

UNIVERSIDAD POLITÉCNICA DE MADRID

Escuela Técnica Superior de Edificación



Gestión y control de los datos BIM a lo largo del Ciclo de Vida en Proyectos de Edificación

TESIS DOCTORAL

Presentada para optar al título de Doctor por:

Samuel Jesús Triana González

Arquitecto

Madrid, 2025



UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Edificación

Doctorado en Innovación Tecnológica en Edificación

Gestión y control de los datos BIM a lo largo del Ciclo de Vida en Proyectos de Edificación

TESIS DOCTORAL

Presentada para optar al título de Doctor por:

Samuel Jesús Triana González

Arquitecto

Bajo la dirección de:
Dr. Rafael Vicente Lozano Díez

Madrid, 2025

Título: Gestión y control de los datos BIM a lo largo del Ciclo de Vida en Proyectos de Edificación

Autor: Samuel Jesús Triana González

Programa de Doctorado: Innovación Tecnológica en Edificación

Dirección de tesis: Dr. Rafael Vicente Lozano Díez, Profesor Doctor, ETSEM (director)

Agradecimientos

Durante los últimos años como doctorando, son muchas las personas que me han ayudado y apoyado para sacar adelante este proyecto de investigación. A todos ellos, deseo expresarles mi más sincero agradecimiento.

Agradecer en primer lugar a mi director de tesis Dr. Rafael Lozano, por su tiempo y dedicación durante todo el proceso de investigación y por compartir su conocimiento y experiencia en todo momento.

En segundo lugar, a toda mi familia y amigos por su apoyo incondicional, por su comprensión y cariño año tras año. En especial, a mis padres por servir de inspiración y por compartir todas sus enseñanzas de vida.

Por último, me gustaría expresar mi agradecimiento a todos los compañeros, profesionales y empresas que me han apoyado durante la ejecución y el desarrollo de la tesis, tanto a nivel personal como profesional.

Abstract

Over the last century, the construction sector has not experienced significant advancements in terms of productivity and efficiency, especially when compared to other industries. Despite efforts to introduce key innovations in project planning and management, no notable progress has been achieved. Moreover, daily practices continue to show serious difficulties in meeting deadlines and budget forecasts, indicating that many effective methods are not being fully utilized. In addition, the lack of standardized protocols in digital procedures related to construction and asset management makes it difficult to create, track, and reuse information efficiently.

However, in recent decades, the landscape has started to change due to the emergence of new technologies and tools designed to optimize construction methods and improve the organization of operational procedures. In this context, interoperability between different environments and data sources emerges as a key factor to improve coordination and project management, which in turn allows for reduced time and costs throughout the entire construction asset lifecycle.

This research addresses the current state of the Building Information Modeling (BIM) methodology to seek greater integration and unlock its full potential. To this end, an extensive bibliographic study and in-depth analysis of construction data structuring and types have been carried out, identifying potential areas for improvement within planning and asset management processes. The goal is to foster overall productivity growth in the construction industry, particularly through significant increases in efficiency in digital environments and the agile and optimal use of 3D information models.

To achieve this goal, the study establishes a link between the BIM methodology and its connectivity with relational databases, presenting a large-scale framework that enables new workflows and practical knowledge for all those involved in building assets. The work positions data connections as fundamental, aligning digital technologies with the extraction of valuable information, and advocates for a strategic approach to digital continuity.

Based on the results, an analysis was conducted that shows a significant increase in the performance of processes and information management, as well as improved reliability in processes and collaboration.

The main contributions are related to the understanding of the global structuring of information involved in building asset management and its potential interconnections. Additionally, original ideas are provided on the challenges and opportunities of using BIM information models in synchronization with relational databases, so that data can be shared from multiple environments and accessed by different specialists.

Resumen

A lo largo del último siglo, el sector de la construcción no ha experimentado avances significativos en términos de productividad y eficiencia, especialmente en comparación con otras industrias. A pesar de los esfuerzos por introducir innovaciones clave en la planificación y gestión de proyectos, no se ha alcanzado un progreso notable. Además, la práctica diaria sigue evidenciando serias dificultades para cumplir con los plazos y costos previstos, lo que indica que muchos métodos efectivos no se están aprovechando en su totalidad. A esto se suma la falta de protocolos estandarizados en los procedimientos digitales relacionados con la construcción y gestión de activos, lo que dificulta la creación, seguimiento y reutilización eficiente de la información.

Sin embargo, en las últimas décadas, el panorama ha comenzado a cambiar gracias a la aparición de nuevas tecnologías y herramientas diseñadas para optimizar los métodos de construcción y mejorar la organización de los procedimientos operacionales. En este contexto, la interoperabilidad entre diferentes entornos y fuentes de datos se presenta como un factor clave para mejorar la coordinación y gestión de los proyectos, lo que, a su vez, permite reducir los tiempos y costos de todo el ciclo de vida del activo de construcción.

La presente investigación aborda el estado actual de la metodología BIM (Building Information Modelling) para buscar una mayor integración y poder obtener su mayor potencial. Con ese fin, se ha realizado un amplio estudio bibliográfico y un análisis a fondo de la estructuración y tipos de datos de construcción, en el que se han identificado potenciales áreas de mejora dentro de los procesos de planificación y gestión de activos. Con ello, se pretende un crecimiento general de la productividad de la industria de la construcción en su conjunto, particularmente mediante aumentos significativos de la eficiencia en los trabajos dentro de entornos digitales y el uso ágil y óptimo de modelos tridimensionales de información.

Para llegar a ese objetivo, el estudio establece un vínculo entre la metodología BIM y su conectividad con bases de datos relacionales, presentando un marco de trabajo a gran escala que posibilita nuevos procesos de trabajo y conocimientos prácticos para todos los involucrados en los activos de edificación. El trabajo posiciona las conexiones entre datos como fundamentales, alineando las tecnologías digitales con la extracción de información de valor, y aboga por un enfoque estratégico de continuidad digital.

Sobre la base de los resultados, se realizó un análisis que demuestra un aumento significativo en el rendimiento de los procesos y gestión de la información, así como de la fiabilidad en cuanto a procesos y colaboración.

Las principales aportaciones están relacionadas con la comprensión de la estructuración global de la información involucrada en la gestión de activos de edificación y sus posibles interconexiones. Adicionalmente, se aportan ideas originales sobre los retos y oportunidades de utilizar modelos de información BIM de forma sincronizada con bases de datos relacionales de tal forma que los datos sean compartidos desde múltiples entornos y accesibles por diferentes especialistas.

Tabla de contenido

Agradecimientos	II
Abstract	III
Resumen	IV
Tabla de contenido	V
Lista de Figuras	VIII
Lista de Tablas	XIV
Abreviaturas y Acrónimos.....	XVII
Estructura de la tesis	XIX
CAPÍTULO 1: HIPÓTESIS, OBJETIVOS Y METODOLOGÍA	1
1.1. Visión general de la investigación	2
1.1.1. Problemática y justificación	3
1.1.2. Planteamiento inicial	3
1.1.3. Motivación y objetivos	4
1.1.4. Alcance y limitaciones.....	5
1.1.5. Aplicaciones utilizadas.....	6
1.2. Metodología	7
CAPÍTULO 2: ANTECEDENTES Y ESTADO DEL ARTE	10
2.1. Antecedentes	11
2.1.1. Digitalización.....	11
2.1.2. Building Information Modeling (BIM).....	19
2.1.3. Gestión de la información BIM	32
2.1.4. Estructura de la información BIM.....	46
2.2. Estado del arte	64
2.2.1. Avances y situación actual de BIM.....	64
2.2.2. OpenBIM	69
2.2.3. Industry Foundation Classes	81
CAPÍTULO 3: DESARROLLO DE LA INVESTIGACIÓN	98
3.1. Búsqueda científica.....	99
3.1.1. Gestión de datos y Bases relacionales SQL.....	99
3.1.2. Análisis del esquema IFC	119
3.1.3. Manipulación de archivos IFC	139
3.1.4. Conexión de bases de datos y modelos BIM	152
3.2. Experiencia directa de expertos BIM	161
3.2.1. Planteamiento inicial	161

3.2.2.	Resultados de la encuesta	164
3.2.3.	Conclusiones de la encuesta.....	174
3.3.	Propuesta de sistematización	175
3.3.1.	Base de datos global.....	175
3.3.2.	Ejes estructurantes de datos.....	177
3.3.3.	Conectividad de datos BIM	179
3.3.4.	Definición de niveles de análisis	187
3.3.5.	Gestión de nuevos roles y responsabilidades.....	192
3.4.	Definición de procesos de la sistematización.....	195
3.4.1.	Creación de base de datos relacional	195
3.4.2.	Integración de datos en modelos BIM.....	196
3.4.3.	Controles de calidad	198
3.4.4.	Consolidación de los datos en la BBDD	199
3.5.	Desarrollo de casos de uso	202
3.5.1.	Recolección y estructuración de datos	202
3.5.2.	Configuración de la base de datos.....	206
3.5.3.	Administración de la base de datos	212
3.5.4.	Datos contractuales y del CDE	216
3.5.5.	Integración de atributos en modelos BIM	220
3.5.6.	Definición de valores de atributos	237
3.5.7.	Configuración de exportación IFC	246
3.5.8.	Extracción de datos IFC.....	252
3.5.9.	Revisiones y chequeos de calidad.....	299
3.5.10.	Consolidación y analítica de datos.....	322
CAPÍTULO 4: ANÁLISIS E INTERPRETACIÓN DE RESULTADOS		338
4.1.	Resultados obtenidos	339
4.2.	Limitaciones de la investigación.....	353
4.3.	Implicaciones teóricas.....	353
4.4.	Implicaciones prácticas.....	354
4.5.	Grado de validez y fiabilidad del estudio	354
CAPÍTULO 5: CONCLUSIONES Y FUTURAS INVESTIGACIONES		355
5.1.	Conclusiones parciales.....	356
5.2.	Conclusiones generales.....	358
5.3.	Futuras líneas de investigación.....	360
Referencias bibliográficas		362
Anexos		370

A.	Comparativa de aplicaciones IFC.....	371
B.	Tipos de datos	372
C.	Comparativa de Sistemas de Bases de Datos	373
D.	Comparativa de Cantidades IFC.....	374
E.	Revisiones en entornos BIM.....	378
F.	Plantilla cuestionario BIM	385

Lista de Figuras

Figura 1.2.1. Metodología de investigación	8
Figura 2.1.1. Digitalización del sector AECO. Fuente* (McKinsey G. Institute)	13
Figura 2.1.2. Innovaciones tecnológicas. Fuente* (Analytical Report)	15
Figura 2.1.3. Gráfico comparativo del esfuerzo con y sin la metodología BIM	20
Figura 2.1.4. Modelo de información BIM de edificio de viviendas.....	21
Figura 2.1.5. Colaboración BIM. Fuente* (buildingSmart International).....	22
Figura 2.1.6. Continuidad digital durante las fases del ciclo de vida	23
Figura 2.1.7. Evolución del ciclo de vida del activo de edificación	24
Figura 2.1.8. Cambio metodológico: procesos, personas y tecnologías.....	28
Figura 2.1.9. Etapas de madurez BIM. Fuente* (UK BIM Framework)	30
Figura 2.1.10. Usos BIM según categoría y fase de proyecto.....	33
Figura 2.1.11. Estados de la información del CDE. Fuente* (ISO 19650).....	42
Figura 2.1.12. Estructura de la información BIM	46
Figura 2.1.13. Códigos de clasificación según tablas del SCFClass v2.0.	55
Figura 2.1.14. Niveles de información BIM. Fuente* (BIMForum)	57
Figura 2.1.15. Tipos de atributos en entornos de construcción digitales.....	59
Figura 2.1.16. Recursos asociados a modelos BIM	60
Figura 2.2.1. Valoraciones en pliegos de edificación (2017-2023)	66
Figura 2.2.2. Estándares y servicios OpenBIM. Fuente* (buildingSmart).....	70
Figura 2.2.3. Información contenida en archivos BCF.....	73
Figura 2.2.4. Sintaxis, Semántica, Taxonomía y Ontología del IFC.....	81
Figura 2.2.5. Esquema IFC completo vs. definición de subconjunto MVD	92
Figura 3.1.1. Claves primarias y foráneas de bases de datos	105
Figura 3.1.2. Gestión y manipulación de bases de datos SQL	107
Figura 3.1.3. Ciclo de vida de los datos.....	108
Figura 3.1.4. Gestión de la calidad de los datos	114
Figura 3.1.5. Vista 3D vs. Formato texto del IFC	119
Figura 3.1.6. Definiciones principales del esquema IFC 4.3.....	120

Figura 3.1.7. Estructura espacial IFC. Fuente* (ISO 16739)	124
Figura 3.1.8. Ejemplos de SweptSolid y FacetedBrep del IFC.....	130
Figura 3.1.9. Relaciones del esquema IFC. Fuente* (ISO 16739).....	132
Figura 3.1.10. Relaciones de un equipo mecánico dentro del IFC	134
Figura 3.1.11. Encabezado y cuerpo del archivo IFC	139
Figura 3.1.12. Relaciones de un muro desde IfcQuickBrowser.....	145
Figura 3.1.13. Herramientas de edición IFC en Blender	147
Figura 3.1.14. Programación visual en Dynamo	148
Figura 3.1.15. Programación con APIs en Visual Studio	149
Figura 3.1.16. Propiedades de Revit en la Microsoft Access	152
Figura 3.1.17. Relaciones de la base de datos exportada desde Revit	153
Figura 3.1.18. Conexión de datos entre Dynamo y MySQL	154
Figura 3.1.19. Inserción de registros con datos BIM en MySQL	155
Figura 3.1.20. Importación de datos de la BBDD en Navisworks.....	156
Figura 3.1.21. Desencadenador y acciones con Workato y ACC Connect	157
Figura 3.2.1. Presentación de encuesta a expertos BIM	163
Figura 3.2.2. Encuesta: Tipos de empresas	164
Figura 3.2.3. Encuesta: Tramos de años de experiencia	164
Figura 3.2.4. Encuesta: Tipos de perfiles profesionales	165
Figura 3.2.5. Encuesta: Exigencia de requisitos de información	165
Figura 3.2.6. Encuesta: Definición de LOIN desde fases iniciales.....	166
Figura 3.2.7. Encuesta: Aspectos relevantes en la gestión BIM	166
Figura 3.2.8. Encuesta: Plantillas de configuración de atributos	167
Figura 3.2.9. Encuesta: Bibliotecas de objetos BIM	167
Figura 3.2.10. Encuesta: Definición de datos y propiedades en el BEP.....	168
Figura 3.2.11. Encuesta: Exigencia de entrega en formatos abiertos.....	168
Figura 3.2.12. Encuesta: Configuración del esquema IFC.....	169
Figura 3.2.13. Encuesta: Conocimientos en formatos openBIM	169
Figura 3.2.14. Encuesta: Controles de calidad en entregables BIM	170
Figura 3.2.15. Encuesta: Controles de calidad en LOIN.....	170

Figura 3.2.16. Encuesta: Controles de calidad en nomenclatura.....	171
Figura 3.2.17. Encuesta: Integración BIM con otras tecnologías.....	171
Figura 3.2.18. Encuesta: Interacción de BIM con BBDD.....	172
Figura 3.2.19. Encuesta: Uso de formatos abiertos como IFC	172
Figura 3.2.20. Encuesta: Trazabilidad de los datos en el ciclo de vida	173
Figura 3.2.21. Encuesta: Conectividad de los datos de la construcción.....	173
Figura 3.3.1. Relación entre BBDD y proyectos BIM.....	176
Figura 3.3.2. Ejes estructurantes de datos de construcción.....	177
Figura 3.3.3. Formatos de trabajo en Autodesk Revit.....	182
Figura 3.3.4. Formatos de trabajo en Navisworks Manage	183
Figura 3.3.5. Formatos de trabajo en BIMCollab Zoom	184
Figura 3.3.6. Formatos de trabajo en Bexel Manager	185
Figura 3.3.7. Formatos de trabajo en Blender con Bonsai	186
Figura 3.3.8. Definición de niveles de análisis	188
Figura 3.3.9. Propuesta de aplicación de la sistematización.....	192
Figura 3.4.1. Proceso de creación y gestión de la BBDD SQL.....	195
Figura 3.4.2. Proceso de integración de datos en modelos BIM.....	196
Figura 3.4.3. Proceso para controles de calidad	198
Figura 3.4.4. Proceso de consolidación y visualización de datos	200
Figura 3.5.1. Tabla con relaciones entre datos de construcción.....	204
Figura 3.5.2. Creación de relaciones de datos desde Visual Basic.....	204
Figura 3.5.3. Generación de tabla de relaciones por IDs.....	205
Figura 3.5.4. Gestión de usuarios y permisos de la BBDD	206
Figura 3.5.5. Esquema de datos de gestión de la organización.....	207
Figura 3.5.6. Importación de datos tabulares a la BBDD SQL.....	209
Figura 3.5.7. Configuración de la base de datos SQL.....	211
Figura 3.5.8. Creación de vistas dentro de la base de datos	214
Figura 3.5.9. Disparador para el control de registros duplicados	215
Figura 3.5.10. Tablas relacionadas con el Plan de Ejecución BIM	216
Figura 3.5.11. Relación entre tabla LOD y tabla de roles BIM.....	217

Figura 3.5.12. Tabla vinculada al documento del BEP	218
Figura 3.5.13. Importación de registro de archivos de ACC en la BBDD	219
Figura 3.5.14. Creación de archivo de parámetros compartidos.....	222
Figura 3.5.15. Creación de parámetros en familia de Revit.....	225
Figura 3.5.16. Creación de parámetros en Revit con XML	228
Figura 3.5.17. Creación de parámetros en Revit con Dynamo.....	231
Figura 3.5.18. Creación de biblioteca de propiedades IFC.....	233
Figura 3.5.19. Incorporación de propiedades en modelo con Blender	235
Figura 3.5.20. Comprobación de nuevas propiedades en BIMVision.....	236
Figura 3.5.21. Tabla de funciones de elementos del SCFClass v02	238
Figura 3.5.22. Asignación de valores a objetos desde Classification Manager	239
Figura 3.5.23. Asignación de valores a proyecto desde Classification Manager ...	239
Figura 3.5.24. Entradas para asignación de valores dependientes.....	241
Figura 3.5.25. Definición de valores dependientes según clasificación.....	243
Figura 3.5.26. Valores rellenados automáticamente desde Dynamo	244
Figura 3.5.27. Valores rellenados automáticamente desde Blender.....	245
Figura 3.5.28. TXT para exportación de propiedades IFC	248
Figura 3.5.29. Exportación de propiedades IFC desde Revit	249
Figura 3.5.30. Configuración de exportación de clasificaciones en Revit	250
Figura 3.5.31. Proceso de generación de los modelos openBIM	251
Figura 3.5.32. Tipos de muros básicos y complejos	252
Figura 3.5.33. Esquema jerárquico de instancia de muro.....	253
Figura 3.5.34. Relación de muro con su nivel asociado	253
Figura 3.5.35. Esquema jerárquico de tipos de muro.....	255
Figura 3.5.36. Relaciones de tipo con instancias de muro.....	255
Figura 3.5.37. Tipos predefinidos de muros IFC	256
Figura 3.5.38. Definición de tipo predefinido personalizado.....	256
Figura 3.5.39. Relaciones de suelo y muros conectados	258
Figura 3.5.40. Relaciones entre muros conectados.....	258
Figura 3.5.41. Relación de espacio con muros perimetrales.....	259

Figura 3.5.42. Composición de materiales de muro	260
Figura 3.5.43. Estructura de capas de muros IFC	260
Figura 3.5.44. Relación de muro con capas. Fuente* (ISO 16739).....	261
Figura 3.5.45. Descomposición de subelementos de muro	264
Figura 3.5.46. Relación del muro con sus capas IFC.....	265
Figura 3.5.47. Atributos principales de identificación de muro	266
Figura 3.5.48. Relación de propiedades y cantidades. Fuente* (ISO 16739)	268
Figura 3.5.49. Definición de múltiples sistemas de clasificación.....	273
Figura 3.5.50. Descomposición de subelementos de muro cortina.....	274
Figura 3.5.51. Puertas y ventanas en muro cortina	275
Figura 3.5.52. Tipos de relaciones de muros con huecos	276
Figura 3.5.53. Relación de muro y puerta con hueco.....	277
Figura 3.5.54. Relación de muro, puerta y hueco. Fuente* (ISO 16739)	277
Figura 3.5.55. Muros y ventanas existentes en la vivienda unifamiliar	279
Figura 3.5.56. Relación de puertas IFC con anfitriones.....	279
Figura 3.5.57. Tipos predefinidos de puertas IFC	281
Figura 3.5.58. Tipos predefinidos de ventanas IFC.....	283
Figura 3.5.59. Relación entre escalera y sus subelementos	284
Figura 3.5.60. Propiedades de tramo de escalera	286
Figura 3.5.61. Cantidades de instancia de armazón estructural	289
Figura 3.5.62. Relaciones entre instancias de objetos y el sistema.....	289
Figura 3.5.63. Atributos de aparatos sanitarios.....	291
Figura 3.5.64. Atributos generales de espacios	293
Figura 3.5.65. Atributos de acabados de espacios	295
Figura 3.5.66. Revisiones de sistemas de clasificación.....	300
Figura 3.5.67. Rutina para generar vista 3D con objetos incorrectos.....	301
Figura 3.5.68. Nueva vista 3D con objetos incorrectos en rojo.....	303
Figura 3.5.69. Creación de especificaciones IDS	305
Figura 3.5.70. Ejecución de chequeos de especificaciones en BIMCollab	306
Figura 3.5.71. Exportación de informe PDF con revisiones BIM.....	309

Figura 3.5.72. Visor de geometría desde Jupyter Notebook.....	312
Figura 3.5.73. Gestión BCF dentro de BIMCollab Zoom.....	312
Figura 3.5.74. Automatización en la validación IFC con Dynamo.....	313
Figura 3.5.75. Informe de calidad desde Model Checker de Revit.....	320
Figura 3.5.76. Procesos de control de calidad de los datos BIM.....	321
Figura 3.5.77. Importación XML de búsquedas en Navisworks	322
Figura 3.5.78. Addin de Navisworks para creación de conjuntos	324
Figura 3.5.79. Tabla de datos desde plantilla XML en Navisworks	326
Figura 3.5.80. Tabla de datos desde plantilla XML en BIMCollab Zoom	327
Figura 3.5.81. SmartViews desde plantilla XML en BIMCollab Zoom.....	329
Figura 3.5.82. Plantilla de conjuntos de objetos XLSX en Bexel	331
Figura 3.5.83. Plantilla de clasificaciones XLSX importadas en Bexel	332
Figura 3.5.84. Tabla de datos a exportar desde Navisworks	333
Figura 3.5.85. Integración de los datos del proyecto en MariaDB.....	335
Figura 3.5.86. Generación de cuadros de mando del proyecto	336
Figura 3.5.87. Proceso de consolidación y analítica de datos	337
Figura 4.1.1. Incremento exponencial en la complejidad del proyecto.....	341
Figura 4.1.2. Ahorro de horas estimadas por proceso y proyecto.....	349
Figura 4.1.3. Propuesta de sistematización para la gestión de activos.....	352

Lista de Tablas

Tabla 1.1. Aplicaciones y herramientas utilizadas.....	6
Tabla 2.1.1. Roles y responsabilidades BIM.....	37
Tabla 3.1.1. Tipos de bases de datos	104
Tabla 3.1.2. Atributos IFC. Fuente* (ISO 16739).....	127
Tabla 3.1.3. Tipos de cantidades IFC. . Fuente* (ISO 16739).....	138
Tabla 3.1.4. Módulos IfcOpenShell. Fuente (OSArch Wiki).....	142
Tabla 3.3.5. Ejemplo de requisitos de información.....	179
Tabla 3.3.6. Definición de nuevos roles de la sistematización	193
Tabla 3.5.7. Extracto de registros de la tabla de usos BIM.....	203
Tabla 3.5.8. Relación entre clasificaciones con clases y tipos IFC	205
Tabla 3.5.9. Campos generados para el control de registros.....	212
Tabla 3.5.10. Unificación de datos de propiedades en tabla conjunta	213
Tabla 3.5.11. Vista LOD final a incorporar en el BEP	217
Tabla 3.5.12. Extracto de relaciones de valores de parámetros	242
Tabla 3.5.13. Atributos IFC dentro de Autodesk Revit	246
Tabla 3.5.14. Propiedades generales del modelo BIM.....	247
Tabla 3.5.15. Extracto de ejemplares de muro de planta baja	254
Tabla 3.5.16. Extracto de tipos de muro predefinidos y personalizados	257
Tabla 3.5.17. Extracto de materiales en capas de muro.....	262
Tabla 3.5.18. Extracto de subelementos de muros	265
Tabla 3.5.19. Extracto de acabados y molduras de muro	266
Tabla 3.5.20. Extracto de atributos de instancia de muro	267
Tabla 3.5.21. Extracción de atributos desde GUID	268
Tabla 3.5.22. Cantidades de instancia de muro IFC4	269
Tabla 3.5.23. Cantidades de instancia de muro IFC2x3	269
Tabla 3.5.24. Extracto de cantidades de muros IFC2x3.....	270
Tabla 3.5.25. Propiedades del pset común de muro IFC2x3	271
Tabla 3.5.26. Extracto de propiedades de muro IFC2x3	271

Tabla 3.5.27. Extracto de recuento de propiedades de muros	272
Tabla 3.5.28. Clasificaciones definidas para instancia de muro	274
Tabla 3.5.29. Subelementos que componen un muro cortina	275
Tabla 3.5.30. Subelementos que componen un muro cortina	275
Tabla 3.5.31. Extracto de ejemplares de puertas y ventanas.....	278
Tabla 3.5.32. Obtención de tipos de puertas en IFC2x3.....	280
Tabla 3.5.33. Obtención de tipos de puertas en IFC4.....	281
Tabla 3.5.34. Obtención de tipos de puertas en IFC4.3.....	282
Tabla 3.5.35. Obtención de tipos en IFC4.3.....	282
Tabla 3.5.36. Obtención de tipos de ventanas en IFC4	283
Tabla 3.5.37. Obtención de subelementos de escaleras	285
Tabla 3.5.38. Propiedades de tramos de escaleras IFC	286
Tabla 3.5.39. Propiedades de armazones estructurales	287
Tabla 3.5.40. Cantidades de armazones estructurales.....	288
Tabla 3.5.41. Extracto de sistemas de instalaciones	290
Tabla 3.5.42. Extracto de instancias de apartatos sanitarios	292
Tabla 3.5.43. Extracto de instancias de segmentos de tuberías.....	293
Tabla 3.5.44. Extracto de instancias de espacios del edificio	294
Tabla 3.5.45. Extracto con información de acabados de espacios	295
Tabla 3.5.46. Datos de proyecto extraídos del modelo IFC	297
Tabla 3.5.47. Datos del edificio extraídos del modelo IFC	298
Tabla 4.1.1. Información de partida para la BBDD	339
Tabla 4.1.2. Recuento de definiciones de proyectos de investigación.....	341
Tabla 4.1.3. Complejidad operativa dentro de modelos BIM	345
Tabla 4.1.4. Resultados de conectividad entre BBDD y software	346
Tabla 4.1.5. Estimación de ahorro en horas de trabajo.....	348
Tabla 4.1.6. Análisis comparativo de detección de errores.....	350
Tabla 4.1.7. Análisis comparativo de la propuesta de sistematización.....	351
Tabla A.1. Comparativa de aplicaciones IFC	371
Tabla B.2. Tipos de datos	372

Tabla C.3. Comparativa de Sistemas de Bases de Datos	373
Tabla D.4. Comparativa de Cantidades IFC	377
Tabla E.5. Revisiones del Entorno Común de datos.....	378
Tabla E.6. Revisiones de documentación BIM.....	379
Tabla E.7. Revisiones del Plan de Ejecución BIM.....	379
Tabla E.8. Revisiones de incidencias	380
Tabla E.9. Revisiones de calidad de los modelos BIM.....	382
Tabla E.10. Revisiones de referencias externas	383
Tabla E.11. Revisiones de planos obtenidos de modelos	383
Tabla E.12. Revisiones de presupuestos asociados a modelos	383
Tabla E.13. Revisiones de planificaciones asociadas a modelos.....	384
Tabla E.14. Revisiones de certificaciones de obra	384
Tabla E.15. Revisiones de modelos As-Built.....	384

Abreviaturas y Acrónimos

AECO	Architecture, Engineering, Construction and Operations <i>(Arquitectura, Ingeniería, Construcción y Operaciones)</i>
AIM	Asset Information Model <i>(Modelo de Información del Activo)</i>
AIR	Asset Information Requirements <i>(Requisitos de Información del Activo)</i>
API	Application Program Interface <i>(Interfaz de Programación de Aplicaciones)</i>
BCF	BIM Collaboration Format <i>(Formato de Colaboración BIM)</i>
BEP	BIM Execution Plan <i>(Plan de Ejecución BIM)</i>
BIM	Building Information Modelling <i>(Modelado de Información de la Construcción)</i>
CAD	Computer Aided Design <i>(Diseño Asistido por Ordenador)</i>
CDE	Common Data Environment <i>(Entorno Común de Datos)</i>
COBie	Construction Operations Building Information Exchange <i>(Intercambio de Datos de Construcción y Operaciones de Activos)</i>
EIR	Exchange Information Requirements <i>(Requisitos de Intercambio de Información)</i>
FM	Facility Management <i>(Gestión de Activos)</i>
GMAO	Computerized Maintenance Management System <i>(Sistema Informatizado de Gestión de Mantenimiento)</i>
GUID	Global Unique Identifier <i>(Identificador Único Global)</i>
HVAC	Heating, Ventilating and Air Conditioning <i>(Calefacción, Ventilación y Aire Acondicionado)</i>

IDM	Information Delivery Manual <i>(Manual de Entrega de la Información)</i>
IFC	Industry Foundation Classes <i>(Clases Fundamentales de la Industria)</i>
IFD	Information Framework Dictionary <i>(Diccionario del Marco de Información)</i>
IPD	Integrated Project Delivery <i>(Entrega Integrada de Proyecto)</i>
LOIN	Level of Information Need <i>(Nivel de Información Necesario)</i>
MEA	Model Element Author <i>(Autor de Elemento de Modelo)</i>
MEP	Mechanical, Electrical and Plumbing <i>(Mecánica, Electricidad y Fontanería)</i>
MVD	Model View Definition <i>(Definición de Vista de Modelo)</i>
OIR	Organizational Information Requirements <i>(Requisitos de Información de la Organización)</i>
PIM	Project Information Model <i>(Modelo de Información de Proyecto)</i>
PIR	Project Information Requirements <i>(Requisitos de Información del Proyecto)</i>
PSET	Property set <i>(Conjunto de Propiedades)</i>
QTO	Quantity Take-Off <i>(Conjunto de Cantidades)</i>
SQL	Structures Query Language <i>(Lenguaje de Consulta de Estructuras)</i>
UPM	Universidad Politécnica de Madrid

Estructura de la tesis

La estructura de la tesis se organiza en cinco capítulos principales que organizan el desarrollo de la investigación:

El capítulo 1 “Hipótesis, Objetivos y Metodología” establece las bases conceptuales y metodológicas de la investigación. Presenta una visión general que expone la problemática, los objetivos principales y específicos, y los alcances y limitaciones del estudio. Además, describe la metodología utilizada, incluyendo una propuesta para evaluar los resultados.

El capítulo 2 “Antecedentes y Estado del Arte” se centra en una revisión bibliográfica y conceptual. Describe el contexto histórico y técnico de la investigación, destacando la digitalización y la evolución del BIM. También analiza avances recientes en la gestión de información y estructuras de datos, identificando vacíos de conocimiento y justificando la relevancia del estudio.

El capítulo 3 “Desarrollo de la Investigación” describe el proceso práctico y teórico llevado a cabo, en el que se desarrolla el proceso de recopilación de información que incluye la construcción de bases de datos relacionales y su conexión con modelos BIM. También se analiza cómo estas herramientas se integran para resolver los problemas planteados. En este capítulo se expone la propuesta de sistematización que plantea una solución para la integración y gestión de datos BIM incluyendo el desarrollo de plantillas configuraciones automatizadas y roles específicos para optimizar los flujos de trabajo en proyectos BIM.

El capítulo 4 “Análisis e Interpretación de Resultados” expone los hallazgos de la investigación, analizando los datos obtenidos y su alineación con los objetivos planteados. También aborda las limitaciones del estudio y sugiere formas de superarlas con diferentes alternativas. Finalmente, evalúa las implicaciones teóricas y prácticas de los resultados en proyectos reales del sector.

El capítulo 5 “Conclusiones y Futuras Investigaciones” ofrece una reflexión final. Resume las principales aportaciones del estudio y propone futuras líneas de investigación para superar las limitaciones identificadas. Concluye destacando la contribución de los hallazgos al avance del conocimiento en el ámbito de la metodología BIM y de la gestión de datos en construcción.

Nota sobre autoría de figuras y tablas

Salvo que se indique expresamente lo contrario en el título o pie de cada figura o tabla, todos los recursos insertados en el presente documento han sido elaborados por el autor. Las figuras o tablas tomadas o adaptadas de otras fuentes se citan adecuadamente en cada caso, incluyendo su referencia correspondiente.

Asimismo, las figuras cuyo título incluye un asterisco (*) han sido modificadas o adaptadas a partir de la fuente original. Estas adaptaciones pueden incluir cambios en el diseño, reorganización de elementos, estilo gráfico o traducción del contenido, manteniéndose siempre la atribución adecuada al autor original.

CAPÍTULO 1: HIPÓTESIS, OBJETIVOS Y METODOLOGÍA

1.1. Visión general de la investigación

A lo largo de las últimas décadas, la digitalización y la aparición de nuevas tecnologías en el sector AECO (Bosch-Sijtsema, Petra et al., 2021) ha permitido a los profesionales de la construcción obtener diferentes beneficios, como ahorrar tiempo durante el desarrollo de proyectos, aumentar la productividad, simplificar los métodos de trabajo o mejorar la calidad de los documentos (Aghimien et al., 2018). Estas tecnologías y procedimientos de trabajo innovadores (Pérez Sánchez et al., 2017) están rompiendo paulatinamente los silos de información y permitiendo la continuidad digital a través de múltiples iniciativas y nuevos enfoques disruptivos (Annunen et al., 2024), lo que garantiza que la información se mantenga coherente, fiable y única al ser generada y actualizada por especialistas.

El uso de la metodología Building Information Modeling (BIM) en proyectos de construcción ha mejorado la integración de procesos y la coordinación entre todos los participantes al basarse en una mejor colaboración y un intercambio óptimo de información (Lou et al., 2020) dentro de un entorno de datos común, si bien ello conlleva un estudio de implementación complejo que requiere de estrategias adecuadas que permitan incorporar todas las actividades y procesos del activo (Olugboyega & Windapo, 2019). A esto se suman múltiples barreras que existen en el sector que impiden una adopción total de la metodología, como la falta de conocimiento de BIM dentro del sector (Fountain & Langar, 2018), la complejidad de la gestión de los procesos de contratación (Matos et al., 2024), factores tecnológicos, retos financieros o nuevas implicaciones legales (Toyin & Mewomo, 2021).

Desde hace algunos años, asociaciones como BuildingSmart promueven la aplicación de la metodología BIM, así como el uso del estándar Industry Foundation Classes (IFC), que establece una estandarización y clasificación de los modelos de información de la construcción. También en el ámbito científico se han llevado a cabo diferentes investigaciones para integrar nuevos esquemas y relaciones de datos dentro del estándar (Okonta et al., 2024), relacionados con el análisis y la eficiencia energética (Ramaji, I. et al., 2020), la automatización en los procesos de clasificación de objetos (Wu, 2019), las integraciones de datos BIM-GIS (Cepa et al., 2023) o las definiciones en la gestión del tiempo de construcción (Yang, 2021).

Sin embargo, existen diferentes limitaciones con el trabajo de estándares abiertos, especialmente con archivos IFC (Lange et al., 2018), dado que no todas las aplicaciones y entornos permiten trabajar con todos los recursos, relaciones y entidades que contiene el estándar, no siempre es posible intercambiar toda la información necesaria entre equipos multidisciplinares y no todos los elementos constructivos quedan modelados dentro del modelo virtual.

1.1.1. Problemática y justificación

Una de las principales dificultades de la industria de la construcción en la actualidad es la falta de coordinación y colaboración entre los agentes intervinientes del proyecto. Son pocas las veces en las que los datos de construcción que se generan en las diferentes fases de un proyecto se reutilizan y son retomadas por los posteriores equipos de trabajo. Concretamente en proyectos de edificación, existe una gran complejidad debida a la coexistencia de múltiples disciplinas, muchas veces en un mismo tiempo y espacio y siendo dependientes unas de otras en gran medida (al Omari et al., 2023).

En la actualidad, aunque empieza a existir un cierto manejo en el uso de la metodología BIM, aún existen “gaps” o desconexiones que limitan el potencial que permite la aplicabilidad de este tipo de procesos. Esto es en gran parte debido a la fragmentación de la industria y al recelo de compartir la información generada (Toyin & Mewomo, 2021).

A raíz de toda esta problemática, se plantea un estudio exhaustivo de los flujos y conexiones existentes de trabajo para generar nuevos procesos de sistematización y estandarización de la información que tiendan a mejorar los flujos de trabajo y alinear el proyecto con las bases de la metodología BIM y de la digitalización de la construcción.

1.1.2. Planteamiento inicial

El planteamiento inicial se basa en la búsqueda de una forma eficiente de aprovechar diferentes herramientas y tecnologías emergentes para integrarlas en los procesos Building Information Modeling (BIM) actuales. Esta integración permitirá enriquecer y acelerar los procesos digitales de construcción, optimizando el manejo de la información y su interoperabilidad entre sistemas.

Uno de los pilares fundamentales de este planteamiento es dar prioridad al dato, poniendo en primer plano la gestión de la información y la conectividad de los datos de construcción entre agentes, fases y especialidades técnicas. Esto permite reducir los errores manuales en la generación y gestión de la información, mejorar la eficiencia en los procesos de diseño y garantizar datos más precisos y estructurados. Asimismo, se promueve una mayor colaboración entre los equipos de trabajo mediante el uso de herramientas digitales colaborativas, lo que facilita una mejor sincronización y trazabilidad de la información.

Para materializar toda esta estrategia, la investigación plantea el desarrollo de un marco de trabajo coordinado que integre la información BIM con otras fuentes de datos de construcción, facilitando su conexión a través de procesos estandarizados y herramientas digitales. Esta interconectividad permitirá mejorar la interoperabilidad entre sistemas, optimizar la gestión de la información constructiva y reducir la fragmentación que históricamente ha afectado a la industria de la construcción.

1.1.3. Motivación y objetivos

La tesis actual tiene como objeto de estudio el enfoque que deben seguir las empresas del sector de la construcción para incorporarse de forma eficiente a los requerimientos actuales que exigen los nuevos procesos de gestión de datos, potenciando el uso de la metodología BIM. Con ello, se pretende analizar los recursos y conocimientos de diferentes perfiles del sector y, para cada caso de estudio, establecer las nociones básicas, conocimientos necesarios y métodos de aplicación específicos que deben seguir para poder alinearse con esta nueva filosofía de trabajo y, por ende, no quedar obsoletos en el mercado actual.

Para conseguir este objetivo, será fundamental el estudio de los tipos de datos que aporta cada empresa al proyecto, la/s fase/s en las que interviene, la/s especialidad/es que trabaja, los usos BIM que pueda desarrollar, la estructura organizacional de la empresa y las responsabilidades actuales del personal, entre otros.

Una vez finalizado el estudio, se propondrán nuevas alternativas y sistemas de implementación que permitan un cambio metodológico progresivo en el que, poco a poco, cualquier empresa pueda establecer nuevos flujos de trabajo a nivel interno, sepa cómo compartir la información con el resto de equipos y pueda colaborar de forma eficiente, aportando la cantidad justa y necesaria de información en cada etapa del proyecto.

Para ello, durante el desarrollo de la investigación se buscarán una serie de objetivos específicos los cuales se enumeran a continuación:

- Definir un nuevo marco de sistematización para las empresas del sector.
- Enlazar la metodología BIM con nuevos procedimientos de trabajo.
- Evaluar los nuevos procesos en diferentes modelos y proyectos de edificación desde el diseño hasta la fase de operaciones para evaluar beneficios.
- Mejorar la transparencia de la comunicación entre el cliente y el equipo de diseño, optimizando el intercambio de información.
- Potenciar el uso de herramientas de revisión automatizada y exhaustiva relativas a calidad y consistencia de la información.
- Reducir los tiempos de trabajo de los equipos, mediante preconfiguraciones y plantillas de trabajo estandarizadas.
- Aplicar los nuevos procesos en diferentes softwares y entornos comerciales, priorizando una solución agnóstica e interoperable.
- Analizar los datos consolidados de construcción a partir de la generación de cuadros de mando interactivos y gráficas de datos.

En conclusión, se busca establecer un sistema interoperable que impulse la generación de datos de calidad por parte de todos los agentes involucrados en el proyecto, que cumpla con los requisitos del cliente y que permita reducir el tiempo de entrega y la probabilidad de retrabajos.

1.1.4. Alcance y limitaciones

El alcance de esta investigación se centra en mejorar la gestión de la información en proyectos de construcción a partir del uso e integración de múltiples tecnologías y metodologías digitales para facilitar la interoperabilidad, el acceso y la gestión centralizada de los datos generados durante el ciclo de vida.

Aunque los datos dentro del entorno BIM pueden ser de diferente índole, clase y función, en el presente estudio se abordan los datos relativos a las entidades de construcción del proyecto, dejando en un segundo plano otros datos generales de carácter contractual o documental, datos de configuraciones de software o plataformas, datos relativos a visibilidad, visualización y gráficos y datos de definiciones anotativas planas o bidimensionales. Además, se prioriza el estudio de los datos alfanuméricos frente a los geométricos, incluyendo datos internos de los propios modelos, datos vinculados y datos asociados desde fuentes externas.

En cuanto al uso de software, la investigación se centra en el desarrollo de procesos a partir de determinadas aplicaciones que cuentan con versiones gratuitas, de estudiante o, en algunos casos, licencias especiales para fines de investigación y que, a su vez, permitan el trabajo con formatos interoperables. Además, se asume que las herramientas de software y los lenguajes de programación a utilizar permiten una integración efectiva, sin mayores restricciones técnicas, dentro del contexto de los sistemas y plataformas de la industria de la construcción. Otras restricciones específicas de las aplicaciones se indican en los apartados de los casos de uso.

También existe una limitación en cuanto a la implantación de la solución en proyectos reales, ya que se han utilizado modelos de proyectos de edificación propios y otros recursos públicos descargados desde plataformas web.

Por último, esta investigación se limita a examinar la conexión de datos BIM con otras fuentes de datos y bases relacionales dentro de un contexto teórico y práctico específico, sin abordar otras posibles integraciones con tecnologías emergentes o sistemas no convencionales que puedan ofrecer otras oportunidades para la gestión de datos en la construcción. Asimismo, queda fuera del alcance de este estudio la implementación en proyectos distintos a edificación, como los de infraestructuras de obras civiles, marítimas, hidráulicas o industriales.

1.1.5. Aplicaciones utilizadas

Durante el desarrollo del proyecto de investigación, se hizo uso de diversas aplicaciones y herramientas digitales que facilitaron el análisis de datos, la gestión de referencias bibliográficas y la redacción del presente documento. El uso combinado de estas herramientas permitió optimizar los procesos de investigación, asegurando rigor metodológico y eficiencia en la recolección, análisis e interpretación de los datos obtenidos. En cuanto a las licencias, la mayoría correspondieron a versiones de prueba o licencias para estudiantes. En algunos casos, se contaba con licencias específicas para fines de investigación, lo que permitió acceder a funcionalidades avanzadas en ciertos programas. A continuación, se detallan las principales y sus respectivas funciones:

Herramientas	Versión	Usos de la herramienta
Microsoft® Word®	24	• Redacción de la investigación
Microsoft® Excel®	24	• Generación de tablas de la investigación • Intercambio de datos BIM y no BIM
Microsoft® Visio®	24	• Esquemas de procesos
Microsoft® PowerPoint®	24	• Gráficos y diagramas
Microsoft® Power BI®	2.14	• Tableros para visualización y análisis de datos
Adobe Acrobat Reader	23	• Visualización y conversión PDF
RefWorks - RCM	24	• Gestión de referencias bibliográficas
Solibri Anywhere	9.13	• Visualización y análisis de IFC
BIMVision	2.27	• Visualización y análisis de IFC
Bexel Manager	24.1	• Federación de modelos BIM • Gestión de datos de construcción
Autodesk Revit	24.2	• Modelado y definición de objetos BIM • Exportaciones/importaciones IFC
Dynamo	2.10	• Procesos de integración de datos en modelos • Procesos de extracción de datos de modelos • Automatizaciones de procesos BIM
Navisworks Manage	24	• Federación de modelos BIM • Gestión de datos de construcción • Complementos de automatización
BIMCollab	7.2	• Modelado de objetos BIM • Exportaciones/importaciones IFC • Definición de información BIM
Blender (con Bonsai)	4.3	• Exportaciones/importaciones IFC • Definición de información BIM
Visual Studio Code	22	• Automatizaciones de procesos BIM • Complementos de automatización
MariaDB (MySQL)	11.6	• Gestión de datos
PhpMyAdmin	4.9	• Gestión de datos
Jupyter NoteBook	7.0	• Automatizaciones de procesos BIM
Google Colaboratory	-	• Automatizaciones de procesos BIM

Tabla 1.1. Aplicaciones y herramientas utilizadas

1.2. Metodología

La presente investigación estudia el desarrollo de una sistematización que permita la gestión de datos en activos edificatorios a partir de la conexión de diferentes software y bases de datos, buscando un mayor control en los procesos de generación, intercambio y consolidación de la información durante todas las fases del activo. Para alcanzar ese objetivo, se utiliza una metodología mixta que combina métodos deductivos e inductivos. El enfoque deductivo se centra en probar hipótesis específicas sobre la precisión y eficiencia de la nueva sistematización a través del desarrollo de casos de uso. Estos casos compararán el desempeño de la nueva sistematización con métodos existentes en términos de precisión, tiempo de uso y facilidad de uso, utilizando métricas obtenidas directamente de los entornos y software utilizados.

Por otro lado, el enfoque inductivo se basará en la observación directa de la nueva sistematización aplicada a diferentes modelos BIM de edificación. A partir de este estudio, se recopilarán datos cualitativos a través de los resultados obtenidos y el análisis de patrones recurrentes. Estos datos permitirán identificar áreas de mejora y desarrollar nuevas teorías sobre la optimización del intercambio de datos. La combinación de estos métodos analiza la efectividad de la nueva propuesta y proporciona una base sólida para su implementación. Para ello, se ha desarrollado un método de investigación que estructura el estudio en tres etapas diferenciadas: el planteamiento inicial; el desarrollo del método; y los hallazgos del método.

Esta metodología de investigación comienza en una primera etapa con la identificación de diferentes problemas de conectividad y control a la hora de gestionar la definición, integración y federación de datos de edificación en entornos y modelos BIM. Esto se debe a que en muchos casos no existe una estrategia clara de flujo de datos ni se cuentan con bases de datos centralizadas de las que se derive o gestione la información de los activos (Lammers & Forth, 2022). Por ello, se propone una estandarización inicial de la información alfanumérica de la organización realizando un estudio detallado de los datos de construcción, analizando fases del activo, usos de aplicaciones BIM, clasificación de objetos, niveles de información necesaria y tipos de datos de construcción, incluyendo ubicación, unidades, materiales, dimensiones, relaciones, capas y subcomponentes, entre otros.

En una segunda etapa, el estudio continúa con una definición general del marco de la solución, estableciendo al mismo tiempo una estrategia para el método de análisis que se utilizará para validar la solución y también el planteamiento de una serie de resultados esperados. La validación de la solución debe ser eficiente y precisa, asegurando que se cumplan los objetivos del proyecto, por lo que el estudio de las automatizaciones es crucial, ya que permite la integración y coordinación de múltiples estándares y fuentes de datos, garantizando que toda la información relevante sea accesible y fácilmente manejable. Python, con su amplia biblioteca de paquetes como IfcOpenShell, proporciona una solución para el desarrollo y ejecución de estas automatizaciones, facilitando la creación de scripts personalizados que puedan manejar tareas específicas de análisis y validación. Durante esta etapa, también se consideran las posibles limitaciones y desafíos asociados con la implementación de automatizaciones entre bases de datos y entornos BIM, y se

realizan pruebas exhaustivas para identificar incompatibilidades de rendimiento que puedan surgir en los diferentes casos de uso propuestos.

Finalmente, en la tercera etapa, se realiza un estudio en el que se analizan los hallazgos encontrados durante el desarrollo del estudio, reflejando resultados, detectando posibles inconsistencias, fallas o desconexiones en los procedimientos de trabajo y generando las conclusiones de la investigación.



Figura 1.2.1. Metodología de investigación

Para la validación de los procesos de sistematización de la investigación, a partir de otros estudios de procedimientos estructurados (Balakrishnan Selvakumaran & Hall, 2022), se ha propuesto la definición de diferentes aspectos de evaluación con el objetivo de comprobar el grado de aplicabilidad de la solución:

- (1) Nivel de complejidad: desafíos que implica llevar a cabo el proceso, incluyendo factores como capacidades técnicas, recursos necesarios y obstáculos potenciales.

- (2) Nivel de automatización: referido al grado en que los procesos, tareas o rutinas se realizan de forma automática, con mínima o nula intervención humana.
- (3) Tiempo de ejecución: el período que tarda el proceso en completarse desde su inicio hasta su fin. Imprescindible para evaluar la eficiencia y el rendimiento de la solución propuesta y para asegurar una experiencia óptima del usuario.
- (4) Facilidad de uso: sencillez y comodidad con la que una persona puede aprender a manejar y utilizar la solución, sin necesidad de asistencia especializada.
- (5) Interoperabilidad: capacidad de trabajar conectado entre diferentes sistemas de datos, independientemente del software o plataformas utilizadas.
- (6) Escalabilidad: posibilidad de ampliar recursos y sistemas para adaptarse a nuevas necesidades o condiciones de la organización.

CAPÍTULO 2: ANTECEDENTES Y ESTADO DEL ARTE

2.1. Antecedentes

2.1.1. Digitalización

La digitalización es el proceso de transformar y adaptar sistemas, procesos y servicios al entorno digital mediante el uso de tecnologías avanzadas. No solo se trata de convertir datos de formato físico a digital, sino también de reestructurar flujos de trabajo, automatizar tareas e integrar plataformas digitales que optimicen la eficiencia, la colaboración y la toma de decisiones. En sectores como la construcción, esto implica la adopción de herramientas y sistemas que permiten gestionar proyectos de manera más eficiente durante todo su ciclo de vida, mejorando la gestión de la información a través de su estandarización, seguridad y reutilización.¹

A menudo, la digitalización se confunde con la digitación, que es un proceso más específico, centrado en la conversión de datos físicos o analógicos a digitales, como el escaneo de documentos o la transcripción manual. A diferencia de la digitalización, que transforma integralmente los procesos y servicios, la digitación se limita únicamente a la conversión de datos sin modificar o mejorar los procesos operativos relacionados.

2.1.1.1. Transformación digital en el sector AECO

La industria de la construcción, históricamente conservadora en la adopción de nuevas tecnologías y metodologías, ha enfrentado dificultades para digitalizarse y optimizar sus procesos. Hasta hace algunos años, el sector no había priorizado el manejo de datos ni fomentado una cultura de intercambio de información que permitiera la conectividad de los datos significativos para una toma de decisiones más informada y colaborativa a lo largo de todo el ciclo de vida del activo. Las últimas innovaciones tecnológicas y metodologías, como el Modelado de Información de la Construcción y la gestión ágil de proyectos, han mejorado algunas tareas específicas, pero hasta el momento no parecen ser suficientes para ofrecer una solución global y transformadora para toda la industria.

En relación con estos nuevos enfoques estratégicos se ha observado que, aunque han mejorado la eficiencia y organización en ciertos aspectos de los proyectos, las soluciones propuestas en muchos casos son limitadas. Existen mejoras puntuales, pero no se termina de lograr una transformación radical que reestructure la industria en su totalidad. El desafío radica en que, aunque la mayoría de los proyectos de construcción son altamente repetitivos, se siguen tratando como casos únicos, lo que impide la estandarización y optimización de los procesos. Esto limita las oportunidades para mejorar la productividad y la eficiencia en todos los proyectos.

También existen complicaciones en la rigidez de las interfaces entre las diferentes fases de un proyecto de construcción y las herramientas de software utilizadas. La falta de integración entre estas fases, junto con la ausencia de un sistema efectivo de retroalimentación de la información desde la construcción hacia la planificación,

¹ UK BIM Framework. Normas, directrices, recursos y protocolos desarrollados por BSI y NIMA para el apoyo en la comprensión de los principios fundamentales de la gestión de la información digital. <https://ukbimframework.org/>

dificulta el análisis de las desviaciones entre lo previsto y lo construido. Este tipo de desconexiones impiden que se logre un perfeccionamiento continuo de los datos de planificación y, por ende, afecta la capacidad para mejorar la precisión de las estimaciones y la calidad de los proyectos en el largo plazo.

Además, y a pesar de los avances en diversas áreas, parece claro que una estrategia integral para la digitalización de la construcción no solo es necesaria, sino crucial.² Esta estrategia debe identificar las raíces de los problemas fundamentales de la industria, tales como la baja productividad, la ineficiencia en los procesos de planificación y ejecución, así como los desafíos relacionados con los cronogramas y los costos.³ Un análisis exhaustivo de las tecnologías, métodos y soluciones de software implementados hasta el momento revela que, aunque existen numerosas herramientas digitales disponibles, muchas de ellas siguen quedándose atrás en cuanto a las posibilidades que podrían ofrecer. Además, es esencial entender los requisitos particulares de cada fase de un proyecto de construcción y cómo interrelacionar la información y los procesos entre ellas para evitar desviaciones en el diseño, el calendario y los costos (Toyin & Mewomo, 2021).

Analizando diferentes fuentes (al Omari et al., 2023; Ershadi & Lijauco, 2024) en las que se estudian las barreras y limitaciones que actualmente enfrenta el sector de la construcción a nivel de digitalización, se han identificado cuatro grandes categorías en las que se pueden agrupar todas estas barreras: organizacionales, tecnológicas, culturales y regulatorias.

En primer lugar, las limitaciones organizacionales son una de las barreras más evidentes. La falta de concienciación sobre la importancia del dato y la falta de gestión documental adecuada dificultan la toma de decisiones basadas en información precisa. A esto se suma la reticencia a realizar grandes inversiones en tecnología debido a la incertidumbre sobre el retorno de inversión (ROI), lo que ralentiza la adopción de nuevas tecnologías en el sector. Además, la rotación constante de los equipos de trabajo genera una falta de inversión en conocimiento y capacitación, lo que impide la continuidad en la gestión digital de la construcción.

Por otra parte, existen las limitaciones tecnológicas, con las que el sector enfrenta una serie de retos relacionados con la dificultad para integrar y actualizar los datos de manera eficiente. Los sistemas actuales no siempre permiten una actualización ágil, lo que puede generar desactualización de la información y errores en la toma de decisiones. Además, el foco en generar nuevos datos en lugar de mejorar los existentes limita el aprovechamiento de la información ya disponible, impidiendo una gestión más estratégica de los datos.

Las limitaciones culturales son otro obstáculo importante para la digitalización en la construcción. La resistencia al cambio se suma a una fuerte dependencia del uso de los métodos tradicionales. Muchos técnicos y directivos muestran escepticismo hacia las nuevas tecnologías, viéndolas como innecesarias o complicadas. Este tipo de mentalidad retrasa la adopción de herramientas digitales y requiere un esfuerzo significativo para cambiar la cultura del sector hacia una mayor apertura a la innovación tecnológica.

² Stimulating favorable investment conditions. European Construction Observatory. 2019.

³ Improving Construction Productivity. McKinsey Global Institute (MGI's). 2017.

Finalmente, las limitaciones regulatorias también juegan un papel crucial en el proceso de digitalización. Las normativas locales y nacionales sobre construcción, seguridad laboral y protección de datos no siempre están alineadas con las tecnologías emergentes. La falta de marcos legales claros para el uso de tecnologías digitales genera incertidumbre entre los profesionales del sector. Además, la ausencia de incentivos gubernamentales para apoyar la adopción de tecnologías digitales en la construcción también limita la capacidad del sector para avanzar en este ámbito.

En comparación con otros sectores industriales, a lo largo del siglo pasado, la industria de la construcción se quedó muy atrás en el desarrollo y la aplicación de logros tecnológicos avanzados. Mientras que numerosas industrias comenzaron rápidamente a adoptar diversas innovaciones, mejoras técnicas y estrategias de gestión, el sector de la construcción no logró seguir esa tendencia. Históricamente, se ha mantenido con un enfoque reticente en cuanto a la incorporación de aplicaciones de estructuras para la gestión, el diseño y la entrega progresivos, excusándose en los argumentos de la producción de un solo elemento, la fabricación artesanal o manual y los ajustes debidos a los continuos cambios de las necesidades de los clientes (Chen, S., 2023; Parusheva & Aleksandrova, 2021; Stepanov et al., 2024).

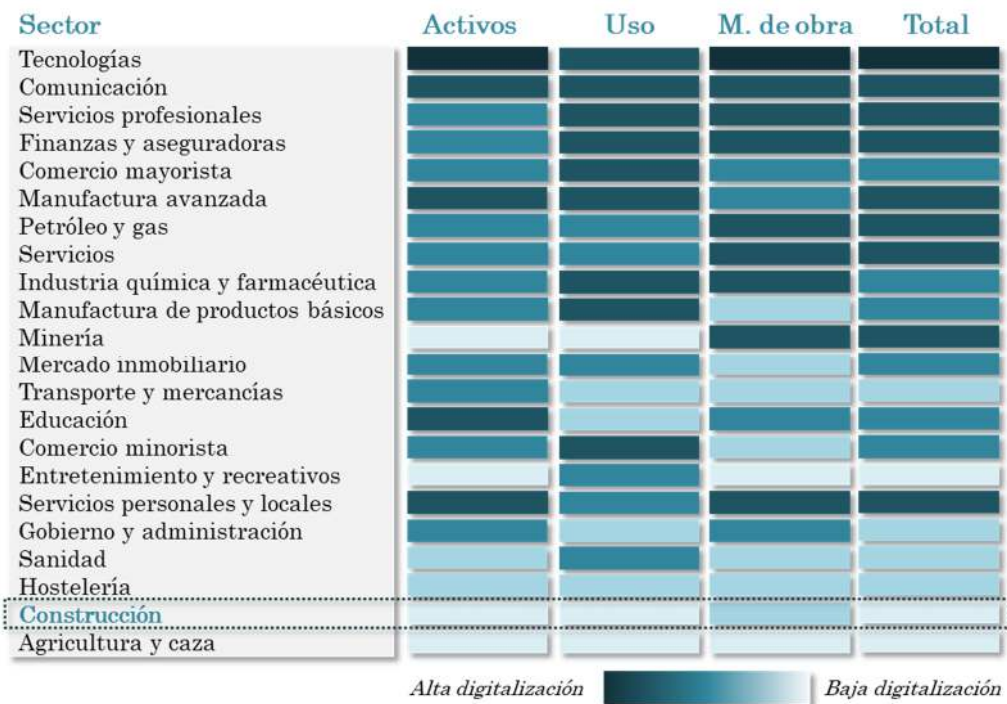


Figura 2.1.1. Digitalización del sector AECO. Fuente* (McKinsey G. Institute)

Además, durante la segunda mitad del siglo pasado, un número creciente de proyectos de construcción se enfrentó a un sistema que incrementaba progresivamente las especificaciones y la complejidad de la planificación y ejecución de las obras. Además, estos proyectos se vieron afectados constantemente por faltas de coordinación entre los diferentes agentes del proyecto y tenían una alta dependencia del resto de los integrantes de la cadena de suministro (Oppong et al., 2017). Por otro lado, las limitaciones de tiempo y presupuesto y los crecientes

desafíos técnicos creaban condiciones difíciles para mantener los proyectos dentro de los plazos previstos (Oesterreich & Teuteberg, 2016). Otras limitaciones relacionadas con los cambios de diseño en las últimas etapas de los proyectos debido a las solicitudes de cambios de los clientes aumentaban aún más el riesgo de aplazamientos y problemas temporales.

La industria de la construcción es de particular importancia para las condiciones económicas internacionales, habiendo proporcionado aproximadamente el 5,4% del producto interno bruto (PIB) europeo y el 4,0% del estadounidense en 2017. También afecta esencialmente a una amplia gama de otras ramas económicas⁴. Además, representó un factor clave para la definición de los objetivos de la UE para el 2020, entre otros, en una evolución inteligente y sostenible.

En cuanto a la gestión de datos,⁵ se estima que el 96% de toda la información capturada en el sector no se utiliza, lo que representa una gran oportunidad perdida para mejorar la eficiencia y la toma de decisiones. Además, el 13% del tiempo de trabajo se invierte en la búsqueda de datos e información de proyectos, evidenciando problemas en la accesibilidad y organización de la información. Otro obstáculo clave es la falta de integración entre herramientas tecnológicas, ya que el 30% de las empresas de construcción utilizan aplicaciones que no se comunican entre sí, dificultando la gestión centralizada de los proyectos. Solo el 8% han implementado sistemas de información para la gestión de proyectos en tiempo real, lo que limita la capacidad de monitoreo y control. Finalmente, el 90% de los datos generados en la industria son no estructurados, incluyendo fotos, historiales de compra y registros de atención al cliente, lo que dificulta su aprovechamiento en la optimización de procesos.

A raíz de todo lo anterior, surge un nuevo enfoque desarrollado en los últimos años denominado Construcción Inteligente⁶ (Smart Construction) que busca transformar los métodos tradicionales mediante la integración de procesos más eficientes y colaborativos. Este enfoque permite mejorar la calidad de los proyectos, optimizar los tiempos de ejecución y reducir los costos mediante un uso más racional y coordinado de los recursos disponibles. Al centrarse en la mejora continua de los procesos, se facilita la planificación, gestión y ejecución de los proyectos de manera más efectiva, alcanzando mejores resultados tanto en el desarrollo como en la entrega final de los mismos.

Una de las principales características de la Construcción Inteligente es su énfasis en la toma de decisiones informadas y basadas en datos, lo que permite una mayor precisión y previsibilidad en las fases de diseño, ejecución y entrega. Al involucrar a todos los actores del proyecto desde las etapas iniciales y mantener una comunicación fluida y constante, se logra un entorno colaborativo que mejora la coordinación y minimiza los riesgos. Esto no solo aumenta la eficiencia operativa, sino que también

⁴ Comisión Europea 2017, Oficina de Análisis Económico de los Estados Unidos.

⁵ Make it, or break it. Reimagining governance, people and technology in the construction industry, Global Construction Survey 2017, KPMG.

⁶ The Construction Innovation Hub es una de las plataformas más relevantes en relación a la innovación y la digitalización en la industria de la construcción. Ofrece múltiples recursos sobre cómo la construcción inteligente y la tecnología pueden transformar el sector.

fomenta un enfoque más proactivo frente a posibles desafíos o problemas durante el ciclo de vida del proyecto.

De igual forma, otro concepto similar es el de Construcción 4.0 (Forcael et al., 2020), el cual se trata de un enfoque más amplio que hace referencia a la digitalización total de la industria de la construcción, inspirado por la Industria 4.0. En este enfoque, se incorporan no solo herramientas de optimización de procesos, sino también una transformación radical de la forma en que se diseñan, fabrican y gestionan los proyectos de construcción.

2.1.1.2. Innovación tecnológica

La adopción de nuevas herramientas digitales y el uso de tecnologías emergentes en los últimos años promovidas por la Construcción Inteligente y la Construcción 4.0, han revolucionado el modo en que los proyectos de construcción se gestionan, desde la fase de diseño hasta las últimas fases del ciclo de vida.



Figura 2.1.2. Innovaciones tecnológicas. Fuente* (Analytical Report)⁷

Una de los cambios fundamentales de las últimas décadas en la gestión de proyectos de construcción es el uso de bases de datos, a través de definiciones de relaciones y restricciones aplicables a conjuntos de datos. La adopción e interconexión de estas bases de datos permite una organización eficiente de la información de proyectos, desde los planos de diseño hasta los presupuestos y cronogramas. Gracias a su capacidad para estructurar grandes volúmenes de datos conectados, las bases de datos relacionales permiten una actualización en tiempo real y una gestión más ágil de la información lo cual facilita la colaboración entre los diferentes actores del proyecto, reduciendo errores y optimizando los procesos de toma de decisiones.

A la par de las bases de datos relacionales, el Cloud Computing (Bello et al., 2020; Mohanaraj et al., 2020) ha ganado protagonismo en la construcción al ofrecer una infraestructura flexible y escalable para almacenar y procesar grandes volúmenes de datos. La computación en la nube permite a las empresas de construcción acceder

⁷ European Construction Sector Observatory. Analytical Report.

a aplicaciones y datos desde cualquier lugar y en cualquier momento, mejorando la colaboración remota entre los equipos de trabajo y asegurando la disponibilidad constante de la información. Además, la nube facilita la integración de diferentes tecnologías, como IoT (Internet de las Cosas) y herramientas de diseño, lo que crea un ecosistema digital cohesivo que mejora la eficiencia global de los proyectos.

En este entorno de datos interconectados, el uso de Big Data se ha vuelto esencial.⁸ La recopilación y análisis de grandes volúmenes de datos generados en cada fase del ciclo de vida de un proyecto (Atuahene et al., 2018) permite identificar patrones, predecir posibles problemas y optimizar recursos. Por ejemplo, el análisis de datos históricos puede ayudar a prever retrasos en la entrega de materiales o identificar factores de riesgo en la obra, lo que permite tomar decisiones informadas para mitigar dichos problemas antes de que ocurran. Además, el Big Data se utiliza para mejorar la gestión de los costos, proporcionando un panorama más claro de los gastos y ayudando a mantener los proyectos dentro del presupuesto.

La inteligencia empresarial (BI) es otra herramienta clave que, cuando se combina con Big Data, permite una visualización más efectiva de los datos y facilita la toma de decisiones estratégicas. Utilizando BI (Golestanizadeh et al., 2023), las empresas de construcción pueden generar informes detallados sobre el desempeño del proyecto, analizar tendencias y optimizar la gestión de recursos. Los cuadros de mando (dashboards) basados en BI permiten a los gerentes de proyecto y ejecutivos tener acceso a información precisa y en tiempo real sobre el estado de los trabajos, lo que mejora la capacidad de respuesta ante imprevistos y aumenta la eficiencia operativa.

Por otro lado, la incorporación de la impresión 3D en la construcción permite la fabricación de componentes de edificios, como estructuras y materiales de acabados, de manera precisa y rápida. Gracias a esta tecnología, los costos de producción se reducen considerablemente, y se mejora la personalización de los diseños arquitectónicos. Además, facilita la creación de formas geométricas complejas que de otra manera serían difíciles de conseguir con métodos tradicionales.

Otra innovación es el uso de drones en la supervisión de obras de construcción los cuales permiten a los gerentes de proyecto monitorear el progreso de la obra de forma remota, capturando imágenes aéreas de alta resolución. Esto no solo mejora la precisión en la recopilación de datos, sino que también aumenta la seguridad en las obras al reducir la necesidad de desplazarse a zonas peligrosas. Los drones también facilitan la creación de modelos tridimensionales del estado actual del activo o del sitio, lo que mejora la planificación y la toma de decisiones en tiempo real.

La inteligencia artificial (IA) también ha comenzado a jugar un papel crucial en la optimización de los procesos de construcción. Mediante el análisis de grandes volúmenes de datos, la IA permite predecir posibles retrasos, fallos en los materiales y problemas en la planificación. Asimismo, los algoritmos de IA pueden optimizar el uso de recursos y materiales, lo que reduce los costos y mejora la eficiencia en el uso de la energía y los materiales. El aprendizaje automático, por su parte, permite la automatización de tareas repetitivas y la mejora continua en la toma de decisiones, facilitando la gestión de proyectos complejos y de gran envergadura.

⁸ The Transformative Power of Building Information Modeling. Digital Transformation. BCG. Boston Consulting Group. 2016.

En cuanto a la gestión de contratos y trazabilidad de la información, otra tecnología emergente es el Blockchain. Su uso en la construcción está poco a poco cambiando la forma en que se gestionan los contratos, los pagos y la trazabilidad de los recursos y materiales. La principal ventaja de esta innovación tecnológica es su capacidad para crear registros inmutables y transparentes que garantizan la integridad y la seguridad de las transacciones. Esto reduce significativamente el riesgo de fraudes y errores en los procesos de contratación y pago. Además, facilita el control de los registros y del seguimiento de los recursos de obra, asegurando que los productos utilizados en la construcción cumplen con los estándares de calidad y origen, lo que también ayuda a mejorar la sostenibilidad de los proyectos. Esta tecnología combinada con el uso de los nuevos modelos de contratos IPD (Entrega integrada de proyectos)⁹, generan una plataforma segura y transparente para la gestión colaborativa, garantizando que todos los participantes estén alineados en cuanto a los plazos, los costos y las responsabilidades. Esto es debido a que los contratos IPD contienen un enfoque colaborativo que involucra a los principales actores del proyecto desde las primeras fases de diseño, con el objetivo de optimizar el rendimiento global y compartir los riesgos y beneficios. Además, a través de contratos inteligentes (smart contracts), Blockchain puede automatizar ciertos procesos dentro de los contratos IPD, como el pago de hitos basados en el cumplimiento de condiciones previamente acordadas. Esto no solo agiliza los procedimientos, sino que también reduce los riesgos asociados a disputas o incumplimientos contractuales, generando una mayor confianza entre los involucrados.

Por su parte, la sostenibilidad también es un tema clave en las innovaciones del sector, y la construcción no es ajena a esta tendencia. Las nuevas tecnologías permiten a los proyectos de construcción minimizar su impacto ambiental, optimizando el consumo de energía, reduciendo las emisiones de CO₂ y promoviendo el uso de materiales reciclables. La integración de sistemas energéticos inteligentes, como los paneles solares y las bombas de calor geotérmicas, junto con el diseño de edificios eficientes en términos de energía, ha propiciado un cambio hacia la construcción de infraestructuras más respetuosas con el medio ambiente.

Además de las tecnologías emergentes, las metodologías ágiles¹⁰ han tenido un impacto significativo en la industria de la construcción. Tradicionalmente, la construcción ha sido vista como un sector con procesos rígidos y lineales, pero la implementación de metodologías ágiles ha permitido mejorar la flexibilidad y la capacidad de adaptación a cambios imprevistos. Con el uso de estas metodologías, los equipos de trabajo son capaces de responder más rápidamente a las necesidades de los clientes, ajustando los plazos y presupuestos de manera más dinámica. La metodología Scrum, por ejemplo, es ampliamente utilizada para la gestión de

⁹ El instituto AIA (American Institute of Architects) es uno de los principales organismos que impulsa este nuevo tipo de contratos y ha desarrollado diferentes documentos tipo. Uno de los más importantes es el modelo de contrato IPD denominado AIA Document C191-2009, el cual define las relaciones y responsabilidades de los distintos participantes.

¹⁰ El Lean Construction Institute (LCI) es una de las organizaciones más destacadas en la promoción de la metodología Lean aplicada al sector de la construcción. LCI proporciona recursos para apoyar la adopción de Lean Construction, como el Last Planner System y otros métodos ágiles aplicados a la construcción. <https://leanconstruction.org/>

proyectos en la construcción, ya que permite trabajar en ciclos iterativos que garantizan resultados rápidos y eficientes.

En este contexto de avance tecnológico y metodológico, la adopción de Building Information Modeling (BIM) ha emergido como una de las metodologías más revolucionarias en la industria de la construcción, ya que no solo mejora la coordinación entre los diferentes actores involucrados en un proyecto, sino que también permite la creación de modelos digitales precisos que pueden ser utilizados a lo largo de todo el ciclo de vida del edificio. Junto a esta metodología, surge también el enfoque de VDC (Virtual Design and Construction) (Hui, 2018), que trata de ampliar los procesos BIM hacia todo el ciclo de vida del activo, incluyendo gestiones de la planificación, la gestión de la construcción, el control de costos y el análisis de riesgos con el objetivo de mejorar la continuidad de la cadena de valor dentro del sector de la construcción.

Por otro lado, los gemelos digitales (Digital Twin) van un paso más allá de la metodología BIM ya que buscan la integración de estos modelos con datos en tiempo real a lo largo del ciclo de vida del activo, combinando el modelo tridimensional BIM con información dinámica proveniente de sensores, dispositivos IoT y sistemas de gestión, lo que permite un seguimiento continuo y preciso del rendimiento del edificio o infraestructura. Esta integración no solo mejora la toma de decisiones durante la fase de diseño y construcción, sino que también optimiza la operación y el mantenimiento, ofreciendo una visión más completa y precisa del comportamiento del activo en su entorno real.

Todo ello también se complementa con el uso de realidades virtuales (VR) y aumentadas (AR), que aportan nuevas dimensiones al diseño, la visualización y la interacción con los proyectos de construcción. Por un lado, la realidad virtual permite a los equipos de trabajo sumergirse en entornos virtuales detallados, lo que facilita la exploración y evaluación de los proyectos antes de su ejecución física. Por otro, la realidad aumentada superpone elementos digitales en el entorno físico, lo que permite a los profesionales visualizar, por ejemplo, cómo se integrarán los sistemas de un edificio en su espacio real durante las fases de construcción y mantenimiento. Estas realidades ofrecen un enfoque más inmersivo e interactivo en la gestión de proyectos, impulsando una mayor colaboración, eficiencia y precisión en todo el ciclo de vida de la construcción, pudiendo llegar también a optimizar la formación del personal y la coordinación entre equipos.

2.1.2. Building Information Modeling (BIM)

El origen del término BIM se remonta a un artículo publicado en 1975,¹¹ en el que se introdujo el concepto de “Building Description System (BDS)”, un prototipo funcional de modelo 3D del edificio con capacidad para extraer secciones y plantas mediante ordenador. En los años siguientes, distintas iniciativas globales promovieron el desarrollo de sistemas de descripción de edificios, denominándose en EE.UU. como “Building Product Models” y en Europa como “Product Information Models”. En 1987, la empresa húngara Graphisoft aportó su visión con el concepto de “Virtual Building” aplicado en ArchiCAD, el primer software CAD que integraba planimetría 2D y modelado 3D. En 1994, se creó el IAI¹² encargado de asesorar el desarrollo de aplicaciones integradas y establecer el primer estándar de intercambio. Dos años antes, en 1992, el término “Building Information Model” fue documentado por van Nederveen y Tolman en la Delft University of Technology, Países Bajos. A partir del año 2000, hitos como la creación de Revit marcaron el inicio del modelado paramétrico, y en 2004 Autodesk popularizó el término BIM como representación digital de procesos constructivos destinados al intercambio e interoperabilidad de información.

Según la norma UNE EN ISO 19650,¹³ BIM se define como el uso de una representación digital compartida de un activo construido para facilitar el diseño, la construcción y la operación, proporcionando una base confiable para la toma de decisiones. Esta metodología es aplicable a edificios, puentes, carreteras e infraestructuras industriales, entre otros.

Desde el punto de vista tecnológico, BIM es entendido como un conjunto de aplicaciones y sistemas de alojamiento de información que optimizan la producción, permitiendo que algunos agentes del sector, como estudios de arquitectura o ingeniería, lo empleen internamente para mejorar su eficiencia sin necesariamente compartir la información con el exterior. No obstante, cuando múltiples agentes utilizan BIM de manera colaborativa, se genera un proceso de interacción que aprovecha su potencial tecnológico. Por otro lado, desde la perspectiva de un proceso, BIM engloba un entorno de trabajo colaborativo entre todos los actores del sector a lo largo del ciclo de vida, apoyado en nuevas tecnologías de software y comunicación, lo que requiere acuerdos sobre responsabilidades y metodologías de trabajo.

La integración de BIM en la gestión de proyectos (Lee, G. & Borrmann, 2020) implica una transformación en la concepción tradicional del ciclo de vida de los activos construidos. Tradicionalmente, el proyecto se limitaba a la fase de diseño y planificación, pero con BIM se entiende como un proceso integral que abarca la planificación, diseño conceptual, diseño constructivo, construcción, explotación e incluso, la rehabilitación o demolición del activo. Esta visión global permite optimizar la toma de decisiones y minimizar costos a lo largo de toda la vida útil de la construcción.

¹¹ The use of computers instead of drawings in building design. Chuck Eastman. 1975.

¹² Consorcio Industrial IAI. International Alliance of Interoperability - EE.UU.

¹³ ISO 19650. Norma internacional de gestión de la información a lo largo de todo el ciclo de vida de un activo construido utilizando el modelado de información para la edificación. 2018.

En los métodos tradicionales, la secuencialidad de las fases de construcción genera ineficiencia al impedir la consideración temprana de distintos puntos de vista. Sin embargo, con BIM se adelanta la intervención de los agentes clave en las fases iniciales del proyecto, favoreciendo la coordinación y optimización de criterios.¹⁴

En el ámbito del diseño digitalizado, la evolución del CAD 2D a BIM ha transformado la planificación y gestión de proyectos. Mientras que CAD estableció la base para la digitalización del diseño en el siglo XX, BIM representa una evolución que incorpora información geométrica, numérica y alfanumérica, con modelos compuestos por objetos individuales cuyas características se almacenan en bases de datos, facilitando la interoperabilidad y colaboración en tiempo real. A diferencia de CAD, que limita su impacto a la fase de diseño, BIM optimiza el proceso a lo largo de todas las etapas del proyecto, asegurando información centralizada y accesible a todos los agentes involucrados.

El siguiente gráfico de Patrick MacLeamy ilustra la diferencia en el valor agregado entre CAD y BIM a lo largo del ciclo de vida del proyecto. Mientras que CAD concentra su utilidad en la fase de diseño, BIM distribuye su valor equitativamente a lo largo de todo el proceso, mejorando la coordinación interdisciplinaria y reduciendo errores en ejecución. A largo plazo, BIM sigue generando valor incluso en la fase de operación y mantenimiento, optimizando costos y promoviendo la sostenibilidad.

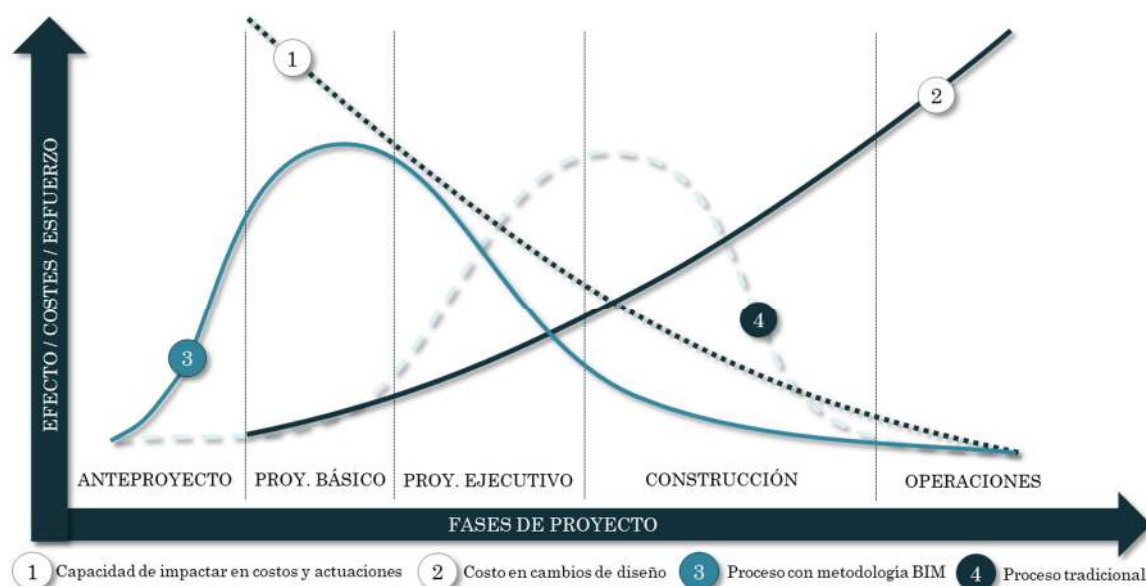


Figura 2.1.3. Gráfico comparativo del esfuerzo con y sin la metodología BIM¹⁵

La implementación de BIM implica la definición temprana de características clave con la participación de todos los agentes responsables, fomentando un enfoque colaborativo que sustituye el modelo tradicional basado en documentación

¹⁴ The New Zealand BIM Handbook 2023. A Guide to Enabling Building Information Modelling (BIM) for the Built Environment. <https://www.biminnz.co.nz/nz-bim-handbook>

¹⁵ Graphic Curve by Patrick MacLeamy from HOK. 2011.

desconectada (Han & Yu, 2018). BIM no solo es un modelo tridimensional, sino un sistema de gestión de datos estructurados y estandarizados que facilita la toma de decisiones basada en información precisa. Más allá de la visualización, la gestión de bases de datos en BIM permite capturar, almacenar y administrar información crucial sobre los objetos del modelo, manteniendo datos accesibles y reutilizables. Su adopción mejora la calidad del trabajo, incrementa la productividad y optimiza la gestión del proyecto, reduciendo costos y tiempos de ejecución.

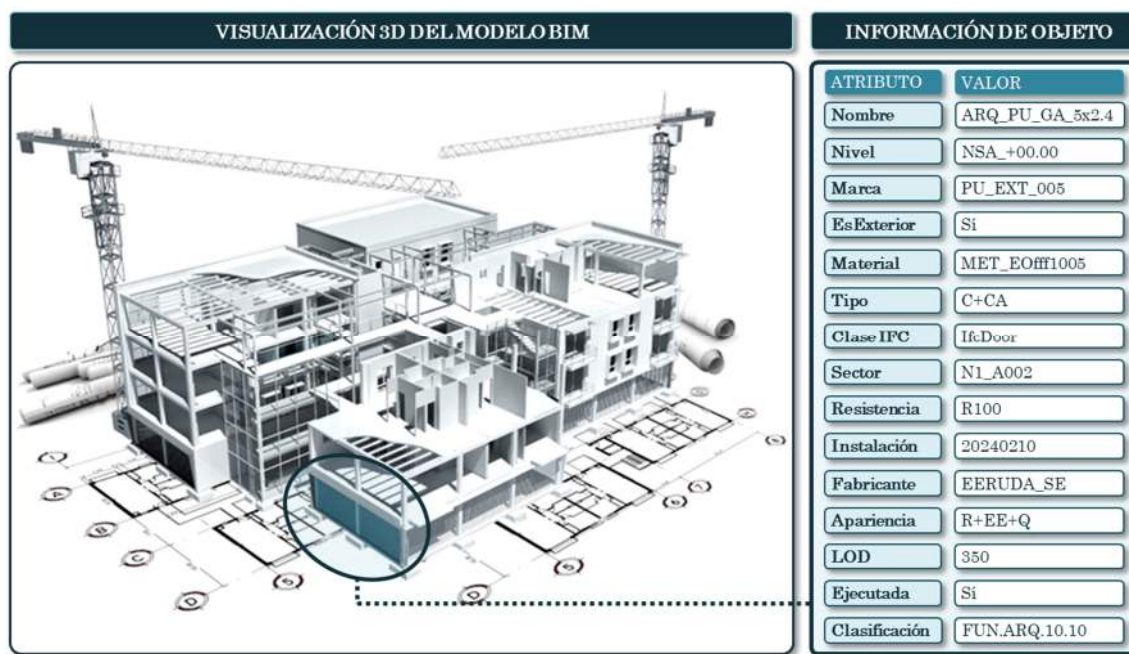


Figura 2.1.4. Modelo de información BIM de edificio de viviendas

Para lograr una correcta implementación, es necesario un cambio cultural hacia un enfoque colaborativo, donde los agentes comprendan la necesidad de compartir beneficios y riesgos para lograr proyectos más eficientes y sostenibles.

Por tanto, esta metodología proporciona un modelo digital integral que elimina la fragmentación de datos, asegurando que cualquier modificación realizada en una parte del modelo se refleje automáticamente en todas las representaciones del proyecto. La parametrización, además, permite realizar cálculos precisos de cantidades, análisis de materiales y extracción de planimetría mediante fórmulas matemáticas y lógicas, facilitando la toma de decisiones y la optimización de recursos en tiempo real. A medida que un proyecto avanza, introducir cambios en la fase de construcción resulta ineficiente y costoso, pero con el uso de la metodología BIM, se hace posible una detección temprana de problemas, lo que permite soluciones más económicas y efectivas.¹⁶

Además, los beneficios de BIM se extienden más allá del diseño, construcción y mantenimiento, integrándose también en otros ámbitos como en la gestión de negocios y la planificación estratégica (Chen, B. et al., 2017) o en la gestión de contratos y licitaciones (Holzer, 2015).

¹⁶ Guía BIM, para la Arquitectura Técnica, BIMAT.2020. Consejo General de la Arquitectura Técnica de España (CGATE).

2.1.2.1. Colaboración de los todos los agentes

La colaboración dentro de la metodología BIM implica la participación activa de todos los agentes involucrados en el proyecto desde sus etapas iniciales (Fountain & Langar, 2018). Arquitectos, diseñadores, ingenieros, contratistas y propietarios acceden a un modelo digital único, que actúa como la fuente centralizada de información. Esta colaboración continua y compartida permite la comunicación clara, la coordinación efectiva y la toma de decisiones informadas a lo largo de todo el proceso de construcción. La transparencia y la integración de la información en tiempo real ayudan a reducir los errores, las duplicidades y los conflictos, generando una sinergia entre los distintos profesionales y asegurando una mayor eficiencia en cada fase del proyecto (Oppong et al., 2017).

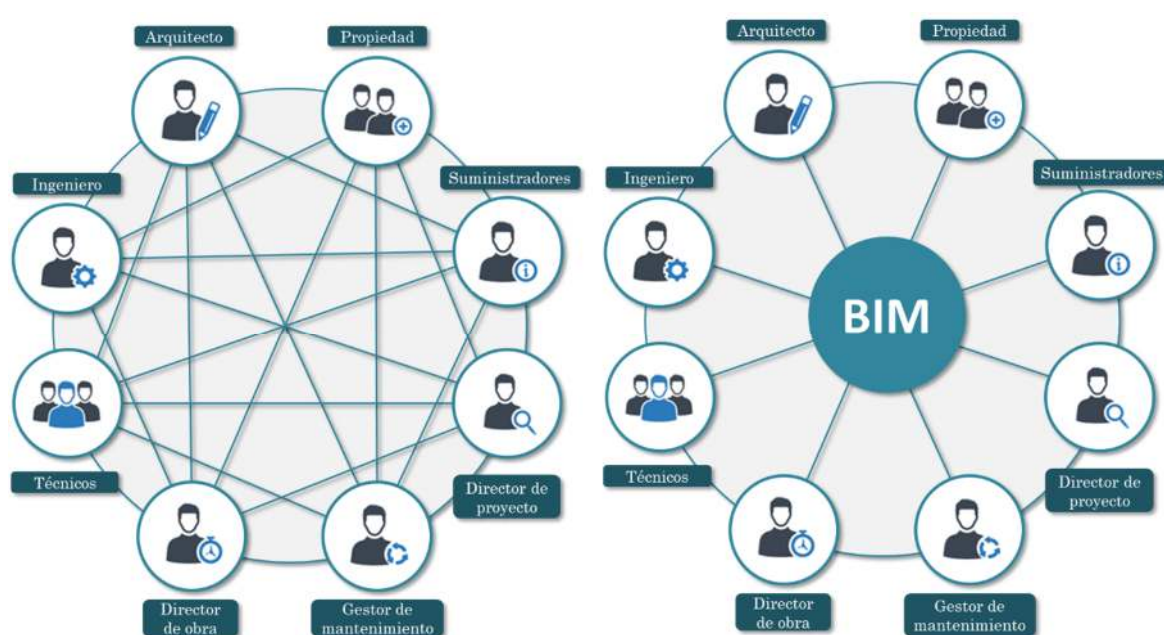


Figura 2.1.5. Colaboración BIM. Fuente* (buildingSmart International)

Para los ingenieros y técnicos, la utilización de BIM facilita una comunicación clara y una coordinación efectiva desde las primeras etapas del proyecto, ya que todos los agentes involucrados tienen acceso a un modelo digital único y centralizado. Esto reduce los errores y duplicidades que a menudo surgen de la falta de información compartida. Al trabajar dentro de un entorno interdisciplinario, BIM promueve la resolución conjunta de problemas, lo cual optimiza los recursos y mejora la toma de decisiones a lo largo del proceso constructivo. Además, también facilita la detección temprana de errores en el diseño, reduciendo modificaciones y optimizando tiempos. La automatización que puede llegar a proporcionar la metodología, incrementa la eficiencia de los procesos y permite a los profesionales maximizar sus inversiones.

Para los propietarios y gestores de activos, BIM ofrece ventajas significativas en términos de fiabilidad y eficiencia a lo largo de todo el ciclo de vida del proyecto.¹⁷ El acceso a un modelo digital preciso permite una planificación más fiable, ya que todos

¹⁷ Guía BIM para Propietarios y Gestores de Activos. BuildingSMART Spanish Chapter.2020.

los cambios y decisiones están registrados y pueden ser fácilmente consultados. Esto reduce los costos inesperados que suelen surgir debido a la falta de información o la malinterpretación de datos. Además, la trazabilidad de decisiones es beneficiosa para los propietarios, ya que ofrece un control más claro sobre los cambios realizados en el diseño o la ejecución del proyecto, facilitando la gestión de los activos a lo largo de su vida útil mediante un sistema integrado que optimiza las operaciones de mantenimiento. Los propietarios pueden acceder a información detallada sobre los componentes del edificio o infraestructura, lo que les permite planificar y gestionar el mantenimiento de manera más eficiente.

2.1.2.2. Continuidad digital

La continuidad digital (Borrmann et al., 2018b) es esencial para asegurar la correcta evolución de la información a lo largo del ciclo de vida del proyecto. Este proceso permite que los datos generados en las diferentes etapas sean constantemente actualizados, lo que garantiza que todos los actores involucrados cuenten con la información precisa y pertinente en cada momento del desarrollo. De este modo, se optimiza la colaboración entre los distintos agentes y se minimizan los errores derivados de la falta de actualización o de acceso a datos desactualizados.

La adopción de la metodología BIM facilita esta fluidez en la transmisión de información, promoviendo la interoperabilidad entre los distintos sistemas y plataformas utilizadas a lo largo del ciclo. A través de BIM, los datos generados en fases anteriores se convierten en recursos valiosos que pueden ser reutilizados y enriquecidos en fases posteriores, mejorando la toma de decisiones y la eficiencia operativa. Esta continuidad se refleja en el gráfico de John Wiley, en el que se compara la reutilización e interoperabilidad de los datos entre los métodos tradicionales y la metodología BIM:

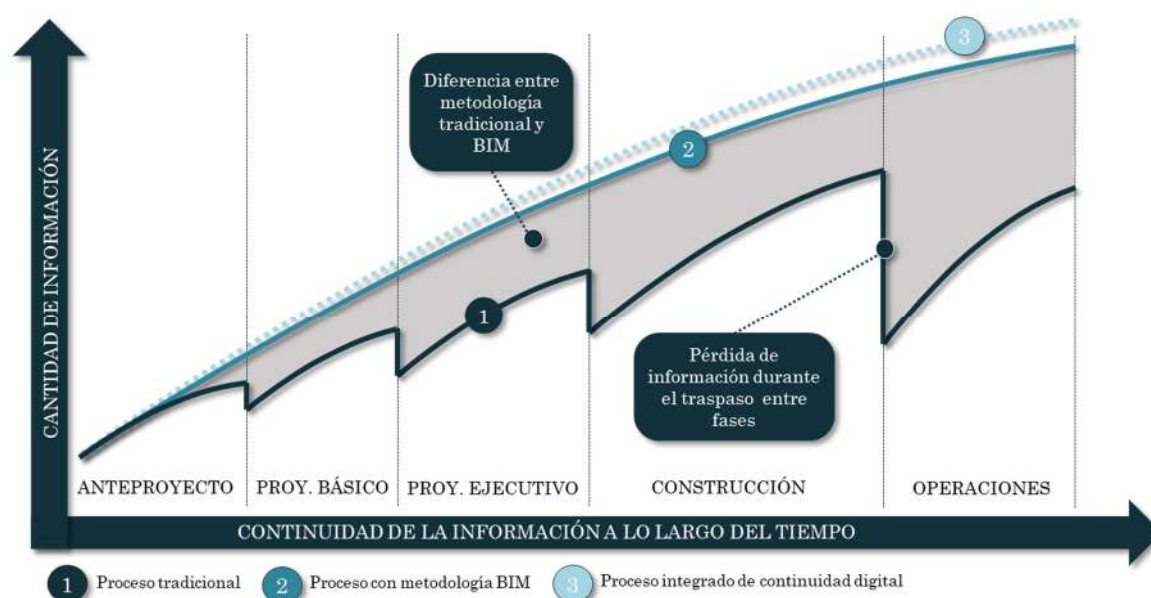


Figura 2.1.6. Continuidad digital durante las fases del ciclo de vida¹⁸

¹⁸ A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers, 3rd ed. Hoboken: John Wiley & Sons, 2018

Un aspecto clave de esta continuidad es la creación de un registro histórico completo que respalde la gestión a largo plazo de los activos. Esta información no solo facilita el mantenimiento y las intervenciones durante la operación, sino que también se convierte en una herramienta fundamental para la gestión predictiva (Tsay et al., 2023). La disponibilidad de datos precisos y actualizados permite anticipar necesidades futuras, optimizando el uso de los recursos y extendiendo la vida útil del activo de manera eficaz.

La transición de un enfoque fragmentado y secuencial a uno digital e integrado presenta un reto significativo, pero también una oportunidad para mejorar la eficiencia global del proyecto. Contar con una base de datos estructurada y bien gestionada es crucial para que la continuidad digital sea efectiva. La correcta organización y almacenamiento de la información no solo garantiza su integridad, sino que también asegura que sea accesible y comprensible por todos los agentes involucrados en el proyecto, independientemente de la fase en la que operen: Planificación, Diseño, Construcción, Explotación y/o Fin de vida útil.



Figura 2.1.7. Evolución del ciclo de vida del activo de edificación

La consideración del ciclo de vida (Sun et al., 2023) enfatiza la necesidad de disponer de toda la información necesaria para el desarrollo de cada fase, así como de la obtención de la información generada al finalizar cada una de ellas para compartir con el resto de agentes involucrados. Mediante BIM, se establecen requisitos concretos que deben cumplir los especialistas y técnicos para desarrollar su actividad, generando en muchos casos información que servirá como base para la ejecución de otras fases posteriores.

2.1.2.3. Centralización de la información

El concepto de "single source of truth", o "única fuente de verdad", implica centralizar toda la información relevante de un proyecto en un único repositorio o sistema digital, asegurando que todos los actores involucrados accedan a los mismos datos actualizados y precisos en tiempo real. Esta centralización de la información previene discrepancias y errores que suelen surgir de la dispersión de datos o de la utilización de versiones desactualizadas en diferentes fuentes. En el contexto BIM,¹⁹ esta centralización es esencial para garantizar la coherencia y la interoperabilidad de los datos a lo largo del ciclo de vida. A partir de una base de datos común, todos los agentes pueden acceder a la información del activo, lo que facilita la colaboración, la coordinación y una toma de decisiones más informada.

De esta manera, se elimina la fragmentación de datos y la duplicación de esfuerzos que caracterizan los métodos tradicionales de gestión. Además, dentro del entorno centralizado, todas las actualizaciones de datos se reflejan inmediatamente, lo que asegura que todos los involucrados trabajen con la versión más reciente y precisa. Esto reduce los riesgos asociados a la gestión de datos dispersos, mejora la trazabilidad de las decisiones y aumenta la eficiencia global del proyecto.

Diversos estudios científicos respaldan la relevancia de contar con una fuente única de verdad en la gestión de proyectos de construcción, remarcando que la adopción de BIM y el establecimiento de una única fuente de verdad (Preidel et al., 2018) contribuyen a mejorar la eficiencia operativa, reducir errores y optimizar la comunicación entre las partes interesadas. También subrayan que la centralización digital de la información reduce la fragmentación y los conflictos derivados del uso de fuentes de datos descoordinadas (Masaki Muto, 2020), destacando también que la centralización facilita la toma de decisiones y la optimización de recursos, ya que permite acceder a datos completos y actualizados en todo momento.

Además de todo ello, toda la información que no se encuentra en modelos BIM se integra y gestiona desde un repositorio documental denominada EDC (Entorno Común de Datos). Este entorno permite no solo almacenar e intercambiar la información, sino que también suele contener diferentes procesos para coordinar y aprobar documentos entre los diferentes agentes involucrados.

2.1.2.4. Estandarización

En el contexto de la ISO 9000,²⁰ un estándar se define como un documento que proporciona especificaciones, directrices o características que pueden ser utilizadas consistentemente para asegurar que materiales, productos, procesos y servicios sean adecuados para su propósito. La estandarización, por su parte, es el proceso de crear, aplicar y adoptar estándares dentro de una industria o sector. Su objetivo es optimizar procesos, mejorar la eficiencia y garantizar que todas las partes involucradas trabajen bajo un mismo marco de referencia. En la metodología BIM, la estandarización está estrechamente relacionada con la implementación de normas como la ISO 19650 en la gestión de proyectos, la adopción de metodologías

¹⁹ Roadmap for Implementation of BIM in the UK Construction Industry. Engineering, Construction and Architectural Management. Khosrowshahi. 2011.

²⁰ UNE-EN ISO 9000:2015. Sistemas de gestión de la calidad. Fundamentos y vocabulario.

estructuradas para la nomenclatura de archivos y la definición de procesos de intercambio de información entre agentes.

Es esencial para garantizar la interoperabilidad y la coherencia entre los diferentes sistemas y actores involucrados, ya que asegura que todos los participantes utilicen un lenguaje común y estructuras de datos compatibles (Hooper, 2015). Además, contribuye a la escalabilidad de los proyectos y permite la integración de herramientas tecnológicas avanzadas.

2.1.2.5. Interoperabilidad

La interoperabilidad es un concepto determinante en los sistemas de información, y su definición varía según el contexto. En el ámbito de la administración pública,²¹ se define la interoperabilidad como la capacidad de los sistemas de información y los procedimientos que soportan estos sistemas para compartir datos y posibilitar el intercambio de información y conocimiento. Esta definición abarca tres dimensiones fundamentales: técnica, semántica y organizativa. Es crucial para la cooperación entre diferentes aplicaciones, la integración de servicios, y el desarrollo de políticas públicas y privadas. Este marco también subraya la importancia de garantizar la neutralidad tecnológica y la adaptabilidad a los avances de las tecnologías de la información y la comunicación (TIC), permitiendo que los ciudadanos elijan libremente las alternativas tecnológicas sin restricciones (Ley 40/2015).

En el entorno BIM, la interoperabilidad se presenta como uno de los mayores retos y beneficios, ya que se refiere a la capacidad de integrar datos provenientes de diferentes plataformas, aplicaciones y herramientas utilizadas por los distintos actores involucrados en el proyecto. Esto se logra mediante el uso de estándares y formatos abiertos, como el IFC²² (Industry Foundation Classes), que permiten el intercambio de datos entre diversas plataformas sin pérdida de información.

La interoperabilidad es crucial para garantizar la fluidez de la colaboración entre equipos multidisciplinares y para evitar problemas derivados de la incompatibilidad entre diferentes herramientas de software. El uso de formatos abiertos no solo facilita el intercambio de datos, sino que también reduce los costos relacionados con la conversión de datos y la resolución de conflictos debido a la falta de compatibilidad entre sistemas (Turk, 2020). Además, la interoperabilidad permite que la información generada en una fase del proyecto sea utilizada y reutilizada en fases posteriores, mejorando la eficiencia y la toma de decisiones.

Por otro lado, la adopción de BIM y su capacidad para integrar datos de diversas fuentes es fundamental para superar las limitaciones impuestas por los métodos tradicionales de trabajo en la construcción, que a menudo se caracterizan por sistemas de información aislados y poco interoperables (Jiang et al., 2019). Además, la interoperabilidad en BIM no solo mejora la comunicación y coordinación entre los equipos, sino que también permite una mejor gestión de los recursos y la optimización de los procesos constructivos.

²¹ Esquema Nacional de Interoperabilidad (ENI) España. Real Decreto 4/2010.

²² ISO 16739:2024. Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries.

2.1.2.6. Trazabilidad

Según la norma ISO 9000, la trazabilidad se define como la capacidad para seguir el histórico, la aplicación o la localización de un objeto. Esta definición enfatiza la importancia de poder rastrear el recorrido y las transformaciones que ha experimentado un producto o servicio a lo largo de su ciclo de vida. Ampliando la concepción del término objeto a cualquier tipo de proceso o recurso de información, la trazabilidad en BIM se puede entender como un seguimiento preciso de todas las decisiones y cambios realizados durante el desarrollo del proyecto, por lo que está íntimamente ligado al control de la información.

En el caso del modelo digital, cada modificación queda registrada, lo que facilita la auditoría y el cumplimiento de los estándares de calidad. Este nivel de detalle también mejora la transparencia y minimiza los riesgos asociados con discrepancias o malentendidos. Es especialmente valiosa en proyectos complejos, donde múltiples agentes trabajan simultáneamente ya que proporciona un registro claro de responsabilidades y ayuda a identificar y resolver conflictos de manera eficiente.

Por otro lado, en relación a la gestión documental,²³ el registro de la información durante todos los hitos del proyecto, permite poder revisar cualquier archivo final con sus anteriores versiones, permitiendo validar cambios, comprobar actualizaciones, verificar quién y cuando realizó el trabajo, restaurar versiones antiguas, etc. También en la mayoría de plataformas BIM, existen funcionalidades que acceden a registros cronológicos de las actividades de miembros, documentos, incidencias, planos y archivos en general.

2.1.2.7. Implementaciones, costes y retorno de la inversión

La implementación de la metodología requiere una inversión inicial significativa en diversos recursos, como software, hardware y capacitación del personal. Sin embargo, los beneficios a largo plazo derivados de su adopción superan ampliamente estos costos iniciales, ofreciendo un retorno de inversión (ROI) notablemente alto. Según múltiples estudios, la implementación BIM contribuye a la reducción de errores, optimización de recursos y mejora de la planificación (Hernán, 2021), lo que a su vez reduce las desviaciones de presupuesto y cronograma en los proyectos, incrementando su viabilidad económica (Smith, 2014). En este sentido, un estudio de KPMG International²⁴ subraya que el 53% de los encuestados en 2018 identificaron a BIM como la tecnología con el mayor potencial de retorno en proyectos de capital, aunque esta cifra disminuyó a un 42% en 2021, y volvió a aumentar hasta un 49% en 2023. Esto demuestra que, a pesar de la aparición de nuevas tecnologías, BIM sigue siendo una herramienta clave para optimizar costos, mejorar la planificación y reducir los riesgos financieros en el sector de la construcción.

La adopción de BIM no solo mejora la gestión de los proyectos en términos económicos, sino que también facilita la evaluación precisa de los costos asociados a cada fase del proceso constructivo. Esta capacidad de prever y controlar los costos permite identificar áreas donde se pueden generar ahorros significativos, contribuyendo a la sostenibilidad y rentabilidad de los proyectos (Chen, B. et al., 2017). Al integrar BIM en los procesos de construcción, se obtiene una visión más

²³ Manual BIM ETS. Euskal Trenbide Sarea. Anejo 10. El CDE. 2014.

²⁴ Familiar challenges, new solutions: KPMG Global Construction Survey. 2023.

clara y detallada de los requerimientos y las implicaciones financieras, lo que facilita una gestión más eficiente de los recursos. Este nivel de control es crucial para la reducción de desperdicios y la mejora de la eficiencia en los proyectos, lo que fortalece aún más la viabilidad económica de las empresas que adoptan esta metodología.

De igual manera, es fundamental contar con un liderazgo sólido que impulse la adopción de la metodología y que entienda que los pilares fundamentales para cualquier cambio metodológico se basan en las personas, las tecnologías y los procesos de la empresa.

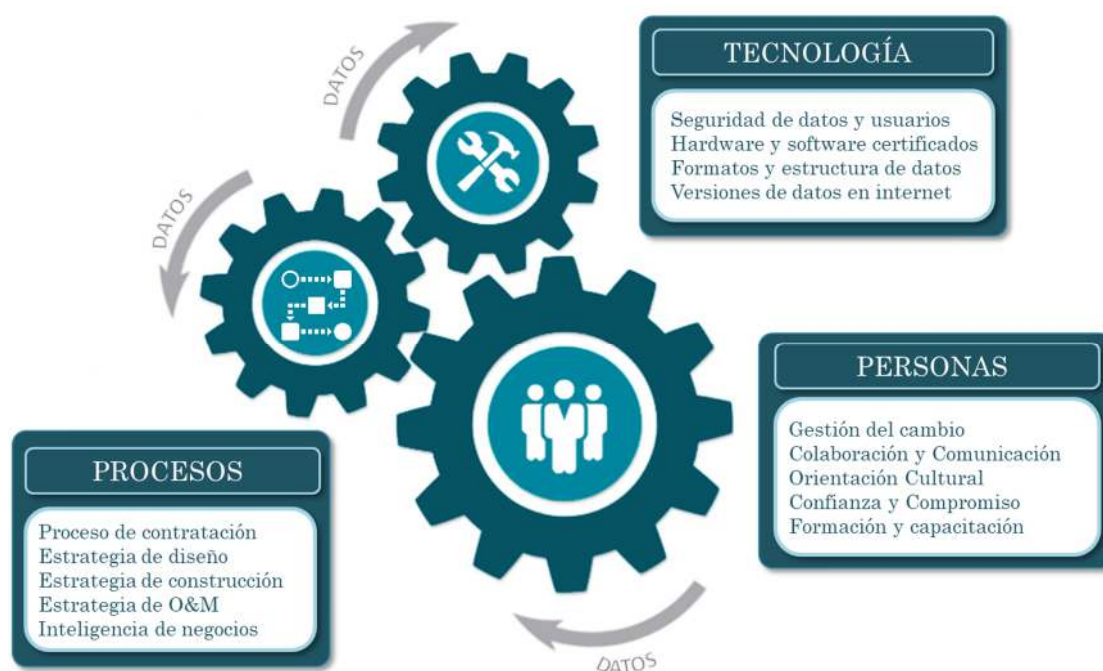


Figura 2.1.8. Cambio metodológico: procesos, personas y tecnologías

Aunque la inversión en tecnología es importante, la gestión del cambio a nivel humano es igualmente crucial. La resistencia al cambio es un desafío común, y es vital que el liderazgo esté comprometido con la transformación. El equipo de implementación, debe analizar los procesos actuales de la empresa, motivar al equipo y garantizar que los cambios se realicen de manera eficiente. Además, debe facilitar la comunicación entre todos los miembros del equipo y asegurar que se sigan los nuevos procedimientos establecidos, supervisando la integración de las nuevas herramientas y prácticas.

Durante estos procesos, la gestión efectiva del equipo humano es un factor crítico. Es necesario realizar un análisis exhaustivo del personal, sus roles, formación y experiencia. Este diagnóstico permite diseñar un plan de formación personalizado que asegure la capacitación adecuada de cada miembro del equipo en relación con las herramientas y procesos de BIM. Asimismo, es importante identificar a aquellos miembros que están más dispuestos al cambio y que tienen la capacidad o la autoridad para redefinir procesos y proponer modificaciones. Estos individuos deben ser seleccionados para formar parte del primer equipo BIM, el cual se encargará de llevar a cabo el primer proyecto piloto. En cuanto a la tecnología, la selección adecuada del software y hardware es crucial para el éxito de la implementación. Las herramientas BIM requieren una infraestructura tecnológica más robusta que las

utilizadas en los métodos tradicionales, lo que implica un análisis detallado de los equipos y software disponibles en la empresa. Las auditorías iniciales deben incluir un inventario de las herramientas tecnológicas existentes, licencias, versiones y capacidad de los sistemas, para evaluar su interoperabilidad en el nuevo entorno BIM. Además, el hardware de la empresa debe cumplir con los requisitos específicos para ejecutar el software BIM, lo que puede requerir una actualización en los equipos, especialmente en cuanto a la memoria RAM, procesador, tarjeta gráfica y almacenamiento. Este análisis previo es esencial para asegurar que la empresa esté preparada para adoptar las nuevas herramientas sin comprometer su capacidad operativa.²⁵

Otro aspecto crucial es la adaptación de los procesos internos a la metodología BIM. Para ello, es necesario realizar un análisis de los flujos de trabajo actuales y determinar cuáles deben ser ajustados o mejorados para cumplir con los nuevos requisitos. Este proceso implica identificar las áreas de la organización donde la integración será más efectiva y crear estándares y procedimientos que guíen la ejecución de proyectos bajo esta nueva metodología (Bryde et al., 2013).

En cuanto al retorno de la inversión (ROI), la adopción de BIM debe ser evaluada en términos cuantitativos y cualitativos. Un análisis de ROI considera tanto los costos iniciales de implementación como los beneficios a largo plazo derivados de la mejora en la eficiencia, reducción de errores y optimización de recursos. Los primeros costos incluyen la adquisición de software y hardware, así como la formación del personal y la adaptación de los procesos. Posteriormente, se deben realizar inversiones estratégicas en la adaptación de la gestión de proyectos, la creación de estándares y la integración de los nuevos procesos.²⁶

A medida que la implementación avanza y se obtienen resultados positivos, el ROI mejora sustancialmente, validando la inversión inicial (Sompolgrunk et al., 2021). Por tanto, las empresas deben estructurar la implementación en fases, comenzando con proyectos piloto más sencillos y avanzando progresivamente hacia proyectos más complejos. Además, es necesario adoptar un enfoque de aprendizaje basado en la experimentación, donde los miembros del equipo puedan aplicar lo aprendido a proyectos reales, lo que mejora la asimilación de los nuevos conocimientos y facilita la adopción de BIM.

2.1.2.8. Etapas de madurez BIM

Las etapas de madurez en la adopción BIM varían entre proyectos y organizaciones, reflejando el grado de implementación tecnológica, la calidad de los procesos y la eficacia de la colaboración entre los equipos. Estas etapas se han desarrollado a partir del modelo²⁷ de niveles de madurez BIM propuesto por en 2008 y evolucionado con la normalización establecida en la ISO 19650. Alcanzar niveles superiores de madurez requiere un compromiso significativo en inversión, capacitación y

²⁵ The Business Value of BIM for Construction in Major Global Markets. McGraw Hill Construction. 2014.

²⁶ BIM Level 2 Benefits Measurement. Introductory note: Approach and benefits framework. PWC. 2018.

²⁷ The UK maturity Model. Bew & Richards. 2008.

estandarización, lo que se traduce en una mayor eficiencia, reducción de riesgos y mejor toma de decisiones.

El grado de madurez BIM se mide a través de diversos factores clave, tales como la utilización de estándares nacionales e internacionales, el modelado 3D de elementos y sistemas, la clasificación de objetos, la implementación de bibliotecas de componentes, el uso de formatos abiertos, la vinculación de tiempos y costos con modelos de información, la adopción de entornos comunes de datos, la generación automatizada de mediciones y planos, la estandarización de procesos, la integración de datos de sostenibilidad y la gestión del ciclo de vida del activo.

Históricamente, los niveles de madurez BIM fueron ampliamente utilizados en Reino Unido a través del modelo Bew-Richards y su adición a la norma PAS1192-2²⁸, el cual definió cuatro niveles principales de madurez.

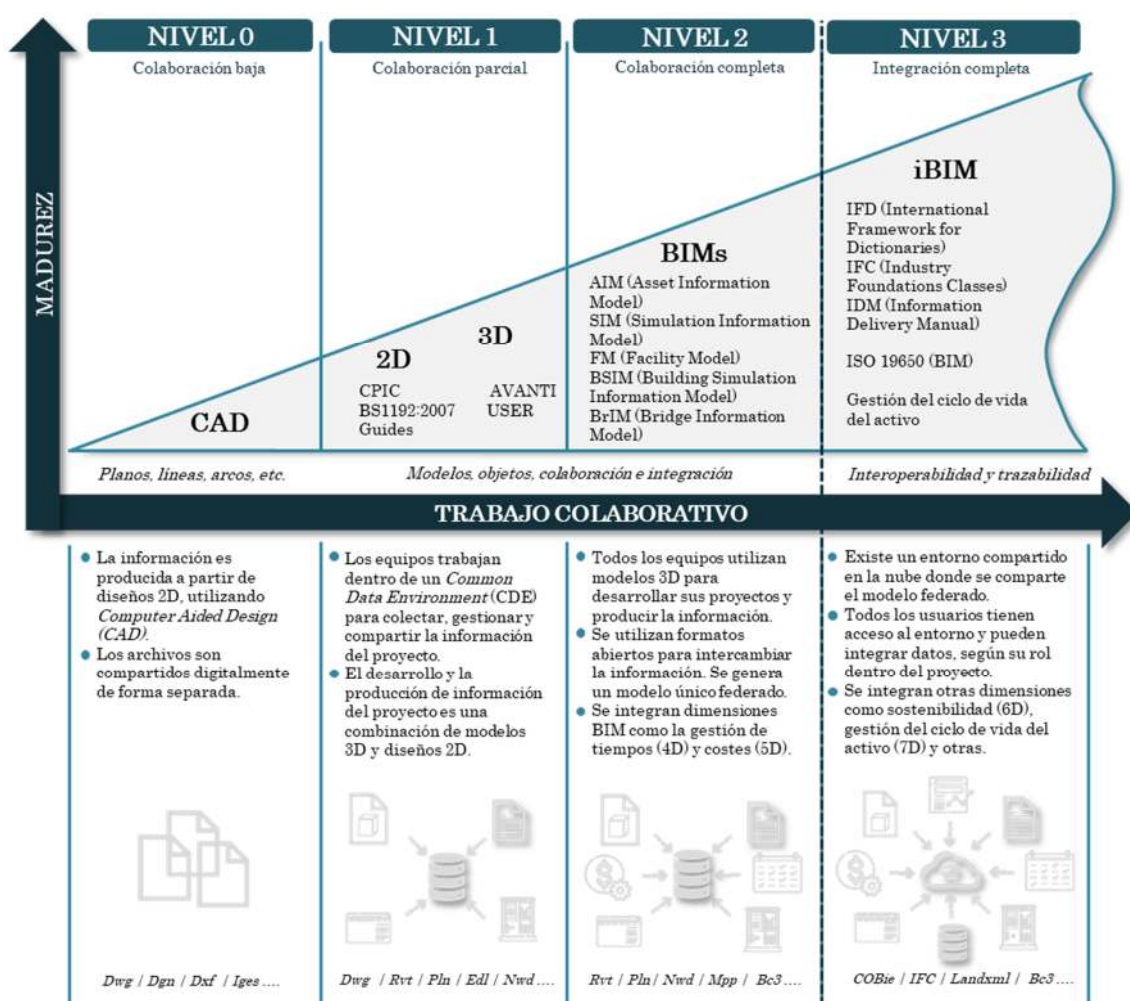


Figura 2.1.9. Etapas de madurez BIM. Fuente* (UK BIM Framework)

Reino Unido ha exigido desde 2016 el Nivel 2 BIM en todos los proyectos financiados con fondos públicos,²⁹ lo que ha impulsado la adopción de metodologías colaborativas

²⁸ PAS1192. *Publicly Available Specification for the collaborative production of information*. 2013. Sustituida gradualmente por la norma ISO 19650.

²⁹ Four things Mark Bew told us about the future of BIM and digital construction. NBS. Richard McPartland. 2017.

en la industria de la construcción. Por otro lado, en el contexto internacional, la ISO 19650 reinterpreta estos niveles en tres etapas de madurez BIM, alineadas con un enfoque progresivo hacia la colaboración y organizado por diferentes capas de información: negocio, información, tecnología y estándares.

- Etapa 1: Modelado basado en objetos. Introduce la gestión normalizada de la información generada por sistemas CAD, utilizando un entorno de datos común (EDC). Sin embargo, los modelos 3D no se comparten entre disciplinas.
- Etapa 2: BIM Colaborativo. Todas las partes trabajan con modelos 3D, intercambiando información en formatos abiertos. La interoperabilidad permite la creación de modelos federados y la coordinación entre disciplinas.
- Etapa 3: BIM Integrado. Representa la colaboración total mediante un único modelo compartido alojado en una base de datos central, garantizando una coordinación completa y reduciendo la redundancia de información. Este enfoque, conocido como 'Open BIM', plantea retos en términos de derechos de acceso a la información y responsabilidad legal.³⁰

La transición entre estas etapas requiere no solo avances tecnológicos, sino también una transformación cultural y organizacional dentro de la industria. A medida que las normativas y estándares continúan evolucionando, la adopción progresiva de BIM en sus diferentes etapas se ha convertido en un objetivo estratégico para mejorar la eficiencia y sostenibilidad en la gestión de proyectos de construcción e infraestructura.

³⁰ Digital Built Britain. Level 3 Building Information Modelling - Strategic Plan. 2015.
https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/410096/bis-15-155-digital-built-britain-level-3-strategy.pdf

2.1.3. Gestión de la información BIM

La gestión de la información BIM se fundamenta en la generación, integración, intercambio, actualización y trazabilidad de los documentos, modelos y recursos asociados al diseño, construcción y mantenimiento de un activo de construcción a lo largo de su ciclo de vida. La correcta implementación de esta gestión implica definir flujos de trabajo, roles y responsabilidades, así como emplear procesos estandarizados de revisión, validación y aprobación de los modelos y la documentación relacionada, garantizando la integridad y fiabilidad de la información en todos las fases e hitos de entrega del proyecto.

2.1.3.1. Guías y estándares

La implementación de Building Information Modeling ha impulsado el desarrollo de un marco normativo internacional y nacional para definir principios, directrices y procedimientos que garanticen su correcta aplicación en proyectos de edificación y obra civil. Uno de los antecedentes clave fue el desarrollo de la serie PAS 1192 en el Reino Unido, elaborada por el British Standards Institution (BSI) entre 2007 y 2018. Basada en la BS 1192:2007, esta serie estableció los fundamentos del BIM Level 2 en el contexto británico, convirtiéndose en un referente internacional. La PAS 1192 incluía directrices para la producción y gestión colaborativa de la información durante el diseño, construcción y operación de activos, abordando además aspectos clave como el intercambio de datos, la seguridad de la información y la gestión de activos en la fase operativa.

Con la voluntad de unificar criterios a nivel global, la serie PAS 1192 dejó de actualizarse en 2018 para dar paso a la serie ISO 19650, publicada ese mismo año. Esta nueva serie adoptó gran parte de los conceptos, estructuras y procesos definidos en la PAS 1192, pero adaptándolos a un marco normativo internacional aplicable a proyectos de cualquier país. En España, la serie ISO 19650 ha sido adoptada como UNE-EN ISO 19650, integrándose así en el marco normativo europeo y nacional. Esta adopción ha sido clave para la consolidación de BIM en España, especialmente en proyectos promovidos por la administración pública.

Aparte de la ISO 19650, otras normas se han ido desarrollando de forma paulatina, ampliando el marco regulatorio de diferentes actividades y procesos BIM (Hjelseth, 2017). Algunas de las más relevantes son la ISO 16739, denominada Industry Foundation Classes (IFC), que define un formato abierto para el intercambio de información entre diferentes software y plataformas BIM, promoviendo la interoperabilidad o la ISO 7817, que establece el Nivel de Información Necesario (LOIN), la cual define los requisitos sobre el nivel de detalle que debe contener la información en cada fase del proyecto.

Además de las normas oficiales, diversas organizaciones y asociaciones sectoriales han elaborado guías y protocolos no vinculantes, adaptados a las particularidades de cada mercado nacional, con el objetivo de facilitar la comprensión de roles, flujos de trabajo y requisitos de información. Estas guías sectoriales complementan las normas formales y han sido esenciales para la difusión de buenas prácticas. Por parte de BuildingSmart Spain, se ha desarrollado una base de información a modo de biblioteca de documentos en la que se pueden consultar todas estas guías nacionales e internacionales.

Pese a estos avances, la implementación efectiva de la ISO 19650 enfrenta importantes dificultades debido a la complejidad de sus requisitos, la diversidad de actores implicados y las diferencias en el grado de digitalización de las empresas. Diversas iniciativas (Abanda et al., 2025; Godager et al., 2022) han analizado los procesos definidos en la serie ISO 19650 y han desarrollado guías interpretativas, plantillas y herramientas de apoyo, que buscan clarificar conceptos clave y facilitar su aplicación práctica en empresas de arquitectura, ingeniería y construcción.

2.1.3.2. Objetivos y usos BIM

Los objetivos BIM representan las metas específicas que un proyecto busca alcanzar mediante la implementación de la metodología BIM y están orientados a la mejora de la colaboración, la reducción de errores, el control de costos, la optimización de procesos constructivos y/o la sostenibilidad del proyecto. Por norma general, estos objetivos se utilizan para definir el valor potencial de BIM del proyecto, pero en algunos casos, también pueden implicar un beneficio directo para quienes participan en este, al aumentar sus conocimientos, capacidades u optimizar sus procesos de trabajo. Con el fin de llevarlos a la realidad, estos se descomponen en usos BIM, que son las aplicaciones concretas de la metodología.³¹

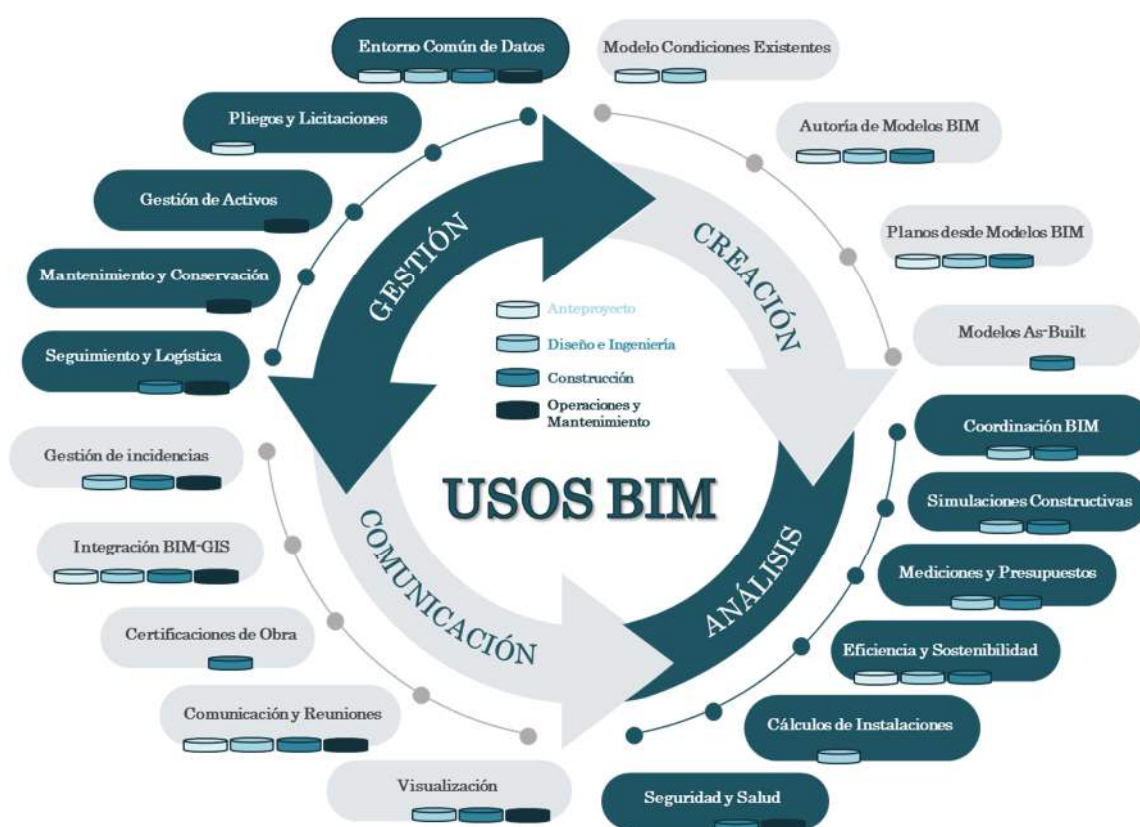


Figura 2.1.10. Usos BIM según categoría y fase de proyecto

Una de las primeras iniciativas que definió y clasificó estos usos fue desarrollada por la Pennsylvania State University en 2010, conocida como la BIM Project Execution Planning Guide, que identificó 25 usos fundamentales (Messner et al., 2011). Esta

³¹ National BIM Guide. NATSPEC. 2022. www.natspec.com.au

guía marcó un hito al establecer un marco estructurado para la selección y priorización de usos BIM en función de los objetivos de cada proyecto. Con el tiempo, otras administraciones y estrategias internacionales han ido incorporando estos usos, como el Reino Unido dentro de su programa BIM o NBBIM US,³² y también otras organizaciones nacionales como la Comisión Interministerial BIM (esBIM), Infraestructuras de Cataluña, Generalitat Valenciana o Euskal Trenbide Sarea (ETS), entre muchos otros. La mayoría de estas iniciativas definen el significado de cada uso y también indican las entradas y salidas necesarias, agentes, relaciones entre usos, categorías generales y fases a las que aplica cada uso.

Por otro lado, un estudio realizado por Becerek-Gerber,³³ identificó en sus resultados que los usos BIM con mayor impacto según los encuestados son la visualización (63.8%), la detección de conflictos (60.7%) y el diseño de edificios (60.4%). También que otros beneficios secundarios son la creación de modelos As-Built (42.1%), el ensamblaje de edificaciones (40.6%) y la secuenciación constructiva (36.2%). Otro estudio realizado por la NBS,³⁴ resaltó que el 70% de los profesionales encuestados generaban modelos en 3D, lo que indica una fuerte adopción de tecnologías de modelado en el diseño arquitectónico. Además, el 74% de los encuestados reportó trabajar de manera colaborativa, lo que refuerza la importancia del BIM como un facilitador de la cooperación interdisciplinaria. Sin embargo, en ese mismo estudio se detectaron también otros usos menos implementados como la integración de modelos en el ciclo de vida del proyecto (37%). Por último, otro estudio realizado por buildingSMART International,³⁵ indicó que los principales usos de BIM en la industria incluyen la coordinación del diseño 2D/3D (85%), la detección de conflictos y controles de calidad (72%) y la revisión/aprobación de diseños (54%). Además, se observa un creciente interés en el intercambio de archivos (54%), la planificación de materiales y cronogramas (43%) y la transferencia de información de activos (39%). Más de la mitad de los encuestados (51%) identificaron la reducción de costos en el diseño y la construcción como una de las principales ventajas, mientras que el 63% destacó la mejora en la comprensión y visibilidad de las decisiones de diseño.

Atendiendo a la inclusión de requisitos BIM en los pliegos de licitaciones públicas en el sector de la edificación en España durante los últimos años,³⁶ más de la mitad (62%) de los pliegos se definieron con usos BIM, destacando como usos más habituales la integración de disciplinas y coordinación 3D (82%), la visualización (79%) y la obtención de planos (58%). Estos resultados reflejan una clara orientación hacia la mejora de la coordinación interdisciplinar y la optimización de la fase de diseño, donde la interoperabilidad y la visualización anticipada de conflictos adquieren un papel clave para la reducción de errores y sobrecostos en obra. Otros usos relevantes, aunque con menor frecuencia, son la obtención de mediciones y presupuestos (54%), la futura gestión de la explotación y mantenimiento (51%) y la

³² BIM Uses Definition Standard. National BIM Standard v04. National Institute of Building Sciences. Estados Unidos.

³³ Exploring Advantages and Challenges of Adaptation and Implementation of BIM in Project Life Cycle. Becerek-Gerber 2010.

³⁴ BIM Survey: Summary of Findings. Malleson. NBS. 2016.

³⁵ Understanding the Role of BIM and CDEs Today and Expectations for the Future. Cumbre Virtual de 2021. buildingSMART International. 2022.

³⁶ Observatorio BIM en España. Inclusión de requisitos BIM en pliegos. Comisión Interministerial BIM. Periodo 2018-2024. Actualización del 18 de septiembre de 2024.

simulación constructiva (44%), evidenciando un interés creciente por extender el uso de BIM más allá de la fase de diseño, hacia la gestión integral del ciclo de vida. Sin embargo, áreas como la realidad virtual/aumentada, el seguimiento de certificaciones y los análisis energéticos y medioambientales presentan aún una baja presencia (entre el 0% y el 8%).

Además de los usos BIM, desde hace algunos años se ha popularizado el concepto de las dimensiones BIM. Estas dimensiones abarcan: 3D, que representa digitalmente un modelo tridimensional del proyecto, considerando este modelo como una base de datos estructurada con información técnica detallada; 4D, que introduce la variable temporal, permitiendo simular la secuencia constructiva y optimizar la planificación del proyecto; 5D, que añade el control de costos en tiempo real, facilitando estimaciones económicas precisas; 6D, que integra parámetros ambientales y de eficiencia energética, permitiendo evaluar el consumo de recursos del proyecto; y 7D, que se enfoca en la gestión de activos y el mantenimiento a lo largo del ciclo de vida del edificio, mediante herramientas de Facility Management y sistemas de Gestión del Mantenimiento Asistido por Ordenador (GMAO).

2.1.3.3. Roles y responsabilidades

La metodología BIM introduce una estructura organizativa que promueve la gestión eficiente de la información, asegurando una coordinación precisa entre los equipos involucrados. Este enfoque no solo mejora la calidad y precisión de los datos, sino que también optimiza la colaboración entre los distintos actores, permitiendo una interacción fluida a través de entornos digitales. La correcta estructuración de estos perfiles dentro del proyecto es esencial para mantener la integridad de los modelos, garantizando que los datos se mantengan actualizados y sean accesibles para todos los interesados, desde los diseñadores hasta los responsables del mantenimiento.³⁷

Además, la flexibilidad de la asignación de estas responsabilidades permite que cada organización adapte la metodología a sus necesidades específicas. Las empresas pueden asignar las responsabilidades en función de la complejidad del proyecto, los recursos disponibles y las particularidades de cada fase, lo que favorece una mayor eficiencia en la ejecución. La adaptabilidad de la estructura organizacional en BIM se refleja en la posibilidad de asignar las funciones de manera que se optimicen los flujos de trabajo y se reduzcan los posibles conflictos o duplicación de esfuerzos.

Todos estos perfiles, se pueden clasificar en tres niveles organizacionales: los designados por el cliente, que abarcan todas las fases del proyecto, desde la planificación hasta el mantenimiento del activo; los de proveedores, vinculados a fases específicas del proyecto, como el diseño, la construcción y el mantenimiento, y que son fundamentales en la implementación de los requerimientos BIM; y los aplicables a grupos o equipos de trabajo, que se centran en tareas concretas y pueden organizarse según o paquetes de trabajo.

A partir del análisis de diferentes estudios (Ahankoob et al., 2019; Bosch-Sijtsema, P. et al., 2019; Godager et al., 2022), estándares y guías BIM,³⁸ se ha elaborado una

³⁷ Guía para implementar y gestionar proyectos BIM: Diario de un BIM manager. David Barco. 2018.

³⁸ Guías BIM FGV, Guía BIM ITeC, Guías ETS, Guía Técnica BIMAT, NBIMS US, AIA's Digital Practice Documents, CIB W78: Information Technology in Construction.

tabla que recoge los perfiles más relevantes, incluyendo las responsabilidades y las habilidades de cada uno de ellos.

Rol BIM	Responsabilidades	Habilidades
Head of BIM / BIM Champion	- Lidera la transformación organizacional hacia la adopción de la metodología BIM.	Conocimiento avanzado en gestión estratégica, liderazgo, adopción tecnológica, gestión del cambio y habilidades de comunicación.
	- Define la estrategia organizacional para la implementación BIM, alineándola con los objetivos del proyecto y de la empresa.	
	- Coordina la capacitación y desarrollo continuo del personal en BIM.	
	- Facilita la adquisición de herramientas y software necesarios para la implementación de BIM.	
	- Actúa como enlace entre la alta dirección y los equipos BIM.	
	- Desarrolla e implementa planes para garantizar la correcta ejecución de los proyectos BIM.	
	- Supervisa la implementación BIM a nivel organizacional y asegura la alineación con los estándares internacionales.	
BIM Manager	- Establece la estrategia a largo plazo para la adopción de BIM dentro de la organización.	Software BIM, habilidades de gestión de proyectos, liderazgo y toma de decisiones estratégicas.
	- Gestiona la implementación de la metodología BIM a nivel del proyecto.	
	- Define los procedimientos y estándares para el desarrollo del proyecto.	
	- Supervisa la ejecución de los proyectos BIM.	
	- Coordina las interacciones entre los diferentes miembros del equipo BIM.	
	- Controla la calidad de la información y los modelos BIM generados.	
Gestor de la Información del Cliente	- Organiza y gestiona la formación interna en BIM.	Conocimiento avanzado de CDE, software de gestión de la información, estándares BIM, y procesos de interoperabilidad.
	- Gestiona la información a lo largo de todo el ciclo de vida del activo, desde el diseño hasta la operación y mantenimiento.	
	- Establece los requerimientos BIM del proyecto, asegurando que se alineen con los objetivos del cliente.	
	- Supervisa el flujo de información entre los equipos, gestionando la interoperabilidad.	
	- Define y controla los límites de responsabilidad y las expectativas entre los diferentes equipos.	
Information Manager	- Asegura que el intercambio de información cumpla con los requisitos del cliente.	Conocimiento avanzado en gestión de la información, estándares y
	- Gestiona la consistencia, coherencia y alineación de la información BIM en todo el proyecto.	
	- Asegura que los estándares BIM y los requerimientos del cliente sean cumplidos en los entregables de cada fase.	

Rol BIM	Responsabilidades	Habilidades
	- Supervisa el intercambio de datos y la gestión de la información en el CDE.	normativas BIM, software de gestión de datos.
	- Establece y mantiene las normativas para la organización y clasificación de los datos.	
	- Garantiza que la información esté disponible y accesible para todos los miembros del equipo en las fases correspondientes.	
Coordinador BIM	- Gestiona una disciplina específica desde el punto de vista de la información.	Software de modelado BIM, conocimiento avanzado de la disciplina que coordina, gestión de equipos, habilidades de comunicación.
	- Realiza el seguimiento y control de las interacciones entre diferentes disciplinas.	
	- Redacta y mantiene actualizado el BEP.	
	- Asegura que los modelos federados sean precisos y completos.	
	- Supervisa la calidad de los entregables BIM.	
Proyectista BIM	- Desarrolla la propuesta técnica de diseño utilizando herramientas BIM.	Conocimiento técnico en diseño, habilidades en herramientas BIM.
	- Produce entregables BIM (planos, informes y documentación técnica).	
	- Asegura que los elementos del modelo sean detallados y representen correctamente las condiciones constructivas.	
	- Colabora con otros miembros del equipo BIM para garantizar la integración de los modelos.	
Modelador BIM	- Crea y actualiza los modelos BIM según los requisitos establecidos.	Habilidades en herramientas de modelado BIM, conocimientos de sistemas constructivos.
	- Gestiona y prepara bibliotecas de elementos estándar.	
	- Colabora con el Coordinador BIM y otros miembros del equipo.	
	- Introduce elementos de anotación en los modelos para crear una documentación técnica detallada.	
Supervisor BIM	- Realiza revisiones periódicas de los modelos para garantizar su calidad y coherencia.	Habilidades en revisión de modelos, conocimientos en normativa BIM y control de calidad.
	- Verifica que los modelos se alineen con los requisitos establecidos.	
	- Supervisa la precisión de la información antes de la entrega al BIM Manager.	
	- Identifica y resuelve inconsistencias o errores en los modelos.	

Tabla 2.1.1. Roles y responsabilidades BIM

En este contexto, la norma ISO 19650 introduce una perspectiva importante al recomendar que el término "roles" sea reemplazado por "funciones", promoviendo una visión más flexible de la gestión de la información. Esta orientación facilita la integración de la metodología BIM en diversas industrias, al permitir que los proyectos se adapten a diferentes escalas y entornos operativos sin perder eficacia ni coherencia. De esta manera, la norma define una serie de procesos y actividades

estructuradas, que son desarrolladas en distintas etapas y organizadas según las realice el adjudicador, el adjudicatario principal u otros adjudicatarios.

2.1.3.4. Requisitos de información

Los Requisitos de Información garantizan una correcta gestión y colaboración en la creación, intercambio y mantenimiento de la información del ciclo de vida del activo de construcción. Según la ISO 19650, son un conjunto de especificaciones sobre la información que debe producirse, cuando debe producirse, su método de producción y su destinatario. Su correcta definición es crucial para asegurar la interoperabilidad, calidad y eficiencia en la gestión del proyecto (Abanda et al., 2025). En cuanto a su organización, están estructurados en cuatro categorías principales, que siguen un flujo secuencial y que están interrelacionadas.³⁹

- OIR (Requisitos de Información de la Organización): definen los objetivos estratégicos de la organización encargada de gestionar el proyecto, estableciendo los requerimientos generales que deben cumplirse en todas las fases del proyecto. Son fundamentales para garantizar que la información producida en el proyecto esté alineada con los objetivos operativos y organizacionales de la entidad.
- PIR (Requisitos de Información del Proyecto): son los requisitos específicos para el desarrollo de un proyecto, detallando qué información debe generarse en cada fase del ciclo de vida del proyecto. Están directamente relacionados con el diseño, la construcción y las entregas del proyecto, estableciendo los estándares y formatos para la creación, validación y entrega de la información durante la fase del proyecto.
- AIR (Requisitos de Información del Activo): se enfocan en la fase de operación y mantenimiento del activo. Estos requisitos derivan de los OIR y se centran en la gestión de la información del activo una vez que el proyecto ha finalizado y se encuentra en uso. Especifican los tipos de información necesarios para mantener y operar el activo de manera eficiente a lo largo de su ciclo de vida, facilitando el mantenimiento, la rehabilitación y la renovación del mismo.
- EIR (Requisitos de Información del Cliente): es el primer documento BIM clave en un proyecto, elaborado por el cliente o un representante del mismo. Define los requisitos necesarios para la correcta ejecución del proyecto bajo la metodología BIM, estableciendo qué datos deben ser producidos y gestionados. Este documento puede ser independiente o formar parte del pliego de licitación del proyecto. Es crucial para que los licitadores comprendan los requerimientos y puedan desarrollar su propio BEP (Plan de Ejecución BIM), asegurando que los objetivos del cliente se cumplan desde el inicio.

Todos ellos están estrechamente interrelacionados y se desarrollan en un orden secuencial (Godager et al., 2022). Los OIR proporcionan el marco estratégico general para la gestión de la información a nivel organizacional. A partir de los OIR, se derivan los AIR, que definen las necesidades específicas para la operación y mantenimiento del activo. Los PIR, que se centran en la ejecución del proyecto, deben

³⁹ UK BIM Framework Standards. BSI and Nima. Information management according to BS EN ISO 19650, Guidance Part 1: Concepts. UK BIM alliance.

alinearse con los OIR para garantizar que la ejecución del proyecto esté en consonancia con los objetivos organizacionales. Además, los EIR son el punto de partida para todo el proceso, ya que definen los requerimientos de información del cliente, proporcionando la base para que los licitadores desarrollen sus planes y estrategias para cumplir con los objetivos establecidos. Posteriormente, documentos como el Plan de Ejecución BIM se basarán en la información contenida en el EIR para coordinar la producción y entrega de la información.

En el caso del EIR, este documento debe desarrollarse una vez se hayan identificado los objetivos y necesidades del proyecto, sirviendo como la referencia principal para la producción de la información durante toda la duración del proyecto. Este documento debe ser claro, detallado y específico para asegurar que todas las partes involucradas estén alineadas en cuanto a los requerimientos de información. A lo largo del ciclo de vida del proyecto, el EIR se mantendrá como la base para la planificación de la ejecución y entrega de la información.⁴⁰ Muchas organizaciones y empresas del sector desarrollan sus propias plantillas de documentos BIM para mantener un orden y contenido bien definido, como es el caso de las plantillas de NBIMS US.⁴¹

2.1.3.5. Plan de ejecución

La ISO 19650 establece la gestión del Plan de Ejecución BIM (BEP, por sus siglas en inglés), y su relación con los otros documentos del ciclo, como el EIR. Este BEP es un documento contractual elaborado por los proveedores de cada fase del proyecto cuyo principal objetivo es definir los procesos de gestión BIM necesarios para cumplir con los requisitos del cliente establecidos en el marco BIM. El documento aborda varios aspectos fundamentales, tales como definir el proceso de ejecución del modelado, establecer el proceso de gestión de la información del proyecto, especificar los procedimientos de intercambio de información, identificar necesidades específicas de la ejecución BIM en el proyecto y establecer la infraestructura tecnológica y las competencias del proveedor para el desarrollo del modelado.⁴²

Su objetivo principal es garantizar que todos los involucrados trabajen de forma coherente y alineada con los requerimientos establecidos por el cliente y, al mismo tiempo, facilitar la entrega de información a lo largo del ciclo de vida. En la mayoría de los casos, el BEP se divide en dos documentos, un BEP de Oferta (Pre-BEP), presentado durante la licitación, y un BEP de Proyecto, desarrollado por el proveedor principal, que detalla la información del BEP de Oferta con mayor especificidad.⁴³

En la fase de licitación, el BEP de Oferta (Pre-BEP) es una respuesta del proveedor a los requisitos del cliente mostrando cómo el proveedor cumplirá con las exigencias BIM. Se presenta como parte de la oferta en el proceso de licitación y no entra en detalles técnicos, pero establece cómo el proveedor planea cumplir con los requerimientos y gestionar la información. Una vez adjudicado el contrato, se elabora un BEP de Proyecto (Post-BEP) con más detalle sobre la ejecución BIM, estableciendo las directrices específicas para el desarrollo del proyecto de acuerdo

⁴⁰ Australia and New Zealand Guide to ISO 19650. BrisBIM. 2018.

⁴¹ Project BIM Requirements Standard. National BIM Standard v04. National Institute of Building Sciences. Estados Unidos.

⁴² Introducción a la Serie EN-ISO 19650. buildingSMART Spain. 2021.

⁴³ UK BIM Framework. Normas, directrices, recursos y protocolos BIM.

con los objetivos y las estrategias establecidas en el Pre-BEP. A través de este documento, también se asegura que se cumplan los requisitos del cliente y se garantice la calidad y precisión de los entregables de información durante las distintas fases del proyecto.⁴⁴ El BEP de Proyecto se convierte en la base sobre la que se desarrollarán otros documentos clave en la implementación BIM, como el MIDP (Master Information Delivery Plan), que gestiona la entrega de información en todo el ciclo de vida del proyecto, y el TIDP (Task Information Delivery Plan), que detalla las entregas por actividad o disciplina. Ambos planes aseguran que la información se entregue en los formatos adecuados, en los plazos establecidos y con los responsables claros para cada tarea.

2.1.3.6. Intercambio de información

El intercambio de información es fundamental para garantizar la interoperabilidad y la eficiencia entre los actores involucrados del proyecto, ya que permite que la información, desde la fase de diseño hasta la de operación y mantenimiento, sea accesible y comprensible para todos los participantes, independientemente del software que utilicen (Turk, 2020). Su correcta gestión asegura que los diferentes sistemas puedan trabajar juntos, facilitando la colaboración y mejorando la calidad del proyecto. Además, este proceso es esencial para asegurar la accesibilidad de los datos en todas las fases, evitando que la información quede atrapada en un sistema tecnológico obsoleto. En proyectos BIM, los formatos de intercambio pueden ser propietarios o abiertos. La elección de uno, otro o ambos depende de varios factores, como los requisitos del cliente, las características del proyecto y las herramientas utilizadas por los distintos actores involucrados.

Por un lado, los formatos propietarios son aquellos cuyo desarrollo y control está restringido a una empresa específica, lo que implica que su especificación no es pública y su uso puede estar limitado por licencias o derechos de autor. Estos formatos pueden generar dependencia de software y dificultar la interoperabilidad entre sistemas, ya que la información contenida en ellos solo puede ser leída y utilizada por aplicaciones que soporten ese formato. Aunque estos formatos son ampliamente utilizados, su naturaleza cerrada plantea dificultades para la colaboración entre distintas plataformas, lo que puede complicar la comunicación y el intercambio de datos entre los diferentes participantes del proyecto.

Por otro lado, los formatos abiertos tienen su especificación disponible públicamente y pueden ser utilizados sin restricciones, lo que promueve la interoperabilidad entre diferentes sistemas y aplicaciones. El uso de estos formatos se hace esencial en BIM, ya que permite que la información sea accesible y utilizable sin depender de un único proveedor o software. Entre los formatos abiertos más relevantes, se encuentran el IFC (Industry Foundation Classes), el BCF (BIM Collaboration Format), el gbXML (Green Building XML), DXF (Drawing Exchange Format), entre otros.

El uso de formatos abiertos es crucial para garantizar la longevidad de los datos, ya que asegura que la información siga siendo accesible y utilizable a lo largo del tiempo, independientemente de las tecnologías que puedan surgir. Además, favorecen la colaboración eficiente entre los diferentes actores del proyecto,

⁴⁴ Australia and New Zealand guide TO ISO 19650.

permitiendo un flujo de información más fluido y reduciendo la posibilidad de errores o malentendidos (Figueiredo Muller et al., 2017).

2.1.3.7. Infraestructura tecnológica

En proyectos de construcción, intervienen diversas disciplinas que utilizan software especializado para resolver requisitos específicos, como estructuras, eficiencia energética, climatización o instalaciones. Cada uno de estos programas debe integrarse dentro de la metodología BIM, ya que, aunque cada disciplina modela el proyecto dentro de su área, es fundamental que el software utilizado sea compatible para garantizar la interoperabilidad entre las distintas herramientas.

Estas herramientas BIM se caracterizan por su capacidad para interpretar y exportar datos entre diferentes aplicaciones a través de formatos de intercambio bien establecidos, sin que se pierda información en el proceso. Esto permite que el flujo de trabajo sea fluido, ya que las herramientas se insertan en un ecosistema en el que diversas aplicaciones se alinean para resolver los requerimientos del proyecto. Las principales empresas de software de modelado ofrecen herramientas integradas que intentan abarcar todas las necesidades del proyecto dentro de su propia marca. No obstante, la verdadera flexibilidad de BIM radica en la capacidad de integrar herramientas de distintas marcas que trabajen juntas de manera eficiente, lo cual no siempre es sencillo debido a las diferencias en los estándares y protocolos utilizados.⁴⁵

Otro punto importante es la definición de las versiones que se utilizarán, ya que la interoperabilidad entre diferentes versiones del mismo software puede presentar dificultades. Con ello se evitan problemas de compatibilidad durante el intercambio de datos y se asegura que el flujo de trabajo se desarrolle sin contratiempos. Además, es recomendable establecer los requisitos mínimos de hardware necesarios para garantizar que las herramientas BIM puedan operar eficientemente, manejar grandes volúmenes de datos y generar modelos complejos sin inconvenientes.

2.1.3.8. Entorno Común de datos

La legislación en el Reino Unido⁴⁶ definió el Entorno Común de Datos (CDE, por sus siglas en inglés) como una fuente única de información para un proyecto, destinada a la recolección, gestión y difusión de los documentos aprobados y relevantes para los equipos multidisciplinares del proyecto. Este entorno puede incluir servidores de proyecto, extranet, sistemas de extracción de documentos u otras herramientas específicas. Si bien el CDE es accesible para todos los agentes del proyecto, cuenta con espacios privados que garantizan la confidencialidad de ciertos datos. Su uso no implica compartir toda la información, sino seleccionar qué datos compartir y en qué momento, siempre de forma digital.⁴⁷

Originalmente desarrollado dentro del contexto de los estándares del Nivel 2 de BIM en el Reino Unido, el CDE ha ido evolucionando para abarcar no solo los datos BIM, sino también información clave como contratos, cronogramas y órdenes de cambio.

⁴⁵ Guías de implementación BIM. Ferrocarrils de la Generalitat Valenciana (FGV). 2020.

⁴⁶ PAS 1192-2:2013. Specification for information management for the delivery phase of construction projects using BIM. British Standards Institution (BSI).

⁴⁷ ¿Qué es un CDE? BuildingSMART Spanish Chapter. Manuel Bouzas Cavada. 2017.

Su función principal es ofrecer una única fuente de información accesible para todos los miembros del equipo, lo que mejora la colaboración y ayuda a evitar errores y duplicaciones. También facilita la gestión de versiones y el control de acceso a la información, lo que asegura que los datos estén actualizados y disponibles de manera precisa a lo largo de todas las etapas del proyecto. Se diferencia de un sistema de gestión de documentos, que se limita al almacenamiento y recuperación de documentos, ya que está diseñado específicamente para facilitar la colaboración en proyectos complejos, como los que implementan BIM. Además, al proporcionar un entorno más adecuado para la gestión de datos, el CDE mejora la eficiencia, calidad y seguridad en la gestión del proyecto. Para que el trabajo colaborativo sea eficiente y la base de datos útil para todos los agentes involucrados, el CDE debe ser un sistema accesible y, por lo general, basado en la nube. Esta accesibilidad facilita la visualización y análisis de datos a través de internet, lo que es fundamental en proyectos de gran envergadura. La selección de este entorno depende de varios factores, como la capacidad de almacenamiento, los requisitos de seguridad, la facilidad de uso y las funcionalidades necesarias.⁴⁸

Según la ISO 19650, el proceso BIM debe gestionarse en un entorno fiable, seguro y estructurado, y el CDE cumple con estos requisitos, ya que centraliza la información y la comparte de una manera organizada entre los agentes involucrados, reduciendo riesgos asociados a la duplicación o desactualización de los datos. Además, dentro del entorno la información es organizada en distintos estados, tales como "Trabajo en curso", "Compartido", "Publicado" y "Archivado", que reflejan el ciclo de vida de los documentos y su acceso permitido en cada fase.

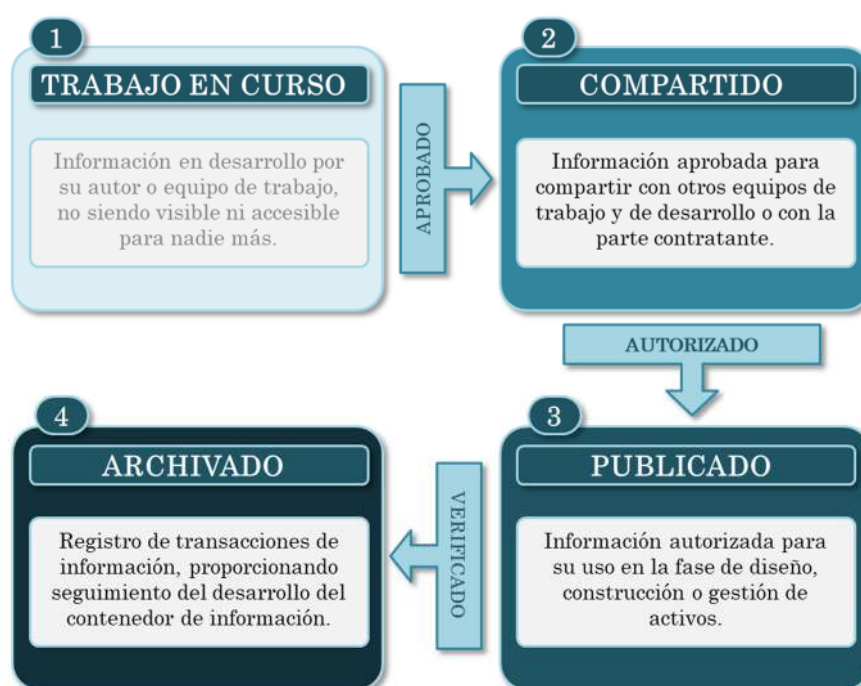


Figura 2.1.11. Estados de la información del CDE. Fuente* (ISO 19650)

⁴⁸ What's a Common Data Environment (CDE) and Why It Matters. Joan Allen. Autodesk Construction Solutions. 2024.

A través de una meticulosa organización y estructuración de los documentos en carpetas, se establece un flujo de trabajo ágil y coordinado, que permite que los equipos multidisciplinares trabajen de forma sincronizada y eficiente. Además, la trazabilidad de la información mediante metadatos y control de versiones asegura que el proyecto se ejecute de acuerdo con los estándares establecidos, manteniendo la calidad y la integridad de los datos en todo momento.

Por otro lado, la gestión del CDE requiere un enfoque estructurado, en el cual el rol del Information Manager es fundamental para mantener la coherencia y sincronización de la información a lo largo del ciclo de vida. La propiedad de la información recae en el propietario del proyecto, quien tiene la autoridad para gestionar el acceso y uso posterior de los datos, de acuerdo con los requerimientos establecidos.

2.1.3.9. Nomenclatura BIM

La nomenclatura en BIM es esencial para la correcta organización, intercambio y gestión de la información. En primer lugar, la nomenclatura de documentos y modelos se refiere al sistema estructurado utilizado para identificar y organizar los archivos dentro del proyecto. Un sistema coherente de nomenclatura permite una localización más rápida de los archivos y una gestión más eficiente durante las diferentes fases del proyecto. Los elementos principales a considerar en la nomenclatura de documentos incluyen un código único para el proyecto, la designación del tipo de documento, la disciplina a la que pertenece el archivo, la fase del proyecto, la versión del documento y la fecha de creación o actualización. Esta estructura facilita el control de las versiones, mejora la trazabilidad de los archivos y permite a los distintos equipos trabajar con la misma información actualizada.⁴⁹

En cuanto a la nomenclatura de objetos y atributos, esta se refiere a la organización de los elementos físicos y funcionales dentro del modelo BIM. Cada tipo de objeto debe tener un nombre único que refleje su identidad dentro del proyecto, garantizando una referencia clara y consistente. Además, los atributos asociados a estos objetos, como los materiales o las dimensiones, deben estar claramente definidos y asociados. Algunas de las guías de objetos más importantes, como la NBS,⁵⁰ utilizan una clasificación detallada de productos y elementos de construcción, organizados jerárquicamente para asegurar que todos los componentes de un proyecto se identifiquen de forma precisa. De forma similar, la guía OBOS⁵¹ proporciona un sistema de clasificación y nomenclatura orientado a la construcción de edificios y sus componentes. En el caso de España, guías como las de la GDO⁵² o la eCOB⁵³ también incluyen apartados específicos para la definición del nombrado de objetos y atributos.

2.1.3.10. Procesos BIM

En el ámbito BIM, los procesos representan secuencias lógicas de actividades que deben seguirse para alcanzar objetivos específicos, con el fin de garantizar la eficacia

⁴⁹ Manual de nomenclatura de documentos al utilizar BIM. BuildingSmart Spain. 2021.

⁵⁰ BIM Object Standard. National Building Specification (NBS). V02, 2023.

⁵¹ The Open BIM Object Standard. Natspec. V10, 2018.

⁵² Guía de desarrollo de objetos BIM. Bimética. V01, 2019.

⁵³ Estándar de creación de objetos BIM. ITeC. V01, 2018.

y la calidad en la ejecución de los proyectos (Abanda et al., 2025). Para visualizar y gestionar estos procesos, se emplean mapas que muestran las interacciones y relaciones entre las diferentes tareas del proyecto, facilitando así la organización y colaboración entre los miembros del equipo. La creación de un mapa de procesos es una herramienta crucial que se desarrolla después de identificar los usos BIM que guiarán la implementación del proyecto. Para ello, es necesario definir las entradas y salidas de cada actividad. Las entradas representan los datos, información o recursos necesarios para comenzar un proceso, mientras que las salidas son los resultados generados por dicho proceso, que luego se utilizan como entrada para otros. Con ello se asegura que cada proceso esté correctamente ejecutado y que la información fluya de manera eficiente entre las distintas fases del proyecto. También estos mapas definen los intercambios de información entre los equipos y las responsabilidades asignadas a cada fase, lo que facilita el control y la trazabilidad del proyecto.

De acuerdo con la ISO 19650, los procesos de gestión de la información se dividen en dos niveles. El nivel 1 aborda los procesos generales de gestión de la información, organizando todos los procesos necesarios para cumplir con los usos BIM, mientras que el nivel 2 se enfoca en los subprocesos detallados para cada uso específico de BIM. El mapa de procesos de nivel 1 debe identificar la secuencia general del proyecto, los responsables de cada actividad, las fases de implementación, los intercambios de información y las versiones de software utilizadas. En el nivel 2, cada subproceso debe desglosarse en actividades jerárquicas, identificando las dependencias entre procesos y los puntos de control para asegurar que se cumplan los estándares establecidos.

Dentro de estos procesos BIM, existen diversas categorías y tipos que son fundamentales para el desarrollo del proyecto. Los procesos de modelado se centran en la creación y modificación de modelos 3D que reflejan fielmente las condiciones y características del proyecto. Los procesos de coordinación, por otro lado, están enfocados en asegurar que todos los modelos de las distintas disciplinas se integren de manera efectiva, evitando conflictos y asegurando la interoperabilidad entre ellos. Los procesos de calidad implican la validación de datos, revisiones de los modelos y la implementación de medidas para asegurar la exactitud y fiabilidad de la información. Dependiendo del tipo de proyecto y su alcance, existen otros procesos que deben ser definidos como la conectividad con el presupuesto o con la planificación de la obra.

2.1.3.11. Seguridad de la información

La seguridad en entornos BIM⁵⁴ es un aspecto fundamental ya que de ello depende la protección de los datos y la operatividad de los proyectos. Dado el carácter colaborativo y la naturaleza compartida de los proyectos BIM, asegurar la protección de la información es crucial para evitar alteraciones no autorizadas, pérdidas de datos y accesos indebidos. La seguridad BIM se fundamenta en tres principios clave de la seguridad de la información: confidencialidad, integridad y disponibilidad. La confidencialidad asegura que solo los usuarios autorizados tengan acceso a la información del modelo, evitando filtraciones y accesos no deseados. La integridad garantiza que los datos no sean alterados de forma indebida, asegurando su precisión

⁵⁴ Seguridad de la Información. BuildingSmart Spain. 2021.

y fiabilidad. Finalmente, la disponibilidad permite que la información BIM esté accesible en todo momento para los usuarios y sistemas que la necesiten, evitando interrupciones que puedan afectar la operatividad del proyecto.

Esta seguridad está especialmente orientada a la protección de los datos dentro del CDE, ya que la ISO 19650⁵⁵ establece normas claras para la gestión de la información, incluyendo la necesidad de acuerdos explícitos entre todas las partes involucradas para garantizar la seguridad y proteger la propiedad intelectual. En este contexto, el protocolo de información debe detallar un sistema de seguridad que contemple el acceso, edición, y control de la información, así como las copias de seguridad. Estos aspectos deben adaptarse a la estructura organizacional y las necesidades particulares del proyecto, considerando que los participantes pueden tener estándares y procesos diferentes.

En cuanto al acceso y control de edición de la información, estos deben ser gestionados de acuerdo a los roles, estableciendo permisos específicos para lectura, edición y publicación de los documentos y modelos. En el caso de las copias de seguridad, estas deben ser realizadas de manera periódica y almacenadas en ubicaciones seguras, tanto locales como en la nube, con protocolos definidos sobre la responsabilidad de cada parte en cuanto a su creación, almacenamiento y restauración en caso de fallo del sistema. Además, las copias de seguridad deben garantizar la integridad de los datos, permitiendo la recuperación de la información en cualquier momento, evitando pérdidas irremediables de datos cruciales.

En el ámbito jurídico, la seguridad también involucra la propiedad intelectual de los modelos. Los contratos deben detallar la autoría de los modelos, especificando quién es responsable de la creación y modificación de los diferentes elementos dentro del modelo y cómo se gestionan los derechos de autor, especialmente cuando el modelo es editado por diversas partes. La propiedad intelectual es un tema complejo que hace necesario un acuerdo claro sobre los derechos de uso, modificación y distribución del modelo.

Finalmente, la seguridad jurídica también involucra la necesidad de que los participantes cuenten con seguros de responsabilidad civil que cubran posibles errores derivados de la información contenida en los modelos. Estos seguros deben garantizar la protección frente a reclamaciones que puedan surgir debido a fallos en la información, defectos en el diseño o problemas derivados de la implementación del modelo. Además, es necesario asignar la propiedad del proceso BIM y gestionar adecuadamente los riesgos asociados con la transferencia del modelo entre equipos, para asegurar que cada etapa del proyecto esté respaldada por un control de calidad y seguridad apropiado, minimizando los riesgos de errores y disputas legales.

⁵⁵ UNE-EN ISO 19650-5:2021. Parte 5: Enfoque orientado a la seguridad en la gestión de la información. AENOR.

2.1.4. Estructura de la información BIM

La información y su estructura de datos son la base de la metodología BIM. La información de la documentación BIM se incluye tanto en los archivos de los modelos, en los documentos de proyecto, en los objetos BIM y en los datos en ellos alojados. Cada dato representa una propiedad o característica específica de una entidad, sistema o elemento del proyecto y aporta información clave para la toma de decisiones, garantizando la coherencia y coordinación entre los distintos equipos y disciplinas. La adecuada estructuración de esta información dentro de los modelos digitales permite que estos actúen como una fuente de datos fiable, centralizada y accesible, facilitando la interoperabilidad entre plataformas y agentes involucrados.

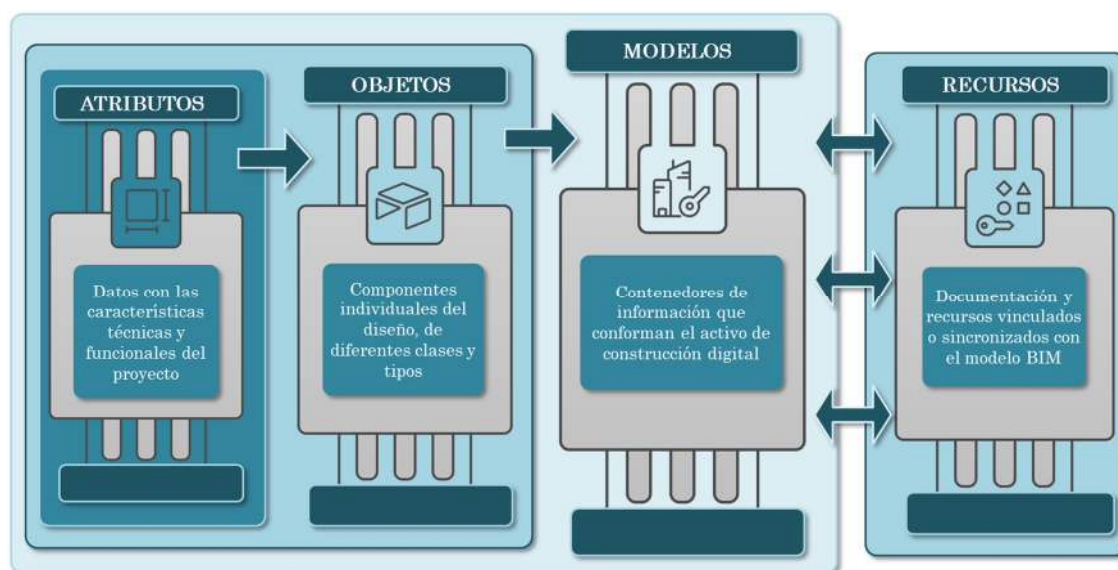


Figura 2.1.12. Estructura de la información BIM

2.1.4.1. Modelos BIM

En el contexto de la gestión digital, es fundamental diferenciar entre distintos tipos de modelos digitales, cada uno con características, funciones y niveles de información particulares. Un modelo 3D representa exclusivamente la geometría tridimensional de un objeto, careciendo de parametrización y de datos adicionales que enriquezcan su contenido. En contraste, un modelo BIM consiste en un modelo tridimensional compuesto por objetos paramétricos, enriquecido con información detallada sobre materiales, especificaciones técnicas y relaciones espaciales entre elementos.

Dentro del entorno BIM, los datos del proyecto se crean, gestionan y almacenan en un contenedor BIM, un modelo de información virtual que, además de incorporar información geométrica, facilita la gestión inteligente de los procesos de construcción. Cada contenedor está compuesto por entidades, sistemas, materiales, espacios y montajes tridimensionales que representan de manera virtual los elementos constructivos que posteriormente serán fabricados y ejecutados en obra. En el contexto de la ISO 19650, un contenedor de información se define como un conjunto de información persistente y recuperable desde un archivo, sistema o aplicación de almacenamiento jerarquizado, que puede contener tanto información estructurada como no estructurada, integrando diferentes tipos de datos y

documentos que conforman la base informativa del proyecto. Así, un modelo BIM es, en esencia, una representación digital de un activo, que no solo almacena datos físicos y geométricos de sus componentes, sino que también integra información no geométrica y establece relaciones entre los distintos elementos que lo conforman. Además, abarca diferentes contenedores de información, tanto gráficos como no gráficos, que incluyen bases de datos, presupuestos, planificaciones, documentación técnica, archivos multimedia y otros recursos para la gestión del proyecto.

A lo largo del ciclo de vida de un activo construido,⁵⁶ es necesario disponer de dos modelos de información esenciales, de acuerdo con la ISO 19650. El primero es el Project Information Model (PIM), que corresponde a la fase de desarrollo y ejecución de la obra. El segundo es el Asset Information Model (AIM), que corresponde a la fase de operación y mantenimiento del activo construido. Ambos modelos están relacionados con los requisitos de información. El PIM se desarrolla a partir del EIR y el AIM, por su parte, está vinculado a los AIR.

Segregación y federación

En el desarrollo de proyectos, es posible concebir una obra como la suma y superposición de distintos sistemas especializados, cada uno de ellos con funciones específicas e interdependientes. Esta concepción sistémica permite abordar el modelado digital desde dos enfoques: la creación de un único modelo integral que contenga todos los sistemas, o bien la elaboración de distintos modelos individuales, cada uno correspondiente a un sistema específico, los cuales se pueden denominar submodelos.⁵⁷

La elaboración de estos submodelos independientes, correspondientes a cada disciplina o sistema, presenta importantes ventajas para la gestión y ejecución de proyectos bajo entornos colaborativos. Cada equipo especializado puede desarrollar su propio submodelo con el grado de detalle necesario, utilizando herramientas específicas y ajustándose a los requisitos de su área de especialidad. Posteriormente, estos submodelos se ponen en común, permitiendo su visualización y análisis conjunto. Esta capacidad de superponer distintos submodelos constituye la base de la federación de modelos, entendida como el proceso mediante el cual múltiples modelos disciplinarios se integran sin alterar su estructura original, preservando la autoría, el nivel de detalle y la especialización de cada uno. Esta federación permite también representar de forma global el proyecto, manteniendo la segregación disciplinaria y la trazabilidad de la información generada por cada equipo.

Además de la división de modelos por disciplina, es común aplicar otros criterios de segregación en función de las necesidades específicas de cada proyecto, como por peso o tamaño del archivo, especialmente en proyectos de gran escala, donde la complejidad y cantidad de información pueden ralentizar el rendimiento de los sistemas de modelado. Al dividir el modelo en secciones manejables, ya sea por zonas geográficas, niveles o sectores constructivos, se optimiza el trabajo y se facilita la colaboración de distintos equipos. Otra estrategia frecuente es la segregación por fases del proyecto, generando modelos específicos para la etapa concreta.

⁵⁶ Introducción a la Serie EN-ISO 19650. buildingSMART Spain. 2021.

⁵⁷ Manual BIM vol. 2. Metodología. Ferrocarrils de la Generalitat Valenciana (FGV). 2020.

Georreferenciación

La georreferenciación en proyectos de edificación BIM es un proceso esencial para garantizar que el modelo se ubique correctamente dentro de un sistema de coordenadas geográficas o cartesianas de referencia, alineado tanto con el entorno real como con otras fuentes de información geoespacial. Esta correcta vinculación espacial es fundamental para asegurar la integración precisa de datos provenientes de distintas disciplinas, así como para la coordinación con infraestructuras existentes, límites de parcela, redes de servicios urbanos y condicionantes topográficos.

En entornos BIM, el proceso de georreferenciación implica definir un sistema de coordenadas de proyecto local y vincularlo con un sistema de coordenadas global, como el sistema UTM (Universal Transverse Mercator), geodésico o el sistema oficial vigente en el emplazamiento.⁵⁸ Esta doble referencia permite que el modelo conserve la precisión geométrica necesaria para el desarrollo detallado de la edificación, al tiempo que asegura su correcta inserción en el entorno territorial. En el proceso, se establecen puntos de control o puntos base compartidos, que actúan como referencias fijas para alinear todos los modelos disciplinares y garantizar la coherencia posicional del federado. Adicionalmente, el empleo de sistemas de coordenadas comunes facilita la interoperabilidad entre plataformas BIM y herramientas de análisis geoespacial (GIS), permitiendo integrar datos ambientales, urbanísticos y de infraestructuras externas (Kralj, 2018).

Fases de proyecto

Las fases comprenden el conjunto de etapas que atraviesa un activo desde su concepción hasta su desmantelamiento, abarcando el diseño, la construcción, la operación, el mantenimiento y, eventualmente, la rehabilitación o demolición. En cada fase, la metodología BIM facilita la integración y evolución de la información, permitiendo que los modelos pasen de contener datos volumétricos y geométricos iniciales a incorporar información técnica, constructiva y de mantenimiento, enriqueciendo progresivamente el modelo y asegurando su utilidad en todo el ciclo de vida del activo.

En la fase de diseño, BIM permite desarrollar un modelo digital coordinado entre disciplinas, donde es posible detectar interferencias y optimizar la solución arquitectónica, estructural y de instalaciones. Este modelo sirve como base para el control de costes y tiempos mediante simulaciones constructivas, lo que mejora la transparencia y facilita la planificación colaborativa (Garber, 2014). Además, durante el diseño detallado, el modelo se convierte en una herramienta fundamental para la verificación normativa y el cumplimiento de estándares de calidad, al tiempo que centraliza toda la información técnica necesaria para la siguiente fase.

Durante la construcción, BIM optimiza la coordinación entre equipos, facilita la gestión de procesos de obra, mejora la seguridad mediante simulaciones y permite un control más preciso de avances, plazos y costes (Utiome, 2015). A través de modelos actualizados y vinculados con sistemas de prefabricación y logística, se agiliza la producción y reduce el desperdicio de materiales, fomentando prácticas

⁵⁸ How to Setup Geo-referencing in a Building or Linear Infrastructure Model. Model Setup IDM. buildingSMART Australasia. V02. 2020.

más sostenibles (N & Padala, 2024). Al finalizar la construcción, el modelo as-built digitaliza el estado final de la obra (Park & Cai, 2017), convirtiéndose en un repositorio clave para la fase siguiente.

En las fases de operación, mantenimiento y rehabilitación, el modelo BIM sirve como base para la gestión integral del activo (Tsay et al., 2023), facilitando la localización de elementos, la programación de inspecciones y mantenimientos, y la integración con sensores IoT para el monitoreo en tiempo real. En proyectos de renovación, el acceso a un modelo actualizado simplifica el diagnóstico y el diseño de las intervenciones. Finalmente, en la fase de demolición, BIM permite planificar la deconstrucción y la gestión de residuos, promoviendo la reutilización de materiales y la aplicación de estrategias de economía circular desde el cierre del ciclo de vida.

Disciplinas y subdisciplinas

Las disciplinas técnicas son áreas clave que abarcan todo el ciclo de vida del activo, desde la concepción hasta su mantenimiento. Las principales dentro de un proyecto de edificación son topografía, arquitectura, estructura, fontanería, electricidad y HVAC. Las disciplinas organizan tanto modelos de información como a los objetos dentro de estos modelos y en muchos de los sistemas de clasificación BIM se definen, como en el caso del SCFClass,⁵⁹ para establecer una organización estructurada de los componentes del proyecto.

Además de la organización por disciplinas, cada una de estas áreas puede desglosarse en subdisciplinas más específicas, como la envolvente del edificio, cimentaciones, iluminación, telecomunicaciones o sistemas de protección contra incendios, cada una con sus propios requisitos y datos técnicos. Mientras que algunas informaciones, como el nombre o código de un elemento, son comunes a todas las disciplinas, cada una requiere datos específicos como parámetros de equipos de instalaciones, cargas estructurales o flujo de aire.

Elementos de referencia

Los elementos de referencia son componentes que permiten establecer una base común de coordenadas, alineación y organización espacial para el desarrollo y la coordinación de los modelos BIM. Entre los más utilizados se encuentran los niveles, las rejillas y los puntos de georreferenciación, cada uno con funciones específicas que garantizan la correcta ubicación y compatibilización de los distintos submodelos que conforman el proyecto de edificación.

Los niveles son planos horizontales virtuales que definen las distintas alturas de un proyecto, como plantas, entresijos, cubiertas o niveles técnicos. Sirven como base para organizar la colocación de elementos constructivos y facilitan la generación de planos de planta y secciones, asegurando la consistencia vertical del modelo. Por otro lado, las rejillas son planos de referencia, generalmente ortogonales o radiales, que permiten subdividir el espacio en módulos o cuadrículas, facilitando la ubicación precisa de elementos. Además de estos elementos básicos, en proyectos complejos pueden incorporarse otros elementos como planos de trabajo personalizados o ejes de alineación específicos para elementos singulares.

⁵⁹ Sistema de Clasificación SCFclass. Railway Innovation Hub.2022.

2.1.4.2. Objetos BIM

Los objetos BIM constituyen la unidad fundamental de modelado, representando los distintos elementos que configuran un modelo digital de un proyecto constructivo. Estos objetos no solo aportan una representación geométrica tridimensional, sino que integran información asociada, necesaria para garantizar la interoperabilidad, la coherencia y la correcta gestión del activo. La estructuración y organización jerárquica de estos objetos es esencial para facilitar la coordinación interdisciplinar, la trazabilidad de la información y la automatización de procesos de análisis y extracción de datos. Además, con ello se consigue relacionar los objetos con sistemas constructivos, espacios funcionales y zonas operativas, estableciendo una red de vínculos semánticos que enriquece la utilidad del modelo.⁶⁰

Una particularidad de los objetos BIM es que no todos los elementos presentes en el modelo virtual tienen un correlato físico directo en la obra construida. Existen objetos cuya función es exclusivamente auxiliar, sirviendo como contenedores de agrupación u organizadores espaciales. Estos objetos, aunque necesarios durante las fases de diseño y modelado, carecen de valor constructivo directo. Por el contrario, determinados elementos físicos de relevancia constructiva y funcional no siempre son modelados de forma explícita debido a limitaciones de las herramientas BIM o a criterios de eficiencia. En estos casos, es necesario establecer estrategias que permitan representar o vincular dichos elementos a través de relaciones indirectas o propiedades adicionales, asegurando que la información esencial quede incorporada al modelo. Estos objetos suelen ser subelementos relacionados con remates, acabados, piezas auxiliares, cables o refuerzos de armadura.

Más allá de la visualización tridimensional, los objetos BIM constituyen la base de un inventario digital que permite identificar, clasificar y gestionar cada elemento modelado. Este inventario, derivado de la información asociada a cada objeto, es clave para el cumplimiento de requisitos de control de calidad, mediciones, planificación, mantenimiento y operación. La extracción de estos datos se realiza mediante herramientas de consulta y generación de listados o tablas, permitiendo su exportación a formatos interoperables para análisis y control externo.

Diversas iniciativas internacionales han trabajado en la estandarización de la creación y gestión de objetos BIM, con el objetivo de garantizar la interoperabilidad, consistencia y reutilización de la información en todas las fases del ciclo de vida de los activos construidos. En el ámbito normativo, destaca la UNE-EN ISO 22014,⁶¹ que establece directrices generales para la definición y gestión de objetos BIM en bibliotecas digitales aplicadas a proyectos de construcción, asegurando su coherencia y adaptabilidad en entornos colaborativos. Complementariamente, iniciativas como el NBS BIM Object Standard (Reino Unido) y el Open BIM Object Standard (OBOS), impulsado por Natspec y Masterpec (Australia y Nueva Zelanda), han desarrollado marcos específicos que definen requisitos de información, propiedades, geometría y comportamiento de los objetos, facilitando su integración en plataformas de trabajo colaborativo.

⁶⁰ BIM Manual. VA BIM Standard. US Department of Veteran Affairs. V2.2. 2017.

⁶¹ UNE-EN ISO 22014:2024. Objetos para librerías usadas en arquitectura, ingeniería, construcción y etapa de uso.

En el contexto español, destacan dos iniciativas relevantes: el eCOB, desarrollado por el Instituto de Tecnología de la Construcción (ITeC), y el GDO-BIM, promovido por Bimetrica. El estándar eCOB se centra en la generación de objetos BIM alineados con la normativa nacional, incluyendo el Código Técnico de la Edificación (CTE), y con una estructura de información compatible con el formato IFC, garantizando su uso a lo largo de todo el ciclo de vida. Por su parte, el GDO-BIM proporciona una guía orientada principalmente a fabricantes, basada en el análisis de los parámetros más utilizados en la industria española, con el fin de facilitar la creación de objetos coherentes y adaptados a las necesidades de los profesionales y empresas del sector de la construcción.

Algunas de estas iniciativas, como el OBOS, se complementan con herramientas adicionales, como el NATSPEC BIM Properties Generator, que ofrece un repositorio de propiedades estandarizadas para facilitar la correcta parametrización de los objetos, fortaleciendo la coherencia de los modelos y su valor como fuentes fiables de información.

Entidades paramétricas

Dentro del software BIM, los objetos se generan como entidades paramétricas, es decir, elementos virtuales que representan componentes físicos o abstractos del proyecto, definidos tanto por su geometría como por conjuntos de parámetros que controlan su comportamiento y propiedades dentro del modelo. La parametrización (Warren, 2019) es el proceso mediante el cual cada entidad recibe un conjunto estructurado de datos y propiedades que la definen. Estos parámetros pueden describir dimensiones, materiales, propiedades físicas, relaciones espaciales, requisitos funcionales, o cualquier otro dato. Los programas BIM permiten que estas propiedades sean dinámicas (Wahbeh, 2017), de modo que cualquier modificación en el valor de un parámetro puede actualizar automáticamente la geometría de la entidad, o viceversa, asegurando la coherencia global entre información gráfica y no gráfica. Este flujo de información también garantiza la consistencia entre las diferentes representaciones del proyecto, como planos, tablas de propiedades y listados de cantidades, y facilita el control de calidad y la gestión de cambios a lo largo de todas las etapas del proyecto.

Este dinamismo se consigue a partir del establecimiento de diferentes relaciones y restricciones entre las entidades, lo cual aumenta la inteligencia del modelo. Las relaciones son esenciales para garantizar la consistencia, especialmente cuando se gestionan elementos que deben ajustarse entre sí, como puertas y ventanas en muros o instalaciones que dependen de la ubicación de los sistemas estructurales. Por otra parte, las restricciones pueden definirse en términos de dimensiones, distancias mínimas o máximas, bloqueos de cotas, ángulos específicos o incluso a partir de fórmulas condicionales relacionando diferentes parámetros, lo que asegura que los elementos se comporten de acuerdo a los requisitos técnicos del proyecto. Muchas de estas relaciones son consideradas asociaciones entre entidades, como en el caso de los objetos y los niveles, en donde se define la posición vertical del objeto dentro del edificio. Otra asociación es la que establecen los objetos y el espacio en el que están contenidos, dando lugar a relaciones adicionales acerca de características del espacio, como el control de la temperatura, la iluminación o el flujo de aire.

Sistemas, grupos y montajes

Los objetos no se gestionan únicamente de forma individual, sino que a menudo son agrupados y relacionados en conjuntos según propiedades compartidas o su pertenencia a un mismo conjunto constructivo. Esta organización permite estructurar el modelo en diferentes niveles de jerarquía, añadiendo capas adicionales de información que facilitan el análisis y el control de los elementos modelados. En el caso de instalaciones, se agrupan los elementos de entrada y salida, así como los de conexión de la instalación. Cada uno de ellos mantiene sus propiedades específicas, pero al asociarse a un mismo sistema, heredan propiedades comunes, como el tipo de fluido, el caudal o la presión. Esta agrupación permite realizar simulaciones de funcionamiento y verificaciones de cumplimiento normativo, ya que permite un estudio conjunto del sistema de la instalación. Además, pueden organizarse en subsistemas, como el tipo de suministro o retorno, el ámbito espacial, el fluido transportado o la función concreta dentro del edificio.

Por otro lado, en proyectos de gran escala o con elementos prefabricados, es común la creación de grupos o montajes, que consisten en la combinación predefinida de varias entidades que, por funcionalidad o por eficiencia de modelado y ejecución, se gestionan como una unidad única. Por ejemplo, una unidad modular prefabricada puede modelarse como un grupo, que integra instalaciones, tabiquería, mobiliario y acabados. Este enfoque agiliza la repetición sistemática de componentes, reduce errores y facilita su coordinación con otros elementos del proyecto.

Categorías de objetos

La organización de las entidades dentro de los softwares BIM se basa en un sistema jerárquico de categorías o clases que agrupan los objetos según su función, naturaleza y uso dentro del modelo. Estas categorías permiten clasificar de forma estructurada los elementos que componen el proyecto, facilitando la gestión, consulta y exportación de datos. Las categorías principales agrupan objetos según su función básica dentro del proyecto, diferenciando entre categorías arquitectónicas (muros, suelos, cubiertas...), estructurales (vigas, pilares, cimentaciones...), de instalaciones (tuberías, conductos, luminarias...) o auxiliares (líneas de referencia, anotaciones, símbolos...). Esta segmentación permite que los programas apliquen comportamientos específicos a cada grupo, como reglas de visualización, propiedades predefinidas o relaciones automáticas entre objetos. Además, dentro de muchas de estas categorías, especialmente en aquellas relacionadas con elementos complejos, es habitual encontrar subcategorías que permiten una clasificación más precisa y detallada, como en el caso de las escaleras, que pueden dividirse en tramos, descansillos y soportes, cada uno con su propio comportamiento y conjunto de propiedades específicas, pero manteniendo su relación funcional como parte de un único sistema constructivo. De manera similar, otras categorías como los muros pueden diferenciar entre muros de carga, muros por capas o muros cortina, los cuales pueden contener subcategorías de montantes, rejillas, puertas, ventanas y/o paneles fijos. Además, las categorías son clave en la parametrización de objetos, ya que cada categoría hereda un conjunto de propiedades comunes que se aplican de forma automática a los objetos que contiene.⁶²

⁶² Manual del usuario Revit. Autodesk. 2011.

Tipos e instancias

Sumada a las categorías, los objetos contienen una segunda estructura jerárquica basada en tipos, que es representada por un conjunto de entidades que comparten una serie de características comunes, sirviendo como plantilla paramétrica para la generación de múltiples instancias. Cada vez que se coloca un objeto basado en un tipo, se crea una instancia, es decir, una manifestación individual y específica de ese tipo dentro del modelo. La relación entre tipos e instancias proporciona un doble nivel de control. Por un lado, el nivel de tipo permite realizar modificaciones globales que afectan a todas las instancias derivadas del tipo, asegurando consistencia y agilizando cambios de diseño o actualización de propiedades comunes. Por otro lado, el nivel de instancia mantiene atributos únicos, en una ubicación exacta, con orientaciones particulares y relaciones específicas con otras entidades.⁶³

Además de lo anterior, los objetos BIM no solo representan elementos sólidos, sino que también incluyen volumetrías vacías, como espacios y zonas. Estos volúmenes vacíos son entidades igualmente paramétricas que definen áreas funcionales, condiciones de confort, requisitos técnicos o relaciones espaciales, y pueden asociarse a sistemas de ventilación, evacuación o seguridad, entre otros. La coexistencia de estos dos tipos de volumetrías permite que los modelos BIM representen el objeto construido y, a la vez, su funcionalidad y la interacción entre espacios y sistemas, lo que resulta fundamental en análisis energéticos, simulaciones ambientales y coordinación de disciplinas.

Los objetos BIM, ya sean tipos o instancias, pueden clasificarse en genéricos o específicos según el grado de definición y vinculación con productos comerciales reales. Los objetos genéricos no están asociados a un fabricante concreto y se utilizan principalmente en fases tempranas del diseño, cuando aún no se han seleccionado los productos definitivos. Estos objetos se desarrollan siguiendo especificaciones normativas, requisitos funcionales o condiciones de proyecto, garantizando un nivel básico de información suficiente para el desarrollo general del modelo. Por su parte, los objetos específicos son aquellos proporcionados por fabricantes o proveedores concretos, incorporando información técnica real y detallada.

En cuanto a cada una de las instancias generadas dentro de un modelo, estas poseen un identificador único, esto es, un código alfanumérico irrepetible dentro del entorno digital que garantiza la trazabilidad de cada elemento. Este identificador es esencial para asegurar que cada objeto pueda ser identificado de forma unívoca, independientemente de las modificaciones que pueda sufrir el modelo.

2.1.4.3. Atributos BIM

Los atributos son unidades de información asociadas a las entidades BIM, que proporcionan detalles esenciales sobre las características específicas de cada elemento. Funcionan como contenedores de información destinados a almacenar un tipo de dato predefinido, asignando un valor a cada objeto. Son necesarios para la gestión y manipulación de datos, ya que permiten que la información vinculada sea fácilmente accesible, modificable y reutilizable durante el proyecto. Cada atributo puede estar asociado a un objeto individual, a un sistema o montaje, a un tipo o categoría específica, o ser global para todo el modelo, dependiendo de su naturaleza.

⁶³ Estándar de creación de objetos BIM (eCOB). ITeC. V01. 2021.

De igual forma, los atributos pueden dividirse entre los específicos, que son de aplicación a tipologías concretas, y los generales o comunes, que aplican a todas las tipologías y categorías de objeto y/o modelo.

Un aspecto relevante en la estructura de atributos es la definición de descripciones, las cuales proporcionan una explicación clara del tipo de información que debe albergar el atributo, permitiendo una mejor comprensión de su función en el modelo. Además, los comentarios asociados a los atributos pueden proporcionar detalles adicionales, consideraciones específicas sobre su uso o sobre la fuente de datos de origen de la información, indicando si debe ser tomada de tablas preestablecidas, proporcionada por terceros o derivada de cálculos automatizados dentro del modelo.

Sistemas de clasificación

Los sistemas de clasificación⁶⁴ surgen de la necesidad de estructurar y caracterizar los datos y de que todas las partes interesadas puedan localizarlos, entenderlos y organizarlos de manera única e inequívoca. Estos sistemas se establecen como una referencia para que todos los conceptos utilizados en los distintos modelos y documentos digitales puedan estar categorizados. Es una forma de agrupar y organizar elementos para una finalidad concreta.

Clasificar permite estandarizar hablando un mismo idioma, organizar elementos en base a un criterio dado. La taxonomía es la ciencia que se ocupa de los principios, métodos y fines de una clasificación. Se aplica para la ordenación jerarquizada y sistemática de un entorno, definiendo los nombres y un lenguaje preciso de los grupos de elementos a clasificar. Las clasificaciones no cuentan con la flexibilidad del lenguaje hablado, es decir: un concepto debe quedar definido inequívocamente por un único texto o código. Y no puede contemplar sinónimos para dicho concepto en clases diferentes. El proceso de clasificación requiere la determinación de características comunes entre objetos, que quedarán adscritos a una clase o tipo. Las agrupaciones de objetos conformarán clases o tipos que definirán la estructura desglosada de una clasificación.

Las clasificaciones más habituales son las de tipo árbol y se diferencian por niveles, donde existe un tipo de jerarquía en función de las relaciones de clases “padre” (niveles superiores) con clases “hijo” (niveles inferiores) y clases “hermanas” (mismo nivel). Es posible clasificar los objetos por todo aquello que tiene que ver con su forma: clasificaciones morfológicas (tipología, material, tamaño...), o bien clasificarlos por lo que significan: clasificaciones semánticas, es decir, por toda aquella información más allá de su morfología (actividades, estados, función...). Cabe la posibilidad de combinar criterios en un determinado orden en las llamadas clasificaciones combinadas.

Cuando se trata de clasificaciones digitales, cada nivel y clase descrita viene acompañado por un código, lo que hace que el sistema sea más preciso y legible tanto para máquinas como para humanos. Estos códigos deben ser inteligibles, es decir, comprensibles y asimilables por quien los lea; lógicos, ya que la codificación de clases y niveles debe seguir un criterio de crecimiento natural y universal (como numeración ascendente con saltos de 5, 10 o 100, o en orden alfabético o temporal);

⁶⁴ Guía de sistemas de clasificación cuando se utiliza BIM. 2022 buildingSMART Spanish Chapter. <https://www.buildingsmart.es/recursos/clasificaci%C3%B3n-en-los-modelos/>

e inequívocos, lo que significa que no debe haber lugar a dudas ni equivocaciones, y no debe existir más de una clase con el mismo código ni dos códigos que representen la misma clase. Además, en un proceso de trabajo colaborativo, sirve de referencia para asegurar la uniformidad de los nombres utilizados y para informar de los diferentes modelos de datos y documentos digitales. De este modo, permite garantizar la interoperabilidad y, por lo tanto, la durabilidad de la información.

En el sector AECO, los proyectos y obras de construcción también acaban resultando muy complejos en prestaciones, funciones, procesos constructivos, gestión o mantenimiento. Por todo ello el sector requiere cada día más, que los objetos que componen las obras se puedan clasificar bajo múltiples criterios.

En la actualidad, existen numerosos sistemas de clasificación (Unifomat, Uniclass, Omniclass, SCFClass, etc.), con diferente alcance y criterio de clasificación, por lo que la elección de este debe corresponderse a las necesidades de las partes interesadas, de modo que se facilite el intercambio colaborativo de información a lo largo de todas las fases del ciclo de vida de un activo construido y su entorno. Cabe señalar que es posible superponer varios sistemas de clasificación, de modo que pueda utilizarse cada uno de ellos con diferentes objetivos. Por ejemplo, una misma viga estructural se puede codificar en base a Omniclass 21 como “21-02 10 10 10 Floor Structural Frame”, en base a Uniclass 2015 como “Ss_20_20 Structural beams” y en base a GuBIMClass como “20.20.20.40 Viguetas”. Por otro lado, como ocurre en el caso del SCFClass⁶⁵, también un mismo sistema puede contener diferentes tablas para asignar clasificaciones según la pregunta a la que responden:

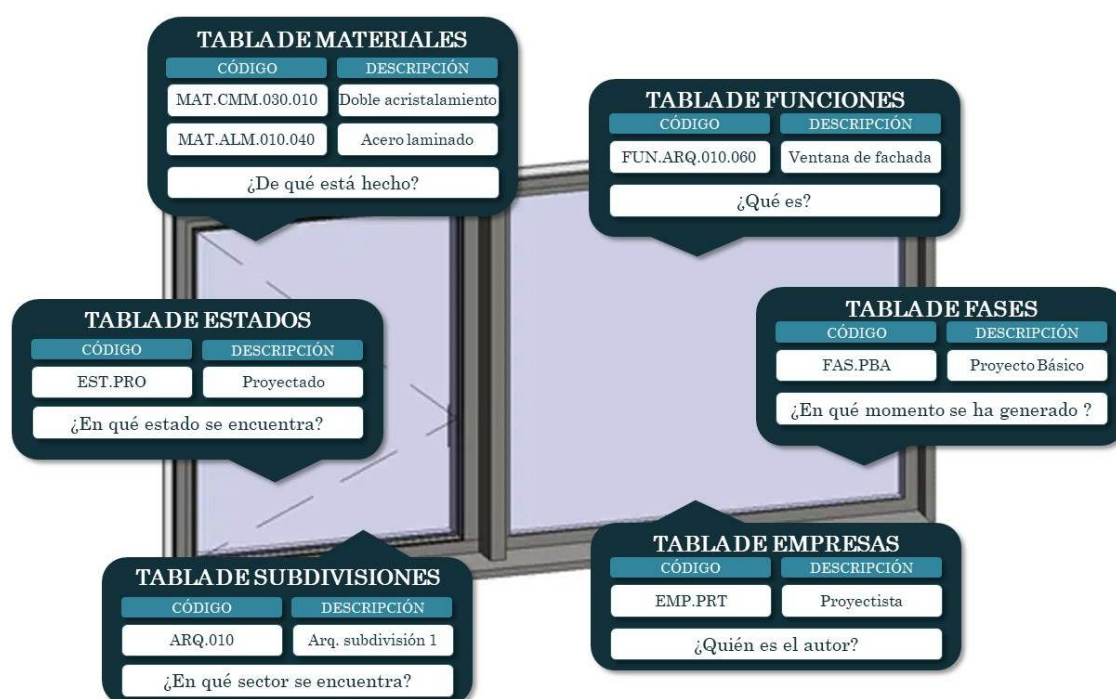


Figura 2.1.13. Códigos de clasificación según tablas del SCFClass v2.0.

⁶⁵ Sistema de clasificación BIM SCFClass del Railway Innovation Hub. 2022.
<https://www.railwayinnovationhub.com/bim/>

Múltiples sistemas pueden convivir para dar respuesta a la suma de necesidades de todos los actores en base al contexto y condiciones de contratación. Incluso, en algunos casos, una misma entidad puede ser clasificada por múltiples códigos de una misma tabla ya que, por ejemplo, puede estar formada por diferentes materiales.

La relevancia de estos sistemas de clasificación incrementa al utilizar la metodología BIM debido a que existe una necesidad por parte de los agentes involucrados de definir la caracterización y tipificación de los objetos, además de establecer una serie de requisitos de información para cada uno de estos grupos de objetos. Si BIM es una base de datos, la clasificación es un criterio de consulta de información de la base de datos. Además, en la norma UNE-EN ISO 7817⁶⁶ se recomienda la utilización de los sistemas de clasificación para disponer un lenguaje de especificación de necesidades de contenido de información de los elementos que forman parte del modelo BIM. El problema que existe con sistemas anteriores a la llegada de la metodología BIM, es que estos no contemplan la organización por objetos, conforme al esquema de clases IFC de la norma UNE-EN ISO 16739-1⁶⁷, sino que en la mayoría de los casos se agrupan por el sistema constructivo.

A nivel de clasificación, estos sistemas definen códigos y descripciones para elementos tangibles (como un muro o una puerta), intangibles (como un espacio de operación o una zona de una vivienda) y para otras entidades abstractas que agrupan distintos elementos (como un modelo BIM o un emplazamiento).

Las tablas que organizan los objetos por Función suelen ser las más solicitadas por promotores ya que son las más utilizadas para gestionar los objetos de construcción. Estas tablas, como la del sistema GuBIMClass⁶⁸, pueden ayudar a buscar, a entender, a explicar partes del proyecto, también a estructurar cómo realizar un estudio de colisiones, o a determinar qué se modelará o qué no y decidir las propiedades o la unidad de medición previstas para cada clase o tipo de objetos.

En los últimos años, debido a la necesidad de encontrar estructuras afines a las diferentes tipologías de proyectos en España, se han desarrollado nuevos sistemas de clasificación además de los mencionados anteriormente como el RCEClass⁶⁹, el FGVclass⁷⁰ o el AeasBIMClass⁷¹. No obstante, el nivel de exigencia de estos sistemas de clasificación dentro de los requisitos BIM de pliegos de administraciones públicas en el sector de la Edificación, sigue siendo muy bajo, alcanzando poco más del 25% del total de licitaciones en los últimos 7 años⁷².

⁶⁶ Norma UNE-EN ISO 7817-1:2025. Nivel de información necesario. Parte 1: Conceptos y principios. (ISO 7817-1:2024). Digitalización de la información para edificación y obra civil.

⁶⁷ Norma UNE-EN ISO 16739-1:2024. Intercambio de datos en la industria de construcción y en la gestión de inmuebles mediante IFC (Industry Foundation Classes).

⁶⁸ Sistema de clasificación de elementos de construcción de acuerdo con su función principal dentro de un entorno BIM. 2017. GUBIMCAT (Grupo de usuarios BIM ubicado en Cataluña).

⁶⁹ Sistema de clasificación multicriterio de la Red de Carreteras del Estado, de aplicación tanto a los elementos 3D de los modelos BIM como al nombrado de los archivos. 2024. Gobierno de España.

⁷⁰ Sistema de clasificación de Ferrocarrils. 2020. Generalitat Valenciana.

⁷¹ Sistema de clasificación BIM Aeas. 2023. Asociación Española de Abastecimientos de Agua y Saneamiento.

⁷² Observatorio de Licitaciones Públicas BIM. CIBIM. Gobierno de España. Datos extraídos desde 2017 hasta 2023. Actualización del 28/12/2023.

Nivel de información necesario

El concepto de Level of Development (LOD, por sus siglas en inglés) ha sido crucial para la estandarización de la información en los modelos BIM a nivel mundial. Su origen data de 2004, cuando Vico Software introdujo la Model Progression Specification (MPS), orientada a mejorar la gestión de datos dentro de los modelos BIM. Inicialmente, LOD hacía referencia al Level of Detail, enfocado en el progreso de la información geométrica del proyecto. Más tarde, en 2008, el American Institute of Architects (AIA) adoptó este concepto en un nuevo protocolo,⁷³ estableciendo la clasificación de niveles LOD 100, 200, 300, 400 y 500.

A lo largo de los años, diversas organizaciones internacionales adaptaron el concepto de LOD. En 2013, BIMForum publicó su propia especificación LOD, basada en los protocolos del AIA, convirtiéndose en un referente para guías y normativas internacionales. En paralelo, el Reino Unido, a través de la PAS 1192-2:2013, introdujo el término Level of Definition, diferenciando el Level of Detail (LOD), relacionado con la geometría, y el Level of Information (LOI), centrado en la información no gráfica del modelo.



Figura 2.1.14. Niveles de información BIM. Fuente* (BIMForum)

Con la publicación de la ISO 19650, el concepto de LOD evolucionó hacia el Level of Information Need (LOIN). A diferencia del modelo tradicional, este enfoque permite definir los requisitos de información en función de los objetivos específicos de cada fase del proyecto, personalizando los criterios según las necesidades del ciclo de vida del activo. Sin embargo, la implementación de este nivel sigue en proceso de adaptación, coexistiendo con los esquemas tradicionales de LOD en muchos proyectos y licitaciones (Utiome, 2015).

Además de la ISO 19650, en 2024 se publicó la ISO 7817, que aborda de manera más detallada los niveles de información necesaria, estableciendo conceptos clave necesarios para especificar las necesidades de información de manera consistente, definiendo las características de los distintos niveles de información y especificando

⁷³ E202-2008: Building Information Modeling Protocol Exhibit.

el grado de detalle y extensión de la información que debe intercambiarse y entregarse a lo largo del ciclo de vida de los activos construidos. Esta norma diferencia los conceptos de información geométrica e información alfanumérica, donde el primero está relacionado con el detalle gráfico de los objetos del modelo, incluyendo detalle, ubicación, dimensionalidad, apariencia y comportamiento paramétrico, y el segundo, que se refiere a la información no gráfica de los elementos, tanto la identificación como el contenido de información. Esto se debe a que los modelos pueden tener un bajo nivel de detalle gráfico, pero contener gran cantidad de información no gráfica, como metadatos, y fórmulas o, al revés, tener altos grados de detalle gráfico pero escasa información.⁷⁴ Adicionalmente a estos conceptos, se desarrolla un tercero relacionado con la documentación asociada a los objetos y modelos de información.

Conjuntos de atributos

Dentro de los contenedores de información, los datos se estructuran y encapsulan a diferentes escalas para garantizar un orden claro de la información y también para luego poder buscarlos o trabajarlos de forma conjunta. Estas agrupaciones forman conjuntos de propiedades que responden a una finalidad común. Cada grupo recibe un nombre específico y su denominación suele seguir un esquema estandarizado basado en el código de grupo y el nombre del conjunto de propiedades. Esta nomenclatura permite estructurar la información de manera ordenada dentro de los modelos BIM, facilitando la integración de los atributos en distintas plataformas y herramientas de modelado.

Tipos de atributos

Los atributos constituyen unidades de información asociadas a cada elemento modelado y permiten describir sus características, facilitando su identificación, análisis y posterior gestión. Dentro del entorno BIM, la primera distinción fundamental que se establece para atributos es según su alcance y aplicabilidad. Existen atributos generales, que son comunes a todos los objetos del modelo, y atributos específicos, definidos para tipologías particulares o disciplinas concretas. Los atributos generales incluyen, entre otros, identificadores globales, nombres de familia o tipo, clasificación en sistemas de codificación, así como datos relativos al estado, autoría o fecha de creación y modificación. En cambio, los específicos responden a las particularidades funcionales o técnicas de cada elemento, atendiendo a su función constructiva o técnica.

De igual forma, los atributos también se organizan por el tipo de dato o formato de la información que contienen. Esta tipología permite la correcta interpretación, intercambio y explotación de los datos del activo, facilitando su integración entre diferentes softwares. Entre los más comunes se encuentran los atributos de texto, que contienen información alfanumérica libre o estructurada, los atributos numéricos, que incluyen valores enteros, decimales o fraccionarios, los atributos booleanos, que representan variables lógicas binarias con valores de verdadero o falso y otros tipos más avanzados, que albergan datos complejos o compuestos, como matrices, tablas o referencias cruzadas, que vinculan los objetos con otras bases de datos o sistemas externos.

⁷⁴ BIM Standards. Construction Industry Council (CIC). Hong Kong. V02. 2021.

De la misma forma, existen diferentes atributos que se organizan en base al uso BIM al que están asociados y dependiendo también de los niveles de información y fases de proyecto que se definan en el alcance del proyecto.⁷⁵



Figura 2.1.15. Tipos de atributos en entornos de construcción digitales

Valores de atributos

Los valores de atributos son los datos que describen las propiedades específicas de objetos. Cada dato relacionado con un valor de cálculo, dimensión, área o cantidad específica se asocia a una unidad de medida estándar que debe ser definida en las fases iniciales de proyecto, permitiendo así que todos los equipos del proyecto utilicen las mismas magnitudes y unidades y facilitando la comprensión y la interoperabilidad entre los distintos actores. A partir de una correcta gestión de unidades de medida, los datos pueden ser fácilmente manipulados, comparados y analizados, y se evitan problemas derivados de conversiones erróneas o ambigüedades en la interpretación (Lou et al., 2020).

En relación a las tolerancias, los valores asignados a los atributos deben tener en cuenta ciertos márgenes de error o variabilidad. Las tolerancias definen el rango dentro del cual un valor puede variar sin que esto afecte el desempeño, la seguridad o la funcionalidad de un elemento del proyecto. En muchos casos, las tolerancias pueden ser específicas para diferentes disciplinas o bien para distintos tipos de modelo, como modelos de entorno, urbanísticos, de anteproyecto, etc.

⁷⁵ Guía de desarrollo de objetos BIM. Bimética. V01, 2019.

Las restricciones de valores también juegan un papel esencial en la gestión de atributos, ya que establecen qué valores son válidos dentro del sistema. Estos pueden estar determinados por criterios técnicos o normativos, y ayudan a evitar que se ingresen datos incorrectos o fuera de contexto. Además, se definen para asegurar la calidad y consistencia de la información, limitando el rango de posibles entradas a aquellos que son físicamente y operativamente posibles. También en muchos casos, estos valores pueden estar sujetos a patrones predefinidos para garantizar consistencia en el uso de mayúsculas, caracteres especiales o estructuras jerárquicas.

2.1.4.4. Recursos BIM

La metodología BIM no solo implica la generación de un modelo, sino que se apoya en un ecosistema de recursos de entrada y de salida de información que garantizan la eficiencia, interoperabilidad, calidad de los datos y control del proyecto. Estos recursos incluyen desde plantillas de trabajo, bibliotecas de objetos y rutinas de automatización, hasta archivos vinculados y extracciones de información. Su integración optimiza la trazabilidad y la transferencia de información entre fases y agentes.

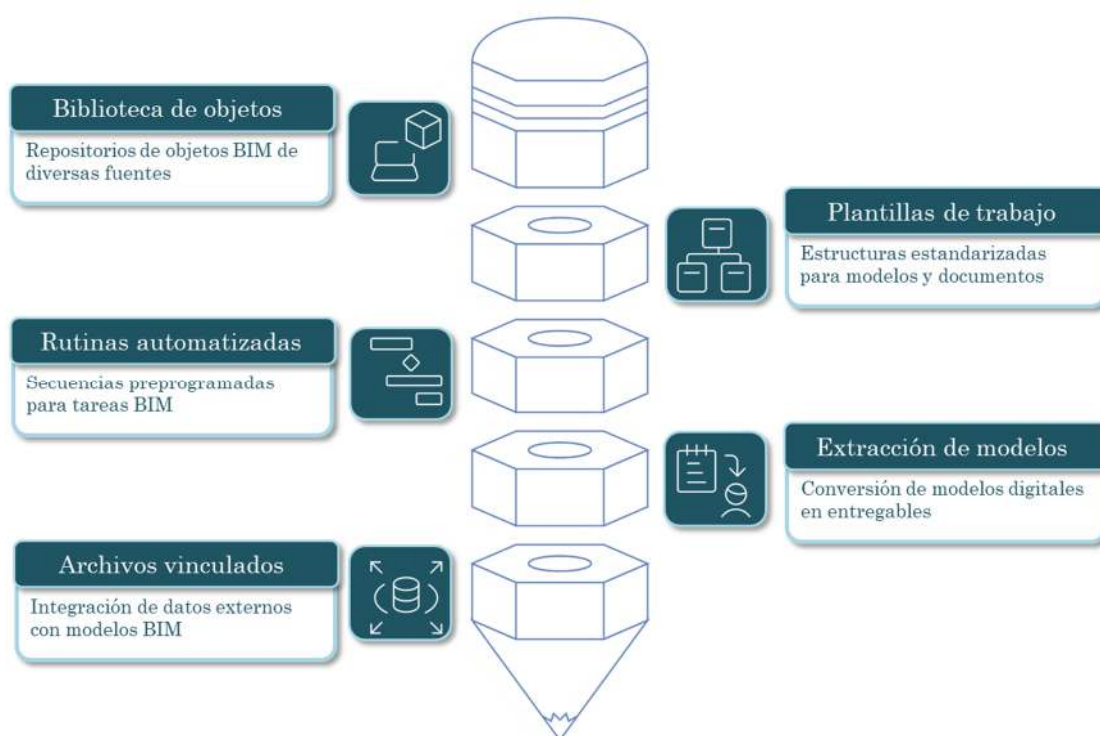


Figura 2.1.16. Recursos asociados a modelos BIM

Bibliotecas de objetos

Las bibliotecas de objetos facilitan la creación y gestión de modelos digitales. En la actualidad, existen diversas fuentes de obtención de objetos BIM, cada una con sus características particulares (Kim et al., 2015). Por un lado, los programas BIM proporcionan bibliotecas predeterminadas de objetos básicos para proyectos estándar, aunque no siempre cumplen con las necesidades del proyecto, ya que están diseñados para cubrir una amplia gama de situaciones y no siempre incluyen propiedades necesarias para cumplir con normativas específicas. Por otro lado, las

empresas fabricantes de productos también juegan un papel importante, ya que muchas ofrecen sus propios objetos para facilitar su inclusión en proyectos. En este caso, los objetos se diseñan específicamente para representar sus productos, incluyendo datos detallados sobre las características del producto. También existen organizaciones que proporcionan objetos genéricos sin referencia comercial. Estos objetos son útiles en las fases iniciales del proyecto, cuando aún no se ha decidido el producto comercial que se empleará en la obra.

Estas bibliotecas de igual forma pueden ser creadas por los propios usuarios, generando la parametrización de los objetos y definiendo la información necesaria en cada caso. Esta opción permite la creación de objetos específicos, ajustados a las necesidades constructivas, estéticas o funcionales del proyecto, aunque requiere habilidades avanzadas en el uso de las herramientas de modelado y una inversión de tiempo considerable. No obstante, la diversidad de plataformas de modelado que requieren adaptaciones específicas de los objetos y las diferentes exigencias normativas entre países complican la integración de los objetos BIM en los modelos.

Plantillas de trabajo

La implementación de plantillas de trabajo en el flujo de proyectos BIM representa un factor clave para asegurar la uniformidad de procesos de modelado y gestión. Estas plantillas no solo definen la estructura inicial de los documentos o modelos, sino que también pueden incorporar parámetros específicos, metadatos de proyecto o configuraciones personalizadas que garantizan la correcta alineación con los requisitos de gestión digital.

En el caso de documentos, las plantillas se utilizan para sentar las bases de la información exigida, así como para la definición de la estructura de cada documento, asegurando la trazabilidad documental y la correcta comunicación entre agentes. Estas plantillas suelen generarse para documentos como el EIR, el BEP, el MIDP y también para otros archivos de apoyo como actas, matrices de interferencias, matrices de atributos, revisiones, etc. Algunas organizaciones nacionales e internacionales impulsan desde hace años el uso de este tipo de plantillas, como la Comisión esBIM de España, el NCS⁷⁶ o el PennState,⁷⁷ ambos de Estados Unidos.

En paralelo, las plantillas de modelos BIM definen la estructura digital de cada modelo según aspectos como la disciplina, la fase o el uso de aplicación. Abarca tanto configuraciones de plantillas de inicio de software, como ajustes gráficos, definiciones de vistas, filtros de exportación, precarga de objetos en modelos, configuraciones de intercambio de datos, controles de calidad, etc. Estas plantillas actúan como guías que permiten estandarizar la producción de los modelos de información.

Rutinas de automatización

Las rutinas son secuencias de instrucciones preprogramadas que permiten la ejecución automática de tareas repetitivas, el control avanzado de modelos digitales y la generación dinámica de información, adaptándose a diferentes plataformas y lenguajes de programación visual o textual. Estas rutinas reducen el margen de error

⁷⁶ BIM Implementation. National CAD Standard (NCS). V06.

⁷⁷ Planning Guides for BIM Implementation. National BIM Standard. PennState College of Engineering's.

en las configuraciones y aseguran el cumplimiento de diferentes aspectos de los modelos BIM. Además, los procesos que aplican estas rutinas, si fueran realizados de forma manual, serían altamente ineficientes y susceptibles a errores.⁷⁸

Algunos de sus principales empleos son la generación de modelos base, donde un script puede crear la estructura inicial de un modelo a partir de parámetros predefinidos, la gestión y validación de metadatos, realizando auditorías masivas de parámetros y detectando inconsistencias en los modelos, la detección, registro y categorización de interferencias de forma automatizada, la extracción de información desde el modelo, generando tablas, listados o informes de datos o el estudio del diseño generativo, en el caso de las primeras etapas del proyecto.

Desde la perspectiva de gestión de proyectos, estas automatizaciones BIM mejoran la productividad operativa, incrementan la consistencia y calidad de los modelos y permiten que procesos analíticos complejos sean parametrizados y ejecutados de forma homogénea, asegurando que los resultados sean comparables y verificables en diferentes proyectos o contextos (Oesterreich & Teuteberg, 2016).

Extracciones de modelos

La extracción de información a partir de los modelos de información constituye un proceso técnico, cuyo objetivo es transformar el modelo digital maestro en un conjunto de entregables adaptados a los diferentes usos de información BIM. Principalmente las extracciones se realizan con el objeto de generar cortes de planos 2D del modelo o nuevos modelos con determinada información en formatos específicos como el IFC o el DXF. Además, la extracción también abarca la generación de mediciones cuantitativas basadas en la geometría y los metadatos alfanuméricos de los objetos BIM, lo que permite la obtención de presupuestos paramétricos y listados de cantidades directamente vinculados al modelo digital.

Otras extracciones comprenden la generación de material audiovisual, como animaciones, recorridos virtuales y simulaciones 4D, útiles para la validación visual y la comprensión multidisciplinar del proyecto o la exportación de informes de interferencias, donde se documentan las incidencias detectadas, la disciplina responsable y la propuesta de resolución. Por otro lado, los registros de auditoría también pueden ser extraídos el objetivo de documentar quién, cuándo y qué modificaciones se realizaron sobre el modelo, facilitando el cumplimiento de requisitos contractuales, auditorías de calidad y procesos de certificación BIM.

Archivos vinculados

Los archivos vinculados asocian información proveniente de distintas fuentes de datos a los modelos BIM sin comprometer la integridad de la información original. Dentro de estos procesos de vinculación, es posible distinguir entre vinculaciones simples y vinculaciones bidireccionales. En las primeras, el archivo externo se conecta al modelo únicamente como una referencia visual o informativa, sin que exista un intercambio de datos activo entre ambos. Algunos ejemplos usuales son planos CAD, nubes de puntos, manuales de mantenimiento y fichas técnicas. En estos casos, el contenido no influye directamente en las propiedades o atributos de los objetos BIM, simplemente se trata de información relacionada con recursos

⁷⁸ Dynamo Language Guide. Autodesk. V02. 2023.

externos. En cambio, en las segundas, se establece un flujo de datos bidireccional entre el modelo y el archivo externo, permitiendo que las modificaciones realizadas en el modelo actualicen información externa y, a su vez, que ajustes realizados en esas plataformas externas actualicen propiedades de los objetos en el modelo. Este tipo de vinculación es habitual en procesos integrados de costes y planificaciones, donde cada objeto del modelo contiene parámetros que se sincronizan con bases de datos o aplicaciones externas.

También en este contexto es habitual la conectividad de los modelos con tablas y bases de referencia externas, que permiten enriquecer los modelos con garantía de que los valores de atributos que se definen son los adecuados (Lawrence et al., 2014). Algunos ejemplos son las bibliotecas de materiales, las tablas de dimensionado paramétrico de objetos, las tablas de sistemas de clasificación y las tablas de consulta de datos. Estas conexiones, que integran atributos paramétricos o metadatos en los modelos, aseguran la consistencia entre la documentación técnica y la ejecución y operación del activo.

2.2. Estado del arte

2.2.1. Avances y situación actual de BIM

La adopción de la metodología BIM ha sido impulsada no solo por la mejora en la colaboración y eficiencia en los proyectos, sino también por el papel clave que han jugado las administraciones públicas al incorporar requisitos BIM en licitaciones y proyectos de obra pública, promoviendo así su implantación a nivel global (Lopez, 2017). Sin embargo, este proceso de transformación digital ha supuesto un importante reto para muchas empresas, que han optado por externalizar parte de los trabajos BIM debido a la complejidad técnica y al cambio metodológico que implica su implementación (Fountain & Langar, 2018).

En el caso de España, la metodología comenzó a ganar presencia a partir de 2010, impulsada por la demanda de algunas empresas en grandes proyectos internacionales. En 2011, se creó el Comité Técnico de Normalización AEN/CTN 41/SC 13 "Organización de modelos de información relativos a la edificación y la obra civil", un comité espejo del internacional ISO/TC 59/SC13, con el objetivo de desarrollar normas y estándares sobre BIM.⁷⁹

En paralelo, se fundó el capítulo español de buildingSMART, conocido como buildingSMART Spanish Chapter (bSSCH), que agrupa a diversos representantes del sector para promover el uso de BIM basado en estándares de datos y procesos.

Con el fin de alinear las estrategias de implantación BIM por parte de los diferentes países que componen la Unión Europea, en 2012 se constituye el denominado EUBIM Task Group, el cual integra representantes de diferentes países europeos y fue el encargado de desarrollar el Manual para la introducción de la metodología BIM por parte del sector público europeo.

En 2014, bSSCH publicó las guías uBIM, orientadas a distintos usos de BIM (modelo arquitectónico, modelo estructural, eficiencia energética, etc.), y el Spanish Journal of BIM, la primera publicación científica sobre esta metodología en España e Hispanoamérica.

En 2015, se llevó a cabo el primer European BIM Summit en Barcelona, que concluyó con la firma del Manifiesto BIMCAT por parte de varias administraciones públicas catalanas. Este documento estableció una hoja de ruta para la adopción de BIM en proyectos de obra pública, con la meta de exigir su uso en todas las obras públicas a partir de 2020. También en 2015 se creó la comisión esBIM, un grupo de trabajo encargado de promover el uso del BIM en el sector de la construcción en España y de desarrollar guías y normativas para facilitar su implementación.⁸⁰ Esta comisión también se encarga de impulsar y garantizar la coordinación de la Administración General del Estado y sus organismos públicos y entidades de derecho público vinculados o dependientes, en la implementación de la metodología en la contratación pública.

⁷⁹ Estándares en apoyo del BIM. Informes de normalización. AENOR. 2016.

⁸⁰ Roadmap BIM España: una historia sobre BIM. BIMChannel. David Barco. 2022.

El año 2018 marcó un hito con la publicación de la Ley de Contratos del Sector Público, que permite a las administraciones exigir BIM en ciertos contratos de obras, concesiones y servicios. Posteriormente, en 2019, se creó la Comisión Interministerial BIM (CIBIM) para coordinar su implementación en la Administración General del Estado. Según un nuevo informe⁸¹ de buildingSMART Spain, la evolución de la adopción de BIM ha sido constante, con un incremento sostenido en los últimos años.

En los últimos años las grandes empresas han liderado el proceso de implementación, mientras que las PYMEs han presentado una adopción más lenta debido a los costos de implementación y la falta de formación específica. En infraestructuras, su crecimiento es más acelerado que en edificación debido a exigencias normativas y licitaciones públicas.

A nivel regional, la implementación BIM es desigual. Según el Observatorio BIM de España, Cataluña lidera la adopción con un 35% de proyectos licitados con requisitos BIM, seguida por la Comunidad de Madrid (16%), la Comunidad Valenciana (12%) y el País Vasco (10%). Sin embargo, regiones como Murcia, La Rioja, Ceuta y Melilla presentan niveles bajos de implantación. Este análisis territorial es clave para entender el impacto de las estrategias autonómicas en la adopción de BIM.

2.2.1.1. Ámbito público

Desde 2019, la Administración pública española ha incorporado progresivamente BIM en licitaciones. Entre 2019 y 2024, se registraron 1.644 licitaciones con requisitos BIM, por un valor total de 8.341 millones de euros. Concretamente en edificación, el número de licitaciones de este periodo es de 1.936. Aparte de este aumento en licitaciones con BIM, se han generado diferentes iniciativas para impulsar el desarrollo de la digitalización en todos los sectores, incluyendo el de la construcción. Por ejemplo, en el Plan España Digital 2025⁸² desarrolla las bases para generar proyectos tractores que incorporen la metodología BIM. Por otro lado, en 2023, el Consejo de ministros aprobó el Plan de Incorporación de la Metodología BIM en la Contratación Pública de proyectos de infraestructuras, estableciendo que, a partir del 1 de abril de 2024, los contratos de obras y concesiones con un valor superior a 5,538 millones de euros deban emplear BIM con un nivel inicial de implementación.

En cuanto a las valoraciones en pliegos, desde los datos del observatorio BIM de la Comisión Interministerial, destacan diferentes criterios técnicos como la evaluación de los perfiles BIM (73%), la adecuación de la metodología a los requisitos específicos del contrato (56%) y la estrategia de implantación BIM (54%). Además, se consideran aspectos como la incorporación de usos BIM adicionales (54%) y la estrategia de desarrollo de los usos BIM requeridos (53%). En términos de puntuación global, que combina criterios técnicos y económicos, los contratos de diseño presentan un promedio de 5,23, mientras que en contratos de obra asciende al 6,41%. La puntuación técnica específica es mayor en los contratos de obra, con un 21,36%, frente al 8,40% en contratos de diseño.

⁸¹ Informe de situación BIM en España 2024. buildingSmart Spain. 2025.

⁸² España Digital 2025: Estrategia para la digitalización de la economía española. Gobierno de España. (2020).



Figura 2.2.1. Valoraciones en pliegos de edificación (2017-2023)

También a partir del Informe acerca del Nivel de Adopción BIM, se destaca que el 73% de los promotores públicos exigen la entrega de modelos BIM en formatos abiertos como IFC. Además, el 39% de los promotores dispone de una guía propia para especificar requisitos BIM, mientras que un 29,3% lo hace a través de pliegos de licitación. Sin embargo, un 17,1% de las administraciones solicita BIM sin detallar ningún requisito específico. En este contexto, un estudio reciente (Pérez-García et al., 2024) detecta que la falta de madurez y preparación de las administraciones y empresas de España dificulta la correcta adopción y gestión BIM en la contratación pública y propone un marco adaptado al nivel de madurez de cada entidad, con requisitos BIM claros y el uso del estándar IDS para mejorar la calidad de los modelos.

En cuanto a la inversión en formación, se han destinado 2,6 millones de euros a subvenciones para capacitación en BIM en colegios profesionales y administraciones. A pesar de este esfuerzo, el 52% de los técnicos de la administración aún consideran que tienen un conocimiento básico o intermedio en BIM. Con respecto a la digitalización y el uso de metodologías como los gemelos digitales, estos están aumentando en ciudades como Madrid, donde ya se han integrado modelos BIM en plataformas urbanas.

Comunidades autónomas como Euskadi han avanzado significativamente en la adopción de BIM. Desde 2022, el 100% de los proyectos de vivienda pública se licitan con BIM. Asimismo, en Cataluña, la aplicación de BIM en proyectos con un presupuesto superior a 2 millones de euros es obligatoria desde 2023.

2.2.1.2. Ámbito privado

En el sector privado, la adopción de BIM ha avanzado de forma desigual. Según el Informe del Nivel de Adopción, el 32% de las organizaciones han implantado BIM, mientras que un 26% están en proceso de hacerlo. Sin embargo, un 34% aún no ha iniciado su implementación. El 60,5% de las empresas que han adoptado BIM lo han aplicado en más de 10 proyectos.⁸³

⁸³ Informe del Nivel de Adopción del BIM en España, en organizaciones y profesionales 2024. BuildingSmart Spain. 2025.

Los principales obstáculos para la adopción de BIM en el sector privado incluyen el alto coste de software y hardware (60%), la falta de tiempo para la transición (47%) y la escasa exigencia por parte de los clientes (42%). A pesar de ello, el 77,3% de las empresas con BIM cuentan con un responsable BIM en su organización. Además, el 46,2% de los gestores de activos utilizan modelos BIM en formato IFC para mejorar la gestión y mantenimiento de sus infraestructuras y edificios.

El impacto de BIM en la empleabilidad también es notable. El 70% de los profesionales encuestados que han conseguido empleo en los últimos 10 años consideran que la experiencia y/o formación en BIM han sido determinantes para su contratación. Este dato subraya el creciente papel de BIM no solo como una herramienta técnica, sino como un factor clave en la competitividad del sector AECO.

Las empresas que han implementado BIM reportan una reducción del 25% en los errores de diseño y construcción, así como un ahorro del 15% en costos operacionales. Sin embargo, las PYMEs siguen enfrentando desafíos como la falta de interoperabilidad entre software y la resistencia al cambio organizacional. Para abordar estos problemas, buildingSMART Spain ha desarrollado un Observatorio de Adopción BIM, proporcionando datos actualizados sobre la evolución de la metodología en el país.

A nivel internacional, España ha fortalecido su presencia en foros y redes sobre BIM, participando en iniciativas europeas para la estandarización y mejora de la metodología. Múltiples empresas han implementado BIM en diferentes proyectos, obteniendo mejoras en eficiencia y reducción de costes. La consultora McKinsey & Company ha señalado que la inversión en herramientas digitales en el sector de la construcción ha aumentado un 45% en los últimos cinco años, con BIM como uno de los principales impulsores.

Además, en los últimos años, se han desarrollado diversas iniciativas de colaboración entre entidades públicas y privadas para fomentar la digitalización en el sector. En particular, buildingSMART Spain ha liderado iniciativas como el Observatorio de Licitaciones BIM, proporcionando datos actualizados sobre el avance de la metodología en España y generando informes trimestrales para el estudio de las tendencias y porcentajes. La evolución positiva de BIM en el país apunta a una consolidación progresiva de su uso en todos los ámbitos del sector AECO.⁸⁴

2.2.1.3. Barreras y retos

A pesar de sus múltiples ventajas, la adopción BIM enfrenta diversas barreras y retos (Lee, G. & Borrmann, 2020) que dificultan su implementación a nivel global. Estos obstáculos pueden clasificarse en barreras externas, ajenas a la empresa, y barreras internas, que dependen de la organización y pueden ser gestionadas para facilitar la transición. Entre las barreras externas, la falta de una normativa específica y estandarizada constituye un desafío significativo. En muchos países, no existen regulaciones claras sobre el uso de BIM, lo que genera descoordinación entre clientes y proveedores. Asimismo, las limitaciones tecnológicas persisten en ciertos ámbitos, con problemas de interoperabilidad entre plataformas que dificultan la colaboración eficiente. Otra barrera importante es el desconocimiento por parte de

⁸⁴ Global openBIM Mandates. A playbook of IFC Mandates from around the world. buildingSmart International. 2025.

los clientes, quienes no siempre poseen un criterio uniforme sobre el grado de aplicación en los proyectos, lo que puede encarecer los procesos sin una adecuada valoración costo-beneficio. En cuanto a las barreras internas, el costo de implementación es un factor determinante. La transición a BIM requiere inversión en licencias de software, actualización de infraestructura tecnológica y formación del personal, lo que puede afectar la productividad a corto plazo. Además, la resistencia al cambio dentro de las organizaciones supone una dificultad adicional, ya que la transformación digital implica modificar procesos tradicionales, generando incertidumbre y reticencia en algunos sectores (Nguyen & Nguyen, 2021).

La falta de estandarización en flujos de trabajo y gestión de datos también representa un desafío operativo (Toyin & Mewomo, 2021). La ausencia de protocolos claros sobre nomenclatura de archivos y estructura de información puede generar ineficiencias y pérdidas de tiempo en la organización de datos relacionados con materiales, instalaciones, costos y mantenimiento, entre otros. Otro problema frecuente es la concepción errónea de BIM como un software específico en lugar de una metodología integral de gestión de la información del proyecto. Muchas empresas limitan su aplicación al uso de herramientas digitales sin incorporar normativas, documentación organizada y roles especializados.

Por otro lado, en diversas organizaciones, la adopción de BIM se realiza únicamente para cumplir con requisitos contractuales sin considerar sus beneficios estratégicos. Este enfoque restringido limita la optimización de procesos internos y su potencial para mejorar la eficiencia y rentabilidad de los proyectos. En otros casos, se subestima su utilidad bajo la creencia de que los procesos tradicionales son suficientes, dificultando la integración de nuevas prácticas digitales en el sector de la construcción y la ingeniería (al Omari et al., 2023).

En referencia al futuro del BIM, según una encuesta realizada por BuildingSmart International a diferentes especialistas y empresas del sector de la construcción,⁸⁵ se espera un aumento en la mejora de la calidad y la reducción de errores en los proyectos, un aspecto destacado por el 77% de los encuestados. En segundo lugar, el 66% de los participantes considera que el desarrollo y uso de gemelos digitales será uno de los beneficios clave, permitiendo una mejor gestión y simulación de activos en tiempo real. Asimismo, el 64% de los encuestados espera que la automatización de procesos constructivos y de diseño juegue un papel fundamental en la optimización de tiempos y recursos. Estos resultados reflejan una tendencia global hacia la integración de tecnologías avanzadas en el ecosistema BIM, consolidando un enfoque centrado en la eficiencia, la predictibilidad y el control integral del proyecto.

⁸⁵ Industry Survey: Understanding the role of BIM and CDEs Today and Expectations for the Future. buildingSMART International. Febrero 2022.

2.2.2. OpenBIM

La colaboración digital en el sector AECO sigue enfrentando importantes desafíos. A menudo, los equipos trabajan con sus propias plantillas y formatos, lo que genera una gran fragmentación en los procesos. Si bien los modelos 3D han mejorado la interoperabilidad, muchos continúan restringidos por formatos propietarios que limitan el control y acceso a los datos. Ante este panorama, OpenBIM surge como una solución clave, proporcionando un enfoque colaborativo y transparente que permite la interoperabilidad entre distintas herramientas, procesos y actores. Este concepto promueve un flujo de trabajo abierto e independiente del software utilizado, fomentando la participación de todos los involucrados en un proyecto sin bloqueos técnicos o comerciales.⁸⁶

OpenBIM permite que los miembros del proyecto colaboren sin importar la plataforma de software que utilicen, garantizando un flujo de información eficiente. Se establece así un lenguaje común para la gestión de datos en la industria, permitiendo proyectos con mayor transparencia y consistencia en la calidad de la información. Esta información generada en un proyecto puede ser utilizada en todas las fases del activo sin necesidad de reintroducir datos, reduciendo errores y optimizando recursos.⁸⁷ Además, tanto pequeños como grandes proveedores de software pueden participar en el ecosistema OpenBIM, fomentando la innovación y la mejora continua. Los usuarios pueden acceder a un abanico más amplio de soluciones digitales compatibles con BIM, adaptadas a sus necesidades específicas.

Los beneficios de OpenBIM se reflejan en diversos aspectos de la gestión de proyectos, como la mejora en la comunicación sin ambigüedades, la optimización de la toma de decisiones basada en datos consistentes, una mayor coherencia en los entregables, el acceso garantizado a la información a lo largo del tiempo y procesos más eficientes y automatizados. Además, este nuevo enfoque contribuye a la sostenibilidad y resiliencia de los activos construidos ya que, al eliminar la dependencia de formatos propietarios, permite que los datos permanezcan accesibles a largo plazo, facilitando su reutilización y actualización conforme evolucionan las tecnologías y los requisitos normativos.⁸⁸

BuildingSMART International (BSI) es la organización principal en la promoción y desarrollo de estándares digitales abiertos en la industria de la construcción. Su objetivo es garantizar la interoperabilidad, la accesibilidad y la confiabilidad del activo construido. Desde su creación en 1995 como la International Alliance for Interoperability (IAI), ha trabajado para desarrollar estándares que permitan la digitalización efectiva de los procesos constructivos. Actualmente, esta organización lidera iniciativas globales para establecer protocolos de intercambio de información, asegurando la sostenibilidad y la eficiencia en el uso de datos. Dentro de su marco de trabajo, desarrolla especificaciones que facilitan la comprensión y aplicación de los datos en entornos BIM.

⁸⁶ Guidance for Regulators on use of OpenBIM. buildingSmart International. 2024.

⁸⁷ Manual BIM de ETS. Euskal Trenbide Sarea. Anejo 12: IFC. Junio 2024.

⁸⁸ Native OpenBIM and the rise of open source in AEC. AEC Magazine. D. & Hooper Moults. 2022.

La comunidad buildingSMART opera con base en principios fundamentales que garantizan un ecosistema abierto y colaborativo. Desarrolla estándares y servicios OpenBIM accesibles a nivel global, promueve la interoperabilidad y el acceso seguro a los datos, crea una infraestructura digital que soporte variaciones locales y nacionales, y mantiene un enfoque basado en la transparencia, neutralidad e inclusión en la toma de decisiones. La organización trabaja constantemente en la evolución de herramientas y procesos digitales que faciliten la adopción de OpenBIM en todo el mundo, contribuyendo así a la modernización de la industria y a la mejora de la calidad en la gestión de activos construidos. A través de colaboraciones con gobiernos, empresas y universidades, buildingSMART también impulsa iniciativas de investigación y desarrollo para mejorar la automatización en el sector, fomentar la digitalización de la cadena de suministro y fortalecer la seguridad y gestión de los datos a largo plazo.

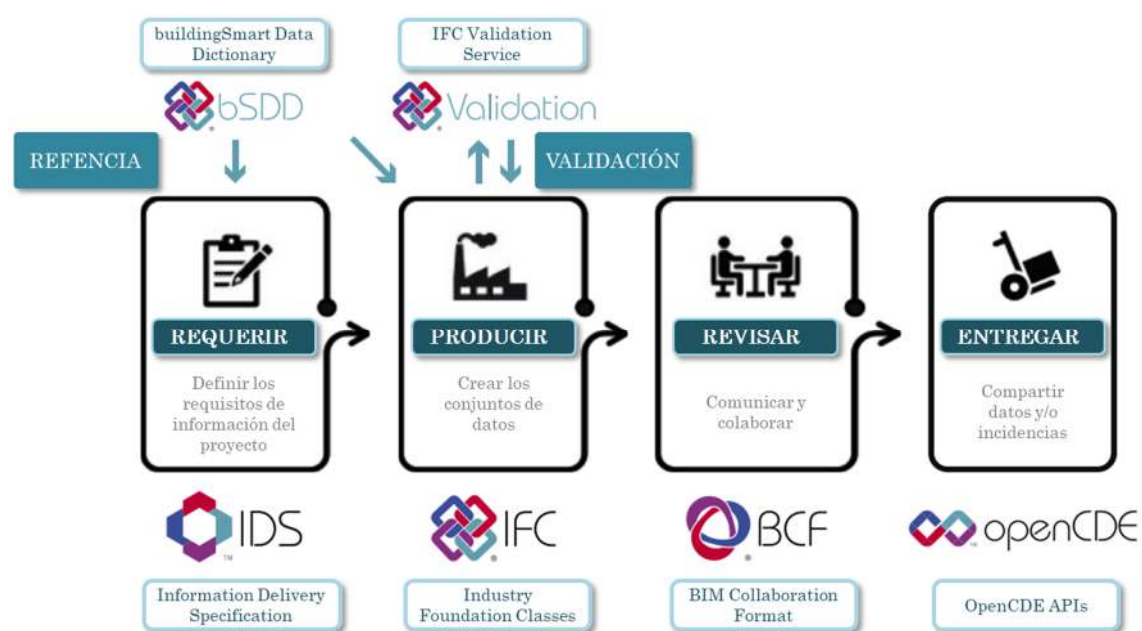


Figura 2.2.2. Estándares y servicios OpenBIM. Fuente* (buildingSmart)

En los próximos años, buildingSMART International tiene establecido en su hoja de ruta el seguir impulsando la digitalización y estandarización de la industria de la construcción mediante el desarrollo de nuevas normativas y certificaciones. Se prevé la integración de nuevos niveles de certificación para software, procesos y profesionales, garantizando la fiabilidad de las soluciones digitales utilizadas en el sector. Además, se pretende explorar la automatización de simulaciones y la optimización del uso de gemelos digitales para mejorar la toma de decisiones en tiempo real. Con el avance de tecnologías como la inteligencia artificial y el aprendizaje automático, la organización buscará trabajar en la creación de ecosistemas digitales más robustos, facilitando el acceso a datos verificados y optimizando los flujos de información en toda la cadena de valor.⁸⁹

⁸⁹ buildingSMART. Strategic Roadmap. buildingSMART International. 2023.

2.2.2.1. Estándares OpenBIM

Los estándares OpenBIM desarrollados por buildingSMART representan un conjunto de especificaciones abiertas, diseñadas para garantizar la interoperabilidad y la eficiencia en la gestión de datos dentro de la industria de la construcción. Estos estándares incluyen diferentes tipos de especificaciones que abordan aspectos variados de los procesos BIM, desde la estructura y el intercambio de datos hasta la definición de requisitos técnicos y la coordinación de flujos de trabajo. La naturaleza abierta de estos estándares permite que cualquier organización o desarrollador de software los implemente sin restricciones, promoviendo un ecosistema más coordinado. Al estar basados en principios de transparencia y colaboración, los estándares OpenBIM facilitan la adopción de prácticas digitales uniformes, permiten la automatización de procesos críticos y apoyan la transformación digital de la industria, al mismo tiempo que garantizan que los datos generados en cada fase del proyecto sean reutilizables y compatibles entre distintas plataformas.

Industry Foundation Classes

Industry Foundation Classes (IFC) es un estándar abierto definido en la norma ISO 16739 que fue creado para facilitar el intercambio de información en proyectos de edificación e infraestructura. Su objetivo es garantizar la interoperabilidad entre diferentes plataformas de software, permitiendo que los datos de un modelo digital sean accesibles y reutilizables a lo largo de todo el ciclo de vida. Este estándar se desarrolla con mayor profundidad en apartados posteriores de la presente investigación.

Information Delivery Manual

Information Delivery Manual (IDM) es un estándar recogido en la norma ISO 29481 y su propósito es establecer una metodología que relacione los procesos involucrados en la construcción de edificios e infraestructuras con la información requerida en cada etapa del ciclo de vida del activo. A través del IDM, se describen y estructuran los flujos de información de manera sistemática, garantizando que los datos se intercambien de forma precisa, confiable y eficiente entre diferentes actores y plataformas tecnológicas.

La metodología IDM tiene como objetivo principal facilitar el intercambio entre aplicaciones de software, desde la planificación y el diseño hasta la construcción, operación y mantenimiento. Para ello, el IDM no solo especifica qué información debe ser intercambiada, sino también quién es responsable de generarla y con qué infraestructura tecnológica se llevará a cabo el intercambio. De este modo, se proporciona un marco claro para la coordinación de procesos y la gestión eficiente de datos en entornos BIM. Además, este estándar estructura el flujo de información mediante la identificación de actividades dentro de un proceso, estableciendo el orden y la relación entre ellas. Asimismo, define las entradas y salidas de información en cada actividad, los responsables del envío y recepción de los datos y las herramientas de software y hardware que deben emplearse.⁹⁰

⁹⁰ Information Delivery Manual. Guide to Components and Development Methods. buildingSmart International. V1.2. 2010.

En la práctica, el IDM se utiliza en combinación con otras metodologías como los Model View Definitions (MVD), que aseguran que los datos intercambiados sean interpretables por las distintas soluciones de software. Un ejemplo de aplicación del IDM es la definición de procesos en el análisis energético de edificaciones, donde se especifica qué información debe ser intercambiada entre disciplinas como arquitectura e ingeniería, garantizando la coherencia y precisión de los datos.

El uso del estándar beneficia tanto a proveedores de software como a usuarios BIM. Para los desarrolladores de software, facilita la creación de herramientas más alineadas con las necesidades de la industria, garantizando la calidad del intercambio de datos y permitiendo la reutilización de componentes. Para los usuarios, el IDM asegura que la información sea comprensible para todos los participantes del proyecto (Jiang et al., 2019). Este enfoque estructurado permite responder preguntas clave sobre los procesos de información en un proyecto, tales como qué actividades deben ejecutarse, quiénes deben desarrollarlo, en qué momento deben realizarse y cómo debe gestionarse la información. Aunque muchas veces pasa desapercibido para los usuarios BIM, el IDM es la base sobre la que se fundamenta la interoperabilidad y el intercambio de datos en entornos digitales, proporcionando modelos de procesos y plantillas para requisitos de información que pueden adaptarse a distintos tipos de proyectos.

Investigaciones recientes sobre los procesos IDM (Gigante-Barrera et al., 2017; Jeon & Lee, 2018; Liu et al., 2023) evolucionan hacia un modelo más integrado y adaptable, con un énfasis en mejorar la entrega de información para procesos de construcción y manufactura. Estos estudios se enfocan en diferentes desarrollos, como la especificación de nuevos contenidos para objetos LOD utilizando el estándar IDM, con el objetivo de mejorar la interoperabilidad entre fabricantes y otros actores del sector, propuestas acerca de la definición de un contenedor basado en IDM que integra diferentes niveles de necesidad de información, permitiendo una lectura más precisa y automatizada de las especificaciones u otras propuestas relacionadas con un desarrollo de proyectos más holístico y menos fragmentado.

BIM Collaboration Format

BIM Collaboration Format (BCF) es un formato estándar de colaboración desarrollado para mejorar la comunicación en proyectos que utilizan metodologías BIM, permitiendo el intercambio de incidencias o asuntos en los modelos a lo largo de todo el proyecto. Su principal función es facilitar la identificación, el seguimiento y la resolución de problemas detectados en los modelos, optimizando la comunicación entre los diferentes actores del proyecto sin necesidad de compartir el modelo completo. Es decir, en lugar de intercambiar los modelos en su totalidad, BCF permite la transmisión de información contextualizada relacionada con las incidencias, mediante archivos que contienen los detalles específicos sobre el problema detectado, como el punto de vista del modelo, los componentes involucrados, los comentarios del agente y las capturas de pantalla que representan las interferencias o conflictos o asuntos detectados.⁹¹

Este formato fue creado para superar las limitaciones de los métodos tradicionales de comunicación en proyectos colaborativos, los cuales a menudo se basan en correos

⁹¹ BCF, mejorando la colaboración. buildingSmart Spain. 2018.

electrónicos, informes en PDF o plataformas que no permiten un seguimiento efectivo de las incidencias. En estos sistemas tradicionales, la comunicación tiende a ser fragmentada y carece de trazabilidad en la resolución de los problemas, lo que puede ocasionar confusión y demoras en la toma de decisiones. Por el contrario, BCF actúa como una plataforma centralizada que permite a todos los agentes del proyecto acceder a un historial completo de incidencias, conocer su estado de resolución, y asignar responsabilidades de manera clara y directa. Además, el estándar permite organizar los asuntos por tipos, clasificándolos por colisiones, defectos, solicitudes, observaciones, incidencias, etc.

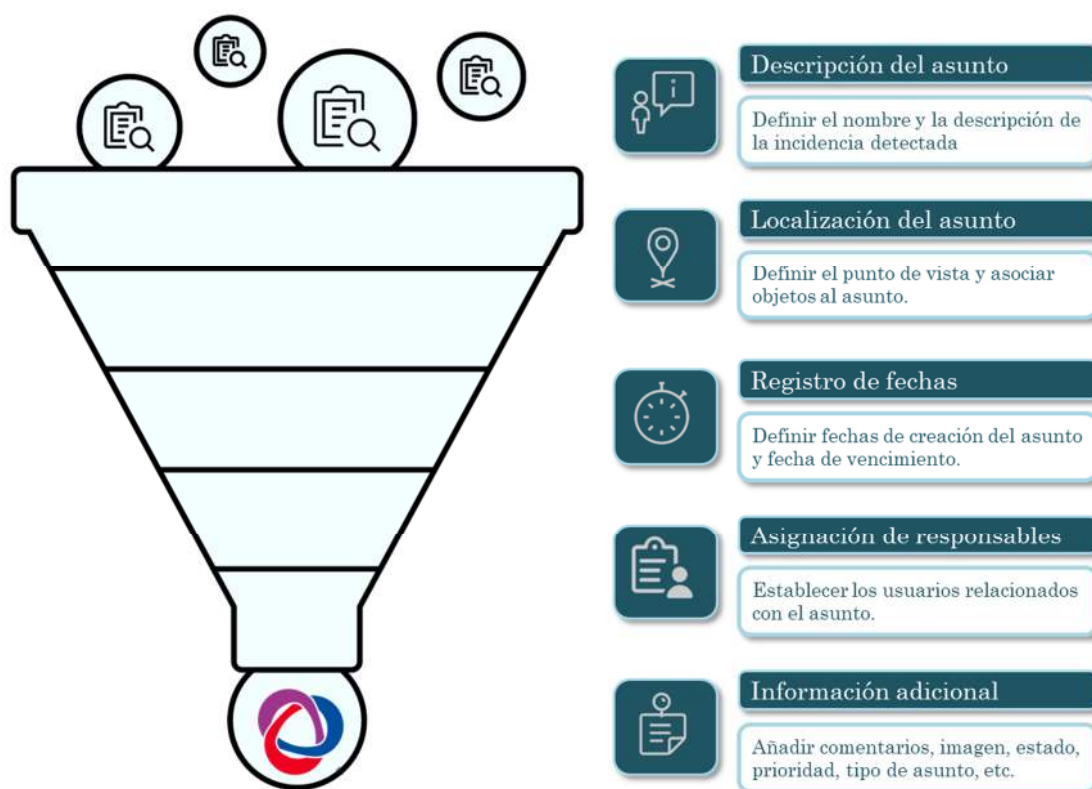


Figura 2.2.3. Información contenida en archivos BCF

El desarrollo del formato BCF comenzó en 2009, concebido inicialmente por miembros del grupo de apoyo de buildingSMART International, como Solibri, Tekla, y el Instituto de Informática Aplicada a la Edificación (IABI) de la Universidad de Munich, con el objetivo de facilitar la interoperabilidad entre distintas herramientas BIM y mejorar los flujos de trabajo basados en el estándar IFC. A medida que la adopción de BIM se ha expandido, el BCF ha ganado relevancia como herramienta clave para gestionar de manera efectiva las incidencias en proyectos de construcción, especialmente en aquellos que involucran equipos multidisciplinares. Con el objetivo de apoyar su implementación, desde buildingSmart se han generado diferentes recursos como mapas de procesos o guías para un mayor entendimiento.⁹²

El funcionamiento de BCF se basa en cinco componentes fundamentales que permiten organizar la información de las incidencias o asuntos de manera eficiente.

⁹² Comunicación con BCF. Gestión de tareas, Obras de rehabilitación, Control de calidad de los modelos y Revisión del diseño. buildingSmart Spain. 2022.

Se trata de cinco preguntas que estructuran la información necesaria para identificar y gestionar cada asunto. “Qué” se refiere a la descripción de la incidencia, identificada por un agente a través de un comentario. “Dónde” señala la ubicación exacta de la incidencia dentro del modelo, sin necesidad de generar capturas de pantalla externas, ya que la incidencia se geolocaliza directamente en el modelo y es visible para todos los involucrados. “Cuándo” establece la fecha de creación o modificación de la incidencia, permitiendo llevar un control temporal. “Quién” identifica al autor del comentario y al responsable de gestionar la incidencia. Finalmente, “Por qué” explica la razón de la incidencia, proporcionando un contexto claro sobre el problema detectado.

En cuanto a la implementación de este estándar, se puede gestionar mediante dos metodologías: basada en archivos o basada en la nube. En el primer método, los archivos se intercambian entre los agentes del proyecto, permitiendo una comunicación asincrónica y fácil de gestionar. Por otro lado, el segundo método permite la gestión centralizada de los archivos a través de una plataforma online, facilitando la colaboración en tiempo real y el acceso desde cualquier lugar, lo que mejora la coordinación y la toma de decisiones. En ambos casos, el uso de BCF permite mantener un registro histórico de las incidencias, lo que facilita el seguimiento y la resolución de los problemas a lo largo del ciclo de vida del proyecto.

Las investigaciones más recientes sobre el BCF reflejan un creciente interés por mejorar la interoperabilidad y la gestión de la comunicación en proyectos basados en OpenBIM (Treldal et al., 2016) explorando su uso como un mediador clave para la gestión de tareas en el diseño de edificios, destacando su potencial para integrar y coordinar flujos de trabajo. Otro estudio reciente (Lee, J. et al., 2016) presenta un método interoperable para compartir información de comunicación a través del estándar, subrayando su importancia para facilitar la colaboración entre diferentes plataformas y actores. Más recientemente, otra investigación (Lourenzi & Geiger, 2022) ha desarrollado BCFViewer, una herramienta que permite visualizar y gestionar estos archivos de colaboración, lo que facilita su adopción y uso práctico en entornos reales.

Information Delivery Specification

Information Delivery Specification (IDS) es un estándar desarrollado para definir los requisitos informativos de manera comprensible tanto para los seres humanos como para los sistemas informáticos. Su objetivo principal es especificar los datos que deben incluirse en un conjunto de datos BIM y verificar su entrega, lo que contribuye a la interoperabilidad y precisión en la gestión de la información. Además, permite la gestión de los datos de forma estandarizada y automatizada, lo que mejora la eficiencia y confiabilidad de los flujos de trabajo BIM, al mismo tiempo que fomenta la colaboración y la transparencia.⁹³

El 2024, buildingSMART International anunció que la versión 1.0 del IDS había alcanzado el estatus de estándar definitivo. Este hito representa un avance significativo en la definición e implementación de flujos de trabajo openBIM y responde a la creciente necesidad de intercambios de datos más precisos y confiables.

⁹³ What is Information Delivery Specification (IDS). buildingSmart International. 2023.

El funcionamiento de IDS se basa en un proceso de selección y validación de datos en dos etapas. En la primera etapa, denominada aplicabilidad, se seleccionan las entidades relevantes dentro de un modelo IFC utilizando diferentes tipos de información. En la segunda etapa, que corresponde a los requisitos, se verifica que las entidades seleccionadas cumplan con los criterios establecidos en el IDS. Entre los aspectos que el estándar puede validar se incluyen atributos de IFC, propiedades y cantidades, clasificaciones, la composición de objetos, así como materiales y tipos de IFC.⁹⁴

El uso de este estándar ofrece una serie de beneficios, como la automatización del proceso de validación, verificando automáticamente si los modelos BIM cumplen con los requisitos establecidos (Berlo et al., 2019). Al ser un estándar openBIM, facilita la interoperabilidad entre diferentes plataformas y software, lo que optimiza el intercambio de datos entre las partes involucradas en el proyecto. Además, al definir con claridad los requisitos informativos, contribuye a la reducción de errores, minimizando las inconsistencias y fallos en la entrega de información. También permite optimizar el flujo de trabajo, ya que modeladores, coordinadores BIM y propietarios de proyectos pueden trabajar bajo un mismo marco de referencia. Asimismo, facilita el cumplimiento de normativas internacionales y la estructuración estandarizada de datos, lo que simplifica la integración con otros sistemas y regulaciones.

A pesar de sus múltiples beneficios, la implementación de IDS enfrenta algunos desafíos (CEROVSEK & Omar, 2024), como la necesidad de contar con conocimientos previos sobre la estructura de IFC y la correcta definición de las reglas de validación. Además, algunos aspectos del proceso, como la verificación geométrica y la detección de interferencias, no están completamente cubiertos por IDS, lo que requiere el uso de herramientas complementarias. No obstante, su integración progresiva dentro del ecosistema openBIM se espera que mejore la estandarización y la automatización en la validación de modelos, optimizando la calidad de la información en toda la cadena de valor del sector de la construcción.

OpenCDE APIs

OpenCDE es una iniciativa que mejora la interoperabilidad a través de un conjunto de estándares basados en servicios web, con el objetivo de superar las limitaciones de los tradicionales CDE, los cuales requieren de la intervención manual para la gestión de archivos y la actualización de documentos. De esta manera, OpenCDE ofrece un enfoque más automatizado e integrado que facilita el intercambio de datos y la colaboración entre diferentes aplicaciones BIM y plataformas CDE.⁹⁵

Este sistema se compone de varias APIs especializadas, entre las cuales destaca la Foundation API, que establece las convenciones y servicios comunes para la interoperabilidad entre diferentes aplicaciones, y la Document API, que permite la carga y descarga de archivos de manera directa y eficiente desde las aplicaciones locales al CDE en la nube. Estas APIs permiten que los usuarios accedan y sincronicen documentos como modelos, planos u otros archivos asociados a un proyecto de forma automatizada, eliminando la necesidad de transferir

⁹⁴ User Manual. Documentation of the IDS. buildingSmart International. 2024.

⁹⁵ OpenCDE Foundation API. buildingSmart International. 2023.

manualmente cada archivo o versión. Otra característica importante es su capacidad para transferir solo las actualizaciones o cambios realizados en los documentos, lo que mejora la eficiencia al reducir la cantidad de datos transferidos y los tiempos de sincronización. Esta automatización también reduce el riesgo de errores humanos durante el proceso de carga y descarga de archivos.

Además, OpenCDE integra el estándar BCF, lo que permite que los equipos de trabajo gestionen y resuelvan problemas de forma más eficiente, promoviendo una comunicación más fluida y ágil entre los diferentes agentes del proyecto. Al eliminar la necesidad de aprender métodos específicos para cada plataforma CDE, OpenCDE facilita la interacción entre distintos proveedores y herramientas, impulsando una mayor colaboración y estandarización en los flujos de trabajo. De esta manera, se establece un ecosistema más interconectado, ágil y con menos riesgo de errores, lo cual es fundamental para la transformación digital del sector de la construcción.

Construction Operation Building information exchange

Construction Operations Building Information Exchange (COBie) es un estándar abierto que permite la transferencia estructurada de datos entre las fases de diseño, construcción y operación de un proyecto, facilitando la continuidad y consistencia de la información. Al igual que el formato IFC, COBie opera como un mecanismo para extraer y organizar datos gráficos y no gráficos de los modelos de información generados durante el desarrollo del proyecto. Sin embargo, su enfoque específico está dirigido a garantizar que la información relevante para la operación y el mantenimiento (O&M) de un activo esté disponible en un formato accesible y comprensible para los equipos de Facility Management. De esta manera, contribuye a cerrar la tradicional brecha de información que suele existir entre las fases del proyecto, consiguiendo reducir tiempos y costos asociados a la recopilación de datos, al mismo tiempo que mejora la toma de decisiones basada en información confiable y actualizada.

Este estándar (Shin et al., 2022) permite que los datos se almacenen y transfieran en hojas de cálculo, normalmente en formato Excel o CSV, donde cada hoja representa un conjunto de información estructurada relacionada con aspectos clave del activo, tales como espacios, sistemas, componentes, equipos y materiales. La estructura de COBie no solo facilita la captura de información de manera ordenada y consistente, sino que también habilita un flujo bidireccional de datos. Esto permite que los Facility Managers incorporen o actualicen información directamente en las tablas y que estos datos puedan ser posteriormente integrados en los modelos BIM, asegurando que el modelo digital refleje con precisión el estado real del activo durante su fase operativa. Además, al formalizar la entrega de información al finalizar la obra, los equipos de gestión reciben un paquete de datos completo, validado y alineado con las necesidades específicas de operación y mantenimiento, lo que facilita la puesta en marcha eficiente del activo.

Uno de los principales valores de COBie radica en su contribución a la gestión eficiente de activos,⁹⁶ ya que proporciona un marco normalizado para documentar y organizar la información necesaria para O&M. A través de su implementación,

⁹⁶ COBie Guide: Commentary on NBIMS-US COBie. National Institute of BUILDING SCIENCES. V05. 2013.

COBie permite eliminar la dependencia de documentos en papel tradicionalmente utilizados en contratos de construcción, reemplazándolos con flujos de trabajo digitales que optimizan el acceso y la actualización de datos. Su aplicación dentro del ciclo de vida del activo comienza en la fase de diseño, donde los arquitectos definen los espacios y las agrupaciones espaciales necesarias según los requerimientos del propietario. Esta información es incorporada en COBie como un subconjunto de datos del diseño esquemático, incluyendo zonificación y fichas de espacios, asegurando que la planificación se alinee con las necesidades operativas del futuro edificio.

Durante la fase de construcción, COBie se nutre de la documentación de diseño y especificaciones de equipos para consolidar información sobre productos y sistemas que serán administrados por el propietario. A medida que la construcción avanza, los contratistas complementan la información con detalles de instalación, pruebas, certificaciones y garantías, enriqueciendo los datos con información precisa sobre cada elemento del activo. De este modo, el estándar permite que la transición entre la fase de construcción y la operación sea más eficiente, facilitando la carga de datos en sistemas de gestión de mantenimiento asistido por ordenador (GMAO) o plataformas de gestión de activos.⁹⁷

Además, COBie se alinea con los principios de Lean Construction al reducir el desperdicio de tiempo y recursos asociados con la recopilación manual de información en etapas avanzadas del proyecto. En lugar de generar manuales de operación y mantenimiento de manera desorganizada, este estándar digitaliza y estructura la información de forma progresiva, asegurando su accesibilidad en el momento oportuno. Al implementar COBie, los propietarios pueden optimizar la gestión de activos al disponer de datos detallados sobre mantenimiento preventivo, históricos de intervenciones, repuestos asociados y garantías, lo que resulta en una mejor planificación de los recursos y una mayor vida útil del activo.

2.2.2.2. Servicios OpenBIM

Los servicios openBIM proporcionan soluciones especializadas que mejoran la colaboración, facilitando la gestión de información, garantizando la validación de datos y optimizando el acceso a recursos compartidos. Estos servicios están diseñados para ser accesibles a una amplia gama de usuarios, promoviendo la transparencia y la eficiencia en los proyectos de construcción. Están disponibles de forma gratuita y son opensource, lo que permite su adaptación y personalización según las necesidades de cada proyecto o plataforma. Esta apertura asegura una interoperabilidad fluida entre diferentes herramientas y sistemas, eliminando barreras tecnológicas y permitiendo a los equipos de trabajo utilizar las mejores soluciones disponibles sin restricciones de licencias o software propietario. Además, al ser desarrollados de forma colaborativa por una comunidad global de expertos, estos servicios están en constante evolución, garantizando su alineación con las mejores prácticas y estándares internacionales.

⁹⁷ Asset Operations Handover -Building Systems Equipment Maintenance. Information Delivery Manual - Part 2: Technical Requirements. buildingSMART International. 2022.

buildingSMART Data Dictionary

La creación de un buildingSMART Data Dictionary (bSDD) o diccionario de datos es un paso importante en la gestión de la accesibilidad, calidad e integridad de los datos dentro del entorno BIM.⁹⁸ Este diccionario funciona como un repositorio centralizado de información que describe el significado, las relaciones con otros datos, el origen, el uso y el formato de los datos empleados en la construcción. Al diseñarlo, es esencial descomponer la información de manera eficaz, optimizando su estructura y terminología. Para ello, es habitual dividir términos complejos en conceptos más simples y fundamentales. Esta descomposición no solo facilita la comprensión de los términos, sino que también permite reutilizar las definiciones en otros contextos, reduciendo la necesidad de crear nuevas definiciones.

A la hora de la creación de los diccionarios, es importante generar diferentes tipos, dependiendo de cada propósito. De esta manera, se pueden generar distintos conjuntos enfocados en clasificaciones de objetos de construcción, listados de materiales, etc. Es importante también evitar duplicidad con otros diccionarios existentes, por lo que, si el contenido ya está disponible, el proceso debe consistir en realizar una entre un diccionario y otro. Sin embargo, en los casos que se necesite ampliar o extender una definición existente, se puede publicar el término como una especialización de uno ya definido.

La organización jerárquica de los datos también juega un papel crucial en el diseño de diccionarios. La estructura jerárquica refleja las relaciones y dependencias entre los términos, facilitando la comprensión de cómo interactúan entre sí. Esta estructura ayuda a los usuarios a comprender rápidamente las conexiones y las dependencias entre los datos. En el caso de los códigos y nombres utilizados en los diccionarios, estos deben ser únicos y fáciles de identificar, evitando el uso de nombres demasiado genéricos o que entren en conflicto con otros diccionarios. El código debe ser reconocible y lo suficientemente corto, sin espacios, ya que este código se utilizará para generar las URI⁹⁹ de los recursos. Además, es importante establecer convenciones claras de nomenclatura para garantizar la coherencia en el diccionario.

Al proporcionar contenido en el bSDD, es fundamental ofrecer suficiente información para que los usuarios comprendan el significado y la aplicación de cada término. Aunque tiene algunos atributos obligatorios y muchos opcionales, siempre se debe proporcionar al menos el código y nombre de cada clase y propiedad, así como una definición que explique el significado semántico del término. Para las clases, también se deben incluir detalles sobre la jerarquía y las entidades relacionadas con el modelo IFC (buildingSMART International, 2020). Para las propiedades, hay que especificar el tipo de dato y las unidades.

El bSDD permite cargar una muestra y añadir más contenido posteriormente, actualizando el contenido a una nueva versión cada vez que se realicen cambios, siendo posible también definir diccionarios en diferentes idiomas.

⁹⁸ Guidelines for creating a data dictionary. buildingSMART International. 2022.

⁹⁹ URI (Uniform Resource Identifier) hace referencia a una cadena de caracteres que identifica de manera única un recurso en la web, como un término o elemento dentro de un diccionario de datos, permitiendo su acceso directo y asegurando la interoperabilidad de los datos en el caso de plataformas BIM.

Use Case Definitions

Use Case Definitions (UCD)¹⁰⁰ son definiciones estandarizadas que describen procesos específicos de intercambio de información dentro de proyectos de construcción e infraestructura, con el objetivo de establecer flujos de trabajo estructurados que garanticen la interoperabilidad entre diferentes herramientas y disciplinas en el entorno BIM. Estas definiciones son fundamentales para la implementación de procesos de colaboración y coordinación en proyectos multidisciplinares, donde la correcta gestión de la información es esencial. Concretamente, especifican qué información debe intercambiarse, en qué formato y en qué etapa del ciclo de vida del activo, lo que facilita la integración de estándares abiertos como IFC, COBie o BCF.

Estos UCD son parte de la iniciativa buildingSMART Use Case Management Service, una plataforma global que documenta, valida y comparte casos de uso estandarizados. Esta iniciativa promueve la adopción de procesos estructurados en la industria de la construcción, para facilitar la digitalización del sector y mejorar la eficiencia en la gestión de proyectos mediante la implementación de estándares abiertos y mejores prácticas basadas en BIM. Así, se avanza hacia una mayor integración y optimización de los flujos de trabajo en la industria, con un enfoque en la mejora continua y la interoperabilidad entre plataformas y agentes.

Validation Service IFC

Validation Service IFC es una plataforma en línea gratuita diseñada para verificar la conformidad de los archivos IFC con los estándares abiertos y las especificaciones técnicas del esquema IFC. Su principal objetivo es garantizar que los modelos IFC generados por diversas plataformas de software cumplan con los requisitos establecidos por buildingSMART International, promoviendo la interoperabilidad y la calidad de los datos dentro del ecosistema BIM. A través de este servicio, se asegura que los flujos de trabajo sean eficientes y los procesos digitales en la industria de la construcción sean optimizados.¹⁰¹

El proceso de validación se realiza mediante la aplicación de reglas automatizadas que analizan tanto la estructura como el contenido de los archivos IFC. Estas verificaciones incluyen la sintaxis STEP, el cumplimiento con el esquema IFC actualizado, y las reglas normativas del estándar IFC. Además, el servicio también realiza verificaciones no normativas, como el cumplimiento de prácticas comunes y las definiciones del bSDD, garantizando la calidad y consistencia de los datos en todo el proceso de intercambio de información. De esta manera, el servicio permite identificar y corregir errores en los modelos antes de su utilización en fases posteriores del proyecto, minimizando el riesgo de problemas durante la construcción y operación del edificio.

Además de la verificación de la conformidad de los modelos IFC, el servicio también apoya la certificación de software. A medida que los archivos IFC se validan, se extraen métricas sobre su calidad, lo que permite generar informes sobre la confiabilidad y consistencia del software que produce estos archivos. Estos informes

¹⁰⁰ bSI Use Case Management Service (UCMS). buildingSmart International. 2021.

¹⁰¹ What is the IFC Validation Service? buildingSmart International. 2020.

se utilizan dentro del programa internacional de certificación de software de buildingSMART, lo que contribuye al mejoramiento continuo de las herramientas utilizadas en la industria de la construcción. Sin embargo, cabe destacar que este servicio tiene limitaciones, ya que no valida reglas específicas de proyectos, países o normativas organizacionales. También es importante señalar que la visualización geométrica de los archivos IFC no está dentro del alcance del servicio, ya que muchos errores no se detectan a través de la visualización o no están relacionados con la representación geométrica del modelo.

Este servicio puede ser utilizado por desarrolladores de software para probar y depurar funciones de exportación e importación de archivos IFC en sus herramientas y de esa manera asegurar la interoperabilidad entre las diversas aplicaciones que utilizan el formato IFC. Los arquitectos e ingenieros, por su parte, pueden verificar que los archivos IFC que reciben de distintas disciplinas de diseño cumplan con el estándar, permitiendo identificar errores que puedan indicar la falta de cumplimiento con el estándar IFC. Los consultores utilizan el servicio para realizar auditorías de calidad y asegurarse de que los archivos IFC cumplan con los estándares necesarios antes de ser utilizados en el proyecto, proporcionando recomendaciones y retroalimentación al equipo de diseño o al propietario del proyecto. Los clientes, finalmente, pueden utilizarlo para validar los archivos IFC que reciben de los actores del proyecto, garantizando que estos cumplan con las restricciones del estándar IFC y puedan ser utilizados en etapas posteriores del ciclo de vida del edificio.

El flujo de trabajo recomendado para la validación de modelos IFC comienza con el uso del Validation Service IFC, lo que asegura que los modelos tengan una estructura de datos válida antes de ser sometidos a otras comprobaciones más específicas del proyecto. Aunque este servicio no es suficiente por sí solo para verificar que los modelos IFC cumplan con todos los requisitos de un proyecto, ofrece una base sólida para garantizar que los datos sean consistentes y correctos, facilitando otros procesos de validación y verificación.

2.2.3. Industry Foundation Classes

El esquema Industry Foundation Classes (IFC) constituye un estándar internacional abierto que desempeña un papel protagonista dentro del ecosistema BIM, al proporcionar un puente de interoperabilidad entre diferentes plataformas y software de modelado de información de construcción.¹⁰² A diferencia de los formatos propietarios, IFC permite un intercambio de datos neutral y estructurado, facilitando la colaboración multidisciplinaria y asegurando que la información de un proyecto pueda ser utilizada en todo su ciclo de vida. Su finalidad es proporcionar un modelo de datos neutral e interoperable que permita el intercambio eficiente de información entre los distintos agentes involucrados en el proyecto de edificación o infraestructura. Este estándar, respaldado por la norma ISO 16739,¹⁰³ define una estructura de datos basada en objetos geométricos enriquecidos con información semántica, que facilita la coordinación entre las plataformas y aplicaciones BIM.

La correcta estructuración y comprensión de la información dentro del esquema IFC se basa en cuatro conceptos clave: la taxonomía, que organiza los elementos del modelo en categorías jerárquicas según la tipología de objeto; la ontología, que define las relaciones y propiedades entre los elementos, como la conexión de un muro con una puerta a través de un hueco; la semántica, que asegura que los términos sean interpretados de manera consistente en diferentes plataformas; y la sintaxis, que establece el formato en que se representan los datos, garantizando su legibilidad y procesamiento adecuado. Este estándar, fundamentado en la interoperabilidad, asegura que la información se pueda utilizar de manera uniforme en distintos entornos digitales sin depender de formatos propietarios, mejorando la comunicación entre disciplinas y reduciendo el riesgo de pérdida o distorsión de datos.



Figura 2.2.4. Sintaxis, Semántica, Taxonomía y Ontología del IFC

¹⁰² BuildingSmart International. BSI Standards. Industry Foundation Classes (IFC).

¹⁰³ En la actualidad, standard ISO 16739-1:2024. Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries. Part 1: Data schema. Edición 2.

El modelo IFC (Borrmann et al., 2018a) organiza los datos en clases que representan elementos constructivos y sus relaciones jerárquicas. Esta jerarquía de información, conocida como 'IFC Tree View', estructura los datos en niveles que van desde la identificación del proyecto hasta los componentes más específicos, facilitando la navegación y consulta dentro del modelo. A través de esta estructura, se pueden definir relaciones entre los elementos, estableciendo dependencias y asociaciones que permiten una representación más precisa y funcional de la información del proyecto. Estos objetos incluyen tanto componentes constructivos físicos, como información abstracta, incluyendo anotaciones 2D, especificaciones de materiales, conexiones MEP, análisis energéticos y planificación de obra, entre otros. Su esquema jerárquico permite estructurar la información de manera coherente y facilita su consulta y uso en diversas aplicaciones de diseño, cálculo, construcción y mantenimiento.¹⁰⁴

Dentro del entorno BIM, es importante diferenciar entre el Esquema IFC, el Archivo IFC y el Estándar IFC, ya que cada uno tiene una función específica y unos niveles de abstracción concretos. En el caso del Esquema, se trata de una estructura conceptual que establece las reglas, clases, atributos y relaciones para organizar la información en un modelo BIM, actuando como una gramática que define cómo deben representarse los datos y asegurando su interoperabilidad entre plataformas y aplicaciones. El Archivo, por otro lado, es la representación concreta de un proyecto de construcción basada en ese esquema, conteniendo instancias específicas de los objetos definidos, como geometría, propiedades físicas y relaciones entre elementos. Mientras que el esquema define la organización de los datos, el archivo .ifc es el contenedor que almacena esos datos y facilita su intercambio entre diversas herramientas. Por último, el Estándar es el marco normativo global que define y regula tanto el esquema como los archivos, proporcionando las directrices y especificaciones técnicas necesarias para garantizar el intercambio de datos en el ciclo de vida.

2.2.3.1. Evolución del IFC

El desarrollo del esquema se inició a principios de la década de 1990, cuando Autodesk promovió la creación de un consorcio con el objetivo de desarrollar un conjunto de clases estándar para mejorar la interoperabilidad entre diversas aplicaciones de software utilizadas en la industria de la construcción, el cual más tarde terminaría denominándose buildingSmart. El objetivo primordial era crear un formato que no solo abarcara la geometría de los modelos, sino que también integrara la gestión de la información del proyecto. Este enfoque integral se consolidó con la publicación en 1997 del primer conjunto de definiciones bajo el nombre de Industry Foundation Classes, un estándar neutral y abierto que permitía el intercambio de datos entre distintos sistemas. Aunque la creación de este estándar se vio influenciada por la necesidad de mejorar la interoperabilidad, el proceso de desarrollo fue acelerado por la percepción generalizada de que los esfuerzos de estandarización impulsados por la ISO no satisfacían las demandas urgentes de la industria.

A medida que la buildingSMART avanzaba en la mejora del estándar, las sucesivas versiones de IFC comenzaron a incluir características cada vez más complejas. El

¹⁰⁴ Native OpenBIM and the rise of open source in AEC. AEC Magazine. D. & Hooper Moulton. 2022.

estándar pasó a incorporar también propiedades detalladas de los elementos del modelo, datos sobre su uso y los requisitos de los distintos agentes involucrados en el proyecto. Esta evolución permitió que IFC se convirtiera en una herramienta crucial para la integración de diferentes disciplinas dentro de un proyecto, lo que facilitaba la colaboración y el intercambio de información entre todos los actores de la cadena de valor.

El verdadero hito en la consolidación de IFC como un estándar internacional se alcanzó en 2013, cuando fue adoptado como parte de la norma ISO 16739. Esta integración no solo garantizó que IFC fuera reconocido globalmente, sino que también impulsó su utilización en proyectos públicos, especialmente en los procesos de licitación y aprobación. La adopción del formato se expandió rápidamente, con su implementación en una amplia variedad de herramientas de software utilizadas para modelado, análisis y gestión de infraestructuras. Su éxito no solo se debe a su apertura, sino también a la facilidad con la que se integra en flujos de trabajo colaborativos, promoviendo la interoperabilidad a nivel mundial y, con ello, el concepto de Open BIM.

En la actualidad, el estándar sigue evolucionando y, aunque en un inicio se centró en la descripción de edificios, IFC ha comenzado a extenderse a otras infraestructuras civiles, asegurando su relevancia en un entorno en constante digitalización (Krijnen, 2021). Gracias a su naturaleza abierta y neutral, sigue siendo la referencia principal para la estructuración y transmisión de datos en BIM, unificando la gestión de la información y asegurando la longevidad y accesibilidad de los datos de los activos de construcción, ampliando sus relaciones y definiciones para englobarlas en futuras versiones.¹⁰⁵

2.2.3.2. Beneficios y limitaciones del IFC

Uno de los mayores beneficios del formato IFC es su capacidad para almacenar y compartir información detallada sobre los modelos de construcción, incluyendo no solo la geometría de los elementos, sino también sus datos, como especificaciones técnicas, cantidades y clasificaciones. Esta representatividad integral de los modelos, accesible a través de archivos .ifc, asegura que todos los participantes del proyecto dispongan de la misma información precisa y actualizada, lo cual mejora la coordinación entre los diferentes equipos de trabajo y reduce la probabilidad de errores gracias a la gestión de datos consistentes de calidad.

El formato IFC también se destaca por su capacidad para facilitar la visualización y verificación de los modelos en 3D, sin necesidad de que todos los colaboradores utilicen el mismo software. Los visualizadores gratuitos disponibles permiten que los profesionales, independientemente de su especialidad o ubicación geográfica, puedan acceder a los modelos y realizar tareas como la federación de archivos, permitiendo la integración de diferentes especialidades de un proyecto.

Además, el ciclo de vida de los activos en la construcción se ve fuertemente reforzado por la durabilidad del formato IFC (Yu et al., 2023). A diferencia de los formatos propietarios, que pueden volverse obsoletos con el tiempo debido a las actualizaciones de software, los archivos .ifc no dependen de versiones específicas de

¹⁰⁵ The Evolution of IFC: The path to IFC5. buildingSmart Spain. 2024.

programas ni de fabricantes, lo que garantiza su accesibilidad y usabilidad a largo plazo. Esta independencia de software asegura que los datos del modelo puedan mantenerse disponibles siempre, incluso en momentos posteriores a la finalización de la obra.

Asimismo, el formato IFC contribuye a la mejora de la transparencia y a la reducción de errores,¹⁰⁶ lo que permite la verificación del cumplimiento normativo y la optimización de los recursos, como el rendimiento energético y las mediciones. Su capacidad para integrar datos sobre el rendimiento ambiental facilita la toma de decisiones informadas en cuanto a la sostenibilidad y la eficiencia de los edificios y las infraestructuras. En este sentido, IFC no solo promueve una mayor colaboración entre los equipos, sino que también mejora la productividad y la calidad de los proyectos de construcción a nivel global, posicionándose como el estándar más usado dentro del ecosistema BIM.

A pesar de sus múltiples ventajas, el estándar IFC presenta ciertas limitaciones que deben ser consideradas al implementarlo en proyectos BIM (Lange et al., 2018; Noardo et al., 2021; Thomas Liebich et al., 2009). Uno de los problemas más frecuentes es la posible pérdida de información al exportar modelos desde su software nativo al formato IFC. Esta pérdida puede afectar a detalles específicos, como texturas, planos o tablas de planificación, lo que limita la capacidad de representar fielmente el modelo original. Además, una vez que un archivo se convierte a formato IFC, no es posible restaurarlo a su formato original sin sufrir pérdidas de datos, a menos que se realice una nueva exportación desde la aplicación de origen. Esta situación puede complicar la actualización y modificación del modelo, ya que requiere la intervención directa en el software original.

Otra limitación inherente al uso de IFC es la dificultad para editar directamente los archivos en este formato. Aunque es posible modificar su contenido, esto generalmente requiere herramientas avanzadas de programación y manipulación de datos, lo que puede resultar poco accesible para los profesionales que no cuentan con los conocimientos técnicos necesarios. Esta falta de facilidad para la edición directa también genera ciertos inconvenientes en términos de flexibilidad y agilidad en el proceso de modificación de modelos a medida que el proyecto avanza.

Adicionalmente, la adopción de versiones más avanzadas del estándar IFC depende en gran medida de su correcta implementación por parte de los desarrolladores de software. En muchos casos, los avances en el estándar no se reflejan inmediatamente en las herramientas disponibles en el mercado, lo que ralentiza su integración efectiva en el flujo de trabajo BIM. Esto puede generar discrepancias entre las diferentes plataformas de software y afectar la interoperabilidad entre ellas, especialmente si no se han implementado completamente las últimas versiones del esquema IFC.

A pesar de estas limitaciones, el formato IFC ha logrado consolidarse como el estándar internacional para la transmisión de datos dentro del sector de la construcción. Además, su certificación bajo la norma ISO refuerza su legitimidad y fiabilidad, posicionándolo como la principal referencia de intercambio de datos BIM.

¹⁰⁶ IFC: what is it and why is it needed? AEC Magazine. Emma Hooper. 2022.

2.2.3.3. Roles en el desarrollo del esquema IFC

El desarrollo del IFC requiere de la participación de distintos grupos de expertos, cada uno con funciones específicas dentro del proceso de definición, integración e implementación del esquema.¹⁰⁷ Por un lado, los grupos de dominio, conformados por expertos en distintas disciplinas de la construcción, definen los requisitos específicos de cada sector, asegurando que el esquema contemple todas las necesidades funcionales y operativas para una adecuada interoperabilidad. Estos grupos trabajan en la identificación de casos de uso, requisitos de datos y procesos necesarios para garantizar que el esquema refleje fielmente las interacciones del mundo real en proyectos de construcción y operaciones de edificación e infraestructuras.

Por otro lado, el desarrollo técnico del esquema IFC está a cargo de un grupo de expertos en modelado de datos, ontologías y estándares de interoperabilidad. Su tarea principal es traducir los requisitos definidos por los grupos de dominio en especificaciones formales y estructuradas. Para ello, emplean metodologías de modelado de datos y diagramación formal, tales como los diagramas IFC-PDEF para la representación de procesos y los diagramas EXPRESS-G para el intercambio de datos. Además, estos expertos garantizan la coherencia y compatibilidad del esquema con versiones anteriores, facilitando la integración con otros estándares y tecnologías de la industria.

Por último, los proveedores de software desempeñan un papel fundamental en la implementación del esquema dentro de aplicaciones comerciales y plataformas BIM, ya que su responsabilidad principal es desarrollar herramientas compatibles con el estándar, asegurando su correcta interpretación y procesamiento en entornos digitales. Estos actores colaboran estrechamente con los expertos técnicos para verificar la viabilidad de las especificaciones, realizar pruebas de interoperabilidad y proporcionar retroalimentación sobre la usabilidad y rendimiento del esquema en aplicaciones prácticas. Asimismo, participan en la validación de la documentación técnica y en la creación de guías de implementación para facilitar su adopción por parte de la comunidad de usuarios.

El proceso de desarrollo de una versión del esquema IFC sigue una serie de pasos esenciales para garantizar su correcta implementación y adopción. Inicialmente, se recopilan los procesos AECO que serán soportados en la nueva versión y se definen los requisitos de información necesarios para cada uno. El siguiente paso es la revisión de la información y la selección de los procesos más adecuados para la nueva versión. A continuación, se especifican las mejoras en el modelo de objetos IFC y se integran en la arquitectura del esquema. Luego, se realiza una síntesis del modelo y se llevan a cabo diversas verificaciones de calidad. Una vez completadas estas fases, se realiza una revisión final antes de la publicación oficial del modelo IFC documentado. Después del lanzamiento, se procede a una implementación piloto para evaluar su funcionalidad, seguida de pruebas de conformidad y certificación. Finalmente, los proveedores de software incorporan la nueva versión en sus productos y la distribuyen comercialmente. El proceso concluye con la recopilación de comentarios de los usuarios para futuras mejoras y actualizaciones del esquema.

¹⁰⁷ IFC Methodology Guide - Methodology for the Development of Industry Foundation Classes. Technical Report. International Alliance for Interoperability (IAI). 1997.

2.2.3.4. Formatos de archivos IFC

El esquema IFC ha dado lugar a la creación de diferentes formatos de archivo¹⁰⁸ para asegurar que los datos puedan ser manejados de manera eficiente y efectiva. Cada formato tiene características técnicas específicas que responden a las diversas necesidades del sector de la construcción, como la capacidad de representar grandes volúmenes de datos, la compatibilidad con múltiples aplicaciones y la preservación de la integridad de la información durante el intercambio entre diferentes plataformas. A continuación, se detallan las especificaciones y ventajas técnicas de los formatos más utilizados.

- El formato estándar .ifc se basa en el lenguaje STEP (Standard for the Exchange of Product model data), que es un estándar globalmente reconocido para el intercambio de datos de productos industriales, permitiendo una representación digital detallada tanto de las características geométricas como de las propiedades funcionales de los productos en todas las etapas del proyecto. Estos archivos siguen un esquema jerárquico estructurado en clases y objetos, incluyendo propiedades específicas de los elementos constructivos, como materiales, dimensiones y relaciones espaciales. Técnicamente, los .ifc son legibles en texto plano, lo que hace que el formato sea accesible a los humanos y fácil de verificar, pero también implica que la eficiencia en el almacenamiento de datos puede verse limitada en modelos muy complejos o de gran tamaño. Sin embargo, su estructura modular y la posibilidad de dividir los archivos en subcomponentes facilitan la gestión de modelos complejos. El uso de texto plano también permite que los archivos sean más fácilmente integrables con otros sistemas que no necesariamente tienen que ser de software CAD o BIM, lo que facilita la interoperabilidad entre distintas plataformas.
- El formato .ifcXML utiliza el lenguaje XML (Extensible Markup Language),¹⁰⁹ que se basa en el esquema STEP-XML, definido en la norma ISO 10303-28. XML es un estándar ampliamente utilizado en el ámbito de la informática y la web para representar datos de forma estructurada y legible por humanos. A través de la codificación de datos en XML, se facilita la interoperabilidad entre diferentes herramientas que utilizan esta tecnología, como los servicios web y las plataformas que emplean tecnologías de la web semántica. En términos técnicos, el formato .ifcXML permite representar datos en un formato jerárquico con etiquetas XML, lo que resulta especialmente útil para la integración con otras herramientas que requieren datos estructurados. Sin embargo, el principal inconveniente de este formato es su tamaño. Dado que el modelo IFC se describe completamente en XML, los archivos .ifcXML pueden llegar a ser considerablemente más grandes en comparación con los archivos .ifc tradicionales. Esto se debe a la sobrecarga asociada con la codificación XML, que agrega etiquetas de datos y metadatos adicionales que aumentan el tamaño del archivo. Aunque el formato es adecuado para la interoperabilidad en aplicaciones que utilizan XML, no es tan eficiente en cuanto al almacenamiento de grandes cantidades de datos complejos, lo que limita su uso para modelos completos de gran escala.

¹⁰⁸ IFC Formats. buildingSmart International. Actualización 2025.

¹⁰⁹ ifcXML Specification Methodology. Specification of the mapping methodology from EXPRESS to XML schema definition for IFC4. buildingSmart International. 2013.

- El formato .ifcZIP es una variante comprimida del archivo .ifc, utilizando el estándar de compresión ZIP. La principal ventaja técnica de este formato es la reducción significativa en el tamaño del archivo, lo que lo convierte en una opción eficaz para almacenar grandes modelos o intercambiar archivos de forma más eficiente. Sin embargo, la compresión ZIP no es un formato nativo de IFC, y por lo tanto, la mayoría de los visores y programas de visualización de modelos BIM no son capaces de leer archivos .ifcZIP directamente. En lugar de ello, se requiere una extracción manual o automática del archivo comprimido para poder acceder a su contenido. Esta necesidad de descomprimir el archivo puede aumentar la complejidad del flujo de trabajo, especialmente cuando se manejan grandes cantidades de archivos comprimidos en proyectos de gran escala.
- Los formatos basados en el lenguaje ifcOWL¹¹⁰ son versiones del esquema IFC diseñadas para aprovechar las ventajas de la web semántica y la interoperabilidad basada en ontologías. Los archivos con las extensiones .ttl (Turtle) y .rdf (Resource Description Framework) son utilizados para describir de manera explícita las relaciones semánticas entre los elementos del modelo BIM. La codificación en ifcOWL permite representar no solo las relaciones geométricas y funcionales, sino también las propiedades contextuales, como la eficiencia energética o las normativas de construcción. La ventaja técnica de los formatos .ttl y .rdf es su capacidad para enlazar datos estructurados de manera flexible y extensible, permitiendo que los datos BIM se integren con otros sistemas basados en tecnologías de la web semántica. Sin embargo, este tipo de codificación tiene una complejidad adicional, ya que requiere un mayor conocimiento de los principios de la web semántica y el manejo de ontologías. Además, los propios creadores del esquema desaconsejan su uso en la práctica, ya que presenta desafíos de implementación. En su lugar, recomiendan el uso de bSDD.

En términos de escalabilidad y soporte de software, el formato .ifc sigue siendo el más utilizado debido a su equilibrio entre tamaño de archivo y capacidad para representar una amplia gama de datos de manera eficiente. Su compatibilidad con más de 67 aplicaciones de software BIM,¹¹¹ todas ellas certificadas por buildingSmart, lo convierte en la opción preferida para proyectos a gran escala, donde la interoperabilidad es esencial. El formato .ifcXML, aunque adecuado para ciertas aplicaciones de integración web, es menos utilizado para proyectos completos debido a su mayor tamaño. Por su parte, el formato .ifcZIP ofrece ventajas en cuanto a almacenamiento, pero su adopción es más limitada debido a la falta de soporte generalizado entre los visores de modelos BIM. Por otro lado, los formatos relacionados con ifcOWL (Pauwels & Terkaj, 2016) son relativamente nuevos y resultan complicados de implementar debido a la dificultad de adaptar el esquema IFC a otras estructuras de datos, encontrando diferentes excepciones y particularidades durante el proceso. Además de los formatos anteriores, existen otros que actualmente se encuentran en desarrollo o bien que están en proceso de revisión, como son el JSON, SQLite o HDF, pero que por el momento no están aprobados de manera oficial.

¹¹⁰ What is IfcOWL? buildingSmart International.

¹¹¹ IFC Certification Participants. buildingSmart International.

<https://technical.buildingsmart.org/services/certification/ifc-certification-participants/>

2.2.3.5. Arquitectura de datos y capas

El esquema IFC tiene sus fundamentos en tecnologías de modelado de datos previamente establecidas, entre las que destacan STEP (Standard for the Exchange of Product Model Data) y EXPRESS. STEP es un estándar internacional desarrollado para la representación e intercambio de datos de productos en diversas industrias, incluidas la manufactura y la construcción. Su objetivo es permitir la interoperabilidad entre sistemas de software mediante una estructura de datos neutral e independiente de plataformas específicas. Dentro de STEP, se define EXPRESS como el lenguaje formal utilizado para describir los esquemas de datos de manera precisa y consistente. Este lenguaje ofrece una sintaxis estructurada basada en principios de orientación a objetos, permitiendo la definición de entidades, atributos y relaciones con un alto grado de formalidad.¹¹²

EXPRESS, regulado por la norma ISO 10303, es un lenguaje declarativo diseñado específicamente para la modelación de datos complejos. Su estructura permite la abstracción de objetos del mundo real en clases denominadas entidades, las cuales pueden tener atributos, relaciones y herencia. También soporta tipos de datos avanzados como listas, conjuntos y estructuras de selección, facilitando la representación de modelos ricos en información. Además, cuenta con un mecanismo de validación mediante reglas de integridad expresadas en bloques WHERE, garantizando la consistencia de los datos. Para complementar su notación textual, EXPRESS incluye un lenguaje gráfico denominado EXPRESS-G, que permite una visualización estructurada de los esquemas de datos. Aunque este lenguaje se emplea para definir esquemas, las instancias de los datos suelen representarse en formatos como archivos físicos STEP, XML o bases de datos, asegurando compatibilidad con diversos sistemas.

En cuanto a la arquitectura del esquema IFC,¹¹³ esta se organiza en una estructura jerárquica que permite la representación modular y coherente de los datos dentro del sector AECO. Su diseño responde a la necesidad de garantizar flexibilidad y evolución del modelo sin comprometer su consistencia interna. Para ello, este modelo se basa en principios de estructuración estrictos que determinan la relación entre las entidades en diferentes niveles, evitando dependencias circulares y asegurando la reutilización eficiente de componentes en distintos contextos.

La arquitectura se segmenta en cuatro capas conceptuales, cada una con un propósito específico dentro de la jerarquía del modelo. En el nivel más fundamental se encuentra la Capa de Recursos o Resource Layer, que proporciona los bloques de construcción esenciales sobre los que se estructuran las capas superiores. Aquí se definen conceptos básicos como medidas, unidades, geometría, costos y actores, elementos que no dependen de entidades superiores y pueden ser referenciados por todas las capas del modelo. Sin embargo, estos recursos no pueden ser utilizados de manera independiente, ya que carecen de identificadores globalmente únicos y requieren una instancia definida en niveles superiores.¹¹⁴

¹¹² The EXPRESS Definition Language for IFC Development. buildingSmart International.

¹¹³ Industry Foundation Classes. IFC Object Model Architecture Guide. International Alliance for Interoperability (IAI). Release 2.0. 1999.

¹¹⁴ IFC 4.3.2.20241204 (IFC4X3_ADD2). Data schema architecture of IFC. buildingSMART International. <https://ifc43-docs.standards.buildingsmart.org/>

Encima de la Capa de Recursos se encuentra la Capa Central, o Core Layer, que establece la estructura fundamental del IFC mediante la definición de conceptos universales aplicables a cualquier disciplina. Su subdivisión incluye el Kernel, que introduce objetos, atributos, relaciones y roles generales, y las Extensiones del Núcleo, que refinan estos conceptos con especializaciones para procesos y productos específicos. En esta capa, las entidades comienzan a incorporar identificadores únicos globalmente y pueden incluir metadatos adicionales, como información sobre propietarios y versiones.

Sobre la Capa Central se encuentra la Capa de Interoperabilidad o Interoperability Layer, que permite la conexión entre diferentes dominios mediante módulos de conceptos compartidos. Su función principal es proporcionar definiciones de entidades que son utilizadas por múltiples disciplinas, evitando redundancias y facilitando la compatibilidad entre modelos especializados. Esta capa introduce elementos constructivos compartidos, que agrupan componentes arquitectónicos comunes, y elementos espaciales comunes, que definen estructuras espaciales aplicables a diversas áreas del sector. La interoperabilidad en este caso se refuerza mediante adaptadores que permiten la integración de modelos externos no armonizados con IFC, garantizando la transferencia de datos.

Finalmente, en el nivel más alto de la jerarquía se encuentra la Capa de Dominio o Core Layer, donde se especializan los modelos en función de áreas específicas del sector, como arquitectura, estructuras o instalaciones. Esta capa define esquemas detallados con atributos específicos para cada disciplina, permitiendo representar con precisión los distintos aspectos de un proyecto. Algunos modelos de dominio pueden integrarse directamente en el núcleo, mientras que otros requieren adaptadores para garantizar su compatibilidad con el resto del ecosistema IFC. En las versiones más recientes de IFC, esta capa ha incorporado nuevas disciplinas como infraestructuras de transporte y gestión portuaria, ampliando el alcance del estándar.

El diseño de capas en la arquitectura IFC garantiza una separación clara entre niveles de abstracción, promoviendo una organización eficiente de los datos y permitiendo la evolución del modelo sin comprometer los procesos de intercambio. Cada capa cumple una función definida dentro del sistema, asegurando que los datos puedan ser representados, compartidos y utilizados de manera coherente en un entorno multidisciplinar.

2.2.3.6. Jerarquía de herencia y relaciones

En los modelos de datos orientados a objetos (Borrmann et al., 2018a), la jerarquía de herencia es un mecanismo esencial que permite la organización estructurada de clases mediante relaciones de generalización y especialización. Este concepto se basa en la reutilización de atributos y comportamientos definidos en clases superiores, permitiendo que las subclases hereden estas características sin necesidad de redefinirlas. La herencia facilita la modularidad y la extensibilidad de los modelos, ya que promueve la organización jerárquica y reduce la redundancia de datos. En un esquema de herencia, las clases más generales, usualmente denominadas superclases, contienen los atributos y comportamientos comunes a varias subclases, mientras que estas últimas pueden agregar especificaciones adicionales o modificar comportamientos preexistentes.

En el contexto de los modelos BIM, la herencia es vital para representar la diversidad de elementos en un entorno estructurado y comprensible ya que, a través de este mecanismo, es posible definir entidades genéricas como elementos constructivos y permitir que subclases más específicas, como muros o vigas, hereden sus propiedades básicas al tiempo que añaden atributos propios.

Dentro del esquema de IFC, la jerarquía de herencia sigue un enfoque semántico que permite la representación precisa de elementos del mundo real mediante la generalización y especialización de clases. Las entidades más generales dentro del modelo IFC están contenidas en la capa central, la cual define las propiedades básicas de objetos genéricos, que a su vez actúan como superclases de objetos específicos. Además, la herencia en IFC permite compartir atributos, relaciones y comportamientos. Por ejemplo, las entidades derivadas pueden heredar relaciones de conexión espacial y estructural.

Por su parte, las relaciones entre objetos dentro del modelo IFC son un componente esencial que permite estructurar y conectar los diferentes elementos de un proyecto de construcción. A diferencia de los modelos que consideran los objetos de forma aislada, IFC establece un sistema de interconexión que describe no solo la existencia de los elementos, sino también sus funciones, dependencias y asociaciones dentro del conjunto del modelo BIM. Estas relaciones posibilitan la representación de estructuras jerárquicas, lo que facilita una organización coherente y detallada de la información. Además, permiten definir conexiones entre elementos constructivos, asegurando que la interacción entre componentes refleje la realidad de la edificación. El modelo emplea el concepto de relaciones objetivadas, en el cual las interconexiones semánticamente relevantes se representan mediante objetos intermedios específicos, los cuales pertenecen a las subclases de tipo relación. Esta estructura permite gestionar eficazmente distintos tipos de relaciones según su propósito y función dentro del modelo.

2.2.3.7. Versiones del Archivo IFC

El desarrollo del esquema IFC a lo largo de los años ha permitido que los diferentes actores del sector puedan trabajar de manera cada vez más eficiente utilizando un formato común de intercambio de información. Desde su creación en 1996, el estándar ha ido adaptándose para cubrir nuevas necesidades y demandas de la industria.¹¹⁵ Las versiones del archivo IFC se agrupan en diferentes tipos, según los cambios realizados en cada una, encontrando versiones principales, versiones menores, adendas, y correcciones.

- La primera gran versión, IFC2x3, publicada en 2005, sigue siendo una de las más utilizadas en la actualidad, gracias a su integración con múltiples herramientas de software como Revit, Allplan, ArchiCAD, entre otras. Esta versión representó un gran avance respecto a las versiones anteriores al introducir un esquema más robusto y extendido que permitió un mejor intercambio de información. Sin embargo, aunque IFC2x3 se mantuvo como un estándar de facto en la industria durante muchos años, su uso ha quedado limitado principalmente al ámbito de la edificación. La principal limitación de esta versión radica en su falta de soporte para proyectos de

¹¹⁵ Autodesk IFC Manual. Understanding IFC. IFC Schema versions. 2021.

infraestructura, como puentes, carreteras, vías ferroviarias y otras estructuras lineales y horizontales. A pesar de su desactualización, IFC2x3 estableció la base para la interoperabilidad de datos en la construcción, y sigue siendo compatible con muchas herramientas de modelado y gestión de proyectos.

- La versión IFC4, lanzada en 2013, marcó un hito importante en la evolución del estándar, convirtiéndose en la versión oficial más reciente y recomendada para proyectos tanto de edificación como de infraestructura. Esta actualización trajo consigo una serie de mejoras significativas respecto a la predecesora, entre las cuales destaca la inclusión de nuevas clases, como la de alineaciones, que facilita la representación de proyectos lineales como carreteras, ferrocarriles y otras infraestructuras horizontales. Con esta incorporación, IFC4 amplió considerablemente su ámbito de aplicación, haciendo que fuera mucho más útil para el modelado y la gestión de proyectos de infraestructura. Además, IFC4 introdujo mejoras sustanciales en la eficiencia del esquema, lo que permitió una reducción significativa del tamaño de los archivos, y una mayor consistencia en el uso del formato. Otro de los aspectos clave de IFC4 fue la mejora en la representación geométrica, permitiendo una mayor precisión en elementos como extrusiones, barridos arbitrarios y superficies no planas. También se mejoró la capacidad de integración con los sistemas de información geográfica GIS, lo que facilitó la gestión de datos espaciales y su aplicación en proyectos de gran escala que requieren un análisis geoespacial. Además, se incorporaron mejoras para la representación de modelos estructurales y análisis, y se amplió la definición de elementos de servicios de construcción, lo que hizo de IFC4 una versión más versátil y preparada para abordar una mayor variedad de proyectos.
- La versión IFC4.3 ADD2, lanzada en 2024, representó un avance significativo, ya que fue aprobada como norma ISO en enero de 2024, y publicada oficialmente como ISO 16739 en abril de 2024.¹¹⁶ Esta nueva versión incluyó una serie de mejoras adicionales enfocadas principalmente en la infraestructura. Por ejemplo, se añadieron nuevas clases para representar elementos específicos de infraestructuras como puentes, vías ferroviarias, carreteras y proyectos marítimos, lo que permitió que el esquema abarcara de manera más eficiente las necesidades de proyectos de infraestructura horizontal y subterránea.¹¹⁷ Aunque IFC4.3 ADD2 se consolidó como un estándar ISO, la implementación de esta versión sigue siendo variable, ya que aún no se han establecido certificaciones de software dedicados para esta versión.¹¹⁸ Esto ha generado que su adopción dependa en gran medida del software utilizado, y en algunos casos, los usuarios deben verificar la compatibilidad de esta versión con sus plataformas antes de implementarla en proyectos activos.

¹¹⁶ IFC 4.3 Formally Approved and Published as an ISO Standard. 2024. buildingSMART International. 2024.

¹¹⁷ The status of IFC 4.3 and the benefit of further extensions as IFC 4.4. BuildingSmart International. Richard Kelly. 2021.

¹¹⁸ IFC Certification Participants. buildingSmart International. Actualización 2025.
<https://technical.buildingsmart.org/services/certification/ifc-certification-participants/>

- Por último, la versión IFC5,¹¹⁹ que está en desarrollo, se espera que continúe con esta tendencia de expansión hacia infraestructuras más complejas. Aunque todavía no se ha lanzado oficialmente, IFC5 incluirá nuevas entidades de infraestructuras subterráneas. Se prevé que esta versión también optimice la precisión en la representación geométrica, favoreciendo aún más el diseño y la construcción digital en sectores como el transporte, la energía y las infraestructuras urbanas.

En definitiva, cada una de estas versiones del esquema refleja los esfuerzos continuos por parte de la comunidad buildingSMART y sus colaboradores para mantener el estándar actualizado y adaptado a las necesidades cambiantes del sector de la construcción. Con cada nueva versión, el esquema mejora su capacidad para representar una gama más amplia de elementos constructivos, lo que asegura su relevancia y aplicabilidad en el futuro de la digitalización.

2.2.3.8. Model View Definition

Model View Definitions (MVD) son un componente esencial dentro del esquema de IFC, ya que optimizan el uso de los modelos en contextos específicos. Dado que el esquema IFC abarca una gran cantidad de datos, no todos son necesarios para todas las aplicaciones. Por ello, las MVD permiten seleccionar subconjuntos de información mínima que son relevantes para un uso determinado, facilitando la interoperabilidad entre diferentes plataformas y software. Estos subconjuntos definen qué datos gráficos y alfanuméricos deben incluirse en el intercambio de información, asegurando que solo la información relevante sea transferida, lo cual no solo mejora la eficiencia sino también protege la propiedad intelectual al limitar la exposición del modelo completo a los agentes que no lo han trabajado.

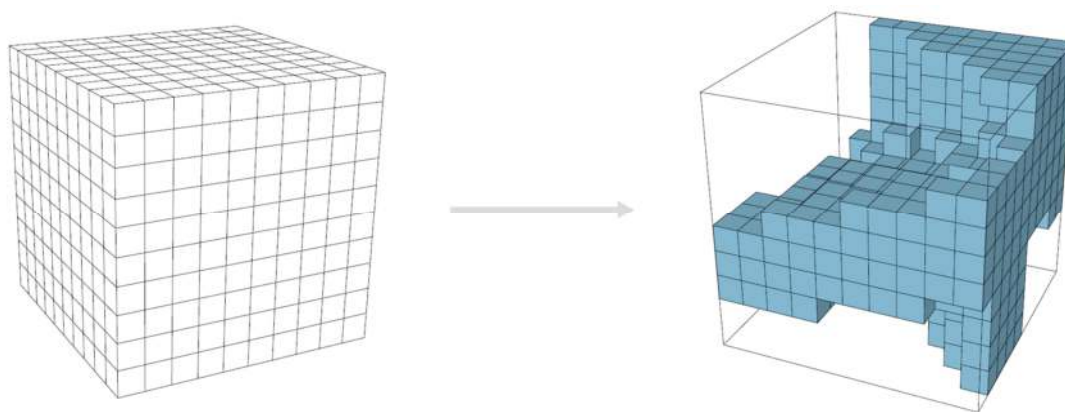


Figura 2.2.5. Esquema IFC completo vs. definición de subconjunto MVD¹²⁰

Cada MVD tiene un formato XML, conocido como mvdXML, que es neutral y se integra fácilmente con las herramientas existentes. Este formato permite segmentar los datos del modelo según las necesidades del proceso, y las MVD ofrecen una forma estandarizada de exportar y compartir las propiedades de los elementos del modelo siguiendo patrones que optimizan el flujo de trabajo. Por ejemplo, la IFC2x3 Coordination View V2.0 es una de las MVD más utilizadas, ya que ofrece una representación interoperable entre programas de modelado, gestión y visualización,

¹¹⁹ Interoperability has no version number. Léon van Berlo. buildingSmart International.

¹²⁰ Manual de IFC para Revit 2.0. Mark Baldwin. Autodesk. 2021.

favoreciendo el intercambio de datos entre aplicaciones de diferentes disciplinas. Sin embargo, no todos los programas permiten la elección de estas MVDs, y algunas versiones como las de IFC4 aún no tienen una implementación oficial.

En cuanto a las aplicaciones de las MVD, su propósito varía según el caso de uso. Para proyectos arquitectónicos, la IFC4 Reference View¹²¹ y la IFC2x3 Coordination View 2.0 son comúnmente utilizadas para representar la geometría y las relaciones espaciales de los elementos del modelo. La primera, particularmente, es adecuada para flujos de trabajo basados en modelos de referencia, donde se transfiere información sin permitir la modificación de la geometría importada. Por otro lado, IFC4 Design Transfer View¹²² introduce una mejora en la transferencia de datos, permitiendo una mayor fidelidad al transferir información entre diferentes herramientas sin perder detalles esenciales del diseño. Sin embargo, todavía no está completamente desarrollada, por lo que la comunidad sigue recomendando el uso de la IFC2x3 Coordination View 2.0 hasta que se finalicen las versiones correspondientes para IFC4.

La implementación de las MVD también facilita el trabajo en sectores específicos. Por ejemplo, en el análisis energético, un modelo arquitectónico puede ser exportado al ingeniero energético con detalles térmicos de materiales y características de confort térmico, lo que permite realizar una simulación precisa del rendimiento energético del edificio. Igualmente, en la estimación de cantidades, los contratistas pueden utilizar el modelo para calcular cantidades y asignar costos basándose en la ubicación precisa de los elementos constructivos.

La disponibilidad de las MVD en la plataforma buildingSMART International también permite que los usuarios consulten las definiciones de vistas de modelo más actualizadas y adopten las mejores prácticas para sus proyectos.¹²³ A pesar de las variaciones en la implementación de estas MVDs entre los programas de software, el uso de las MVD adecuadas asegura que los datos se transfieran de manera eficiente, estandarizada y con el menor riesgo de incompatibilidades o pérdida de información crítica durante el intercambio de datos entre aplicaciones.

En la actualidad las MVD se siguen ampliando para abordar nuevos desafíos, como los proyectos de construcción modular de varios pisos (Ramaji, Issa, J. & Memari, 2017), con nuevas extensiones para facilitar la integración entre módulos prefabricados, optimizando el intercambio de datos, nuevas metodologías para clasificar y determinar los requisitos de intercambio de datos entre los modelos OpenBIM y las simulaciones energéticas (Karlapudi, 2020), lo que mejora la integración de los modelos arquitectónicos y los sistemas de análisis energético o el uso de lenguajes genéricos (Deng et al., 2020) para generar subconjuntos de datos específicos mediante conjuntos de selección, facilitando exportaciones controladas de distintas aplicaciones. Estas iniciativas son ejemplos que buscan complementar las MVD existentes y mejorar la eficiencia de los procesos de intercambio de información, permitiendo un manejo más preciso y adaptado.

¹²¹ MVD. Purpose of the Reference View. buildingSMART International. 2019.

¹²² Autodesk IFC Manual. Understanding IFC. Model View Definitions (MVD).

¹²³ MVD Database. Model View Definition. buildingSmart International.

<https://technical.buildingsmart.org/standards/ifc/mvd/mvd-database/>

2.2.3.9. Generación de archivos IFC

Con el objetivo de conseguir una correcta generación del archivo IFC, es fundamental seguir un proceso sistemático que asegure la coherencia geométrica, semántica y de datos (Huahui & Deng, 2018). En primer lugar, es importante señalar que este tipo de archivos no forman parte de la línea de trabajo del modelo nativo, sino que es una captura de la geometría y datos, de tal forma que cualquier persona pueda acceder, consultar y revisar dicha información. Sin embargo, la coordinación de la información procedente de distintos softwares pone de manifiesto una cierta inoperancia de los procesos BIM, debido a la problemática relacionada con la adaptación de los distintos softwares a la hora de exportar e importar las distintas versiones del esquema IFC. Así pues, aunque el IFC es un recurso valioso en el ámbito de la coordinación BIM, es más útil como un registro o captura final de los datos y la geometría del modelo para un tiempo, fase o hito de entrega concreto.

El primer paso en la generación de un modelo IFC es la organización estructural del proyecto. Una jerarquía clara y bien definida es crucial para garantizar que todos los elementos sean correctamente interpretados durante la exportación y en plataformas de software diferentes. Esta estructuración incluye la correcta disposición de los niveles y elementos de referencia, así como la agrupación adecuada de los objetos que conforman el modelo. Asimismo, es necesario establecer una nomenclatura coherente y estandarizada, asegurando que cada componente del modelo tenga una identificación única y precisa. También se deben eliminar elementos redundantes o innecesarios que puedan generar inconsistencias o dificultar la interpretación del modelo en software de terceros. La organización de la estructura del modelo permite no solo una exportación más limpia, sino también una mejor gestión de la información durante todas las fases.¹²⁴

La generación de la geometría del modelo es un aspecto fundamental para garantizar una exportación efectiva, siendo necesario modelar los elementos con precisión, respetando dimensiones, proporciones, escalas y relaciones espaciales. Es importante también evitar la presencia de geometrías mal definidas, tales como sólidos abiertos, elementos superpuestos o intersecciones incorrectas, ya que estos pueden generar errores durante la exportación y visualización en otros softwares. La parametrización adecuada de los objetos permite que el modelo mantenga su flexibilidad y que los cambios en dimensiones y propiedades se reflejen de manera automática en el archivo IFC.

Cada software de diseño BIM tiene una base de datos estructurada según sus necesidades y agrupa los elementos como considera más eficiente. De esta manera, se pueden dividir este tipo de programas en:

- Software Nativo IFC con gestión directa de objetos IFC. En este caso el software tiene una base de datos estructurada según sus necesidades específicas. Por ejemplo, ArchiCAD organiza los elementos en Clases, Allplan y Tekla Structures utilizan Componentes y objetos, y Aecosim agrupa en Grupos de datos. Estos grupos de objetos se traducen a clases IFC de manera directa cuando se exportan los archivos. Las tablas de mapeo IFC, que vienen

¹²⁴ The openBIM bootcamp: TOP 10 IFC tips for Revit users. Lejla Secerbegovic & Angel Velez. 2022.

predefinidas en este tipo de softwares, permiten que los elementos se conviertan correctamente a clases IFC. Los usuarios pueden enriquecer o ajustar estas tablas de mapeo si alguna categoría no encaja correctamente, evitando que se genere un elemento "proxy" o elemento sin una clase IFC específica.

- Software con mapeado o conversión IFC. Algunos programas, como Revit o Civil 3D, no gestionan objetos IFC de forma nativa. En su lugar, utilizan un proceso de exportación en el que los objetos dentro del software se mapean a clases IFC mediante tablas de mapeo. Esto significa que los elementos del modelo se convierten en entidades IFC durante la exportación, pero no se gestionan como tales dentro del flujo de trabajo del software. Si hay conceptos que no encajan directamente en el mapeo predefinido, el usuario puede ajustar las tablas para mejorar la exportación y evitar que los elementos se conviertan en "proxy".
- Software que requiere complementos para trabajar con IFC. Existen otros programas, como Blender, Rhinoceros y AutoCAD, que no tienen soporte nativo para gestionar IFC. Para estos casos, es necesario instalar complementos o plugins adicionales para poder trabajar con archivos IFC. Estos complementos permiten la importación, exportación y en algunos casos, la visualización de modelos IFC, pero requieren de un paso adicional para poder manejar este formato de archivo.

Además de lo anterior, las clasificaciones de objetos proporcionan un marco de referencia que ayuda a identificar elementos de manera uniforme según normativas internacionales. La correcta agregación de estas clasificaciones permite que el modelo IFC sea más accesible y compatible con otros sistemas de gestión de datos. De igual forma, los modelos deben contener atributos y propiedades adicionales que permitan describir cada elemento de manera detallada. En la mayoría de los casos, existe una configuración adicional acerca del correcto geoposicionamiento, representado las coordenadas según el sistema de referencia utilizado.

Cabe señalar que desde buildingSmart se ha publicado una serie de fichas de aplicaciones openBIM que recogen las capacidades de diversas herramientas de software,¹²⁵ detallando sus funcionalidades relacionadas con los estándares abiertos, como el IFC. La información proviene directamente de los fabricantes de software y sirve como guía para garantizar flujos de trabajo compatibles. Sin embargo, aparecer en este listado no implica que las aplicaciones estén certificadas por el Programa Oficial de Certificación de buildingSMART.

2.2.3.10. Visualización de archivos IFC

Los archivos IFC deben revisarse cuidadosamente después de la exportación para asegurarse de que se haya realizado correctamente la conversión o guardado de los elementos del modelo. Para este propósito, es recomendable utilizar un visualizador IFC especializado en lugar de abrir o vincular el archivo en un software de creación BIM, ya que estos programas no están diseñados específicamente para la revisión y visualización detallada de archivos IFC. Los visores IFC son herramientas esenciales que permiten inspeccionar y analizar archivos IFC sin modificar el modelo. Además

¹²⁵ Fichas Aplicaciones openBIM. buildingSmart Spain.

<https://www.buildingsmart.es/recursos/fichas-aplicaciones-openbim/>

de ofrecer la visualización gráfica del modelo en 3D, muchos de estos visores permiten explorar la estructura jerárquica del modelo, acceder a propiedades detalladas de los elementos y realizar análisis de calidad e interoperabilidad. Los visores más populares son generalmente gratuitos e incluyen opciones de ventanas de árbol para navegar por la estructura del modelo y ventanas de propiedades para consultar detalles adicionales de los objetos seleccionados.

Dentro de las aplicaciones y entornos de Autodesk, uno de los visualizadores más conocidos es el Autodesk Viewer, una herramienta gratuita basada en la web que soporta más de 50 formatos de archivos, incluyendo IFC. Además de permitir visualizar modelos 3D, ofrece funcionalidades como compartir y comentar sobre los archivos. Autodesk Docs, es otra opción basada en la misma tecnología que Autodesk, pero ofrece funciones extendidas para la gestión de documentos y proyectos, lo que la convierte en una excelente opción para equipos que trabajan de manera colaborativa con archivos IFC. Navisworks Freedom, por su parte, es una solución más completa que ofrece funcionalidades de visualización de propiedades y estructura de los archivos IFC, y además permite ver simulaciones constructivas, incidencias y otros recursos que se hayan creado desde un software de coordinación.

El Open IFC Viewer, desarrollado por Open Design Alliance, es otro visor muy rápido y compatible con las versiones más recientes de IFC, incluida la IFC 4.3, aunque no de manera oficial. BIMvision, desarrollado por Datacomp, es otro visor conocido por su ligereza y facilidad de uso, aunque algunas funciones avanzadas pueden estar limitadas en la versión gratuita. BIMcollab Zoom, desarrollado por BIMcollab, es una plataforma robusta diseñada para la visualización y colaboración entre equipos, la cual permite la incorporación de varios archivos IFC dentro de un mismo proyecto. DDS-CAD Viewer, creado por Nemetschek, es otra herramienta eficiente para la visualización de archivos IFC con soporte para datos de diseño de edificación.

Existen también herramientas basadas en web, como el IFC WebGL, que permiten visualizar representaciones gráficas en 3D de los elementos contenidos en un archivo IFC y, aunque en algunos casos no muestran una estructura en árbol, ofrecen información clave al seleccionar cada elemento, como la clase IFC, el nombre del elemento, y el identificador del elemento. El IFC JAVA Viewer, una aplicación web creada en Java, también permite visualizar modelos 3D y dispone de un árbol de visualización que permite activar o desactivar elementos, leer atributos y resaltar la línea de código correspondiente en el archivo IFC, facilitando una inspección más detallada del archivo.

2.2.3.11. Gestión de la calidad IFC

La coordinación efectiva entre todas las disciplinas involucradas en el proyecto, respaldada por documentos como el BEP y el MIDP, asegura que el modelo mantenga su integridad y precisión a lo largo de su ciclo de vida. Además, la revisión periódica del modelo para verificar aspectos como la consistencia de los identificadores, la ausencia de elementos duplicados y la correcta alineación del modelo federado, es esencial para una gestión eficiente y libre de errores.

La gestión de la información es otro aspecto crítico que requiere de criterios claros y consistentes. Es necesario que se definan correctamente las propiedades y atributos para cada componente del modelo. Esto incluye asegurarse de que los elementos

tengan propiedades correctamente especificadas (Wang, G., 2014). También es importante validar las mediciones de cantidades y confirmar que las propiedades de los componentes sean consistentes a lo largo de las diferentes fases del proyecto. Por ejemplo, los valores de las propiedades, como los espesores de las capas de material o las dimensiones de los elementos, deben ser exactos y mantenerse consistentes.¹²⁶

El proceso de implementación del IFC debe incluir una verificación regular de la estructura del modelo y de las directrices de modelado (Leygonie & Motamedi, 2021). Esto incluye la validación de los requisitos de información y la confirmación de que todos los componentes del modelo estén correctamente definidos y clasificados. Las revisiones deben comprobar también que las configuraciones de exportación estén correctamente implementadas y que cada componente tenga un nombre, tipo y atributos claros. Esto es crucial para asegurar que el modelo sea útil no solo durante todas las etapas posteriores de operación y mantenimiento.

Además de los aspectos estructurales y geométricos, existen otras verificaciones relacionadas con el archivo IFC en sí. El archivo debe cumplir con varios criterios técnicos (Zadeh et al., 2015), como la correcta codificación del archivo digital según el estándar, y no debe superar el tamaño máximo especificado. También debe estar geo-referenciado correctamente en coordenadas y debe contar con todas las entidades geométricas generadas en el software propietario original. De igual forma, se debe asegurar que las entidades tengan materiales adecuados para facilitar la visualización y que el árbol de jerarquía esté correctamente estructurado y alineado con los requisitos. Es igualmente importante que las entidades estén correctamente mapeadas a IFC, evitando al máximo el uso de la entidad tipo proxy. Las entidades deben estar nombradas adecuadamente para poder localizar y referenciar cada instancia fácilmente, y deben contener toda la información no gráfica que haya sido solicitada. También se debe garantizar que no existan identificadores de objeto duplicados, lo cual puede causar problemas de coordinación.¹²⁷ Finalmente, existen herramientas como IFCCompressor que pueden ayudar a reducir el tamaño de los archivos IFC. Estos compresores son especialmente útiles para disminuir hasta un 70% el tamaño del archivo, lo que facilita su manejo y reduce los tiempos de apertura en visores. Mantener los archivos de tamaño optimizado también mejora la eficiencia del proceso y facilita el trabajo con modelos grandes y complejos en plataformas de visualización y coordinación.

Las características más comunes de los softwares de coordinación y verificación de calidad de IFC, como pueden ser Navisworks, Solibri o BIMCollab, incluyen la capacidad de detectar conflictos entre modelos de diferentes disciplinas, asegurar la interoperabilidad entre diversas herramientas BIM, y facilitar la visualización y revisión de modelos federados. Adicionalmente, en la mayoría de los casos, permiten la validación de los datos, asegurando que los modelos sean precisos, estén correctamente clasificados y cumplan con los estándares establecidos. Además, ofrecen herramientas para la comprobación de diferentes versiones de un mismo archivo, la detección de errores dentro de los modelos y la mejora de la colaboración entre los diferentes equipos a través de la integración en plataformas web.

¹²⁶ Digital Quality Management for IFC Project. Tobias Schmidt, TUV SUD. Autodesk. 2021.

¹²⁷ Beyond e-permitting: Framing the Business Case for Automated Rule Checking in AEC in the Era of Big Data. Technical Report. buildingSMART International. 2019.

CAPÍTULO 3: DESARROLLO DE LA INVESTIGACIÓN

3.1. Búsqueda científica

3.1.1. Gestión de datos y Bases relacionales SQL

3.1.1.1. Fundamentos de datos

En la era digital (Serra, 2021), el manejo y análisis de datos se ha convertido en una herramienta esencial para la toma de decisiones estratégicas en diversos ámbitos. La cantidad y diversidad de datos disponibles ha crecido de forma exponencial, convirtiéndose en un activo clave en el sector de la construcción. Este fenómeno ha impulsado el desarrollo de tecnologías avanzadas dedicadas a la captura, almacenamiento, procesamiento y análisis de datos. Por lo tanto, comprender los fundamentos de los datos se ha vuelto crucial para garantizar su correcta gestión y aplicación, a fin de asegurar elecciones informadas y efectivas en distintos contextos.

La adecuada gestión de datos permite transformar volúmenes masivos de información en activos valiosos. En este contexto, la correcta administración de los datos resulta fundamental para evitar la aparición de información incorrecta, incompleta o inexacta.¹²⁸ Este apartado examina los elementos esenciales que rigen la gestión de datos, proporcionando un marco conceptual necesario para su manejo en entornos digitales complejos y dinámicos.

Datos vs. Información

Una de las distinciones más importantes en el manejo de datos es la diferenciación entre datos e información. Un dato se puede definir como la representación simbólica de una variable, que puede ser tanto cuantitativa como cualitativa.¹²⁹ Estos datos pueden adoptar diversas formas, como valores alfanuméricos, coordenadas espaciales o representaciones algorítmicas. En su estado más básico, no poseen un contexto ni una interpretación, lo que significa que carecen de significado por sí mismos. Son simplemente hechos, cifras o variables aisladas que no pueden ser utilizadas directamente para tomar decisiones, ya que no ofrecen una visión comprensible o relevante por sí mismos.

Una correcta organización y categorización de los datos es esencial para garantizar que estos puedan ser interpretados y procesados de manera adecuada. En este sentido, es crucial que los datos sean estandarizados y estructurados de forma que faciliten su almacenamiento, transmisión y análisis. La fragmentación debe superarse mediante la integración de estos datos en formatos coherentes y unificados, que permite su uso posterior en diferentes sistemas y procesos sin perder su integridad o calidad.

Por otro lado, la información surge cuando los datos son procesados, organizados y estructurados de manera adecuada, otorgándoles significado, contexto y relevancia. Por tanto, la información es el resultado de un proceso de transformación que permite interpretar y dar sentido a los datos. Este proceso generalmente involucra el uso de algoritmos, reglas de negocio y modelos analíticos que estructuran los datos

¹²⁸ Fundamentos de Bases de Datos. Abraham Silberschatz. V04. 2002.

¹²⁹ Cambio de paradigma, del dato al conocimiento. Asociación española para la calidad (AEC). 2012.

para extraer valor útil, haciendo que los mismos adquieran un contexto significativo. A diferencia de un dato aislado, la información tiene un propósito claro y puede ser utilizada para interpretar, analizar y comprender fenómenos o hechos específicos. Este proceso de transformación convierte los datos en recursos valiosos para la toma de decisiones (Wang, Y., 2024), generando conclusiones relevantes, identificando patrones y resolviendo problemas específicos. Su correcta categorización, manipulación y almacenamiento también permite transformar el conocimiento potencial en soluciones aplicables y escalables. En disciplinas científicas, técnicas y empresariales, una buena organización de los datos facilita la replicabilidad y consistencia de los análisis, mejorando la calidad de los activos y optimizando la eficiencia en la gestión de proyectos.

Fuentes de datos y estructuras de datos

Las fuentes de datos¹³⁰ son todos aquellos medios de los cuales procede la información, que satisfacen las necesidades de conocimiento de una situación o problema presentado y, que posteriormente son utilizados para lograr unos objetivos esperados. Estas fuentes pueden ser primarias, cuando provienen de observaciones directas o mediciones originales, o secundarias, cuando son recopiladas a partir de otras fuentes preexistentes.

Por otro lado, para garantizar un procesamiento eficiente, los datos deben estructurarse de manera coherente, evitando redundancias y mejorando la integridad de la información. Esta organización no solo influye en la eficiencia de las consultas y la velocidad de recuperación, sino que también determina la escalabilidad de los sistemas y su capacidad de adaptación a distintos volúmenes y tipos de datos.¹³¹

Existen múltiples enfoques para estructurar los datos (Serra, 2021), desde modelos jerárquicos y en red hasta bases de datos relacionales y no relacionales. La elección del modelo depende de factores como la complejidad de las relaciones entre los datos, el volumen de información y los requisitos de acceso. En los sistemas modernos, la organización eficiente de los datos es clave para garantizar la interoperabilidad entre diferentes plataformas y aplicaciones, permitiendo el análisis avanzado y la integración de múltiples fuentes de información. En este contexto, cualquier dato puede categorizarse como estructurado, no estructurado o semiestructurado.

- Los datos estructurados están muy organizados y se comprenden fácilmente por el lenguaje de máquina. Es necesario que tengan una estructura antes de ser integrados en las fuentes de datos. Por lo general, se clasifican como datos cuantitativos y se agrupan en filas y columnas dentro de bases de datos. Al estar organizados, pueden evitar la redundancia y duplicidad de datos. Una de las características fundamentales de este tipo de datos es que son de fácil consulta y manipulación. Sin embargo, requieren de un cierto esfuerzo para ser definidos en un principio, ya que deben adaptarse al esquema de la fuente. Ejemplos de este tipo los podemos encontrar en los modelos BIM, en los que los datos están estructurados y almacenados de una forma ordenada y

¹³⁰ Fuentes de información para la recolección de información cualitativa y cuantitativa. Ubaldo E. Miranda. 2009.

¹³¹ NSW Data Governance. Data Analytics Centre. V1.1. 2021.

jerarquizada y también en otro tipo de fuentes como bases de datos relacionales, sistemas CRM y hojas de cálculo.

En estas soluciones, se incluyen datos tipo nombre, identificadores, fechas, direcciones, geolocalización, dimensiones, etc., los cuales pueden ser generados por máquina, como registros automáticos, datos de sensores y de captura automática, y los que se generan por humanos, como los datos de entrada manuales para registros de información.

- Los datos no estructurados se clasifican con mayor frecuencia como datos cualitativos y no pueden procesarse y analizarse utilizando herramientas y métodos convencionales. Es complicado deconstruirlos ya que no tienen un modelo predefinido. Se trata, por lo tanto, de información que existe en una estructura no normalizada y no identificable. Estos datos suelen trabajarse en bases de datos semiestructuradas o desestructuradas como las NoSQL y en otras soluciones como los Data Lakes. Algunos formatos que entran dentro de esta tipología son los HTMLs, archivos de audio, imagen y video, textos, presentaciones, planos y esquemas 2D, documentos de licitación, actas de reunión, etc.
- Los datos semiestructurados contienen una cierta estructura básica, pero el contenido general no se encuentra estructurado. Muchas veces se caracterizan por tener propiedades organizativas para facilitar su análisis como etiquetas semánticas. Un ejemplo son los correos electrónicos, los cuales tienen ciertos metadatos estructurados como el asunto, la fecha y el destinatario, pero el resto no tiene una estructura definida.¹³²

Tipos de datos

Los datos pueden clasificarse según su naturaleza y estructura, lo que influye en la forma en que son almacenados y procesados. Una clasificación relevante, además de por su estructura, es según el tipo de contenido que representan. Los datos numéricos son aquellos que pueden ser cuantificados y utilizados en operaciones matemáticas, como valores enteros o decimales. Los datos alfanuméricos incluyen caracteres y números en combinación, generalmente utilizados en etiquetas, nombres o identificadores. Además, existen los datos categóricos, que representan variables discretas con un conjunto limitado de valores posibles, como colores o tipos de productos. Otros datos, como los temporales, incluyen información de fechas y horas, fundamentales en análisis cronológicos y series temporales.¹³³ Con el objetivo de detallar cada tipo de dato relevante en entornos de bases de datos y ecosistemas de datos, en el anexo B del presente estudio se detallan los principales tipos según su naturaleza, relacionándolos con los de las bases de datos y con los del formato IFC.

La correcta elección de estos tipos de datos permite optimizar el almacenamiento y garantizar la integridad de la información. En bases de datos relacionales, los tipos de datos están estrictamente definidos para asegurar coherencia y evitar errores en las operaciones. En sistemas de datos masivos, la diversidad de tipos requiere enfoques flexibles que permitan combinar diferentes formatos sin perder eficiencia. Este procesamiento de grandes volúmenes de datos heterogéneos ha impulsado el

¹³² Fundamentos Técnicos de openBIM. Estructuras de Datos, Estándares y Protocolos de Intercambio. buildingSmart Spain. 2025.

¹³³ Data Types and Structures: Detailed Explanation. National Instruments (NI). 2011.

desarrollo de técnicas avanzadas de gestión, como la normalización¹³⁴ y la indexación, para mejorar la rapidez y precisión en la recuperación de información.

Esquemas y metadatos

Los esquemas de datos son modelos que describen cómo se organizan y estructuran los datos dentro de un sistema. En bases de datos relacionales, los esquemas establecen la disposición de tablas, columnas y relaciones, permitiendo una representación coherente de la información. En sistemas no relacionales, los esquemas pueden ser más flexibles, adaptándose a estructuras dinámicas y datos semiestructurados.¹³⁵

Por otro lado, los metadatos son datos que describen otros datos, proporcionando información sobre su estructura, origen y características. En sistemas de gestión de datos, permiten catalogar, buscar y acceder a los datos de manera eficiente. Existen diferentes tipos de metadatos, incluyendo descriptivos, que proporcionan detalles sobre el contenido y contexto de los datos; estructurales, que definen la organización interna de los datos; y administrativos, que incluyen información sobre permisos de acceso, fechas de modificación y estándares de almacenamiento.

El uso de metadatos y esquemas es esencial ya que permite integrar múltiples fuentes de información con distintos formatos. En entornos BIM o Bigdata, donde los volúmenes de información son extremadamente grandes y diversos, la correcta gestión de metadatos y esquemas facilita la organización y recuperación eficiente de datos. Tecnologías como los catálogos de datos y los sistemas de gestión de metadatos han sido desarrolladas para mejorar la administración de grandes repositorios de información, asegurando la trazabilidad y gobernanza de los datos.

3.1.1.2. Bases de datos relacionales y SQL

En el campo de la gestión de datos, las bases de datos relacionales y el lenguaje SQL han sido fundamentales para el almacenamiento, organización y recuperación eficiente de grandes volúmenes de información. Estas bases ofrecen una estructura organizada y flexible que permite mantener los datos en tablas interrelacionadas, lo que facilita el acceso y la manipulación de la información. El lenguaje SQL, por su parte, es el estándar de facto para interactuar con bases de datos relacionales, permitiendo realizar consultas complejas, actualizaciones y manipulaciones de datos de manera eficaz. Este apartado aborda los conceptos clave relacionados con las bases de datos, el modelado de datos, el uso del lenguaje SQL, y la seguridad y control que se implementan para proteger la integridad de la información.

Tipos de bases de datos

Una base de datos es un conjunto organizado de datos que comparten un contexto común, y su propósito es ser almacenado de manera sistemática para ser administrado y utilizado en un futuro. Para garantizar su eficacia en la organización, almacenamiento y recuperación, se utiliza un modelo de base de datos, que define la estructura y las relaciones entre los datos, así como las limitaciones y restricciones para su manejo adecuado. Este modelo puede conceptualizarse como una biblioteca

¹³⁴ Further Normalization of the Data Base Relational Model. IBM Research Laboratory.

¹³⁵ Digital Library Standards & Metadata – The Basics. 2007.

o almacén, en el que la información se organiza de tal manera que facilita su posterior búsqueda, utilización y administración.¹³⁶

En este contexto, es importante destacar algunas características clave de estas bases de datos. En primer lugar, la independencia de los datos, tanto lógica como física, que asegura que las modificaciones en el almacenamiento o en la organización interna no afecten a las aplicaciones que dependen de esos datos. Además, la integridad juega un papel esencial, ya que garantiza que los datos se mantengan consistentes y correctos, lo que es fundamental para asegurar la fiabilidad de la información. La optimización es otro aspecto crucial, ya que busca mejorar la eficiencia en la ejecución de consultas complejas. Por otro lado, la seguridad es una característica indispensable, ya que controla el acceso a los datos, protegiendo su integridad frente a accesos no autorizados. Asimismo, los mecanismos de respaldo y recuperación permiten garantizar la protección de los datos y su restauración en caso de fallos. La accesibilidad es igualmente relevante, ya que permite el acceso a los datos mediante lenguajes de programación, facilitando su integración en diversas aplicaciones. Finalmente, el acceso multiusuario posibilita que diferentes usuarios interactúen con la base de datos simultáneamente sin comprometer la coherencia de la información, lo que resulta esencial en entornos colaborativos.

Por otro lado, la estructura de las bases de datos se organiza en muchos casos a partir de tablas, que son conjuntos de datos interrelacionados entre sí. Estas tablas se componen de: columnas, que representan tipos específicos de información asociada a una entidad, y filas, que contienen los datos correspondientes a cada instancia de esa entidad. Esta organización permite una estructuración eficiente y lógica de los datos, facilitando su acceso y manipulación. A partir de esta estructura, surge el modelo de datos, que es una representación abstracta que define cómo se organiza la información dentro de la base de datos. Este modelo establece las relaciones simbólicas entre los datos, basándose en las entidades del mundo real. Su objetivo principal es crear un lenguaje común que permita una comunicación efectiva entre los distintos actores del sistema, asegurando la estandarización en la gestión de los datos y facilitando la colaboración entre los usuarios.

Adicionalmente, las bases de datos se pueden clasificar según su variabilidad, dividiéndolas en dos tipos principales: estáticas y dinámicas. Las bases de datos estáticas se utilizan principalmente para consultas y lecturas de datos históricos, permitiendo realizar análisis sobre el comportamiento pasado de la información. En cambio, las bases de datos dinámicas son más flexibles, ya que permiten la modificación continua de los datos, incluyendo su edición, adición y eliminación, lo que las hace aptas para aplicaciones que requieren actualizaciones constantes.

Otras clasificaciones de bases de datos se organizan según su finalidad, arquitectura y modelo de administración, cada una diseñada para responder a necesidades específicas en términos de velocidad, escalabilidad, integridad y flexibilidad. Desde bases de datos bibliográficas, hasta bases de datos geoespaciales, la diversidad de tipos responde a requerimientos específicos en distintos sectores.¹³⁷ Además, los modelos de administración de bases de datos han evolucionado desde las estructuras

¹³⁶ Fundamentos de Bases de Datos. Abraham Silberschatz. V04. 2002.

¹³⁷ Comparing Database Types: How Database Types Evolved to Meet Different Needs. Prisma. 2019.

jerárquicas y de red hasta las modernas bases de datos relacionales y no relacionales. A continuación, se presenta una clasificación de los principales tipos de bases de datos, considerando sus características y usos comunes.

Base de Datos	Descripción	Usos Comunes
Bibliográficas	Almacenan registros de información como autores, fechas, títulos y otros datos documentales.	Bibliotecas, investigación.
Directorios	Organizan información con un enfoque personalizado en los campos registrados.	Gestión de usuarios, directorios.
De Texto Completo	Permiten búsquedas eficientes dentro de documentos completos.	Motores de búsqueda, análisis de documentos.
Jerárquicas	Datos estructurados en árbol con relaciones padre-hijo.	Aplicaciones heredadas, gestión de archivos.
De Red	Permiten múltiples relaciones padre-hijo, con una estructura más flexible que la jerárquica.	Sistemas bancarios, telecomunicaciones.
Relacionales	Organizan datos en tablas con claves primarias y foráneas.	Aplicaciones empresariales, e-commerce.
No Relacionales (NoSQL)	Diseñadas para escalabilidad horizontal y datos no estructurados.	Big Data, redes sociales, IoT.
Multidimensionales	Optimizadas para el análisis de datos en múltiples dimensiones.	Business Intelligence, análisis financiero.
Orientadas a Objetos	Basadas en almacenamiento de objetos en bases de datos.	Aplicaciones científicas, BIM, modelado 3D.
Documentales	Almacenan documentos en formato JSON, XML o BSON.	Aplicaciones web, gestión de contenido.
Deductivas	Utilizan lógica formal para inferir información.	Inteligencia artificial, sistemas expertos.
Geoespaciales	Gestionan información geográfica con soporte para coordenadas y mapas.	Sistemas GIS, logística, cartografía.

Tabla 3.1.1. Tipos de bases de datos

Bases de datos relacionales

Las bases de datos relacionales representan un modelo estructurado de almacenamiento de información basado en la organización tabular de datos y sus relaciones a través de claves. Este enfoque, propuesto por Edgar F. Codd en 1970, marcó un hito en la gestión de datos al proporcionar una metodología fundamentada en la teoría de conjuntos y la lógica matemática. Su capacidad para estructurar datos

de manera eficiente ha permitido su adopción generalizada en diversos sectores, garantizando flexibilidad, escalabilidad e integridad en el manejo de grandes volúmenes de información.¹³⁸

El modelo relacional organiza la información en tablas compuestas por registros y atributos, estableciendo relaciones mediante claves primarias y foráneas. Estos vínculos no solo aseguran la unicidad de los registros y evitan la redundancia de datos, sino que también optimizan la coherencia y consistencia del sistema.

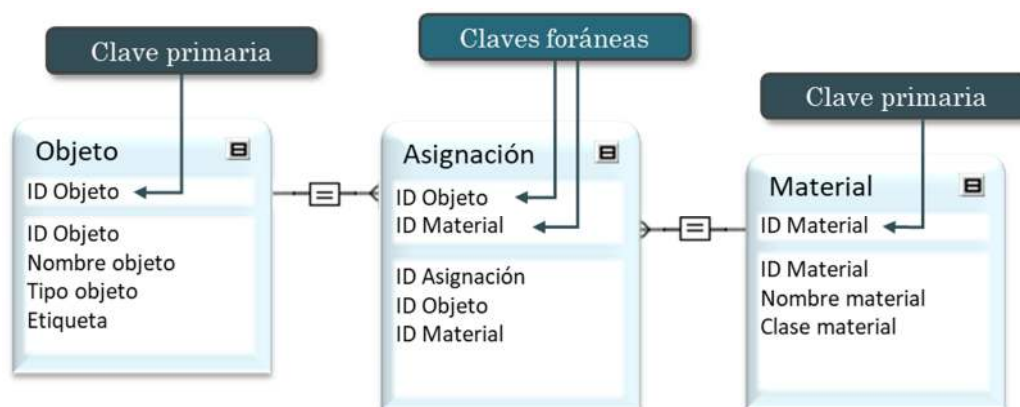


Figura 3.1.1. Claves primarias y foráneas de bases de datos

Dentro de este modelo, existen diversos elementos fundamentales como las relaciones, que almacenan datos estructurados, las restricciones, que imponen condiciones sobre los atributos para garantizar la validez de los datos, y las claves, que permiten la identificación y vinculación de registros. Asimismo, los procedimientos almacenados, ejecutados en lenguajes como SQL, facilitan la automatización de tareas. Para garantizar la calidad y organización de los datos almacenados, la normalización juega un papel crucial en la gestión de bases de datos relacionales. Este proceso permite reducir redundancias mediante la segmentación de la información en múltiples tablas interrelacionadas, lo que evita anomalías en la actualización de datos y mejora la eficiencia del almacenamiento. No obstante, en ciertos contextos, la desnormalización se presenta como una alternativa válida cuando es necesario optimizar la velocidad de recuperación de información, aun cuando ello implique un mayor consumo de almacenamiento y una gestión más compleja de la coherencia de datos.

El correcto funcionamiento de las bases de datos relacionales se fundamenta en los principios ACID (Atomicidad, Coherencia, Aislamiento y Durabilidad), que garantizan la confiabilidad de las transacciones. La atomicidad impide que una transacción quede incompleta, la coherencia mantiene la validez del estado de la base de datos, el aislamiento evita interferencias entre transacciones concurrentes y la durabilidad asegura que los cambios sean permanentes incluso en situaciones de fallo del sistema. Estos principios son complementados por mecanismos de seguridad que regulan el acceso a los datos a través de roles y privilegios, asegurando que solo los usuarios autorizados puedan realizar modificaciones o consultas, lo que refuerza la protección de la información.

¹³⁸ MySQL 8 Reference Manual. Oracle. 2024.

A lo largo del tiempo, las bases de datos relacionales han demostrado ser una solución robusta y confiable para la gestión de información estructurada. Su fundamento en la teoría de conjuntos y la lógica matemática, sumado a su capacidad para garantizar integridad, optimización y seguridad en la manipulación de datos, las consolida como una herramienta indispensable en entornos empresariales y científicos que requieren sistemas eficientes, escalables y de alta disponibilidad.

Sistemas de gestión de bases de datos

Los sistemas de gestión de bases de datos (SGBD) desempeñan un papel esencial en la organización y gestión de los datos (Sweta, 2015), independientemente de su estructura o volumen, ya que permiten a los usuarios interactuar con grandes cantidades de datos de manera eficiente y segura, organizando la información en estructuras optimizadas que facilitan su acceso y actualización. Los SGBD modernos incluyen capacidades avanzadas, como la replicación de datos, la distribución de información en múltiples ubicaciones o la capacidad de manejar grandes volúmenes de datos de manera escalable y eficiente.¹³⁹ Además, destacan por tener un procesamiento y almacenamiento a gran escala, permitiendo la distribución de los datos en múltiples conjuntos, asegurando que las operaciones de acceso y actualización sean rápidas y resilientes e incluyendo también mecanismos de control de acceso.

Asimismo, estos sistemas de gestión cuentan con herramientas para el análisis en tiempo real de los datos, lo que permite a las organizaciones obtener información valiosa rápidamente. La optimización del rendimiento es otro aspecto clave, con la implementación de técnicas como la partición de datos y la indexación avanzada, que permiten realizar consultas más rápidas incluso en grandes volúmenes de información. La gestión de bases de datos en la nube, por su parte, ha introducido un nuevo nivel de flexibilidad, permitiendo a las organizaciones escalar sus capacidades de almacenamiento sin necesidad de infraestructura física, lo que reduce significativamente los costos operativos y aumenta la accesibilidad.

En el anexo C del presente documento se ha generado una tabla con las características más importantes de los principales sistemas de gestión de bases de datos relacionales del mercado.

Lenguaje SQL y sus componentes

El lenguaje SQL (Structured Query Language)¹⁴⁰ es un estándar de programación utilizado para gestionar y manipular datos en bases de datos relacionales. SQL proporciona un conjunto de instrucciones que permiten realizar diversas operaciones sobre los datos, como la consulta, actualización, inserción y eliminación de registros. Entre los componentes principales de SQL se incluyen las instrucciones DDL (Data Definition Language), DML (Data Manipulation Language) y DCL (Data Control Language), cada una de las cuales cumple con una función específica en el manejo de datos.

Las instrucciones DDL permiten definir la estructura de la base de datos, creando o modificando tablas, vistas y otros objetos. Ejemplos de comandos DDL son CREATE,

¹³⁹ Sistemas gestores de bases de datos. MCGRAHILL. 2014.

¹⁴⁰ Structured Query Language. SQL. Relational System Corporation. V1.2. 2001.

ALTER y DROP. Por otro lado, las instrucciones DML se utilizan para manipular los datos dentro de las tablas, permitiendo la inserción (INSERT), actualización (UPDATE), eliminación (DELETE) y consulta (SELECT) de los registros. Las instrucciones DCL, por su parte, se enfocan en el control de acceso y los privilegios de los usuarios, a través de comandos como GRANT y REVOKE, que asignan permisos a los usuarios para realizar determinadas operaciones.

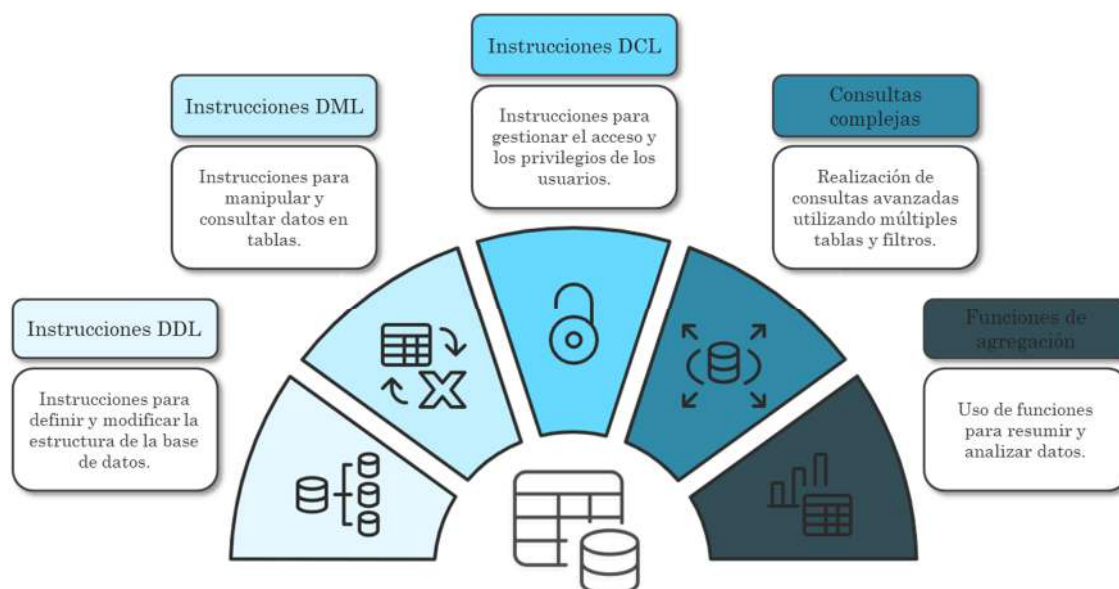


Figura 3.1.2. Gestión y manipulación de bases de datos SQL

Este lenguaje es altamente flexible y permite realizar consultas complejas, uniendo múltiples tablas mediante operaciones como JOIN y filtrando los resultados con cláusulas como WHERE, GROUP BY y ORDER BY. Además, facilita la agregación de datos a través de funciones de agrupamiento y agregación, como SUM(), AVG(), y COUNT(), lo que lo convierte en una herramienta útil para el análisis de datos.

Las consultas en SQL son fundamentales para la extracción y manipulación de datos en bases de datos relacionales. La instrucción principal para realizar consultas es SELECT, que permite extraer información de una o más tablas, según los criterios establecidos. Las consultas pueden incluir múltiples cláusulas que permiten filtrar, ordenar y agrupar los datos de manera eficiente. La cláusula WHERE se utiliza para especificar condiciones que los registros deben cumplir, mientras que ORDER BY permite organizar los resultados, y GROUP BY se usa para agrupar los datos según ciertos criterios, permitiendo el cálculo de agregados.¹⁴¹

Una de las características más poderosas de SQL es la capacidad de unir tablas mediante la instrucción JOIN. Esta operación permite combinar registros de diferentes tablas en función de una columna común, como una clave foránea, lo que permite consultas más complejas y detalladas. Además de las consultas, SQL permite la manipulación de datos a través de las instrucciones INSERT, UPDATE y DELETE, las cuales permiten agregar nuevos registros a las tablas, modificar los registros existentes y eliminar los que ya no son necesarios.

¹⁴¹ SQL Basics and Advanced. IMS, University of Lucknow. 2021.

Seguridad y control de bases de datos

La seguridad y el control en las bases de datos (Omotunde & Ahmed, 2023) son fundamentales para proteger la integridad de los datos y garantizar que solo los usuarios autorizados puedan acceder y manipular la información. En SQL, la gestión de la seguridad se lleva a cabo mediante la definición de roles y permisos, que permiten asignar diferentes niveles de acceso a los usuarios. Los permisos pueden ser específicos para cada operación y pueden ser otorgados o revocados utilizando instrucciones como GRANT y REVOKE. Además, la seguridad de las bases de datos también incluye la implementación de medidas de autenticación y autorización. La autenticación asegura que los usuarios que acceden a la base de datos sean quienes dicen ser, mediante métodos como contraseñas o autenticación de dos factores. La autorización, por otro lado, determina qué acciones puede realizar cada usuario dentro de la base de datos, basándose en los roles y permisos asignados.

3.1.1.3. Gestión del ciclo de vida

El ciclo de vida de los datos describe las diversas etapas por las que pasan los datos desde su creación hasta su eliminación. Comprender este ciclo es crucial para garantizar una gestión adecuada de los datos, asegurando su integridad, disponibilidad y relevancia a lo largo del tiempo.¹⁴² A continuación, se detalla un análisis de cada una de las fases clave del ciclo de vida de los datos.

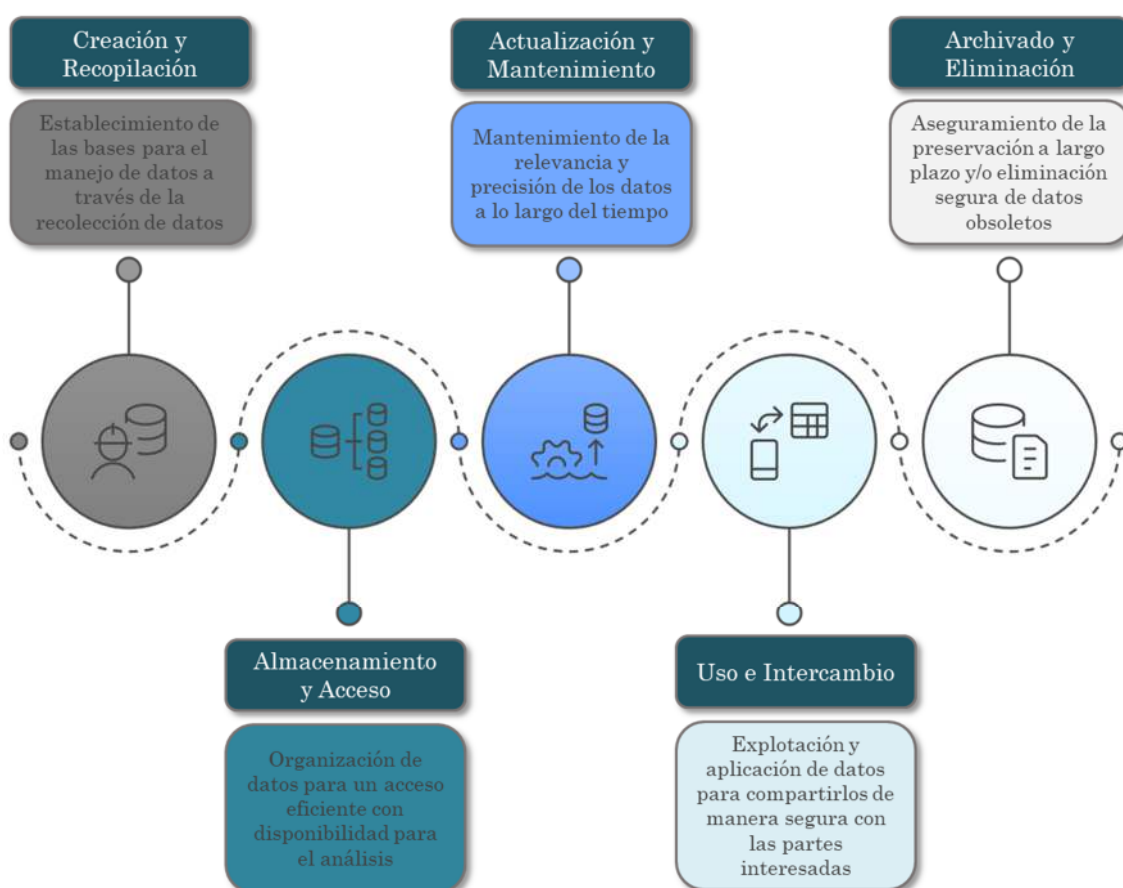


Figura 3.1.3. Ciclo de vida de los datos

¹⁴² Database Lifecycle Management. Achieving Delivery for Databases. Grant Fritchey.

Creación y recopilación de datos

La primera fase en el ciclo de vida de los datos es su creación y recopilación, una etapa fundamental que establece las bases para todo el proceso posterior (Jaganathan Manonmani, 2025). Los datos pueden ser generados de manera directa o indirecta, a través de diversas fuentes. Entre las fuentes de datos primarios se incluyen las observaciones, mediciones, experimentos, encuestas y registros de sensores, mientras que los datos secundarios se obtienen a partir de información ya recopilada en otros contextos, como bases de datos previas o investigaciones publicadas.

La creación de datos debe basarse en metodologías rigurosas (El Arass & Souissi, 2018) para asegurar su fiabilidad. El proceso de recopilación implica la selección de variables relevantes y la determinación de métodos y herramientas adecuadas para su captura. Dependiendo del ámbito, los datos pueden ser de naturaleza cualitativa (como entrevistas o descripciones textuales) o cuantitativa (como mediciones numéricas o datos estadísticos). En ambos casos, la precisión y la coherencia son esenciales, ya que los datos recopilados forman la base sobre la cual se construirán análisis e interpretaciones posteriores.

Almacenamiento y acceso de datos

El almacenamiento y acceso de datos es una fase crítica en el ciclo de vida de los mismos,¹⁴³ ya que asegura que los datos estén organizados, sean accesibles y puedan utilizarse de manera eficiente para su análisis posterior. En esta fase, la elección de las tecnologías y sistemas adecuados para el almacenamiento y el acceso depende de varios factores, como el volumen de datos, la velocidad de acceso requerida y las necesidades específicas de la organización.

El almacenamiento de datos se clasifica en dos tipos principales: primario y secundario. El almacenamiento primario incluye tecnologías como la memoria RAM, que proporciona acceso rápido y directo a los datos. Este tipo de almacenamiento es adecuado para datos que requieren procesamiento inmediato y constante. Por otro lado, el almacenamiento secundario se refiere a soluciones de almacenamiento a largo plazo, como discos duros, sistemas en la nube y data lakes, que permiten almacenar grandes volúmenes de datos de manera eficiente, tanto estructurados como no estructurados.

Las tecnologías de almacenamiento han avanzado significativamente en los últimos años. Los discos de estado sólido (SSD) han reemplazado en muchas ocasiones a los discos duros tradicionales (HDD) debido a su mayor velocidad y fiabilidad. Además, los sistemas distribuidos permiten gestionar grandes volúmenes de datos de manera eficiente y accesible, facilitando la distribución de los datos entre diferentes nodos para optimizar el rendimiento.

Para garantizar un acceso rápido y eficiente, los datos se organizan en estructuras como índices y particiones. Estas estructuras mejoran la velocidad de consulta y recuperación de la información, haciendo que el acceso sea más eficiente y eficaz. El acceso a los datos debe gestionarse con cuidado, implementando políticas de control de acceso para asegurar que solo los usuarios o sistemas autorizados puedan obtener

¹⁴³ NCSS Data Management Guide. National Council of Social Service. 2021.

y manipular la información. La protección de la privacidad y la seguridad de los datos son esenciales para evitar el uso indebido o la pérdida de información sensible.

En sistemas más avanzados, como aquellos empleados en Big Data, el acceso a los datos distribuidos se facilita mediante arquitecturas distribuidas como Hadoop o bases de datos NoSQL. Estas soluciones permiten el acceso paralelo y eficiente a grandes volúmenes de datos dispersos, lo que mejora el rendimiento en entornos donde los datos están distribuidos entre múltiples nodos.

Uno de los desafíos clave en esta fase es garantizar la integridad y escalabilidad de los datos. A medida que los volúmenes de datos crecen, es crucial emplear métodos que permitan gestionar el almacenamiento de forma eficiente sin sacrificar el rendimiento. Técnicas como la compresión de datos y la redundancia de almacenamiento son fundamentales para mantener la integridad y la disponibilidad de los datos. Además, un sistema jerárquico de almacenamiento puede ser útil para organizar los datos según su importancia o frecuencia de acceso. Este enfoque mejora el rendimiento al almacenar los datos más relevantes o más frecuentemente utilizados en niveles de almacenamiento más rápidos.

Actualización y mantenimiento de datos

El ciclo de vida de los datos no concluye con su almacenamiento, sino que también es esencial mantener los datos actualizados y en buen estado a lo largo de su existencia (Zhang & Ye, 2011). Este mantenimiento involucra varias prácticas necesarias, como la validación de su calidad, la corrección de errores y la eliminación de duplicados. A medida que los datos envejecen, pueden volverse obsoletos o requerir ajustes para reflejar cambios en los contextos en los que fueron recolectados. Las actualizaciones periódicas son cruciales para garantizar que los datos sigan siendo relevantes y útiles. Los datos desactualizados pueden generar análisis inexactos, lo que afecta la calidad de las decisiones tomadas a partir de ellos. Por esta razón, las organizaciones deben establecer procedimientos claros y regulares para actualizar los datos y la integración de nuevas fuentes de información que complementen o reemplacen datos previos.

El mantenimiento continuo de los datos incluye la monitorización constante de los sistemas de almacenamiento, asegurando que los datos no se pierdan ni se corrompan durante el almacenamiento o la transferencia. Además, las copias de seguridad regulares son necesarias para proteger la información ante fallos del sistema. Es fundamental que los procesos de actualización se realicen de manera controlada, utilizando transacciones que aseguren la consistencia e integridad de los datos. Esto se logra mediante protocolos que garantizan que las actualizaciones sean atómicas, es decir, que se efectúen por completo o no se efectúen en absoluto, evitando que los datos se corrompan durante el proceso.

El mantenimiento también abarca la optimización periódica de los sistemas de almacenamiento, como la reorganización de índices, la limpieza de registros obsoletos y la reconfiguración de los sistemas para asegurar un rendimiento óptimo. Las bases de datos distribuidas requieren un enfoque especializado en el mantenimiento, ya que gestionan la replicación de datos entre varios nodos, lo que asegura la disponibilidad continua y la tolerancia a fallos. Además, es necesario

validar las actualizaciones para garantizar que la información sigue siendo precisa y libre de errores.

Un aspecto igualmente relevante en el mantenimiento de los datos es la implementación de políticas de archivado. Los datos menos utilizados o históricos se mueven a soluciones de almacenamiento más económicas pero igualmente seguras. Este proceso permite optimizar el rendimiento y la capacidad de los sistemas de almacenamiento, reduciendo los costos operativos a largo plazo. Finalmente, la seguridad es un factor crucial en el mantenimiento de los datos, ya que es necesario proteger la información tanto en tránsito como en reposo, evitando accesos no autorizados.

Uso e intercambio de datos

Una vez que los datos están correctamente almacenados y actualizados, su uso y intercambio se convierten en la siguiente fase crucial en su ciclo de vida. Los datos tienen un valor significativo cuando se emplean para análisis, investigaciones, toma de decisiones o creación de productos y servicios. Este proceso puede involucrar la interpretación de los datos mediante métodos estadísticos, algoritmos de machine learning o análisis cualitativos, dependiendo de la naturaleza de los datos y los objetivos perseguidos.¹⁴⁴

En muchos casos, los datos se comparten entre diferentes actores, como organizaciones, instituciones académicas o socios comerciales, lo que fomenta la colaboración y permite maximizar el valor de la información. El intercambio de datos debe realizarse con transparencia y bajo estrictos protocolos de seguridad para garantizar que la información no sea mal utilizada o divulgada sin el debido consentimiento.

La interoperabilidad de los datos también juega un papel clave en esta fase. Para que los datos sean utilizados y compartidos eficientemente, deben estar disponibles en formatos y estándares comunes que permitan su integración con otras bases de datos y sistemas de información. Los frameworks de interoperabilidad y las plataformas de datos abiertos están facilitando el intercambio de datos a nivel global, lo que permite la creación de nuevas aplicaciones y modelos predictivos.

Archivado y eliminación de datos

El ciclo de vida de los datos no termina con su uso ya que, eventualmente, los datos llegan a un punto en el que ya no son necesarios para las operaciones o decisiones inmediatas. En esta fase, los datos deben ser archivados o eliminados, según corresponda.

El archivado de datos (Jaganathan Manonmani, 2025) implica su transferencia a sistemas de almacenamiento a largo plazo, donde permanecen accesibles para su consulta futura, pero no ocupan espacio en los sistemas de acceso rápido. El archivado de datos debe realizarse de forma que asegure su integridad y accesibilidad durante años o incluso décadas. Los sistemas de archivo deben cumplir con normativas específicas sobre la conservación de datos, como las normativas de

¹⁴⁴ NCSS Data Management Guide. National Council of Social Service. 2021.

protección de datos personales o las regulaciones industriales, que exigen la retención de ciertos datos durante períodos prolongados.

Por otro lado, la eliminación de datos se refiere al proceso de destruir o hacer inaccesibles los datos que ya no son necesarios o que se consideran obsoletos. Esto debe llevarse a cabo de acuerdo con políticas de seguridad y cumplimiento, asegurando que los datos se eliminen de manera irreversible y sin dejar rastro que pueda ser recuperado. En la actualidad, la eliminación segura de datos es una preocupación crítica, especialmente en lo que respecta a la protección de la privacidad y la seguridad de la información.

3.1.1.4. Calidad de los datos

Los datos constituyen un recurso de gran valor para las empresas y organizaciones, ya que facilitan el uso, modificación y redistribución de la información de la empresa. Sin embargo, la utilidad de estos datos depende en gran medida de su calidad, ya que errores en su estructura, inconsistencias o falta de documentación pueden dificultar su interpretación y aplicación en diversos contextos. La implementación de estándares y buenas prácticas resulta fundamental para garantizar la calidad, accesibilidad e interoperabilidad de los datos, permitiendo su correcta reutilización.

Según la norma ISO 8402,¹⁴⁵ la calidad se define como el conjunto de propiedades y características de un producto o servicio que le confiere su aptitud para satisfacer unas necesidades expresadas o implícitas. Existen diversas metodologías y estándares que permiten evaluar y mejorar esta calidad y su adopción facilita la validación y mejora de los datos, asegurando su utilidad y fiabilidad.

Diversos problemas pueden comprometer la calidad de los datos (Pongoh et al., 2024). Uno de los más frecuentes es la publicación en formatos no procesables, como documentos PDF o imágenes, lo que dificulta su extracción y análisis automatizado. Para evitar esta limitación, es esencial el uso de formatos abiertos como CSV, JSON, XML o RDF, que permiten una manipulación eficiente y la integración con otras fuentes de información. Asimismo, es fundamental utilizar una codificación de caracteres estandarizada, como UTF-8, para garantizar la correcta representación de caracteres especiales y evitar errores en su procesamiento. La asignación de nombres claros y consistentes a las columnas de datos también contribuye a mejorar su comprensibilidad y facilita su documentación en metadatos estructurados.¹⁴⁶

Otro aspecto crítico es la completitud de los datos. La ausencia de valores en los conjuntos de datos puede afectar la precisión de los análisis y generar interpretaciones erróneas. Para mitigar este problema, es necesario documentar adecuadamente los valores nulos y emplear convenciones uniformes, como el uso de "NULL" o "NA", evitando la presencia de celdas o datos vacías o la aplicación de múltiples convenciones dentro de un mismo valor. La duplicidad de registros representa otra problemática relevante, especialmente en la integración de datos provenientes de distintas fuentes. La implementación de procesos de validación y

¹⁴⁵ ISO 9000:2015. Quality management systems — Fundamentals and vocabulary.

¹⁴⁶ Guía práctica para la mejora de la calidad de datos abiertos. 2022. Ministerio de Asuntos Económicos y Transformación Digital.

normalización permite identificar y eliminar registros repetidos, asegurando la coherencia de la información y reduciendo el riesgo de sesgos en los análisis.

La estandarización de los valores de datos es un factor clave para garantizar su interoperabilidad y facilitar su comparación con otras fuentes. En muchos casos, la falta de uniformidad en la denominación de entidades, unidades de medida o categorías dificulta la correlación entre diferentes fuentes. Para evitar esta situación, el uso de vocabularios controlados y estándares reconocidos es clave. Asimismo, resulta fundamental proporcionar una cantidad adecuada de datos, asegurando que la información generada sea suficientemente detallada para su análisis, sin resultar excesivamente fragmentada ni sobrecargada de registros irrelevantes.

El uso de formatos estandarizados para la representación de datos temporales y numéricos es otro aspecto esencial para mejorar la calidad y consistencia de los datos. En el caso de las fechas, se recomienda emplear el estándar ISO 8601, que establece una representación uniforme en la forma AAAA-MM-DD. Para los valores numéricos, se sugiere utilizar el punto como separador decimal y evitar el uso de separadores de millar o espacios en blanco, previniendo así errores en el procesamiento automático. La mezcla de escalas numéricas dentro de un mismo conjunto debe evitarse, ya que cambios inconsistentes en las unidades de medida pueden generar confusión y dificultar la interpretación de los datos. En caso de que un cambio de escala sea necesario, debe documentarse adecuadamente y, cuando sea posible, ofrecer los datos en ambas escalas para facilitar su reutilización.

Los conjuntos de datos que incluyen información geográfica requieren un tratamiento específico para garantizar su correcto procesamiento. Las coordenadas geográficas deben publicarse en grados decimales y en columnas separadas para latitud y longitud, evitando representaciones en grados sexagesimales que dificulten su integración en sistemas de información geográfica. Además, es recomendable emplear formatos especializados, como IFC, landXML, SHP o KML, que permiten una mejor manipulación y visualización de datos espaciales. La publicación de este tipo de información en formatos abiertos y estandarizados facilita su uso en diversas aplicaciones y mejora su interoperabilidad con otras fuentes de datos.

Otro problema frecuente en la generación de datos es la inclusión de subtotales, totales y agrupamientos dentro de los conjuntos de información. Este tipo de agregaciones genera inconsistencias y limita la reutilización de los datos en su nivel más desagregado. Para evitar esta situación, hay que proporcionar los datos en su forma más granular posible, permitiendo que los usuarios realicen sus propios cálculos en función de sus necesidades. De manera similar, la fragmentación de datos y su dispersión en múltiples fuentes dificultan su localización y acceso. Para mejorar la accesibilidad, es importante la creación de bases centralizadas que integren toda la información disponible y permitan una búsqueda eficiente.

El cumplimiento en la generación y entrega de datos requiere además una adecuada documentación y especificación de su estructura. La descripción de metadatos permite mejorar la interoperabilidad y facilita su integración con otras plataformas.

La existencia de metadatos claros y completos contribuye a mejorar la transparencia y fiabilidad de los datos, optimizando su potencial de reutilización.¹⁴⁷

Por otra parte, el uso de herramientas especializadas (Seiß, 2019) facilita la validación y mejora de la calidad de los datos, a partir del uso de diversas plataformas y aplicaciones que permiten la detección de errores de formato, la conversión entre distintos estándares y la verificación de la coherencia de la información. En este contexto, la gestión de calidad en proyectos sigue un proceso estructurado que abarca la planificación de la calidad y el control de calidad.

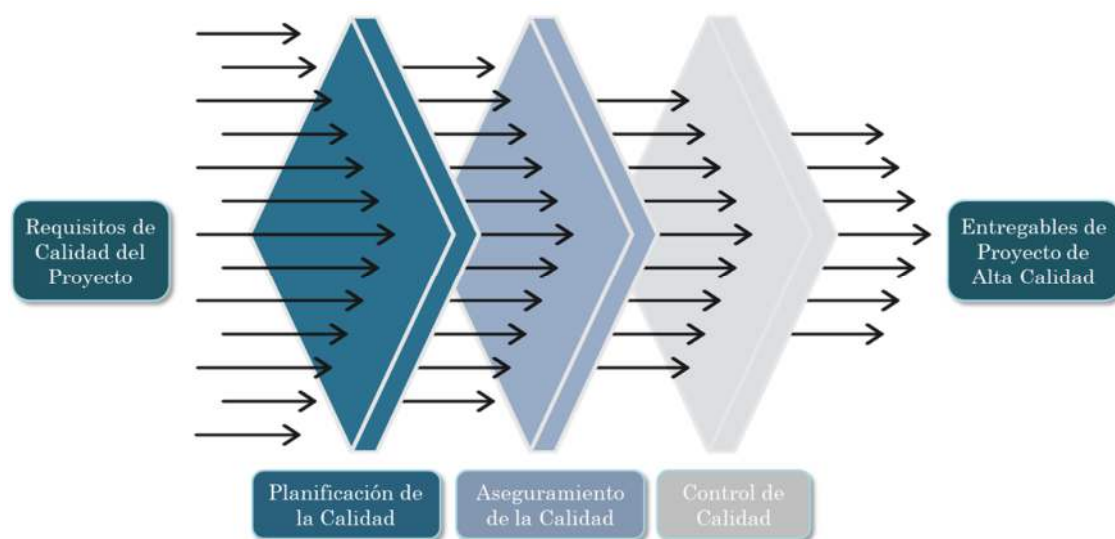


Figura 3.1.4. Gestión de la calidad de los datos

Planificación de la calidad

La planificación representa la fase preliminar en la gestión de calidad dentro de un proyecto y se desarrolla antes de su inicio. Su propósito es definir los requisitos de calidad, establecer los estándares a seguir y diseñar los procesos necesarios para asegurar que los resultados cumplan con las expectativas establecidas. En esta etapa se identifican los objetivos de calidad, se determinan los criterios de aceptación y se diseñan los procedimientos para su cumplimiento. Además, implica la selección de metodologías, herramientas y recursos adecuados para la gestión de la calidad, garantizando que el equipo de trabajo esté alineado con los estándares requeridos y cuente con la capacitación necesaria.

Uno de los principios fundamentales de la planificación de la calidad es la prevención de errores y defectos a lo largo del desarrollo del proyecto. Esto se logra mediante la optimización de procesos, la implementación de controles preventivos y la creación de protocolos de auditoría que permitan la detección temprana de desviaciones. En el contexto de la gestión de datos, la planificación de la calidad adquiere una relevancia particular, ya que implica la definición de estructuras de datos, la implementación de reglas de validación y la estandarización de formatos con el fin de asegurar la consistencia y confiabilidad de la información. Asimismo, abarca la

¹⁴⁷ Data.europa.eu. Data Quality Guidelines. Publication Office of the European Union. 2021.

definición de procedimientos para la captura, almacenamiento y mantenimiento de datos, minimizando los riesgos de errores y pérdidas de información.

Para garantizar la fiabilidad, accesibilidad y utilidad de los datos desde su origen, resulta esencial adoptar un enfoque estructurado basado en estándares. Un elemento clave en este proceso es la definición de criterios de calidad, la implementación de mecanismos de control y la asignación de responsabilidades dentro de las entidades encargadas de la generación, gestión y entrega de los datos. La estrategia de calidad debe comenzar con la identificación de los requisitos específicos que deben cumplir los datos, considerando su propósito y el perfil de los usuarios que los emplearán. Para ello, es necesario establecer métricas y umbrales para evaluar la exactitud, completitud, coherencia, accesibilidad, interoperabilidad y trazabilidad de los datos.

La implementación de protocolos de recolección con controles de validación automatizados ayuda a evitar la introducción de datos inconsistentes o incompletos desde su origen. Los sistemas de captura deben diseñarse para detectar anomalías en tiempo real, emitiendo alertas sobre posibles incoherencias y permitiendo su corrección inmediata. Asimismo, la estandarización de formatos y estructuras de datos desde el inicio del proceso reduce la necesidad de transformaciones posteriores, disminuyendo la probabilidad de errores derivados de la conversión entre distintos modelos de datos.

La planificación de la calidad debe incluir la gestión adecuada de los metadatos, ya que estos proporcionan información esencial sobre la estructura, origen y restricciones de los datos publicados. Una documentación clara y completa favorece la reutilización de los datos y asegura que los usuarios comprendan su naturaleza y limitaciones. Para ello, se recomienda el uso de esquemas de metadatos normalizados, que permitan describir conjuntos de datos de manera estructurada y compatible con otras plataformas. Asimismo, la planificación debe contemplar la especificación de versiones y actualizaciones de los datos, garantizando la coherencia con versiones anteriores y minimizando la pérdida de información.

El acceso a los datos debe organizarse de manera que garantice tanto su disponibilidad como la eficiencia en su distribución. La implementación de infraestructuras escalables y mecanismos de consulta optimizados permite atender la demanda de información sin comprometer el rendimiento de los sistemas. El uso de interfaces de programación de aplicaciones (APIs) facilita el acceso dinámico a los datos y reduce la necesidad de descargas masivas, mejorando la eficiencia en el procesamiento. Asimismo, la planificación de la calidad debe contemplar la adopción de formatos adecuados para distintos tipos de datos, asegurando que los usuarios puedan utilizarlos sin necesidad de conversiones adicionales.

Finalmente, la planificación de la calidad debe incluir estrategias de formación y sensibilización dirigidas a los responsables de la generación de datos. La capacitación en buenas prácticas de gestión de datos contribuye a mejorar la calidad desde su origen y reduce la necesidad de correcciones posteriores. Además, la promoción de la colaboración entre equipos y empresas favorece la adopción de estándares comunes y la armonización de procedimientos, lo que fortalece la interoperabilidad de los datos a nivel global.

Aseguramiento de la calidad

El aseguramiento de la calidad es un enfoque preventivo orientado a garantizar que los procesos involucrados en la gestión de datos sean adecuados y eficaces para la producción de resultados de alta calidad. Este enfoque analiza las estrategias de calidad en el contexto del manejo de datos, abarcando auditorías, revisiones de procesos y certificaciones que aseguran el cumplimiento de los requisitos establecidos.¹⁴⁸ El punto fundamental de este aseguramiento es la prevención de defectos mediante la optimización de procesos y la mejora continua. Su aplicación en proyectos de datos implica la definición de estándares, la implementación de auditorías y la revisión sistemática de procedimientos con el objetivo de garantizar que la información generada sea precisa y confiable. Este enfoque es esencial para evitar problemas en etapas posteriores del procesamiento de datos, minimizando el impacto de errores y mejorando la eficiencia operativa.¹⁴⁹

Las principales estrategias para el aseguramiento de la calidad dentro de los entornos de datos se detallan a continuación:

- **Optimización de Procesos:** La definición de estándares y metodologías adecuadas permite mejorar la calidad desde el inicio. Esto incluye la implementación de normas para el tratamiento de datos faltantes, la estandarización de formatos y la definición de protocolos de validación.
- **Auditorías y Revisiones de Calidad:** Se llevan a cabo inspecciones periódicas de los procesos para identificar desviaciones respecto a los estándares definidos. Estas auditorías incluyen la verificación de la integridad de los datos y su conformidad con los criterios establecidos.
- **Capacitación y Mejora Continua:** La formación constante del equipo de trabajo en metodologías de aseguramiento de la calidad es esencial para fortalecer las buenas prácticas y promover la mejora continua de los procesos.
- **Trazabilidad y Registros de Calidad:** La documentación del historial de modificaciones, correcciones y mejoras realizadas en los datos facilita la rendición de cuentas y permite un análisis retrospectivo para optimizar los procedimientos.

Control de la calidad

El control de calidad es un enfoque correctivo centrado en la inspección, verificación y validación de los productos o servicios antes de su entrega. Su propósito principal es identificar defectos y corregirlos antes de que los entregables lleguen al usuario final o a la siguiente fase del proyecto. A través de inspecciones, pruebas y monitoreo, el control de calidad (Lino et al., 2024) garantiza que los procesos y resultados finales cumplan con los estándares definidos, minimizando errores y asegurando la fiabilidad de los datos procesados.

Este control en procesos de datos implica la aplicación de inspecciones y pruebas que permiten verificar la conformidad de los productos o proyectos entregados con los

¹⁴⁸ Use of Digital Technologies for QA/QC of MiC Modules. Construction Industry Council (CIC). 2021.

¹⁴⁹ Data Quality. The Field Guide. Thomas C. Redman. 2018.

estándares establecidos. Esto abarca la validación de datos, el análisis de precisión, la realización de pruebas unitarias y la revisión de reportes, asegurando que cada parte cumpla con los requisitos de calidad definidos. La verificación de resultados finales es una etapa esencial en la que se revisan exhaustivamente los datos generados para garantizar la ausencia de errores en la base de datos final. Este proceso incluye la detección y eliminación de datos duplicados, la identificación de valores atípicos y la corrección de inconsistencias en los informes generados.

Para garantizar la exactitud sintáctica, los datos deben cumplir con formatos predefinidos. Por ejemplo, los números de identificación deben ajustarse a una estructura establecida, como el caso de un número de expediente, que debe expresarse con una combinación específica de valores alfanuméricos. En términos de exactitud semántica, se verifica que los valores registrados sean coherentes con la realidad, como en el caso de los nombres, los cuales deben corresponder a valores válidos y esperados dentro del contexto del sistema.

El rango de exactitud es otro aspecto fundamental, ya que ciertos valores deben pertenecer a un conjunto predefinido. Un ejemplo es la categorización de tipos de incidencias, los cuales deben limitarse a opciones específicas para garantizar uniformidad y evitar errores en la clasificación. La integridad referencial juega un papel clave en la calidad de los datos, asegurando que las relaciones entre entidades sean consistentes. Esto significa que ciertos atributos, como un código de provincia en una tabla de municipios, deben existir en la tabla de referencia para evitar inconsistencias y datos huérfanos.

Cuando se detectan fallos en los datos, se implementan acciones correctivas inmediatas para alinear los resultados con los estándares establecidos. Este mecanismo de corrección permite evitar la propagación de errores en etapas posteriores del proceso y preservar la integridad de la información. Para complementar estas acciones, el monitoreo de la calidad se lleva a cabo mediante mecanismos de supervisión que permiten la detección temprana de problemas emergentes y la aplicación de medidas correctivas de manera proactiva. Este monitoreo continuo facilita además la identificación de tendencias y patrones de errores recurrentes, lo que contribuye a una mejora sistemática de los procesos.

La consistencia del formato es otro criterio básico en el control de calidad, ya que permite que los datos mantengan una estructura uniforme a lo largo de las diferentes tablas y sistemas. Un ejemplo es la validación de direcciones de correo electrónico, las cuales deben seguir una sintaxis específica que combine nombres, dominios y extensiones de país. Asimismo, la consistencia semántica asegura que los valores cumplan con reglas lógicas, como la validación de fechas en registros de personas, donde la fecha de creación de un usuario debe ser igual o anterior a la fecha actual.

El riesgo de consistencia también debe considerarse en los procesos de control de calidad. Por ejemplo, los identificadores únicos, como los números de identificación, deben ser distintos para cada entidad registrada en la base de datos. La detección de duplicados es un procedimiento esencial para evitar la redundancia y asegurar la unicidad de la información.

La validación en fases o hitos clave del proceso garantiza que cada etapa cumpla con los estándares de calidad antes de avanzar a la siguiente. Estas validaciones

incluyen pruebas de integridad de datos y revisiones de la documentación asociada, asegurando que la información procesada mantenga su fiabilidad. La automatización de pruebas de calidad es un elemento crucial en este contexto, ya que el uso de herramientas automatizadas para la validación de datos incrementa la eficiencia de las revisiones. Estas herramientas permiten detectar anomalías de manera rápida y precisa, reduciendo el margen de error humano y optimizando la supervisión de la calidad.¹⁵⁰

Otro aspecto fundamental del control de calidad es la evaluación de la precisión y consistencia de los datos,¹⁵¹ donde se analizan métricas clave para verificar que la información sea uniforme a lo largo de diferentes sistemas y bases de datos, permitiendo la detección de discrepancias y asegurando la coherencia de la información procesada. Adicionalmente, las pruebas de rendimiento se llevan a cabo para evaluar la capacidad del sistema de datos bajo condiciones de carga extrema. Estas pruebas permiten identificar posibles fallos en la integridad y estabilidad de la información almacenada, asegurando que los sistemas puedan operar eficientemente en diversas condiciones de uso.

Antes de que los datos sean utilizados o publicados, se lleva a cabo una revisión final de calidad con el fin de confirmar su integridad, exactitud y completitud. Esta revisión abarca un análisis exhaustivo de la información y una validación de su conformidad con los estándares definidos. Además, la trazabilidad y el registro de errores juegan un papel clave en el control de calidad, ya que la documentación de todas las incidencias detectadas, junto con detalles sobre su causa raíz y las acciones correctivas implementadas, facilita un análisis retrospectivo que contribuye a la mejora de los procesos futuros.

¹⁵⁰ Automated BIM data validation integrating open-standard schema with visual programming language. Advanced Engineering Informatics. 2019

¹⁵¹ UNE-ISO/IEC 25024:2023. Ingeniería de sistemas y software. Requisitos y evaluación de la calidad de sistemas y software. Medición de la calidad de los datos.

conceptos abstractos, procesos y personas. Con ello, se consigue definir como se proyecta, se construye y se opera el conjunto de objetos y sistemas que conforman el activo de construcción.

En cuanto a la estructura general del esquema (Borrmann et al., 2018a), el IFC organiza la información mediante una jerarquía estructurada, en la que cada entidad hereda atributos y comportamientos de sus entidades superiores. La base de esta jerarquía es *IfcRoot*, que define la raíz principal del esquema a través de identificadores y atributos generales. A continuación de esta primera definición, *IfcObjectDefinition* agrupa entidades que representan objetos o conjuntos de objetos, *IfcRelationship* establece las conexiones entre diferentes elementos del modelo e *IfcPropertyDefinition* define las propiedades específicas de los elementos, como dimensiones, materiales o características de desempeño, permitiendo una descripción más detallada de los objetos dentro del modelo.



Figura 3.1.6. Definiciones principales del esquema IFC 4.3

El concepto de clase dentro de este esquema IFC sigue los principios de la programación orientada a objetos. Una clase es un modelo que define atributos y métodos comunes para un conjunto de objetos. La estructura de estas clases permite trabajar con la herencia, la encapsulación y el polimorfismo, aportando flexibilidad y robustez en la representación digital de los activos construidos. Cada una de estas clases puede contener propiedades predefinidas y atributos que garantizan una representación estandarizada de los elementos constructivos. Por ejemplo, la clase *IfcWall* define las propiedades generales de los muros dentro de un modelo IFC, incluyendo atributos como material, espesor y relaciones con otros elementos. Estas clases IFC presentan una estructura heterogénea, ya que cada entidad responde a distintos usos y necesidades dentro del proceso constructivo. Algunas clases definen elementos físicos, mientras que otras establecen relaciones o proporcionan información complementaria.

El lenguaje EXPRESS se utiliza en la definición del esquema para establecer la estructura de cada clase, organizando atributos, relaciones y comportamientos específicos. De esta manera, se garantiza que las clases sean interpretadas de manera uniforme en diferentes plataformas BIM, facilitando la interoperabilidad y coherencia de los datos en todas las etapas del proyecto.

Cabe señalar que el esquema IFC ha evolucionado significativamente en el ámbito de la edificación desde sus primeras versiones hasta la actual versión 4.3, con la incorporación de nuevas clases y la eliminación de otras obsoletas para mejorar la representación de los modelos BIM. A lo largo de este proceso, se han refinado diversas entidades para optimizar la organización y el nivel de detalle de los elementos constructivos.¹⁵³ En IFC 4, por ejemplo, se reorganizaron las clases relacionadas con espacios y zonas y también se introdujeron mejoras en las definiciones de componentes de instalaciones, facilitando una mejor clasificación de los sistemas mecánicos, eléctricos y de fontanería. Por otro lado, en la versión 4.3, aunque el enfoque general se ha ampliado hacia infraestructuras, el esquema de edificación ha mantenido su refinamiento con ajustes en la semántica de clases existentes y la consolidación de categorías que permiten una mayor coherencia y precisión de los modelos de información.¹⁵⁴

En los últimos años, además del estudio y desarrollo de la versión 5 del esquema por parte de la comunidad de buildingSmart, se han publicado diferentes estudios para ampliar las clases y definiciones del esquema, abordando otros áreas y recursos de la construcción. Estos nuevos enfoques (Ait, 2021; Zhao, 2020) han explorado diferentes escenarios para clases IFC de arquitectura, ingeniería y obra con el fin de mejorar su eficiencia y aplicabilidad en la gestión de proyectos. Otros estudios (Emunds, 2021; Ramaji, I. et al, 2020) han explorado el uso de archivos IFC para evaluar el desempeño energético y optimizar el consumo en edificaciones o para involucrar dentro del esquema todo el ciclo de vida del activo, mejorando su gestión. También la integración del 4D (Yang, 2021) ha permitido una planificación más precisa de proyectos, fortaleciendo la vinculación de datos temporales y la eficiencia en cronogramas de construcción.

¹⁵³ La evolución de IFC: el camino hacia IFC5. Píldoras BIM. buildingSmart Spain. 2024.

¹⁵⁴ IFC 4.3. Annex F: Changelogs IFC 4, IFC 4.1, IFC 4.2, IFC 4.3_ADD2. buildingSmart International.

3.1.2.2. Definiciones de objetos

La entidad *IfcObjectDefinition* representa una generalización de cualquier objeto o proceso tratado semánticamente, ya sea como un tipo o como una instancia concreta. Esta definición de objeto puede ser aplicada tanto a tipos de objetos, que son plantillas o categorías dentro del modelo, como a ejemplares específicos, que corresponden a instancias particulares de dichos objetos. A través del atributo heredado *Name*, cada *IfcObjectDefinition* puede ser nombrado de forma que el nombre sea reconocible por el usuario y se utilice como etiqueta para identificar dicho ejemplar. Adicionalmente, el atributo heredado *Description* permite incluir una descripción detallada que enriquece la comprensión del objeto dentro del modelo.

Con respecto a los tipos, la entidad *IfcTypeObject* representa definiciones genéricas de elementos que comparten características comunes. Estas definiciones de tipo incluyen propiedades y atributos que son inherentes a todas las instancias derivadas de ese tipo. Esta metodología no solo optimiza el tamaño del archivo al evitar la duplicación de información, sino que también facilita la gestión y modificación de propiedades comunes, ya que cualquier cambio en el tipo se refleja automáticamente en todas sus instancias. Las entidades para tipos de objetos se pueden localizar dentro del esquema IFC bajo las entidades de la clase *IfcBuiltElementType*.¹⁵⁵

Otra característica importante dentro de la jerarquía de *IfcObjectDefinition* es el concepto de contexto o *IfcContext*. Un contexto define el entorno o ámbito dentro del cual un tipo de objeto o instancia de objeto se establece, ya sea en un proyecto específico o en una biblioteca de objetos. Este concepto permite clasificar y estructurar los objetos en función de sus roles y ubicaciones dentro del proyecto.

Objetos

Los objetos en el esquema IFC, denominados *IfcObject*, son unidades independientes de información que pueden contener o hacer referencia a otras piezas de información. Estos objetos participan en diversas relaciones fundamentales para la estructura y organización del modelo, permitiendo una representación integral del proyecto. Dentro de estos objetos, se distinguen dos grandes categorías: los objetos físicamente tangibles y los objetos intangibles. Los primeros son elementos constructivos concretos y perceptibles que forman la edificación, como muros que definen espacios, o pilares y forjados que sostienen la estructura. Por otro lado, los segundos incluyen elementos que, aunque no son construcciones en sentido estricto, son partes esenciales del contexto del proyecto. Ejemplos de esta categoría son los espacios interiores, definidos por límites virtuales, pero con características funcionales y de uso o los sistemas de coordenadas de referencia.

En conjunto, los objetos en IFC no solo representan componentes físicos o virtuales del proyecto, sino que también se interrelacionan de manera que permiten una gestión y organización coherente de la información a lo largo del ciclo de vida del edificio. Esta amplitud de la definición de *IfcObject* se articula en seis conceptos fundamentales que proporcionan una estructura más detallada para la información del proyecto:

¹⁵⁵ Manual de IFC para Revit 2.0. Instrucciones detalladas para gestionar archivos IFC.

- *IfcProcess* (Procesos): la noción de objeto dentro del esquema se extiende a la dinámica del proyecto, incorporando las actividades y secuencias de trabajo necesarias para su realización. Esto incluye tareas específicas de construcción, mecanismos de control como la gestión de costes a través de partidas presupuestarias, y la identificación de los agentes y las personas que participan activamente en las diversas etapas.
- *IfcActor* (Agentes): concepto enfocado en la identificación y caracterización de todos los agentes que intervienen en el ciclo de vida de la construcción. Permite registrar información crucial sobre cada participante, como su nombre o denominación, su dirección de contacto, la organización a la que pertenece y el rol específico que desempeña dentro del proyecto.
- *IfcGroup* (Grupos): esta definición se caracteriza por la creación de agrupaciones lógicas de objetos basadas en criterios específicos. Estos grupos no poseen una representación geométrica ni una ubicación espacial intrínseca. Su función primordial es la de organizar y categorizar elementos según una característica compartida o una relación funcional.

Dentro de esta definición se encuentran los sistemas o *IfcSystem*, los cuales se definen como conjuntos de elementos que comparten una función común y se dividen en sistemas de construcción, como cimentaciones o elementos de fachada y sistemas de distribución, como sistemas eléctricos, de fontanería o de climatización. Las zonas y los sistemas de análisis estructural también se engloban dentro de estos sistemas de construcción.

- *IfcControl* (Controles): estos controles actúan como un marco abstracto para definir las restricciones y directrices que influyen en la planificación y la ejecución del proyecto. Abarca factores como las demandas y requisitos del cliente, las limitaciones presupuestarias, el cumplimiento de las normativas y estándares de construcción vigentes, y los plazos de entrega establecidos. Al formalizar estas restricciones dentro del modelo, se facilita la toma de decisiones, asegurando la coherencia del proyecto.
- *IfcResource* (Recursos): este tipo de objeto proporciona el medio para modelar la información relativa a los recursos necesarios para la ejecución de los procesos. Esto incluye datos sobre costes asociados, planificación de su utilización y otros impactos derivados de su uso. Aunque ciertos recursos, como la maquinaria pesada, pueden no tener una representación geométrica detallada en el modelo, esta clase permite incorporar información esencial para la gestión económica y temporal del proyecto, facilitando la estimación de costes y la programación de las actividades.
- *IfcProduct* (Productos): este concepto constituye una categoría fundamental dentro del esquema IFC, definiéndose como la clase raíz para la definición de toda entidad discernible en un proyecto de construcción digital. Su alcance abarca diferentes componentes esenciales, que no solo representan elementos constructivos tangibles, sino que también engloba abstracciones espaciales, elementos destinados al análisis estructural, elementos conceptuales como rejillas y anotaciones gráficas, entre otros. Además, en su función como clase base, esta definición sirve como un contenedor genérico para incluir características básicas como atribuciones de materiales, representaciones geométricas explícitas y una ubicación precisa dentro del sistema de

coordenadas espaciales del proyecto. Esta capacidad de vinculación con la información geométrica y posicional es crucial para la integración y coherencia del modelo digital del edificio.

Gracias a esta organización inherente de la clase *IfcProduct*, se establece una estructura que facilita la comprensión y la gestión de la información del modelo, ya que permite vincular las diferentes entidades con sus correspondientes atributos y relaciones, contribuyendo a la integridad semántica del modelo IFC y facilitando la interoperabilidad entre diversas herramientas de software.¹⁵⁶

Organización espacial

Dentro de las definiciones de *IfcProduct*, se encuentra la estructura espacial, conocida como *IfcSpatialStructureElement*. Esta estructura se caracteriza por contener siempre una entidad inicial denominada *IfcProject*, de la que cuelga una o varias entidades de la clase *IfcSite*. Debajo de estas, se anexan en el siguiente nivel uno o varios edificios e infraestructuras como *IfcBuilding*, *IfcRoad*, *IfcRailway* o *IfcMarineFacility*. En el caso de proyectos de edificación, cada edificio tiene una serie de niveles conocidos como *IfcBuildingStorey*, que dividen la construcción verticalmente por pisos. Estas entidades IFC contienen definiciones generales que aportan información acerca de datos generales del activo de construcción como el nombre del proyecto, el autor, las coordenadas, las alturas de cada planta del edificio, etc.

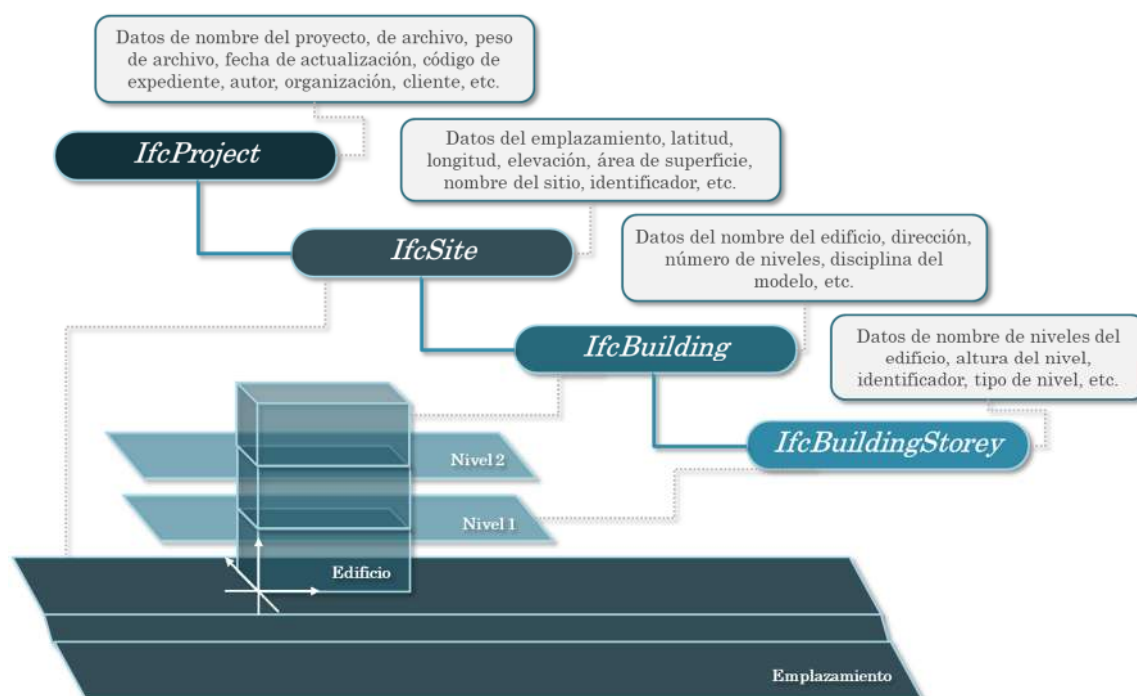


Figura 3.1.7. Estructura espacial IFC. Fuente* (ISO 16739)

Además de las entidades anteriores, la estructura espacial también define que cada nivel puede contener espacios, los cuales simbolizan áreas o volúmenes funcionales

¹⁵⁶ IFC Methodology Guide - Methodology for the Development of Industry Foundation Classes. Technical Report. International Alliance for Interoperability (IAI).

con información espacial del edificio y se definen dentro del esquema como *IfcSpace*.¹⁵⁷ Estos espacios pueden también ser agrupados en zonas para cumplir con diferentes objetivos relacionados con cálculos energéticos o agrupaciones de habitaciones dentro de cada vivienda del edificio.

Elementos de construcción

La entidad *IfcElement* representa los componentes individuales de una construcción, tales como muros, vigas y otros elementos físicos. Estos objetos poseen propiedades físicas inherentes que definen su materialidad y estructura dentro del activo construido. Estas definiciones permiten la representación tridimensional de los elementos y garantizan la interoperabilidad entre diversas herramientas BIM. Dentro de esta entidad, se encuentra *IfcBuiltElement*, una subentidad que agrupa los elementos funcionales de la construcción, como cimientos, pilares, forjados, muros, ventanas, puertas y cubiertas. Estos elementos se caracterizan por quedar incorporados de manera permanente en la obra y son comunes a todas las disciplinas del sector AECO.

Dentro del modelo IFC, algunos de estos elementos desempeñan múltiples funciones dentro de un proyecto, formando parte de distintos sistemas de construcción. Estos elementos multifuncionales son esenciales para la interoperabilidad y la colaboración multidisciplinar, ya que afectan a diversas áreas del diseño y la construcción. Por ejemplo, un muro puede cumplir simultáneamente funciones estructurales, de separación de espacios y de soporte de instalaciones. Modificar su ubicación genera impactos en otras partes del modelo, ya que puede alterar la configuración de los elementos conectados, requerir un nuevo cálculo de cargas estructurales, modificar el comportamiento térmico de los espacios adyacentes o afectar el diseño de las instalaciones que lo atraviesan. Para evitar conflictos y optimizar la colaboración en entornos BIM, es fundamental documentar de manera precisa la función de cada elemento y sus interdependencias dentro del modelo. Debido a esta transversalidad, las definiciones *IfcBuiltElement* engloban elementos compartidos, conocidos como *IfcSharedBldgElements*.

A través de este enfoque jerárquico, entidades más específicas como *IfcWall* e *IfcSlab* heredan características de *IfcBuiltElement* y pueden definir atributos adicionales según sus requerimientos. Este mecanismo de herencia proporciona flexibilidad al modelo y asegura una estandarización en la representación de los elementos constructivos. Además, el esquema IFC utiliza enumeraciones de tipos predefinidos que ofrecen una clasificación detallada dentro de cada entidad. *IfcSlab*, por ejemplo, puede especificarse como FLOOR, ROOF o LANDING. En caso de que ninguna clasificación predefinida sea adecuada, se puede utilizar la opción USERDEFINED, permitiendo una personalización acorde a las necesidades del proyecto.

La correcta asignación de estas entidades y categorías es fundamental ya que, si un elemento no se clasifica correctamente, su interpretación en diferentes softwares puede ser errónea. Si por ejemplo un forjado se exporta incorrectamente como un *IfcBuildingElementProxy* en lugar de *IfcSlab*, perderá atributos esenciales como altura, espesor de capa y materialidad. Esto puede generar problemas en análisis estructurales, cálculos de eficiencia energética y coordinación de instalaciones.

¹⁵⁷ IfcSpatialStructureElement. IFC 4.3. buildingSmart International.

Además de los elementos estructurales y constructivos, IFC también contempla los elementos de mobiliario y de distribución dentro del modelo digital. *IfcFurnishingElement* abarca elementos como escritorios, sillas, estanterías y armarios, permitiendo su inclusión en el diseño de interiores y facilitando la gestión del espacio en proyectos de arquitectura. Por otro lado, los elementos de distribución, representados por la entidad *IfcDistributionElement*, agrupan todos los elementos que participan dentro de un sistema. Esto incluye elementos de servicios dentro de sistemas de calefacción, refrigeración, ventilación, fontanería, drenaje, instalaciones eléctricas, redes de comunicación y redes de sensores de monitoreo.

Además de los anteriores, existen otros tipos de entidades que amplían la aplicabilidad del esquema en diversos aspectos de la construcción. *IfcElementAssembly* permite la agrupación de múltiples elementos individuales en un conjunto estructurado, facilitando la representación de sistemas modulares o prefabricados dentro del modelo. Esta entidad es especialmente útil para gestionar ensamblajes complejos, como fachadas modulares, sistemas de soporte o unidades de instalaciones técnicas. Otros elementos se incluyen mediante entidades como *IfcGeographicElement*, que permite representar objetos relacionados con el entorno del proyecto, como elementos topográficos, vegetación o infraestructuras urbanas. Su integración en modelos mejora la contextualización de los proyectos y permite una mejor planificación del emplazamiento y su impacto en el entorno. Por su parte, *IfcTransportElement*, abarca elementos de comunicación o transporte como ascensores, escaleras mecánicas y otros sistemas de transporte. En cuanto a elementos que no cuentan con una representación física tangible dentro del modelo pero que cumplen funciones relevantes en el diseño y la coordinación, el esquema establece la definición *IfcVirtualElement* relativa a elementos virtuales. Estos elementos pueden incluir zonificaciones, espacios de seguridad o recorridos de circulación previstos en el modelo.

Atributos

Los atributos representan las propiedades comunes a todos los objetos que pertenecen a una clase específica dentro del modelo de información. Estos atributos son fundamentales para describir las características estáticas de un objeto. Además, muchos de estos atributos se asocian a relaciones que son objetos pertenecientes a otras clases del esquema IFC. A menudo, estas relaciones son opcionales, lo que introduce un grado de flexibilidad en la representación de las entidades dentro del modelo. Los atributos, como los valores de las propiedades de las clases, deben ser declarados y definidos adecuadamente en el momento de instanciar un objeto dentro de un archivo. Esta instancia se genera mediante la inserción de los valores de los atributos en la clase. El orden posicional en que estos atributos son definidos al instanciar la clase es importante, y aquellos atributos que representan objetos complejos no se definen como simples textos, sino que se identifican mediante su identificador del esquema Express en el lugar correspondiente.

Cuando se instancian objetos en un archivo IFC, los atributos definen el estado del objeto a través de los valores asignados. Estos valores pueden corresponder a diferentes tipos de datos simples, o bien pueden ser tipos de datos derivados, que proporcionan una descripción más precisa de las características del objeto.

Con respecto a la jerarquía del esquema, los atributos más básicos se encuentran definidos en la clase *IfcRoot*. Estos atributos se van sumando a otros de diferentes subtipos y van siendo heredados por las clases derivadas. Esta herencia de atributos es un principio fundamental en el diseño del modelo de datos, ya que asegura que las clases más especializadas hereden las propiedades comunes sin necesidad de redefinirlas.¹⁵⁸

En términos de implementación, los atributos en IFC se representan mediante tipos de datos simples como *IfcReal*, *IfcInteger* o *IfcString*, los cuales permiten describir la información de los objetos de una forma estructurada y uniforme. Además, algunos atributos en IFC se definen mediante tipos de datos enumerados, que ofrecen un rango de valores posibles para un atributo determinado. Este tipo de dato permite un control más riguroso sobre las opciones que un atributo puede tomar, restringiendo su valor a una lista predefinida. Este enfoque es útil cuando un atributo debe representar una categoría específica o un conjunto de valores limitados, como el tipo de material o el estado de una tarea en un proyecto de construcción.

Atributo	Descripción	Ejemplo
<i>GlobalId</i>	Identificador único global para el objeto dentro del modelo, garantizando su unicidad. ¹⁵⁹	2f3d9b56-e7c5-4b50-b6d7-8a1f34f1a593
<i>Name</i>	Nombre asignado al objeto, utilizado para identificarlo o clasificarlo dentro del modelo.	Ventana de Oficina
<i>Description</i>	Descripción textual que proporciona más detalles sobre el objeto o su función en el modelo.	Ventana de aluminio con doble acristalamiento
<i>ObjectType</i>	Tipo de objeto o categoría a la que pertenece el elemento.	Ventana
<i>Tag</i>	Etiqueta o código de identificación del objeto, a menudo utilizado para el seguimiento y la gestión.	A1-V-001
<i>OwnerHistory</i>	Información sobre el historial del propietario del objeto, incluyendo detalles sobre el autor y el momento de creación.	Usuario1
<i>LongName</i>	Nombre largo del objeto o entidad, proporcionando una descripción más detallada.	Ventana de oficina con marco de aluminio, doble acristalamiento, con apertura lateral
<i>PredefinedType</i>	Tipo predeterminado del objeto, describiendo un valor predefinido que refleja su categoría funcional.	LIGHTDOME
<i>ObjectPlacement</i>	Especifica cómo está ubicado el objeto, incluyendo posición y orientación.	Ubicación en el muro M1-2132, a 2 metros del suelo

Tabla 3.1.2. Atributos IFC. Fuente* (ISO 16739)

¹⁵⁸ IFC Attributes and Properties. Wiki.OSArch. Actualización 2023.

¹⁵⁹ IFC GUID. Background Information. IFC Implementation Guidance. buildingSmart International.

Posicionamiento y orientación

El posicionamiento espacial de los objetos se logra a través de un conjunto de clases y mecanismos que permiten la manipulación de coordenadas locales y globales, esenciales para la representación geométrica del modelo en su totalidad. A lo largo de este proceso, clases como *IfcLocalPlacement*, *IfcCartesianPoint* e *IfcDirection* definen y transforman la ubicación y orientación de los objetos.

El concepto de ubicación de objetos en el contexto de IFC involucra una estructura jerárquica de sistemas de coordenadas que, a su vez, permite el posicionamiento relativo de los objetos. Dentro de esta estructura, el atributo *ObjectPlacement* define la posición de cada objeto dentro del modelo. Este atributo puede estar vinculado a diferentes subclases de *IfcObjectPlacement*, como *IfcGridPlacement* o, de forma más común, *IfcLocalPlacement*. Cada una de estas subclases ofrecen un mecanismo distinto para posicionar los objetos, adaptándose a las necesidades específicas del modelo y su estructura.¹⁶⁰

IfcLocalPlacement, en particular, es fundamental para definir la ubicación de los objetos dentro de un sistema de coordenadas relativo. Esta clase se utiliza para establecer un origen de coordenadas local a partir del cual se especifica la posición de un objeto en relación con otros elementos del modelo. Dentro de la definición, se encuentra *PlacementRelTo*, que define el objeto de referencia respecto al cual se establecen las coordenadas locales y *RelativePlacement*, que representa el desplazamiento relativo del objeto dentro de este sistema de coordenadas, utilizando la clase *IfcAxis2Placement*. Esta última clase, a su vez, define un sistema de referencia tridimensional con base en tres parámetros fundamentales: el origen, que es un punto en el espacio, la dirección del eje Z y la dirección de los ejes X e Y. Estos mecanismos permiten que los objetos se ubiquen correctamente en relación con otros componentes dentro del modelo, manteniendo una coherencia espacial.

En cuanto a la orientación de los objetos, la clase *IfcDirection* define la dirección de un vector en el espacio tridimensional utilizando un conjunto de valores denominados *DirectionRatio*. Estos valores representan las proporciones relativas entre los ejes del sistema de coordenadas, permitiendo que el objeto se oriente de manera precisa dentro del espacio modelado. La utilización de *IfcDirection* en combinación con *IfcLocalPlacement* asegura que la orientación de los objetos se ajuste correctamente a las relaciones espaciales con otros elementos.

A medida que se avanza en el desarrollo de un modelo de información, es posible que se necesite trasladar los objetos desde su sistema de coordenadas local a un sistema de coordenadas global. Este proceso de transformación de coordenadas se realiza mediante el uso de matrices de transformación. Para obtener las coordenadas globales de un objeto a partir de sus coordenadas locales, se multiplica el vector de coordenadas locales por la matriz de transformación correspondiente, y posteriormente se suma el desplazamiento relativo de la referencia superior del sistema de coordenadas. Este proceso asegura la coherencia del modelo al integrar elementos que provienen de diferentes sistemas de referencia.

En el ámbito de la georreferenciación, la ubicación de los objetos se extiende más allá de las coordenadas locales y globales, vinculándose a sistemas de coordenadas

¹⁶⁰ *IfcLocalPlacement*. IFC 4.3. buildingSmart International.

geográficas o planimétricas. La georreferenciación implica asignar un conjunto de coordenadas de mapa a los objetos, lo cual requiere la definición de parámetros adicionales, como el marco de medición, la unidad de medida, y la proyección cartográfica o sistema de referencia espacial. Además, se necesita un datum o eslipsoide de referencia que sirva como base para la proyección de coordenadas. El estándar IFC, al integrar estos sistemas de coordenadas geográficos, permite que los modelos sean colocados con precisión dentro de un contexto espacial más amplio, facilitando la interoperabilidad con otros sistemas de información geográfica GIS y la correcta integración con el entorno real.

Aunque las entidades principales del esquema no han cambiado a lo largo de las versiones del IFC, en el caso de la versión 2x3, las capacidades de transformación de coordenadas entre sistemas locales y globales eran más limitadas. Además, la capacidad para manejar sistemas de coordenadas georreferenciados y realizar transformaciones complejas entre diferentes sistemas de referencia aún no estaba tan desarrollada como en las versiones posteriores. IFC 4 permitió una mayor flexibilidad y precisión en la representación de la localización de objetos, mejorando el soporte para la georreferenciación de modelos y la integración con sistemas de información. En la actualidad, IFC 4.3 ha permitido una especificación de las coordenadas geográficas y una integración con GIS aún más detallada, facilitando la definición de marcos de medición, unidades de medida, y proyecciones cartográficas.

Representación geométrica

La representación geométrica de los objetos IFC describe de manera detallada la forma, el volumen y la orientación de los objetos dentro del espacio físico del modelo. En este contexto, se emplean varias clases dentro del estándar para estructurar la geometría de los objetos como *IfcProductDefinitionShape*, que juega un papel clave al asociar las formas de los objetos a través de la clase *IfcShapeRepresentation*, la cual proporciona una representación completa de la geometría de un objeto al definir sus componentes visuales y espaciales dentro del modelo. A través de esta clase, los objetos se representan mediante diferentes categorías de formas geométricas, cada una con un propósito específico que responde a los requerimientos de representación y análisis dentro del modelo. Estas formas incluyen representaciones de volumen, masas, ejes y superficies, entre otras, y se clasifican según el tipo de información que se desea modelar o analizar. Por ejemplo, el Centro de Gravedad se utiliza para describir las masas de los objetos dentro del modelo permitiendo realizar análisis estructurales o dinámicos que requieren conocer cómo se distribuyen las cargas dentro de la edificación.¹⁶¹

Otra de las formas geométricas más importantes es el *Box*, que define un volumen mínimo alrededor de un objeto y se utiliza para simplificar la representación de geometrías más complejas, especialmente en etapas preliminares de diseño o cuando no se necesita un alto nivel de detalle en la forma. Este tipo de representación permite realizar simulaciones de colisiones o análisis de espacio sin la necesidad de computar formas geométricas complejas, lo que mejora la eficiencia en el modelado de grandes volúmenes de datos. En paralelo, el *Axis*, que representa un eje, resulta útil para definir trayectorias o ubicaciones de objetos estructurales tales como

¹⁶¹ *IfcProductDefinitionShape*. IFC 4.3. buildingSmart International.

columnas, vigas y otros elementos que requieren una referencia precisa para su alineación dentro del espacio del edificio.

Cuando se requiere una descripción más detallada y precisa de la forma tridimensional de los objetos, el *Body* ofrece una representación completa de la geometría de un objeto, generando una visión realista y completa de su volumen y características. Otra definición es *Brep*, basada en superficies cerradas, es otra forma geométrica comúnmente utilizada en IFC, especialmente en modelado avanzado, que permite representar objetos complejos a través de sus límites definidos por superficies. Este tipo de modelado es adecuado para describir objetos con geometrías irregulares, como fachadas, techos o elementos arquitectónicos complejos.

Una de las técnicas más utilizadas en la representación geométrica de elementos constructivos dentro de IFC es el uso de *SweptSolid*, que se refiere a la creación de sólidos mediante la extrusión o el barrido de un perfil a lo largo de una trayectoria específica. Este método se utiliza comúnmente para representar objetos alargados como muros, vigas, columnas y otros elementos lineales. El *IfcExtrudedAreaSolid* es uno de los ejemplos más comunes de esta categoría, donde la geometría se genera extruyendo un perfil 2D a lo largo de una dirección para crear una forma 3D.

Por otro lado, el *IfcFacetedBrep* ofrece una representación de objetos mediante superficies cerradas que definen su volumen. Este modelo es utilizado para representar sólidos a través de polígonos y superficies facetadas, lo que lo convierte en una herramienta útil para describir formas geométricas complejas que no pueden ser fácilmente representadas mediante extrusión. Esta clase se emplea con frecuencia en el modelado de detalles arquitectónicos o componentes que tienen geometrías no uniformes y requieren una mayor precisión en su descripción.

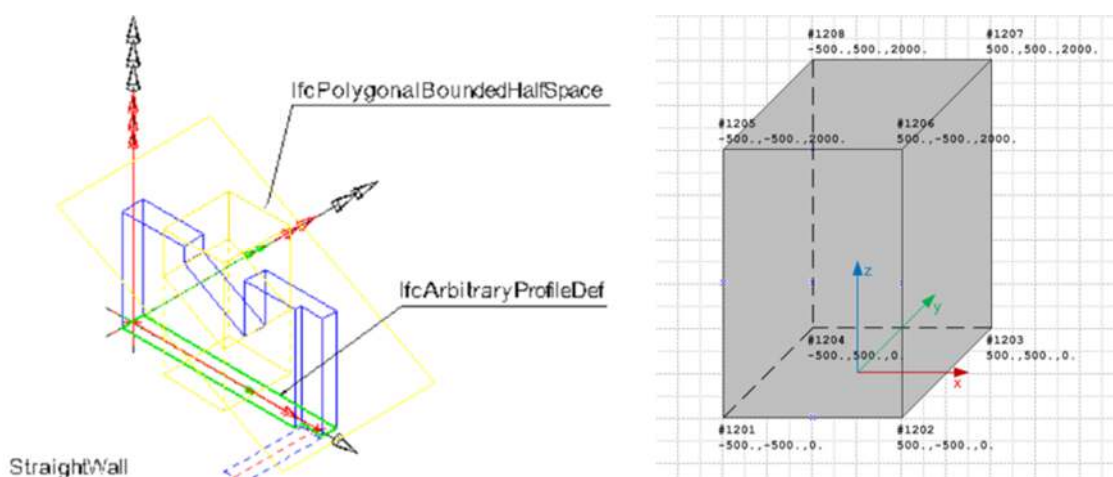


Figura 3.1.8. Ejemplos de SweptSolid y FacetedBrep del IFC¹⁶²

Estas formas geométricas definidas en el esquema IFC también están vinculadas a unidades de medida específicas, lo que permite establecer un control preciso sobre la escala y las dimensiones de los objetos dentro del modelo. En muchos casos, las unidades de medida son fundamentales para asegurar que los objetos sean representados correctamente dentro del espacio físico.

¹⁶² *SweptSolid* y *FacetedBrep*. ISO/CD 10303-42:1992 y buildingSmart International.

Materiales

Los materiales representan las materias primas o componentes básicos que constituyen los elementos de construcción en un modelo BIM. Estos abarcan una amplia gama de sustancias, desde elementos naturales hasta compuestos industriales. En IFC, los materiales se definen de acuerdo con su uso específico en la construcción, reflejando con precisión los distintos tipos de materiales utilizados en cada parte del proceso constructivo.¹⁶³

Aunque los materiales no son objetos en el sentido tradicional dentro del modelo IFC ya que no tienen un identificador GUID, pueden estar asociados a varias entidades constructivas mediante relaciones como *IfcRelAssociatesMaterial*. Su identificación dentro del modelo se realiza mediante su nombre y, a menudo, se complementa con propiedades adicionales definidas en conjuntos como *IfcMaterialProperties*, que permiten describir atributos como densidad, conductividad térmica, resistencia al fuego, entre otros. A nivel visual, los materiales pueden asociarse con propiedades gráficas básicas, como el color, la transparencia y la textura, pero para representaciones fotorrealistas aún es necesario complementar el modelo IFC con otras tecnologías como motores de renderizado o herramientas de realidad virtual.

En la versión IFC 2x3, la representación de materiales se limitaba a propiedades básicas como el nombre del material, su color, el orden de capas en materiales multicapa y el espesor de dichas capas. Sin embargo, a partir de IFC4, la representación de los materiales ha sido ampliada y mejor estructurada. La entidad base *IfcMaterial* sigue siendo utilizada para definir un material individual, pero se ha mejorado su capacidad para describir combinaciones de materiales y estructuras jerárquicas a través de entidades como *IfcMaterialLayerSet*. Además, en la versión 4.3, se han incorporado mejoras que permiten una definición más rica y precisa que incluye la orientación de capas, la posibilidad de combinación de materiales por función o ubicación, e incluso la asociación con perfiles estructurales o acabados superficiales.

Por último, aunque la representación de materiales en IFC aún puede resultar limitada para aplicaciones avanzadas de simulación como análisis energético detallado, acústico o visualización hiperrealista, su estructura ha sido diseñada para permitir extensibilidad. Esto permite añadir propiedades personalizadas o utilizar extensiones de esquema para adaptarse a requerimientos específicos de proyecto, software o dominio.

3.1.2.3. Relaciones

El concepto *IfcRelationship*¹⁶⁴ en el esquema IFC es otra de las definiciones clave para estructurar la información dentro de un modelo de información. Las relaciones establecidas entre los diferentes objetos del modelo permiten representar con precisión las conexiones funcionales, espaciales y lógicas, proporcionando una semántica estructurada y garantizando la interoperabilidad de los datos. A continuación, se detallan los tipos de relaciones del esquema con sus subtipos más relevantes.

¹⁶³ IfcMaterial. Wiki.OSArch. Actualización 2023.

¹⁶⁴ IfcRelationship. IFC 4.3. buildingSmart International.

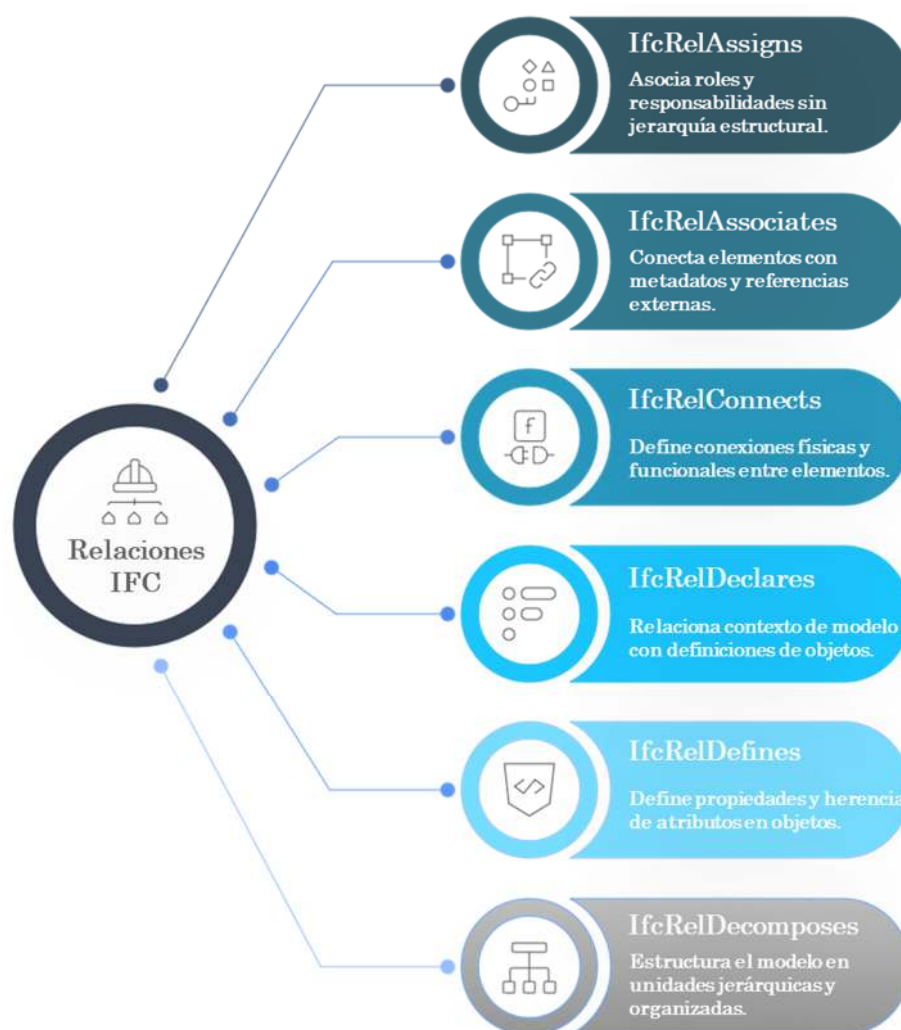


Figura 3.1.9. Relaciones del esquema IFC. Fuente* (ISO 16739)

- *IfcRelAssigns* (Relación de asignación): permite asociar un conjunto de objetos con uno o más elementos de referencia sin establecer una jerarquía o dependencia estructural. Esta relación es utilizada para asignar roles, responsabilidades o recursos a determinados elementos dentro del modelo. En el ámbito de la planificación y gestión de proyectos, *IfcRelAssigns* facilita la vinculación de tareas constructivas con equipos de trabajo, el establecimiento de relaciones entre actores y entidades de diseño, y la asociación de elementos a paquetes de trabajo específicos. Su flexibilidad permite modelar estructuras organizativas y operativas sin alterar la jerarquía intrínseca de los objetos involucrados.
- *IfcRelAssociates* (Relación de asociación a recursos externos): establece conexiones entre elementos del modelo y fuentes de información externas, tales como documentos, especificaciones técnicas, normativas y clasificaciones estandarizadas. Esta relación permite enriquecer el modelo con metadatos y referencias externas, proporcionando información adicional que contribuye a la toma de decisiones y al cumplimiento de regulaciones.
- *IfcRelConnects* (Relación de conexión): establece las conexiones físicas, lógicas o funcionales entre los elementos del modelo, asegurando la

coherencia e integración de sistemas estructurales, instalaciones y equipos electromecánicos. Estas relaciones son esenciales para el análisis de desempeño, simulaciones de carga y compatibilidad entre disciplinas. Dentro de esta definición, destacan *IfcRelContainedInSpatialStructure*, la cual define la ubicación espacial de los elementos dentro de la estructura del edificio, organizándolos en niveles como pisos o zonas específicas y *IfcRelConnectsElements*, que define la relación entre distintos elementos constructivos, como la unión entre muros y forjados. Otra relación perteneciente a esta clase es *IfcRelFillsElement*, que define como un elemento de relleno, como una ventana, se inserta dentro de un hueco en un muro.

- *IfcRelDeclares* (Relación de contexto): establece una relación entre un contexto de modelo, como un proyecto o una biblioteca de objetos, y las definiciones de objetos dentro de este. Esta relación estructura el modelo definiendo qué elementos forman parte de un proyecto y cómo se organizan dentro de un sistema de información centralizado. Además, facilita la gestión de bibliotecas de objetos reutilizables y la aplicación de propiedades contextuales a múltiples elementos dentro del modelo, asegurando la coherencia en el uso de componentes BIM.
- *IfcRelDecomposes* (Relación de jerarquía y descomposición): representa las relaciones de jerarquía y descomposición dentro de un modelo BIM, permitiendo dividir elementos complejos en unidades más pequeñas y organizadas de manera lógica. Esta relación es fundamental para estructurar el modelo en distintos niveles, como la segmentación de un edificio en niveles y zonas, la descomposición de un sistema estructural en sus componentes individuales o la subdivisión de ensamblajes en subcomponentes. Además, facilita la gestión eficiente de la información a lo largo del ciclo de vida del activo, permitiendo diferentes niveles de detalle según la fase del proyecto. Dentro de este tipo de relación, existen tres subtipos principales que establecen diferentes formas de relación entre los elementos del modelo. *IfcRelAggregates* define una relación jerárquica de todo-parte de manera no ordenada y no espacial, con una conexión bidireccional implícita. Se utiliza para organizar la estructura del modelo, como la relación entre un emplazamiento y los edificios dentro de este o la subdivisión de un edificio en pisos o niveles. Por otro lado, *IfcRelNests* establece una relación de todo-parte similar a la agregación, pero con una diferencia clave, es ordenada y no espacial, lo que implica que los elementos anidados tienen una secuencia específica dentro de su agrupación. Se emplea en casos donde el orden de los elementos es relevante, como la descomposición de un proceso de construcción en una secuencia de tareas o en la estructura interna de un componente mecánico con sus subcomponentes organizados de manera jerárquica. Por último, *IfcRelVoidsElement* representa la relación entre un elemento constructivo y un vacío que forma parte de él, permitiendo modelar huecos dentro de muros, losas o vigas. Al vincular un elemento con su vacío, se mantiene la coherencia geométrica y funcional del modelo, asegurando que los espacios reservados para instalaciones o accesos sean correctamente gestionados dentro del entorno BIM.
- *IfcRelDefines* (Relación de herencia): agrupa relaciones que permiten asignar propiedades, características y tipos comunes a varios elementos de un modelo

BIM, evitando redundancia y facilitando la actualización de información.¹⁶⁵ Dentro de esta clase de relación existen tres subtipos principales: *IfcRelDefinesByObject* permite que varios elementos hereden características de un objeto de referencia, asegurando consistencia en elementos repetidos como puertas o ventanas. *IfcRelDefinesByProperties* asigna propiedades a uno o más elementos sin modificar su geometría o tipo, permitiendo agregar atributos personalizados como datos de resistencia térmica o especificaciones de mantenimiento. *IfcRelDefinesByType* vincula elementos con un tipo predefinido, garantizando que compartan características comunes y permitiendo actualizaciones centralizadas en todas sus instancias.

En conjunto, la diversidad y la especificidad de los subtipos de *IfcRelationship* confieren al esquema IFC una potente capacidad para modelar las complejas interdependencias que caracterizan los proyectos de construcción. Estas relaciones no solo estructuran la información geométrica, sino que también enlazan los aspectos funcionales, temporales y de gestión, consolidando el modelo como un repositorio de información coherente y exhaustivo para la totalidad de la edificación. En la siguiente imagen se muestra un ejemplo de posibles relaciones para una instancia de un equipo mecánico.

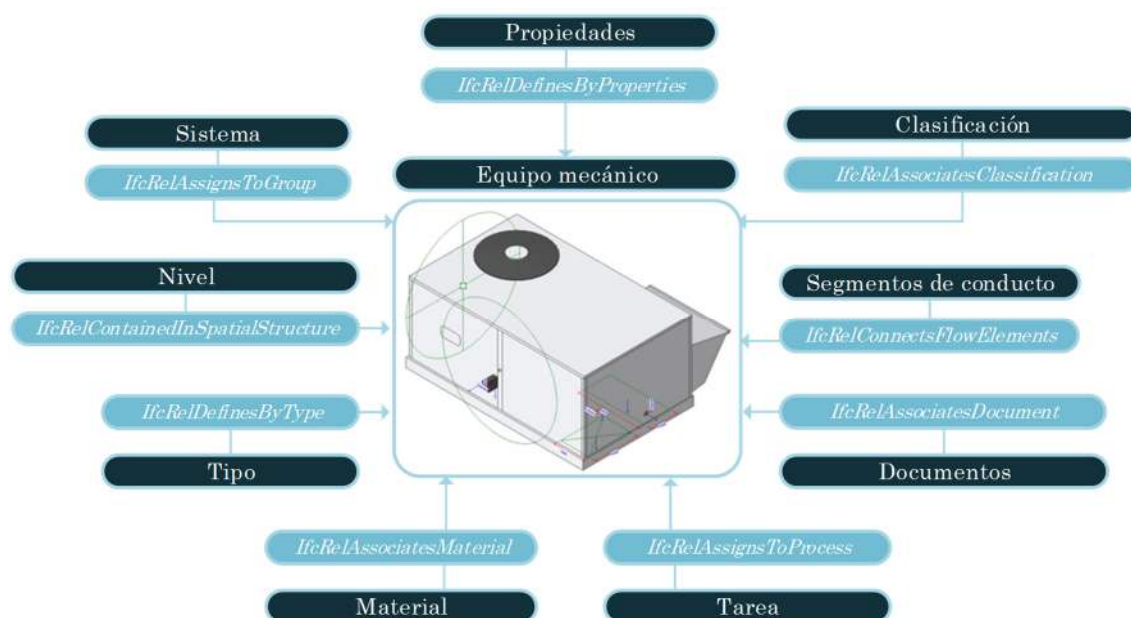


Figura 3.1.10. Relaciones de un equipo mecánico dentro del IFC

Otro aspecto fundamental de las relaciones del IFC es la cardinalidad, que se refiere a la cantidad de instancias que pueden existir en una relación entre diferentes objetos dentro del modelo. Esta propiedad no se aplica a los atributos de una clase, sino a las relaciones que definen cómo los objetos se conectan o interactúan entre sí. Cada relación dentro del esquema puede tener una cardinalidad definida que especifica cuántas instancias de una clase pueden estar asociadas con una instancia de otra clase. Estas relaciones se determinan según las necesidades del proyecto y la naturaleza de los elementos involucrados. Por ejemplo, una relación entre un edificio y sus plantas puede tener una cardinalidad de uno a muchos, donde un edificio puede

¹⁶⁵ IFC File. ACCA Software. Whitepaper. 2021.

contener múltiples plantas, pero cada planta pertenece a un solo edificio. Este concepto es crucial para representar las interacciones y dependencias entre los elementos IFC, garantizando que se pueda interpretar correctamente la estructura de un proyecto de construcción, tanto en términos de sus componentes individuales como de las relaciones entre ellos. Esta cardinalidad por tanto establece restricciones y permite definir la cantidad mínima o máxima de instancias involucradas en una relación, lo que ofrece una mayor precisión y flexibilidad a la hora de modelar los diferentes aspectos de un proyecto BIM.

3.1.2.4. Definiciones de propiedades

La clase *IfcPropertyDefinition* actúa como marco conceptual para la estructuración y la organización de la información descriptiva asociada a las entidades del modelo. Su función primordial radica en establecer la semántica de las propiedades, permitiendo la definición abstracta de conjuntos de atributos que posteriormente pueden ser instanciados y asignados a múltiples objetos dentro del modelo de información de la construcción.¹⁶⁶

En su jerarquía, *IfcPropertySetDefinition* se presenta como el esqueleto para la creación de agrupaciones lógicas de propiedades relacionadas. Estos conjuntos de propiedades representan colecciones nombradas de atributos que describen características específicas de los objetos. La vinculación entre estos objetos y sus respectivos conjuntos de propiedades se materializa mediante la relación *IfcRelDefinesByProperties*, estableciendo una conexión explícita entre la entidad y su información descriptiva. Además de esta relación, la clase *IfcProperty* define la naturaleza y el tipo de cada propiedad individual que compone un conjunto de propiedades. Actúa como un contenedor abstracto para los diversos tipos de información que pueden ser relevantes para describir un objeto o un recurso de construcción.

Dentro del esquema, estos conjuntos de propiedades se denominan *PropertySets* y *QuantitySets*, y se definen con nombres específicos que suelen comenzar con los prefijos *Pset* o *Qto* respectivamente. Los *Psets* se utilizan para agrupar propiedades adicionales que no están directamente incluidas en la definición básica de una entidad, mientras que los *Qtos* asocian información de cantidades. Ambas definiciones proporcionan información detallada sobre los elementos del modelo, permitiendo una mayor flexibilidad y precisión en la representación de los objetos dentro del modelo BIM.

En cuanto a la estandarización de estos conjuntos, el esquema propone la utilización de grupos predefinidos, aunque su uso no es obligatorio. Además, pueden identificarse otros tipos de grupos según el origen de la información, como grupos definidos por el promotor del encargo, grupos generados por defecto por los programas BIM o grupos generados por el usuario, creados para atender necesidades específicas o para mejorar la eficiencia durante el proceso de desarrollo.

Propiedades

La especialización de *IfcProperty* se manifiesta a través de sus subtipos. *IfcComplexProperty* permite la definición de propiedades con una estructura interna

¹⁶⁶ The EXPRESS Definition Language for IFC Development. buildingSmart International.

más elaborada, capaces de albergar información compleja y anidada. En contraste, *IfcSimpleProperty* se centra en la definición de propiedades con valores directos y unívocos, con subtipos como *IfcPropertySingleValue*, *IfcPropertyEnumeratedValue*, *IfcPropertyBoundedValue*, *IfcPropertyTableValue*, *IfcPropertyReferenceValue* o *IfcPropertyListValue*, los cuales especifican las diferentes maneras en que estos valores pueden ser representados y gestionados. Desde valores únicos y discretos hasta listas ordenadas y referencias a otras propiedades, estos subtipos proporcionan la flexibilidad necesaria para capturar la diversidad de información inherente a los elementos de un proyecto de edificación.

Estas propiedades permiten estandarizar la forma en que se define y se organiza la información descriptiva, facilitando la interoperabilidad semántica entre diferentes plataformas y software BIM. Además, la capacidad de definir propiedades de manera abstracta y luego instanciarlas para múltiples objetos optimiza la gestión de la información y promueve la creación de modelos ricos en datos.

Los subtipos de *IfcSimpleProperty* ofrecen diversos mecanismos para definir y representar la información asociada a objetos individuales dentro del esquema IFC. Cada uno de estos subtipos está diseñado para capturar diferentes naturalezas de datos, permitiendo una descripción detallada y precisa de los elementos de un modelo BIM.

- *IfcPropertySingleValue* se utiliza para definir una propiedad que posee un único valor, el cual puede ser de tipo numérico o descriptivo. Esta definición incluye atributos como el Nombre de la propiedad, el Valor Nominal asignado, el Tipo de valor, y la Unidad de medida si corresponde. Externamente, se puede entender como la forma más directa de asignar un atributo a un objeto, representando una característica unívoca y bien definida.
- *IfcPropertyEnumeratedValue* define una propiedad cuyo valor debe seleccionarse de un conjunto predefinido de valores posibles, conocido como enumeración. Este subtipo se caracteriza por su Nombre, el Valor seleccionado, el Tipo de valor, y la referencia a la propia enumeración que contiene los valores permitidos. El mayor beneficio de este tipo de propiedad es que consigue limitar las posibles entradas a un conjunto controlado de opciones.
- *IfcPropertyBoundedValue* se emplea para definir una propiedad que se especifica mediante un rango de valores, definido por un límite inferior y un límite superior. Los atributos de este subtipo son el Nombre, el *UpperBoundValue*, el *LowerBoundValue*, el Tipo de valor y la Unidad. Su aplicación es útil para indicar tolerancias dimensionales, rangos de temperatura operativa o límites de peso admisibles para un componente. Su uso es de gran utilidad cuando se necesita verificar requisitos o normativas, ya que asegura que las propiedades de un objeto se encuentren dentro de los límites aceptables.
- *IfcPropertyTableValue* define una propiedad a través de una tabla bidimensional, donde la primera columna establece un conjunto de valores definitorios y la segunda columna contiene los valores de la propiedad correspondientes. Los atributos incluyen *DefiningValues*, *DefiningValueType*, y *DefiningUnit*. Un caso de uso típico es la

representación de curvas de rendimiento de un equipo en función de diferentes condiciones operativas.

- *IfcPropertyReferenceValue* permite asignar un valor a una propiedad mediante una referencia a otra propiedad compleja predefinida dentro del esquema IFC. Este mecanismo facilita la reutilización de información ya estructurada y la creación de relaciones. Este tipo de propiedad promueve la coherencia del modelo al evitar la redundancia de información y al establecer vínculos explícitos entre diferentes aspectos de un objeto.
- *IfcPropertyListValue* se utiliza para definir una propiedad que consiste en una lista ordenada de uno o más valores únicos. Sus atributos son el Nombre, *ListValue*, el Tipo de los valores en la lista y la Unidad si aplica. Un ejemplo de su aplicación es en las capas que componen un elemento constructivo multicapa, donde el orden de las capas es relevante. Gracias a este tipo de propiedad es posible representar conjuntos de características o componentes donde el orden o la multiplicidad son importantes.

A pesar de la flexibilidad del esquema IFC, muchos programas actuales limitan el trabajo solo a valores del tipo *IfcPropertySingleValue*, lo que restringe la capacidad de gestionar información más compleja dentro del modelo. En cuanto a los tipos de datos, el esquema IFC admite varios formatos los cuales han sido definidos mayormente en el Anexo B del presente documento.

Cantidades

En el contexto del esquema IFC, el concepto de cantidad o quantity describe aspectos físicos medibles de los objetos constructivos. A diferencia de las propiedades, que representan características más generales o cualitativas, las cantidades están intrínsecamente asociadas a magnitudes físicas expresables en unidades del Sistema Internacional u otras unidades de medición. El mecanismo central para manejar este tipo de datos en IFC se organiza a través de la clase *IfcQuantitySet* y sus subtipos especializados. Esta clase abstracta no se utiliza directamente para instanciar objetos, pero sirve como supertipo común para la estructura *IfcElementQuantity*, la cual actúa como un contenedor estructurado de cantidades asignadas a un objeto específico del modelo. Estas cantidades pueden representar el volumen de un pilar de hormigón, la longitud de una tubería, o la superficie de una cubierta inclinada. A diferencia de un *IfcPropertySet*, que puede contener tanto propiedades simples como complejas sin asociarse necesariamente a unidades, *IfcElementQuantity* tiene una finalidad exclusivamente cuantitativa, lo que la hace crucial en tareas como el cómputo de materiales, la estimación de costos o la planificación de obra.¹⁶⁷ Con el objetivo de analizar las propiedades de cantidades de las diferentes versiones del IFC, se ha generado una tabla detallada en el anexo D del presente estudio.

Las cantidades contenidas en un *IfcElementQuantity* no se representan como valores sueltos, sino como instancias de subtipos de la clase *IfcPhysicalQuantity*. Dentro de dicha entidad, *IfcPhysicalSimpleQuantity* representa la unidad básica e indivisible de información cuantitativa, definiendo el tipo y la semántica de la cantidad. Su función es registrar un valor aislado, aunque semánticamente significativo, que se define por alguno de los siguientes conceptos:

¹⁶⁷ *IfcElementQuantity*. IFC 4.3. buildingSmart International.

Subtipo	Unidad	Tipo de dato	Descripción técnica
<i>IfcQuantityLength</i>	Longitud (m, mm, ft)	<i>IfcLengthMeasure</i>	Longitud lineal, como altura, perímetro, profundidad o espesor.
<i>IfcQuantityArea</i>	Superficie (m ² , ft ²)	<i>IfcAreaMeasure</i>	Área de un componente, como superficie neta o bruta de muros, losas, techos.
<i>IfcQuantityVolume</i>	Volumen (m ³ , ft ³)	<i>IfcVolumeMeasure</i>	Volumen físico del objeto, útil para cálculos de materiales o excavación.
<i>IfcQuantityCount</i>	Cantidad entera	<i>IfcCountMeasure</i>	Unidades discretas: número de tornillos, luminarias, placas, etc.
<i>IfcQuantityWeight</i>	Masa (kg, t, lb)	<i>IfcMassMeasure</i>	Peso o masa del componente o de sus partes.
<i>IfcQuantityTime</i>	Tiempo (s, min, h)	<i>IfcTimeMeasure</i>	Tiempo estimado de fabricación, montaje, uso o mantenimiento.
<i>IfcQuantityNumber</i>	Porcentaje, coeficiente, etc.	<i>IfcNumericMeasure</i>	A partir de IFC 4.3. Registra valores numéricos genéricos que no encajan en las otras categorías.

Tabla 3.1.3. Tipos de cantidades IFC. . Fuente* (ISO 16739)

Cada una de estas cantidades incluye atributos clave como *Name*, *Description*, *Unit*, y *Value*. La unidad puede ser definida de forma local en la cantidad o heredada del contexto del modelo, lo que permite flexibilidad en la representación. Además, un elemento puede tener varias cantidades del mismo tipo, pero con propósitos distintos: por ejemplo, un muro puede tener un volumen neto y un volumen bruto, representados ambos como instancias separadas de *IfcQuantityVolume* diferenciadas por su nombre, en este caso *NetVolume* y *GrossVolume*.

Desde el punto de vista semántico, las cantidades tienen una mayor precisión y relevancia operativa que las propiedades genéricas, ya que son utilizadas directamente por aplicaciones de cálculo. En este sentido, la mayoría de los programas BIM generan automáticamente los *IfcElementQuantity* a partir de la geometría del modelo, aunque también es posible definirlos manualmente para reflejar ajustes de diseño, tolerancias o métodos constructivos específicos. Cabe destacar que estas entidades deben responder a criterios objetivos de medición. Esto implica una responsabilidad en cuanto a la definición exacta de qué se está midiendo y con qué criterios y también al uso de nombres de cantidades estandarizados como *Length* o *Height* para facilitar el intercambio entre sistemas.

Dentro del mismo marco, el estándar IFC contempla situaciones en las que es necesario agrupar múltiples cantidades relacionadas semánticamente, bajo una estructura compuesta. Para estos casos, se utiliza el subtipo *IfcPhysicalComplexQuantity*. Esta entidad permite encapsular un conjunto coherente de cantidades simples bajo un nombre común, ofreciendo así una representación jerárquica y modular de información cuantitativa. Esta definición puede incluir metadatos adicionales como una unidad común, una descripción y una lista de calificaciones, que permiten distinguir entre diferentes variantes o enfoques.

3.1.3. Manipulación de archivos IFC

La manipulación de archivos IFC va más allá de la visualización gráfica, abriendo posibilidades de análisis profundos de los datos estructurales que contienen. A pesar de que los visores gráficos permiten la representación tridimensional de los modelos, existen aplicaciones especializadas que ofrecen una visión alfanumérica de los datos dentro del archivo, permitiendo al usuario explorar la información de manera detallada sin la necesidad de representar la geometría. Estas herramientas son necesarias para evaluar las propiedades contenidas en el archivo, como datos de clases, propiedades e interacciones. Diferentes aplicaciones desarrolladas en los últimos años facilitan estos trabajos con archivos IFC, permitiendo generar resúmenes detallados de la estructura de los archivos, exportar información y verificar la conformidad del archivo con respecto a las versiones del IFC 2x3, 4 y 4.3. Estos procesos son cruciales para asegurar la integridad y coherencia de los datos, especialmente en proyectos grandes y complejos donde la gestión eficiente de la información BIM es esencial y también para la extracción y análisis de datos que los software y plataformas del mercado actual no son capaces de procesar.

3.1.3.1. Estructura del archivo

Al abrir un archivo IFC desde un editor de texto, se encuentran dos partes diferenciadas, el encabezado y el cuerpo. El primero contiene información general sobre el modelo de construcción, la versión de IFC, la fecha de creación, el esquema y la *MVD*, mientras que el segundo contiene información sobre los objetos, la geometría y los atributos de la propia construcción.¹⁶⁸

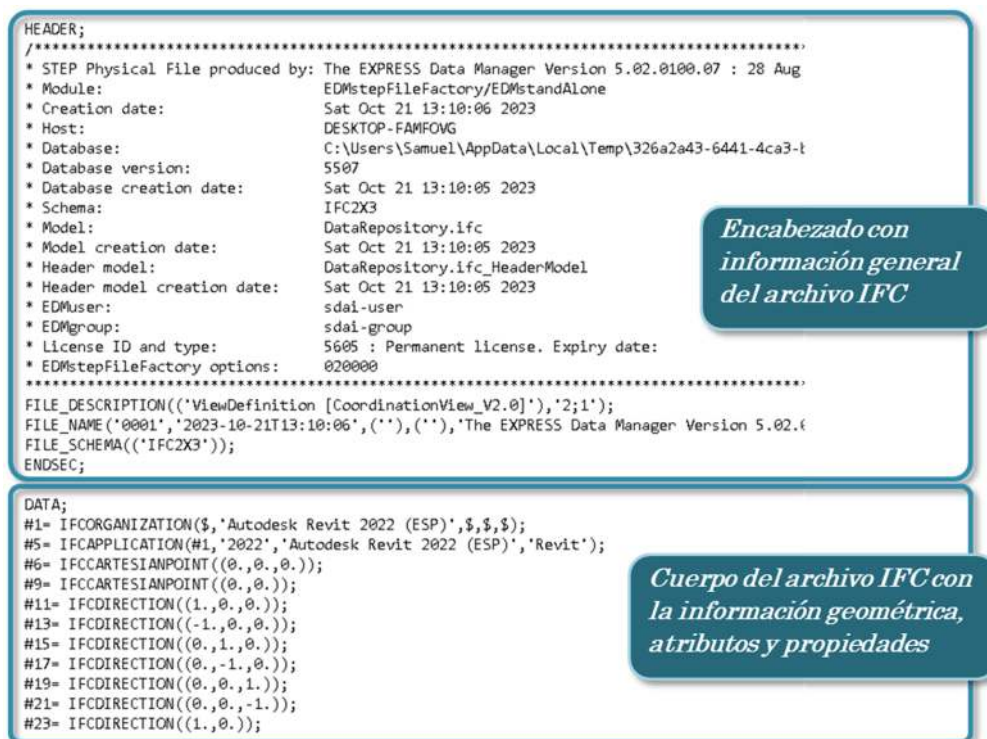


Figura 3.1.11. Encabezado y cuerpo del archivo IFC

¹⁶⁸ Implementation Guide for IFC Header Section. V01. buildingSmart International. 2008.

Dentro del cuerpo del esquema, la definición de cada elemento comienza con una línea que clasifica el objeto, lo identifica de manera única y le da nombre.¹⁶⁹ Un ejemplo puede ser la definición del emplazamiento *IfcSite* del proyecto:

```
#34642=IFCSITE('3Ilj7B0LnBjf0mv',#41,'Sup:411452',#34641,#34634);
```

Además, en la propia línea de definición, el objeto hace referencia a otras líneas en la estructura del archivo, que se pueden identificar mediante el símbolo de almohadilla (#) que las precede. Estas líneas lo describen con más detalle y, a su vez, hacen referencia a otras líneas con diferentes tipos de información relacionada:

```
#34642= IFCSITE('3Ilj7B0LnBjf0mv',#41,'Sup:411452',#34641,#34634);
  ↘ [ #41= IFCOWNERHISTORY(#38,#5,NOCHANGE,,$,$,$,1518553926);]
    ↘ [ #38= IFCPERSONANDORGANIZATION(.....,$);]
    ↘ [ #5= IFCAPPLICATION(...,'2019','AutodeskRevit (ESP)');]
  ↘ [ #34641= IFCLOCALPLACEMENT($,#34640);]
    ↘ [ #34640= IFCAXIS2PLACEMENT3D(.....);]
  ↘ [ #34634= IFCPRODUCTDEFINITIONSHAPE($,$,#34629,#34631);]
    ↘ [ #34629= IFCSHAPE REPRESENTATION(...,'Body','Surface'(...));]
    ↘ [ #34631= IFCSHAPE REPRESENTATION(...,'FootPrint'(...));]
```

Esta estructura de referencias continúa hasta la generación total de un modelo lógico de datos que ofrece una descripción detallada de cada objeto. Gracias a este sistema, los atributos específicos se almacenan una sola vez y pueden ser utilizados por otros componentes por medio de referencias. Esto reduce considerablemente el tamaño del archivo y significa, por ejemplo, que los componentes con el mismo tipo harán referencia a las mismas líneas de definición del tipo dentro del modelo de datos.

3.1.3.2. Librerías

Una librería es un componente de software que reúne un conjunto estructurado de funciones, clases y métodos diseñados para resolver tareas específicas o proporcionar funcionalidades concretas dentro de un entorno de programación. En lugar de desarrollar desde cero la lógica necesaria para ciertas operaciones, los desarrolladores pueden aprovechar estas bibliotecas ya construidas, que encapsulan algoritmos, estructuras de datos y flujos de procesamiento optimizados, facilitando el desarrollo, mantenimiento y escalabilidad del código. Estas librerías actúan como una capa intermedia entre el lenguaje de programación y los datos o sistemas que se desean manipular, y son fundamentales en la construcción de aplicaciones modulares y eficientes.¹⁷⁰ Dentro del ecosistema BIM, estas librerías cumplen un rol esencial al ofrecer una interfaz estructurada para interactuar con los complejos esquemas de datos que definen los modelos digitales de edificación.

En este contexto, las librerías enfocadas en la manipulación de archivos IFC proporcionan mecanismos para leer, escribir, analizar y modificar modelos construidos bajo el estándar a partir de la programación. Estas librerías se generan

¹⁶⁹ IFC - Industry Foundation Classes. Wiki.OSArch. Actualización 2023.

¹⁷⁰ The Python Standard Library. Python 3.13.2 documentation.

debido a que la manipulación de la información del IFC directamente sin herramientas especializadas resulta inviable debido a la complejidad del esquema. Por ello, estos recursos ofrecen un acceso simplificado pero potente a la estructura semántica y geométrica de los archivos IFC, lo que permite integrar esta información en aplicaciones personalizadas, automatizar procesos como la detección de errores, realizar transformaciones masivas sobre entidades o adaptar modelos a requerimientos normativos específicos.

El uso de lenguajes como *Python*, *C++*, *C#* o *JavaScript* en el desarrollo de estas librerías responde a la diversidad de entornos de trabajo en los que se emplean modelos IFC. Mientras que *Python* es ampliamente utilizado en procesos de análisis, automatización y ciencia de datos, *C++* se orienta a tareas de alto rendimiento como la visualización y el procesamiento geométrico, y *C#* se integra naturalmente en plataformas basadas en *.NET* utilizadas en entornos corporativos. *JavaScript*, por su parte, permite trasladar el procesamiento BIM al entorno web, haciendo posible la visualización e interacción con modelos IFC directamente desde el navegador. En todos los casos, las librerías permiten explorar estructuras jerárquicas del modelo, acceder a atributos de entidades, crear o eliminar componentes, gestionar relaciones entre objetos, e incluso aplicar reglas automatizadas.

Estas funcionalidades optimizan tareas de edición y validación a la vez que hacen posible nuevas formas de integración entre modelos BIM y tecnologías emergentes como la inteligencia artificial, el aprendizaje automático, la visualización en realidad aumentada, el análisis energético o la planificación automatizada de obra. De este modo, las librerías IFC se constituyen como piezas clave en la evolución del BIM hacia flujos de trabajo más abiertos, programables e interconectados.

IfcOpenShell

Entre las librerías más consolidadas del ecosistema de desarrollo BIM, destaca IfcOpenShell,¹⁷¹ una librería de código abierto diseñada para trabajar con datos IFC. Su desarrollo comenzó como un proyecto académico con el propósito de ofrecer una alternativa gratuita y de alto rendimiento a las herramientas comerciales para la lectura y escritura de archivos IFC, y con el tiempo ha evolucionado hasta convertirse en un componente central en numerosas aplicaciones BIM, tanto en entornos de escritorio como en flujos de trabajo automatizados en servidores.

A nivel técnico, IfcOpenShell implementa el estándar IFC en sus versiones 2x3, 4 y 4.3, garantizando compatibilidad con las exigencias actuales de interoperabilidad. Su núcleo está programado en *C++*, lo que le permite un procesamiento rápido y eficiente de estructuras complejas, mientras que su interfaz en *Python*, ampliamente utilizada por la comunidad, facilita la creación de scripts personalizados para automatizar tareas. Una de las características más destacadas en este sentido es su capacidad para generar geometría 3D a partir de descripciones geométricas contenidas en los archivos IFC, lo cual es clave para la visualización, la verificación de modelos y la conversión a otros formatos (Pan et al., 2023; Silva et al., 2024). Otra característica importante es la capacidad para convertir archivos IFC a otros formatos tridimensionales como *OBJ*, *DAE*, *STL*, *GLTF* o *STEP*, lo que facilita su integración con motores de juego, motores de renderizado o software de simulación.

¹⁷¹ IfcOpenShell 0.8.1 documentation. The open source IFC toolkit and geometry engine.

Además, la librería ofrece soporte para el análisis de propiedades no geométricas, como atributos alfanuméricos, relaciones entre entidades o estructuras jerárquicas del modelo (Omotunde & Ahmed, 2023; Serra, 2021; Sobhkhiz, Soroush and Zhou Yu-Cheng and Lin Jia-Rui and El-Diraby Tamer, 2021; Sweta, 2015; Zhang & Ye, 2011). Esto es particularmente relevante para flujos de trabajo orientados a la gestión de obra, operaciones de mantenimiento, auditorías de calidad o integración con bases de datos externas. Todos estos flujos se organizan y desarrollan en diferentes módulos contenidos dentro de la librería IfcOpenShell:

Módulo	Descripción	Uso BIM
BCF	Gestión de incidencias mediante el formato BCF, permite vincular observaciones al modelo.	Coordinación y revisión de modelos
BIMServer-Plugin	Interfaz para conectar IfcOpenShell con servidores BIMServer, permitiendo el trabajo en red y control de versiones.	Integración y automatización
BIMTester	Herramienta para validar el cumplimiento de reglas en modelos IFC usando archivos YAML.	Validación y control de calidad
bSDD	Conector con el bSDD para enriquecer semánticamente los modelos IFC.	Vinculación semántica y normalización
ifc2CA	Conversión de modelos IFC a estructuras compatibles con análisis estructurales.	Ingeniería estructural
ifc4D	Añade y gestiona información temporal en modelos IFC para simular fases de constructivas.	4D. Planificación de obra
ifc5D	Permite asociar información de costes al modelo IFC, integrando presupuestos en el entorno BIM.	5D. Presupuestación y control económico
IfcCityJSON	Conversión entre IFC y CityJSON para conectar modelos BIM con datos geoespaciales urbanos.	Integración BIM-GIS
IfcClash	Herramienta en desarrollo para detectar interferencias geométricas en modelos IFC.	Coordinación técnica y revisión
IfcCSV	Exportación de datos IFC a formato CSV para análisis o integración con otros sistemas.	Interoperabilidad y extracción de datos
IfcDiff	Comparación entre versiones de archivos IFC para detectar cambios o actualizaciones.	Gestión de cambios y control de versiones
IfcFM	Adaptación del IFC para Facility Management, permitiendo su uso en operación y mantenimiento.	Gestión de activos e infraestructuras
IfcMax	Exportación de modelos IFC hacia entornos como 3ds Max para visualización y animación.	Visualización avanzada
IfcPatch	Aplicación de transformaciones automatizadas en modelos IFC para corrección o adaptación.	Corrección y limpieza de modelos
IfcSverchok	Integración de IFC con el sistema de nodos visuales de Sverchok para modelado generativo.	Modelado paramétrico y visual
IfcTester	Entorno de prueba para ejecutar reglas, scripts y evaluaciones sobre modelos IFC.	Desarrollo y testeo de scripts

Tabla 3.1.4. Módulos IfcOpenShell. Fuente (OSArch Wiki)

Su enfoque de código abierto ha promovido una comunidad activa que contribuye a su desarrollo continuo, reporta errores, mejora la documentación y propone nuevas funcionalidades. Gracias a su licencia permisiva,¹⁷² puede integrarse sin restricciones en aplicaciones comerciales y académicas, lo que ha impulsado su adopción en universidades, oficinas de arquitectura, empresas de ingeniería y desarrolladores de software BIM en todo el mundo. Su papel es cada vez más central en la transición hacia flujos de trabajo digitales abiertos, colaborativos y centrados en los datos.

xBIM Toolkit

xBIM Toolkit es otra librería de software abierta y gratuita diseñada para trabajar con el estándar IFC. Su principal propósito es proporcionar a los desarrolladores herramientas para gestionar modelos BIM en aplicaciones personalizadas. Esta librería está escrita en *C#* y está orientada principalmente a integrar funcionalidades BIM en sistemas propios, sin depender de soluciones comerciales. Una de sus características principales es la posibilidad de leer y escribir archivos IFC, permitiendo la interoperabilidad entre diferentes programas. La librería también proporciona una serie de componentes que permiten la manipulación y visualización de los datos del modelo, haciendo posible la integración en diversas aplicaciones, como visualizadores 3D, simuladores de construcción o herramientas de análisis de datos.¹⁷³

Otra característica es su capacidad para gestionar grandes volúmenes de datos típicos en los modelos BIM, utilizando estructuras de datos eficientes que optimizan el rendimiento. Además, la librería está diseñada para ser extensible, lo que permite crear y agregar nuevas funcionalidades de acuerdo con las necesidades específicas del proyecto. Esto incluye la capacidad de realizar cálculos relacionados con el análisis de energía, la planificación de proyectos y la secuencia constructiva. Además, la accesibilidad y la facilidad de uso proporcionan una API que facilita su integración en diversos entornos de desarrollo. A través de su interfaz de programación, es posible acceder a las entidades definidas en el modelo y manipular sus propiedades o relaciones con otros elementos.

Por otro lado, xBIM incluye herramientas para la visualización en 3D, lo que permite a los usuarios ver el modelo en un entorno gráfico interactivo. Esto mejora la comprensión del proyecto a la vez que facilita la identificación de problemas o inconsistencias dentro del modelo. En términos de compatibilidad, ofrece un alto grado de integración con otros sistemas y plataformas, lo que lo convierte en una opción potente para proyectos en los que se requieren múltiples herramientas o una integración fluida con otros servicios de software. Su comunidad activa y su continuo desarrollo garantizan que se mantenga actualizado con las últimas normativas y estándares de la industria.

¹⁷² IfcOpenShell está licenciado bajo los términos de la GNU Lesser General Public License, versión 3.0 (LGPL-3.0), lo que permite su uso, modificación e integración tanto en proyectos libres como comerciales, siempre que se mantenga la compatibilidad con los términos de dicha licencia.

¹⁷³ Xbim Essentials & Xbim Geometry. Xbim, the extensible Building Information Modelling toolkit for the .NET platform. 2022.

Ifc++

Ifc++, también conocida como IfcPlusPlus, es otra librería de código abierto desarrollada también para trabajar con archivos IFC, escrita en *C++* y diseñada para permitir la lectura, escritura y manipulación de archivos IFC de manera eficiente, facilitando su integración en aplicaciones que requieren la gestión de modelos BIM. Al ser una herramienta de código abierto, está disponible para ser utilizada y modificada por la comunidad, lo que favorece la colaboración y la personalización según requisitos del proyecto o del cliente. Su principal ventaja es la capacidad para trabajar con grandes volúmenes de datos complejos, garantizando un buen rendimiento y permitiendo la manipulación en tiempo real de los modelos.¹⁷⁴

La funcionalidad de Ifc++ incluye, además, herramientas para la visualización en 3D de modelos BIM. La visualización es uno de los aspectos más importantes en los proyectos de construcción, ya que facilita la comprensión del diseño, la detección de posibles problemas de interferencia y la validación de los requisitos antes de pasar a las fases de construcción. La librería permite mostrar los modelos en tres dimensiones y navegar por ellos de manera interactiva, lo que mejora la toma de decisiones y la colaboración entre los diferentes actores involucrados en el proyecto. Además de su enfoque en la visualización y manipulación de modelos, Ifc++ permite realizar diversas operaciones sobre los datos de los modelos de manera modular, lo que facilita la implementación de nuevas características y la integración con otros sistemas. Asimismo, la librería se mantiene actualizada para cumplir con los estándares y las normativas de la industria, lo que garantiza que se pueda utilizar en proyectos actuales sin problemas.

Ifcjs

Ifcjs es otra de las librerías de código abierto para el trabajo con IFC, pero en este caso, basada en *JavaScript*. Está diseñada para facilitar la manipulación, visualización y análisis de datos BIM directamente en aplicaciones web. Al estar desarrollada en *JavaScript*, permite la integración sencilla en entornos web, lo que la convierte en una opción ideal para desarrollar aplicaciones accesibles a través de navegadores sin la necesidad de dependencias adicionales o instalaciones de software pesados.¹⁷⁵

La librería ofrece herramientas para la visualización de modelos 3D, para la extracción de información y para la renderización de manera interactiva desde el navegador a partir del uso del motor *Three.js*. Otra ventaja clave es que se integra fácilmente con otras tecnologías web, como *HTML*, *CSS* y frameworks populares de *JavaScript*. Esto permite construir interfaces de usuario personalizadas para interactuar con los modelos. Su facilidad de uso también radica en su diseño modular, el cual integra solo las funcionalidades necesarias para sus aplicaciones específicas, sin tener que sobrecargar el código. Además, es altamente extensible, lo que significa que los desarrolladores pueden agregar nuevas características o adaptar la librería a las necesidades particulares de sus proyectos. En cuanto a su comunidad de desarrolladores, se mantiene activa, con múltiples creaciones de plugins y herramientas adicionales que amplían las capacidades de la librería.

¹⁷⁴ About Ifc++. IfcQuery project for reading, writing, and viewing IFC files.

¹⁷⁵ Open BIM Components. That Open Company. Actualización 2025.

3.1.3.3. Automatización, programación y entornos

El trabajo con modelos BIM basados en IFC no se limita únicamente a su visualización o edición mediante interfaces gráficas convencionales. A medida que los flujos de trabajo OpenBIM se consolidan en entornos colaborativos y multidisciplinares, se hace necesario implementar estrategias más avanzadas para la manipulación, análisis y gestión automatizada de estos modelos. En este contexto, la programación ofrece un conjunto de herramientas y entornos que permiten extender las capacidades del formato, habilitando nuevas funcionalidades.

Este apartado presenta una clasificación funcional de las principales herramientas empleadas para la automatización y programación aplicadas a archivos IFC. Se distinguen dos enfoques principales, la programación visual, que permite manipular modelos mediante flujos gráficos, y la programación textual, que posibilita el desarrollo de scripts y aplicaciones personalizadas. Adicionalmente, se incluyen plataformas complementarias que facilitan la ejecución, integración o colaboración en entornos locales, en la nube o en aplicaciones web. Esta estructura busca proporcionar una visión sistemática de las posibilidades técnicas actuales en el procesamiento avanzado de modelos IFC.

Aplicaciones para visualización y análisis básico

La visualización de archivos IFC constituye una etapa fundamental en el proceso de revisión, coordinación y validación de información en entornos BIM abiertos. Existen diversas aplicaciones orientadas específicamente a la inspección visual de archivos IFC, sin requerir funcionalidades avanzadas de edición o programación. Estas herramientas permiten explorar la estructura jerárquica del modelo, identificar elementos según su clase o función, y comprender la organización semántica de los datos, lo cual resulta especialmente útil en tareas de auditoría o supervisión técnica. Entre estas aplicaciones, se encuentran ggIfc Tree Viewer e Ifc Quick Browser, que ofrecen una navegación simplificada por la estructura del modelo, permitiendo una lectura clara de las entidades y sus relaciones sin necesidad de abrir el archivo en un entorno de modelado completo.

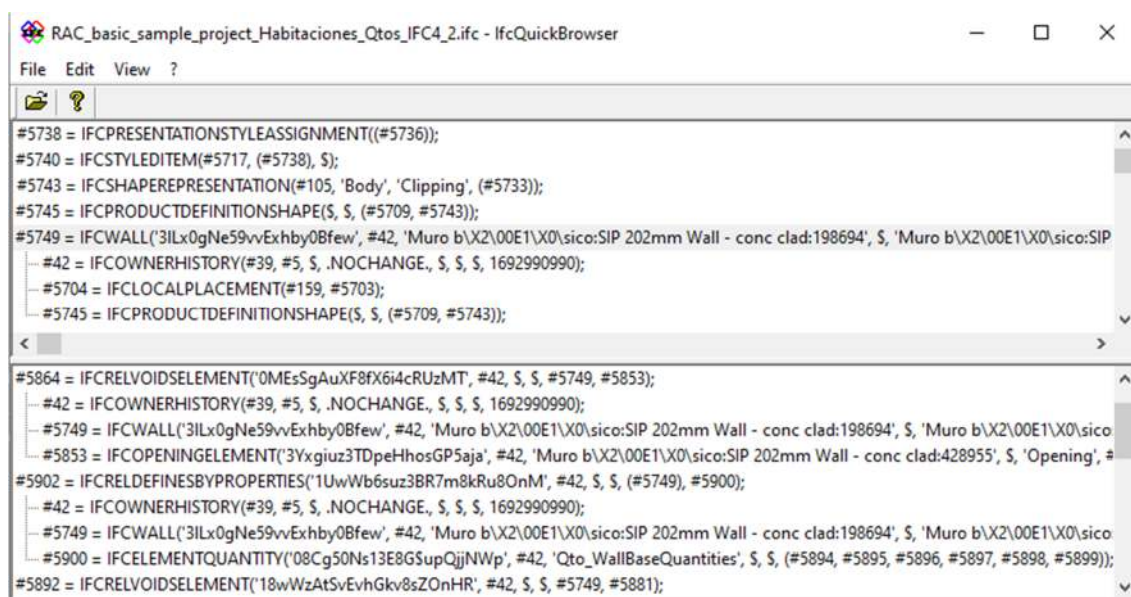


Figura 3.1.12. Relaciones de un muro desde IfcQuickBrowser

Además de las herramientas centradas en la visualización estructural, existen soluciones orientadas al análisis cuantitativo de los datos contenidos en el modelo, particularmente aquellas que permiten su exportación a formatos de hoja de cálculo. Este tipo de aplicaciones responde a la necesidad de integrar la información BIM con procesos de planificación, control o presupuestación mediante la transformación de datos estructurados en tablas editables. En este contexto, Ifc File Analyzer, desarrollado por el NIST,¹⁷⁶ constituye una herramienta de escritorio que permite procesar archivos IFC y generar informes en formato *.xls* o *.csv*. Esta funcionalidad facilita la extracción automatizada de propiedades, clases y cantidades asociadas a los elementos del modelo, haciendo posible su análisis fuera del entorno BIM. Estas soluciones enfocadas en la visualización y la extracción de datos desempeñan un papel clave en la democratización del acceso a la información contenida en los modelos IFC. Al operar sin depender de licencias propietarias y con requisitos técnicos mínimos, permiten que diferentes perfiles profesionales, incluso aquellos no especializados en modelado tridimensional, puedan participar en diferentes procesos de desarrollo y gestión.

Edición de archivos

La edición de archivos IFC es necesaria cuando se requiere ajustar, enriquecer o depurar modelos sin comprometer su integridad estructural ni su interoperabilidad. A diferencia de las herramientas de visualización o extracción de datos, los entornos de edición permiten modificar directamente el contenido del archivo, ya sea a nivel de propiedades, geometría o estructura semántica. Estas operaciones se realizan con especial cuidado, ya que el formato IFC presenta una estructura compleja basada en relaciones entre entidades que, al ser alteradas sin conocimiento técnico, pueden generar inconsistencias o pérdida de información.

En su forma más básica, esta edición puede realizarse manualmente a través de editores de texto convencionales como Bloc de notas o NotePad++. Esta práctica, sin embargo, es altamente arriesgada, ya que la intervención directa sobre el código IFC puede comprometer las relaciones geométricas y semánticas del modelo. Su uso suele restringirse a tareas puntuales y muy controladas, como el reemplazo de nombres de tipos de objetos a través de técnicas de búsqueda y reemplazo.

Frente a las limitaciones de la edición manual, algunas empresas de software BIM han lanzado aplicaciones que permiten una edición más controlada y organizada. Un ejemplo es usBIM.editor de ACCA Software, una herramienta que permite visualizar modelos IFC y también editarlos de manera avanzada. Con esta aplicación es posible modificar propiedades existentes, añadir nuevas entidades, reemplazar geometrías y gestionar clasificaciones personalizadas, manteniendo siempre la coherencia y la consistencia del modelo de información.

Otras soluciones de herramientas de software libre de código abierto han ido surgiendo gradualmente para ofrecer una aproximación más robusta y segura al tratamiento de archivos IFC. Una de las soluciones más destacadas es la combinación de Blender con el complemento Bonsai,¹⁷⁷ anteriormente conocido como BlenderBIM. Esta integración permite trabajar con modelos directamente en un

¹⁷⁶ National Institute of Standards and Technology. Estados Unidos.

¹⁷⁷ Bonsai 0.8.1 documentation. Exploring an IFC model. OSArch community. 2024.

entorno tridimensional, sin depender de plataformas propietarias, y se basa en el uso de la biblioteca IfcOpenShell para interpretar y manipular archivos IFC. Gracias a esta arquitectura, la aplicación permite la edición paramétrica de objetos, el control de versiones, la inspección de elementos y la modificación de propiedades. De manera similar, FreeCAD¹⁷⁸ mediante su módulo BIM, también proporciona un entorno de modelado paramétrico con soporte nativo para IFC. El soporte de este último para planos 2D es más maduro en comparación con Bonsai, sin embargo, ambas herramientas se complementan en diversos aspectos, ofreciendo alternativas flexibles según las necesidades del proyecto.

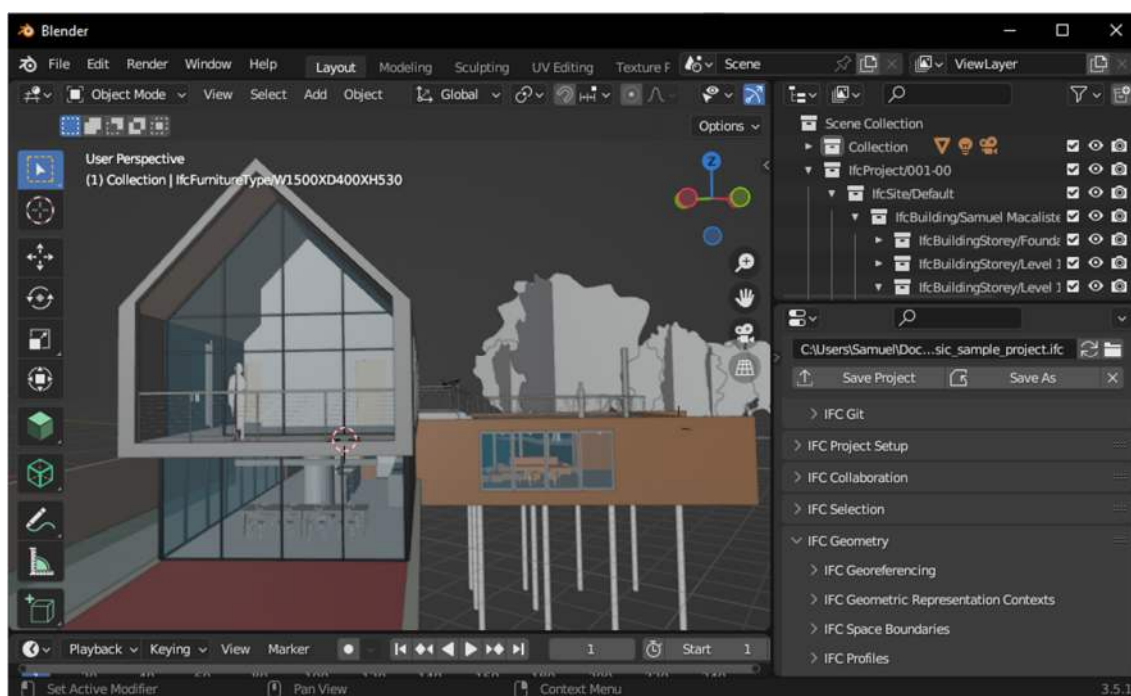


Figura 3.1.13. Herramientas de edición IFC en Blender

Existen además aplicaciones especializadas en la optimización y depuración de modelos IFC que, aunque no ofrecen modelado geométrico, permiten la edición estructurada de datos. Entre ellas se encuentra SimpleBIM,¹⁷⁹ una herramienta diseñada para limpiar y organizar archivos antes de su exportación o reutilización. A través de interfaces intuitivas, esta plataforma facilita el filtrado de información, la eliminación de datos innecesarios y la modificación de atributos, contribuyendo a mejorar la calidad y la consistencia del modelo.

Programación visual

La programación visual ha emergido como una nueva metodología en el ámbito de la gestión de modelos BIM, permitiendo a los profesionales interactuar con datos complejos de manera intuitiva mediante interfaces gráficas. Esta aproximación facilita la creación y automatización de flujos de trabajo sin necesidad de una codificación textual extensa, lo que democratiza el acceso a técnicas avanzadas de manipulación de datos del archivo IFC.

¹⁷⁸ Native IFC. FreeCAD Documentation. Actualización 2025.

¹⁷⁹ Clean-up Models. SimpleBIM Features. 2025.

Una de las herramientas más destacadas en este contexto es Dynamo,¹⁸⁰ un entorno de programación visual que se integra estrechamente con diferentes programas de Autodesk como Revit, Civil 3D o FormIt, aunque también dispone de una versión standalone, denominada Sandbox, que permite programar de manera independiente. En cualquier caso, esta aplicación permite acceder y modificar la estructura de datos de los modelos BIM, facilitando la creación de algoritmos que automatizan tareas repetitivas y mejoran la eficiencia en el diseño paramétrico. Su sistema basado en nodos y conexiones permite definir relaciones y secuencias de acciones de manera gráfica, lo que resulta especialmente útil en la generación y modificación de geometrías complejas.

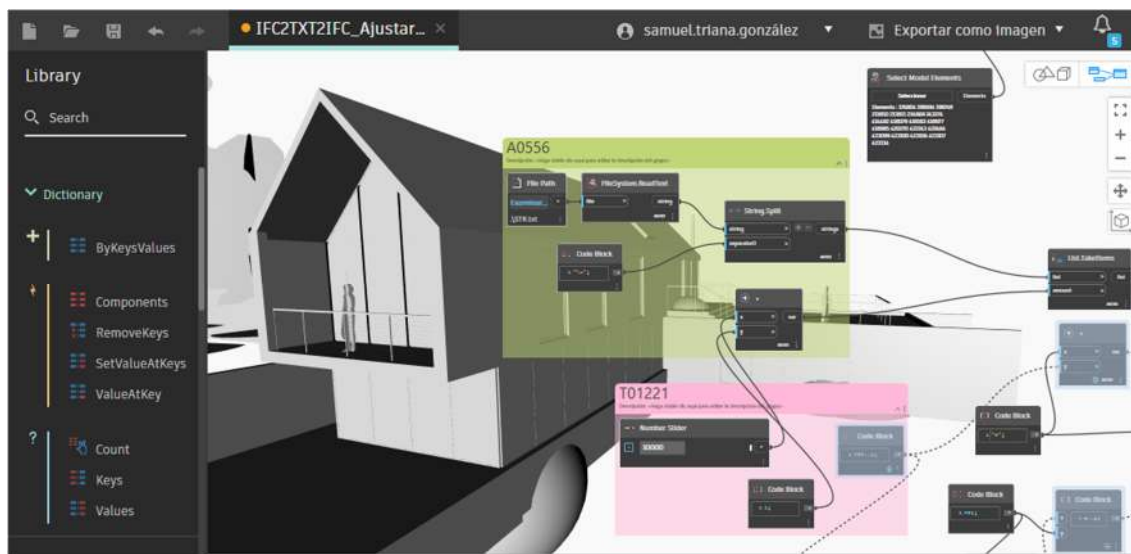


Figura 3.1.14. Programación visual en Dynamo

Además de Dynamo, existen otras herramientas de programación visual que han cobrado relevancia en los flujos de trabajo BIM basados en IFC. Por un lado, Grasshopper,¹⁸¹ que permite el desarrollo de geometrías paramétricas y el diseño generativo mediante componentes visuales interconectados. Su funcionalidad en entornos BIM se ha visto potenciada por plugins como Geometry Gym o la integración con plataformas de gestión, que habilitan la interoperabilidad con formatos abiertos. Por su parte, FME¹⁸² ofrece una amplia interfaz para transformar y convertir datos entre múltiples formatos, permitiendo flujos visuales complejos que incluyen la lectura, edición y exportación de archivos IFC, además de su integración con otras fuentes de datos como GIS o bases relacionales. Asimismo, KNIME¹⁸³ ha comenzado a incorporar capacidades orientadas al procesamiento de modelos IFC, facilitando el uso de técnicas de minería de datos, estadística avanzada y aprendizaje automático sobre información extraída directamente de modelos constructivos.

Estas plataformas de programación visual no solo mejoran la eficiencia y calidad del trabajo con datos IFC, sino que también promueven una cultura de automatización, trazabilidad y apertura. Al reducir las barreras técnicas de entrada a la

¹⁸⁰ The Dynamo Primer. Mode Lab, Parallax Team and Autodesk. v2.0. 2019.

¹⁸¹ Entorno gráfico integrado en Rhinoceros 3D enfocado en la programación visual.

¹⁸² Feature Manipulation Engine. Safe Software.

¹⁸³ Plataforma open Source de análisis de datos modular y visual.

programación, facilitan la integración de disciplinas diversas dentro del flujo BIM, alineándose con los principios fundamentales del OpenBIM y fortaleciendo la interoperabilidad en entornos colaborativos de edificación.

Programación textual

La programación textual constituye un enfoque clave en entornos BIM donde se requiere un control detallado sobre la lógica operativa, la manipulación estructurada de datos y la automatización de procesos asociados al modelo IFC. A diferencia de los entornos visuales, en los que la lógica se construye mediante componentes gráficos, la escritura directa de código permite implementar algoritmos complejos, validar estructuras normativas, generar reportes automatizados y desarrollar soluciones personalizadas con alta granularidad y escalabilidad. Este tipo de programación resulta especialmente eficaz en contextos de análisis avanzado, transformación semántica de modelos y vinculación con sistemas externos.

Cabe señalar que existen entornos híbridos, como Dynamo que, aunque se clasifican dentro de la programación visual, permiten incorporar nodos de scripting mediante Python y diferentes librerías, habilitando funcionalidades adicionales que trascienden la manipulación exclusivamente gráfica. No obstante, su enfoque sigue orientado a la construcción visual del flujo de datos y a la accesibilidad para usuarios no especializados en programación, por lo que se diferencian conceptualmente de los entornos puramente textuales, en los que la interacción con el modelo se produce exclusivamente a través de código fuente.

Los entornos de desarrollo integrados, como Visual Studio Code y PyCharm,¹⁸⁴ son ampliamente utilizados para la creación de soluciones BIM mediante lenguajes como Python, C# y JavaScript. Estas plataformas ofrecen funcionalidades de edición avanzada, depuración, gestión de entornos virtuales y control de versiones, que permiten mantener una trazabilidad precisa del desarrollo e incrementar la modularidad del código. Su compatibilidad con extensiones específicas facilita la conexión con APIs orientadas al análisis de datos.

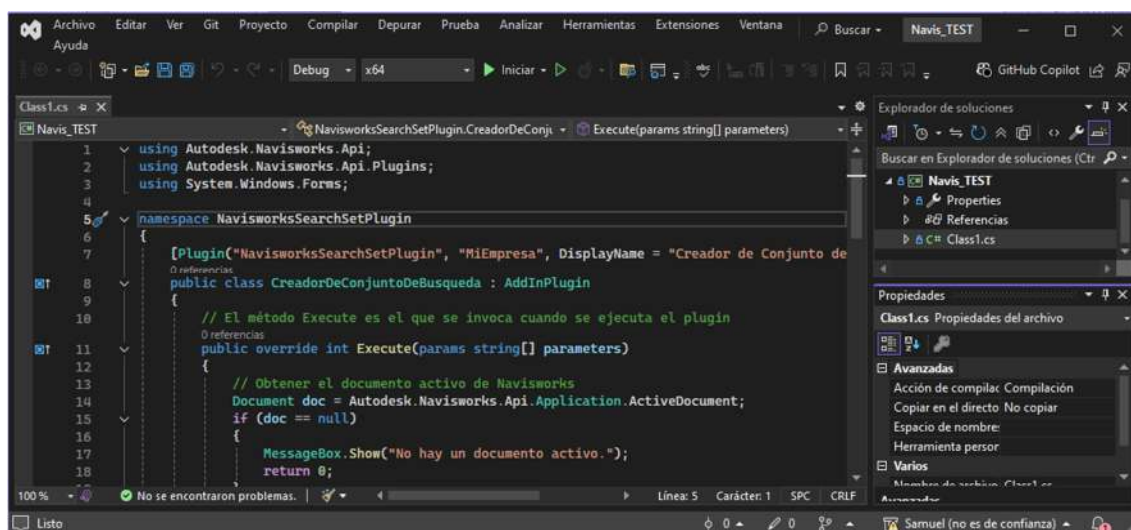


Figura 3.1.15. Programación con APIs en Visual Studio

¹⁸⁴ Learn Python in your IDE. PyCharm for Education. 2025.

En flujos de trabajo orientados a la experimentación computacional y la validación interactiva, los entornos basados en notebooks, como Jupyter,¹⁸⁵ permiten ejecutar scripts organizados en celdas, favoreciendo un enfoque exploratorio y reproducible. La posibilidad de combinar instrucciones de código con anotaciones técnicas y visualizaciones interactivas convierte este entorno en una herramienta útil para la extracción, transformación y análisis de datos IFC. En su modalidad remota, Google Colaboratory ofrece capacidades equivalentes sin requerimientos de instalación local, posibilitando el uso compartido de recursos computacionales y facilitando la colaboración distribuida en contextos de investigación o desarrollo de soluciones escalables.

Para el despliegue de aplicaciones web con capacidades de interacción con modelos IFC, se ha extendido el uso de librerías JavaScript que permiten visualizar y manipular el modelo directamente en el navegador. Ifc.js destaca en este ámbito por ofrecer visualización 3D, navegación semántica y edición básica de propiedades sin necesidad de software propietario ni infraestructura local. Estas funciones se ejecutan en el entorno del cliente mientras que, en el servidor, tecnologías como Node.js¹⁸⁶ permiten gestionar archivos, peticiones y servicios relacionados con datos IFC. La combinación de Ifc.js en el navegador y Node.js en el servidor habilita el desarrollo de aplicaciones web completas para la gestión de modelos BIM en entornos distribuidos.

La adopción de flujos de trabajo basados en programación textual permite un alineamiento más estricto con los principios de interoperabilidad, reproducibilidad y modularidad que sustentan el enfoque OpenBIM. A través de la escritura explícita de lógica computacional, es posible desarrollar herramientas que respondan a necesidades específicas del ciclo de vida del modelo BIM, facilitando procesos de análisis estructurado y verificación automática.

Plataformas complementarias y entornos de ejecución

Dentro del ecosistema BIM, existen plataformas tecnológicas que, si bien no constituyen herramientas de edición directa del modelo, desempeñan funciones esenciales en la integración, ejecución, colaboración y tratamiento avanzado de datos. Estas tecnologías actúan como infraestructuras de soporte para el desarrollo y despliegue de soluciones openBIM, facilitando la interoperabilidad, el trabajo colaborativo distribuido y el procesamiento a gran escala.

BIMserver¹⁸⁷ constituye una de las principales soluciones de código abierto orientadas a la gestión centralizada de modelos IFC. Diseñado específicamente para entornos colaborativos, este servidor permite el almacenamiento estructurado, la consulta, la modificación remota y el control de versiones de archivos IFC mediante su API. La arquitectura de la plataforma permite implementar flujos de trabajo asincrónicos entre usuarios y agentes, promoviendo la interoperabilidad entre aplicaciones de terceros y facilitando la validación automatizada en entornos multiusuario. Además, el uso de contenedores virtualizados, a través de tecnologías

¹⁸⁵ Project Jupyter Documentation. Jupyter Notebook and JupyterLab web-based notebook authoring and editing applications. 2015.

¹⁸⁶ Entorno de ejecución JavaScript, desarrollado por Ryan Dahl en 2009.

¹⁸⁷ BIMserver - an open-source Building Information Model server. BIMserver wiki. 2024.

como Docker,¹⁸⁸ permite encapsular aplicaciones BIM junto con sus dependencias, asegurando su ejecución en entornos reproducibles, escalables y aislados del sistema anfitrión. Esta estrategia despliega soluciones que dependen de bibliotecas específicas como IfcOpenShell o xBIM, evitando problemas de compatibilidad entre sistemas operativos o versiones. Los contenedores también facilitan la integración de flujos de procesamiento automático en arquitecturas distribuidas o basadas en microservicios.

En cuanto al análisis masivo de datos, plataformas como BigQuery¹⁸⁹ ofrecen infraestructuras especializadas para el tratamiento computacional de grandes volúmenes de información provenientes de modelos IFC. Mediante el uso de lenguajes de consulta estructurada y herramientas de aprendizaje automático, es posible transformar datos de construcción en conocimiento accionable. Estas capacidades permiten realizar minería de datos, evaluación de tendencias, segmentación de atributos y generación de modelos predictivos aplicables en procesos de mantenimiento, eficiencia energética o gestión operativa del entorno construido.

Speckle¹⁹⁰ representa una plataforma orientada a la interoperabilidad y transmisión estructurada de datos BIM en entornos colaborativos. Permite la sincronización de modelos entre herramientas como Revit, Rhino o cualquier software compatible con IFC, y ofrece funcionalidades de visualización, versionado y control de cambios. Su arquitectura basada en servicios web facilita la integración con flujos de trabajo distribuidos y la transmisión de datos en tiempo real, lo que refuerza su utilidad en entornos de diseño colaborativo y automatización de procesos entre disciplinas.

Estas plataformas y entornos complementarios expanden significativamente las posibilidades operativas del ecosistema IFC, al proporcionar soporte para la ejecución, colaboración, integración de servicios y análisis de datos. Su incorporación estratégica refuerza la infraestructura digital sobre la cual se desarrollan y operan los modelos BIM, alineándose con los principios de escalabilidad, interoperabilidad y sostenibilidad tecnológica que fundamentan la transición hacia entornos construidos más inteligentes y conectados.

¹⁸⁸ Docker Engine. Manuals. Open-Source containerization technology for building and containerizing your applications.

¹⁸⁹ BigQuery. Recursos de documentación. Descripción general. 2025.

¹⁹⁰ Speckle Docs. User Guide. 2025.

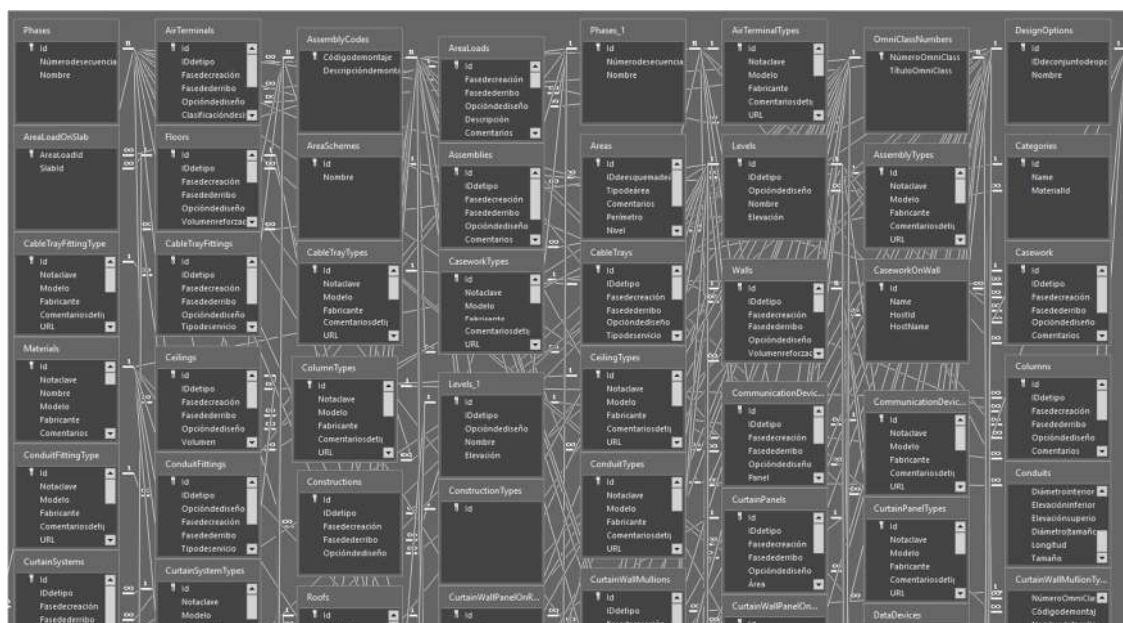
Actualmente, existen diversas soluciones tecnológicas que permiten establecer conexiones dinámicas y bidireccionales entre modelos BIM y bases de datos externas. Estas conexiones posibilitan la sincronización automatizada de datos, la extracción de información específica del modelo, y la actualización en tiempo real de atributos y propiedades vinculados a elementos constructivos. En la mayoría de los casos, las aplicaciones o plataformas incorporan funcionalidades que permiten estas integraciones mediante APIs o plugins personalizados.

3.1.4.1. Software de autoría BIM

[illegible]

¹⁹¹ Autodesk Revit DB Link. Autodesk 2024.

Uno de los aspectos clave en esta relación es la estructura relacional de la base de datos, en la que los registros se identifican por campos únicos que aseguran la coherencia de los datos durante procesos sucesivos de exportación e importación. Esto permite que campos y tablas adicionales permanezcan intactos, preservando la integridad de la información aún tras múltiples iteraciones del modelo. Sin embargo, es fundamental que las operaciones de importación y exportación se realicen con la misma base de datos utilizada originalmente, ya que de lo contrario pueden producirse errores o inconsistencias en los datos del modelo.



Otras formas de conexión con bases de datos externas se realizan a partir del uso de la programación, utilizando complementos o el uso de Dynamo, mediante paquetes específicos desarrollados para la conexión con bases de datos como MySQL o VM SQL Server. En este contexto, el uso de herramientas como Slingshot,¹⁹² permite a los usuarios conectar Grasshopper o Dynamo con sistemas de bases de datos relacionales, ofreciendo una interfaz entre modelos de diseño y repositorios estructurados de datos.

¹⁹² Paquete de Dynamo desarrollado por Nathan Miller de Proving Ground.

generar la cadena de conexión, y *Command.MySQL_Command* para ejecutar comandos de estructura o manipulación de datos en el servidor.¹⁹³ Esta arquitectura no sólo permite la escritura de datos, sino también la lectura y actualización de registros desde Dynamo, permitiendo la creación de flujos de trabajo bidireccionales entre el modelo BIM y la base de datos.

De esta manera, se pueden desarrollar procesos en los que, por ejemplo, datos espaciales del modelo de Revit sean utilizados para establecer relaciones entre objetos de mobiliario y las habitaciones que los contienen. Esta relación espacial permite extraer datos relevantes de los espacios y asignarlos como parámetros a los elementos de mobiliario. Posteriormente, estos datos son estructurados en listas y enviados a la base de datos MySQL mediante sentencias SQL generadas desde Dynamo.

La creación de nuevos esquemas y tablas en el servidor MySQL también se ejecuta desde Dynamo, utilizando bloques de código para construir sentencias del tipo *CREATE*, en las que se especifican los nombres de las columnas, tipos de datos y claves primarias. En el caso de una tabla de muros, cada registro puede incluir el identificador único de Revit como clave principal, y adicionalmente diferentes campos que definan otros datos como materiales, tipo de familia, dimensiones, etc.

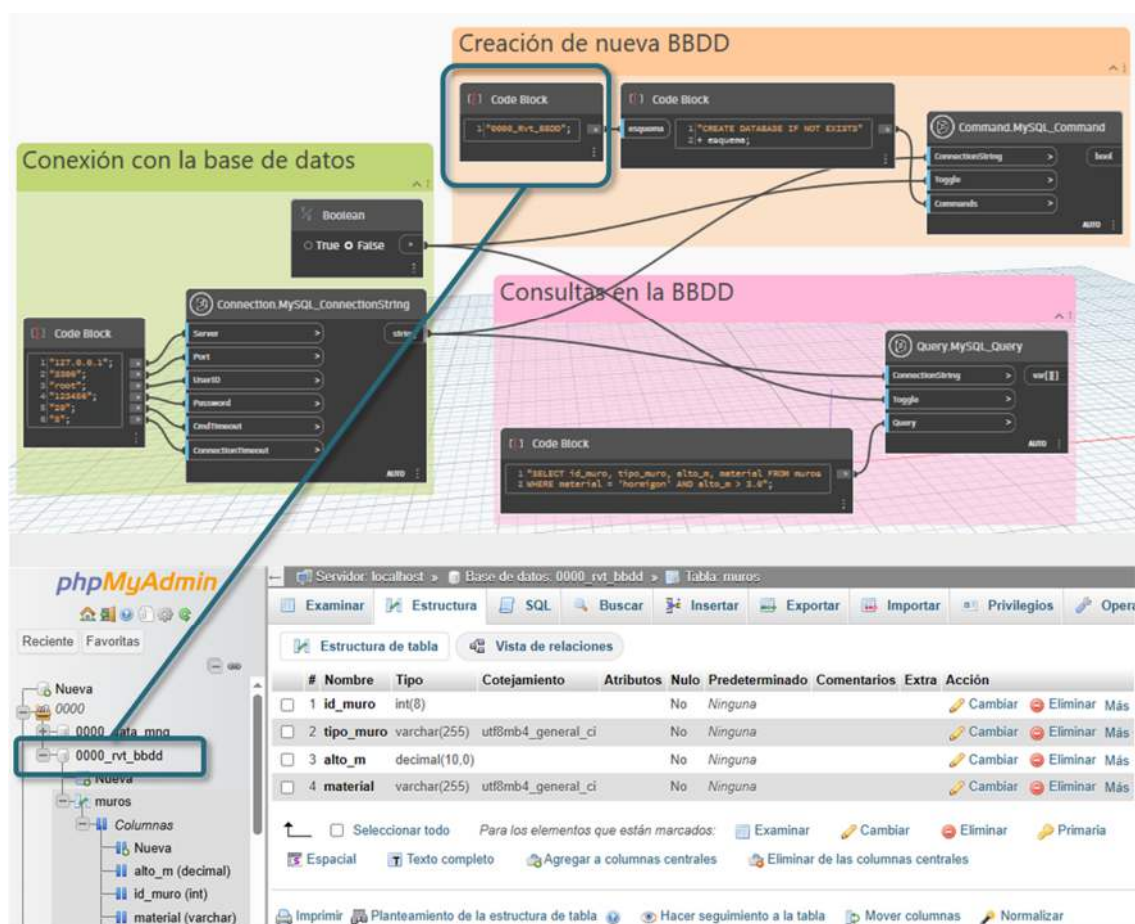


Figura 3.1.18. Conexión de datos entre Dynamo y MySQL

¹⁹³ Driving a SQL Database with Dynamo. Revit beyond BIM. Dieter Vermeulen. 2015.

Una vez que se crean las tablas dentro de la base de datos, se puede realizar una recolección de datos de los objetos del modelo, generando listas de valores mediante nodos de extracción de parámetros como *Element.GetParameterValueByName*¹⁹⁴ y posteriormente se estructuran todos ellos como cadenas SQL a través del nodo *SQL.InsertInto*. Estas cadenas finalmente son enviadas al servidor MySQL utilizando *Command.MySQL_Command*, completando así el ciclo de transferencia de información desde el entorno de modelado hacia el repositorio externo.



Figura 3.1.19. Inserción de registros con datos BIM en MySQL

Una vez los datos se almacenan en el servidor, pueden ser manipulados desde herramientas como MySQL Workbench o phpMyAdmin,¹⁹⁵ permitiendo, por ejemplo, modificar valores, completar registros faltantes o realizar análisis mediante consultas SQL personalizadas. Asimismo, las modificaciones realizadas en la base de datos pueden ser recuperadas desde Dynamo, utilizando nodos de consulta como *Query.MySQL_Query*, lo cual habilitaba la retroalimentación de información hacia el modelo de Revit.

Al tratarse de un paquete de código abierto, el proyecto se mantiene disponible bajo licencia GNU, y el repositorio en línea contiene los scripts y librerías que permiten replicar e incluso adaptar estas prácticas en entornos BIM, aunque para ello es necesario un dominio avanzado del lenguaje de programación y conocimiento profundo acerca de la gestión de bases de datos relacionales.

Sin embargo, estos sistemas o herramientas presentan limitaciones importantes, ya que mayormente se enfocan en herramientas específicas y no pueden ser aplicadas o desarrolladas en otros entornos diferentes. Además, existe una incapacidad de operar directamente con archivos OpenBIM, lo que restringe su aplicación a modelos nativos. Asimismo, existen problemas de sincronización vinculados al idioma del software, a la longitud de ciertos valores, como descripciones o códigos de montaje y a restricciones a la hora de definir ciertos tipos de datos y modificar determinados valores de atributos. Por último, no es posible gestionar la información de modelos vinculados de manera conjunta, lo que hace complejo el estudio global de los datos del proyecto.

¹⁹⁴ The Dynamo Primer. Mode Lab, Parallax Team and Autodesk. v2.0. 2019.

¹⁹⁵ phpMyAdmin User Guide. The phpMyAdmin devel team. Actualización 2025.

3.1.4.2. Software de coordinación BIM

En el caso de programas de coordinación, Navisworks Manage posee una herramienta denominada DataTools,¹⁹⁶ la cual permite vincular información externa proveniente de bases de datos con los objetos del modelo 3D, enriqueciendo los datos disponibles para revisión y análisis. Esta herramienta se utiliza principalmente para conectar archivos de proyecto con datos alfanuméricos adicionales, provenientes de fuentes como bases SQL Server, Access, Excel o archivos CSV, sin necesidad de modificar directamente el modelo original. A través de esta vinculación, es posible visualizar y consultar propiedades adicionales asociadas a elementos del modelo, lo cual resulta útil en procesos de coordinación, control de obra o planificación.

El sistema permite mapear campos específicos entre los identificadores del modelo y los datos externos, generando una relación dinámica que se actualiza al recargar el modelo o el archivo de datos. Esto facilita, por ejemplo, la incorporación de información de costos, plazos, códigos de clasificación o atributos personalizados desde sistemas de gestión de proyectos o ERP, directamente dentro del entorno de revisión de Navisworks. La visualización de estos datos se hace posible mediante el panel de propiedades extendidas, sin alterar la estructura interna del archivo NWD.

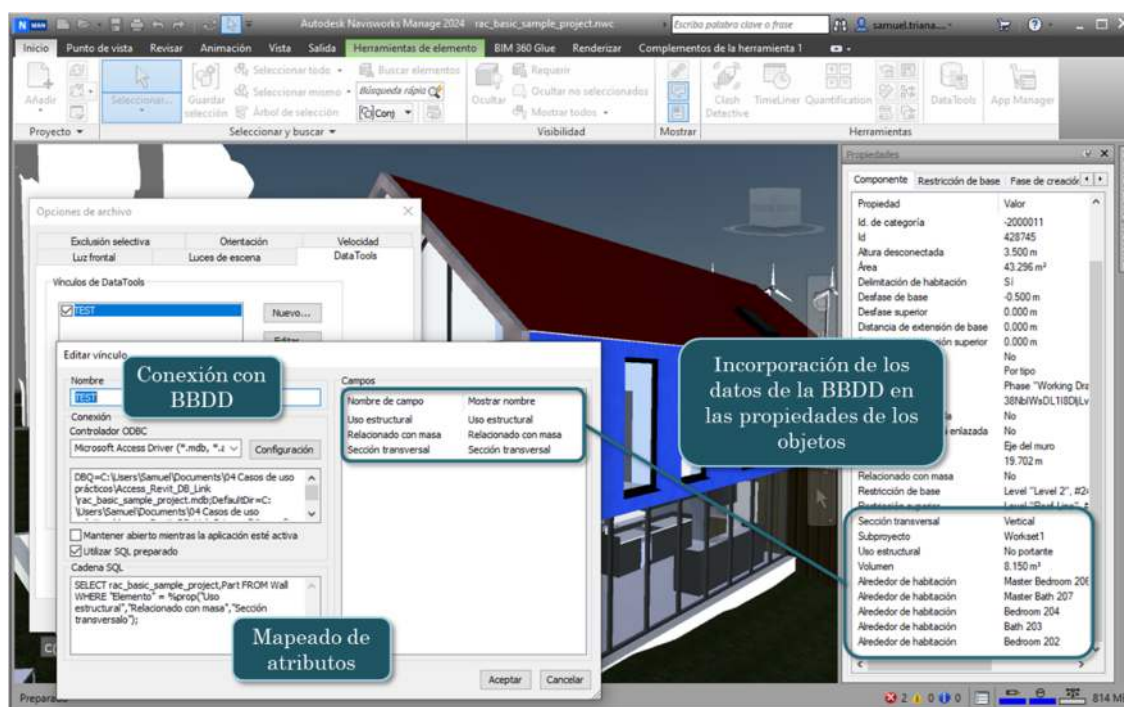


Figura 3.1.20. Importación de datos de la BBDD en Navisworks

Sin embargo, esta herramienta no realiza modificaciones permanentes en el modelo ni incorpora los datos a los objetos de manera nativa, sino que actúa como una capa de información vinculada. La utilidad se asocia a revisiones colaborativas o auditorías técnicas, en las que se requiere complementar la geometría con datos externos sin pasar por procesos de edición desde plataformas de modelado.¹⁹⁷

¹⁹⁶ Datatools. Autodesk Navisworks Manage. Autodesk Support. 2023.

¹⁹⁷ Linking Excel Data to Navisworks Model. Autodesk Enterprise Priority. 2017.

3.1.4.3. Plataformas y soluciones en la nube

Otra de las aproximaciones más flexibles para conectar modelos BIM con bases de datos relacionales es el desarrollo de pipelines ETL (Extract, Transform, Load) personalizadas. Mediante el uso de la API REST de plataformas como Autodesk Construction Cloud (ACC),¹⁹⁸ es posible extraer datos como incidencias, tareas o usuarios, y cargarlos en bases como MySQL o SQL Server. Al alojar estos datos en esquemas relacionales optimizados, se habilita su explotación mediante herramientas analíticas o su integración con sistemas ERP y de planificación. Bases como MySQL resultan estratégicas en entornos nativos en la nube, mientras que SQL Server y Oracle destacan por su capacidad de procesamiento en entornos empresariales más complejos.

Como alternativa al desarrollo personalizado, han emergido otro tipo de soluciones denominadas Low-Code/No-Code (Parimi, 2025) como Portable, Workato o Autodesk Construction Cloud Connect, que permiten integrar plataformas BIM con bases como MySQL, Oracle o PostgreSQL sin necesidad de programación textual. Estas plataformas utilizan conectores preconfigurados, facilitando la configuración de flujos de datos seguros, replicables y monitoreados en tiempo real. Además, en el caso de ACC Connect, es posible definir flujos de trabajo activados por desencadenadores en las propias bases de datos. Por ejemplo, la modificación de un campo en MySQL puede generar automáticamente una incidencia en ACC, o actualizar información en otras aplicaciones de la nube Autodesk.

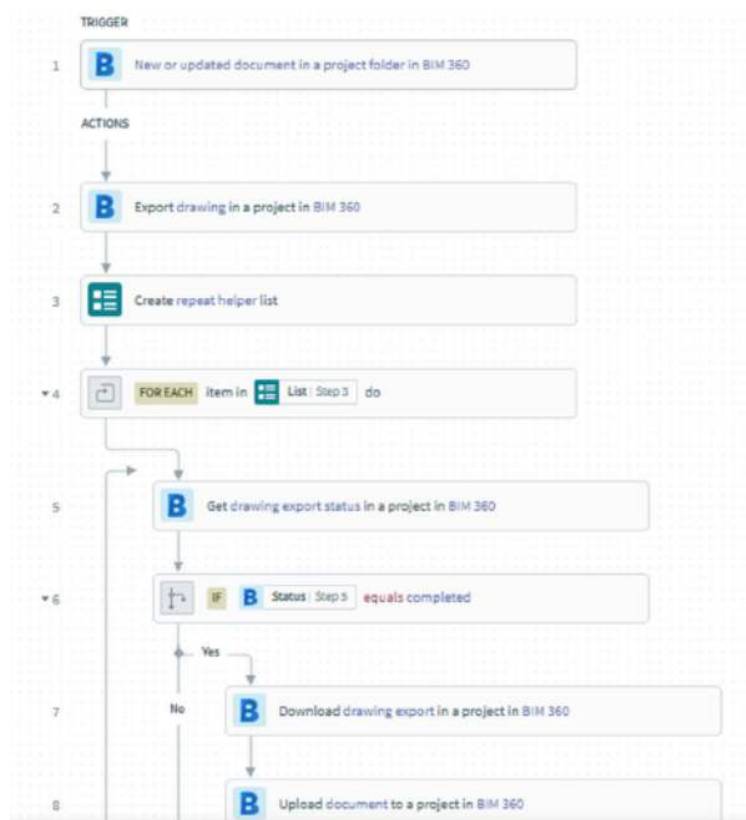


Figura 3.1.21. Desencadenador y acciones con Workato y ACC Connect

¹⁹⁸ Integration Partner Ecosystem. Autodesk Construction Cloud.

Por otro lado, herramientas como Plannerly permiten definir requerimientos de información y atributos estructurados, por ejemplo, la longitud permitida de un conducto, que pueden sincronizarse con modelos en Revit o IFC. Otras soluciones como BIM&CO permiten normalizar y estructurar datos BIM en línea, mientras que herramientas como Speckle o Simplebim actúan como puentes entre modelos IFC y sistemas de bases o fuentes de información externas.

Si bien las integraciones ofrecen un enorme potencial transformador, ya que permiten automatización, trazabilidad, análisis avanzado y decisiones basadas en datos, también implican importantes desafíos técnicos. La implementación de arquitecturas de datos conectadas con modelos BIM requiere inversiones sustanciales en infraestructura tecnológica, personal especializado en desarrollo y administración de APIs, y recursos continuos para mantenimiento y escalabilidad. Además, exige conocimientos avanzados en autenticación, manejo de estructuras anidadas, programación de integraciones y despliegue de pipelines ETL personalizados. Para muchos equipos, especialmente en proyectos de menor escala o sin una estructura tecnológica madura, este nivel de complejidad puede resultar excesivo.

Frente a ello, otras alternativas más ligeras y accesibles que no requieran el despliegue completo de integraciones a gran escala pueden ser de gran valor. El uso de hojas de cálculo inteligentes, la explotación directa de propiedades generadas desde modelos IFC, o incluso el empleo de plataformas colaborativas con capacidades básicas de análisis sumadas a pequeñas herramientas y automatizaciones que conecten los datos entre sistemas BIM y bases de datos, pueden resolver las principales necesidades de conectividad de manera efectiva.

3.1.4.4. Investigaciones relacionadas

En los últimos años, diversos estudios han explorado este tipo de integraciones como una vía para optimizar la interoperabilidad de datos, automatizar procesos y mejorar la calidad en el sector de la construcción. Una de las propuestas más relevantes (Le et al., 2020) aborda el desarrollo de un sistema integrado que vincula software de autoría BIM, hojas de cálculo, bases de datos y herramientas de programación visual para facilitar el análisis de costos del ciclo de vida de los edificios. Este enfoque busca reducir la dependencia de procesos manuales propensos a errores mediante la automatización del flujo de datos desde los modelos BIM hacia una base de datos relacional, lo que permite realizar estimaciones de costos más precisas y sostenibles desde una perspectiva económica. El sistema se diseñó para apoyar decisiones de contratación basadas en criterios de sostenibilidad y fue validado mediante un prototipo funcional en un caso de estudio de edificación institucional, demostrando beneficios concretos en términos de eficiencia y trazabilidad de la información.

En un enfoque complementario, otro estudio (Hewavitharana & Perera, 2018) propone un marco para integrar sistemas BIM y ERP con el fin de mejorar la sostenibilidad y la eficiencia de los procesos digitales dentro de las empresas constructoras. A través del uso de tecnologías interoperables y conectividad basada en bases de datos, esta propuesta busca alinear la información generada en las fases de diseño y construcción con los flujos operativos y financieros gestionados desde el ERP. La investigación enfatiza que esta integración no solo optimiza el rendimiento operativo, sino que también apoya objetivos estratégicos como la trazabilidad de

recursos, el control de emisiones y la evaluación de impacto ambiental, fundamentales en el contexto de la construcción sostenible.

Desde el ámbito de la gestión de activos, se han desarrollado marcos de trabajo que conectan modelos BIM con bases de datos relacionales para cubrir necesidades de Facility Management (FM). Un ejemplo destacado es el estudio (Kang & Choi, 2015) que plantea una conexión directa entre datos del modelo BIM y los sistemas de FM, facilitando la centralización de toda la información del ciclo de vida del activo en una única base de datos. Esta propuesta busca crear un repositorio de información compartido, accesible y actualizado, que permita al personal de mantenimiento y operación consultar atributos clave del edificio, registrar incidencias y realizar análisis sin necesidad de navegar en entornos complejos de modelado. La arquitectura del sistema incluye un modelo de base de datos extensible y un conjunto de servicios de interoperabilidad diseñados para integrar datos gráficos y no gráficos.

Por otro lado, se ha propuesto un esquema de base de datos relacional simplificado y espacialmente habilitado para transformar modelos BIM, en formato IFC, en estructuras consultables con SQL. Esta investigación (Barzegar et al., 2021) destaca la necesidad de contar con bases de datos capaces de realizar consultas eficientes sobre datos geométricos y semánticos, integrando funciones espaciales sin depender de software externo. La propuesta incluye un proceso detallado de limpieza, validación y mapeo de entidades IFC hacia una base de datos que gestiona tanto la información semántica como las representaciones geométricas mediante geometrías poliédricas, lo que permite ejecutar consultas espaciales como análisis de accesibilidad, relaciones topológicas o delimitaciones.

Otra línea de investigación (Alzraiee, 2020) se ha centrado en el desarrollo de sistemas de estimación de costos que superen las limitaciones de los enfoques actuales basados únicamente en cantidades, proponiendo un marco que vincula directamente los elementos BIM con bases de datos de costos, utilizando consultas SQL para establecer una relación semántica y cuantitativa entre elementos del modelo y tareas constructivas específicas. Este sistema permite crear un entorno de estimación de costos más preciso y contextualizado, que se ha demostrado casi tan fiable como las estimaciones manuales, pero con un ahorro significativo en tiempo.

También se han explorado marcos de trabajo para conectar modelos BIM con sistemas de Facility Management mediante la sincronización de datos entre plataformas como Autodesk Revit y bases de datos relacionales como SQLite. Una de estas investigaciones (Trabulci, 2020) emplea Dynamo para extraer atributos del modelo y alimentar una base de datos bidireccional que permite tanto la actualización como la consulta de parámetros no gráficos. Este sistema fue validado en un caso práctico que consistía en la gestión de luminarias en un edificio universitario, y evidenció mejoras notables en la accesibilidad a la información, la trazabilidad y la integración con herramientas de visualización como Power BI.

Finalmente, otra propuesta relevante (Solihin, 2016) explora la transformación de datos BIM en una base de datos relacional estructurada, optimizada para realizar consultas analíticas. A través de una arquitectura que combina datos geométricos y atributos semánticos, se logra una representación completa del modelo que facilita consultas cruzadas y análisis complejos desde SQL, apoyando flujos de trabajo colaborativos y multidisciplinarios en proyectos de gran escala.

3.1.4.5. Esquema IfcSQL

El uso del IFC permite la extracción de información estructurada de archivos para su integración en otros sistemas y plataformas de gestión. Esta metodología facilita la organización y consulta de datos, permitiendo su análisis y reutilización en distintos proyectos. Sin embargo, el proceso de conversión de archivos IFC a bases de datos presenta desafíos significativos, especialmente en lo referente a la estructura de los datos y la eficiencia en su manipulación.¹⁹⁹

Uno de los enfoques recientes para el almacenamiento de datos IFC en bases de datos relacionales es IfcSQL,²⁰⁰ un formato experimental desarrollado con el propósito de mejorar la gestión y accesibilidad de los datos extraídos de archivos IFC. Esta propuesta busca representar la estructura jerárquica de los datos IFC dentro de un esquema relacional, permitiendo consultas más estructuradas y la integración con herramientas analítica. Algunos de los beneficios que ofrece este enfoque incluyen la capacidad de almacenar múltiples modelos en una sola base de datos, la posibilidad de acceso multiusuario y la garantía de integridad referencial entre entidades.

A pesar de estos avances, IfcSQL presenta importantes limitaciones que afectan su eficiencia y aplicabilidad. Dado que los datos IFC están diseñados bajo un modelo orientado a objetos, su conversión a un esquema relacional puede generar estructuras excesivamente complejas, lo que incrementa la cantidad de tablas necesarias y dificulta las consultas. En particular, los tipos de datos como listas y selecciones no son fácilmente representables en una base de datos relacional, lo que puede comprometer la integridad y precisión de la información almacenada. Otro desafío es la falta de escalabilidad en proyectos de gran tamaño, donde el rendimiento de las consultas puede verse afectado por la necesidad de realizar múltiples uniones de tablas.²⁰¹

Dado que IfcSQL aún se encuentra en fase experimental, su adopción en entornos de producción requiere una evaluación cuidadosa de sus ventajas y desventajas. Alternativas desde otros enfoques de conexión podrían ofrecer una solución más eficiente, al buscar otro tipo de adaptaciones para manejar relaciones complejas entre entidades y mejorar el rendimiento en la extracción de datos. En este contexto, la conversión de archivos IFC a bases de datos sigue siendo un área de investigación en desarrollo, en la que es necesario explorar estrategias que optimicen tanto la accesibilidad como la integridad de los datos en el sector de la construcción.

¹⁹⁹ IFC for Relational Databases. Discusión técnica sobre bases de datos relacionales para IFC. BuildingSMART. 2020.

²⁰⁰ IfcSQL Project. Bernhard Simon Bock. BuildingSMART. 2019.

²⁰¹ IfcSQL Documentation. GitHub. Estructuras de almacenamiento y limitaciones.

3.2. Experiencia directa de expertos BIM

3.2.1. Planteamiento inicial

La revisión bibliográfica aporta una visión global del estado y la evolución de la metodología BIM, abarcando gran cantidad de investigaciones y teorías las cuales muchas veces no llegan a profundizar lo suficiente en los temas específicos del estudio. Por el contrario, la percepción de los expertos permite la obtención pormenorizada de información actualizada, específica y contextualizada de diferentes áreas, ya que estos usuarios cuentan con un profundo conocimiento en el campo de estudio concreto. Estas experiencias directas son fundamentales para recoger de primera mano datos relacionados con experiencias recientes, innovaciones o tendencias emergentes.

Con el objeto de definir una metodología para la recolección de estas experiencias, se ha optado por el método de encuesta para obtener un análisis cuantitativo, ya que se trata de una herramienta dinámica que permite analizar respuestas a cuestiones concretas, estandarizando estas cuestiones de una misma manera para todos los participantes. Al tratarse de un área de específica, se toma un segmento de la población, los cuales deben cumplir con diferentes requisitos en cuanto a experiencia y conocimientos del uso y la aplicación de la metodología BIM. Para el caso del presente estudio, este segmento, también denominado muestreo, es no probabilístico²⁰² y de carácter estratégico, ya que por la naturaleza de la investigación no todos los individuos de la población pueden cumplir con los criterios exigidos para participar. De estos criterios, los principales que se han considerado son:

- Experiencia mínima de 10 años en el sector de la construcción.
- Experiencia mayor a 5 años en el campo específico de la metodología BIM.
- Desarrollo de proyectos BIM nacionales de gran impacto o escala.
- Perfiles BIM, con roles de coordinación, dirección o consultoría de alto nivel dentro de proyectos u obras desarrolladas que se desarrollen con la metodología.
- Conocimiento de estándares y formatos OpenBIM.

En referencia a la selección de las personas a encuestar, se realiza un análisis de perfiles profesionales desde distintas redes sociales y páginas web de empresas públicas y privadas. Una vez revisados los datos de cada usuario, se realiza un primer acercamiento para explicar el contexto y propósito del estudio y, en los casos en los que el usuario se muestra interesado, se comparte el cuestionario vía web para que pueda ser autoadministrado.

Por otro lado, con el objetivo de recabar resultados que puedan ser representativos, se establece un tamaño muestral mínimo de 50 expertos y al menos de 30 empresas

²⁰² Evaluación y calidad. Guía de orientación para la realización de estudios de análisis de la demanda y de encuestas de satisfacción. Ministerio de Administraciones Públicas. 2006. Madrid, España.

diferentes que trabajen en proyectos de edificación, pudiendo pertenecer al área de proyectos, obras, mantenimiento o, incluso, otros sectores como consultoría, investigación y docencia.

De este modo, se genera un conjunto estructurado de preguntas categorizadas con una serie de posibles alternativas de respuesta, las cuales pretenden recoger el conocimiento de expertos en la materia. Previamente, se realiza un estudio exhaustivo para establecer los rangos y variables de las posibles respuestas, de manera que se puedan seleccionar opciones de diferentes grados de complejidad y desde múltiples puntos de vista. La finalidad del estudio es recabar opiniones acerca de lo que es importante o no para los encuestados, buscando un esquema representativo que aporte un panorama general de la percepción del conjunto de individuos.

Las cuestiones se han enumerado dentro de un formulario generado a través de Google Forms²⁰³ en el que primero se define un bloque de preguntas de identificación del usuario y, seguidamente, se organizan 5 áreas de conocimiento relacionadas con BIM, las cuales son:

- A) Planificación inicial BIM: relativo a la definición de requisitos previos y a la gestión desde el comienzo del proyecto de todo el ciclo de vida del activo.
- B) Trabajo con atributos y propiedades: aborda el trabajo con plantillas, bibliotecas y la definición de la información necesaria para generar los modelos BIM.
- C) Trabajo con formatos abiertos: relacionado con estándares OpenBIM y su grado de aplicación en proyectos de construcción.
- D) Controles de calidad: se enfoca en el nivel de control de entregables que existe para garantizar el cumplimiento de los requisitos solicitados por el cliente.
- E) Futuro del BIM: en referencia a posibles interacciones con otras metodologías y herramientas emergentes del sector de la construcción.

Para minimizar al máximo los posibles sesgos o errores debidos a respuestas parciales o incompletas, se añade un apartado al final del formulario para que los usuarios puedan añadir sus propios comentarios o aclaraciones a cualquiera de las cuestiones planteadas.

En cuanto a la estructura interna del formulario, la finalidad de la encuesta se expone en el primer apartado de introducción, donde se indica de forma clara que se trata de un estudio para conocer el nivel de madurez en la gestión de procesos y datos BIM desde la perspectiva de los expertos BIM del sector AECO, que la duración estimada del formulario es de menos de 10 minutos y se presenta a los responsables de la encuesta.

²⁰³ Herramienta en la nube de Google que permite la creación de formularios con el propósito de recopilar información de manera ordenada, generando gráficos de los resultados obtenidos.

Encuesta Gestión Datos BIM - PhD - UPM

Introducción:

El presente cuestionario forma parte del trabajo de investigación "Gestión y control de los datos BIM a lo largo del ciclo de vida en proyectos de edificación" desarrollado por [D. Samuel J. Triana González](#) y que está dirigiendo el [Doctor Rafael Lozano Díez](#) en la **Universidad Politécnica de Madrid (UPM)**. Este trabajo aborda el uso de los datos BIM a través formatos abiertos y su interacción con otras metodologías y entornos digitales.

Objetivo:

El objetivo de la encuesta es **conocer el nivel de madurez en la gestión de procesos y datos BIM** desde la perspectiva de los expertos BIM del sector AECO.

Tiempo estimado:

Esta encuesta tiene una **duración aproximada de 5 minutos**.

Muchas gracias por su apoyo y colaboración.

**La información recogida en este cuestionario será exclusivamente utilizada para fines académicos y de investigación. Asimismo, es un modo de colaborar con el resto del sector para definir el grado actual de integración de la metodología y de poder determinar posibles mejoras en procesos y métodos de aplicación.*

Las respuestas y la identidad de los participantes serán confidenciales. Al responder a este cuestionario, el participante consiente el tratamiento de los datos por parte del investigador y, en cualquier caso, se le recuerda que se encuentra bajo el amparo de la **Ley Orgánica de Protección de Datos Personales y Garantía de los Derechos Digitales.*

Figura 3.2.1. Presentación de encuesta a expertos BIM

En lo referente a la creación de las cuestiones, se estudia de forma pormenorizada cada una de ellas con el fin de que el conjunto de resultados pueda validar los objetivos de partida que propone el presente estudio, aportando información verídica y clara. Todo ello, de una manera que permita acoger toda la casuística posible y siendo excluyentes unas respuestas de otras, por lo que cada cuestión debe responderse siempre con solo una de las opciones planteadas.

En ninguno de los casos, se le obliga al encuestado a añadir información sensible relacionada con la empresa, proyectos o con información personal, evitando preguntas indiscretas y buscando contestaciones directas e inequívocas a preguntas sencillas redactadas en un lenguaje fácilmente comprensible para el público objetivo. De igual forma, se le indica al usuario que la información que aportada es tratada de manera confidencial y solo para los fines de la investigación.

En muchas de estas cuestiones, se ha planteado una escala tipo Likert²⁰⁴ con un rango definido de 5 posibles puntuaciones con el objetivo de simplificar la elección, pero permitiendo respuestas con matices suficientes. Esta escala asigna un valor de 1 para respuestas totalmente en desacuerdo, y llega hasta un valor máximo de 5 para cuestiones con las que se está totalmente de acuerdo. Se establece el valor de 3 para una categoría neutra en la que el usuario estaría indiferente ante la cuestión. En otros casos, las preguntas tienen unos rangos definidos para identificar el nivel de importancia o de conocimiento que posee el encuestado en referencia a diferentes temáticas. Antes del envío a los participantes, se procede a la realización de la validación del cuestionario a partir de un pretest en el que se solicita a una comisión de expertos altamente cualificados y competentes en la materia para que evalúen si la calidad y la procedencia de las preguntas son comprensibles, si se han

²⁰⁴ Artículo "A technique for the measurement of attitudes" de Rensis Likert, publicado en 1932 en Archives of Psychology.

contemplado todas las posibles variables y si todas las escalas y rangos de respuesta establecidos son aceptables.

La encuesta se abrió a los participantes a finales de 2023 y se cerró a principios de 2024, permaneciendo activa durante unos 4 meses. Esto permitió aumentar considerablemente el número de respuestas, ya que los usuarios contaron con un largo margen de tiempo para responder al formulario.

3.2.2. Resultados de la encuesta

Una vez recogida y analizada la información de los individuos integrantes de la muestra, se organizan y presentan los resultados obtenidos de manera conjunta. Al haber establecido diferentes configuraciones dentro del formulario, no es necesario codificar a cada usuario ni revisar posibles respuestas erróneas o inconsistentes, ya que la propia plataforma gestiona el tratamiento de la información de cada encuestado y de manera automática genera una serie de representaciones gráficas con el conjunto de resultados obtenidos.

En cuanto al nivel de participación, cumplió con las premisas iniciales en cuanto a encuestados, incluyendo la participación de más de 40 empresas diferentes del sector de la edificación, lo cual era un requisito fundamental para el estudio. De estas empresas, la gran mayoría fueron ingenierías (51%), siguiendo consultoras (23,5%), centros de enseñanza o investigación (9,8%) y constructoras (7,8%).

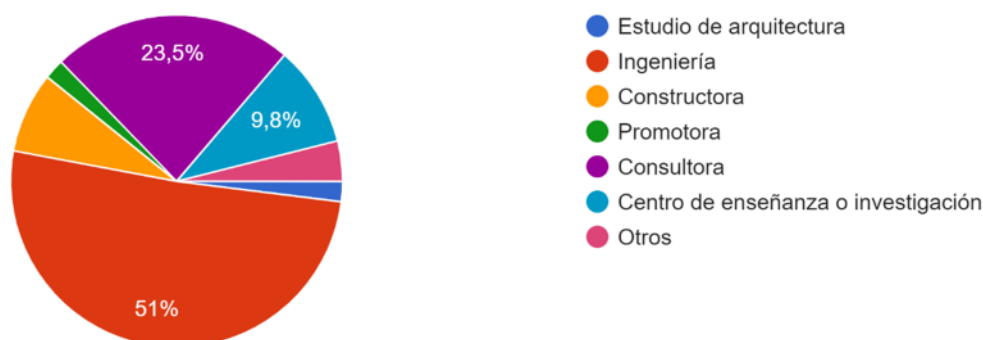


Figura 3.2.2. Encuesta: Tipos de empresas

En relación a los años de experiencia de los encuestados, casi la mitad de los mismos (49%) se encontraban entre 5 y 10 años de experiencia y más de un tercio (35,3%) tenían más de 10 años.

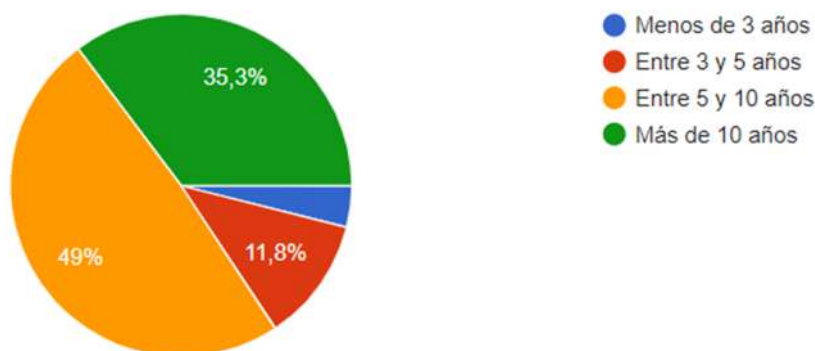


Figura 3.2.3. Encuesta: Tramos de años de experiencia

En referencia al rol de los encuestados, existen múltiples perfiles dependiendo del tipo de empresa, los niveles de experiencia y de los grados de cada puesto. Sin embargo, la gran mayoría de los encuestados (58,8%) tienen perfiles de responsables, directivos o coordinadores BIM, lo cual favorece una visión holística en el uso de la metodología BIM y de los proyectos que tiene la empresa desarrollados y los que tiene en activo.

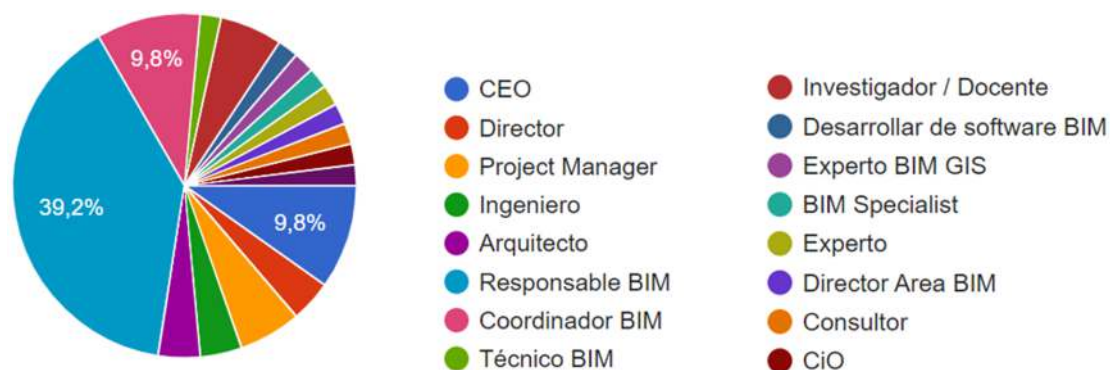


Figura 3.2.4. Encuesta: Tipos de perfiles profesionales

Las cuestiones del primer bloque relativo a la planificación inicial BIM, hacen referencia al nivel de implicación y colaboración de todos los stakeholders desde las fases iniciales del proyecto de construcción, estableciendo niveles de información necesarios, procedimientos de intercambio y otros aspectos relevantes para continuidad de la coordinación y gestión del activo.

La primera cuestión de este bloque tiene relación con el cumplimiento de la norma ISO 19650 y del establecimiento de diferentes documentos BIM como el EIR (Exchange Information Requirements) o el AIR (Asset Information Requirements) para definir los requisitos de información y de intercambio entre las partes interesadas. El siguiente gráfico muestra como solo el 17,6% de los encuestados trabajan la totalidad de los proyectos con una visión global del ciclo de vida del activo desde un inicio de manera planificada. Cabe destacar que este tipo de documentos deben ser generados por el cliente o administración interesada en construir el activo y deben ser ellos quienes fomenten la colaboración temprana entre todas las partes.

¿Es usual que en los proyectos en los que trabaja se exija desde las primeras fases de proyecto el cumplimiento del EIR, AIR u otros requisitos de información de construcción?

51 respuestas

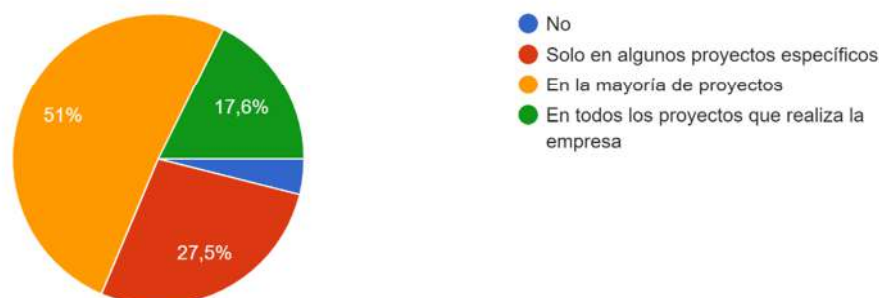


Figura 3.2.5. Encuesta: Exigencia de requisitos de información

La siguiente pregunta se refiere a los niveles de información gráfica y no gráfica que contendrán los modelos de información BIM. Esto incluye niveles de definición, clases y tipos de objetos y sistemas, grupos de atributos, tipos de datos y valores permitidos. Es un punto fundamental ya que, si estos niveles se establecen desde fases tempranas, no existen ambigüedades durante el desarrollo del proyecto u obra y cada equipo sabe concretamente hasta donde tiene que llegar en cada hito de entrega, generando la información justa y adecuada en cada momento. Los resultados muestran que algo más de un tercio de los encuestados (37,3%) trabajan en proyectos que tienen un planteamiento conjunto de todo el ciclo de vida desde el inicio, otro tercio (33,3%) solo trabaja la continuidad en las fases de diseño y construcción. El resto de los participantes (29,4%) no contemplan el uso de estos niveles, lo definen de una manera genérica o solo establecen los requisitos para la fase de diseño.

¿Se establecen desde fases tempranas los Niveles de información (LOD / LOIN) de las entidades BIM?

51 respuestas

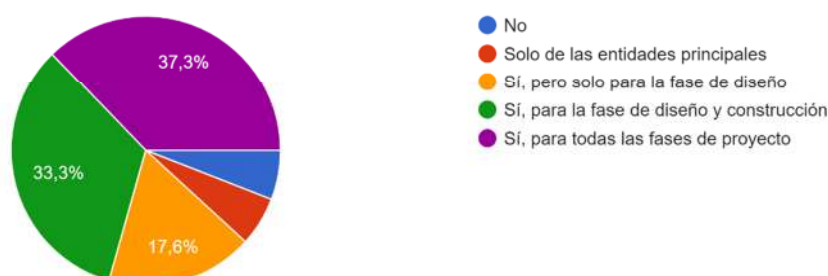


Figura 3.2.6. Encuesta: Definición de LOIN desde fases iniciales

La última cuestión del bloque abarca el grado de importancia que percibe cada usuario en relación a diferentes aspectos que deben tenerse en cuenta a la hora de planificar un proyecto que cuenta con la aplicación de la metodología BIM, definiendo para cada uno de los aspectos un valor de rango que puede encontrarse entre “No importante” o llegar a ser “Imprescindible”, pasando en medio por diferentes grados de importancia. En el resultado obtenido se observa que los principales aspectos a tener en cuenta son la estandarización, la interoperabilidad y la centralización de la información, teniendo marcadas la mayoría de respuestas (más del 90%) como “Imprescindibles” o de “Importancia alta”.

De los siguientes puntos, ¿Cuál es el nivel importancia que considera para una buena planificación y gestión de la información BIM? Responda una opción por fila

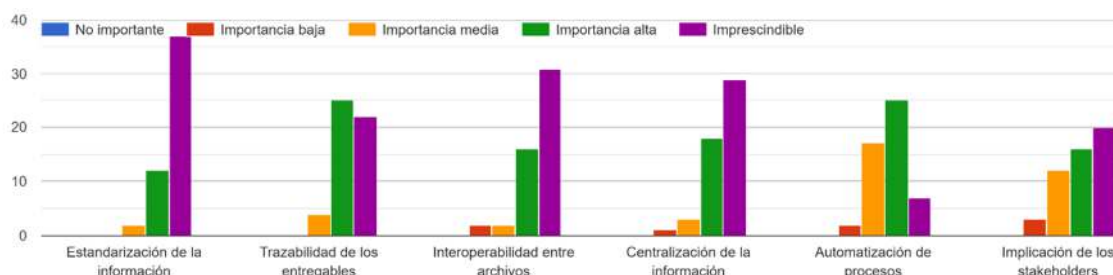


Figura 3.2.7. Encuesta: Aspectos relevantes en la gestión BIM

El segundo bloque de cuestiones relativo al trabajo con atributos y propiedades, se centra en el trabajo de los datos de construcción, teniendo en cuenta recursos y herramientas que permitan el desarrollo del proyecto de una manera controlada y estructurada.

En primer lugar, se pregunta acerca de plantillas preconfiguradas que contengan las definiciones de los atributos necesarios. Aunque la gran mayoría de encuestados responden de forma afirmativa, destaca un porcentaje considerable (17,6%) de los usuarios que no trabajan con este tipo de archivos. Y menos de la mitad (39,2%) son los que realmente están involucrados en proyectos que tienen planificada una continuidad digital para el activo de edificación.

¿Trabaja con plantillas tipo "Parámetros compartidos" o similares?

51 respuestas



Figura 3.2.8. Encuesta: Plantillas de configuración de atributos

En segundo lugar, se consulta sobre bibliotecas de objetos BIM, las cuales se utilicen para diferentes proyectos y estén ya preparadas para ser incluidas en los modelos BIM. En este caso, el porcentaje de usuarios que no tiene estas bibliotecas es menor (13,7%), mientras que la amplia mayoría sí que las tiene y, al menos, contemplan su uso para la fase de diseño.

¿Trabaja con bibliotecas de objetos BIM personalizadas y adaptadas a los proyectos que realiza su empresa?

51 respuestas



Figura 3.2.9. Encuesta: Bibliotecas de objetos BIM

En tercer lugar, se hace referencia al control de los requisitos de información dentro del Plan de Ejecución BIM, el cual es una evolución de los documentos iniciales generados por el cliente y debe ser actualizado periódicamente conforme al avance

de las fases del proyecto. Del resultado obtenido, se extrae que gran parte de los participantes (39,2%) no consideran que exista una estructura clara de información dentro de este documento y que los requisitos se especifican de forma muy genérica.

Por norma general, ¿en el BEP se establecen de forma correcta las propiedades, elementos, unidades, tipos de datos, etc. que requiere el proyecto?

51 respuestas

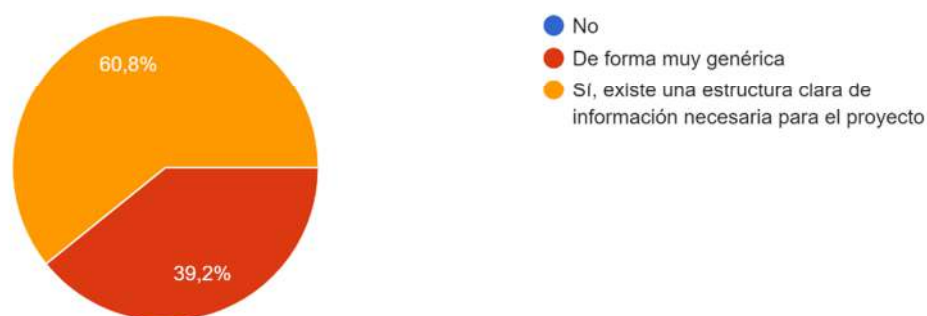


Figura 3.2.10. Encuesta: Definición de datos y propiedades en el BEP

Continuando con el tercer bloque del cuestionario, este hace referencia a los formatos abiertos, los cuales se basan en archivos interoperables que no dependen de casas comerciales de software o plataformas y que pueden ser visualizados o analizados a partir de diferentes herramientas BIM. El objeto de este bloque es conocer el estado de implantación de estos formatos abiertos dentro de las empresas nacionales a partir del punto de vista de los expertos. Al considerarse el formato IFC como el más importante de todos ya que se trata del contenedor de información de la construcción, se le da una mayor presencia en las cuestiones del bloque.

La primera cuestión aborda el nivel de exigencia a la hora de generar entregables con formato IFC en proyectos de edificación. Estos formatos son muy importantes ya que no dependen de versiones de software nativos y contienen un esquema estandarizado e interoperable para todos los agentes intervinientes del proyecto. La gran mayoría de respuestas indican que sí se solicita el formato, sin embargo, solo algo más de la mitad de los encuestados (52,9%) apuntan a que se trata de una exigencia para la totalidad de los proyectos en los que se trabaja en la actualidad.

En los proyectos que trabaja actualmente, ¿Se exige que los entregables se generen con formatos abiertos tipo IFC?

51 respuestas

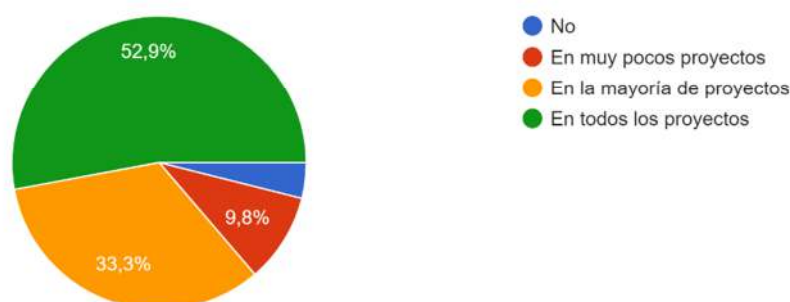


Figura 3.2.11. Encuesta: Exigencia de entrega en formatos abiertos

En relación a la estructura y contenido del esquema IFC, la siguiente pregunta busca conocer el grado de fidelidad que percibe el encuestado a la hora de generar estos entregables. Se muestra en el siguiente gráfico que más de la mitad de los participantes (52,9%) no cuentan con plantillas o preconfiguraciones para la generación de estos archivos y que un porcentaje considerable (23,5%), solo realiza configuraciones básicas para la creación o exportación de los archivos.

En cuanto al trabajo con modelos IFC, ¿Se respetan los conjuntos de propiedades, nombres de propiedades y resto de información del estándar?

51 respuestas

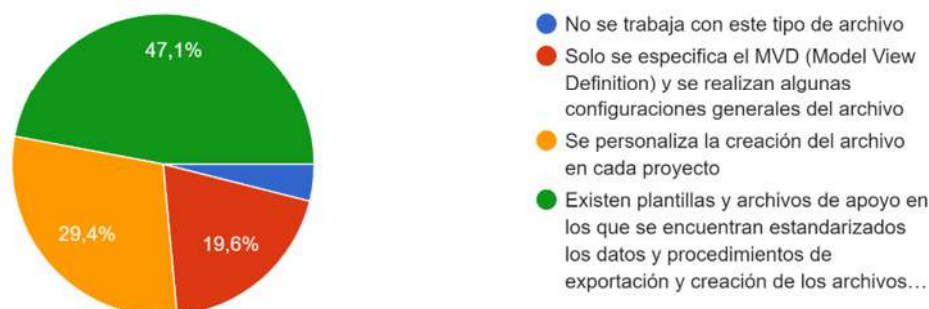


Figura 3.2.12. Encuesta: Configuración del esquema IFC

Para finalizar el bloque, se solicitó a los encuestados que indicaran en una matriz de preguntas con diferentes grados, el conocimiento que poseían acerca de los estándares y recursos openBIM que actualmente existen en el mercado. La gran mayoría (más del 80%) respondió tener conocimientos “Intermedios” o “Avanzados” en cuanto al formato IFC y al BCF, mientras que los menos conocidos, con conocimientos fundamentalmente “Básicos”, fueron el COBie y el IDS.

De los siguientes estándares y recursos BIM, indique el nivel de conocimiento que posee. Responda una opción por fila

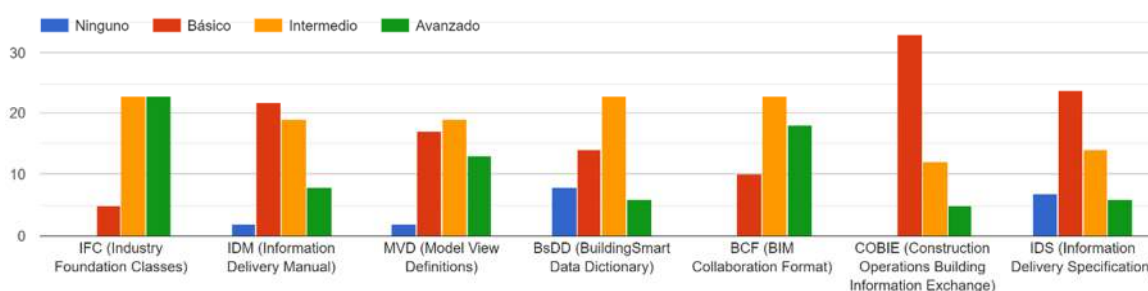


Figura 3.2.13. Encuesta: Conocimientos en formatos openBIM

El cuarto bloque se compone de una serie de cuestiones relacionadas con controles de calidad aplicados a los entregables BIM. Estos controles tienen como objetivo principal garantizar que los requisitos iniciales establecidos por el cliente han sido satisfechos y que el nivel de información de los entregables es el que se ha solicitado.

En cuanto a la existencia de un sistema de control de calidad dentro de la empresa para llevar a cabo las revisiones, más de un tercio de los encuestados (37,3%) indicó

que no existe un procedimiento global de empresa, sino que cada equipo aplica sus propios criterios o incluso que no existe un proceso normalizado.

¿Tiene establecido un sistema de control de calidad de los entregables BIM que se generan?

51 respuestas

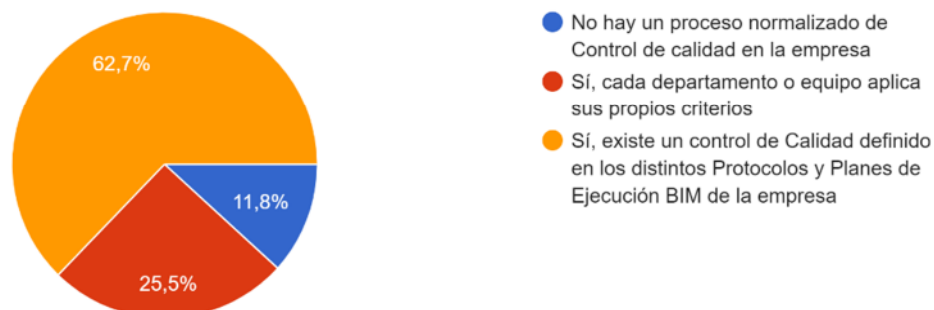


Figura 3.2.14. Encuesta: Controles de calidad en entregables BIM

En anteriores cuestiones se solicitaba al usuario conocer si se exigían los Niveles de información para el proyecto, sin embargo, en esta cuestión, la pregunta se enfoca a saber si esos niveles posteriormente y antes de la entrega son validados siendo unos criterios de calidad. Solo un pequeño porcentaje (15,7%) respondió que no se realizaban estos chequeos y, por otro lado, un tercio de los encuestados (33,3%) contestó que no solo los realizaban, sino que también se valían de rutinas y procesos automáticos o semiautomáticos para realizar las comprobaciones.

¿Se realizan controles de calidad relativos al Nivel de información (LOD / LOIN) de las entidades BIM?

51 respuestas

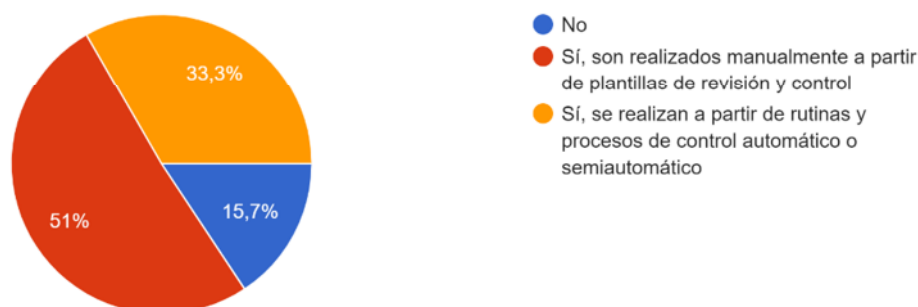


Figura 3.2.15. Encuesta: Controles de calidad en LOIN

Para cerrar el bloque, se solicitó a los usuarios que indicaran si también se realizaban en los proyectos controles relativos a la nomenclatura de los archivos y elementos BIM. De nuevo, un pequeño porcentaje (11,8%) respondió que no se realizaban, mientras que casi la mitad (43,1%) afirmó que los generaban a partir de procesos automáticos o semiautomáticos.

¿Se realizan controles de calidad relativos a la nomenclatura de archivos y elementos?

51 respuestas

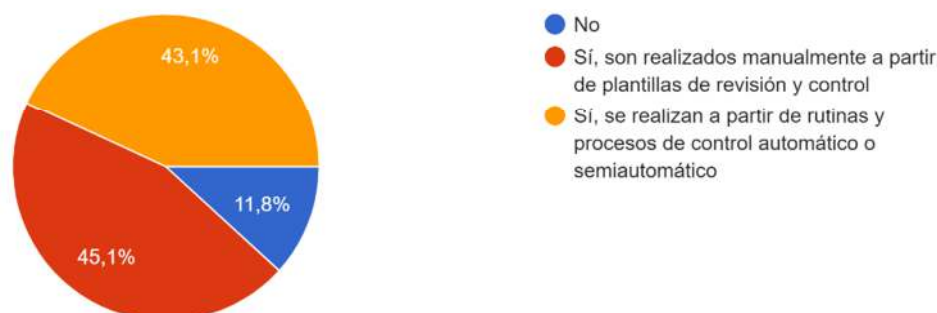


Figura 3.2.16. Encuesta: Controles de calidad en nomenclatura

Para finalizar la encuesta, se generó un último bloque denominado Futuro del BIM, en el que las cuestiones se enfocaron en conocer la percepción actual de los expertos en cuanto al rumbo que tomará la metodología y a las posibilidades que tiene la misma de conectividad o interacción con otras herramientas o tecnologías emergentes. Todas las preguntas de este bloque utilizan un rango que empieza en 0, para opiniones “Muy en desacuerdo” y termina en 5, para opiniones “Muy de acuerdo”.

La primera cuestión del bloque está relacionada con la percepción del experto en cuanto a la integración del BIM con otras tecnologías. En este caso, el gráfico muestra que la tendencia de los resultados se acerca a percepciones que están “De acuerdo” o “Muy de acuerdo”, sumando entre las dos un 90,2% de las opiniones.

¿Considera que la metodología BIM tiende a integrarse y colaborar cada vez más con otras tecnologías / herramientas emergentes?

51 respuestas

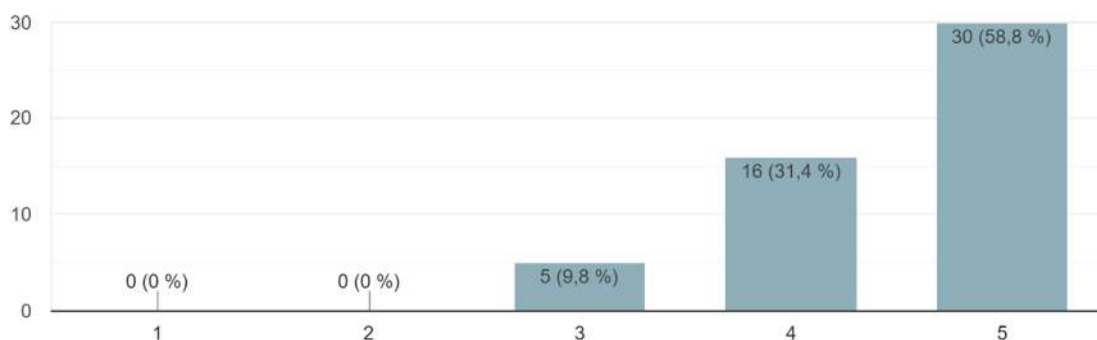


Figura 3.2.17. Encuesta: Integración BIM con otras tecnologías

La siguiente pregunta se enfoca en la interacción o conectividad entre los modelos de información BIM y otras bases de datos externas. En este caso, las opiniones que están de acuerdo superan el 95% de los resultados.

¿Considera útil la interacción entre bases de datos BIM y otras bases de datos externas?

51 respuestas

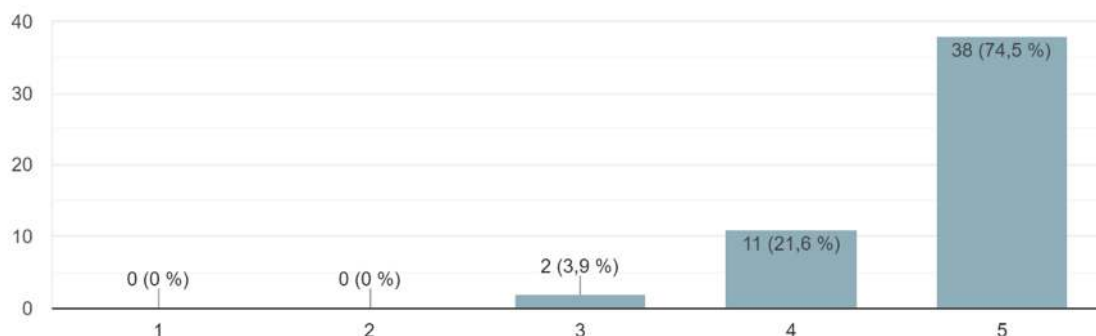


Figura 3.2.18. Encuesta: Interacción de BIM con BBDD

En cuanto al uso de los formatos abiertos como el IFC para una mejora de la gestión de los proyectos, los usuarios han respondido con opiniones más repartidas entre las diferentes opciones. La mayoría (68,6%) está “Muy de acuerdo” en su uso, pero existen algunos encuestados que no lo consideran importante o que tiene una importancia media (11,7%).

¿Considera importante el uso de formatos abiertos como el IFC para una mejor gestión en los proyectos del sector AECO?

51 respuestas

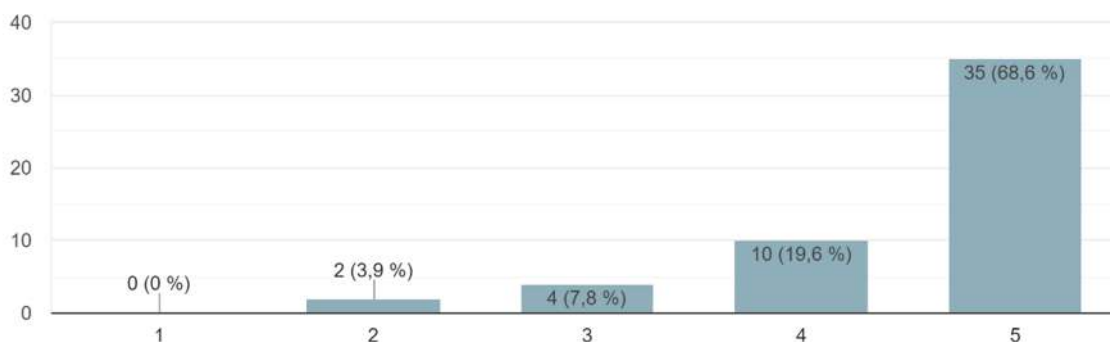


Figura 3.2.19. Encuesta: Uso de formatos abiertos como IFC

La siguiente pregunta, aborda el tema de la trazabilidad de la información durante todo el ciclo de vida del activo. Este punto sí que se percibe de forma casi unánime como fundamental, ya que las respuestas que están “De acuerdo” o “Muy de acuerdo” son del 98%.

¿Considera importante la trazabilidad de los datos a lo largo del ciclo de vida de un activo de construcción?

51 respuestas

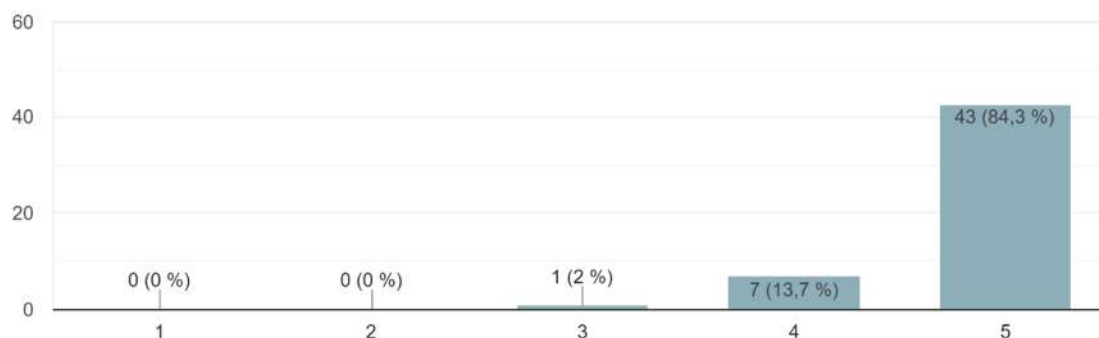


Figura 3.2.20. Encuesta: Trazabilidad de los datos en el ciclo de vida

Por último, se hace referencia a la creación y el uso de una base de datos centralizada que permita gestionar toda la información conjunta de los activos de construcción, relacionando los datos de los modelos BIM con otros relativos a proveedores, mantenimiento, fichas técnicas, seguridad y salud y otros datos importantes del ciclo de vida. En este caso, los resultados muestran que casi tres cuartos de los encuestados (74,5%) están “Muy de acuerdo” y casi el resto de participantes (17,6%), están “De acuerdo” con la propuesta.

¿Considera provechoso el uso de una base centralizada que pueda integrar los datos BIM de los activos construidos con otros datos relacionados c... proveedores, contratación, mantenimiento, etc.?

51 respuestas

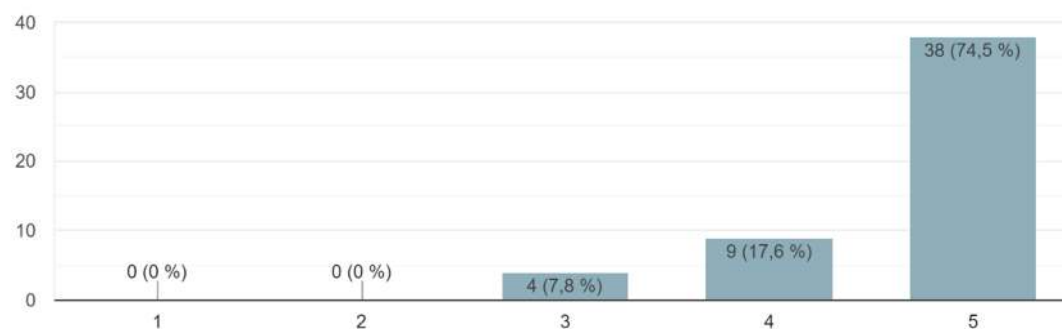


Figura 3.2.21. Encuesta: Conectividad de los datos de la construcción

3.2.3. Conclusiones de la encuesta

El análisis del cuestionario ha permitido obtener una visión detallada y representativa del estado actual de la metodología BIM en el sector de la edificación. En general, los resultados evidencian un grado de madurez intermedio en la adopción de BIM, con un conocimiento generalizado de sus principios y herramientas, pero aún con áreas clave por consolidar o implementar.

Entre los hallazgos más destacados, se encuentra la importancia que otorgan los profesionales a aspectos como la estandarización, la interoperabilidad y la centralización de la información, considerados elementos imprescindibles para una correcta gestión BIM. Sin embargo, aspectos como la planificación inicial aún presenta carencias, especialmente en la implementación sistemática de requisitos de información y en la definición de niveles de información desde las fases tempranas del proyecto.

En cuanto al trabajo con datos, aunque es notable la existencia de plantillas y bibliotecas BIM en muchas empresas, la continuidad digital no está plenamente garantizada, y los Planes de Ejecución BIM en muchos casos carecen de una estructura clara y concreta. Del mismo modo, los resultados reflejan una implementación parcial de los formatos abiertos como el IFC, los cuales, a pesar de ser reconocidos como necesarios, no siempre se utilizan con estándares o procesos de calidad consistentes.

Los controles de calidad sobre los entregables BIM también representan un punto de mejora importante, ya que una parte significativa de los participantes manifiesta la ausencia de procedimientos normalizados o el uso limitado de herramientas de validación automatizada.

Por último, en cuanto a la visión de futuro, los expertos del sector muestran una fuerte confianza en la evolución de BIM hacia una integración más amplia con otras tecnologías, bases de datos externas y sistemas de gestión centralizada, destacando la trazabilidad de la información y la gestión de datos durante todo el ciclo de vida como elementos estratégicos.

3.3. Propuesta de sistematización

Con base en el análisis realizado en los apartados previos, correspondientes a los antecedentes, el estado del arte y la búsqueda científica, se propone una solución que se enfoca en conseguir generar una sistematización que permita una mayor colaboración BIM entre todos los agentes intervinientes del proyecto, una gestión de los datos más estructurada y controlada y una agrupación de todos los datos del proyecto en una fuente centralizada de información. De esta manera, se genera un nuevo procedimiento cuyo objetivo principal es estandarizar la forma de proceder con las tareas y actividades relacionadas con la construcción y administración de datos desde un enfoque sistemático y estructurado, basado en un conjunto de reglas y técnicas que guían el proceso de trabajo durante las diferentes fases del activo de construcción. Además, se prioriza el establecimiento de un enfoque flexible y agnóstico, que se pueda adaptar a diferentes condiciones o entornos, pero que siempre sea gestionado a partir de los mismos procesos y configuraciones de datos. De este modo, se establece un marco general que define y optimiza el trabajo de construcción de información de activos en función de un marco general de aplicación.

3.3.1. Base de datos global

La construcción de una base de datos global se plantea como el núcleo estructural de la sistematización propuesta, funcionando como repositorio relacional unificado para gestionar la totalidad de los datos organizativos y de proyectos en el marco de operaciones vinculadas a la edificación y a modelos BIM. Esta base de datos relacional gestiona el almacenamiento centralizado de los datos, habilitando a la vez una interoperabilidad entre sistemas y asegurando la trazabilidad, actualización y explotación de los datos a lo largo del ciclo de vida de los activos construidos.

La base de datos se articula mediante un esquema relacional compuesto por múltiples entidades normalizadas, diseñadas para reflejar la granularidad y jerarquía de la información técnica, contractual y operativa que la organización gestiona. Las tablas principales están agrupadas por dominios funcionales que representan diferentes dimensiones del negocio. Esta organización permite mantener una estructura modular, donde cada nuevo proyecto se incorpora mediante esquemas específicos asociados a contenedores de información únicos por activo, trabajado desde un sistema de gestión de bases de datos.

Los datos generales de la organización, tales como los requisitos de intercambio de información, matrices de responsabilidades, sistemas de clasificación, taxonomías de objetos y conjuntos de propiedades, son previamente descompuestos, auditados y reestructurados en tablas relacionales. Este análisis pormenorizado asegura la integridad semántica y técnica de los datos, permitiendo su integración mediante claves primarias y foráneas con los modelos de datos provenientes de proyectos específicos. Asimismo, se implementan reglas de validación automatizadas para verificar la conformidad y consistencia de estos datos antes de ser integrados.

En cuanto a los datos de proyectos, la base de datos está configurada para recibir y transformar la información contenida en modelos openBIM mediante procedimientos de extracción, transformación y carga de datos, a partir de configuraciones específicas por tipo de activo, donde se definen reglas de mapeo entre entidades IFC

y tablas relacionales internas. Además, se generan otros procedimientos que permiten integrar otros datos del proyecto relacionados con registros de entregables, planificaciones de tiempos y costes y otros registros como incidencias, actas de reunión, informes de transmisión o revisiones activas del proyecto.

Esta base de datos relacional no pretende sustituir los Entornos Comunes de Datos (CDE) existentes, sino complementarlos, integrándose como una capa de sistematización que permite conectar verticalmente la información operativa de los proyectos con los sistemas internos de toma de decisiones de la organización y facilitando así una analítica transversal que permite, por ejemplo, realizar consultas predictivas, comparar rendimientos entre activos similares o generar reportes históricos de mantenimiento vinculados directamente con los modelos digitales.

La actualización periódica de los datos ejecuta en intervalos definidos para capturar modificaciones de los modelos BIM y actualizar las entidades relacionadas en la base de datos. También se establecen mecanismos de versionado lógico por activo y por fase, lo que permite mantener un histórico de evolución del dato.

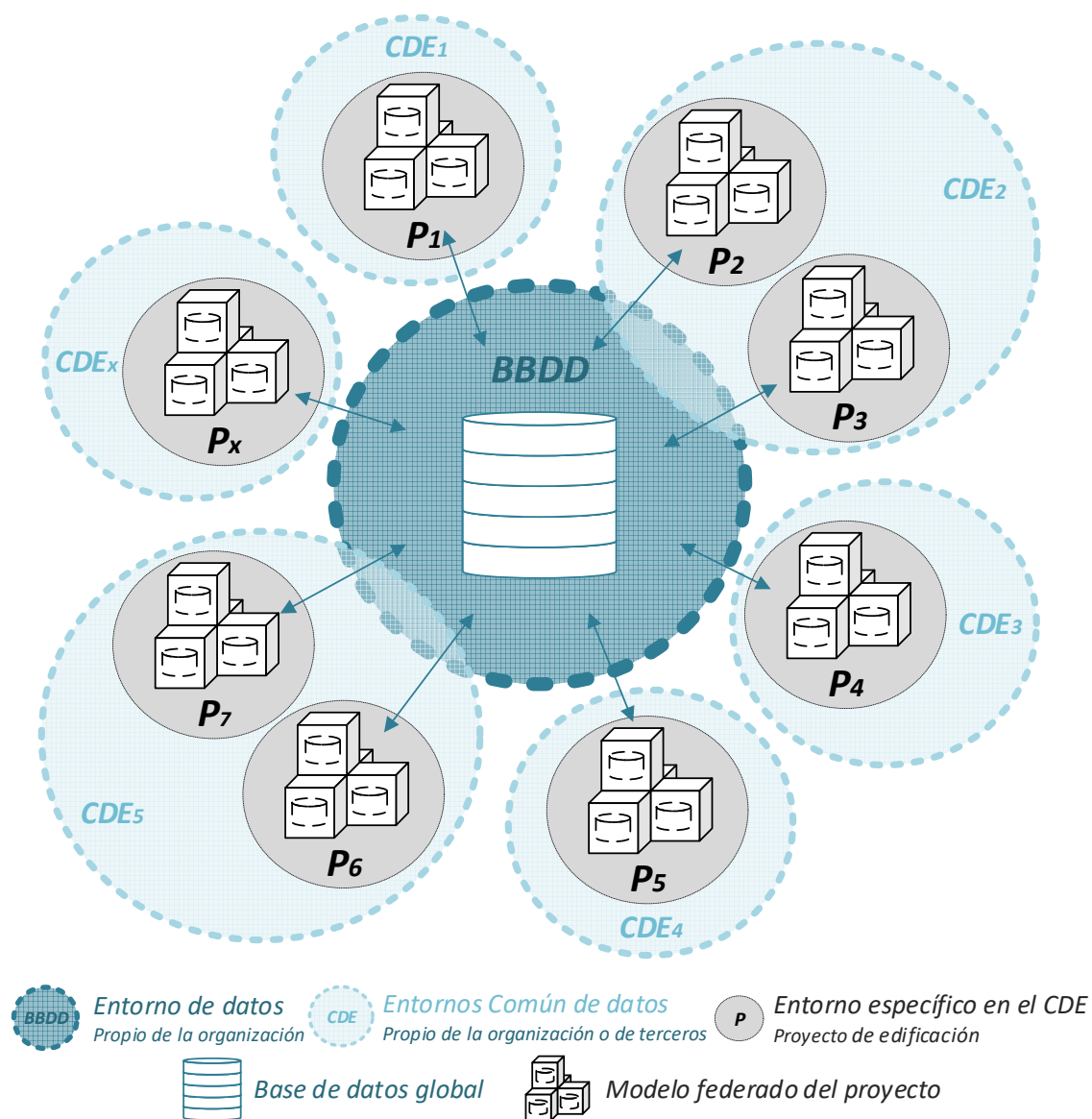


Figura 3.3.1. Relación entre BBDD y proyectos BIM

3.3.2. Ejes estructurantes de datos

En el marco de esta base de datos global, la estructuración de la información en función de tres ejes fundamentales, como son las fases del proyecto, los usos de la información y las disciplinas técnicas, se configura como un principio clave para asegurar la eficiencia, la trazabilidad y la capacidad analítica del sistema. Las fases del proyecto actúan como un criterio temporal que permite organizar los datos según su generación, vigencia y aplicabilidad, estableciendo una lógica evolutiva que favorece la validación progresiva y la gestión por hitos.

Los usos de la información, por su parte, introducen una dimensión funcional que clasifica los datos según su finalidad concreta, como pueden ser la coordinación entre disciplinas, la estimación de costes, la planificación o la preparación para el mantenimiento operativo del activo construido. Esta categorización funcional permite que los sistemas de consulta y exportación estén orientados a las necesidades específicas de los usuarios y de las herramientas con las que interactúan, alineándose siempre con los requisitos específicos para cada proyecto.

A estos dos ejes se suma la disciplina técnica como una dimensión organizativa esencial en el contexto de proyectos multidisciplinarios, donde intervienen diversas áreas de construcción. Su inclusión permite segmentar la información según un ámbito de responsabilidad, lo que facilita la gestión descentralizada de los datos, el control de versiones específicas y la integración federada de modelos parciales.

Esta triple estructuración optimiza los mecanismos de filtrado, búsqueda y exportación dentro del sistema a la vez que permite implementar estrategias de acceso diferenciadas, auditorías más precisas y flujos de trabajo automatizados que responden a la complejidad creciente de los entornos colaborativos. En consecuencia, el diseño del modelo de datos contempla estos tres ejes como pilares fundamentales para lograr una gestión coherente, interoperable y orientada al ciclo de vida completo del proyecto.

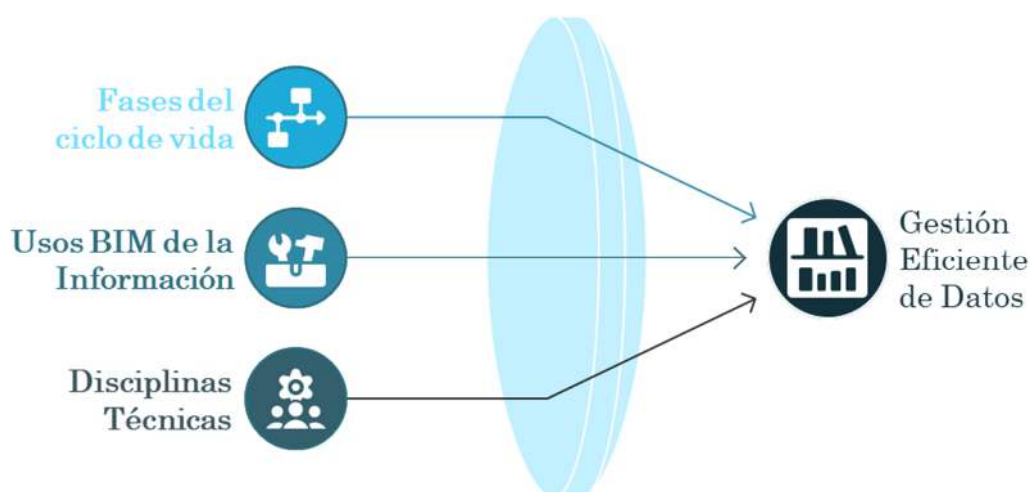


Figura 3.3.2. Ejes estructurantes de datos de construcción

En el contexto de la estructuración de estos datos, los atributos que deben ser extraídos desde la base de datos global no dependen exclusivamente de combinaciones entre disciplinas, fases y usos BIM, sino que cada uno de estos ejes

genera su propio conjunto de requisitos de información. Estos requisitos, a su vez, definen propiedades específicas que deben estar disponibles en los modelos o bases asociadas. Como resultado, se construyen tres conjuntos independientes de atributos, a partir de la extracción de forma separada, normalizando para eliminar duplicados, y consolidando finalmente la información en una tabla de atributos para el proyecto concreto, que se materializa en un archivo exportado, como una hoja de cálculo, donde se integran todas las propiedades requeridas.

$$A = \text{set} (f (R_D) \cup f (R_F) \cup f (R_U))$$

A: Conjunto final de atributos únicos integrados en la tabla de proyecto.

R_D: Requisitos definidos por las disciplinas técnicas.

R_F: Requisitos definidos por las fases del proyecto.

R_U: Requisitos definidos por los usos BIM.

f (R_x): Función de extracción de atributos a partir de los requisitos R_x,

siendo $x \in \{D, F, U\}$.

set(·): Unión sin duplicados entre los atributos extraídos por cada eje.

Un aspecto clave de esta estructura es que cada propiedad está asociada explícitamente a una clase de objeto del modelo BIM o del sistema de información relacional utilizado. Esto significa que los atributos no se asignan de forma genérica, sino que cada uno contiene un campo adicional que especifica a qué clase constructiva, funcional o espacial debe ser aplicado. Estas clases ya están previamente definidas en la estructura semántica del modelo, y las relaciones entre objetos y propiedades están establecidas en la ontología o esquema de datos del proyecto. Esta vinculación permite que los atributos extraídos se integren correctamente en los modelos BIM o en las tablas del sistema, asegurando la coherencia semántica de la información. De este modo, el sistema no sólo garantiza una cobertura completa de requisitos por eje, sino que además mantiene la integridad estructural al posicionar cada atributo exactamente en la clase de objeto correspondiente.

Un ejemplo básico de aplicación sería el siguiente, en el que se van integrando los atributos requeridos según el eje de origen y la especificación del eje, definiendo también las clases de aplicación para cada atributo extraído. Inicialmente se obtienen los atributos de cada eje por separado, luego se realiza una unificación de todos los atributos dentro de una misma tabla de datos.

Nº	Eje de origen	Especificación del eje	Atributo	Clases de objeto
1	Disciplina	Arquitectura	ID	Todos los objetos
2	Disciplina	Arquitectura	Nombre	Todos los objetos
3	Disciplina	Arquitectura	Material	Muro
4	Disciplina	Estructuras	TipoElemento	Pilar, Viga

Nº	Eje de origen	Especificación del eje	Atributo	Clases de objeto
5	Disciplina	Arquitectura	Alto	Puerta, Ventana
6	Fase	Construcción	FechaInicio	Todos los objetos
7	Fase	Construcción	FechaFin	Todos los objetos
8	Fase	O&M	Estado	Todos los objetos
9	Uso BIM	Mantenimiento	FechaInspección	Equipo, Sistema
10	Uso BIM	Coordinación	Nivel	Espacio, Zona
11	Uso BIM	Gestión de activos	Proveedor	Sistema MEP

Tabla 3.3.5. Ejemplo de requisitos de información

En el desarrollo de los casos de uso del presente estudio, se especifican otras casuísticas que pueden estar relacionadas con el idioma, restricciones o requerimientos de los softwares o de formatos de importación de archivos y de segregación de modelos, dependiendo de tiempos y especialidades técnicas.

3.3.3. Conectividad de datos BIM

La conectividad de datos entre los entornos BIM y las bases de datos habilita el intercambio bidireccional, automatizado y semánticamente consistente de información entre los modelos digitales y los sistemas relacionales de gestión de datos.

La estrategia de conectividad adoptada requiere, como punto fundamental, un estudio profundo y detallado de los formatos de intercambio y estructuras de datos que permite gestionar cada software BIM utilizado dentro del presente estudio. Esta tarea es crítica, dado que los modelos generados en plataformas como Autodesk Revit, BIMCollab o Navisworks presentan notables diferencias en cuanto a sus capacidades de exportación, estructuración interna, codificación semántica y soporte para atributos personalizados. Así, se identifican no solo los formatos de salida, sino también las APIs disponibles, los mecanismos de automatización soportados y la capacidad de personalización de plantillas y esquemas de datos.

En este contexto, se hace indispensable la definición y desarrollo de un conjunto de herramientas técnicas que garanticen la consistencia de la conexión, incluyendo:

- Plantillas de exportación preconfiguradas que aseguren la estructura coherente y repetible de los datos exportados desde los modelos BIM hacia el sistema de base de datos.
- Esquemas de mapeo paramétrico, que vinculan entidades del modelo con tablas y campos específicos en la base de datos relacional.
- Automatizaciones y scripts desarrollados para ejecutar procesos de extracción, transformación y carga de manera periódica.
- Soluciones intermedias de gestión de datos, que sirven como capa de abstracción entre el software de modelado y la base de datos, traduciendo

estructuras de datos complejas a modelos relacionales sin pérdida de semántica ni granularidad.

Todo este proceso de integración exige un conocimiento preciso del funcionamiento interno de cada herramienta BIM, especialmente en lo que respecta a la definición de parámetros, gestión de propiedades personalizadas, jerarquías espaciales, asignación de sistemas de clasificación y la implementación de metadatos estructurados. El objetivo final es construir una conectividad robusta y flexible que permita tanto la consulta directa desde la base de datos, como la posibilidad de inyectar datos desde sistemas externos hacia los modelos BIM.

Maria DB

En el marco de esta integración de entornos BIM con bases de datos relacionales, resulta esencial comprender los formatos de archivo propios de los sistemas, así como las posibilidades que ofrecen para la importación y exportación de datos estructurados. La elección adecuada del formato de intercambio condiciona no solo la eficiencia del flujo de información, sino también la capacidad de conservar la integridad relacional de los datos provenientes de los modelos BIM.

El sistema de MariaDB utiliza de forma nativa archivos de almacenamiento binario que no son manipulables directamente por el usuario, sino gestionados internamente por el motor de la base de datos. Estos incluyen extensiones como *.frm* para el esquema de tablas, *.ibd* para los datos y los índices cuando se utiliza el motor de almacenamiento InnoDB, y *.ibdata* como fichero compartido de espacio de tablas en configuraciones que así lo requieren. No obstante, en el contexto de interoperabilidad con aplicaciones externas, el trabajo cotidiano con estos gestores se centra principalmente en formatos de importación y exportación legibles y estructurados, que permiten la conexión con hojas de cálculo.

Entre los formatos más comunes para la exportación de datos se encuentra el *.SQL*, que contiene un volcado de la base de datos o de tablas específicas, incluyendo tanto las sentencias *DDL* (Data Definition Language) para la recreación de la estructura como las sentencias *DML* (Data Manipulation Language) para la inserción de datos. Este formato es especialmente útil para respaldos, replicación o migración de entornos. Por su parte, los formatos *.CSV* y *.TSV* permiten una exportación directa de conjuntos de datos estructurados, facilitando su edición en hojas de cálculo y su uso posterior en herramientas BIM para la sincronización de datos.

Asimismo, los motores de base de datos permiten la importación de datos desde archivos externos mediante instrucciones como *LOAD DATA INFILE*, lo que permite integrar masivamente información de tipo numérico, categórico o jerárquico en tablas previamente definidas. Esta funcionalidad resulta clave en flujos donde los requisitos BIM se gestionan en matrices externas (por ejemplo, Excel o Google Sheets) y deben luego consolidarse en una base de datos para su trazabilidad, consulta o vinculación con los objetos del modelo digital. También es posible, mediante scripts automatizados o herramientas intermedias, importar archivos *JSON* o *XML* estructurados, lo que permite representar jerarquías de atributos más complejas o estructuras anidadas, manteniendo su lógica relacional durante la integración con modelos BIM.

Autodesk Revit

En el caso de los procesos BIM basados en Autodesk Revit, los distintos formatos de importación y exportación cumplen funciones complementarias en relación con la documentación gráfica, la colaboración interdisciplinar, la visualización y, especialmente, la estructuración e interoperabilidad de los datos. La comprensión detallada de las capacidades y limitaciones de estos formatos resulta esencial para diseñar flujos de trabajo coherentes, asegurar la trazabilidad de la información y permitir su conexión con bases de datos relacionales.

Los formatos nativos del programa constituyen el núcleo del entorno de modelado, siendo fundamentales para la organización del proyecto y la estandarización del contenido. El archivo *.RVT* representa la unidad básica del proyecto y contiene tanto los elementos modelados como las vistas, tablas, parámetros compartidos y configuraciones. Su estructura interna permite mantener consistencia entre disciplinas, fases y niveles de detalle. Junto a este, los archivos *.RFA* actúan como unidades independientes que contienen familias de objetos, las cuales pueden insertarse en el archivo de proyecto, garantizando modularidad y reutilización. Por su parte, los archivos *.RTE*, usados como plantillas de proyecto, y *.RFT*, empleados para la creación de familias, permiten establecer parámetros iniciales, normas gráficas y configuraciones estándar.

En cuanto a los formatos de exportación, estos varían según el propósito de uso, distinguiéndose entre salidas gráficas sin datos estructurados, como *.PDF*, *.DWF*, *.DWFx* y *.DWG*, y formatos orientados a la interoperabilidad de datos. Los primeros se utilizan para documentación estática y coordinación externa, careciendo de propiedades paramétricas o semánticas. En contraste, el formato *.IFC* se posiciona como el estándar abierto para el intercambio de modelos BIM con su estructura de datos íntegra. Además, Revit permite exportaciones en *.GBXML* para análisis energético, así como en *.NWC* para procesos de coordinación y planificación con Navisworks.

En contraposición, existen formatos orientados explícitamente a la exportación o importación de conjuntos de datos estructurados. Uno de los más versátiles es el formato *.CSV*, que permite exportar tablas de planificación directamente desde Revit hacia hojas de cálculo. También con el complemento de conexión de bases de datos es posible conectar el modelo con bases de datos de Access. Asimismo, los informes generados en *.HTML*, si bien menos estructurados, pueden documentar alertas del modelo, auditorías internas o listados de verificación.

Adicionalmente, en muchos entornos BIM avanzados se recurre a extensiones específicas como el complemento de Herramientas de Interoperabilidad BIM, desarrollado para facilitar el intercambio estructurado de información con bases de datos externas o sistemas empresariales. Este complemento habilita exportaciones en formatos orientados a plantillas de requisitos, como *.XML* o *.XLSX*, que permiten estructurar atributos de objetos, clasificaciones, sistemas y parámetros en función de las necesidades específicas del proyecto o del entorno normativo. Asimismo, posibilita la generación de matrices de datos que integran múltiples fuentes y referencias.

Por último, el uso de Dynamo, la plataforma de programación integrada en Revit, introduce un nivel adicional de flexibilidad en la creación, manipulación y exportación de datos, posibilitando además la generación de rutinas que permitan extraer información del modelo y exportarla en formatos como *.CSV*, *.HTML* o incluso conectarse directamente con bases de datos a través de scripts personalizados.

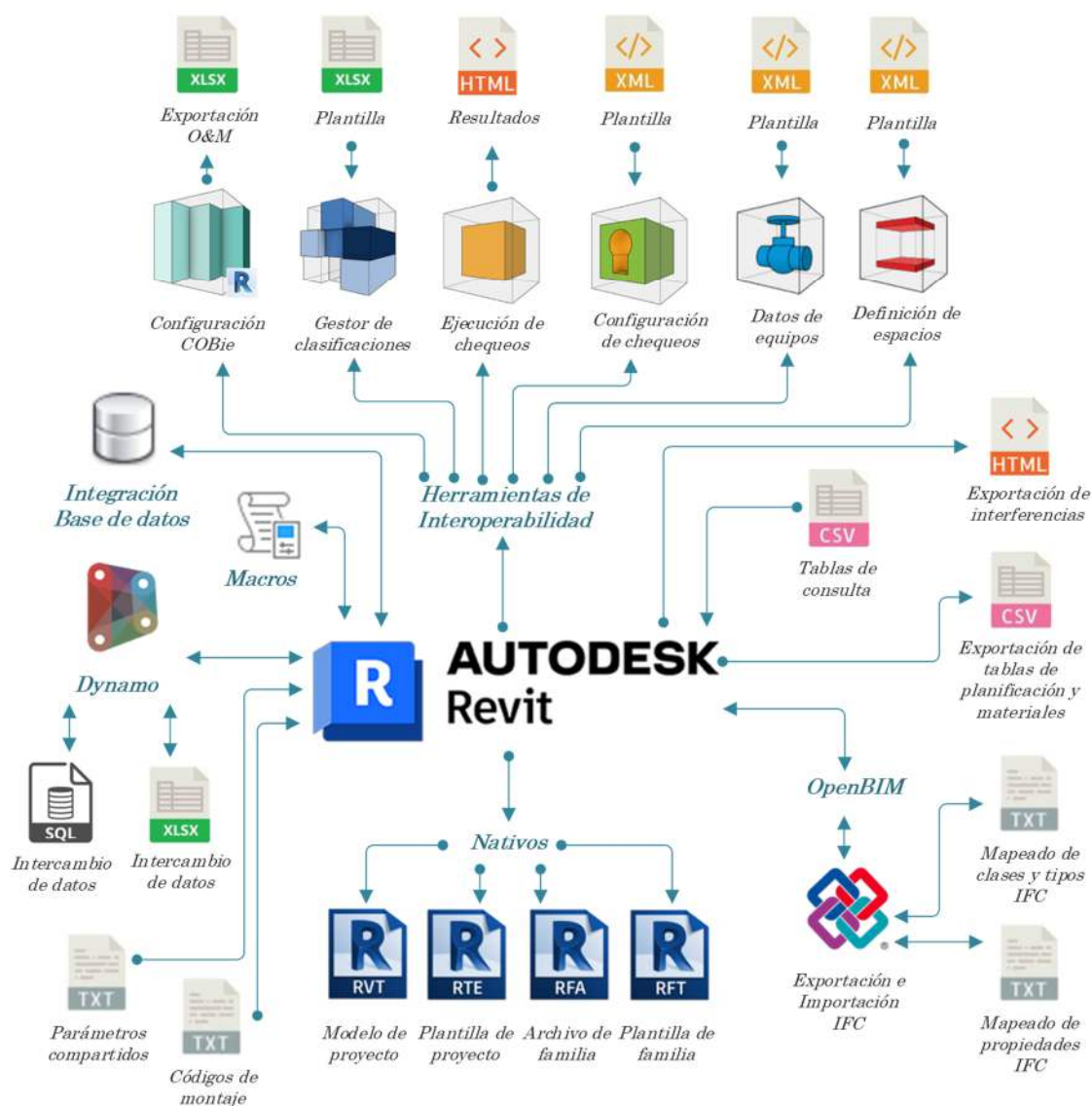


Figura 3.3.3. Formatos de trabajo en Autodesk Revit

Navisworks Manage

Navisworks, como una de las principales herramientas para la coordinación, revisión y gestión de proyectos en el entorno BIM, utiliza formatos de archivo específicos que permiten tanto la integración de modelos de diversas disciplinas como la colaboración entre distintos equipos. Sus formatos nativos son *.NWD*, *.NWF* y *.NWC*, que cumplen funciones distintas pero complementarias en el manejo de modelos. El archivo *.NWD* es un archivo “congelado” o estático que contiene una versión específica del modelo, con toda la información consolidada en una única unidad. Este formato es utilizado principalmente para la entrega final del modelo, permitiendo

una visualización del modelo sin modificaciones. Por otra parte, *.NWF* es el archivo de trabajo, que se utiliza para ir añadiendo las capas de información relacionadas con la coordinación de los modelos BIM. Por su parte, el archivo *.NWC* es más ligero y está diseñado como una versión optimizada para la carga rápida y la visualización de cada uno de los modelos generados.

En cuanto a los formatos de importación y exportación de modelos, Navisworks ofrece varias opciones que permiten la integración de información entre diferentes plataformas BIM, incluyendo archivos *.IFC*, *.DWG*, *.SKP*, *.DGN*, *.RCP*, etc.

En cuanto a la exportación de datos desde Navisworks, además de la posibilidad de generar archivos en formatos nativos, es posible exportar informes en formatos como *.CSV* o *.XML*. Estos formatos permiten la extracción de datos del modelo, tales como la detección de interferencias, la planificación temporal, o las listas de materiales, lo que facilita la integración con otros sistemas de gestión de proyectos, plataformas de planificación o bases de datos relacionales.

En cuanto a formatos de plantillas de datos, Navisworks permite utilizar principalmente archivos *.XML* para gestionar herramientas como el Inspector de selección, los conjuntos de selección y las propiedades rápidas.

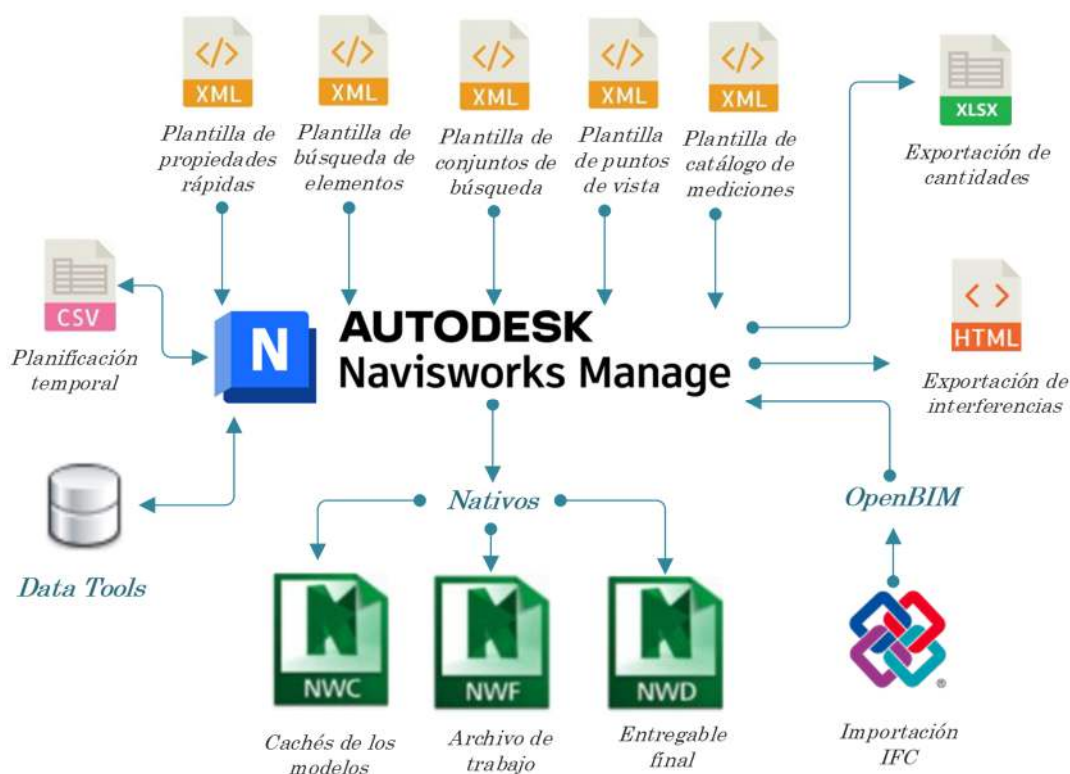


Figura 3.3.4. Formatos de trabajo en Navisworks Manage

Además, es posible acceder a la API del software para incorporar otros procesos de mayor complejidad relacionados con filtros de datos o con modificaciones gráficas o de visibilidad de los objetos BIM.

BIMCollab

BIMcollab también ofrece una variedad de formatos para importar y exportar información, lo que permite que los usuarios integren fácilmente modelos de diferentes sistemas de modelado y gestionen de manera eficiente.

El formato principal de archivo utilizado es el *.BCP*, que actúa como contenedor para almacenar información de los modelos BIM. Este formato almacena tanto los datos geométricos como los atributos asociados de todos los modelos *.IFC* o nubes de puntos con formato *.ISA* que sean incorporados a este modelo federado.

Por otro lado, BIMcollab también permite la exportación de información en formatos como *.CSV* y *.XML*, lo cual se realiza para generar informes detallados sobre el estado de las incidencias o para la extracción de conjuntos de datos específicos.

En relación a formatos de plantillas de datos, es posible utilizar archivos *.XML* para definir las SmartViews o vistas inteligentes. Además, en las últimas versiones se ha habilitado la posibilidad de trabajar con formatos *.IDS* para el control de la calidad de los datos BIM.

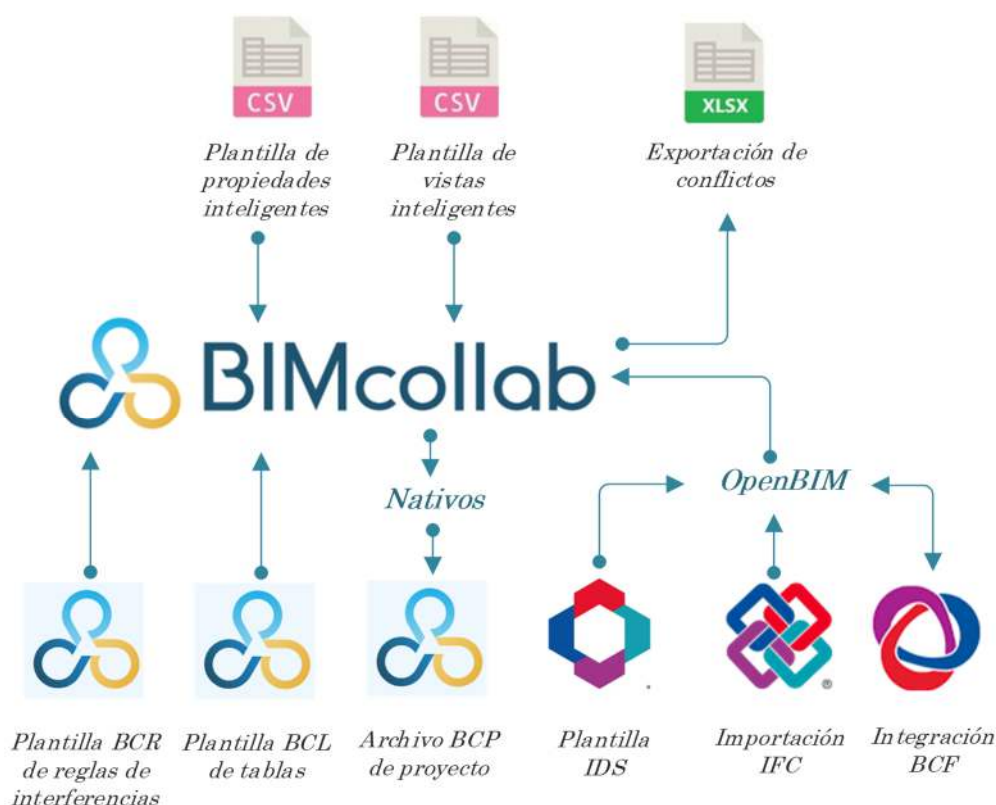


Figura 3.3.5. Formatos de trabajo en BIMCollab Zoom

Bexel Manager

Bexel Manager es una plataforma de gestión de proyectos BIM integral que gestiona la coordinación de proyectos de construcción. La plataforma está diseñada para facilitar la integración de información de diversas disciplinas y garantizar la interoperabilidad entre distintos sistemas BIM.

En este caso, uno de los formatos principales utilizados es el archivo *.BXM*, que contiene información relacionada con la gestión del proyecto BIM, como la programación de tareas, la gestión de costos, la planificación de recursos y el control de calidad. Este formato permite almacenar de manera estructurada toda la información del proyecto y facilita su revisión, seguimiento y actualización por parte de los equipos de trabajo.

Además del formato *.BXM*, Bexel es compatible con formatos de importación y exportación que permiten la integración con otros sistemas BIM y la exportación de datos a otras herramientas de gestión o análisis. Entre los formatos más utilizados para la importación se encuentran *.IFC*, lo que facilita la interoperabilidad con herramientas BIM.

Bexel también soporta la exportación de datos en formatos como *.CSV* y *.XML*. La exportación a *.CSV* es especialmente útil para generar informes tabulares, como el seguimiento de costos, el análisis de los recursos utilizados o la programación de actividades. El formato *.XML* se utiliza para la exportación de datos estructurados que pueden ser importados por otras plataformas o sistemas empresariales, permitiendo la integración de los datos con otros sistemas.

Bexel también permite la exportación de datos a formatos de coordinación como *.NWD*, lo que facilita la visualización y detección de interferencias entre diferentes disciplinas dentro del modelo BIM.

En relación a desgloses personalizados y a conjuntos de selección de objetos o sistemas de clasificación, Bexel también permite trabajar con formatos *.XLSX* para importar las consultas o agrupaciones de objetos BIM.

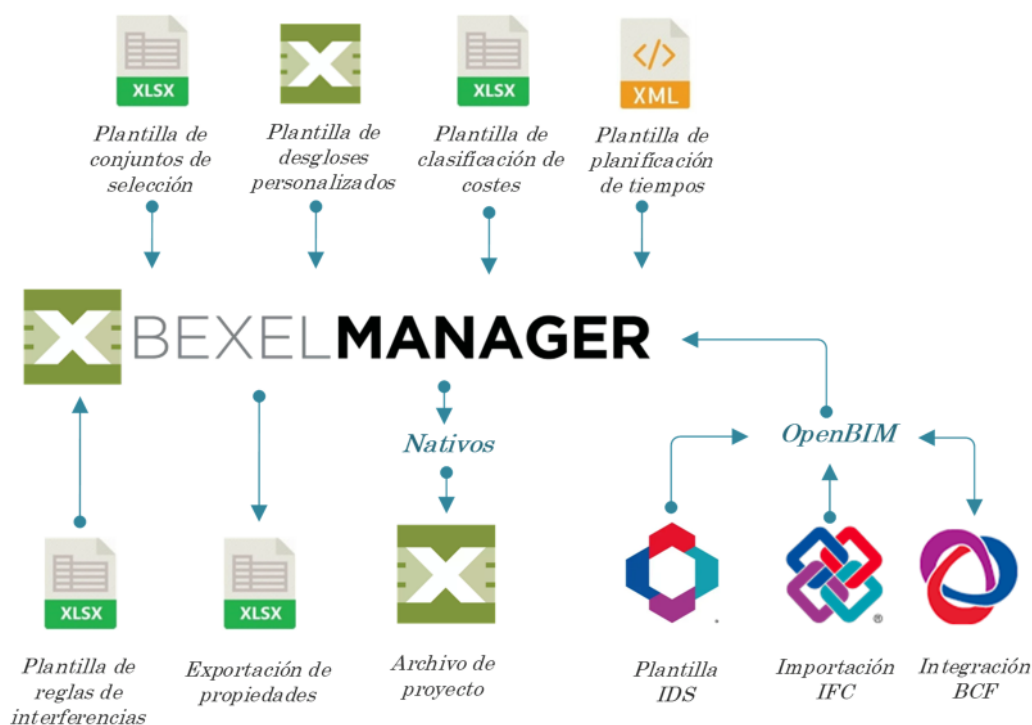


Figura 3.3.6. Formatos de trabajo en Bexel Manager

Blender Bonsai

Blender es una herramienta utilizada principalmente en la visualización de proyectos BIM que, junto con el complemento Bonsai, permite una integración entre la visualización avanzada y la gestión del ciclo de vida de un proyecto BIM.

El formato nativo es el archivo *.BLEND*, que contiene todos los elementos del proyecto, incluyendo modelos 3D, texturas, animaciones y materiales. Este archivo permite la manipulación total de los proyectos dentro de Blender, pero no está directamente vinculado a los flujos de trabajo BIM. Sin embargo, Blender es compatible con la importación de modelos en formatos como *.IFC*, lo que permite integrar información de otras plataformas BIM.

En cuanto a la exportación, Blender permite la salida de modelos en formatos como *.IFC*, *.FBX*, *.OBJ* o *.STL*, que se pueden utilizar en otras plataformas de visualización o fabricación. De manera similar, Bonsai facilita la exportación de datos de gestión en formatos como *.CSV* o *.XML*. El formato *.CSV* se utiliza para la exportación de datos tabulares, como la programación de actividades y los costos, mientras que el formato *.XML* permite estructurar los datos para su uso en otros sistemas de gestión.

Además, es posible trabajar con la biblioteca de IfcOpenShell desde el entorno de programación para gestionar la entrada o salida de datos de manera avanzada, dentro de los modelos *.IFC*.

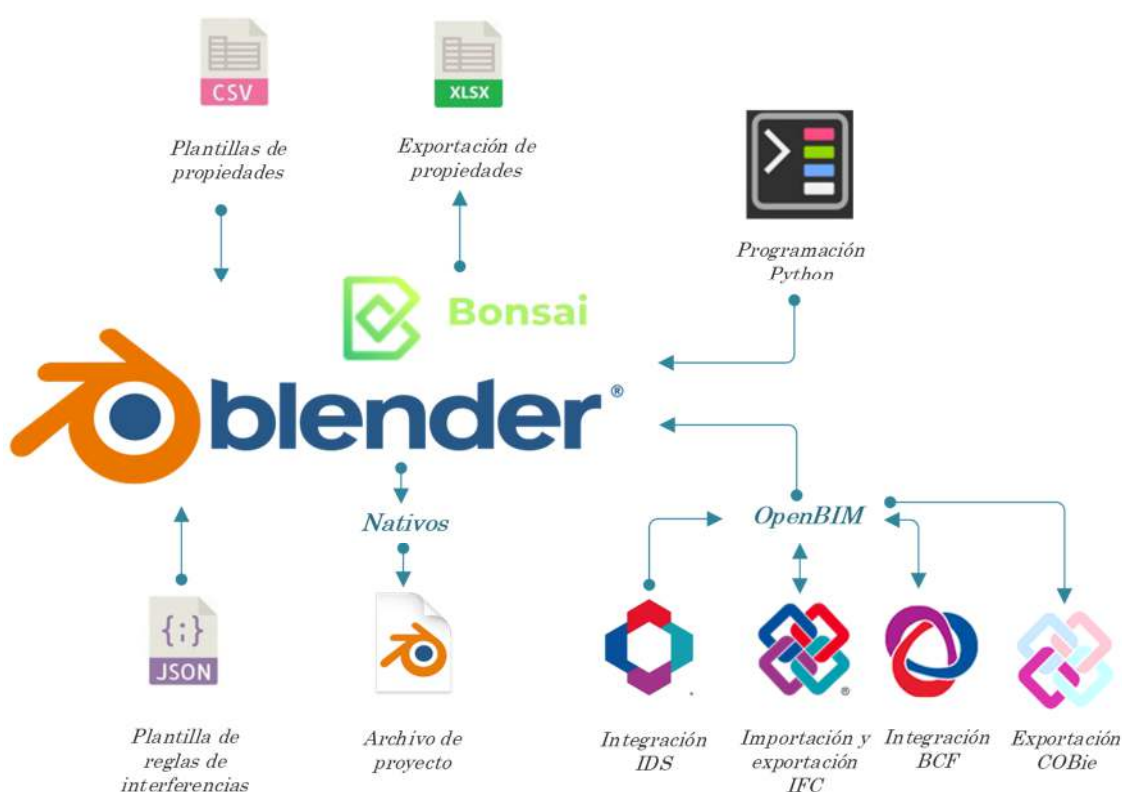


Figura 3.3.7. Formatos de trabajo en Blender con Bonsai

Autodesk Construction Cloud

Autodesk Construction Cloud (ACC) es una plataforma integrada de gestión de construcción que conecta equipos, proyectos y datos en tiempo real. En este caso, para el presente estudio, solo se tienen en cuenta los formatos de exportación de datos desde la plataforma.

ACC ofrece varias opciones para la extracción de datos. Los archivos pueden ser exportados en formato *.XLSX* para realizar análisis de datos y seguimiento de la programación. Además, los datos del proyecto, como los informes de progreso, las revisiones, las incidencias, los registros de archivos y las notificaciones, pueden ser exportados en formatos estándar que faciliten la integración con otros sistemas de gestión.

PowerBI

Power BI es una herramienta de visualización de datos que permite conectar, transformar, analizar y visualizar datos de diferentes fuentes en tiempo real. Su flexibilidad para importar y exportar datos lo convierte en una plataforma integral para el análisis y la representación visual de información.

El formato de proyecto es el *.PBIX*, que es el archivo de trabajo principal. Este archivo contiene todo el trabajo realizado en Power BI, incluidas las consultas de datos, las visualizaciones creadas, los informes y las conexiones a las fuentes de datos. Este formato permite guardar el diseño del informe y las configuraciones, y es utilizado para compartir proyectos entre diferentes usuarios.

Para la importación, Power BI es altamente flexible y permite la conexión a una amplia variedad de fuentes de datos. Entre los formatos de archivo se incluyen *.CSV*, *.XLSX*, y *.TXT*, lo que facilita la integración de datos desde hojas de cálculo, bases de datos o sistemas externos. Además, el programa puede conectarse a fuentes de datos como bases de datos SQL, proporcionando la capacidad de integrar desde diversas aplicaciones y sistemas de gestión de proyectos.

La plataforma también soporta la importación de archivos *.JSON* para intercambiar datos estructurados entre sistemas, permitiendo la integración de datos provenientes de plataformas que exportan información en este formato.

3.3.4. Definición de niveles de análisis

La parte práctica del estudio se basa en una clasificación en cuatro niveles de aplicabilidad que permite una evaluación detallada y precisa de la sistematización propuesta en diferentes contextos de complejidad y alcance. Para ello, se implementan los procesos de desarrollo dentro de diferentes entornos y software BIM y de bases de datos, estableciendo diferentes preconfiguraciones, puentes, automatizaciones o plugins con el objetivo de garantizar la conectividad y veracidad de la información en todos los casos.

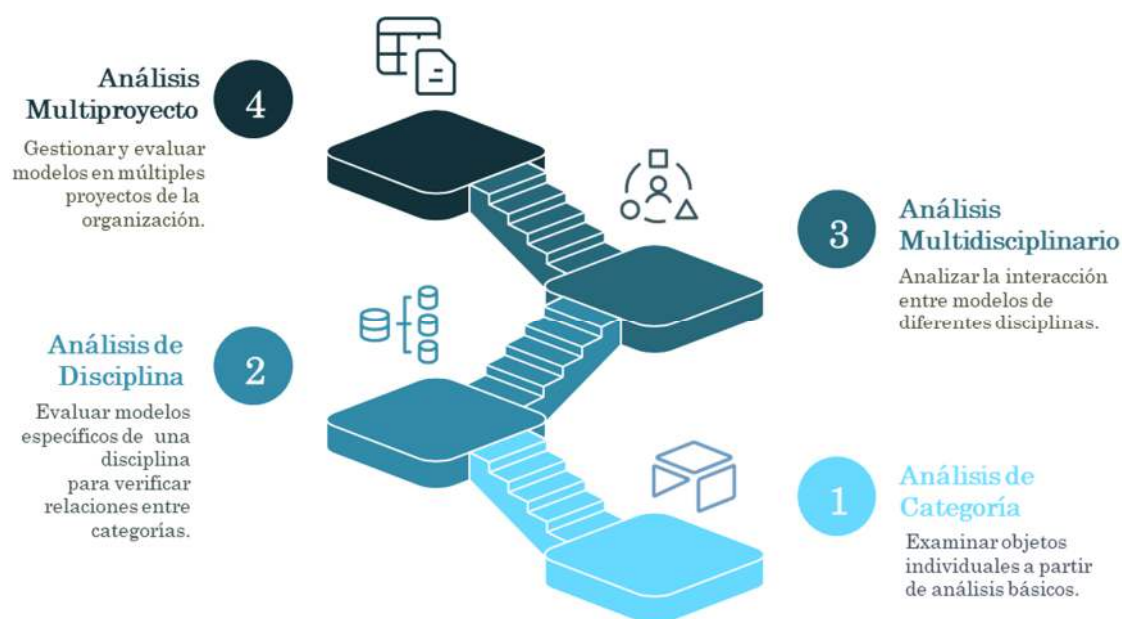


Figura 3.3.8. Definición de niveles de análisis

Cabe señalar que para la investigación se establecen dentro de los modelos BIM una serie de elementos y conjuntos de propiedades específicos, seleccionados de diferentes usos, disciplinas y fases BIM con el fin de tener una muestra general de la solución propuesta. Además, se han estudiado los modelos OpenBIM con diferentes versiones del esquema IFC para identificar diferentes procesos de conexión entre bases de datos.

Análisis a nivel de categoría

El análisis a nivel de categoría o clase de objeto parte de la clasificación fundamental de los elementos del modelo BIM en función de su categoría tipológica, entendida como el conjunto de objetos que comparten una función constructiva, un comportamiento técnico y una representación estandarizada. Trabajar a nivel de categoría proporciona una forma eficaz de estudiar estructuras de datos específicas y preparar reglas de control automatizadas. Asimismo, se facilita el diagnóstico temprano de errores y la comparación entre diferentes herramientas o versiones del modelo, posibilitando así revisiones de conectividad e interoperabilidad.

Para el presente estudio, se opta por trabajar con la categoría de muros de manera individual, por considerarse una de las categorías básicas, estructurantes y universales dentro de cualquier modelo de edificación. Los muros definen los límites físicos de los espacios a la vez de participar en otros aspectos relacionados con la compartimentación, la estabilidad estructural, el aislamiento térmico y acústico, la seguridad contra incendios, y la interacción con otros sistemas técnicos. Su presencia transversal en todas las fases del ciclo de vida los convierte en una categoría clave para el estudio detallado a nivel de objeto.

En relación al esquema IFC, estos muros se definen mediante la entidad *IfcWall* o su subclase *IfcWallStandardCase*, dependiendo del tipo de geometría y del método de definición adoptado. Esta categoría posee una complejidad semántica considerable, ya que está asociada a múltiples relaciones de tipo espacial, material, funcional y constructivo. Entre las más relevantes se encuentran:

- Relaciones jerárquicas: los muros siempre deben estar anidados dentro de entidades espaciales como niveles o zonas funcionales.
- Relaciones de agregación o descomposición: en muchos casos los muros forman parte de un ensamblaje más complejo, o también puede contener subelementos como capas de materiales, acabados o refuerzos modelados de forma explícita.
- Asignación de propiedades: la clase de muros dispone de conjuntos de propiedades estandarizados que pueden incluir propiedades personalizadas, lo que exige un análisis cuidadoso de los atributos asociados como el tipo de resistencia, tipo de soporte o función de carga.
- Relaciones de materiales: cada muro puede estar vinculado a una composición material definida por capas, lo que influye directamente en los cálculos de rendimiento térmico o estructural.
- Interacción geométrica: los muros pueden presentar uniones complejas con otros elementos como puertas o ventanas además de relaciones con huecos.

Estas múltiples capas de relaciones y anidamientos hacen del muro una entidad ideal para evaluar la calidad semántica y la consistencia de la información a nivel de objeto. Además, al ser una categoría presente en todos los softwares BIM y ampliamente soportada por las implementaciones IFC, su análisis permite contrastar distintos niveles de interoperabilidad y calidad de exportación entre plataformas.

Este enfoque detallado proporciona una base metodológica sólida para validar cómo se estructuran y representan los datos en una escala inicial, y sirve como punto de partida para construir reglas de validación, estructuras de datos reutilizables, y mecanismos de conexión con bases de datos externas.

Análisis a Nivel de Disciplina

El análisis a nivel de disciplina independiente se centra en la evaluación específica de los elementos pertenecientes a una sola especialidad técnica. En este desarrollo, el foco se ha puesto en la disciplina de arquitectura, por su papel estructurante en la definición formal y funcional del modelo BIM, y por ser habitualmente el punto de partida en el desarrollo de proyectos. Este análisis se ha realizado a partir de un modelo BIM correspondiente a una vivienda unifamiliar, que ha sido depurada para contener exclusivamente elementos propios de la disciplina arquitectónica. En total, se han trabajado 21 clases distintas, incluyendo muros, puertas, ventanas, suelos, cubiertas, escaleras, barandillas, entre otras, cada una con múltiples instancias, para un análisis amplio y representativo de los objetos más habituales en este tipo de proyectos.

El estudio aborda tanto la estructura jerárquica del modelo como la coherencia en las relaciones entre elementos. Todos los objetos deben estar correctamente vinculados a niveles espaciales, estableciendo una estructura clara que permita la definición, consulta y extracción de la información. Asimismo, se ha revisado la correcta conexión geométrica entre clases arquitectónicas. Por ejemplo, se ha comprobado que las puertas y ventanas estén adecuadamente insertadas en muros, o que las escaleras estén definidas como elementos compuestos jerárquicamente.

Otro aspecto fundamental del análisis es la verificación de la estructura informativa del modelo. Se ha evaluado la correcta asignación de propiedades a cada clase arquitectónica, tanto mediante conjuntos estandarizados del esquema IFC como mediante propiedades personalizadas relevantes para la edificación. Este desarrollo incluye la completitud de los datos, la validez de los valores asignados y la coherencia en el uso de tipos, unidades y nomenclaturas. Se ha puesto especial atención en la correcta aplicación de tipos y en la consistencia entre las instancias del modelo.

El control de calidad también abarca aspectos como la homogeneidad en la clasificación y codificación de los elementos, el uso correcto de sistemas de clasificación y la ausencia de errores sistemáticos, como elementos sin tipo, propiedades incompletas o relaciones mal definidas.

Análisis a Nivel Multidisciplinario

Este análisis multidisciplinario se orienta en la evaluación conjunta de modelos pertenecientes a distintas disciplinas técnicas involucradas en un proyecto de edificación. A diferencia del enfoque centrado en una única disciplina, este nivel se enfoca en estudiar cómo interactúan y se coordinan los modelos parciales generados por equipos diferentes, lo que permite identificar problemas de compatibilidad, interferencias y diferencias semánticas que afectan directamente la calidad del modelo federado y la eficiencia del flujo de trabajo colaborativo.

En el marco de esta investigación, se plantea trabajar con modelos correspondientes a un edificio de viviendas colectivo, compuesto por tres modelos disciplinares: arquitectura, estructura y MEP (instalaciones mecánicas, eléctricas y de fontanería). Estos modelos incluyen un total de 38 clases distintas del esquema IFC, lo que permite un análisis amplio de los distintos tipos de objetos utilizados en la práctica profesional. El modelo se configura como un entorno realista y lo suficientemente complejo para analizar los desafíos de la interoperabilidad y la coordinación entre diferentes disciplinas.

Además de las clases constructivas propias de cada especialidad, el análisis incluye entidades espaciales y abstractas del modelo, fundamentales para su organización estructural. Se consideran clases como *IfcProject*, *IfcSite*, *IfcBuilding*, *IfcBuildingStorey*, *IfcSpace* e *IfcZone*, cuya correcta definición y uso es clave para garantizar la consistencia en la organización de los modelos. Estas entidades permiten también establecer vínculos jerárquicos y espaciales entre los distintos elementos, y sirven de base para la coordinación entre disciplinas.

Por otro lado, se estudia la coherencia semántica e informativa entre los modelos, evaluando que los objetos compartan estructuras de datos compatibles, que los espacios estén correctamente definidos y referenciados por los elementos de instalaciones, y que existan correspondencias entre objetos. Las relaciones definidas en el esquema IFC son clave para modelar y analizar las interacciones entre entidades de los modelos.

Asimismo, se incorpora al análisis la variabilidad en la implementación del esquema IFC por parte de los diferentes programas utilizados por cada disciplina. Se identifican diferencias en la estructura de los modelos exportados, el uso de propiedades, o la definición de relaciones, lo que permite estudiar las implicaciones

que esto tiene en la interoperabilidad y definir estrategias de normalización y validación.

Análisis a Nivel Multiproyecto

A nivel multiproyecto, el análisis se centra en la gestión y evaluación de modelos BIM no como entidades individuales, sino como parte de un conjunto más amplio de proyectos de la organización. Este enfoque permite estudiar cómo se estructura, se comparte y se reutiliza la información entre distintos modelos dentro de un portafolio de proyectos, así como detectar patrones, incoherencias o desviaciones que puedan surgir cuando se manejan múltiples fuentes de datos en paralelo. La dimensión multiproyecto resulta eficaz para avanzar hacia una visión organizativa de la información BIM, donde la consistencia, trazabilidad y gobernanza de los datos se vuelven prioritarias.

Para este nivel de análisis, se trabaja con la información procedente de los modelos utilizados en los niveles, incorporando además un tercer modelo BIM correspondiente a un edificio de viviendas de mayor escala y mayor nivel de definición, que contiene un total de 47 clases IFC. Este modelo complementario permite ampliar el espectro de análisis y comparar cómo varía la estructura de la información en función de la escala del proyecto, el detalle de modelado y el uso de clases específicas en contextos más complejos.

El objetivo principal es estudiar la integración de datos entre proyectos, identificar discrepancias en la codificación de la información, evaluar la consistencia de la estructura espacial y semántica entre modelos, y analizar cómo se gestionan elementos comunes, como conjuntos de propiedades, tipologías, nomenclaturas o estructuras jerárquicas. También se observa la reutilización o replicación de ciertos patrones de gestión, y cómo estas prácticas pueden afectar la coherencia o la calidad de los datos a escala global.

Un aspecto relevante de este análisis es la verificación de la coherencia entre instancias equivalentes de diferentes proyectos, por ejemplo, cómo se definen y documentan los muros, puertas o sistemas MEP en edificios con distintas dimensiones, pero con requisitos funcionales similares. Se analiza también la estabilidad y compatibilidad de las relaciones IFC entre modelos exportados por distintas herramientas o versiones de software.

Se presta especial atención a la gestión de versiones y evolución del modelo, lo cual es crítico cuando se trabaja con múltiples proyectos en diferentes fases o con revisiones sucesivas. La trazabilidad de los cambios, la conservación de las relaciones entre elementos, y la consistencia de la información a lo largo del tiempo se convierten en indicadores clave en este nivel de análisis.

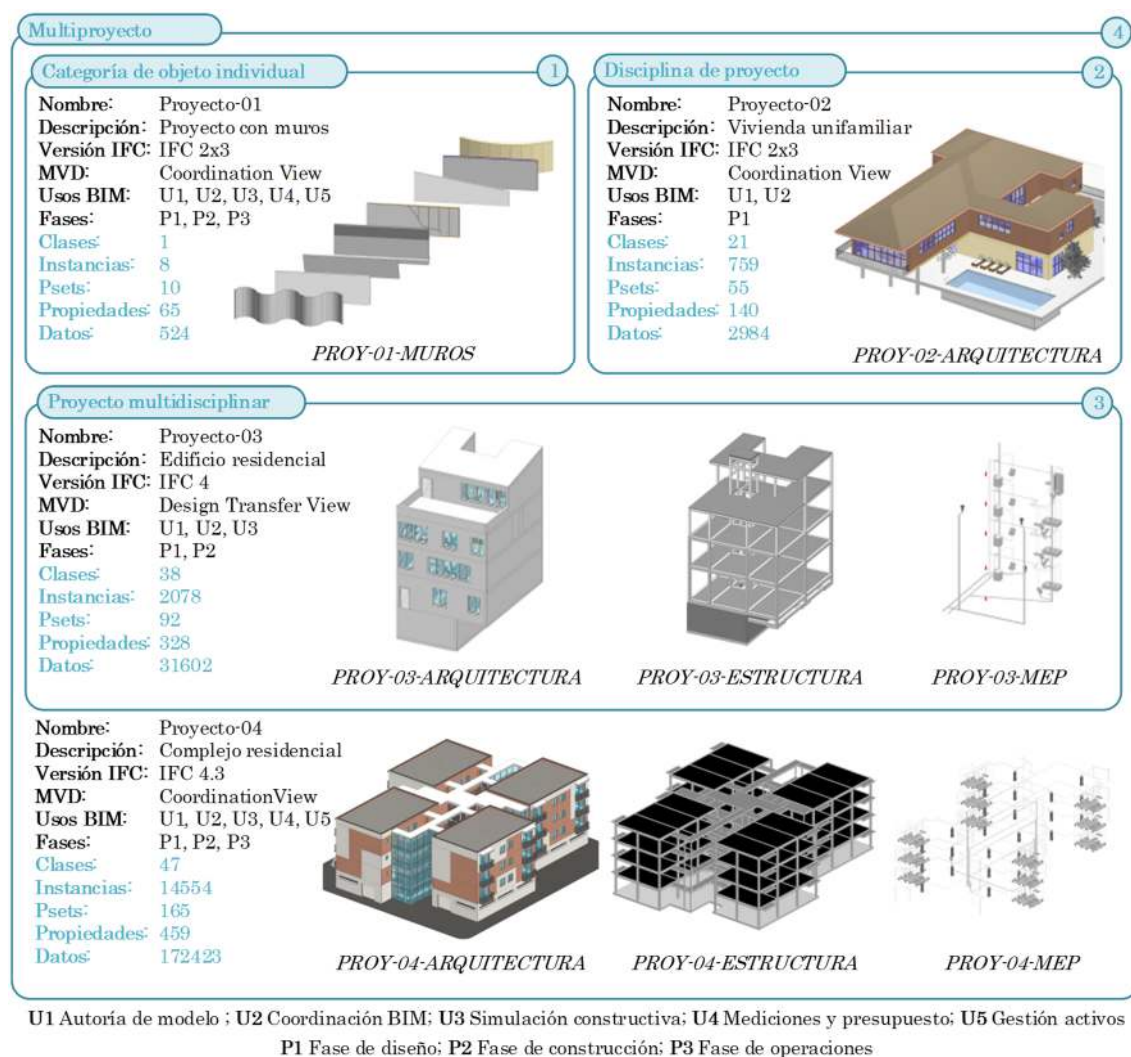


Figura 3.3.9. Propuesta de aplicación de la sistematización

Una vez establecido el alcance y la estrategia de la sistematización, se procede a la aplicación práctica, en la cual se utilizan diferentes modelos BIM, junto con la creación de una base de datos SQL, diferentes scripts para la gestión de datos, un conjunto de hojas predefinidas para la gestión de la información y otros archivos de apoyo con el objetivo de verificar la viabilidad y confiabilidad del proceso.

3.3.5. Gestión de nuevos roles y responsabilidades

A partir de un estudio detallado de los diferentes roles BIM actuales y de las nuevas necesidades que introduce la sistematización propuesta, se establecen cuatro perfiles clave, que cuentan con permisos específicos y una serie de responsabilidades clave. Aunque algunos de estos perfiles ya existen en los organigramas BIM actuales, la nueva propuesta plantea nuevas responsabilidades para cada uno de ellos.

Rol	Permisos	Responsabilidades clave
Administrador en la gestión de datos global (R1)	Acceso total a todas las bases de datos y entornos BIM. Capacidad de crear, modificar y eliminar datos. Realizar copias de seguridad. Autorización para gestionar permisos de otros usuarios en el sistema de datos.	Monitorear y mantener la calidad y precisión de los datos en los sistemas. Implementar políticas y procedimientos. Realizar auditorías y revisiones periódicas para asegurar el cumplimiento de las regulaciones de datos. Coordinarse con otros departamentos. Gestionar copias de seguridad y planes de recuperación ante desastres. Llevar a cabo la consolidación de los datos del proyecto BIM dentro de la base de datos global de la empresa. Generar informes y análisis basados en datos.
Gestor BIM de datos (R2)	Acceso total a las plataformas BIM y al software utilizado. Autorización para integrar y actualizar datos entre diferentes sistemas BIM y bases de datos. Permiso para gestionar el acceso de otros usuarios a los datos BIM.	Supervisar la integridad y precisión de los datos BIM durante todo el ciclo de vida del proyecto. Coordinar la integración y federación de datos BIM de todas las disciplinas de activos de construcción. Establecer y mantener estándares de calidad. Facilitar la colaboración entre diferentes equipos.
Coordinador BIM (R3)	Acceso a todos los modelos BIM del proyecto. Uso de automatismos para controles de calidad de datos BIM. Capacidad para revisar, comentar y solicitar modificaciones a los modelos BIM.	Coordinar diferentes equipos y disciplinas para asegurar la correcta integración de los modelos BIM. Revisar y verificar la calidad y precisión de los modelos BIM en relación con los requisitos del proyecto. Facilitar la colaboración entre los equipos de diseño, ingeniería y construcción. Identificar y comunicar conflictos en los modelos BIM.
Líder de disciplina BIM (R4)	Acceso a modelos BIM relacionados con una disciplina específica. Capacidad para crear, modificar y revisar modelos BIM dentro del área de especialidad técnica.	Supervisar la creación y actualización de modelos BIM para garantizar que cumplan con los requisitos del cliente. Coordinar con otros líderes de disciplinas para garantizar la correcta integración de modelos. Revisar y resolver problemas relacionados con modelos BIM de una disciplina específica.

Tabla 3.3.6. Definición de nuevos roles de la sistematización

Si bien los perfiles definidos anteriormente constituyen los roles principales dentro del nuevo esquema de gestión BIM conectado a bases de datos, en contextos donde los entornos tecnológicos son más complejos, o donde es necesario incorporar plataformas adicionales provenientes de otras áreas de la organización, podrían contemplarse perfiles adicionales que respondan a esas exigencias específicas. Estos roles adicionales no son imprescindibles en todos los casos, pero su incorporación

podría ser clave en proyectos de gran envergadura, alta exigencia normativa o fuerte componente de interoperabilidad entre sistemas.

Entre ellos, podría considerarse el perfil de Especialista en Interoperabilidad BIM-Datos, cuya función se centraría en facilitar la integración avanzada entre modelos BIM y sistemas de bases de datos relacionales, empleando para ello soluciones de conexión o APIs que aseguren una transferencia de información precisa y coherente entre entornos digitales.

Asimismo, podría ser útil la figura del Analista de Datos, con competencias orientadas al tratamiento, análisis e interpretación de datos provenientes tanto de modelos como de sistemas externos, capaz de transformar grandes volúmenes de información en indicadores estratégicos que informen la toma de decisiones en las diferentes fases del proyecto.

En contextos especialmente regulados o donde la trazabilidad de la información digital sea prioritaria, podría ser necesario incorporar un Auditor Digital BIM, encargado de revisar de manera independiente la alineación de los datos y modelos con los estándares contractuales, normativos y organizativos, garantizando así la integridad documental y el cumplimiento de los requisitos establecidos.

Finalmente, en entornos donde se busque una adaptación avanzada de las herramientas a los procesos propios de la organización, podría resultar conveniente contar con un Desarrollador de Soluciones BIM, capaz de programar herramientas específicas, automatizar flujos de trabajo o generar conectores entre plataformas.

Estos perfiles, aunque secundarios respecto a los roles centrales de la propuesta, podrían convertirse en actores estratégicos para asegurar una implementación eficaz del sistema en escenarios de mayor sofisticación tecnológica o integración interdepartamental.

3.4. Definición de procesos de la sistematización

La solución de estudio tiene como enfoque conseguir generar un entorno que permita una mayor colaboración BIM entre todos los agentes intervinientes del proyecto, una gestión de los datos más estructurada y controlada y una agrupación de toda la información de la organización dentro de una base de datos transversal, interoperable y actualizada. Para ello, se propone dividir el marco general de aplicación en cuatro procesos relacionados con la creación de la base de datos global de la organización, la extracción e integración de los datos dentro de los modelos BIM, la ejecución de controles de calidad de la información y, por último, la consolidación total de los datos de proyectos.

3.4.1. Creación de base de datos relacional

A continuación, se presenta un esquema que describe un flujo estructurado para la creación e integración de la base de datos SQL, que funciona como herramienta principal para la gestión de datos en entornos organizacionales y de proyectos. Este flujo se compone de dos procesos fundamentales: la adquisición y estructuración de datos, y la configuración técnica y operativa de la base de datos.

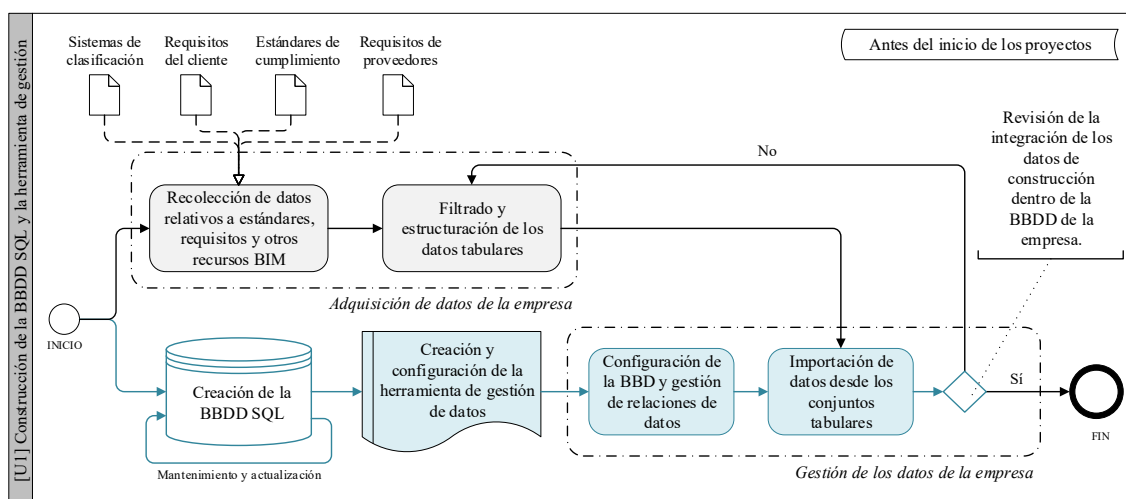


Figura 3.4.1. Proceso de creación y gestión de la BBDD SQL

Adquisición y estructuración de datos organizacionales

El proceso inicial se centra en la recolección de datos relevantes desde múltiples fuentes corporativas, como sistemas de clasificación, requisitos del cliente, estándares de cumplimiento y requisitos de proveedores. Esta información incluye tanto datos tabulares como no estructurados, y se enfoca especialmente en los requerimientos técnicos asociados a metodologías BIM.

Una vez recolectados, los datos pasan por un proceso de filtrado y estructuración, lo que permite transformarlos en conjuntos tabulares coherentes. Este paso es esencial para asegurar la calidad, integridad y normalización de los datos, ya que define los tipos de datos, las relaciones entre entidades y los identificadores clave para su posterior uso dentro de la base de datos.

Creación y configuración de la BBDD SQL

Paralelamente, se desarrolla la base de datos SQL, la cual incluye la definición de su estructura general, su arquitectura lógica y la asignación de roles y permisos de acceso para los distintos usuarios. Esta etapa también contempla la creación de una herramienta de gestión de datos que permite la interacción entre usuarios y la base de datos de forma segura y eficiente.

Con la base de datos ya definida, se procede a su configuración, lo cual abarca la definición de relaciones entre tablas, esquemas de normalización y reglas de integridad referencial. A continuación, mediante scripts automatizados, se lleva a cabo la importación de los datos tabulares estructurados, asegurando que todas las propiedades y definiciones se integren correctamente en el sistema.

Gestión y mantenimiento continuo

Una vez integrada la información, se realiza una revisión continua de la integración de nuevos datos. Esto permite evaluar la consistencia con respecto a la estructura global de la base de datos y verificar que no se generan conflictos en identificadores o claves. Este proceso garantiza que la base de datos permanezca actualizada y adaptable frente a nuevas necesidades organizacionales. Cabe destacar que, en las fases iniciales, únicamente se incorporan datos organizativos generales. Posteriormente, a medida que los proyectos avanzan, se agregan datos específicos de cada uno, lo que permite centralizar modelos técnicos para su gestión, consulta e intercambio entre equipos. Esta modularidad asegura que cualquier nuevo conjunto de datos pueda ser integrado sin alterar la lógica estructural preexistente, simplemente actualizando claves o relaciones.

3.4.2. Integración de datos en modelos BIM

Este apartado detalla el proceso de extracción y transferencia de datos desde la base de datos SQL hacia los modelos BIM, facilitando la generación de modelos enriquecidos con información estandarizada y adaptada a las necesidades específicas de cada fase de proyecto.

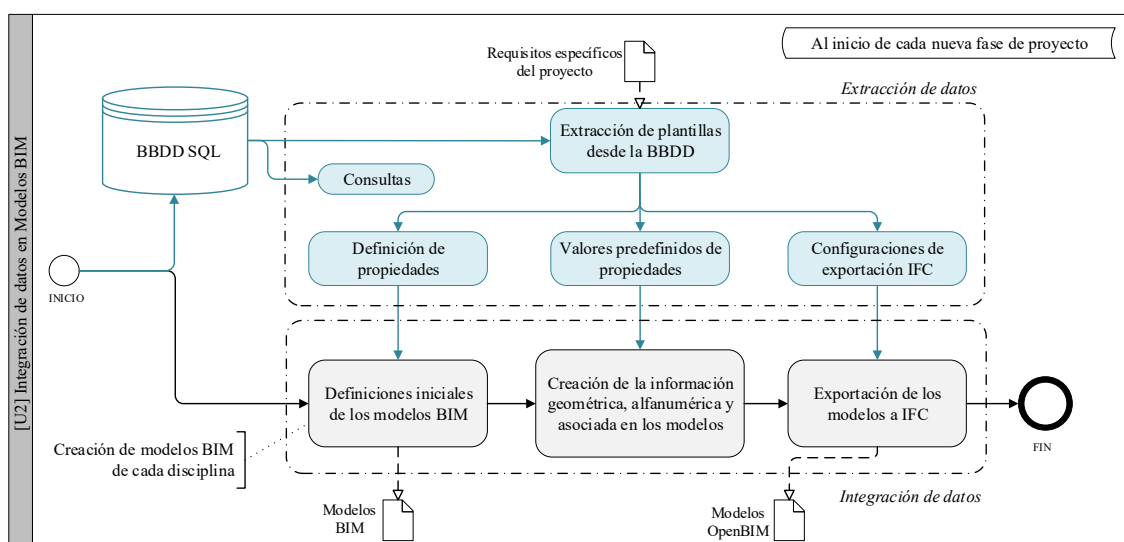


Figura 3.4.2. Proceso de integración de datos en modelos BIM

Extracción de datos desde la BBDD

Al inicio de cada nueva fase de proyecto, se accede a la BBDD mediante consultas SQL personalizadas y diferentes soluciones de programación. Estos trabajos permiten filtrar y extraer datos específicos según los requerimientos del proyecto, tales como definiciones de materiales, propiedades técnicas, listados de valores y configuraciones relacionadas con los usos BIM. Este proceso permite acceder a información detallada y también mapear equivalencias entre diferentes estándares y lenguajes, como versiones del esquema IFC, sistemas de clasificación o estructuras corporativas propias.

Para garantizar la eficiencia, se utilizan plantillas y scripts previamente almacenados, los cuales permiten automatizar las consultas más comunes y estandarizar la estructura de datos resultante. Esto minimiza la intervención manual, reduce la posibilidad de errores y optimiza los flujos de trabajo entre equipos de modelado y gestión.

Procesamiento y agrupación de datos

Los datos extraídos se procesan internamente para organizar la información en conjuntos tabulares estructurados. Estos conjuntos se agrupan según el tipo de modelo, la disciplina involucrada y la fase del proyecto. Asimismo, se categorizan según el uso BIM al que se destinarán, lo cual permite personalizar la carga de datos para cada disciplina sin duplicar procesos ni generar redundancia informativa.

Entre los aspectos configurables en esta fase se encuentran:

- Definición de nuevas propiedades a incorporar en los modelos.
- Asignación de valores predeterminados para acelerar la inserción de datos estándar.
- Configuraciones de exportación IFC, esenciales para garantizar la compatibilidad con estándares OpenBIM y plataformas colaborativas.

Integración en los modelos BIM

Una vez obtenida y agrupada la información, se procede a su asociación directa con los modelos BIM generados por cada disciplina. Esta integración se realiza mediante scripts automatizados, tanto visuales como textuales, lo cual garantiza que la incorporación de datos se realice de forma homogénea y trazable.

El proceso contempla tres operaciones clave:

- Definición inicial de propiedades en los modelos, creando los contenedores de datos requeridos.
- Asignación de información alfanumérica y relacional, asociada con los objetos del modelo.
- Configuración de las reglas de exportación IFC, permitiendo que los modelos nativos puedan convertirse a formatos interoperables manteniendo la estructura de datos requerida por agentes externos.

El resultado es un modelo BIM enriquecido, normalizado y exportable, adecuado para su uso en entornos colaborativos y plataformas digitales. Este enfoque permite replicar el proceso de forma iterativa al inicio de cada fase de proyecto, asegurando la coherencia de la información y facilitando estrategias como la trazabilidad de requisitos y la gobernanza de datos.

3.4.3. Controles de calidad

Una vez integrados los datos en los modelos BIM y generadas sus respectivas exportaciones a formatos IFC, se activa un proceso sistemático de control de calidad orientado a verificar la idoneidad, completitud y coherencia de los modelos conforme a los requisitos específicos del proyecto. El proceso, tal como se representa en el esquema, se implementa de forma periódica durante todas las fases del proyecto y se apoya en datos estructurados extraídos desde la base de datos SQL.

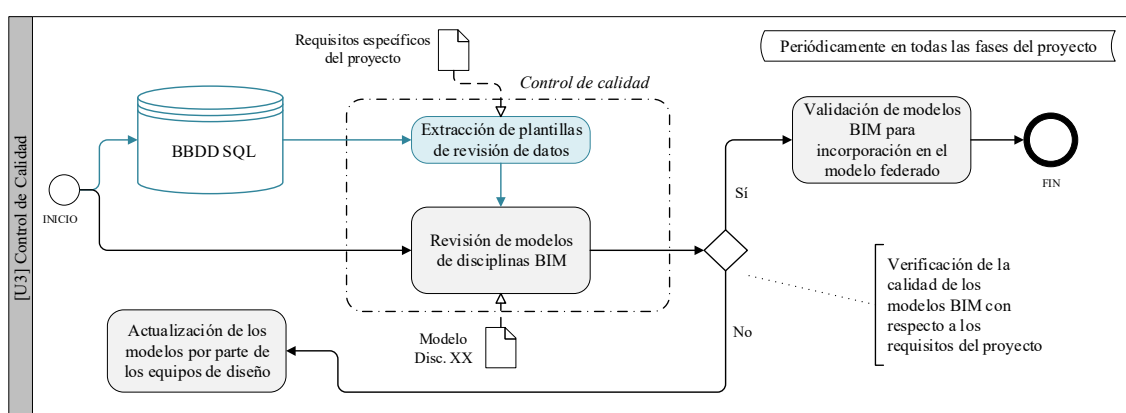


Figura 3.4.3. Proceso para controles de calidad

Extracción de plantillas de revisión de datos

La base de datos SQL contiene plantillas configuradas específicamente para la verificación de modelos IFC, las cuales definen los criterios y parámetros de revisión. Estas plantillas pueden incluir listas de entidades, atributos obligatorios, nomenclaturas normalizadas y configuraciones específicas de exportación para distintos usos BIM.

La extracción de estas plantillas garantiza la trazabilidad y estandarización de los criterios de revisión, asegurando que todos los modelos sean validados bajo las mismas condiciones técnicas y normativas.

Revisión técnica de los modelos por disciplina

Una vez disponibles las plantillas, se procede a la revisión de modelos BIM por disciplina, utilizando tanto herramientas automatizadas como análisis manual por parte de los coordinadores y gestores BIM. Este paso incluye un amplio conjunto de comprobaciones estructuradas, entre ellas:

- Validación de la nomenclatura de entidades y la coherencia de clases y tipos.
- Existencia y completitud de la información alfanumérica asociada a cada objeto del modelo.

- Revisión de la estructura de conjuntos de propiedades, incluyendo tipos de datos, unidades de medida y formatos.
- Verificación de valores enumerados, límites mínimos y máximos para datos numéricos y validación de formatos para campos específicos.
- Comprobación de coherencia en los sistemas de clasificación: códigos, descripciones y relaciones entre elementos.
- Control de que los modelos contengan únicamente clases de entidades permitidas por disciplina, garantizando la no duplicidad de información.
- Revisión de la versión IFC utilizada y su adecuación con las exigencias del cliente o normativa vigente.

Otras revisiones más detalladas que contemplan las de otros usos BIM se han especificado en el Anexo E del presente estudio.

Verificación y validación de modelos

En función del resultado de las revisiones, el proceso de calidad sigue dos posibles caminos:

- Si el modelo cumple los criterios establecidos, se realiza la verificación de modelos BIM y su incorporación al modelo federado. Esta acción consolida el modelo como una fuente válida de información para procesos de coordinación interdisciplinar, planificación y control.
- Si el modelo no supera los controles, se emite una notificación al equipo de diseño correspondiente para que se realice la actualización del modelo, corrigiendo los errores detectados y volviendo a iniciar el ciclo de revisión.

Este enfoque de control iterativo, estructurado y automatizado permite mantener un nivel alto de calidad en los modelos BIM, lo cual es esencial en entornos colaborativos y proyectos basados en datos. Este sistema además garantiza que los modelos IFC cumplan con los requerimientos del proyecto, siendo confiables y consistentes para su uso y reduciendo riesgos durante la ejecución.

3.4.4. Consolidación de los datos en la BBDD

El proceso de integración de datos provenientes de los modelos BIM en la base de datos relacional, busca centralizar la información de los proyectos en un entorno estructurado, facilitando su análisis, reutilización y gestión. Este enfoque evoluciona desde una visión fragmentada de los proyectos hacia una plataforma corporativa de gestión integral.

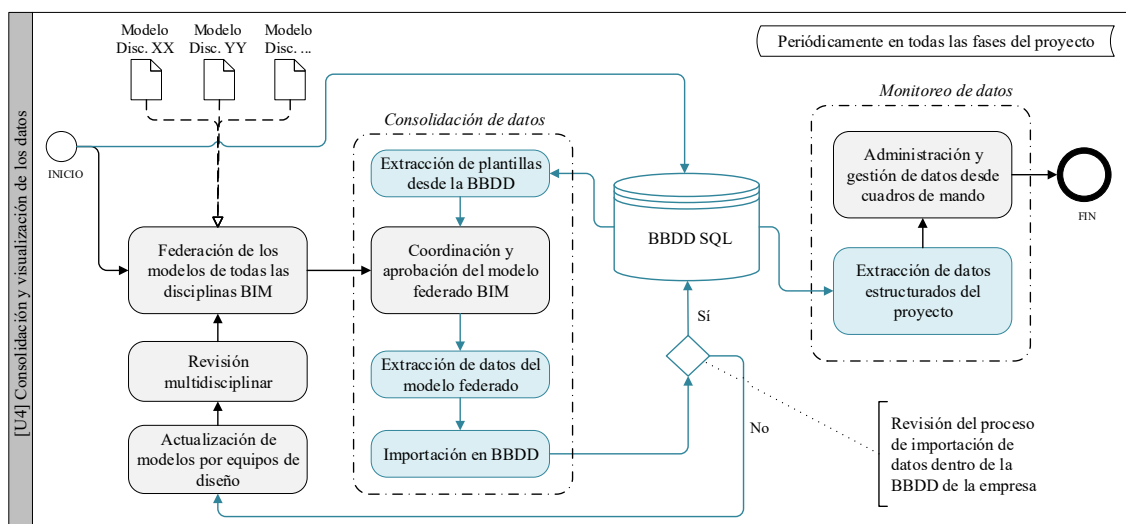


Figura 3.4.4. Proceso de consolidación y visualización de datos

Federación de modelos y extracción de información estructurada

El proceso de datos se inicia con la federación de los modelos generados por las distintas disciplinas involucradas en el proyecto. Esta federación permite coordinar la información proveniente de los modelos arquitectónicos, estructurales y de instalaciones, entre otros, integrándolos en una única representación coherente del activo de construcción. Una vez realizada, se procede a su revisión desde una perspectiva multidisciplinar con el fin de validar interferencias, omisiones o inconsistencias. Los resultados de dicha revisión son considerados por los equipos de diseño para actualizar sus modelos antes de la siguiente iteración.

Una vez consolidado y aprobado el modelo federado, se extrae la información estructurada directamente desde los archivos IFC utilizando configuraciones automatizadas que definen con precisión qué entidades y atributos deben ser considerados. Estas configuraciones permiten filtrar la información de manera coherente y reproducible, asegurando que los datos exportados se correspondan con los requerimientos del sistema de destino. A través de este mecanismo, los datos extraídos del modelo BIM pasan a formar parte del flujo de información hacia la base de datos relacional.

Importación, trazabilidad e integración de datos en SQL

Con los datos estructurados disponibles, se procede a su importación en la base de datos SQL. Cada entidad IFC es representada en forma de tabla, con campos que contienen tanto atributos básicos como propiedades extendidas. Esta organización explota de manera relacional los datos y habilita consultas complejas sobre el modelo digital del edificio.

Un aspecto fundamental del proceso es la trazabilidad. Para garantizar que cada elemento del modelo pueda ser rastreado de forma única a lo largo del tiempo, se emplea el identificador único global o GUID propio del esquema IFC. A diferencia de otros atributos, el GUID permanece inalterado incluso cuando se modifican las propiedades de un objeto, lo cual permite mantener el vínculo histórico entre los elementos del modelo y los registros almacenados en la base de datos.

Durante la importación, se aplican controles automáticos para verificar la integridad y consistencia de los datos. En caso de detectar errores o incongruencias, se activa un procedimiento de revisión que analiza el proceso de transferencia de datos entre el entorno BIM y la base de datos, con el objetivo de identificar la causa del problema y asegurar la calidad del conjunto de información importado.

Actualización continua y gestión del entorno de datos

La base de datos relacional no constituye un sistema estático, sino que se mantiene en constante actualización. A través de módulos automatizados de supervisión, se administra el flujo de datos que proviene del modelo federado y se asegura su correcta incorporación a la estructura del sistema. Esta gestión incluye tanto el seguimiento de cambios como la verificación del cumplimiento de estándares de calidad y completitud de la información.

Además, el sistema se conecta con cuadros de mando y herramientas de visualización que permiten a los responsables del proyecto realizar extracciones periódicas de datos estructurados, con el fin de alimentar análisis técnicos, económicos y operativos. Esta dinámica favorece un entorno colaborativo y transparente, en el que la información es accesible y verificable por todos los agentes del proyecto.

Este enfoque consolida una plataforma tecnológica escalable, capaz de adaptarse a los requerimientos específicos de distintos proyectos sin perder coherencia en su estructura general. Esta capacidad de evolución contribuye a la formación de un ecosistema corporativo de gestión de datos, donde la experiencia acumulada en cada intervención se convierte en un activo reutilizable en contextos futuros.

Además, al disponer de la información en un formato relacional y estructurado, se facilita la aplicación de metodologías de análisis avanzadas. Las organizaciones pueden realizar simulaciones complejas, detectar patrones de comportamiento, optimizar procesos constructivos o anticipar necesidades de mantenimiento. De este modo, la integración de los modelos BIM en una base de datos SQL no solo resuelve necesidades operativas inmediatas, sino que también habilita una estrategia de gestión del conocimiento más amplia, fundamentada en datos precisos, trazables y actualizados de forma continua.

3.5. Desarrollo de casos de uso

3.5.1. Recolección y estructuración de datos

En cuanto a la gestión de los datos de la construcción, el primer caso comienza con una recolección y pretratamiento de los datos brutos de la organización y de diferentes sistemas y fuentes de información habitualmente utilizados en el desarrollo de proyectos para normalizar los conjuntos de datos y proponer nuevas relaciones entre tablas.

La mayoría de los datos de partida que van a ser almacenados y gestionados desde la base de datos SQL van a estar relacionados con el esquema IFC, por lo que se recolecta y organiza la siguiente información:

- **Propiedades.** Directamente extraídas del esquema IFC, pudiendo añadir también otras dentro de conjuntos personalizados que no estén en el esquema.
- **Descripciones.** Cada propiedad tiene una descripción en la que se explica para que sirve o su objetivo principal.
- **Tipos de propiedades.** Cada propiedad pertenece a un tipo específico como texto, número, booleano, etc.
- **Conjuntos de propiedades.** Las propiedades del esquema IFC están definidas dentro de psets, aunque pueden existir otros creados por el usuario debido a necesidades concretas de proyecto.
- **Tipos de valores.** El valor de la propiedad puede ser simple, o bien puede tener condicionantes, como es el caso de valores predefinidos o enumeraciones.
- **Clases relacionadas.** Cada propiedad puede ser asignada a una o varias clases del esquema, pudiendo ser estas abstractas, de organización espacial o de sistemas, objetos o elementos tangibles.
- **Disciplinas.** Organización de la información en dominios concretos o bien clasificados dentro de los elementos de construcción compartidos.
- **Fases.** Cada propiedad puede trabajarse desde una o varias etapas del proyecto de edificación.
- **Usos BIM.** Según los usos requeridos para cada proyecto, existirán determinadas propiedades que serán necesarias.

A estas primeras capas de información, se les van añadiendo otras relacionadas con sistemas de clasificación, mapeos de entrada y salida de software BIM, traducciones entre español e inglés y otros datos de construcción relevantes.

Para esta preparación de los datos, es fundamental que exista una relación y entendimiento entre los tipos de datos existentes en las tablas CSV o XLSX, con los tipos permitidos dentro de las bases SQL y con los definidos en el esquema IFC. En el anexo B del presente estudio pueden revisarse las equivalencias entre todos ellos.

En un nivel inicial, y con el objetivo de cumplir las reglas de normalización de las bases de datos relacionales, se van generando tablas independientes con los registros de cada ítem según corresponda. Cada registro debe tener una columna que defina el identificador y otra con el nombre del ítem. Seguidamente, y dependiendo de la información requerida, existirán otros datos como descripciones o traducciones.

BUse_ID	BUse_Name	BNombre_Uso	BDescripcion_Uso
U00001	Design Model Authoring	Autoría de modelo de diseño	Materialización del proceso de diseño en un modelo digital tridimensional. Este modelo debe servir como la fuente principal para generar documentación, coordinación y comunicación.
U00002	BIM Coordination	Coordinación BIM	Verificación de la integración espacial de los elementos inter e intra disciplinares de los modelos bajo análisis, asegurando la coherencia y constructibilidad de la representación.
U00003	Constructive Simulation	Simulación constructiva	Proceso mediante el cual se añade información temporal a un modelo 3D respecto a cuándo los objetos que contiene son contruidos, instalados o desmontados, con el fin de validar cronogramas y gestionar espacios de ocupación.
U00004	Quantities and Budget	Mediciones y presupuesto	Uso de modelos BIM como fuente principal para obtener y verificar mediciones y presupuestos.
U000XX	...	...	...

Tabla 3.5.7. Extracto de registros de la tabla de usos BIM

Una vez que se definen las tablas de manera independiente, se generan otras basadas en estas primeras que relacionan los datos entre tablas. Dentro de las bases SQL, realmente serán los identificadores los que crearán las claves primarias y foráneas, por lo que será necesaria una conversión de los datos de cada registro para poder realizar estas relaciones.

Figura 3.5.1. Tabla con relaciones entre datos de construcción

Para ello, en este caso se genera una macro en la que se implementa un procedimiento que transforma las relaciones no estructuradas en una forma relacional para conectar las propiedades con las clases IFC. A partir de una hoja que contiene propiedades junto con un listado de clases asociadas, se recurre a otras hojas de referencia que vinculan tanto las propiedades como las clases IFC con identificadores únicos. A través de este cruce de datos, y descomponiendo las relaciones múltiples, se genera una nueva hoja con una tabla plana donde cada fila representa una relación individual entre una propiedad y una clase, utilizando sus respectivos códigos únicos.



Como resultado, se obtiene una nueva hoja con todas las relaciones entre IDs de las propiedades con sus respectivas clases IFC.

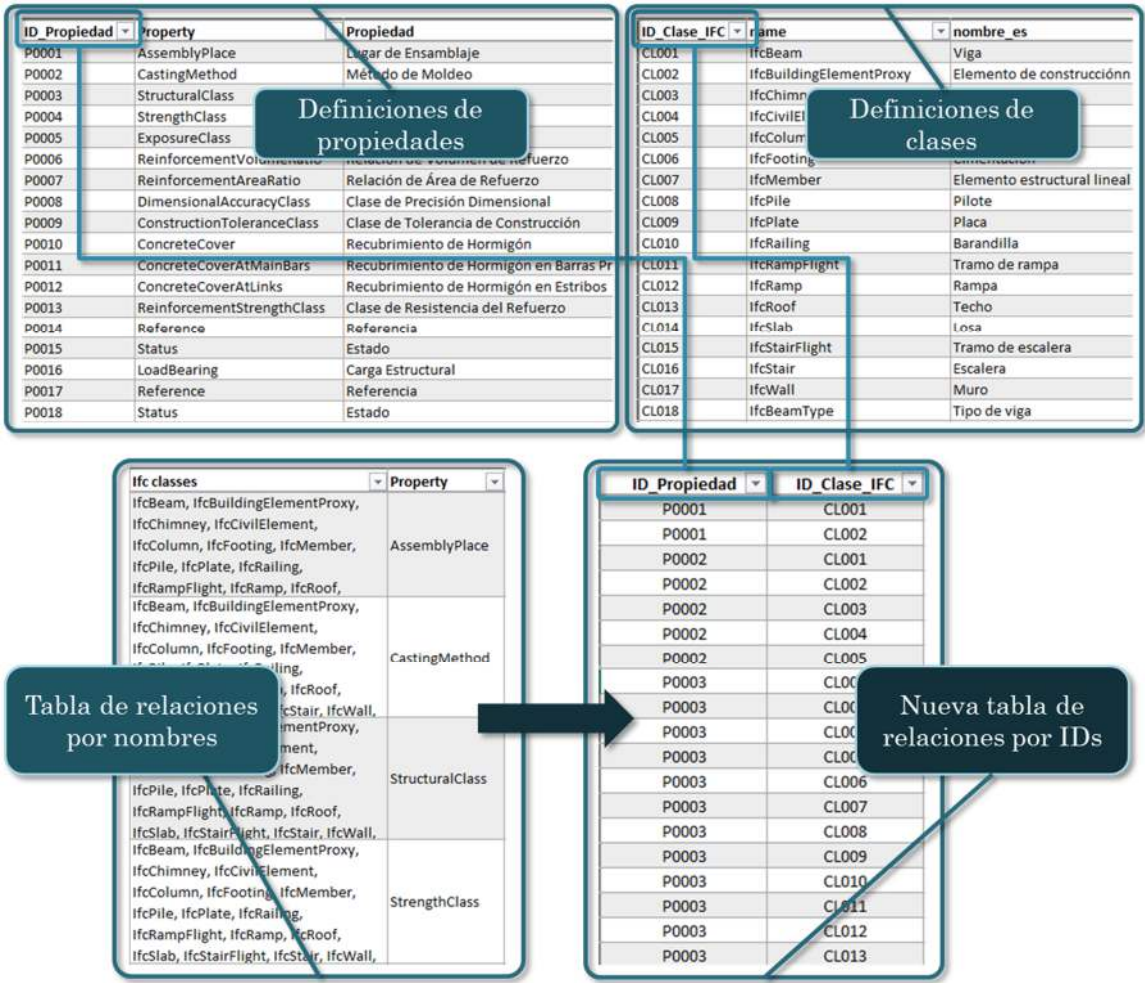


Figura 3.5.3. Generación de tabla de relaciones por IDs

En el caso de los sistemas de clasificación, en muchas de las versiones actuales las propias tablas del sistema ya contienen asignaciones para clases y tipos predefinidos IFC para cada ítem, por tanto, se puede realizar el mismo proceso, en este caso, para relacionar cada código de clasificación con el ID de la clase y el tipo asignado. La relación es de 1-1, por lo que las relaciones pueden generarse en una misma tabla de manera conjunta.

id_clasificacion	codigo_clasificacion	descripcion_clasificacion	id_clase_ifc	id_tipo_ifc
P001	FUN.10.10	Pilar	CL0101	TY1001
P002	FUN.10.20	Losa	CL0102	TY1002
P003	FUN.10.30	Viga	CL0103	TY1003
P004	FUN.10.40	Escalera	CL0104	TY1004
P005	FUN.10.50	Rampa	CL0105	TY1005

Tabla 3.5.8. Relación entre clasificaciones con clases y tipos IFC

3.5.2. Configuración de la base de datos

Para el desarrollo de este caso, se selecciona MariaDB como base de datos relacional debido a su rendimiento, estabilidad y su naturaleza de código abierto, lo que permite una compatibilidad total con otras bases de datos como MySQL. Se utiliza la herramienta phpMyAdmin de la plataforma de desarrollo XAMPP para la administración general de datos, ofreciendo una interfaz altamente intuitiva que simplifica las tareas complejas de configuración y gestión de datos. Además, se estableció Apache como servidor para gestionar las solicitudes del navegador web.

Para la configuración inicial de la base de datos, se crea un esquema inicial para almacenar la información, seguido de la configuración de usuarios y permisos para gestionar el acceso a los datos. Estos usuarios se configuran en base a los perfiles desarrollados en el apartado de roles de la propuesta de sistematización, generando un usuario para cada perfil.

Vista global de las cuentas de usuario

☐	Nombre de usuario	Nombre del servidor	Contraseña	Privilegios globales	Grupo de usuario	Conceder
☐	Administrador de gestión de datos global	%	Sí	ALL PRIVILEGES	Gestores y analistas	Sí
☐	Analista de Datos	%	Sí	SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, FILE, INDEX, ALTER, CREATE TEMPORARY TABLES, CREATE VIEW, EVENT, TRIGGER, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EXECUTE		No
☐	Auditor Digital BIM	%	Sí	SELECT, INSERT, UPDATE, DELETE, FILE		No
☐	Coordinador BIM	%	Sí	SELECT, INSERT, UPDATE, DELETE, FILE	Usuarios BIM	No
☐	Especialista en Interoperabilidad	%	Sí	SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, FILE, REFERENCES, INDEX, ALTER, SHOW DATABASES, CREATE TEMPORARY TABLES, CREATE VIEW, EVENT, TRIGGER, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EXECUTE		No
☐	Gestor BIM de datos	%	Sí	SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, FILE, INDEX, ALTER, CREATE TEMPORARY TABLES, CREATE VIEW, EVENT, TRIGGER, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, EXECUTE		No
☐	Líder de disciplina BIM	%	Sí	SELECT, INSERT, UPDATE, DELETE, FILE	Usuarios BIM	No

Figura 3.5.4. Gestión de usuarios y permisos de la BBDD

Los campos del esquema de MariaDB incluyen nombres, descripciones, tipos de datos permitidos, longitud de caracteres, valores nulos e identificaciones de clave. Los registros que se incluyan en un futuro y que no cumplan estos criterios serán rechazados, lo cual mejorará la calidad de la información en comparación con los métodos tradicionales no estructurados.

En primer lugar, se desarrolla un esquema de base de datos relacional denominado *gest_0000_datos_empresa*, diseñado específicamente para representar de forma estructurada los procesos administrativos y operativos típicos de una empresa del sector de la construcción.

La creación de este esquema se lleva a cabo mediante código, utilizando el conector *mysql-connector-python* para interactuar con el sistema de gestión de bases de datos y utilizando Anaconda y Jupyter Notebook para desarrollar scripts de creación de esquemas, de tablas e importación de registros.


```

# Importación de librerías
import mysql.connector
from mysql.connector import errorcode

# Configuración de conexión
config = {'user': 'XXXXXX', 'password': 'XXXXXX', 'host':
'localhost', 'raise_on_warnings': True}

# Conectar a MySQL
cnx = mysql.connector.connect(**config)
cursor = cnx.cursor()

# Crear esquema
cursor.execute("CREATE DATABASE IF NOT EXISTS gest_0000_datos_empresa
DEFAULT CHARACTER SET 'utf8mb4'")
cursor.execute("USE gest_0000_datos_empresa")

# Crear las tablas
cursor.execute("""
CREATE TABLE IF NOT EXISTS clientes (id_cliente INT AUTO_INCREMENT
PRIMARY KEY, nombre VARCHAR(100), email VARCHAR(100), telefono
VARCHAR(20), direccion TEXT, tipo_cliente ENUM('público','privado'))""")

...

# Finalizar
cnx.commit()
cursor.close()
cnx.close()

```

El diseño de la base de datos identifica y modela los componentes generales de la organización, como son clientes, proyectos, empleados, maquinaria, materiales, compras, asignaciones de personal, uso de maquinaria y pagos. Cada uno de estos aspectos se representa mediante tablas interrelacionadas mediante claves foráneas, respetando principios de normalización y buenas prácticas de integridad referencial. La tabla proyectos, por ejemplo, se relaciona con clientes, empleados, materiales y maquinaria, permitiendo una trazabilidad de recursos humanos, materiales y financieros involucrados en cada obra.

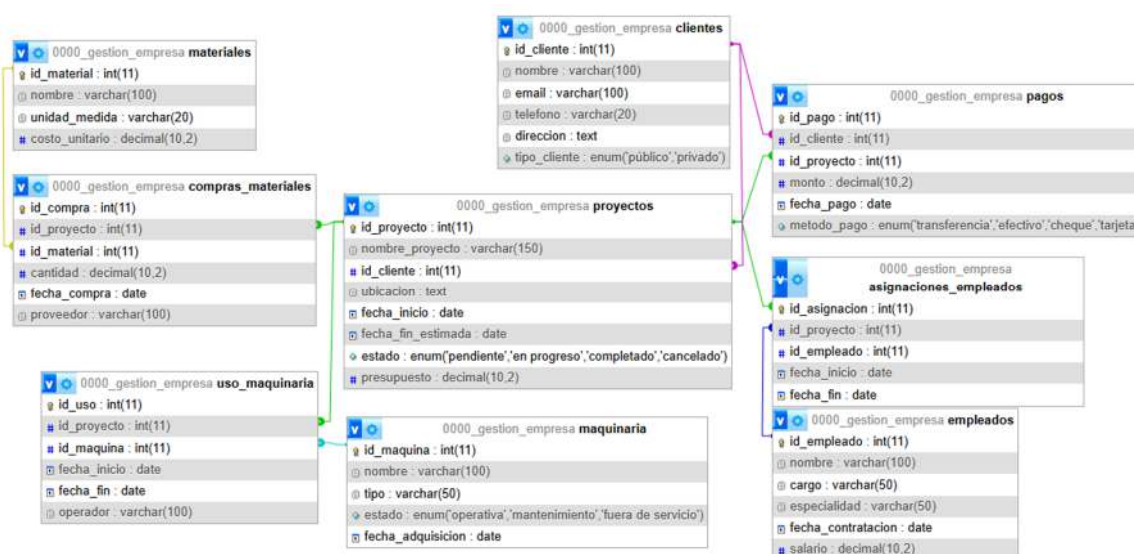


Figura 3.5.5. Esquema de datos de gestión de la organización

En segundo lugar, se integran los datos estructurados de construcción y se establecen asociaciones entre usos, fases, requisitos, sistemas de clasificación, disciplinas, clases, tipos, conjuntos de propiedades y propiedades, seguidas de relaciones de nivel inferior como unidades y tipos de datos. Este enfoque busca definir los términos de la organización a todos los niveles y vincularlos con definiciones de terceros de sistemas como GuBIMClass, Omniclass y BIMForum, así como de las versiones 2x3, 4 y 4.3 de IFC. Esta personalización adapta las estructuras de las tablas a las necesidades de la organización, lo que permite una escalabilidad fácil y rápida.

Para este subproceso, se toma una muestra representativa de las propiedades IFC de cada disciplina o dominio del esquema y, una vez estructurados y organizados los datos, se transfieren a la base de datos SQL mediante rutinas de automatización. Para casos en los que los datos se encuentren dentro de archivos CSV, hay que generar el código estableciendo primero la conexión con la base de datos, extraer luego los datos del archivo y, por último, incluirlos en las tablas correspondientes.

```
# Conexión a la base de datos
cnx = mysql.connector.connect(user=XXXX, password=XXXX,
    host='localhost', database='gest_0002_datos_ifc')
cursor = cnx.cursor()
def sanitize(text):
    if text is None:
        return ""
    return text.replace("'", "")
# Para evitar duplicados en la sesión
properties_cache = {}
with open('propiedades.csv', newline="", encoding='utf-8-sig') as csvfile:
    reader = csv.DictReader(csvfile)
    reader.fieldnames = [f.strip() for f in reader.fieldnames]
    for row in reader:
        prop_name = row['Property'].strip()
        translated_name = row.get('Propiedad', "").strip()
        description = row.get('Description', "").strip()
        if prop_name in properties_cache:
            property_id = properties_cache[prop_name]
        else:
            # Verificar si la propiedad ya existe en la DB
            cursor.execute("SELECT id FROM properties WHERE name = %s",
                (prop_name,))
            result = cursor.fetchone()
            if result:
                property_id = result[0]
            else:
                # Insertar propiedad
                cursor.execute(
                    "INSERT INTO properties (name, translated_name, description)
                    VALUES (%s, %s, %s)",
                    (prop_name, translated_name, description))
                cnx.commit()
```

```

    property_id = cursor.lastrowid
    properties_cache[prop_name] = property_id
    # Insertar relaciones con ifc_classes
    ifc_classes = [cls.strip() for cls in row['Ifc classes'].split(',')]
    for ifc_class in ifc_classes:
        cursor.execute("""
            INSERT INTO property_ifc_classes (property_id, ifc_class_id)
            VALUES (%s, (SELECT id FROM ifc_classes WHERE name = %s))
            """, (property_id, ifc_class))
    cnx.commit()
    cursor.close()
    cnx.close()

```

Una vez que se finaliza el proceso, los datos pueden ser revisados dentro de la tabla de referencia. En el caso de que existan errores, estos pueden deberse a la estructura del CSV, errores en el nombrado de los encabezados, valores nulos o tipos de datos incorrectos en alguna columna.

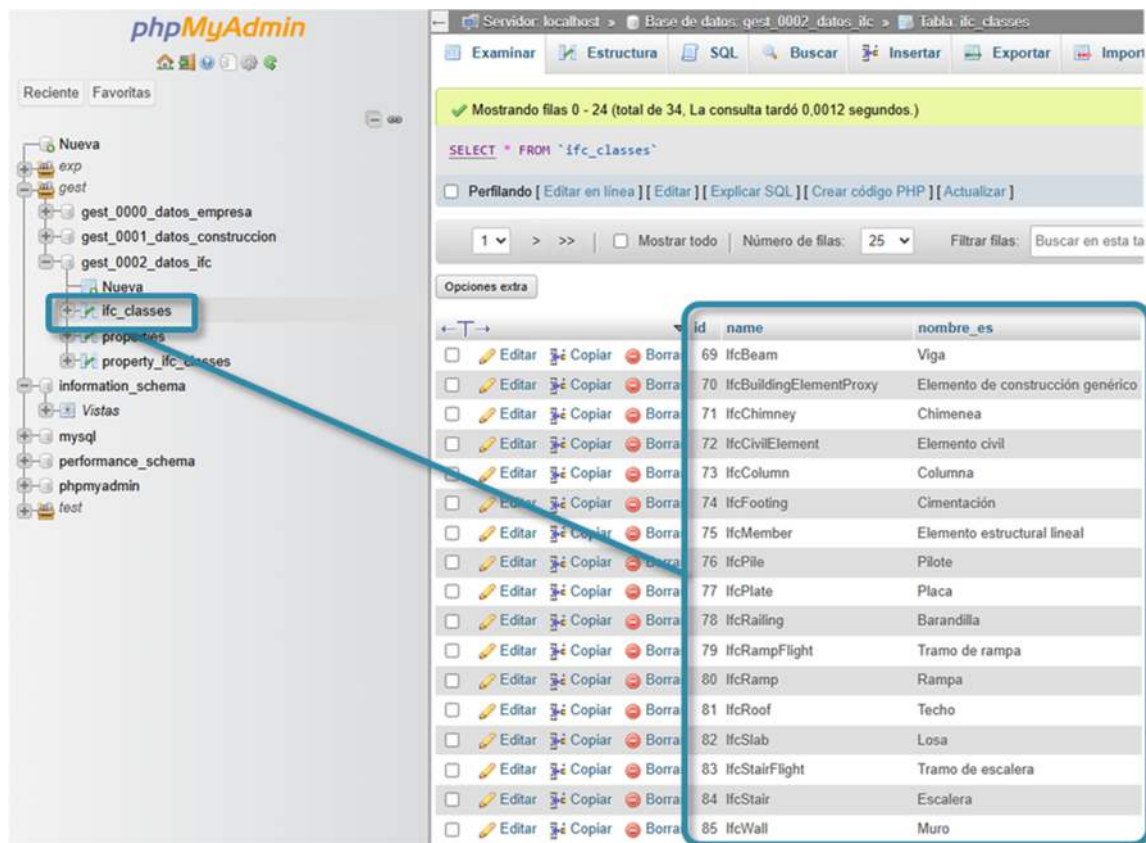


Figura 3.5.6. Importación de datos tabulares a la BBDD SQL

Un apunte importante es que es fundamental utilizar dentro de estos procesos de transformación de datos, librerías de apoyo para poder gestionar correctamente los tipos de datos y los procesos de conexión. Algunas de estas librerías necesarias pueden ser:

- Conexión y manejo de bases de datos: las librerías *pandas.io.sql* y *sqlalchemy* permiten interactuar con bases de datos SQL, facilitando la ejecución de

consultas y la transferencia eficiente de datos hacia y desde estructuras de Python como DataFrames. Por su parte, *pymysql* es un conector específico para bases de datos MySQL y MariaDB que permite establecer la comunicación entre Python y el motor de base de datos, posibilitando realizar consultas, inserciones y actualizaciones directamente desde código Python.

- Manipulación y análisis de datos: *pandas* es esencial para la manipulación y análisis de datos en estructuras tabulares, ya que proporciona herramientas para limpiar, transformar y resumir conjuntos de datos. Por otro lado, *numpy* complementa estas capacidades a partir del soporte para cálculos numéricos avanzados mediante matrices multidimensionales y funciones matemáticas.
- Manejo de fechas y tiempos: para trabajar con fechas y horas en Python, se puede utilizar el módulo *datetime*, que incluye las clases *datetime* y *date*. Estas permiten crear, manipular y formatear objetos temporales con precisión, facilitando tareas como calcular diferencias entre fechas, filtrar datos por rangos temporales o convertir cadenas de texto en fechas.

Una manera más automática de realizar estos procesos de carga de datos en la BBDD es a partir del uso de archivos XLSX, los cuales pueden extraer la información de varias hojas a la vez para realizar el volcado de datos de una manera más eficiente. En el siguiente código se extrae la información de la primera hoja del archivo para enviar los datos a la base SQL.

```
# Lectura del archivo XLSX
FilePath= 'C:\\Users\\Samuel\\Desktop\\Requisitos_info.xlsx'
xlsx=pd.ExcelFile(FilePath)
# Conexión con la base de datos
database_username = 'XXXX'
database_password = 'XXXX'
database_ip      = 'localhost'
database_name    = 'test_data_repo'
database_connection = db.create_engine('mysql+pymysql://{0}:{1}@{2}/{3}'.
                                     format(database_username, database_password,
                                     database_ip, database_name))
# Importación de los datos en la BBDD
connection = database_connection.connect()
metadata = db.MetaData()
for sheet_name in xlsx.sheet_names:
    print(sheet_name)
    mysql_table = 'arq_' + sheet_name
    print('Table - ' + mysql_table)
    df=pd.read_excel(FilePath, sheet_name=sheet_name, index_col=None)
    df.to_sql(mysql_table, connection, if_exists='append', index=False)
connection.close()
```

Previamente a este volcado, las claves primarias y foráneas se han definido para establecer vínculos estructurados entre diferentes tablas, garantizando la integridad referencial y facilitando la navegación y consulta de la información. La clave primaria especifica de manera única cada registro, mientras que la clave foránea actúa como un enlace que referencia esa identidad en otra tabla.

Finalmente, se integran el resto de tablas con datos generales y específicos de construcción, generando así la base de datos de la organización de la cual se irán extrayendo más adelante los tipos de datos según las necesidades concretas de cada proyecto de edificación.

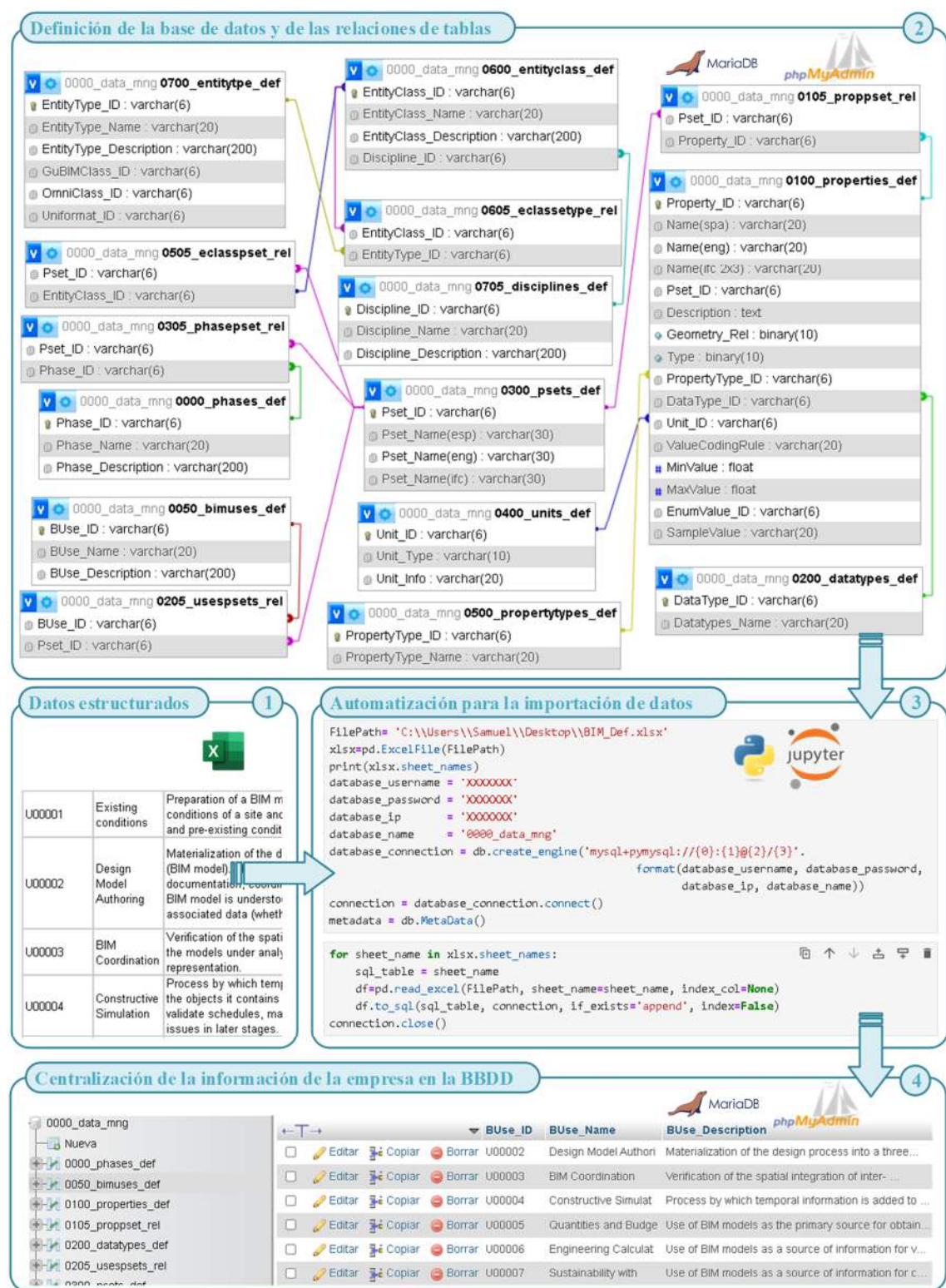


Figura 3.5.7. Configuración de la base de datos SQL

3.5.3. Administración de la base de datos

En este caso de uso se desarrollan diferentes aspectos relevantes relacionados con la gestión y administración de la base de datos, buscando un correcto funcionamiento, integridad y eficiencia a largo plazo. Esta administración de la base no se limita únicamente al almacenamiento de información, sino que incluye la implementación de herramientas y procedimientos que garanticen la coherencia semántica de los datos, la facilidad de acceso para distintos usuarios y la automatización de reglas lógicas conectadas al modelo de información.

Con respecto al control y gestión de los registros que van siendo generados en las tablas, es de gran utilidad incluir atributos adicionales referidos al estado y validez de cada dato. Siguiendo este objetivo, se ha procedido a la incorporación de un conjunto de campos comunes en todas las tablas principales del modelo relacional. Esta ampliación estructural incluye cuatro columnas clave: un campo de estado denominado “Activo” que determina si un registro está vigente u obsoleto; un campo de autoría denominado “Creado por”, que permite identificar al responsable de la creación del dato; y dos campos temporales referidos a la fecha de creación y la fecha de actualización de los datos. Estos elementos, añadidos de manera homogénea en todas las tablas, permiten establecer una base sólida para la trazabilidad, el control de versiones y la auditoría de los datos a lo largo del ciclo de vida del proyecto.

```
ALTER TABLE propiedades
ADD COLUMN activo BOOLEAN NOT NULL DEFAULT TRUE,
ADD COLUMN creado_por VARCHAR(100) DEFAULT 'XXXX',
ADD COLUMN fecha_creacion DATETIME NOT NULL DEFAULT
CURRENT_TIMESTAMP,
ADD COLUMN fecha_actualizacion DATETIME NOT NULL DEFAULT
CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP;
```

La implementación de estos campos responde a la necesidad de fortalecer la transparencia y la responsabilidad en la manipulación de los datos, asegurando que cualquier modificación pueda ser rastreada y justificada dentro de un marco de gobernanza de datos. Este enfoque facilita la administración de los registros y también permite desarrollar funciones avanzadas como la validación por parte de usuarios autorizados, el archivado de registros obsoletos sin necesidad de eliminarlos físicamente, y la integración con mecanismos de seguridad o análisis histórico.

id	nombre	activo	creado_por	fecha_creacion	fecha_actualizacion
1	Height	1	admin	17/05/2025 18:20	17/05/2025 18:20
2	Width	1	admin	17/05/2025 18:20	17/05/2025 18:20
3	Material	1	admin	17/05/2025 18:20	17/05/2025 18:21
4	FireRating	1	admin	17/05/2025 18:20	17/05/2025 18:20
5	LoadBearing	1	admin	17/05/2025 18:20	17/05/2025 18:20

Tabla 3.5.9. Campos generados para el control de registros

Además, las consultas SQL desempeñan un papel central como instrumento para la extracción, análisis y manipulación de la información almacenada. La sintaxis estructurada de este lenguaje permite realizar desde operaciones básicas, como selecciones, filtrados y ordenamientos, hasta consultas avanzadas que implican la combinación de múltiples tablas mediante uniones, agrupamientos y agregaciones. En el caso de la base de datos desarrollada para el entorno de construcción, estas consultas permiten generar tablas semánticamente enriquecidas que revelan patrones de asociación entre diferentes tipos de datos de la organización, facilitando su interpretación desde una perspectiva técnica y de gestión.

La siguiente consulta SQL dentro de phpMyAdmin extrae los datos de tres tablas relacionadas, unificando la información. Partiendo de la tabla de propiedades, la de clases IFC y la que establece las relaciones entre ambas, se realiza una agrupación de las clases asociadas a cada propiedad en una sola celda utilizando la función *GROUP_CONCAT*, permitiendo obtener un resumen de las propiedades y todas las clases IFC a las que están vinculadas.

```

SELECT
    p.id AS id_propiedad,
    p.nombre AS nombre_propiedad,
    GROUP_CONCAT(c.nombre ORDER BY c.nombre SEPARATOR ',') AS
    clases_relacionadas
FROM
    relaciones_relaciones r
JOIN
    relaciones_propiedades p ON r.property_id = p.id
JOIN
    relaciones_ifc_classes c ON r.ifc_class_id = c.id
GROUP BY
    p.id, p.nombre
ORDER BY
    p.nombre;
```

Como resultado, se unifican los datos de la tabla de propiedades y de clases IFC, generando así una visión consolidada para tareas de análisis semántico o verificación de reglas y relaciones entre datos.

id_propiedad	nombre_propiedad	clases_relacionadas
4	FireRating	IfcDoor, IfcWindow
1	Height	IfcDoor, IfcSlab, IfcWall, IfcWindow
5	LoadBearing	IfcBeam, IfcSlab, IfcWall
3	Material	IfcBeam, IfcDoor, IfcWall
2	Width	IfcDoor, IfcWall, IfcWindow

Tabla 3.5.10. Unificación de datos de propiedades en tabla conjunta

Esta consulta SQL puede ser además guardada como una vista para futuros análisis. Las vistas son representaciones lógicas y virtuales de los datos, derivadas de una o

varias tablas mediante una consulta SQL definida previamente. Las vistas no almacenan físicamente los datos, sino que se actualizan dinámicamente en función de las tablas subyacentes, lo que permite abstraer la complejidad estructural del modelo de datos y facilitar el acceso a información relevante.

En este caso de uso, que se orienta a la gestión de datos, las vistas desempeñan un papel importante para la simplificación de consultas complejas, la mejora de la seguridad, y la estandarización de la presentación de resultados. Así, aunque las tablas de clases y propiedades vayan siendo ampliadas, esta vista generada irá mostrando siempre las versiones más recientes de los datos relacionados.

CREATE VIEW *vista_propiedades_agrupadas*

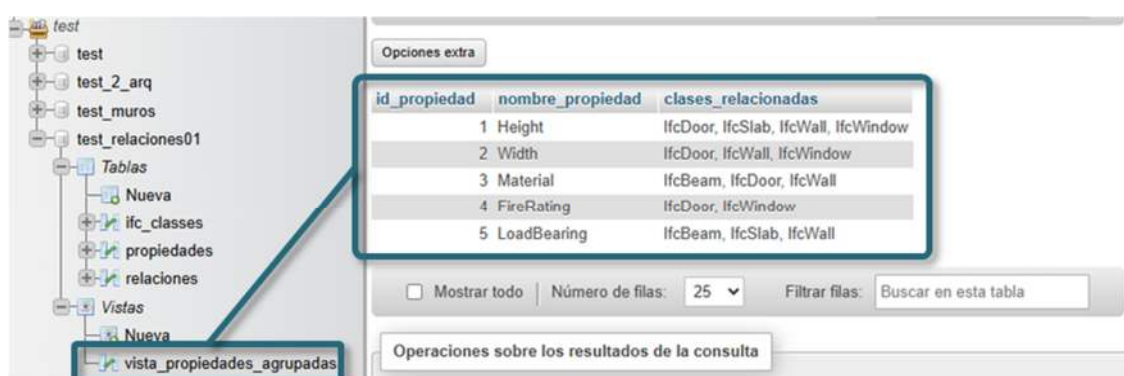


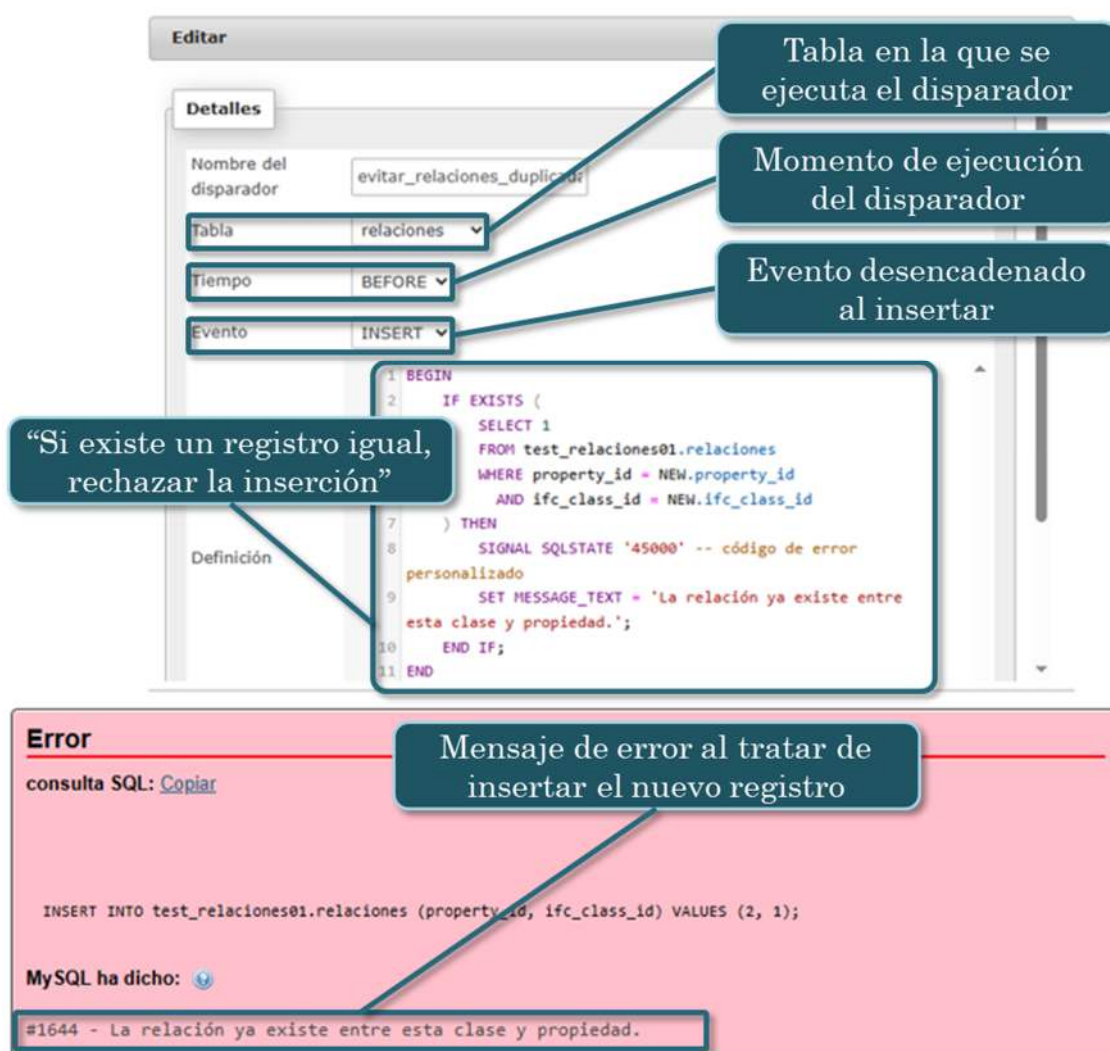
Figura 3.5.8. Creación de vistas dentro de la base de datos

Como se ha mostrado en otros casos, estas consultas y operaciones de datos también pueden ser realizadas desde entornos externos de programación como Jupyter Notebook, pudiendo conectar la información con otros entornos o plataformas.

```
import mysql.connector
import pandas as pd
# Conexión a la base de datos
conn = mysql.connector.connect(host='localhost', user='XXXX',
                               password='XXXX', database='relaciones')
# Consulta SQL
query = """
SELECT
    p.id AS id_propiedad,
    p.nombre AS nombre_propiedad,
    GROUP_CONCAT(c.nombre ORDER BY c.nombre SEPARATOR ', ') AS
    clases_relacionadas
FROM
    relaciones r
"""
# Ejecutar y mostrar resultados
df = pd.read_sql(query, conn)
display(df)
# Cerrar conexión
conn.close()
```

Por otro lado, los disparadores resultan de utilidad para controlar diferentes aspectos de la base de datos, relacionados especialmente con la organización y normalización de cada una de las tablas. Estos disparadores o triggers son un mecanismo que ejecuta automáticamente un bloque de código cuando ocurre un evento específico, como una inserción, actualización o eliminación en una tabla.

En el siguiente caso, se ha creado un disparador que se activa antes de insertar un nuevo registro en la tabla de relaciones, con el objetivo de garantizar la integridad de los datos evitando la duplicación de relaciones entre clases IFC y propiedades. El disparador verifica si ya existe una fila con la misma combinación de ID de propiedad e ID de clase IFC. Si encuentra una coincidencia, detiene la inserción lanzando un error personalizado, lo que previene la creación de registros duplicados que podrían causar inconsistencias en el modelo de datos.



The image shows a database management interface for creating a trigger. The trigger is named 'evitar_relaciones_duplicadas' and is associated with the 'relaciones' table. It is configured to execute 'BEFORE' the 'INSERT' event. The trigger's definition is as follows:

```

1 BEGIN
2   IF EXISTS (
3     SELECT 1
4     FROM test_relaciones01.relaciones
5     WHERE property_id = NEW.property_id
6           AND ifc_class_id = NEW.ifc_class_id
7   ) THEN
8     SIGNAL SQLSTATE '45000' -- código de error
9     personalizado
10    SET MESSAGE_TEXT = 'La relación ya existe entre
11    esta clase y propiedad.';
12   END IF;
13 END

```

Annotations on the interface include:

- Tabla en la que se ejecuta el disparador:** Points to the 'relaciones' table selection.
- Momento de ejecución del disparador:** Points to the 'BEFORE' time selection.
- Evento desencadenado al insertar:** Points to the 'INSERT' event selection.
- “Si existe un registro igual, rechazar la inserción”:** Points to the trigger logic that checks for existing records and signals an error.
- Mensaje de error al tratar de insertar el nuevo registro:** Points to the error message displayed in the MySQL console: '#1644 - La relación ya existe entre esta clase y propiedad.'

The error message is shown in a pink box with the text: 'Error', 'consulta SQL: Copiar', and the SQL statement: 'INSERT INTO test_relaciones01.relaciones (property_id, ifc_class_id) VALUES (2, 1);'. Below this, it says 'MySQL ha dicho:' followed by the error message.

Figura 3.5.9. Disparador para el control de registros duplicados

3.5.4. Datos contractuales y del CDE

Tanto los requisitos de información exigidos por la organización, como el BEP del proyecto y el Entorno Común de Datos, contienen datos estructurados que se organizan en diferentes apartados y que pueden ser gestionados desde la base de datos relacional. Esto es de gran utilidad ya que pueden generarse relaciones entre el esquema general de gestión de la empresa con el esquema específico del proyecto que se esté desarrollando.

Como punto de partida, cualquier proyecto BIM debe definir aspectos como los agentes involucrados, el plan de entrega de información máster, los usos requeridos o la matriz de atributos, entre muchos otros. Toda esta información puede ser estructurada en formatos tabulares y ser gestionada desde la base de datos.

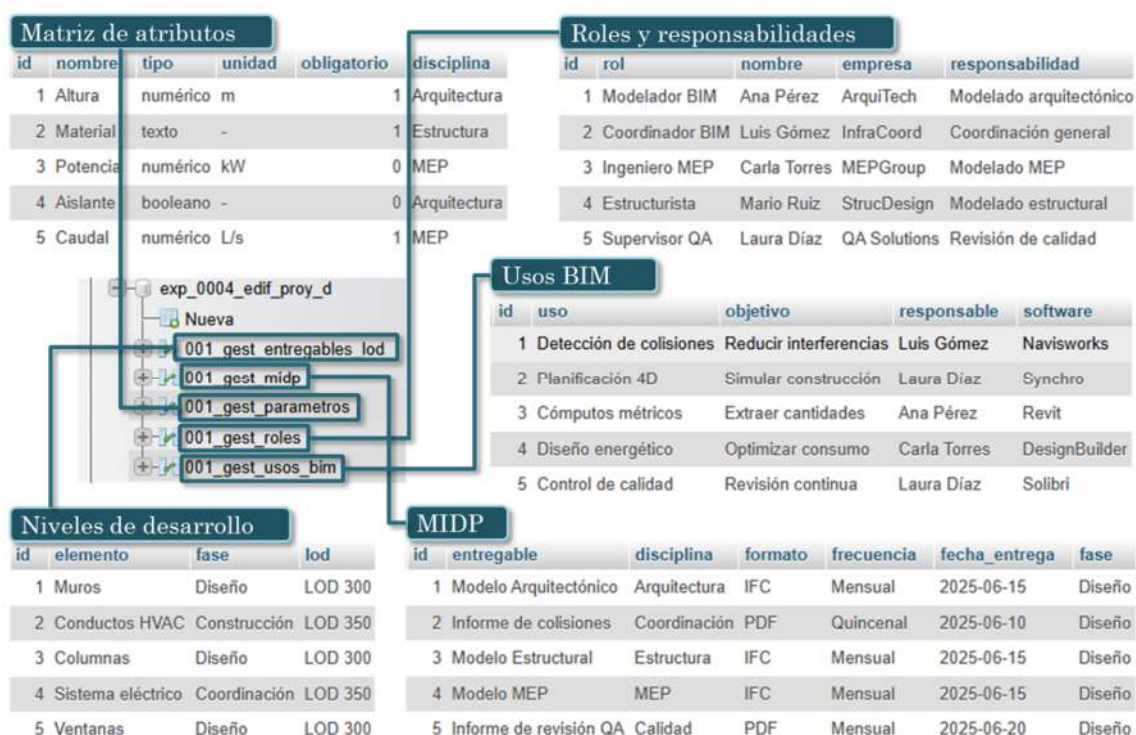


Figura 3.5.10. Tablas relacionadas con el Plan de Ejecución BIM

De este modo, la base de datos se posiciona como la fuente original y principal de la información del proyecto, manteniéndose siempre actualizada dependiendo de los cambios que puedan irse desarrollando en el proyecto, como sustituciones de agentes, modificados de contrato con nuevos usos o nuevos atributos, ampliación de entregables dependientes de la segregación y el tamaño de los modelos BIM, etc. Además, y al igual que sucede en casos de uso anteriores, estas tablas pueden relacionarse para mantener la normalización y evitar la duplicidad de información. Por ejemplo, la tabla de Niveles de Desarrollo de objetos BIM puede relacionarse con los usuarios responsables de llevar a cabo la ejecución de cada tipo o clase de objeto, a partir de la relación de los IDs de estos responsables con cada uno de los registros de la tabla de Niveles de Desarrollo.

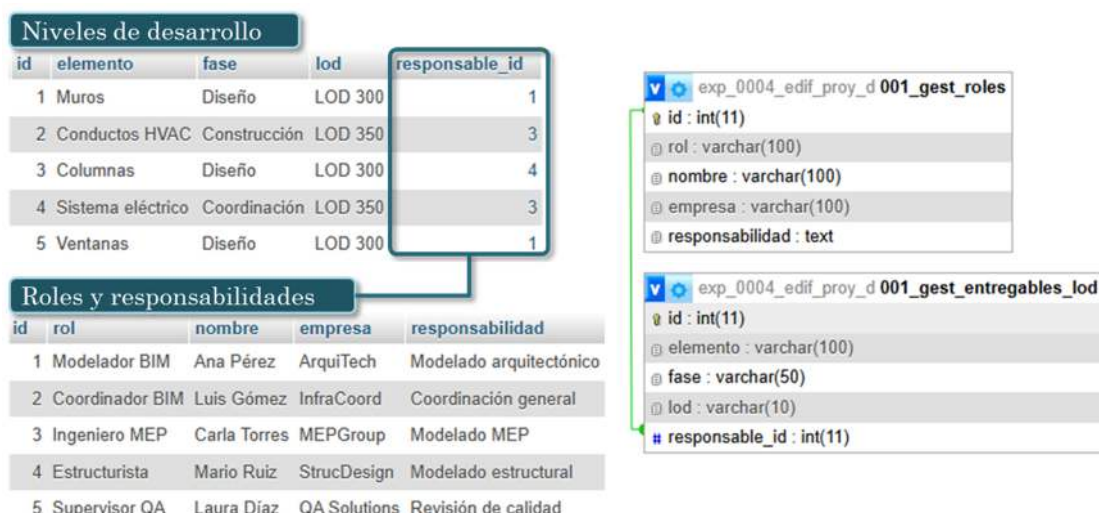


Figura 3.5.11. Relación entre tabla LOD y tabla de roles BIM

Para generar la tabla solo con los datos que interesan, la cual será la que se añada en el Plan de Ejecución BIM, se crea una vista en la que solo se extraigan los datos relevantes de ambas tablas, sustituyendo el ID del responsable por su nombre y ocultando el ID de cada clase de objeto.

```

CREATE VIEW vw_001_gest_entregables AS
SELECT
    e.elemento,
    e.fase,
    e.lod,
    r.nombre AS responsable
FROM
    001_gest_entregables_lod e
JOIN
    001_gest_roles r ON e.responsable_id = r.id;
  
```

elemento	fase	lod	responsable
Muros	Diseño	LOD 300	Ana Pérez
Conductos HVAC	Construcción	LOD 350	Carla Torres
Columnas	Diseño	LOD 300	Mario Ruiz
Sistema eléctrico	Coordinación	LOD 350	Carla Torres
Ventanas	Diseño	LOD 300	Ana Pérez

Tabla 3.5.11. Vista LOD final a incorporar en el BEP

Una vez configuradas todas las vistas y tablas necesarias, se exportan desde Jupyter Notebook mediante código Python a un archivo XLSX del cual luego pueden ser añadidas dentro del Plan de Ejecución BIM del proyecto. Para ello, es necesario trabajar con el conector de SQL y la librería de pandas para gestión de tablas.


```

import mysql.connector
import pandas as pd
#Conexión a la base de datos
conn = mysql.connector.connect(host="localhost", user="XXXX",
    password="XXXX", database="exp_0004_edif_proy_d")
#Obtener nombres de tablas 001_gest_ excluyendo la de entregables
tables_query = """
SELECT table_name
FROM information_schema.tables
WHERE table_schema = 'exp_0004_edif_proy_d'
AND table_name LIKE '001_gest_%'
AND table_name != '001_gest_entregables_lod';"""
tables = pd.read_sql(tables_query, conn)['table_name'].tolist()
#Agregar la vista final como reemplazo
tables.append('vw_001_gest_entregables_con_responsables')
#Exportar a Excel
output_path = "exp_0004_edif_proy_d.xlsx"
with pd.ExcelWriter(output_path, engine="openpyxl") as writer:
    for table in tables:
        df = pd.read_sql(f"SELECT * FROM `{table}`", conn)
        sheet_name = table[:31]
        df.to_excel(writer, sheet_name=sheet_name, index=False)
#Cerrar conexión
conn.close()
print(f'Archivo generado: {output_path}')

```

Una opción adicional en relación a actualizaciones de estas tablas dentro del BEP, es la vinculación de las mismas dentro del documento de Word mediante vínculos dinámicos. Al sobrescribir el archivo Excel con una versión actualizada, los datos en el documento Word también se actualizarán. Cada vez que exista una nueva versión de las tablas, al abrir el documento, aparecerá un aviso para actualizar los vínculos, y al aceptarlo, las tablas insertadas reflejarán la información más reciente contenida en el Excel.

1.2 Usos BIM

La correcta definición de los Usos BIM es un componente clave para el desarrollo eficiente de un proyecto bajo metodología BIM. Este apartado establece qué usos se aplicarán durante las distintas fases del ciclo de vida del proyecto, especificando su propósito, responsables, entregables y nivel de desarrollo requerido. La identificación y planificación de los Usos BIM permite al equipo alinearse en torno a objetivos comunes, maximizar el valor del modelado y garantizar una gestión de la información coherente con los requerimientos del cliente y los estándares del proyecto. A continuación, se detallan los usos seleccionados para este proyecto, su justificación y su implementación prevista en el entorno colaborativo.

id	uso	objetivo	responsable	software
1	Detección de colisiones	Reducir interferencias	Luis Gómez	Navisworks
2	Planificación 4D	Simular construcción	Laura Díaz	Synchro
3	Cómputos métricos	Extraer cantidades	Ana Pérez	Revit

Figura 3.5.12. Tabla vinculada al documento del BEP

Con respecto a los datos de gestión de proyectos que van siendo generados dentro del Entorno Común de Datos, todos ellos también pueden ser integrados dentro de la BBDD si se exportan periódicamente en formatos tabulares y posteriormente se integran de manera estructurada como nuevas tablas dentro del esquema específico del proyecto. Si bien es cierto que estos entornos ya poseen herramientas para gestionar estos datos a partir del uso de diferentes atributos para cada registro y filtros y agrupaciones dependientes de estos atributos, la incorporación de estos datos a la BBDD puede generar otras relaciones más avanzadas entre diferentes capas y niveles de información del proyecto, aumentando la calidad, el control y la gestión del activo.

Autodesk Construction Cloud (ACC) es un ejemplo robusto de CDE que incluso ofrece soluciones específicas que conectan directamente los datos del entorno con la base de datos como MySQL. No obstante, estos desarrollos conllevan un coste considerable y necesitan actualizaciones periódicas dependiendo de las necesidades y actualizaciones que vayan surgiendo dentro de los esquemas de la BBDD.

Por ello, en este caso de estudio no se considera este tipo de soluciones complejas, sino que se gestionan los datos a partir de la extracción en formatos como XLSX para posteriormente, integrarlos dentro del esquema del proyecto.

Dentro de este entorno de ACC, es posible exportar diferentes tablas relacionadas con el registro de archivos, las incidencias registradas, los informes de transmisión, las revisiones realizadas y pendientes, los miembros del proyecto y otros registros relacionados con formularios, solicitudes de información o planificaciones de obra. Esta exportación puede realizarse de manera manual para luego automatizar la entrada de los datos dentro del proyecto de la BBDD. Cada tabla puede guardarse con una fecha o hito específico para luego poder llevar a cabo una relación temporal.



Mostrando filas 0 - 14 (total de 15, La consulta tardó 0,0004 segundos.)

`SELECT * FROM 'h01_registro_archivos_acc'`

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25

Tabla de registro de archivos cargados en ACC correspondientes al Hito 01 del proyecto

nombre_archivo	ruta_archivo	version	creador	fecha_carga	ultima_modificacion	estado_revision	tamano
Plano_001.pdf	/Documentos/Arquitectura/Plantas/Plano_001.pdf	4	carla torres	2025-02-20 00:00:00	2025-03-02 00:00:00	Aprobado	2223 KB
Plano_002.pdf	/Documentos/Arquitectura/Plantas/Plano_002.pdf	3	luis garcia	2024-11-30 00:00:00	2024-12-08 00:00:00	En revisión	1548 KB
Plano_003.pdf	/Documentos/Arquitectura/Plantas/Plano_003.pdf	4	luis garcia	2024-12-11 00:00:00	2024-12-20 00:00:00	En revisión	1549 KB
Plano_004.pdf	/Documentos/Arquitectura/Plantas/Plano_004.pdf	1	luis garcia	2025-02-21 00:00:00	2025-02-25 00:00:00	Rechazado	1563 KB
Plano_005.pdf	/Documentos/Arquitectura/Plantas/Plano_005.pdf	5	ana perez	2025-02-13 00:00:00	2025-02-17 00:00:00	Aprobado	2521 KB
Plano_006.pdf	/Documentos/Arquitectura/Plantas/Plano_006.pdf	1	ana perez	2025-02-26 00:00:00	2025-03-07 00:00:00	Aprobado	1082 KB
Plano_007.pdf	/Documentos/Arquitectura/Plantas/Plano_007.pdf	1	luis garcia	2024-12-13 00:00:00	2024-12-17 00:00:00	En revisión	2139 KB
Plano_008.pdf	/Documentos/Arquitectura/Plantas/Plano_008.pdf	4	mruiz	2024-11-27 00:00:00	2024-12-05 00:00:00	Rechazado	2301 KB
Plano_009.pdf	/Documentos/Arquitectura/Plantas/Plano_009.pdf	1	mruiz	2025-02-13 00:00:00	2025-02-20 00:00:00	En revisión	1030 KB
Plano_010.pdf	/Documentos/Arquitectura/Plantas/Plano_010.pdf	3	ana perez	2025-04-03 00:00:00	2025-04-07 00:00:00	Aprobado	903 KB
Plano_011.pdf	/Documentos/Arquitectura/Plantas/Plano_011.pdf	1	luis garcia	2025-05-05 00:00:00	2025-05-07 00:00:00	En revisión	502 KB
Plano_012.pdf	/Documentos/Arquitectura/Plantas/Plano_012.pdf	5	carla torres	2024-12-11 00:00:00	2024-12-17 00:00:00	Rechazado	2109 KB
Plano_013.pdf	/Documentos/Arquitectura/Plantas/Plano_013.pdf	4	mruiz	2025-02-28 00:00:00	2025-03-09 00:00:00	En revisión	734 KB
Plano_014.pdf	/Documentos/Arquitectura/Plantas/Plano_014.pdf	4	mruiz	2025-01-21 00:00:00	2025-01-31 00:00:00	Rechazado	1042 KB
Plano_015.pdf	/Documentos/Arquitectura/Plantas/Plano_015.pdf	2	luis garcia	2024-11-24 00:00:00	2024-12-02 00:00:00	Rechazado	2403 KB

Figura 3.5.13. Importación de registro de archivos de ACC en la BBDD

3.5.5. Integración de atributos en modelos BIM

En este estudio de uso se aborda la manipulación de las propiedades extraídas de la base de datos para integrarlas dentro de los modelos BIM. A partir de este caso, se realizan diferentes aproximaciones utilizando los tres primeros niveles de análisis referidos a modelo de categoría, modelo de disciplina, y modelos multidisciplinarios.

De la misma manera que en casos anteriores, durante el proceso extracción, los datos se seleccionan desde la BBDD en función de los requisitos del proyecto, incluyendo las clases de objetos BIM, los usos y las fases de aplicación, diferenciando los conjuntos de información para cada modelo de disciplina. El resultado conecta los datos de varias tablas de esquema en un único archivo para cada modelo de disciplina. A diferencia de los datos del esquema, la información extraída se unifica y agrupa para su aplicación conjunta dentro de las tablas de Excel exportadas. Esto incluye la agrupación de las clases de entidad definidas para cada propiedad y las clases y tipos IFC permitidos para categorías específicas de objetos BIM.

Para el caso concreto de Autodesk Revit, el trabajo de creación de parámetros dentro de modelos BIM a partir de la información filtrada y extraída de la base de datos, tiene tres aplicaciones. Primero, la creación del archivo de parámetros compartidos, después la incorporación de los parámetros dentro de archivos de familia y, por último, la creación de estos dentro de los archivos de proyecto de Revit.

En primer lugar, los parámetros compartidos extienden la información de los objetos del modelo más allá de lo que proporcionan los parámetros internos de Revit, ya que permiten personalizar los datos del modelo y asegurar una adecuada extracción de información para planificación, documentación y análisis. Sin embargo, la gestión manual de estos archivos, mediante la edición directa de sus líneas y campos estructurados, conlleva riesgos importantes, como la introducción de errores sintácticos, uso de identificadores de grupo incorrectos, duplicaciones o inconsistencias en los tipos de datos, que pueden afectar a la integridad de los modelos y a la interoperabilidad entre diferentes equipos o disciplinas.

Para evitar esta edición manual, se desarrolla un script automatizado que genera un archivo TXT de parámetros compartidos que estructura y estandariza el proceso de definición de dichos parámetros mediante programación, utilizando el entorno CPython3 desde Dynamo y basándose en los conjuntos de datos extraídos desde la BBDD. Este script actúa directamente sobre la API de Revit, utilizando clases como *ExternalDefinitionCreationOptions* y *ForgeTypeId* para generar parámetros que sean procesados correctamente dentro del programa.

```
import clr
clr.AddReference("RevitServices")
from RevitServices.Persistence import DocumentManager
from RevitServices.Transactions import TransactionManager
clr.AddReference("RevitAPI")
from Autodesk.Revit.DB import *
import os
#Entradas al nodo de Python
grupos = IN[0]
nombres = IN[1]
```

```

tipos_texto = IN[2]
#Diccionario de tipos válidos
type_dict = {
    "Texto": SpecTypeId.String.Text,
    "Entero": SpecTypeId.Int.Integer,
    "Longitud": SpecTypeId.Length,
    "Número": SpecTypeId.Number,
    "Área": SpecTypeId.Area,
    "Volumen": SpecTypeId.Volume,
    "Sí/No": SpecTypeId.Boolean.YesNo}
#Crear archivo TXT si no existe
descargas = os.path.join(os.path.expanduser("~"), "Downloads")
ruta_parametros = os.path.join(descargas, "ParametrosCompartidos.txt")
if not os.path.exists(ruta_parametros):
    with open(ruta_parametros, 'w') as f:
        f.write("# Archivo de Parámetros Compartidos de Revit\n")
        f.write("*META\tVERSION\tMINVERSION\nMETA\t2\t1\n")
        f.write("*GROUP\tID\tNombre\n")
        f.write("*PARAM\tGUID\tNombre\tTipoDato\tGrupo\tVisible\tDescripción\n")
#Obtener acceso al archivo y aplicación
app = DocumentManager.Instance.CurrentUIApplication.Application
doc = DocumentManager.Instance.CurrentDBDocument
app.SharedParametersFilename = ruta_parametros
archivo_param = app.OpenSharedParameterFile()
#Validación de archivo
if archivo_param is None:
    OUT = "Error al abrir archivo de parámetros"
else:
    definiciones_creadas = []
    #Iterar sobre los parámetros a crear
    for i in range(len(nombres)):
        nombre = nombres[i]
        grupo_nombre = grupos[i]
        tipo = tipos_texto[i]
        grupo = archivo_param.Groups.get_Item(grupo_nombre)
        if grupo is None:
            grupo = archivo_param.Groups.Create(grupo_nombre)
        existe = any(definicion.Name == nombre for definicion in grupo.Definitions)
        if existe:
            definiciones_creadas.append("Ya existe: " + nombre)
            continue
        if tipo in type_dict:
            tipo_id = type_dict[tipo]
        else:
            definiciones_creadas.append("Tipo inválido: " + tipo)
            continue
        opciones = ExternalDefinitionCreationOptions(nombre, tipo_id)
        opciones.Visible = True

```



```

opciones.UserModifiable = True
opciones.Description = "Generado desde Dynamo"
definicion = grupo.Definitions.Create(opciones)
definiciones_creadas.append("Creado: " + nombre)
OUT = definiciones_creadas

```

Esto implica definir el nombre del parámetro, el tipo de dato, la visibilidad, la opción de ser modificable, y la descripción, todo ello encapsulado en una estructura estándar que Revit puede interpretar de forma nativa y sin ambigüedad. Además, se asegura que el archivo generado cumpla con el formato oficial requerido por Autodesk, incluyendo secciones de metadatos, definición de grupos y los parámetros propiamente dichos. En la siguiente imagen se muestra el proceso realizado con un grupo de parámetros enfocados en el control de actividades de la planificación constructiva 4D.

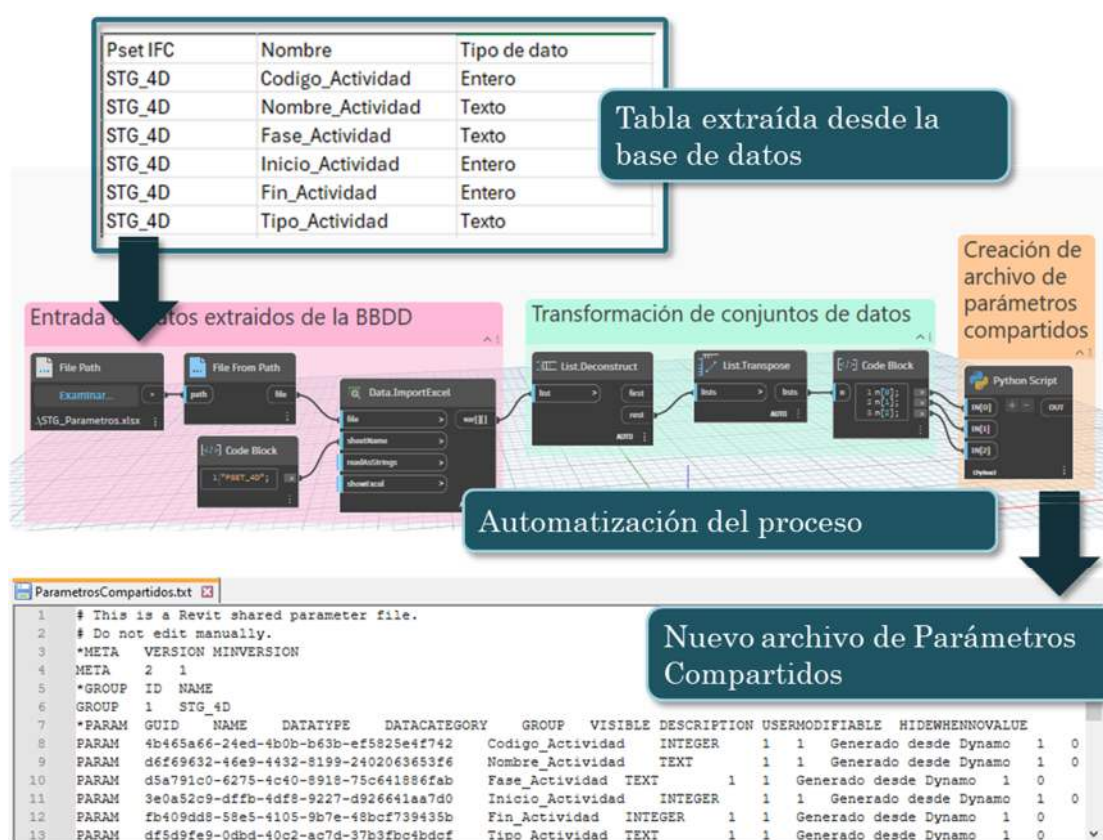


Figura 3.5.14. Creación de archivo de parámetros compartidos

Desde una perspectiva de ingeniería de software, esta automatización se alinea con los principios de reproducibilidad, trazabilidad y control de calidad. Generar el archivo mediante programación hace que el proceso sea registrado, verifica las condiciones de existencia previa de parámetros, y mantiene consistencia entre múltiples proyectos o plantillas, habilitando una gestión masiva y escalable de parámetros en entornos de trabajo colaborativos y multidisciplinares.

Una vez que se encuentra generado el archivo TXT, este puede ser utilizado dentro de las familias de Revit con el objetivo de integrar en la familia estos parámetros y que de esta manera la información sea reutilizable, coherente y accesible. A

diferencia de los parámetros de proyecto, que son los otros tipos de parámetros que pueden ser trabajados en Revit, los compartidos en una familia viajan con ella, manteniendo su GUID y estructura, lo que permite que sean etiquetables, panificables y reconocidos por normas y sistemas externos sin necesidad de redefinirlos en cada archivo.

Esto es especialmente útil para desarrollar bibliotecas de familias inteligentes, en las que la geometría y los datos están estandarizados desde el origen. Así, una familia puede insertarse en distintos proyectos conservando los atributos de la organización, como códigos, especificaciones técnicas o propiedades de planificación constructiva.

Para insertar los parámetros en la familia, se genera un nuevo script en CPython3 para Dynamo que automatiza la incorporación de parámetros compartidos dentro de una familia de la categoría Puertas de Revit, utilizando entradas dinámicas del nombre de parámetro, la definición de tipo o instancia y el grupo de organización dentro del programa.

En primer lugar, el script accede al documento activo de Revit y configura la ruta del archivo de parámetros compartidos. Posteriormente, se realiza una búsqueda dentro de este archivo para identificar la definición externa correspondiente a cada nombre de parámetro especificado. Se mapea además un grupo lógico de parámetros a su correspondiente *GroupId* de la API de Revit, lo cual organiza semánticamente los parámetros en la interfaz de usuario de Revit.

Durante la transacción del documento, necesaria para modificar la familia en Revit, se verifica que el parámetro no exista previamente en la familia para evitar duplicación. Si el parámetro es válido y no está presente, se agrega a la familia mediante el *FamilyManager.AddParameter*, especificando si será un parámetro de tipo o de ejemplar.

```
import clr
clr.AddReference('RevitAPI')
clr.AddReference('RevitServices')
import Autodesk.Revit.DB as DB
from RevitServices.Persistence import DocumentManager
from RevitServices.Transactions import TransactionManager
#Entradas desde Dynamo
param_names = IN[0]
is_instances = IN[1]
group_names = IN[2]
shared_params_path = IN[3]
doc = DocumentManager.Instance.CurrentDBDocument
app = DocumentManager.Instance.CurrentUIApplication.Application
#Configurar archivo de parámetros compartidos
app.SharedParametersFilename = shared_params_path
sharedFile = app.OpenSharedParameterFile()
#Mapeo a GroupTypeId
group_typeid_map = {
    "IdentityData": DB.GroupTypeId.IdentityData,
    "Materials": DB.GroupTypeId.Materials,
```

```

    "Graphics": DB.GroupTypeId.Graphics,
    "Geometry": DB.GroupTypeId.Geometry,
    "Constraints": DB.GroupTypeId.Constraints,
    "Text": DB.GroupTypeId.Text,
    "Data": DB.GroupTypeId.Data,
    "Construction": DB.GroupTypeId.Construction,}
resultados = []
if sharedFile is None or sharedFile.Groups is None:
    OUT = ["El archivo de parámetros compartidos no es válido o no tiene grupos."]
else:
    TransactionManager.Instance.EnsureInTransaction(doc)
    try:
        for i, param_name in enumerate(param_names):
            is_instance = is_instances[i] if i < len(is_instances) else True
            group_key = group_names[i] if i < len(group_names) else "IdentityData"
            group_type_id = group_typeid_map.get(group_key,
DB.GroupTypeId.IdentityData)
            extDef = None
            for group in sharedFile.Groups:
                for defn in group.Definitions:
                    if defn.Name.strip() == param_name:
                        extDef = defn
                        break
            if extDef:
                break
            if extDef is None:
                resultados.append(f"Parámetro '{param_name}' no encontrado.")
                continue
            existe = any(p.Definition.Name == param_name for p in
doc.FamilyManager.Parameters)
            if existe:
                resultados.append(f"Parámetro '{param_name}' ya existe.")
                continue
            doc.FamilyManager.AddParameter(extDef, group_type_id, is_instance)
            resultados.append(f"Parámetro '{param_name}' agregado en grupo
'{group_key}'.")
        TransactionManager.Instance.TransactionTaskDone()
        OUT = resultados
    except Exception as e:
        TransactionManager.Instance.ForceCloseTransaction()
        OUT = [f"Error: {str(e)}"]

```

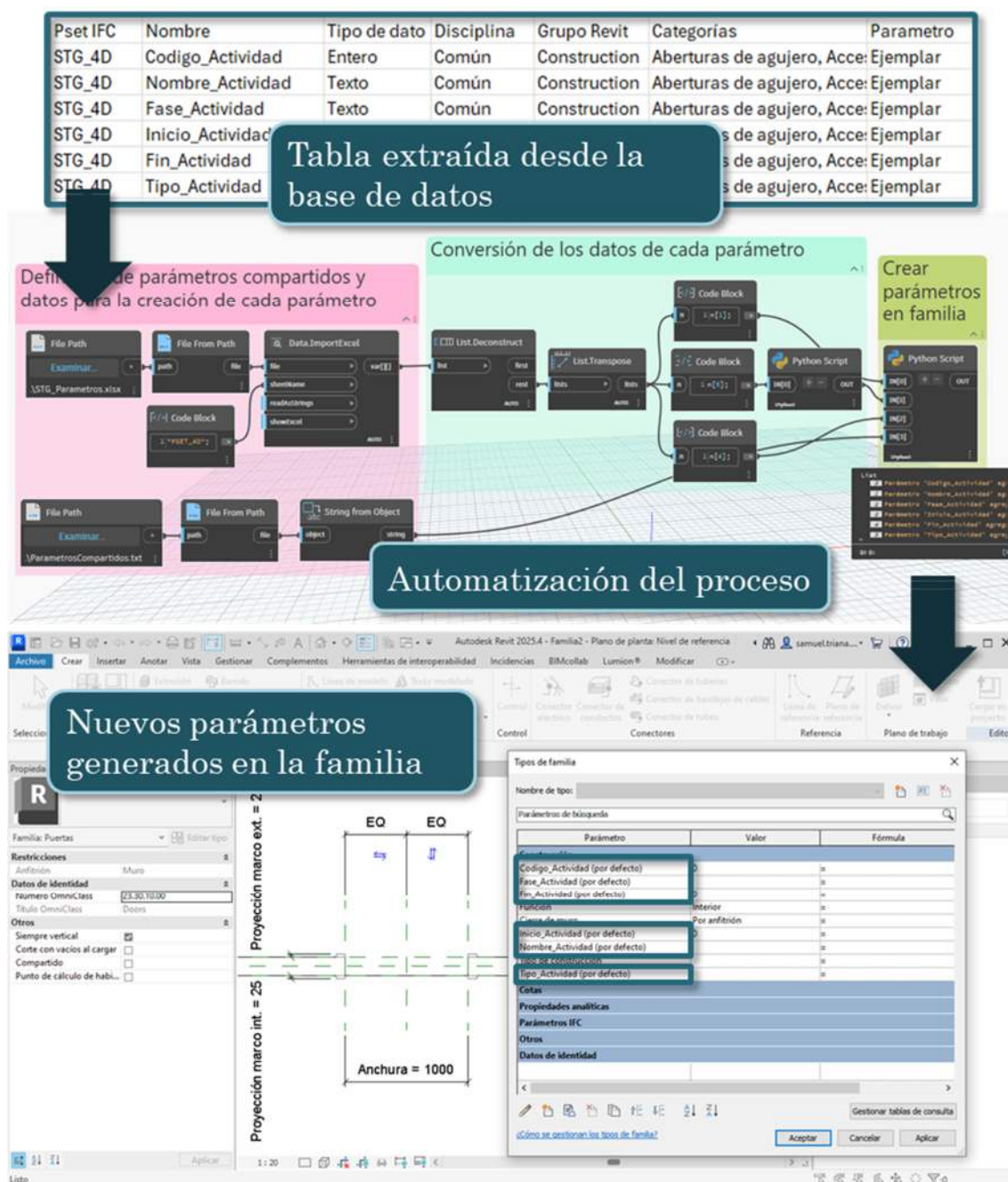


Figura 3.5.15. Creación de parámetros en familia de Revit

Además de añadir los parámetros dentro de las familias, estos pueden ser incluidos en los proyectos de Revit para garantizar el trabajo del conjunto del modelo BIM. Este proceso puede llevarse a cabo a partir del uso de la herramienta de Parámetros compartidos que contiene el complemento de Herramientas de interoperabilidad BIM de Revit, la cual crea los parámetros en el proyecto y los asigna a las categorías que hayan sido definidas en la plantilla.

Para generar la plantilla y que esta se defina directamente con los datos obtenidos de la BBDD, es necesario crear un archivo XML que respete y organice los datos de la misma manera que lo hace la herramienta. Para ello, se genera un código automatizado en Visual Studio Code que crea el archivo XML y que, a partir del

archivo de parámetros compartidos y de la plantilla de datos con el resto de información, se asignan las características de cada parámetro.

En primer lugar, el código lee el archivo de texto de parámetros compartidos de Revit. Sus datos son extraídos y almacenados como una lista de diccionarios para su posterior procesamiento. Simultáneamente, se lee un archivo XLSX que contiene la información complementaria de cada parámetro, como el grupo al que pertenece, el tipo de dato, la disciplina asociada, el grupo de Revit, las categorías del modelo y el tipo de parámetro, sea este de ejemplar o de tipo. Esta hoja de cálculo es interpretada como un DataFrame de Pandas, lo que permite un acceso eficiente y estructurado a la información.

Posteriormente, el script combina la información proveniente de ambas fuentes utilizando una lógica de mapeo predefinida para traducir los valores textuales a identificadores estándar utilizados por Revit. Por ejemplo, los tipos de datos como *Integer* o *Text* se mapean a sus representaciones estándar junto con las unidades correspondientes, mientras que las categorías del modelo se asocian a sus identificadores internos de Revit. Además, se determina si el parámetro debe aplicarse a instancias o tipos, y se define el grupo de clasificación dentro de la interfaz de Revit.

Finalmente, se construye un árbol XML con esta información, siguiendo una jerarquía específica que Revit puede interpretar al importar el archivo. Cada parámetro se encapsula dentro de una estructura XML que incluye tanto sus propiedades como las categorías a las que se aplicará. El archivo resultante se convierte en un texto legible y se guarda en un archivo con la codificación y encabezado requeridos por los estándares de interoperabilidad de Revit.

```
import xml.etree.ElementTree as ET
import pandas as pd
import xml.dom.minidom as minidom
#Leer archivo TXT de parámetros compartidos
txt_file_path = "C:/Users/samuel.triana/Downloads/ParametrosCompartidos.txt"
with open(txt_file_path, "r", encoding="utf-8") as file:
    shared_param_txt = file.read().strip().splitlines()
#Leer archivo Excel con datos de cada parámetro
excel_file_path = "C:/Users/samuel.triana/Downloads/STG_Parametros.xlsx"
df = pd.read_excel(excel_file_path)
#Transformar datos del TXT
parametros_txt = []
for line in shared_param_txt:
    if line.startswith("PARAM"):
        parts = line.split("\t")
        parametros_txt.append({
            "GUID": parts[1],
            "Name": parts[2],
            "DATATYPE": parts[3]})
#Diccionarios de mapeo
unit_map = {
    "INTEGER": ("Integer", "autodesk.spec.spec.int64-2.0.0", "Entero"),
```

```

    "TEXT": ("Text", "autodesk.spec:spec.string-2.0.0", "Texto")}
group_under_map = {
    "Construction": "autodesk.parameter.group:construction-1.0.0"}
category_id_map = {
    "Puertas": "-2000023",
    "Ventanas": "-2000014"}
#Crear estructura XML base
root = ET.Element("SharedParametersConfig", {
    "Version": "9.2",
    "Name": "Crear Parametros Interoperabilidad",
    "Author": "Samuel Triana",
    "Description": "",
    "Image": ""})
params_elem = ET.SubElement(root, "Parameters")
#Combinar datos TXT y Excel
for param in parametros_txt:
    row = df[df["Nombre"] == param["Name"]].iloc[0]
    tipo, unit_type, unit_display = unit_map.get(param["DATATYPE"], ("", "", ""))
    is_instance = "True" if row["Parametro"] == "Ejemplar" else "False"
    categorias = [cat.strip() for cat in row["Categorías"].split(",")]
    group_under = group_under_map.get(row["Grupo Revit"],
    "autodesk.parameter.group:other-1.0.0")
    param_elem = ET.SubElement(params_elem, "Parameter", {
        "GUID": param["GUID"],
        "Name": param["Name"],
        "Group": row["Pset IFC"],
        "Type": tipo,
        "UnitType": unit_type,
        "UnitTypeDisplay": unit_display,
        "Description": "",
        "IsInstance": is_instance,
        "GroupUnder": group_under,
        "AlignGroupValues": "True",
        "UserModifiable": "True",
        "HideWhenNoValue": "False"})
    cats_elem = ET.SubElement(param_elem, "Categories")
    for cat in categorias:
        cat_id = category_id_map.get(cat, "-999999")
        ET.SubElement(cats_elem, "Category", {"ID": cat_id, "Name": cat})
#Guardar archivo XML
xml_str = ET.tostring(root, encoding='utf-8', method='xml')
parsed = minidom.parseString(xml_str)
pretty_xml = parsed.toprettyxml(indent=" ", encoding="utf-8")
with open("ParametrosInteroperabilidad.xml", "wb") as f:
    f.write(b'<?xml version="1.0" encoding="UTF-8" standalone="yes"?>\n')
    f.write(pretty_xml)

```

El resultado es el archivo XML el cual es añadido como plantilla dentro del complemento de Revit. Una vez insertado, se ejecuta la creación de los parámetros

dentro del proyecto y se verifica que se hayan generado correctamente en las categorías asignadas.



Figura 3.5.16. Creación de parámetros en Revit con XML

Otra forma de realizar el proceso de creación de parámetros en el proyecto de Revit es a partir de la realización de un script dentro de Dynamo que, utilizando los mismos archivos de entrada que utiliza el XML anterior, genere de manera automática los parámetros tomando los de referencia del TXT de parámetros compartidos.

Una característica fundamental de este código es la normalización y validación de las categorías asignadas, que permite una coincidencia insensible a mayúsculas y minúsculas entre los nombres suministrados y las categorías existentes en el proyecto. Esto garantiza la correcta asociación de los parámetros a los elementos

adecuados, minimizando errores derivados de inconsistencias tipográficas o variaciones en la nomenclatura. Además, el proceso incorpora un control exhaustivo de errores y condiciones particulares, tales como la ausencia del archivo de parámetros compartidos, la inexistencia de definiciones coincidentes o la duplicación de parámetros en el proyecto, proporcionando retroalimentación precisa y detallada para cada caso. Asimismo, se asegura que los parámetros se vinculen correctamente según su tipo de instancia o tipo compartido, respetando la estructura de grupos definida por el usuario y el esquema de la API de Revit.

Finalmente, el script maneja las transacciones de manera explícita para garantizar la integridad del modelo durante la modificación, cerrando adecuadamente las operaciones ante errores y manteniendo la estabilidad del entorno de trabajo.

```
import clr
clr.AddReference('RevitAPI')
clr.AddReference('RevitServices')
import Autodesk.Revit.DB as DB
from RevitServices.Persistence import DocumentManager
from RevitServices.Transactions import TransactionManager
# Entradas de Dynamo
param_names = IN[0]
is_instances = IN[1]
group_names = IN[2]
shared_params_path = IN[3]
categorias_lista = IN[4]
doc = DocumentManager.Instance.CurrentDBDocument
app = DocumentManager.Instance.CurrentUIApplication.Application
#Configurar archivo de parámetros compartidos
app.SharedParametersFilename = shared_params_path
sharedFile = app.OpenSharedParameterFile()
#Mapeo de grupos a GroupTypeId
group_typeid_map = {
    "IdentityData": DB.GroupTypeId.IdentityData,
    "Materials": DB.GroupTypeId.Materials,
    "Graphics": DB.GroupTypeId.Graphics,
    "Geometry": DB.GroupTypeId.Geometry,
    "Constraints": DB.GroupTypeId.Constraints,
    "Text": DB.GroupTypeId.Text,
    "Data": DB.GroupTypeId.Data,
    "Construction": DB.GroupTypeId.Construction,}
#Función para obtener categoría
def get_category_by_name(name):
    if not name:
        return None
    name_lower = name.strip().lower()
    for cat in doc.Settings.Categories:
        if cat.Name.strip().lower() == name_lower:
            return cat
    return None
```

```

resultados = []
if sharedFile is None or sharedFile.Groups is None:
    OUT = ["El archivo de parámetros compartidos no es válido o no contiene grupos."]
else:
    TransactionManager.Instance.EnsureInTransaction(doc)
    try:
        bindingMap = doc.ParameterBindings
        for i, param_name in enumerate(param_names):
            is_instance = is_instances[i] if i < len(is_instances) else True
            group_key = group_names[i] if i < len(group_names) else "IdentityData"
            group_type_id = group_typeid_map.get(group_key,
DB.GroupTypeId.IdentityData)
            #Construir CategorySet a partir de la lista correspondiente
            category_set = DB.CategorySet()
            categorias_no_encontradas = []
            categorias_param = categorias_lista[i] if i < len(categorias_lista) else []
            for cat_name in categorias_param:
                cat = get_category_by_name(cat_name)
                if cat:
                    category_set.Insert(cat)
                else:
                    categorias_no_encontradas.append(cat_name)
            if category_set.IsEmpty:
                resultados.append(f"Parámetro '{param_name}': No se asignó a ninguna
categoría válida (categorías indicadas: {categorias_param})")
                continue
            #Buscar definición en archivo compartido
            extDef = None
            for group in sharedFile.Groups:
                for defn in group.Definitions:
                    if defn.Name.strip() == param_name:
                        extDef = defn
                        break
            if extDef:
                break
            if extDef is None:
                resultados.append(f"Parámetro '{param_name}' no encontrado en el archivo
de parámetros compartidos.")
                continue
            #Verificar si ya existe el parámetro en el proyecto
            if bindingMap.Contains(extDef):
                resultados.append(f"Parámetro '{param_name}' ya existe en el proyecto.")
                continue
            #Crear binding o asociación para instancia o tipo
            binding = DB.InstanceBinding(category_set) if is_instance else
DB.TyBEPinding(category_set)
            #Insertar binding
            success = bindingMap.Insert(extDef, binding, group_type_id)
            if success:

```

```

msg = f"Parámetro '{param_name}' creado en proyecto con categorías:
{categorias_param}"
if categorias_no_encontradas:
    msg += f" (No encontradas: {categorias_no_encontradas})"
resultados.append(msg)
else:
    resultados.append(f"No se pudo agregar el parámetro '{param_name}'.")
TransactionManager.Instance.TransactionTaskDone()
OUT = resultados
except Exception as e:
    TransactionManager.Instance.ForceCloseTransaction()
OUT = [f"Error general: {str(e)}"]

```

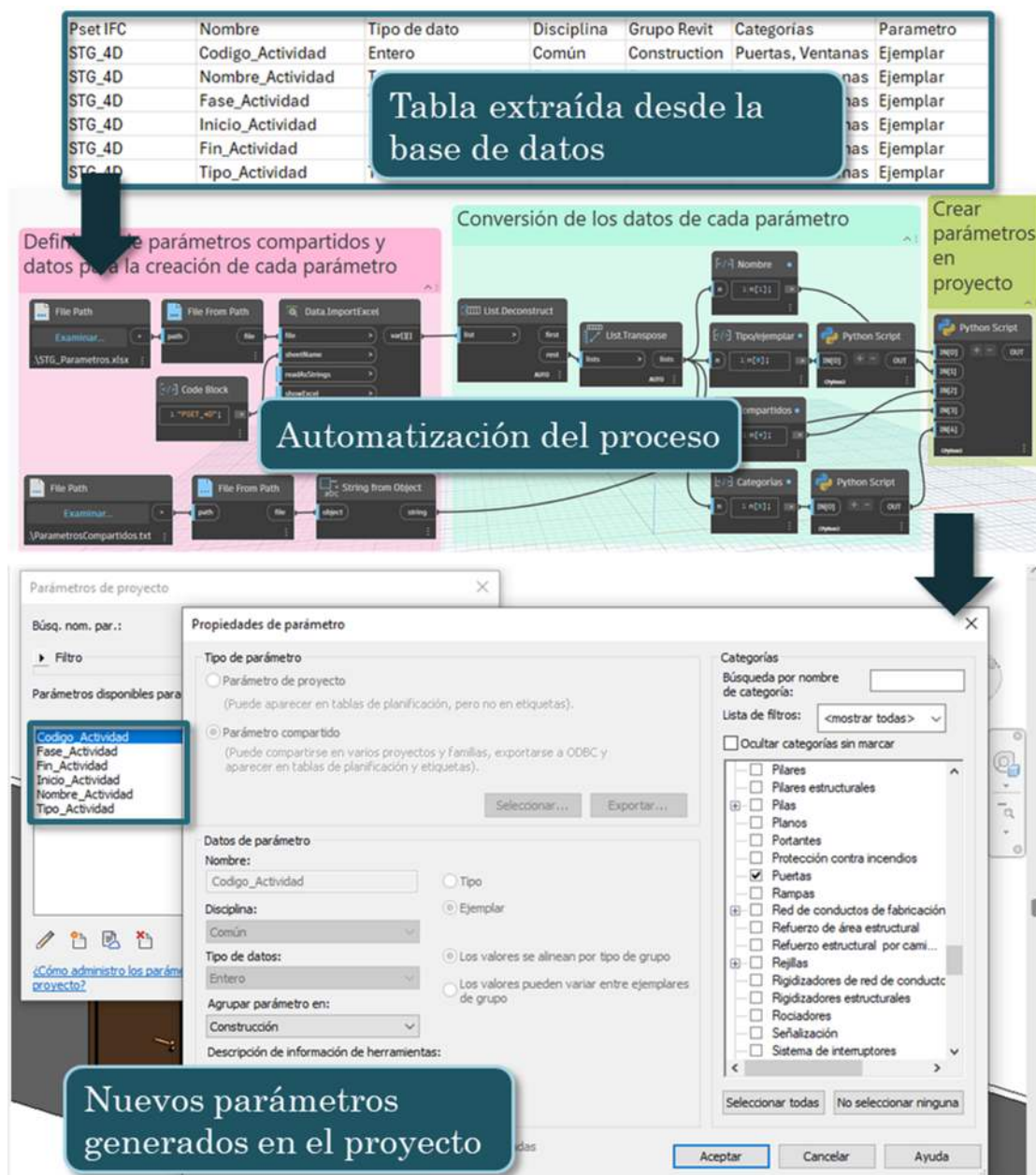


Figura 3.5.17. Creación de parámetros en Revit con Dynamo

En el caso de utilizar Blender y Bonsai, se genera una definición de conjunto de propiedades, denominada Property Set Template, almacenada en un archivo IFC independiente, que actúa como una librería semántica, buscando ser similar al archivo de parámetros compartidos de Revit. Esta librería contiene la estructura normalizada de propiedades aplicables a objetos de una clase específica, o bien a múltiples clases IFC. Al importar dicha plantilla en otro archivo IFC de proyecto, se automatiza la generación de propiedades y su asociación a los elementos correspondientes mediante relaciones explícitas. Este enfoque favorece la consistencia de datos entre modelos, la trazabilidad del origen de las definiciones y la escalabilidad de la gestión de propiedades en flujos OpenBIM, manteniendo la independencia respecto a las herramientas de autoría, ya que esta plantilla de propiedades se podría reutilizar en otros softwares BIM.

La implementación de este procedimiento mediante una primera rutina en IfcOpenShell permite la creación directa del modelo IFC a nivel de esquema, lo que asegura que la estructura resultante se mantenga conforme a la norma ISO 16739. Este script permite definir propiedades personalizadas directamente dentro de entidades IFC en un proyecto BIM. Para ello, el script debe leer la tabla de datos estructurada en formato Excel extraída de la BBDD, donde cada fila contiene el nombre de una o varias clases IFC, el nombre del conjunto de propiedades y un conjunto de propiedades con sus valores asociados. Mediante la interfaz de programación, el script identifica, para cada fila, toda la información relevante para generar el nuevo archivo de referencia.

```
import ifcopenshell
from ifcopenshell.api import run
import pandas as pd
from datetime import datetime
def create_pset_from_excel(excel_path, output_path):
    # Leer Excel
    df = pd.read_excel(excel_path)
    if not {"PsetName", "PropertyName", "PropertyType"}.issubset(df.columns):
        raise ValueError("El Excel debe tener columnas: 'PsetName', 'PropertyName', 'PropertyType'")
    # Crear archivo IFC4
    library = ifcopenshell.file(schema="IFC4")
    # Encabezado
    header = library.wrapped_data.header
    header.file_name.name = "Library_STG"
    header.file_name.time_stamp =
datetime.now().replace(microsecond=0).isoformat()
    header.file_name.author = ("Samuel Triana",)
    header.file_name.originating_system = "Blender"
    header.file_name.preprocessor_version = "IfcOpenShell 0.8.2"
    # Crear el PsetTemplate
    pset_name = df["PsetName"].iloc[0]
    stg_pset = run(
        "pset_template.add_pset_template",
        library,
```



```

name=pset_name,
template_type="PSET_TYPEDRIVENOVERRIDE",
applicable_entity="IfcElement")

# Crear propiedades
prop_templates = []
for _, row in df.iterrows():
    prop_template = run(
        "pset_template.add_prop_template",
        library,
        pset_template=stg_pset,
        name=row["PropertyName"],
        primary_measure_type=row["PropertyType"],
        template_type="P_SINGLEVALUE")
    prop_templates.append(prop_template)

# Relacionar propiedades al template
run(
    "pset_template.edit_pset_template",
    library,
    pset_template=stg_pset,
    attributes={"HasPropertyTemplates": prop_templates})

# Guardar IFC
library.write(output_path)
print(f"Librería IFC generada: {output_path}")

# Ruta del Excel y archivo de salida IFC
if __name__ == "__main__":
    excel_path = "C:/Users/Samuel/Downloads/pset_excel.xlsx"
    output_ifc = "C:/Users/Samuel/Downloads/LIBRARY_STG.ifc"
    create_pset_from_excel(excel_path, output_ifc)

```

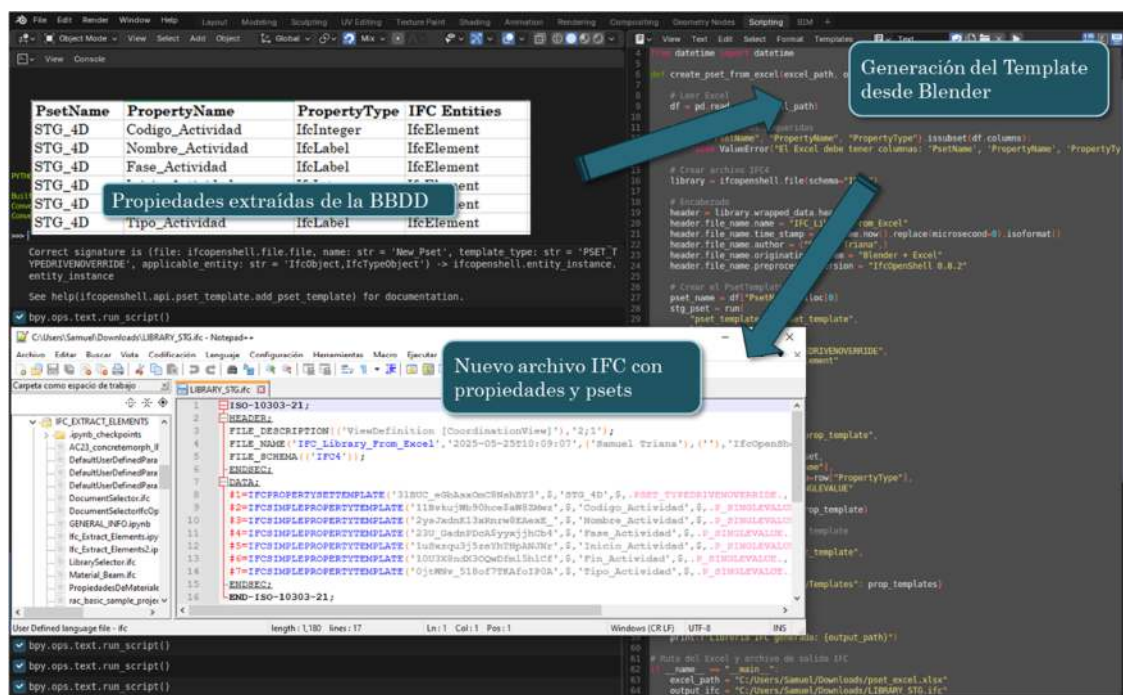


Figura 3.5.18. Creación de biblioteca de propiedades IFC

Una vez que se ha generado la biblioteca de propiedades, esta puede ser insertada en todos los modelos del proyecto. Para ello, se genera una segunda rutina que lee y extrae los datos de la biblioteca y se los asigna a los elementos IFC del modelo. Al finalizar, el modelo modificado incorpora las propiedades adicionales dentro de la estructura IFC, de manera que, una vez exportado, pueden ser reconocidas y visualizadas por cualquier software IFC compatible. Esto permite enriquecer el modelo BIM con información adicional relevante para diferentes disciplinas o fases del proyecto sin alterar la geometría, facilitando la gestión y análisis posterior.

```

import ifcopenshell
source_ifc_path = "C:/Users/Samuel/Downloads/LIBRARY_STG.ifc"
target_ifc_path = "C:/Users/Samuel/Downloads/Proj_2_ARC.ifc"
output_ifc_path = "C:/Users/Samuel/Downloads/Proj_2_ARC.ifc"
pset_name = "STG_4D"
#Cargar los archivos IFC
lib_ifc = ifcopenshell.open(source_ifc_path)
dst_ifc = ifcopenshell.open(target_ifc_path)
#Buscar los elementos dentro de la plantilla IFC
template = next((t for t in lib_ifc.by_type("IfcPropertySetTemplate") if t.Name ==
pset_name), None)
if not template:
    raise Exception(f"No se encontró el template '{pset_name}'.")
#Función para crear propiedades desde plantilla
def crear_propiedades_from_template(ifc_file, template):
    props = []
    for tpl in template.HasPropertyTemplates:
        prop_name = tpl.Name
        measure_type = tpl.PrimaryMeasureType or ""
        # Crear el valor como entidad IFC según el tipo
        if "Integer" in measure_type:
            value = ifc_file.create_entity("IfcInteger", 0)
        elif "Real" in measure_type:
            value = ifc_file.create_entity("IfcReal", 0.0)
        elif "Length" in measure_type:
            value = ifc_file.create_entity("IfcLengthMeasure", 0.0)
        elif "Text" in measure_type:
            value = ifc_file.create_entity("IfcText", "")
        else:
            value = ifc_file.create_entity("IfcLabel", "")
        prop = ifc_file.create_entity("IfcPropertySingleValue", prop_name, None, value,
None)
        props.append(prop)
    return props
#Buscar las clases o clase IFC
walls = dst_ifc.by_type("IfcElement")
for elem in elements:
    #Eliminar PSet existente si existe
    to_remove = []
    for rel in getattr(elem, "IsDefinedBy", []):

```

```

definition = rel.RelatingPropertyDefinition
if definition.is_a("IfcPropertySet") and definition.Name == pset_name:
    to_remove.append((rel, definition))
for rel, pset in to_remove:
    dst_ifc.remove(pset)
    dst_ifc.remove(rel)

#Crear nuevo conjunto de propiedades
propiedades = crear_propiedades_from_template(dst_ifc, template)
#Crear el PropertySet
pset = dst_ifc.create_entity(
    "IfcPropertySet",
    ifcopenshell.guid.new(),
    None,
    pset_name,
    None,
    propiedades)

#Crear la relación entre el PSet y los elementos
rel = dst_ifc.create_entity(
    "IfcRelDefinesByProperties",
    ifcopenshell.guid.new(),
    None,
    "Relación PSet aplicada automáticamente",
    None,
    [elem],
    pset)

#Guardar el IFC con las nuevas propiedades
dst_ifc.write(output_ifc_path)

```

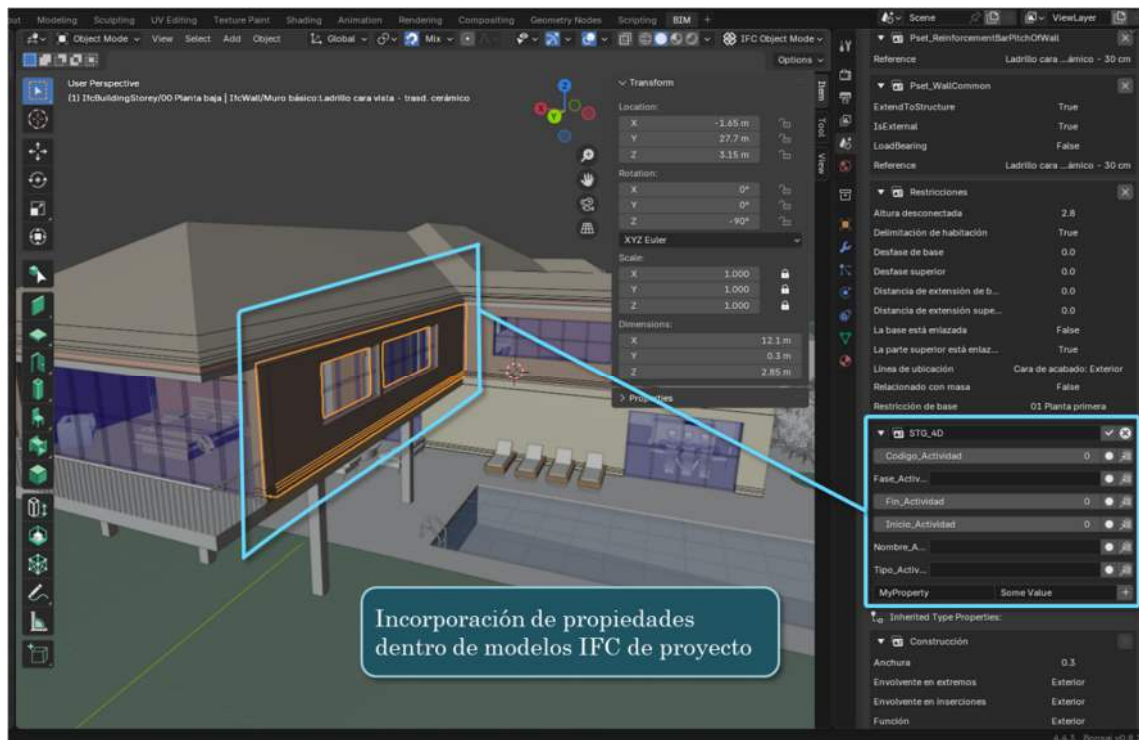


Figura 3.5.19. Incorporación de propiedades en modelo con Blender

En este caso en el que se han generado las propiedades dentro del modelo arquitectónico del proyecto 2, la plantilla se ha leído desde un modelo de biblioteca donde los *IfcPropertySetTemplate* definen la estructura y tipos de datos, y se ha aplicado a todas las entidades *IfcElement* del modelo de destino, generando nuevas instancias *IfcPropertySingleValue* válidas según el esquema IFC4.

Este proceso es de alta utilidad cuando se requiere enriquecer modelos con datos específicos de planificación, faseado o control de obra, sin alterar su geometría ni depender de la introducción manual. En casos en los que un modelo es compartido por diferentes equipos, estas rutinas pueden ser de gran utilidad, ya que el segundo equipo solo genera las nuevas propiedades de datos alfanuméricos, sin afectar a la geometría o relaciones espaciales del modelo IFC.

Otro aspecto relevante con estos procesos de trabajo es la correspondencia exacta entre los tipos de propiedad definidos, como *IfcInteger* o *IfcLabel*, y las instancias creadas, así como la correcta eliminación y recreación de relaciones *IfcRelDefinesByProperties* para evitar duplicidades o inconsistencias. Además, se asegura que los nombres de los conjuntos de propiedades coincidan exactamente entre archivos y que las entidades de destino sean del tipo esperado.

Cabe destacar que, para el caso concreto del conjunto de propiedades 4D, no se han trabajado las definiciones propias del esquema como *IfcTask* debido a que, al incorporar más adelante los modelos en los softwares de coordinación, la gran mayoría de estos no leen correctamente los datos de planificación. Por otro lado, se realiza una comprobación desde un visor BIM para garantizar que las propiedades se han definido correctamente. Lógicamente, los valores aparecen vacíos ya que es en el siguiente caso de uso en el que se aborda este tema.

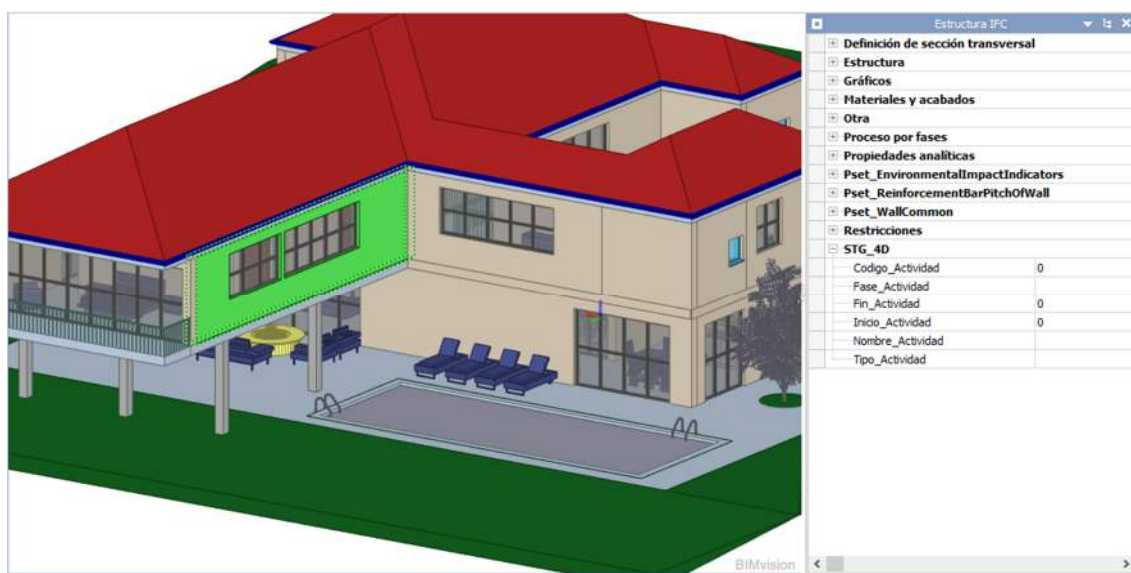


Figura 3.5.20. Comprobación de nuevas propiedades en BIMVision

3.5.6. Definición de valores de atributos

Una vez que se han establecido los atributos dentro de los programas de autoría BIM, existen diferentes formas de acotar y definir los valores que pueden ser asignados. En este caso de estudio se ejecutan diferentes flujos para realizar de una manera más eficiente el relleno de cada propiedad, pudiendo ser este de instancia, de tipo, del proyecto, del edificio, del emplazamiento o de elementos de referencia como niveles.

Dependiendo de los requisitos y el tipo de proyecto, los archivos exportados desde la base relacional incluyen datos sobre los posibles valores de los parámetros del proyecto. El desarrollo consiste en que el usuario seleccione los valores adecuados mediante ventanas emergentes para instancias o tipos de elementos preseleccionados, utilizando herramientas existentes o scripts ejecutados mediante Dynamo o Visual Code para evitar alteraciones de datos.

Dentro De Autodesk Revit, la herramienta Classification Manager del complemento de Herramientas de Interoperabilidad BIM permite cargar plantillas con valores de clasificaciones para asignar a los elementos del modelo. A partir de una rutina de automatización los datos tabulares extraídos de la BBDD se configuran para encajar con la estructura que exige la herramienta.

```
import pandas as pd
from openpyxl import Workbook
from openpyxl.utils.dataframe import dataframe_to_rows
# Cargar datos desde Excel de la BBDD
input_file = r'C:\Users\Samuel\F_Definicion_valores\Clasificaciones_BBDD.xlsx'
df = pd.read_excel(input_file, header=None)
# Asignar nombres de columna a partir del contenido esperado
df.columns = ['NUMBER', 'DESCRIPTION', 'LEVEL']
#Datos del encabezado de la tabla configurables
header_data = {
    'TITLE': 'Productos',
    'DESCRIPTION': 'Elements / Funciones RIH v02',
    'VERSION': 'REV 07 REVIT 2024',
    'FUNCTION': 'Element',
    'NUMBER PARAMETER': 'Codigo_Elemento',
    'DESCRIPTION PARAMETER': 'Descripcion_Elemento',}
#Crear workbook con openpyxl
wb = Workbook()
ws = wb.active
#Escribir datos superiores (encabezados)
ws['A1'] = 'TITLE'
ws['B1'] = header_data['TITLE']
ws['A2'] = 'DESCRIPTION'
ws['B2'] = header_data['DESCRIPTION']
ws['A3'] = 'VERSION'
ws['B3'] = header_data['VERSION']
ws['A4'] = 'FUNCTION'
ws['B4'] = header_data['FUNCTION']
```



```

ws['A5'] = 'NUMBER PARAMETER'
ws['B5'] = header_data['NUMBER PARAMETER']
ws['A6'] = 'DESCRIPTION PARAMETER'
ws['B6'] = header_data['DESCRIPTION PARAMETER']
start_row = 8
#Escribir encabezados de la tabla en la fila 8
table_headers = df.columns.tolist()
for col_idx, header in enumerate(table_headers, start=1):
    ws.cell(row=start_row, column=col_idx, value=header)
#Escribir datos de la tabla a partir fila 9
for r_idx, row in enumerate(dataframe_to_rows(df, index=False, header=False),
start_row + 1):
    for c_idx, value in enumerate(row, start=1):
        ws.cell(row=r_idx, column=c_idx, value=value)
# Guardar archivo Excel
output_file = r'C:\Users\Samuel\F_Definicion_valores\Plant_Classification.xlsx'
wb.save(output_file)
print(f'Archivo Excel generado: {output_file}')

```

Tabla extraída desde la base de datos

FUN.ENT	Entorno	2
FUN.ENT.010	Símbolos	3
FUN.ENT.010.010	Origen de coordenadas	4
FUN.ENT.010.020	Red o malla	4
FUN.ENT.020	Áreas arboladas	3
FUN.ENT.020.010	Árbol	4
FUN.ENT.020.020	Arbusto	4
FUN.ENT.030	Parcelas	3
FUN.ENT.030.010	Parcela completa	4
FUN.ENT.030.020	Subparcela	4
FUN.ENT.030.030	Área	4

Tabla final para utilizar como plantilla de asignación de valores

TITLE	Productos	
DESCRIPTION	Elements / Funciones RIH v02	
VERSION	REV 07 REVIT 2024	
FUNCTION	Element	
NUMBER PARAMETER	Codigo_Elemento	
DESCRIPTION PARAMETER	Descripcion_Elemento	
NUMBER	DESCRIPTION	LEVEL
FUN.ENT	Entorno	2
FUN.ENT.010	Símbolos	3
FUN.ENT.010.010	Origen de coordenadas	4
FUN.ENT.010.020	Red o malla	4
FUN.ENT.020	Áreas arboladas	3
FUN.ENT.020.010	Árbol	4
FUN.ENT.020.020	Arbusto	4
FUN.ENT.030	Parcelas	3
FUN.ENT.030.010	Parcela completa	4
FUN.ENT.030.020	Subparcela	4
FUN.ENT.030.030	Área	4

Figura 3.5.21. Tabla de funciones de elementos del SCFClass v02

Todas las tablas de valores que se generen, se unifican en el archivo XLSX que servirá como plantilla de asignación de valores dentro de la herramienta. Para toda esta configuración es clave asignar correctamente la Función de cada tabla, pudiendo estar enfocada al activo, a los elementos, a los sistemas o a espacios del modelo BIM. Asimismo, en la parte del encabezado superior es necesario que se definan el nombre del parámetro o parámetros de Revit que van a ser asociados a la clasificación de la tabla.

Con la plantilla preparada, se abre el modelo de Revit, se carga el archivo XSLX dentro del Classification Manager y se van seleccionando los objetos o conjuntos de los objetos del modelo para ser definidos por cada una de las clasificaciones establecidas en la plantilla. Es necesario que previo a la asignación de los valores,

los parámetros hayan sido creados correctamente, lo que implica la definición de las categorías asociadas, la definición del tipo de parámetro y la correcta nomenclatura del mismo.



Figura 3.5.22. Asignación de valores a objetos desde Classification Manager

En otros casos, la información es relativa a datos generales del proyecto, no a información específica de los objetos de construcción. Para definir este tipo de datos, el parámetro es necesario que haya sido definido en la categoría de Información de Proyecto y que el grupo de valores predefinidos se seleccione desde el apartado de Instalación, el cual hace referencia a parámetros generales del modelo de información BIM.

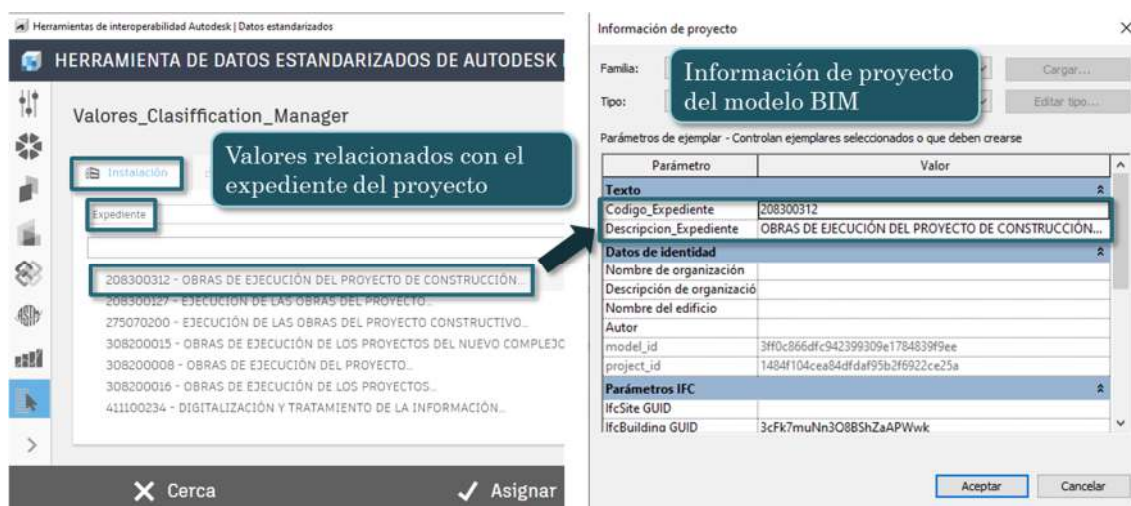


Figura 3.5.23. Asignación de valores a proyecto desde Classification Manager

Es necesario destacar en estos procesos que la herramienta utilizada solo permite establecer relaciones de 1 a 1, por lo que no es posible añadir dos códigos o descripciones a un mismo objeto, sino que el proceso tendría que realizarse

manualmente o con programación. No obstante, la gran mayoría de objetos solo tienen un código específico relacionado con su función o tipo de producto, zonificación, estado, fuente o tipo de activo. Además, este proceso asegura la correcta definición de la clasificación y asegura que el código y la descripción sean coherentes entre sí.

Aunque esta plantilla no está pensada para valores de parámetros independientes, también puede ser utilizada para definir un listado o enumeración de valores para una propiedad concreta. Simplemente se asignan los valores en la columna del código y al asignar la clasificación desde Revit, lo que ocurre es que solo se rellena la propiedad definida. Esto es útil para parámetros que deben tener un listado concreto de valores como puede ser Estado_Expediente, con valores como *Licitacion*, *Adjudicado*, *En_Desarrollo*, *Paralizado*, *Finalizado*.

Por otro lado, existen valores que son dependientes de otros que, una vez que los primeros son definidos, los segundos pueden ser agregados a partir de una programación en Dynamo que contenga una relación entre dichos valores. Por ejemplo, si existe un código definido por Función para un tipo de objeto, dicho código puede tener una relación directa y preestablecida con una disciplina, una subdisciplina, una clase IFC y un tipo predefinido IFC.

La rutina para la asignación de estos valores dependientes comienza con la recolección desde Dynamo de todos los elementos de las categorías de objetos constructivos que existen en el modelo abierto en Revit, excluyendo las relacionadas con información de proyecto, materiales, enrutamientos, habitaciones, espacios, áreas y zonas de climatización. Una vez obtenidos los elementos, se extrae el valor que tienen dentro del parámetro Código_Elemento para poder más tarde revisar los valores dependientes.

```
import clr
clr.AddReference('RevitServices')
import RevitServices
from RevitServices.Persistence import DocumentManager
clr.AddReference('RevitAPI')
import Autodesk
from Autodesk.Revit.DB import *

# Documento activo
doc = DocumentManager.Instance.CurrentDBDocument
#Excluir categorías por BuiltInCategory, si existen
excluded_builtins = []
for name in [
    "OST_ProjectInformation", "OST_Materials", "OST_PipeSegments", "OST_Rooms",
    "OST_Areas", "OST_Spaces", "OST_ZoneSchemes"]:
    try:
        excluded_builtins.append(getattr(BuiltInCategory, name))
    except:
        pass
excluded_ids = [int(b) for b in excluded_builtins]
# Obtener todas las categorías válidas
def get_model_categories(doc):
    settings = doc.Settings.Categories
```

```

model_cats = []
for cat in settings:
    try:
        name_lower = cat.Name.lower()
        if (
            cat.CategoryType == CategoryType.Model and
            not cat.IsTagCategory and
            cat.Id.IntegerValue not in excluded_ids and
            all(keyword not in name_lower for keyword in excluded_name_keywords)):
            model_cats.append(cat)
    except:
        pass
return model_cats
# Obtener elementos por categoría
def get_elements_by_category(category):
    collector = FilteredElementCollector(doc).
    OfCategoryId(category.Id).WhereElementIsNotElementType()
    return list(collector)
model_cats = get_model_categories(doc)
elements_by_category = [get_elements_by_category(cat) for cat in model_cats]
# Salida
OUT = elements_by_category

```

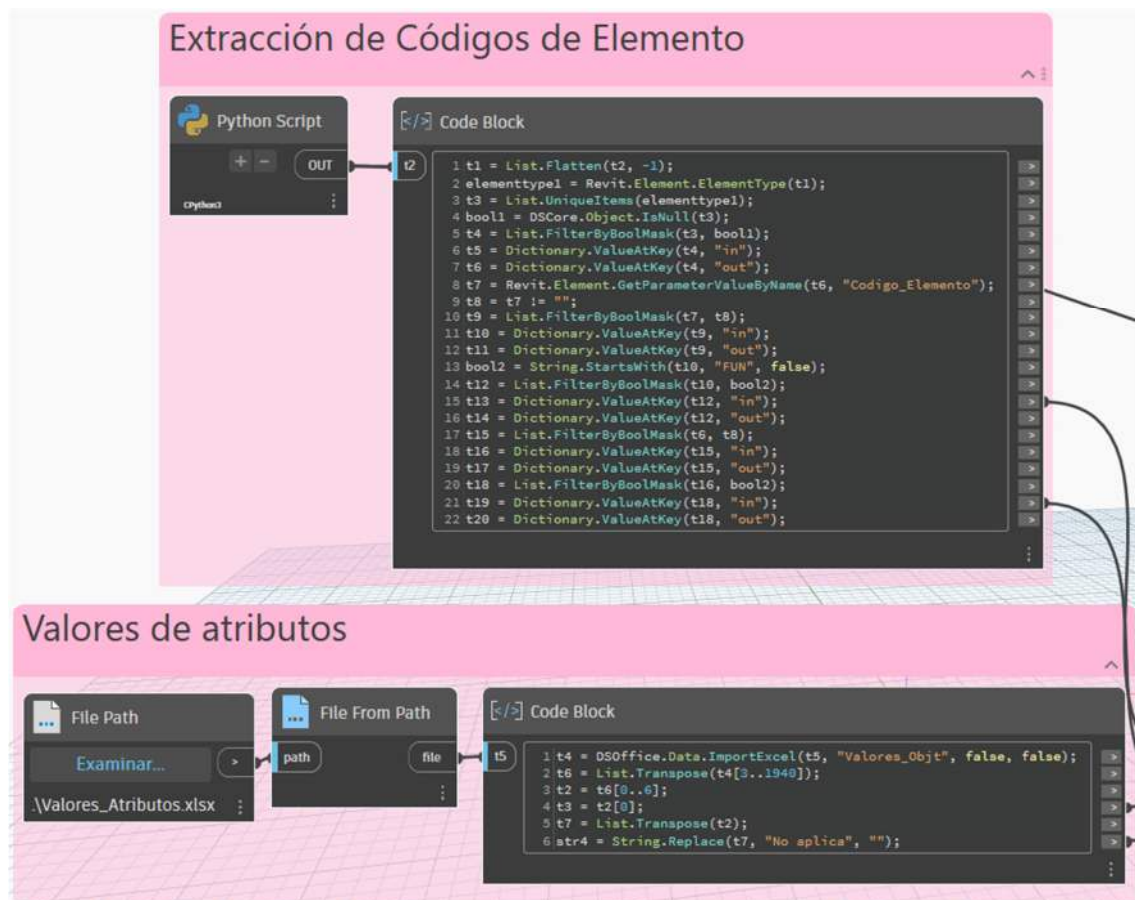


Figura 3.5.24. Entradas para asignación de valores dependientes

A la vez, se extraen los datos del Excel que ha sido exportado desde la base de datos de la organización, el cual contiene en cada fila los valores a asignar a cada tipo de objeto asociados a cada código y descripción de elemento. En este caso, se utilizan las definiciones establecidas en la versión 2 del sistema de clasificación SCFClass, añadiendo además una relación con acrónimos de disciplinas y subdisciplinas.

Codigo_Elemento	Descripcion_Elemento	Disciplina	Sub disciplina	Clase IFC	Tipo predefinido IFC
FUN.ENT.010.010	Origen de coordenadas	URB	ENT	Ifc Cartesian Point	No aplica
FUN.ENT.010.020	Red o malla	URB	ENT	IfcGrid	USERDEFINED
FUN.ENT.020.010	Árbol	URB	ENT	Ifc Geographic Element	VEGETATION
FUN.ENT.020.020	Arbusto	URB	ENT	Ifc Geographic Element	VEGETATION
FUN.ENT.030.010	Parcela completa	URB	ENT	Ifc Annotation	ASSUMEDAREA
FUN.ENT.030.020	Subparcela	URB	ENT	Ifc Annotation	ASSUMEDAREA
FUN.ENT.030.030	Área	URB	ENT	Ifc Annotation	ASSUMEDAREA
FUN.ENT.030.040	Vallado parcelario	URB	ENT	IfcWall	USERDEFINED

Tabla 3.5.12. Extracto de relaciones de valores de parámetros

Para la asignación de los valores dentro de los parámetros de los elementos, se generan dos códigos enfocados en la asignación para tipos y para ejemplares. Las configuraciones IFC se consideran de ejemplar debido a que alguna instancia de familia podría tener definiciones diferentes al resto. Por el contrario, disciplina y subdisciplina se consideran siempre de tipo, ya que el código de la clasificación también está asociado al tipo.

Durante el proceso de definición, es importante asignar dentro de los elementos de tipo los parámetros de tipo y dentro de las instancias cada parámetro de ejemplar, por ello, se genera otro código Python que permita extraer de cada tipo de familia los ejemplares que contiene para poder asignarles dichos parámetros específicos de cada objeto.

```

import System
import clr
clr.AddReference('RevitAPI')
from Autodesk.Revit.DB import *
clr.AddReference("RevitServices")
import RevitServices
from RevitServices.Persistence import DocumentManager
# Recolectar todas las instancias de cada tipo de familia
def InstancesOfFamilyType(ft, doc):
    collector = FilteredElementCollector(doc)
    bic = System.Enum.ToObject(BuiltInCategory, ft.Category.Id.IntegerValue)

```



```

    collector.OfCategory(bic)
    return [x for x in collector.ToElements() if x.GetTypeId().IntegerValue ==
ft.Id.IntegerValue]
famtypes = UnwrapElement(IN[0])
inputdoc = UnwrapElement(IN[2])
if not inputdoc: doc = DocumentManager.Instance.CurrentDBDocument
elif inputdoc.GetType().ToString() == "Autodesk.Revit.DB.RevitLinkInstance": doc =
inputdoc.GetLinkDocument()
elif inputdoc.GetType().ToString() == "Autodesk.Revit.DB.Document": doc = inputdoc
else: doc = DocumentManager.Instance.CurrentDBDocument
if isinstance(IN[0], list): OUT = [InstancesOfFamilyType(x, doc) for x in famtypes]
else: OUT = InstancesOfFamilyType(famtypes, doc)

```



Figura 3.5.25. Definición de valores dependientes según clasificación

Siempre que la tabla de datos relacionados se haya cargado desde el archivo XLSX, los parámetros hayan sido definidos previamente y los valores del código del elemento del sistema de clasificación se encuentren rellenos en los ejemplares, la automatización funcionará correctamente.

Para aquellos objetos que no tengan definido el valor del código, el valor asociado permanecerá también vacío y, en el caso de que los objetos ya tengan definidos valores, estos se sobrescriben para evitar posibles errores manuales.

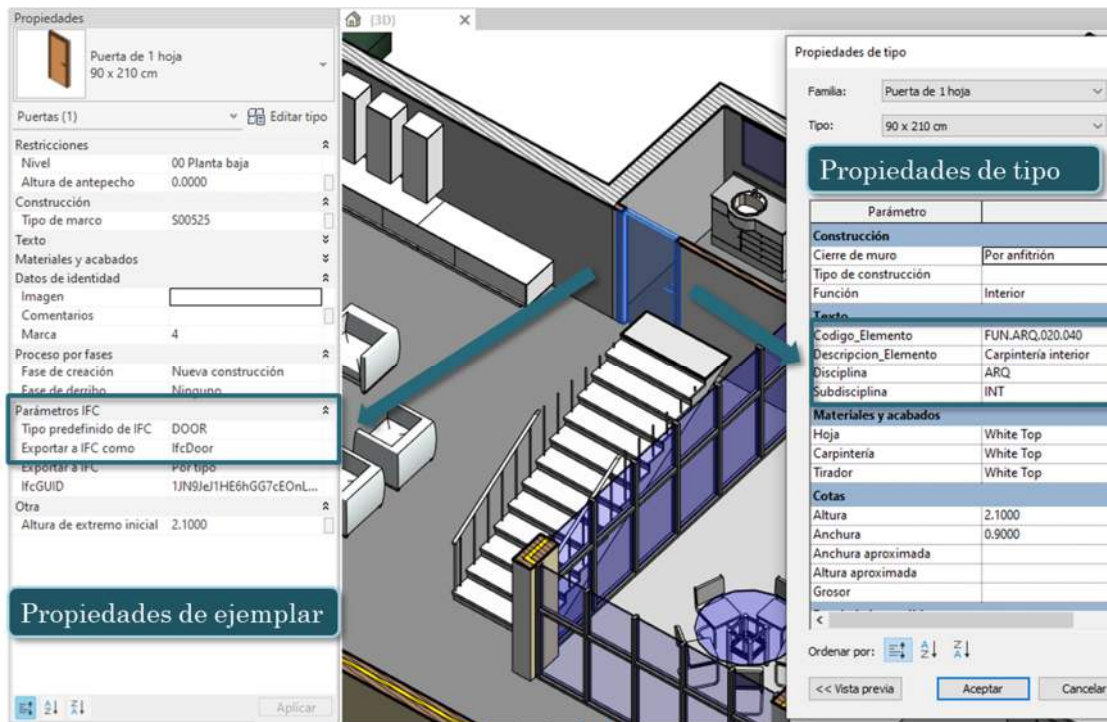


Figura 3.5.26. Valores rellenados automáticamente desde Dynamo

Una aproximación al proceso realizado utilizando Blender y Bonsai sería la definición de las relaciones desde un código Python que acceda al modelo IFC de trabajo y actualice los datos de los valores. Utilizando propiedades de planificación de actividades, sabiendo por ejemplo el código de una actividad, es posible definir para cada objeto otros datos relacionados con tiempos, tipos de construcción o información relacionada.

```
import bpy
import ifcopenshell
import ifcopenshell.util.element
def debug_print_psets(elem):
    for rel in getattr(elem, "IsDefinedBy", []):
        if rel.is_a("IfcRelDefinesByProperties"):
            pset = rel.RelatingPropertyDefinition
            for prop in getattr(pset, "HasProperties", []):
                value = prop.NominalValue.wrappedValue if prop.NominalValue else
                None
                print(f" {prop.Name}: {value}")
# Trabajar en el IFC actual
ifc_file_path = bpy.context.scene.BIMProperties.ifc_file
ifc_file = ifcopenshell.open(ifc_file_path)
elem_updated = 0
for elem in ifc_file.by_type("IfcElement"):
    #Revisar en todos los conjuntos de propiedades
    for rel in getattr(elem, "IsDefinedBy", []):
        if rel.is_a("IfcRelDefinesByProperties"):
            pset = rel.RelatingPropertyDefinition
            props = {}
```

```

for prop in getattr(pset, "HasProperties", []):
    if prop.is_a("IfcPropertySingleValue"):
        props[prop.Name] = prop.NominalValue.wrappedValue if
prop.NominalValue else None
if props.get("Codigo_Actividad") == "CARP0001N":
    updates = {}
    if props.get("Inicio_Actividad") != "20/05/2024":
        updates["Inicio_Actividad"] = "20/05/2024"
    if props.get("Fin_Actividad") != "30/05/2024":
        updates["Fin_Actividad"] = "30/05/2024"
    if updates:
        ifcopenshell.api.run(
            "pset.edit_pset", ifc_file, pset=pset, properties=updates)
        walls_updated += 1
        print(f"Updated properties: {updates}")
#Guardar los cambios en el archivo IFC
ifc_file.write(ifc_file_path)

```

Como resultado de la anterior operación, se puede apreciar que los valores de las propiedades dentro de los objetos del IFC en Blender se han definido correctamente para el código de actividad específico.

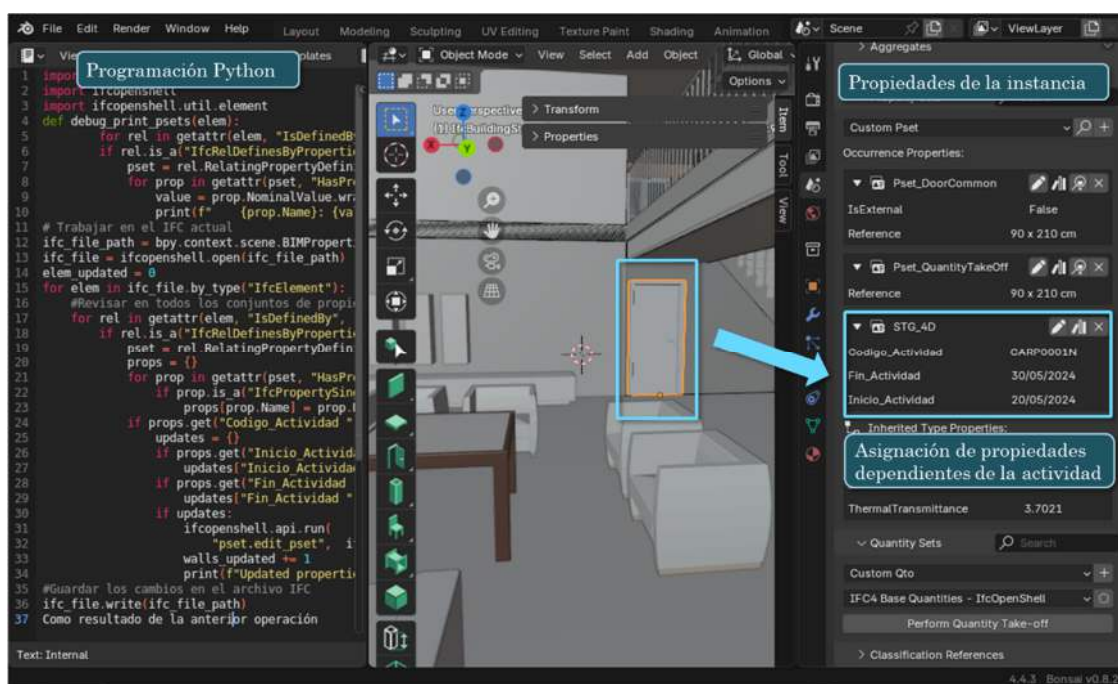


Figura 3.5.27. Valores rellenados automáticamente desde Blender

3.5.7. Configuración de exportación IFC

Una importante aplicación práctica de la sistematización se realiza en el contexto de la configuración de exportación de los archivos IFC, realizando diferentes operaciones de mapeado y conversión para poder obtener el archivo OpenBIM desde el nativo. Esto engloba configuraciones relativas a las clases de cada entidad, la organización de propiedades y cantidades y el cumplimiento de la estructura de datos del esquema IFC.

En el caso concreto de Autodesk Revit, los atributos de entidades IFC no existen como tal, sino que son parámetros del programa definidos con otros nombres o bien necesitan ser creados para poder realizar la exportación correctamente. A continuación, se detalla un listado con los más relevantes incluyendo sus principales características:

Parámetro	Descripción	Nivel IFC relacionado	Nativo de Revit
Número de proyecto	Nombre o código del proyecto.	IfcProject.Name	Sí
Nombre del edificio	Nombre del edificio.	IfcBuilding.Name	Sí
SiteName	Nombre del emplazamiento del proyecto.	IfcSite.Name	No
IfcName	Nombre del objeto IFC, reemplaza el nombre por defecto del modelo.	IfcObject.Name	No
IfcTag	Etiqueta específica de la instancia IFC.	IfcObject.Tag	Sí
IfcExportAs	Define la clase IFC a la que se exportará el objeto BIM.	Determina la entidad IFC	Sí
IfcDescription	Descripción adicional del objeto para propósitos de documentación.	IfcObject.Description	No
IfcGUID	Identificador único global del objeto.	IfcGloballyUniqueId	Sí
IfcExportType	A veces usado junto con IfcExportAs para definir subtipos específicos.	Tipo predefinido de la entidad IFC	Sí
IfcObjectType	Tipo de objeto IFC, describe la función o categoría personalizada.	IfcObject.ObjectType	No

Tabla 3.5.13. Atributos IFC dentro de Autodesk Revit

En casos anteriores ya se han definido valores como la clase y el tipo predefinido IFC. Para los valores de nombres y datos relacionados con el proyecto, edificio o emplazamiento, al tratarse de la misma información para todos los modelos de un mismo proyecto, basta con establecer el código del expediente y luego automatizar el volcado del resto de los datos desde una tabla extraída de la base de datos que incorpore la información del contrato.

Parámetro	Valor
Codigo_Expediente	208300312
Descripcion_Expediente	OBRAS DE EJECUCIÓN DEL PROYECTO DE CONSTRUCCIÓN DE...
Inicio_Expediente	10/02/2024
Fin_Expediente	30/10/2024
Empresa_Responsable	UTE
Enlace_Ubicacion	https://acc.autodesk.eu/docs/files/projects/
Sistema_Coordenadas	ETRS89 UTM-30N
Número de proyecto	208300312
Nombre_Proyecto	Proy_03_Edificio_Viviendas
SiteName	Emplazamiento Madrid
Nombre de edificio	Edificio de viviendas

Tabla 3.5.14. Propiedades generales del modelo BIM

De estos parámetros, todos los que Revit no gestiona de manera interna y que han sido creados por el usuario, necesitan una configuración adicional para la exportación IFC, ya que el programa no sabe por defecto en que pset añadirlos o a qué entidad IFC van asociados. Para realizar este trabajo, se utiliza una plantilla TXT que realiza un proceso de personalización de propiedades IFC en el cual se especifica los nombres de los psets nuevos y se indica qué parámetros van en cada uno, identificando además si cada parámetro es de tipo o ejemplar y a las clases IFC que irá enlazado al exportarse.

Junto a estos parámetros de proyecto, el TXT permite añadir todas las definiciones de parámetros para otras clases IFC, sean generales para *IfcElement*, para espacios como *IfcSpace* o bien para determinadas clases como *IfcWall*, *IfcDoor*, *IfcSlab*, etc. Este archivo puede ser creado mediante programación con Python desde Jupyter Notebook, extrayendo los datos desde la base de datos y reestructurando el XLSX para convertirlo en el TXT.

```
import pandas as pd
from collections import defaultdict
import os
#Ruta del archivo Excel de entrada y salida TXT
ruta_excel = r"C:\Users\Samuel\ G_Configuracion_IFC\propiedades_BBDD.xlsx"
ruta_salida = r"C:\Users\Samuel\ G_Configuracion_IFC\propiedades_IFC.txt"
#Mapeo de tipos desde el Excel al formato IFC
tipo_mapeo = {
    "texto": "Text", "booleano": "Boolean", "entero": "Integer", "numero": "Number"}
#Leer archivo Excel
df = pd.read_excel(ruta_excel)
#Verificación de columnas requeridas
columnas_requeridas = ["Grupo", "Nombre_Atributo", "Tipo_Dato", "Aplicación",
    "Clase_IFC"]
if not all(col in df.columns for col in columnas_requeridas):
    raise ValueError(f'El Excel debe contener las columnas: {',
        '.join(columnas_requeridas)}')
```



```

#Agrupación por PropertySet
property_sets = defaultdict(list)
for _, row in df.iterrows():
    grupo = str(row["Grupo"]).strip()
    nombre = str(row["Nombre_Atributo"]).strip()
    tipo_dato = str(row["Tipo_Dato"]).strip().lower()
    aplicacion = str(row["Aplicación"]).strip().lower()
    clase_ifc = str(row["Clase_IFC"]).strip()
    tipo_aplicacion = "I" if aplicacion == "instancia" else "T"
    tipo_convertido = tipo_mapeo.get(tipo_dato)
    if not tipo_convertido:
        raise ValueError(f"Tipo de dato no reconocido: '{tipo_dato}' en atributo '{nombre}'")
    clave = (grupo, tipo_aplicacion, clase_ifc)
    property_sets[clave].append((nombre, tipo_convertido))
#Crear carpeta si no existe
os.makedirs(os.path.dirname(ruta_salida), exist_ok=True)
#Escribir archivo de salida
with open(ruta_salida, "w", encoding="utf-8") as f:
    for (grupo, tipo_aplicacion, clase_ifc), atributos in property_sets.items():
        f.write(f"PropertySet:\t{grupo}\t{tipo_aplicacion}\t{clase_ifc}\n")
        for nombre_atributo, tipo in atributos:
            f.write(f"\t{nombre_atributo}\t{tipo}\n")

```

Grupo	Nombre_Atributo	Tipo_Dato	Aplicación	Clase_IFC
Expediente	Inicio_Expediente	Texto	Instancia	IfcProject
Expediente	Fin_Expediente	Texto	Instancia	IfcProject
Expediente	Codigo_Expediente	Texto	Instancia	IfcProject
Expediente	Descripcion_Expediente	Texto	Instancia	IfcProject
Expediente	Empresa_Responsable	Texto	Instancia	IfcProject
Generales	Disciplina	Texto	Tipo	IfcElement
Generales	Es_Exterior	Booleano	Instancia	IfcWall, IfcDoor, IfcSlab, IfcColumn

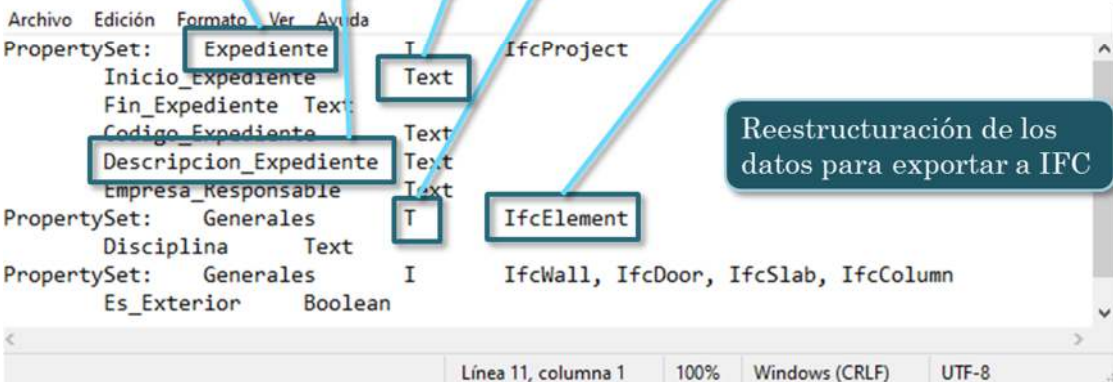


Figura 3.5.28. TXT para exportación de propiedades IFC

Además de la configuración de exportación de propiedades personalizadas, en Revit existe un segundo TXT que se encarga de realizar una conversión de parámetros del programa nativo a definiciones específicas del esquema IFC. En el caso de estudio se han definido los nombres de las propiedades en español, por lo que esta configuración

puede ser utilizada para mapear los nombres de Revit a los del esquema y además añadirlos en el pset que corresponda. Con el siguiente script se toman los datos del XLSX de la base relacional y se convierten a la estructura que del TXT de Revit.

```
import pandas as pd
import os
#Rutas de entrada y salida
ruta_excel = r"C:\G_Configuracion_IFC\mapeado_atributos_ifc.xlsx"
ruta_salida = r"C:\G_Configuracion_IFC\mapeo_atributos.txt"
#Leer Excel
df = pd.read_excel(ruta_excel)
#Validar columnas esperadas
columnas_esperadas = ["Conjunto IFC", "Nombre IFC", "Nombre"]
if not all(col in df.columns for col in columnas_esperadas):
    raise ValueError(f"El Excel debe contener las columnas: {';'.join(columnas_esperadas)}")
#Crear carpeta si no existe
os.makedirs(os.path.dirname(ruta_salida), exist_ok=True)
# Guardar archivo TXT sin encabezados, separado por tabuladores
df.to_csv(ruta_salida, sep='\t', index=False, header=False)
```

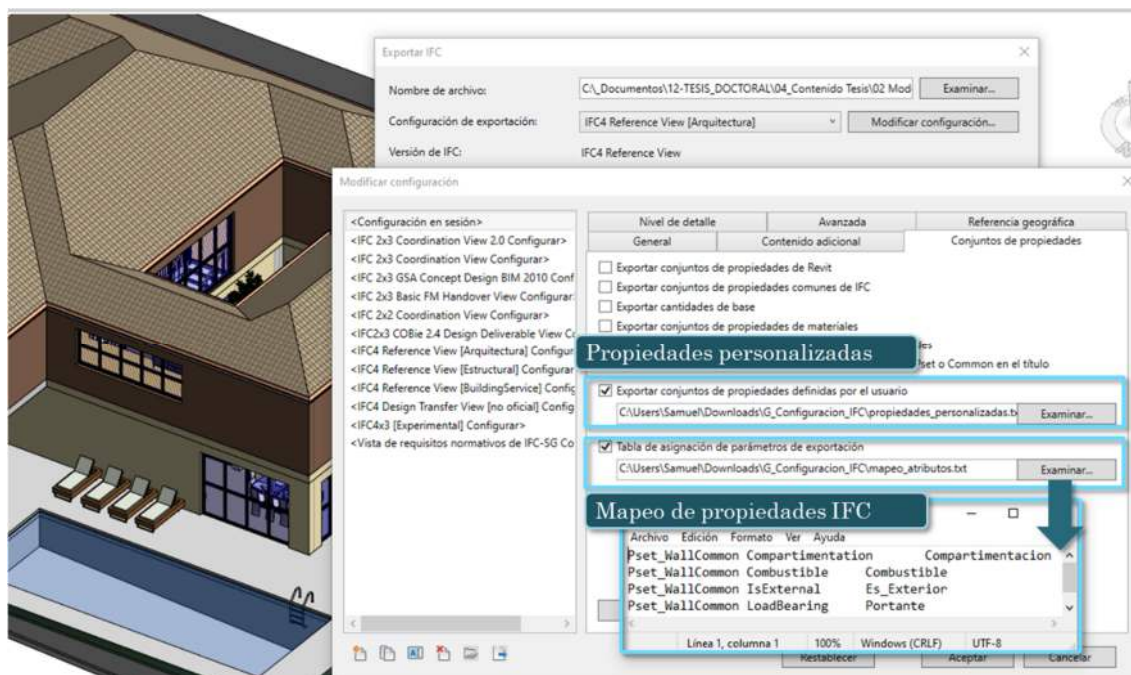


Figura 3.5.29. Exportación de propiedades IFC desde Revit

Por último, las clasificaciones deben exportarse también como *IfcClassification*. Para ello, se genera una pequeña automatización con Dynamo que concatena el código y descripción de la clasificación, divididos por un “:” y los añade en un parámetro denominado *Clasificacion_Funcion*, el cual se asigna en la configuración de exportación para que puedan generarse las definiciones IFC. En dicha configuración, se establece la información general del sistema de clasificación utilizado, como el nombre, la tabla seleccionada, la versión y la fecha de actualización.

```

doc = DocumentManager.Instance.CurrentDBDocument
#Categorías de la disciplina arquitectura
categorias_constructivas = [BuiltInCategory.OST_Walls...]
#Función para coleccionar los tipos de las categorías
def obtener_tipos_elemento(categoria):
    return
    FilteredElementCollector(doc).OfCategory(categoria).WhereElementIsElementType
    ().ToElements()
tipos_todos = []
for cat in categorias_constructivas:
    tipos_todos.extend(obtener_tipos_elemento(cat))
TransactionManager.Instance.EnsureInTransaction(doc)
contador_actualizados = 0
for tipo in tipos_todos:
    param_codigo = tipo.LookupParameter("Codigo_Elemento")
    param_desc = tipo.LookupParameter("Descripcion_Elemento")
    param_destino = tipo.LookupParameter("Clasificacion_Funcion")
    if param_codigo and param_desc and param_destino:
        codigo = param_codigo.AsString()
        descripcion = param_desc.AsString()
        if codigo and descripcion:
            nuevo_valor = "{}:{}".format(codigo.strip(), descripcion.strip())
            param_destino.Set(nuevo_valor)
            contador_actualizados += 1
TransactionManager.Instance.TransactionTaskDone()
OUT = "Elementos actualizados: {}".format(contador_actualizados)

```

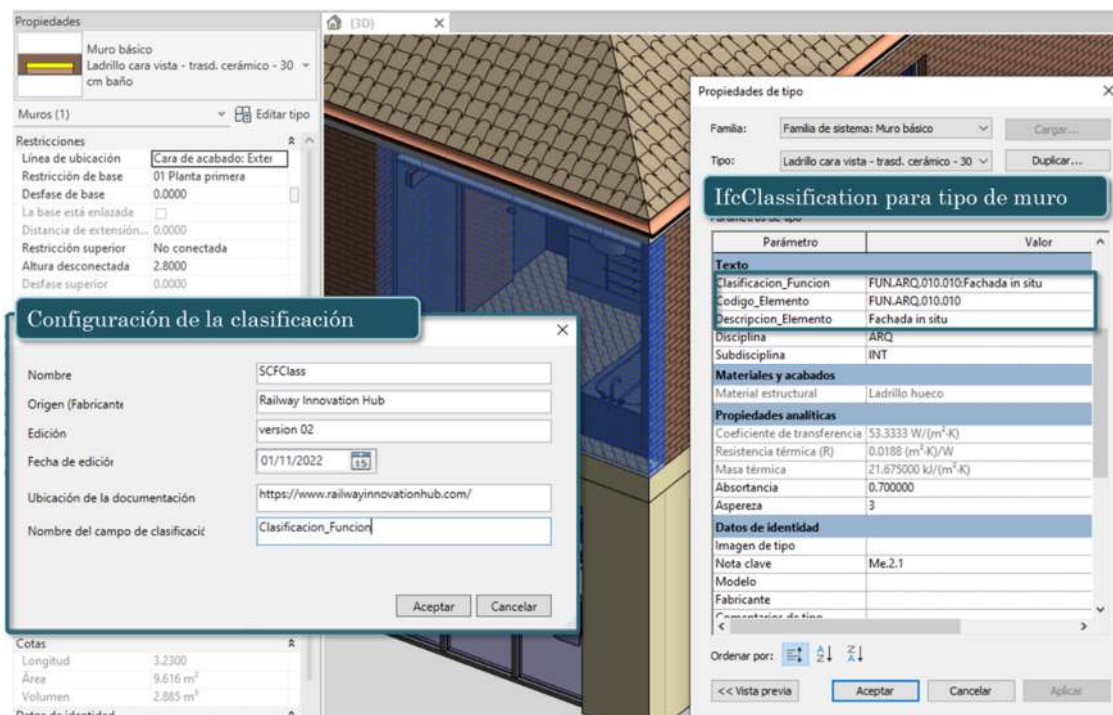


Figura 3.5.30. Configuración de exportación de clasificaciones en Revit

El proceso definido hasta el momento muestra una metodología eficiente para la integración de datos desde la BBDD relacional en los modelos BIM, mediante la extracción de información, la automatización en la creación de propiedades, la asignación de valores predefinidos y la configuración de exportación a IFC, apostando por un enfoque sistemático y abierto que permite optimizar la gestión del proyecto a lo largo de todo el ciclo de vida.

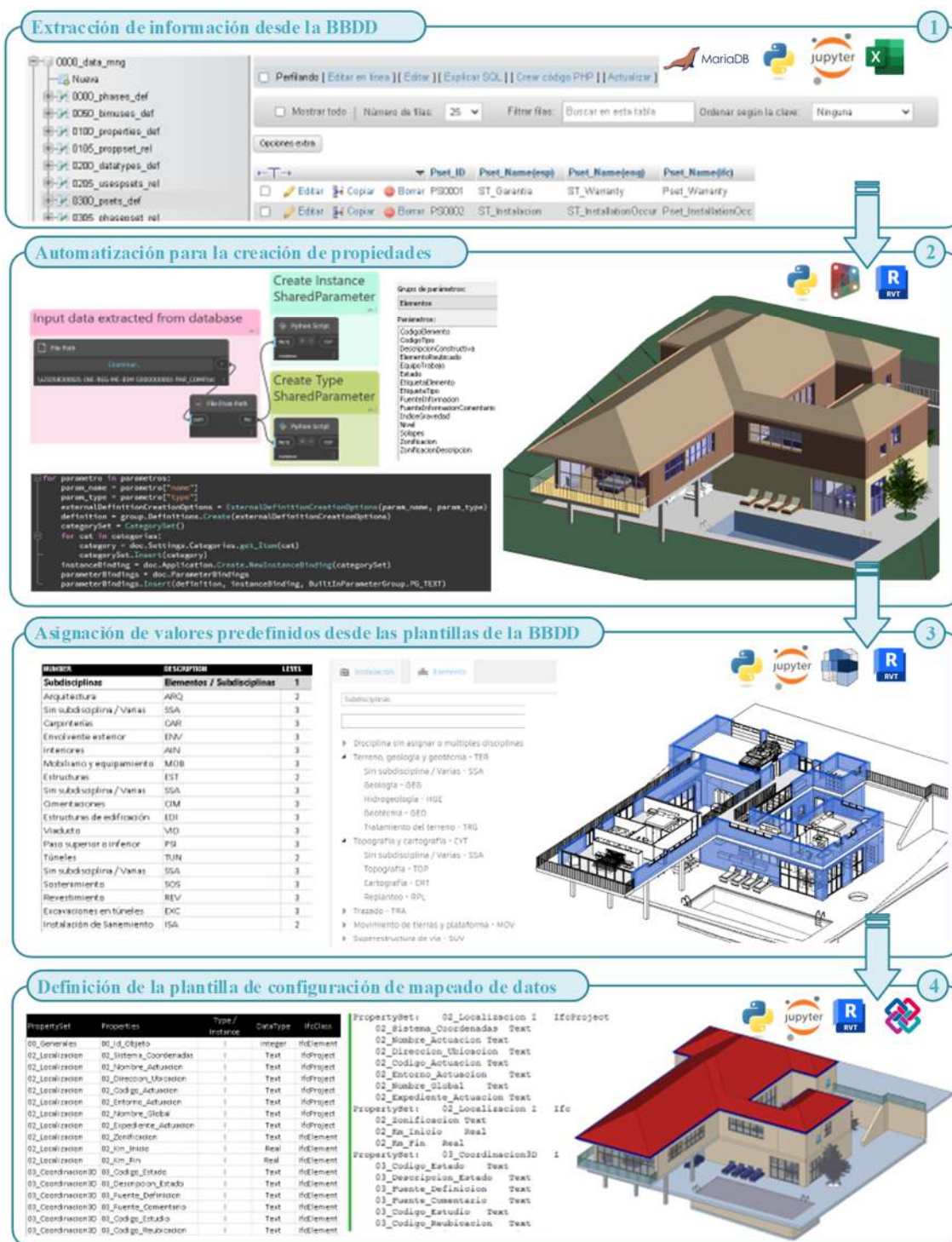


Figura 3.5.31. Proceso de generación de los modelos openBIM

3.5.8. Extracción de datos IFC

En este caso se llevan a cabo diferentes análisis del esquema IFC con el objetivo de realizar procesos de extracción de la información relevante que se encuentra almacenada en los modelos. Utilizando IfcOpenShell como herramienta de acceso y manipulación de datos dentro de entornos de programación como Jupyter Notebook y librerías como Pandas para la estructuración de los datos, se exploran los diferentes tipos de entidades IFC, así como objetos instanciados, sus propiedades asociadas y las relaciones que estructuran el modelo. Estos análisis permiten una comprensión más profunda de la organización y el contenido del archivo IFC, sentando las bases para posteriores tareas de procesamiento e integración de datos.

3.5.8.1. Muros

Según el estándar IFC²⁰⁵, un muro representa una construcción vertical que puede delimitar o subdividir espacios. Los muros suelen ser elementos planos verticales, o casi verticales, a menudo diseñados para soportar cargas estructurales. Dentro del esquema IFC, los muros se definen como instancias *IfcWall*. Adicionalmente, existe otro tipo de instancia denominada *IfcWallStandardCase* la cual en futuras versiones quedará obsoleta ya que estará englobada en la propia *IfcWall*. La diferencia entre estas clases es que *IfcWall* engloba todo tipo de muros, mientras que *IfcWallStandardCase* solo contempla muros básicos con un solo espesor a lo largo de su recorrido, teniendo como eje base del camino una línea recta o un arco. No obstante, esta clase puede tener desplazamientos del camino del muro, una o varias capas y una altura constante o variable a lo largo de su recorrido.

En las últimas versiones de IFC, los muros básicos contienen un nuevo concepto denominado *IfcMaterialLayerSetUsage* con el que se definen las propiedades específicas de este tipo de muros dentro de *IfcWall*.

Para el resto de casos, no se utiliza la definición anterior, ya que se trata de muros que pueden tener un espesor variable a lo largo de su recorrido (por ejemplo, muros poligonales), muros con secciones transversales no rectangulares (por ejemplo, muros de contención en forma de *L*), y muros inclinados o muros que tienen solo geometría *Brep* (modelo de representación de contorno) o *NURBS* (B-splines racionales no uniformes) y otras superficies suaves.

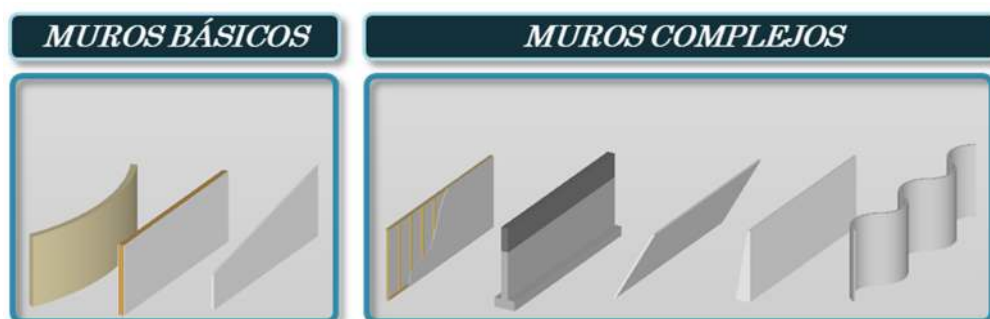


Figura 3.5.32. Tipos de muros básicos y complejos

²⁰⁵ IFC Specifications Database. Official final buildingSMART standard. IFC 4.3.2.20240128 (IFC4X3_ADD2).

A nivel de estructura dentro del estándar, los muros se encuentran dentro de los elementos de construcción, como otros elementos del esquema como suelos, puertas, cubiertas, etc.

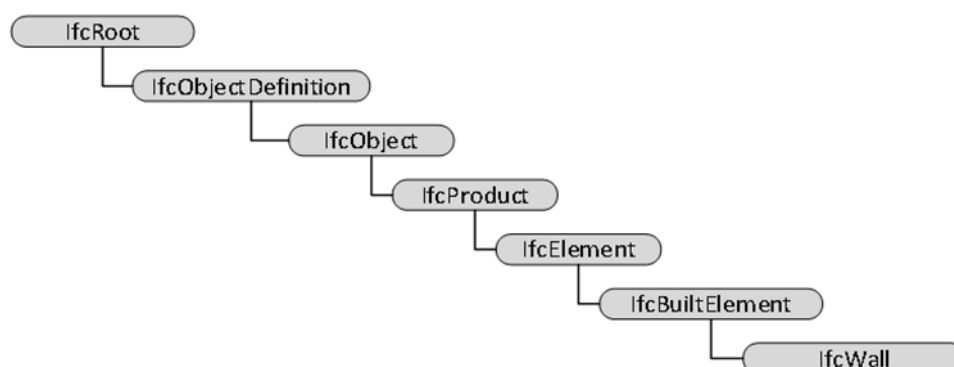


Figura 3.5.33. Esquema jerárquico de instancia de muro

Al igual que el resto de elementos dentro del esquema IFC, los muros pueden tener diferentes tipos de relaciones según las clases o entidades con las que se conecten. La principal relación que siempre debe existir es la del contenedor en el que está definido el muro. Por lo general, este es un nivel del edificio con clase *IfcBuildingStorey* y se relaciona a través de la definición *IfcRelContainedInSpatialStructure*.

///Relación de muros con el nivel asociado

#779=IFCRELCONTAINEDINSPATIALSTRUCTURE('29TFp',#18,\$,('#137,#183'),#111);

///Muros asociados al nivel

∨ [#137=IFCWALL('1fRQ',#18,'MF3:162779',\$,, '#121,#136','162779')]

∨ [#183=IFCWALL('1fRr9',#18,'MF3:162779',\$,, '#171,#182','162793')]

///Nivel en el que se asocian los muros

∨ [#111=IFCBUILDINGSTOREY('29TFpHBYX',#18,'Nivel 1',\$, 'N1',#110,\$, 'Nivel1',.ELEMENT.,0.)]

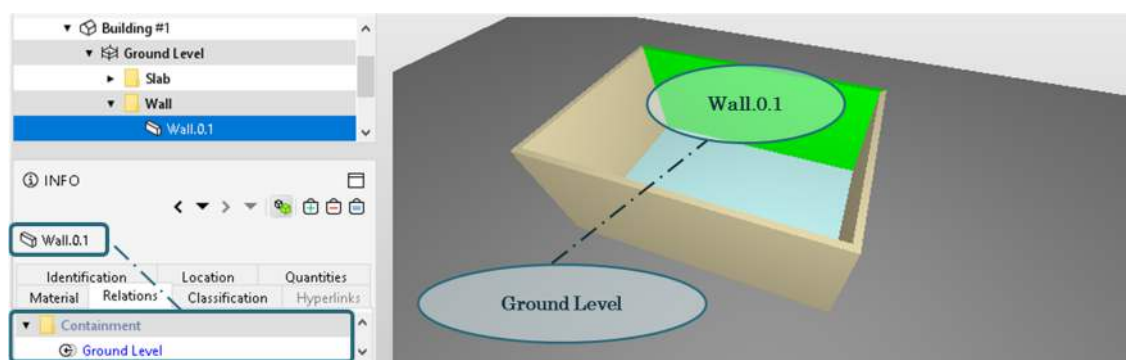


Figura 3.5.34. Relación de muro con su nivel asociado

A diferencia de muchas otras relaciones que suelen ser del tipo 1 – 1, en la imagen anterior se puede observar que la relación con el nivel es de 1 – varios, por lo que en una definición se relacionan todas las instancias contenidas en el nivel. Este mismo muro está relacionado con el proyecto debido a que el nivel se encuentra contenido

dentro del edificio, este a su vez en el emplazamiento y este último dentro del proyecto.²⁰⁶

También en la imagen se aprecia que el muro está contenido en el nivel de suelo, ya que el nivel asociado a un muro siempre se considera el que se encuentra en su arranque, es decir, en la cara inferior de su volumetría.

Para obtener los muros establecidos dentro de un nivel, es posible buscar los elementos relacionados que se encuentran contenidos en dicho nivel. Para ello, se accede al nivel y, dentro de este, se utiliza la definición de relación *ContainsElements* para luego llegar a los *RelatedElements*. En el siguiente ejemplo se muestra cómo obtener los nombres e identificadores de los muros contenidos dentro del 'Nivel 1' del proyecto a partir del filtrado de elementos que contengan solo la clase *IfcWall*.

```
model = ifcopenshell.open('Muros_por_niveles.ifc')
bdsty = model.by_type('IfcBuildingStorey')[0]
rltdele = bdsty.ContainsElements[0].RelatedElements
wallsinlvl = []
for w in rltdele:
    aa = w.ObjectType
    bb = w.ObjectType, w.GlobalId
    if 'Muro' in aa:
        wallsinlvl.append(bb)
df = pd.DataFrame(wallsinlvl)
df.columns = ['Nombre de tipo', 'IfcGUID']
display(df)
```

Id	Nombre de tipo	IfcGUID
0	Muro básico:Partición con capa de yeso	1fRr9QxcvC4x5k5Gw6DFp4
1	Muro básico:Partición con capa de yeso	1fRr9QxcvC4x5k5Gw6DFps
2	Muro básico:Muro medianero con capa de yeso	1fRr9QxcvC4x5k5Gw6DFk_
3	Muro básico:Muro medianero con capa de yeso	1fRr9QxcvC4x5k5Gw6DFIL
4	Muro básico:Por defecto - 10 cm	1fRr9QxcvC4x5k5Gw6DFeu

Tabla 3.5.15. Extracto de ejemplares de muro de planta baja

Al igual que el resto de elementos de construcción englobados en *IfcBuiltElement*²⁰⁷, los muros heredan parte de sus propiedades de los tipos de muros *IfcWallType*. Las instancias de un mismo tipo, pueden compartir diferentes propiedades como información de materiales, espesores de capas y formas de la volumetría. No obstante, una instancia de muro solo puede pertenecer a un tipo concreto.

²⁰⁶ IFC 4.3.2.20240128 (IFC4X3_ADD2). Definición y relaciones de *IfcProject*.

²⁰⁷ A partir de la versión IFC4x3 la entidad pasa de ser denominada *IfcBuildingElement* a *IfcBuiltElement*, convirtiéndose en una clase no abstracta.

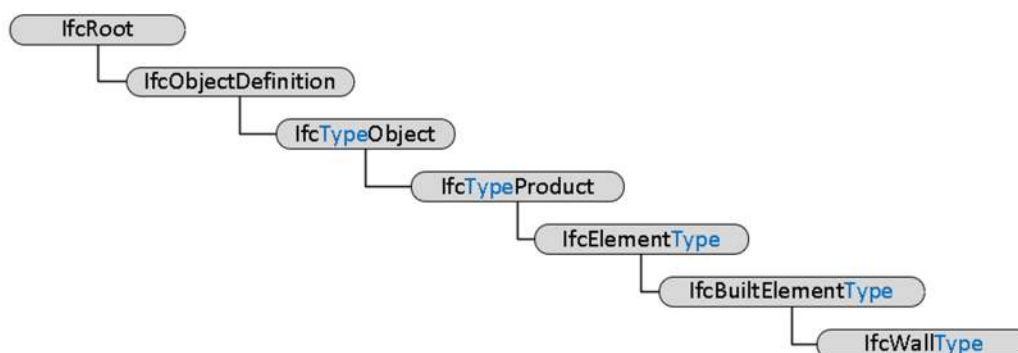


Figura 3.5.35. Esquema jerárquico de tipos de muro

La relación que conecta cada una de las instancias de muro con su tipo es la denominada *IfcRelDefinesByType*, la cual es la que siempre se utiliza para establecer relaciones de este tipo para cualquier clase de objeto IFC.

///Relación de muros con su tipo

#256=IFCRELDEFINESBYTYPE('23fZ1IO\$rxXc',#18,\$,\$,(\$137,#183,#206,#229),#156);

///Instancias de muro asociadas al tipo

∨ [#137=IFCWALL('1YQexjo\$XEYg',#18,'Tabique10:162774',\$,'MuroE1sico:10',#121,#136,'162774')]

∨ [#183=IFCWALL('1YQexjo\$XEYg',#18,'Tabique10:162775',\$,'MuroE1sico:10',#171,#182,'162775')]

///Tipo al que pertenecen las instancias de muro

∨ [#156=IFCWALLTYPE('15W4T\$rob6kQ',#18,'Tabique10',\$,\$,(\$168),\$',14460,\$,..STANDARD.)]

Por otro lado, dentro de cualquier aplicación BIM, al seleccionar alguna de las instancias, se podrá ver el tipo de muro relacionado.

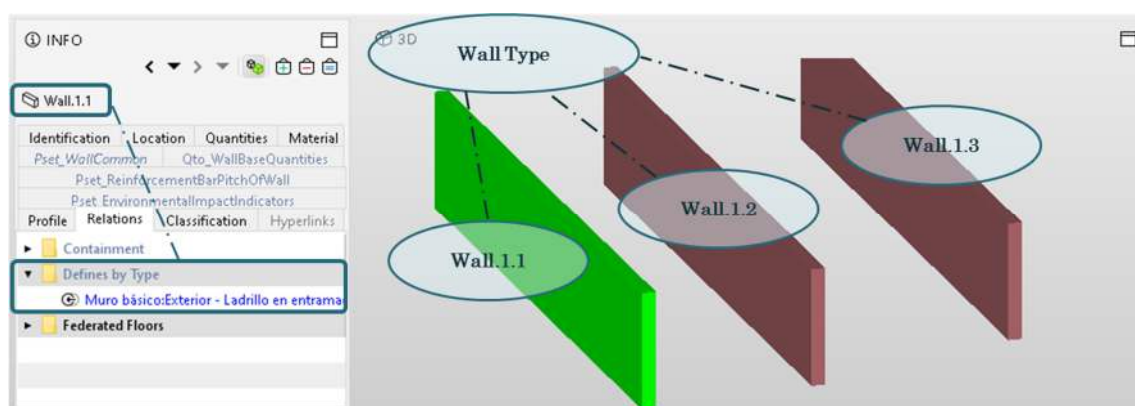


Figura 3.5.36. Relaciones de tipo con instancias de muro

Como en muchas otras entidades, este tipo de muro contiene una enumeración de tipos predefinidos que identifican la clase de muro y que se define en el atributo *PredefinedType*. Algunos ejemplos se muestran a continuación:

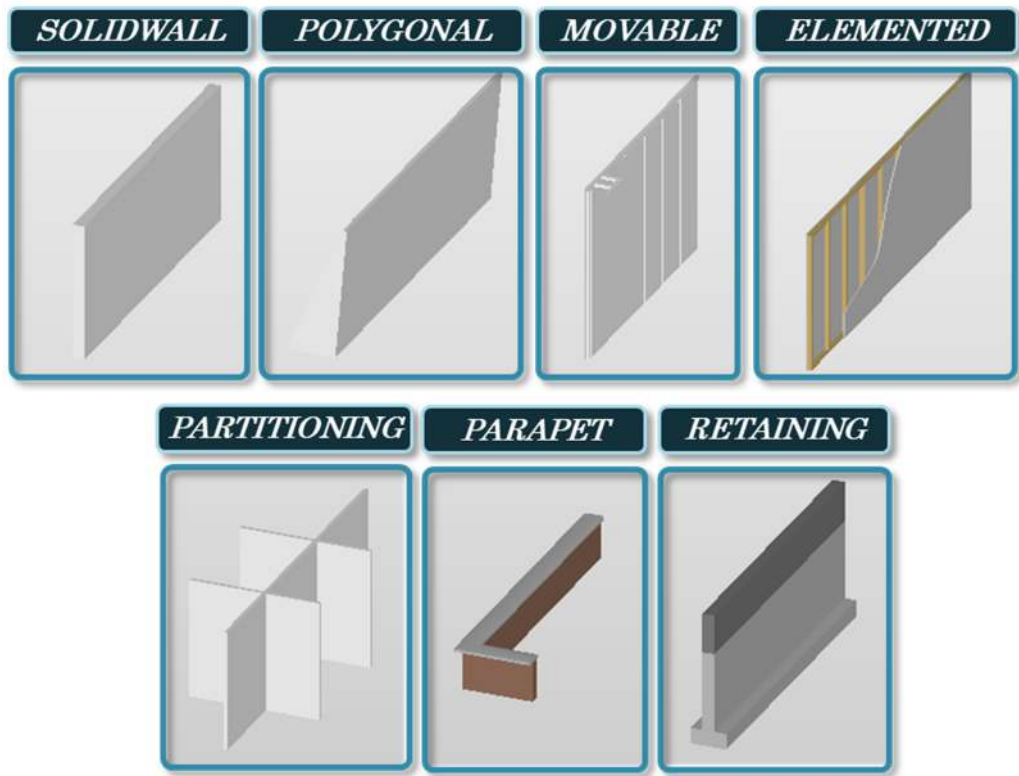


Figura 3.5.37. Tipos predefinidos de muros IFC

Es importante la verificación de estos tipos predefinidos dependiendo de la versión de IFC a utilizar. Por ejemplo, a partir de la versión IFC4_ADD2_TC1 aparecen los tipos *MOVABLE*, *PARAPET*, *PARTITIONING* y *SOLIDWALL*²⁰⁸.

Cuando no exista un tipo específico en la enumeración del objeto, el tipo predefinido será *USERDEFINED*, y en el atributo de ObjectType se definirá el tipo personalizado. Algunos ejemplos pueden ser muros diafragma *DIAPHRAGMWALL*, muros combinados *COMBIWALL* o muros pantalla *SLURRYWALL*.²⁰⁹

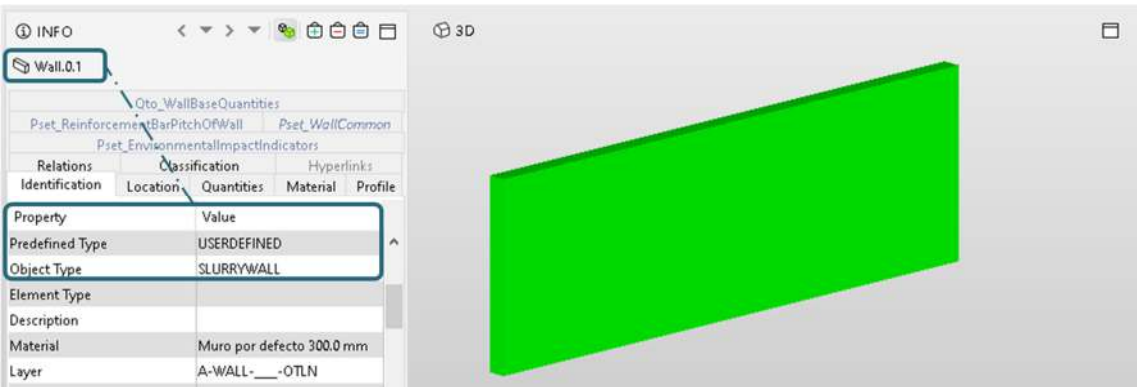


Figura 3.5.38. Definición de tipo predefinido personalizado

²⁰⁸ Enumeración de tipos predefinidos de muros. IFC4_ADD2_TC1 - 4.0.2.1 [Official]. 1996-2020 buildingSMART International Ltd.

²⁰⁹ UC 00 – Minimum Requirements for Walls. BIM IDS v1.5. Ifc Class and Type. TechBiM and MatBiM.

Para obtener los datos de cada tipo predefinido y tipo personalizado de muro, es necesario iterar todos los tipos existentes extrayendo la información relevante y, en el caso del valor de tipo de objeto personalizado, hay que buscar el valor dentro de un ejemplar de muro, ya que en los atributos del tipo de muro puede no aparecer, dependiendo del software del que se haya extraído el archivo.

```
model = ifcopenshell.open('Muros_Predefinidos_4x3.ifc')
pdtpes = model.by_type('IfcWallType')
walls = []
for p in pdtpes:
    nm = p.Name, p.PredefinedType, p.Types[0].RelatedObjects[0].ObjectType,
    p.GlobalId
    walls.append(nm)
df2 = pd.DataFrame(walls)
df2.columns = ['Nombre de tipo', 'Tipo predefinido', 'Tipo personalizado', 'IfcGUID']
display(df2)
```

Id	Nombre de tipo	Tipo predefinido	Tipo personalizado	IfcGUID
0	Muro básico:Partición con capa de yeso	USERDEFINED	SLURRYWALL	0t2nGKfbH1Vh HNqEzOo50
1	Muro básico:Partición con capa de yeso	MOVABLE	NA	2nmIrxOwIZD3 NHBIBkKo
2	Muro básico:Partición con capa de yeso	ELEMENTEDWALL	NA	2bb_fi7WSRl_G M2uUkn0D2
3	Muro básico:Partición con capa de yeso	RETAININGWALL	NA	0ZPrHKZdJU8p viJCo5dHBX
4	Muro básico:Partición con capa de yeso	STANDARD	NA	0qjzuCWvUYeHz H_FwdGMBI
5	Muro básico:Partición con capa de yeso	PARTITIONING	NA	0bRIHKs6GI1T9 9BlS\$2yIA

Tabla 3.5.16. Extracto de tipos de muro predefinidos y personalizados

Otras relaciones que pueden establecerse para muros son las definidas por *IfcRelConnectElements* con las que se relaciona la instancia con, por ejemplo, el suelo o losa donde se apoya la cara inferior del muro.

///Relación de muro con suelo

#409= IFCRELCONNECTSELEMENTS('1_c3omXFD7UPfF9thiX',#201,\$,\$,\$,#292,#407);

///Muro que se conecta con el suelo

↘ [#292= IFCSLAB ('20WrgxqpbDtferON_k1P2X',#201,'Slab #1',\$,\$,\$,#352,#302,\$,\$,BASESLAB.)]

///Suelo que se conecta con el muro

↘ [#407= IFCWALL ('0edwXkNgP03AySQjQQlat2',#201,'Wall #3',\$,\$,\$,#416,#423,\$,\$,NOTDEFINED.)]

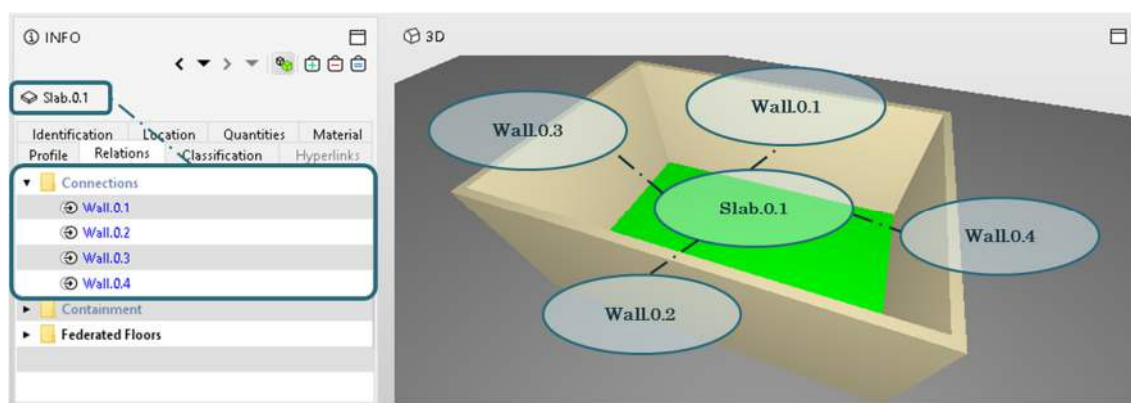


Figura 3.5.39. Relaciones de suelo y muros conectados

Una relación diferente es la que se establece entre muros, identificada con el término *IfcRelConnectsPathElements*, la cual define una relación para cada una de las conexiones que se realizan entre muros dentro de un modelo BIM. Esta unión de muros puede darse en esquina o bien entre un extremo de muro y una parte intermedia de otro muro (unión tipo *T*), pudiendo ser esta ortogonal o inclinada.

///Relación de muros conectados

#432=IFCRELCONNECTSPATHELEMENTS('0N2CJ8VSz1yx6iStYDzqLN',#201,\$,\$,\$,#407,#429);

///Muros conectados por la relación

▼ [#407=IFCWALL('0edwXkNgP03AySQjQQlat2',#201,'Wall #3',\$,\$,\$,#416,#423,\$,..NOTDEFINED.)]

▼ [#429=IFCWALL('27D5wuD5z86hr07xvbZLKW',#201,'Wall #4',\$,\$,\$,#438,#445,\$,..NOTDEFINED.)]

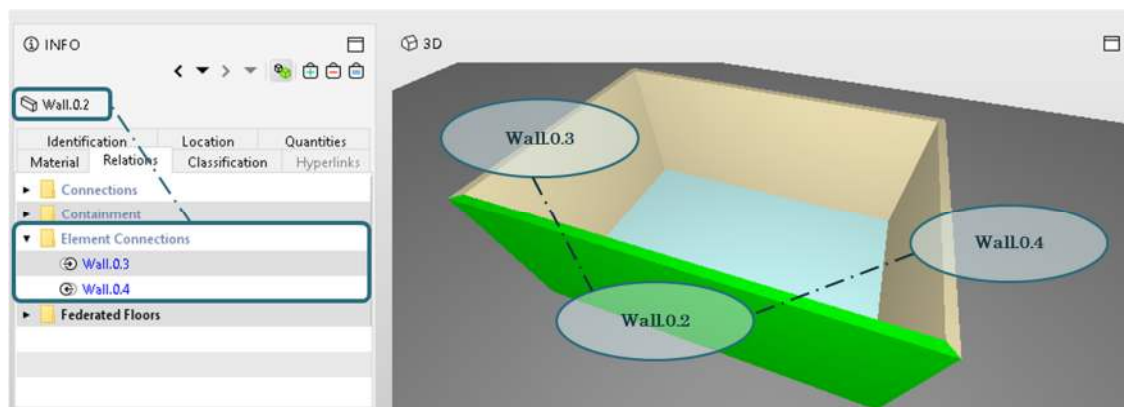


Figura 3.5.40. Relaciones entre muros conectados

Adicionalmente, un muro es capaz de saber el espacio o espacios que divide a través de la relación *IfcRelSpaceBoundary*²¹⁰, la cual establece una conexión entre la superficie del muro (y de otros elementos contenidos o anexados a él) y la cara o caras del volumen del espacio que están en contacto directo con este.

²¹⁰ Esta definición permite que los datos puedan ser utilizados para diferentes tipos de cálculos y análisis como en el caso del artículo *Extended maptree: a representation of fine-grained topology and spatial hierarchy of BIM*, en el que se proponen nuevos modelos de relaciones topológicas dentro del IFC.

Gracias a esta relación entre el espacio y el objeto físico, es posible la obtención de la topología del espacio. Las definiciones que permiten obtener la relación son *RelatingSpace* y *RelatedBuildingElement*. Algunos estudios de los últimos años han utilizado este tipo de relaciones para identificar redes de navegación en el interior de edificios a partir de sistemas de triangulación del espacio y otras estrategias de trabajo geométrico.²¹¹

///Relación de muro con espacio

#340=IFCRELSPACEBOUNDARY('1rn4BqkKpt',#20,'2ndLvl',\$, #119,#304,#176,PHY,..INTERNAL.);

///Espacio relacionado con el muro

↘ [#119=IFCSPACE('2MCxi2ZaDEuxPVzR',#20,'1',\$, \$, #111,#118,'Habitac.ELEMENT,..SPACE.,\$)]

///Muro delimitador del espacio

↘ [#304=IFCWALL('2MCxi2ZaD',#20,'MF3n con capa de yeso',#289,#303,'167375',..NOTDEFINED.)]

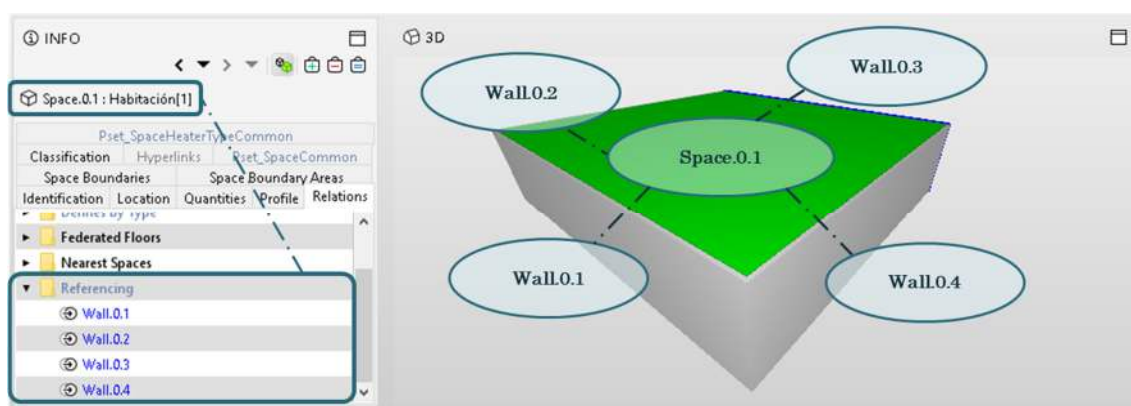


Figura 3.5.41. Relación de espacio con muros perimetrales

En referencia a las capas o sustratos de material que puede contener un muro, también existe una relación específica la cual se define como *IfcRelAssociatesMaterial*. La asociación permite entender el contenido de capas a través de la relación de las instancias de muro con una definición para el conjunto de capas denominada *IfcMaterialLayerSetUsage*, la cual se relaciona con una segunda definición llamada *IfcMaterialLayerSet* la cual contiene cada una de las capas definidas como *IfcMaterialLayer*. En el caso de *IfcMaterialLayerSetUsage*, se define la información del desfase del muro con respecto a la línea de referencia y también el sentido de creación de las capas del muro, pudiendo este ser positivo o negativo.

///Relación de muro con capas

#232=IFCRELASSOCIATESMATERIAL('0ZtRDUvpWpEcUTBNu10Int',#18,\$, \$, (#157),#192);

///Muro delimitador del espacio

#157=IFCWALL('2uZcYbK4n8CAESr_FB5qZg',#18,'MuroLadrillo:162765',\$, 'Muro,..NOTDEFINED.);

///Conjunto de uso de capas

#192=IFCMATERIALLAYERSETUSAGE(#193,..AXIS2,..NEGATIVE.,0.014,\$);

²¹¹ Indoor navigation supported by the Industry Foundation Classes (IFC): A survey. Automation in Construction 121 (2021).

///Conjunto de capas de muro

#193=IFCMATERIALLAYERSET((#188,#189,#190,#191),'MuroLadrillo',\\$);

///Capas de muro

#188=IFCMATERIALLAYER(#158,0.014999999999999999,\\$, 'Mortero beis',\\$, 'Varios',\\$);

#189=IFCMATERIALLAYER(#162,0.12,\\$, 'Ladrillo',\\$, 'Enladrillado',\\$);

#190=IFCMATERIALLAYER(#174,0.089999999999999997,\\$, 'Aislamiento t\X\E9rmico',\\$, 'Varios',\\$);

#191=IFCMATERIALLAYER(#181,0.014999999999999999,\\$, 'Yeso',\\$, 'Placa de yeso laminado',\\$);

A su vez, la capa de material indica en su primer atributo el material con el que fue definida. El segundo define el espesor de la capa. En el siguiente ejemplo se muestra en metros dentro del archivo IFC, mientras que, en el visor, la configuración está definida para que cada capa se muestre con unidades en milímetros.

///Capa de material

#191=IFCMATERIALLAYER(#181,0.014999999999999999,\\$, 'Mortero beis',\\$, 'Varios',\\$);

///Material asociado a la capa

#181=IFCMATERIAL('Mortero beis',\\$, 'Varios');

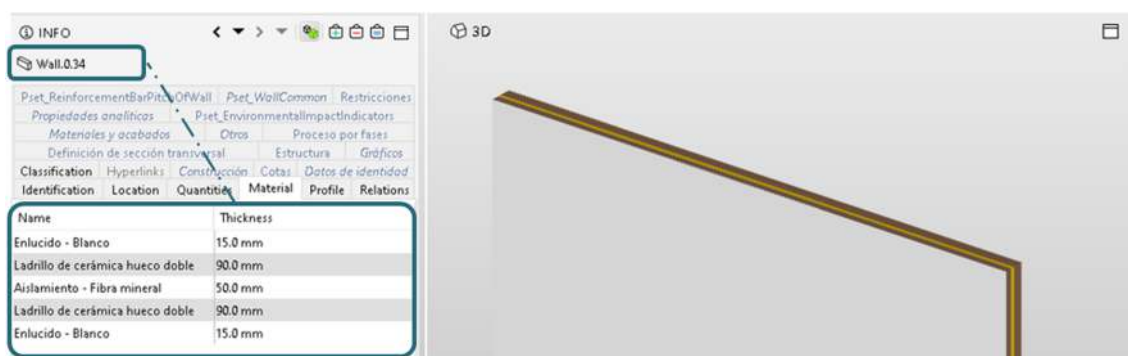


Figura 3.5.42. Composición de materiales de muro

En cuanto a los atributos de las capas, cada una de ellas cuenta con un nombre, un material específico, un espesor y también otros datos relativos a si es ventilada o no, la categoría del material, la prioridad de la capa para saber qué capas conectan antes en los encuentros de muro o cuales son cortadas por otros elementos como pilares, y una descripción propia de la capa.

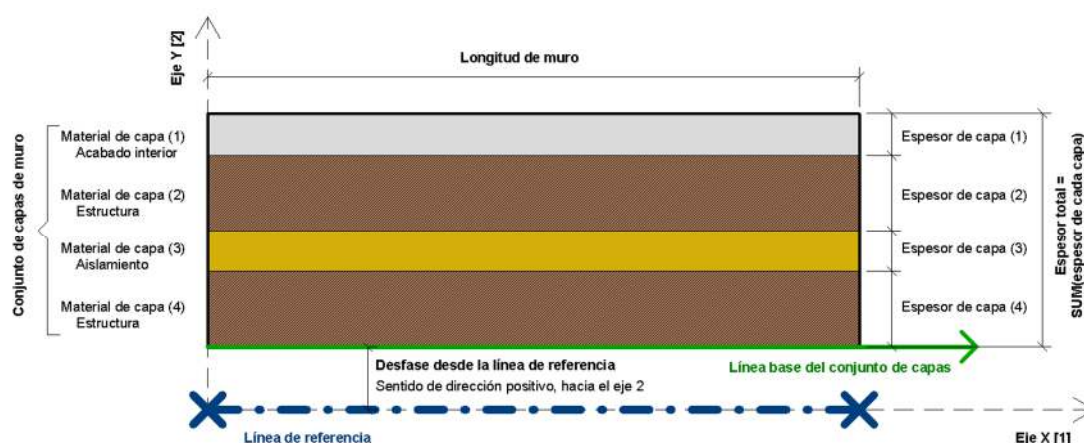


Figura 3.5.43. Estructura de capas de muros IFC

La asignación del conjunto de capas puede especificarse tanto para el *IfcWall* como para el *IfcWallType*. En el siguiente esquema del IFC se muestra la relación de una instancia de muro con el conjunto de capas de material, pudiendo observar que un muro puede contener diferentes asociaciones de material y que un conjunto de capas también puede estar relacionado con varias instancias o tipos de muro.

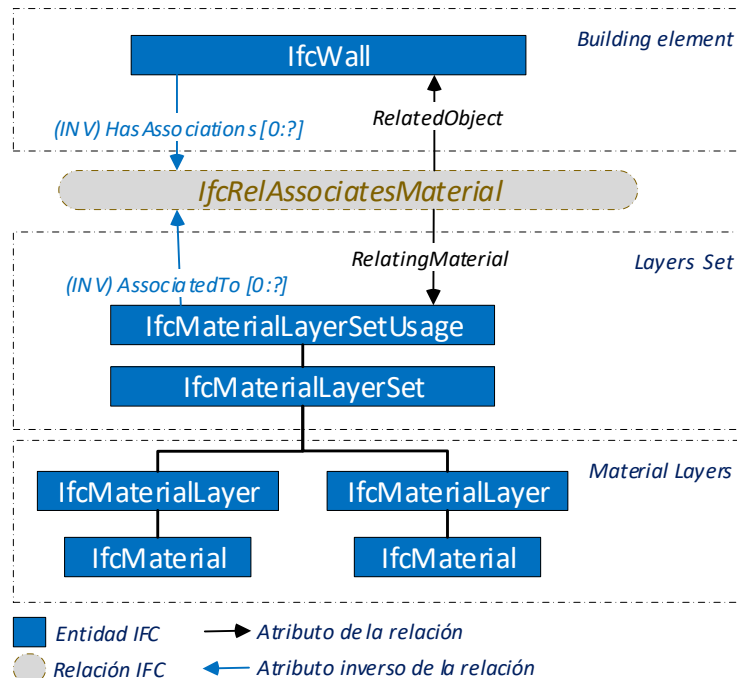


Figura 3.5.44. Relación de muro con capas. Fuente* (ISO 16739)

Además, es posible obtener la información de todas las capas de un tipo de muro, identificando los nombres, espesores y categorías de cada una de las capas de material. Para ello, es necesario identificar la relación de tipo *IfcRelAssociatesMaterial* del tipo de muro.

```

model = ifcopenshell.open('Muro_Capas.ifc')
mtrInfo = []
wllTp = model.by_type('IfcWallType')[0]
assWll = wllTp.HasAssociations
for a in assWll:
    if a.is_a() == 'IfcRelAssociatesMaterial':
        for d in a.RelatingMaterial.MaterialLayers:
            info = d.Material.Name, d.LayerThickness, d.Category, wllTp.Name,
            wllTp.GlobalId
            mtrInfo.append(info)
df4 = pd.DataFrame(mtrInfo)
df4.columns = ['Nombre Material', 'Espesor', 'Categoría',
'Tipo de muro', 'GUID Tipo de Muro']
display(df4)

```

Id	Nombre Material	Espesor	Categoría	Tipo de muro	GUID Tipo de Muro
0	Mortero beis	0.015	Varios	Muro básico:Ladrillo enfoscado - trasd. PYL - ...	0wQknZQiXEOBcv2M5Gzx_v
1	Ladrillo	0.12	Enladrillado	Muro básico:Ladrillo enfoscado - trasd. PYL - ...	0wQknZQiXEOBcv2M5Gzx_v
2	Aislamiento térmico	0.09	Varios	Muro básico:Ladrillo enfoscado - trasd. PYL - ...	0wQknZQiXEOBcv2M5Gzx_v
3	Yeso	0.015	Placa de yeso laminado	Muro básico:Ladrillo enfoscado - trasd. PYL - ...	0wQknZQiXEOBcv2M5Gzx_v

Tabla 3.5.17. Extracto de materiales en capas de muro

Las definiciones de estilos se encuentran en *IfcMaterialDefinitionRepresentation*, el cual establece una relación entre las propiedades de representación del estilo y el propio material. Este concepto es aplicable a materiales generales de un objeto y también a capas de objeto.

///Relación de representación del material

#161=IFCMATERIALDEFINITIONREPRESENTATION(\$,\$, (#160),#158);

///Material

#158=IFCMATERIAL('Mortero beis',\$, 'Varios');

///Representación del estilo del material

#160=IFCSTYLEDREPRESENTATION(#99,'Style','Material', (#159));

#159=IFCSTYLEDITEM(\$, (#153), \$);

#153=IFCSURFACESTYLE('Mortero beis',.BOTH., (#152));

#152=IFCSURFACESTYLERENDERING(#151,0.,\$, \$, \$, \$, IFCNORMALISEDRAIOMEASURE(0.5));

#151=IFCCOLOURRGB(\$,0.8901960784313725,0.84313725490196079,0.6705882352941176);

Por otro lado, la propia instancia de muro tiene sus propiedades de forma, representación geométrica, información de presentación asociada y definiciones de representación topológica como vértices, aristas y caras. Todo ello, se engloba en *IfcProductDefinitionShape*. A continuación, se muestra un ejemplo de una instancia de muro:

///Instancia de muro

#157=IFCWALL('2uZcYbK4n8CAESr_FB5qZg',#18,'MuroLadrillo:162765',\$, #156,.NOTDEFINED.);

///Definición de forma del muro

#156=IFCPRODUCTDEFINITIONSHAPE(\$,\$, (#145,#155));

///Curva 2D con los datos de la geometría en planta del muro

#145=IFCSHAPEREPRESENTATION(#100,'Axis','Curve2D', (#144));

#144=IFCPOLYLINE((#4,#143));

#143=IFCCARTESIANPOINT((8.400000000000004,0.));

///Extrusión de sólido de muro rectangular

#155=IFCSHAPEREPRESENTATION(#101,'Body','SweptSolid', (#150));


```
#150=IFCEXTRUDEDAREASOLID(#148,#149,#9,2.9999999999999996);
#148=IFCRECTANGLEPROFILEDEF(,AREA,,$,#147,8.400000000,0.24000000174);
#149=IFCAXIS2PLACEMENT3D(#3,$,$);
#147=IFCAXIS2PLACEMENT2D(#146,#12);
#146=IFCCARTESIANPOINT((4.2000000000000002,0.));
```

Además de la definición de la forma de la instancia, cada ejemplar de muro o de cualquier tipo de objeto, debe posicionarse en el espacio tridimensional a partir de la definición *IfcLocalPlacement*, ya sea estableciendo una relación con un punto base o de referencia, una relación de un subcomponente con su componente principal, o bien a través de sus coordenadas absolutas.

```
///Instancia de muro
#157=IFCWALL('2uZcYbK4n8CAESr_FB5qZg',#18,'MuroLadrillo:162765',$, #142,.NOTDEFINED.);
///Definición del posicionamiento
#142=IFCLOCALPLACEMENT(#111,#141);
///Posicionamiento del punto de coordenadas
↘ [#111=IFCLOCALPLACEMENT(#107,#110); ]
    ↘ [#110=IFCAXIS2PLACEMENT3D(#3,$,$); ]
    ↘ [#107=IFCLOCALPLACEMENT(#118,#106); ]
        ↘ [#106=IFCAXIS2PLACEMENT3D(#3,$,$); ]
        ↘ [#118=IFCLOCALPLACEMENT($,#117); ]
            ↘ [#117=IFCAXIS2PLACEMENT3D(#3,$,$); ]
                ↘ [#3=IFCCARTESIANPOINT((0.,0.,0.)); ]
///Posicionamiento relativo del muro
↘ [#141=IFCAXIS2PLACEMENT3D(#140,#9,#7); ]
    ↘ [#140=IFCCARTESIANPOINT((-10.83296250603,0.522507486864,0.)); ]
```

Como ocurre con el resto de instancias que se modelan en un archivo IFC, los objetos heredan la información relativa a la persona responsable de generar el archivo, su respectiva organización y, por otro lado, la aplicación con la que se ha generado el archivo y su organización. Todo ello está registrado en el historial de propiedad, dentro de la definición *IfcOwnerHistory*.

```
///Registro de propiedad
#18=IFCOWNERHISTORY(#17,#2,$,.NOCHANGE,,$,$,1711911020);
///Autoría de modelo
↘ [#17=IFCPERSONANDORGANIZATION(#15,#16,$); ]
    ↘ [#15=IFCPERSON($,','samuel.jesus.triana.gonzalez',$); ]
    ↘ [#16=IFCORGANIZATION($,'TesisDoctoral',,$,$); ]
///Aplicación utilizada
↘ [#2=IFCAPPLICATION(#1,'2024','Autodesk Revit(ESP)',Revit); ]
    ↘ [#1=IFCORGANIZATION($,'Autodesk Revit 2024 (ESP)',,$,$); ]
```

Un muro puede descomponerse en partes como en un marco con la función de parte portante y paneles a cada lado. La relación que permite conectar el muro con sus subelementos es *IfcRelAggregates*. Estos objetos agregados pueden tener diferentes clases IFC como *IfcMember* (postes) *IfcBuildingElementPart* (paneles de yeso o similares), *IfcElementAssembly* (armazones de particiones interiores) e *IfcMechanicalFastener* (cierres o conexiones estructurales).

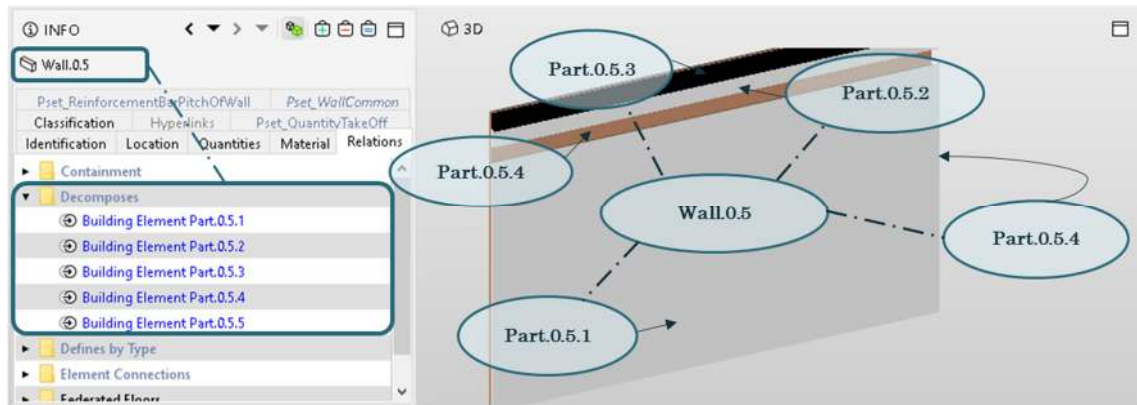


Figura 3.5.45. Descomposición de subelementos de muro

///Relación del muro con sus partes

```
#314=IFCRELAGGREGATES('0TM5igWk',#18,$,$,#125,(#144,#188,#221,#247));
```

///Instancia de muro

```
↘ [#125=IFCWALL('0TM5iIWk',#18,'Montaje:Wll:167931',.ELEMENTEDWALL);]
```

///Subelementos de muro

```
↘ [#144=IFCBUILDINGELEMENTPART('1txeC657B',#18,'MANSORY',#129,#143);]
```

```
↘ [#188=IFCBUILDINGELEMENTPART('1txeC65OR',#18,'CLADDING',#173,#187);]
```

```
↘ [#221=IFCBUILDINGELEMENTPART('1t65O2',#18,'INSULATION',#206,#220);]
```

```
↘ [#247=IFCBUILDINGELEMENTPART('1t65OF',#18,'MANSORY',#235,#246);]
```

La definición *Get_parts* de *IfcOpenShell* permite extraer las subelementos contenidos dentro de una instancia de muro. De igual forma, es posible obtener los subelementos del muro si se accede a la relación de agregados a través de *IsDecomposedBy* y luego se buscan los *RelatedObjects*.

Para diferenciar cada una de las partes que componen el muro, se puede definir el tipo personalizado de objeto en cada caso y de esa manera, tener los subcomponentes bien diferenciados.

```
model2 = ifcopenshell.open('Muro_Montaje.ifc')
comwall = model2.by_type('IfcWall')[0].IsDecomposedBy[0].RelatedObjects
atribos = []
for i in comwall:
    at = (i.is_a(), i.Name, i.GlobalId, i.PredefinedType, i.ObjectType)
    atribos.append(at)
df6 = pd.DataFrame(atribos)
df6.columns = ['IfcClass', 'Nombre', 'GUID', 'Tipo predefinido', 'Tipo personalizado']
display(df6)
```

Id	IfcClass	Nombre	GUID	Tipo predefinido	Tipo personalizado
0	IfcBuildingElementPart	Muro básico:Ladrillo hueco - 12,5 cm:162778	1txeOCClZ8T5\$MnC657	USERDEFINED	MANSORY
1	IfcBuildingElementPart	Muro básico:Acabado de yeso - 3,5 cm:162826	1txeOCClZ8T5\$MnC650	USERDEFINED	CLADDING
2	IfcBuildingElementPart	Muro básico:Aislamiento térmico - 5 cm:162835	1txeOCClZ8T5\$MnC650	USERDEFINED	INSULATION
3	IfcBuildingElementPart	Muro básico:Ladrillo hueco - 8 cm:162846	1txeOCClZ8T5\$MnC650	USERDEFINED	MANSORY
4	IfcBuildingElementPart	Muro básico:Acabado de yeso - 3,5 cm:162855	1txeOCClZ8T5\$MnC650	USERDEFINED	CLADDING
5	IfcBuildingElementPart	Barrido de muro:Rodapié:167753	0Dpr5s6MT1t0bBkNcfMQ	USERDEFINED	SKIRTING BOARD
6	IfcBuildingElementPart	Barrido de muro:Cornisa:167755	0Dpr5s6MT1t0bBkNcfMO	USERDEFINED	MOLDING

Tabla 3.5.18. Extracto de subelementos de muros

Generalmente, los muros se modelan como multicapa y la exportación IFC mantiene estos muros multicapa, incluidos los acabados. Sin embargo, en algunos casos es útil tener las capas por separado para tener un mayor control de las mediciones. Por ejemplo, un *IfcWall* puede dividirse en la capa portante del muro y el resto de capas y subcomponentes pueden ser otras entidades como *IfcCovering* con diferentes tipos predefinidos. En el caso de acabados de muro, siempre se tratan de capas que no soportan cargas y puede posicionarse en el exterior o en el interior del muro. Si el proceso de creación del muro, se realiza a partir de esta segunda opción, cada objeto será independiente y habrá que definirle a cada entidad los tipos y clases que correspondan.

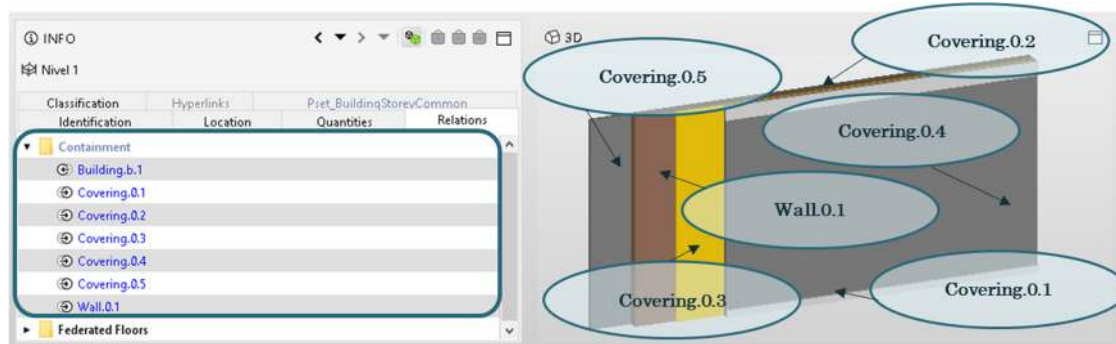


Figura 3.5.46. Relación del muro con sus capas IFC

Para obtener los valores de los diferentes elementos, habrá que ir extrayendo la información de cada clase de objeto, así como de sus atributos.

```

model2 = ifcopenshell.open('Muro_Subelementos.ifc')
clss = ['IfcWall', 'IfcCovering']
atrib2 = []
for c in clss:
    elem = model2.by_type(c)
    for e in elem:
        at = (e.is_a(), e.PredefinedType, e.Name, e.GlobalId)
        atrib2.append(at)
df7 = pd.DataFrame(atrib2)
df7.columns = ['IfcClass', 'Tipo predefinido', 'Nombre', 'GUID']
display(df7)

```

Id	IfcClass	Tipo predefinido	Nombre	GUID
0	IfcWall	STANDARD	Muro básico: Ladrillo hueco - 12,5 cm:168090	3BF68LjHH5Jgpg133qz3Tm
1	IfcCovering	CLADDING	Muro básico: Acabado de yeso - 3,5 cm:168091	3BF68LjHH5Jgpg133qz3Tn
2	IfcCovering	INSULATION	Muro básico: Aislamiento térmico - 5 cm:168092	3BF68LjHH5Jgpg133qz3Ts
3	IfcCovering	CLADDING	Muro básico: Acabado de yeso - 3,5 cm:169710	0Sej0zjvf72AM15Ck1Pteh
4	IfcCovering	MOLDING	Barrido de muro: Cornisa:169724	0Sej0zjvf72AM15Ck1Ptev
5	IfcCovering	SKIRTINGBOARD	Barrido de muro: Rodapié:169725	0Sej0zjvf72AM15Ck1Pteu

Tabla 3.5.19. Extracto de acabados y molduras de muro

Como cualquier entidad IFC, los muros contienen una serie de atributos, algunos heredados y otros específicos de la clase, que permiten obtener los datos generales del objeto BIM. Estos atributos se almacenan en la definición IFC con nombres en inglés, en el caso de ser definiciones del esquema.



Figura 3.5.47. Atributos principales de identificación de muro

En el siguiente ejemplo se obtienen los atributos generales con sus respectivos valores de una instancia de muro IFC:

```
instanciamuro = model.by_type("IfcWall")[0]
infodemuro = instanciamuro.get_info().items()
df55 = pd.DataFrame(infodemuro)
df55.columns = ['Nombre de atributo', 'Valor']
display(df55)
```

Id	Nombre de atributo	Valor
0	id	462
1	type	IfcWall
2	GlobalId	3u10zYbZT1pQ0ayM5A7QsV
3	OwnerHistory	[[None, , samuel.jesus.triana.gonzalez, None,...
4	Name	Muro básico:Partición con capa de yeso:167382
5	Description	None
6	ObjectType	Muro básico:Partición con capa de yeso
7	ObjectPlacement	[[[#147=IfcLocalPlacement(\$,#146), #31=IfcAxis...
8	Representation	[None, None, ([#114=IfcGeometricRepresentation...
9	Tag	167382

Tabla 3.5.20. Extracto de atributos de instancia de muro

También es posible realizar la búsqueda inversa en la que, por ejemplo, se tiene el GlobalId (*GUID*) de una instancia y se quiere obtener la información de dicho objeto. En estos casos, se utiliza la definición *by_guid*, incluyendo el código de la instancia.

```
model = ifcopenshell.open('Relaciones_Muros.ifc')
wall = model.by_guid('1RIApVCs9FrfdqXiYtYX2E')
def obtener_atributos_basicos(entidad):
    atributos = {
        "GlobalId": entidad.GlobalId,
        "Name": entidad.Name,
        "ObjectType": getattr(entidad, 'ObjectType', None),
        "PredefinedType": getattr(entidad, 'PredefinedType', None),
        "Tag": getattr(entidad, 'Tag', None),
    }
    return atributos
datos = obtener_atributos_basicos(wall)
df = pd.DataFrame([datos])
print(df)
```


GlobalId	Name	ObjectType	PredefinedType	Tag
1RlApVCs9FrfdqXiYtYX2E	Muro básico:Partición con capa de yeso:169952	None	SOLIDWALL	169952

Tabla 3.5.21. Extracción de atributos desde GUID

Además de los atributos generales, existen también propiedades que definen otras características del objeto como su clasificación, su ubicación global, representación geométrica... Todas ellos están siempre agrupadas a partir de conjuntos, pudiendo estos ser de cantidades (*Qtos*) o de propiedades (*Psets*).

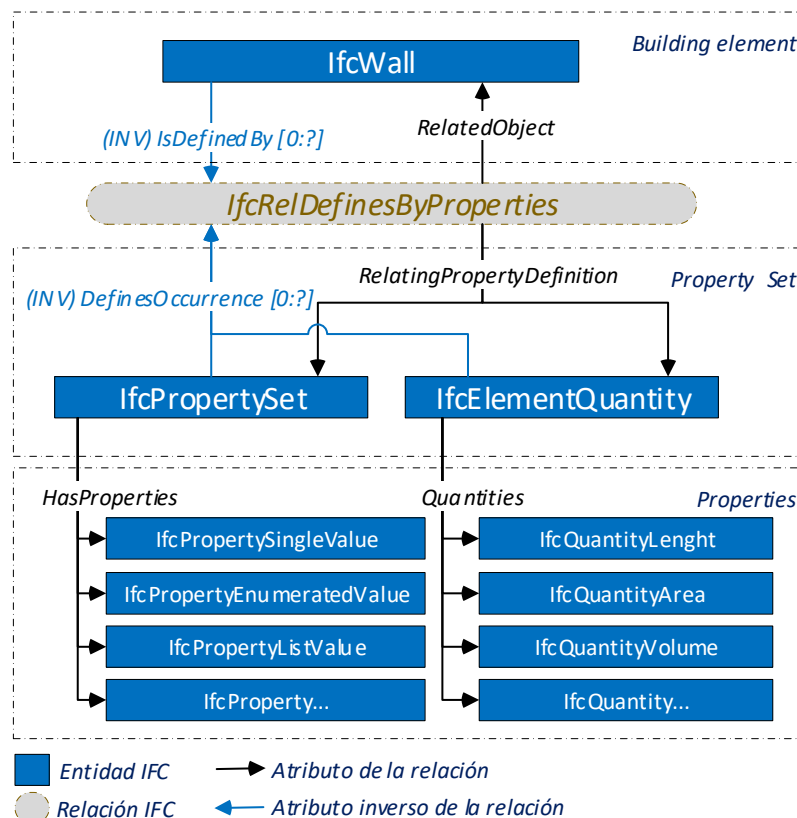


Figura 3.5.48. Relación de propiedades y cantidades. Fuente* (ISO 16739)

En el caso de los conjuntos de cantidades, a partir de la versión IFC 4 se añade un tipo por encima de *IfcElementQuantity*, denominado *IfcQuantitySet*.

Para obtener a través de *IfcOpenShell* las cantidades de un muro, se puede utilizar la definición *Get_psets*, indicando a continuación la instancia de muro y el tipo de propiedad de cantidades.

```

murosmodelo = model.by_type("IfcWall")
muromod1 = murosmodelo[0]
cantidades = ifcopenshell.util.element.get_psets(muromod1, qtos_only=True)
qtodata = []
for qto_name, qto_data in cantidades.items():

```

```

for prop_name, prop_value in qto_data.items():
    qtodata.append({
        "Wall GlobalId": muromod1.GlobalId,
        "Quantity Set": qto_name,
        "Property": prop_name,
        "Value": prop_value
    })
qtodf1 = pd.DataFrame(qtodata)
display(qtodf1)

```

Id	Wall GlobalId	Quantity Set	Property	Value
0	3lLx0gNe59vvExhby0Bfew	Qto_WallBaseQuantities	Height	6498.67729
1	3lLx0gNe59vvExhby0Bfew	Qto_WallBaseQuantities	Length	6404
2	3lLx0gNe59vvExhby0Bfew	Qto_WallBaseQuantities	Width	202
3	3lLx0gNe59vvExhby0Bfew	Qto_WallBaseQuantities	GrossFootprintArea	1.293608
4	3lLx0gNe59vvExhby0Bfew	Qto_WallBaseQuantities	GrossSideArea	22.35499
5	3lLx0gNe59vvExhby0Bfew	Qto_WallBaseQuantities	GrossVolume	4.515708

Tabla 3.5.22. Cantidades de instancia de muro IFC4

Dependiendo del esquema IFC utilizado y la definición de los conjuntos de cantidades, también es posible extraer los datos para la versión IFC 2x3, en este caso, transponiendo los datos para que cada cantidad sea una columna:

```

qto2x3 = ifcopenshell.util.element.get_pset(muromod1, 'Qto_WallBaseQuantities',
prop=None, should_inherit=True)
dfqto2x3 = pd.DataFrame(list(qto2x3.items()), columns=["Property", "Value"]).T
display(dfqto2x3)

```

0	1	2	3	4	5	6
Property	Height	Length	Width	GrossFootprintArea	GrossSideArea	GrossVolume
Value	6498.6	6404	202	1.293608	22.35499	4.515708

Tabla 3.5.23. Cantidades de instancia de muro IFC2x3

Una vez que se obtienen los valores una instancia de muro, se puede realizar el mismo proceso dentro de un bucle para extraer la información de todas las instancias de muros del proyecto y generar un listado con la información de cada uno de ellos. Para ello, es necesario tener en cuenta que en la versión de estudio pueden existir tanto *IfcWall* como *IfcWallStandardCase*.

```

muros = model.by_type("IfcWall") + model.by_type("IfcWallStandardCase")
rows = []
for wall in muros:
    qto = ifcopenshell.util.element.get_pset(wall, 'Qto_WallBaseQuantities',
    should_inherit=True)
    if qto:
        cleaned_qto = {k: getattr(v, 'wrappedValue', v) for k, v in qto.items()}
        cleaned_qto["GlobalId"] = wall.GlobalId
        rows.append(cleaned_qto)
qtowllsdf = pd.DataFrame(rows)
cols = ["GlobalId"] + [c for c in qtowllsdf.columns if c != "GlobalId"]
qtowllsdf = qtowllsdf[cols]
display(qtowllsdf)

```

Id	GlobalId	NetSide Area	Net Volume	Nominal Width	Nominal Height	Nominal Length	Tipo
0	38NbIWsdL1I8DljLvn67bV	43.2	8.094	202	3300	19702	SIP 202mm Wall - conc clad
1	38NbIWsdL1I8DljLvn67cw	43.2957	8.15	202	3500	19702	SIP 202mm Wall - conc clad
2	0Intrd6l1AGwfcnKJK5r3q	40.014	4.468	120	3000	11380	Interior - Particion
3	0NWseyvsH7_gBW225aGtuD	24.781	6.801	280	3500	9628.063	CL_W1
4	2OulbipBrAkwQNv_KJRV9c	42.7518	8.635	202	3900	15202	Wall - Timber Clad

Tabla 3.5.24. Extracto de cantidades de muros IFC2x3

Por otro lado, si lo que se busca es obtener las propiedades del muro, se pueden obtener sus *Psets* y seguidamente las propiedades que contiene utilizando la definición de *HasProperties* y buscando los nombres y sus valores.

```

wllprop = []
wall = model.by_type('IfcWall')[0]
pstwll = []
for definition in wall.IsDefinedBy:
    if definition.is_a('IfcRelDefinesByProperties'):
        property_set = definition.RelatingPropertyDefinition
        pstwll.append(property_set)
for property in pstwll[6].HasProperties:
    bb = aaa[6].Name, property.Name,
    property.NominalValue.wrappedValue
    wllprop.append(bb)
propdf = pd.DataFrame(wllprop)
propdf.columns = ['Pset', 'Nombre de atributo', 'Valor']
display(propdf)

```

Id	Pset	Nombre de atributo	Valor
0	Pset_WallCommon	LoadBearing	False
1	Pset_WallCommon	Reference	SIP 202mm Wall - conc clad
2	Pset_WallCommon	ExtendToStructure	False
3	Pset_WallCommon	IsExternal	True
4	Pset_WallCommon	ThermalTransmittance	47.545455

Tabla 3.5.25. Propiedades del pset común de muro IFC2x3

Para la extracción de la totalidad de *psets* y propiedades contenidas en un objeto BIM, puede usarse la definición de *get_psets* con la que se obtienen los nombres de cada conjunto, las propiedades contenidas en cada uno y también sus respectivos valores:

```

wall01 = model.by_type("IfcWall")[0]
wall_type1 = ifcopenshell.util.element.get_type(wall01)
print(f"Tipo del muro: {wall_type1.Name}")
psets = ifcopenshell.util.element.get_psets(wall_type1)
rows = []
for pset_name, properties in psets.items():
    for prop_name, prop_value in properties.items():
        # Obtener el valor real
        value = getattr(prop_value, 'wrappedValue', prop_value)
        rows.append({
            "Property Set": pset_name,
            "Property": prop_name,
            "Value": value})
dfpp = pd.DataFrame(rows)
display(dfpp)

```

Id	Property Set	Property	Value
0	Construcción	Anchura	202
1	Construcción	Función	Exterior
2	Datos de identidad	Marca de tipo	WT1
3	Datos de identidad	Nota clave	F10
4	Materiales y acabados	Material estructural	Timber Insulated Panel
5	Otros	Categoría	Muros
6	Otros	Nombre de familia	Muro básico
7	Propiedades analíticas	Masa térmica	33244.2
8	Propiedades analíticas	Resistencia térmica (R)	0.021033

Tabla 3.5.26. Extracto de propiedades de muro IFC2x3

Un estudio interesante es la obtención del recuento de propiedades que contiene cada pset aplicado a todas las instancias de muro existentes en el proyecto. Con ello se consigue detectar rápidamente si existen ejemplares que no tienen definidas todas las propiedades o bien que sus valores no han sido rellenados.

```
walls = model.by_type("IfcWall") + model.by_type("IfcWallStandardCase")
all_pset_names = set()
wall_data = []
for wall in walls:
    psets = ifcopenshell.util.element.get_psets(wall)
    if not psets:
        continue
    summary = {"GlobalId": wall.GlobalId}
    for pset_name, props in psets.items():
        all_pset_names.add(pset_name)
        summary[f"{pset_name}_Count"] = len(props)
    wall_data.append(summary)
df = pd.DataFrame(wall_data)
for pset in all_pset_names:
    col = f"{pset}_Count"
    if col not in df.columns:
        df[col] = 0
cols = ["GlobalId"] + sorted([col for col in df.columns if col != "GlobalId"])
df = df[cols]
display(df)
```

Id	GlobalId	Base Quantities	Cotas	Datos de identidad	Estructura	Prop. Analít.	Pset_Wall Common
0	38Nb1WsDL1 I8DljLvn67bV	5	4	7	4	6	6
1	38Nb1WsDL1 I8DljLvn67cw	5	4	7	4	6	6
2	0Intrd6l1AG wfcnKJK5r3q	5	4	4	4	3	5
3	0NWseyvsH7_ gBW225aGtuD	5	4	6	7	6	6
4	2OulbipBrAkw QNv_KJRV	5	4	6	4	3	5

Tabla 3.5.27. Extracto de recuento de propiedades de muros

Un muro puede asignarse a diferentes sistemas de clasificación dependiendo de su función o de otros aspectos constructivos. Para cada uno de ellos, por un lado, se encuentra la definición de *IfcClassification*, que establece el nombre y los datos generales de la clasificación y, por otro, el *IfcClassificationReference*, que contiene los datos específicos de la referencia. En cualquier caso, siempre se establece una

relación entre el objeto y la clasificación a través de *IfcRelAssociatesClassification*. Dado que se trata de una asociación, a la hora de iterar para extraer los datos del muro, es necesario filtrar cada una de las asociaciones enlazadas al muro para solo obtener las definidas para clasificaciones.

///Instancia de muro

```
#138 = IFCWALLSTANDARDCASE('00GgiaRG, #18, 'ME1sico, $, #122, #137, '162855');
```

///Relaciones con clasificación OmniClass

```
#184 = IFCRELAASSOCIATESCLASSIFICATION('3e$p_2oe145xWw7R', #18, $, $, (#138), #183);
↳ [#183 = IFCCCLASSIFICATIONREFERENCE('21-02 20 10 20', 'Exterior Wall Construction', #177);]
  ↳ [#177 = IFCCCLASSIFICATION('http://www.omniclass.org/', '(OCCS)', #178, 'OmniClass');]
```

///Relaciones con clasificación MasterFormat

```
#186 = IFCRELAASSOCIATESCLASSIFICATION('2ULdP2vkP5HhSvYv', #18, $, $, (#138), #185);
↳ [#185 = IFCCCLASSIFICATIONREFERENCE('03 45 13', 'Faced Architectural Precast', #179);]
  ↳ [#179 = IFCCCLASSIFICATION('www.csinet.org', '2016 Edition', #180, 'MasterFormat');]
```

///Relaciones con clasificación UniFormat

```
#188 = IFCRELAASSOCIATESCLASSIFICATION('1oKuDMRUb68xrflv', #18, $, $, (#138), #187);
↳ [#187 = IFCCCLASSIFICATIONREFERENCE('B2011', 'Exterior Wall Construction', #181);]
  ↳ [#181 = IFCCCLASSIFICATION('www.uniformat.com/', 'II (E1557-97)', #182, 'UniFormat');]
```

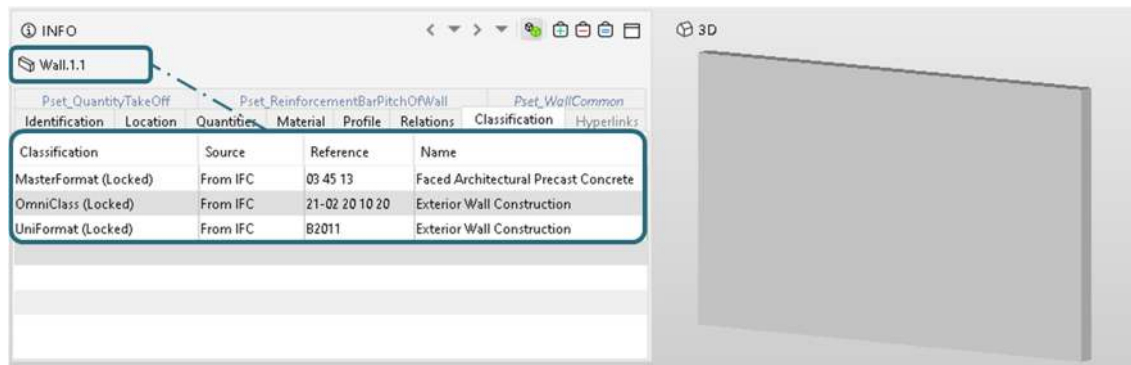


Figura 3.5.49. Definición de múltiples sistemas de clasificación

```
wlls = model4.by_type('IfcWall')
classification_name_list = []
classification_list = []
for product in wlls:
    for has_associations in product.HasAssociations:
        if has_associations.is_a('IfcRelAssociatesClassification'):
            if has_associations.RelatingClassification.Name is not None:
                classification_list.append([
                    str(has_associations.RelatingClassification.ReferencedSource.Name),
                    str(has_associations.RelatingClassification.ReferencedSource.Source),
                    str(has_associations.RelatingClassification.ReferencedSource.Edition),
                    str(has_associations.RelatingClassification.ItemReference),
                    str(has_associations.RelatingClassification.Name)])
dfclass = pd.DataFrame(classification_list)
dfclass.columns = ['Nombre de clasificación', 'Código de clasificación', 'Edición', 'Fuente', 'Descripción de clasificación']
display(dfclass)
```

Id	Nombre de clasificación	Código de clasificación	Edición	Fuente	Descripción clasificación
0	OmniClass	www.omniclass.org/	(OCCS)	21-02 20 10 20	Exterior Wall Construction
1	MasterFormat	www.csinet.org/masterformat	2016 Edition	03 45 13	Faced Architectural Precast Concrete
2	UniFormat	www.uniformat.com/	II (E1557-97)	B2011	Exterior Wall Construction

Tabla 3.5.28. Clasificaciones definidas para instancia de muro

3.5.8.2. Muros cortina

Los muros cortina, denominados dentro del esquema IFC como *IfcCurtainWall*, son diferentes al resto de los muros ya que están formados por subelementos como montantes o paneles. Se trata de agregados que mantienen una relación con el muro cortina del tipo *IfcRelAggregates*.

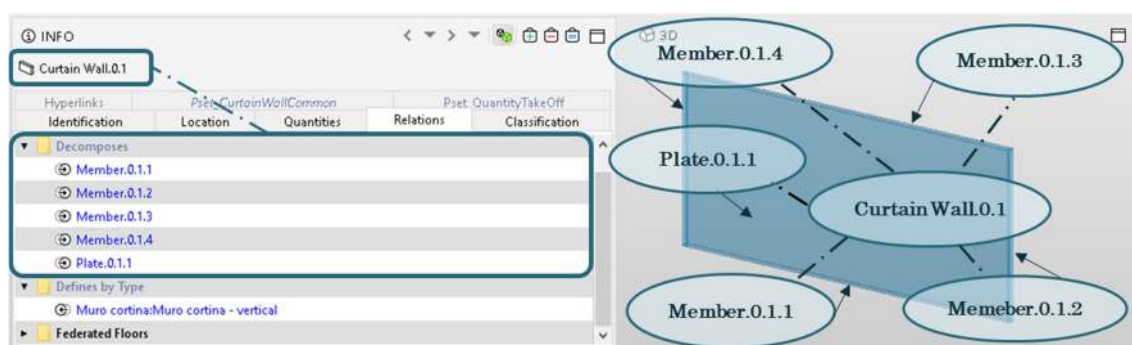


Figura 3.5.50. Descomposición de subelementos de muro cortina

Atendiendo a la descomposición de este tipo de muros, se pueden obtener los subelementos que contiene el muro cortina y sus atributos correspondientes.

```

model = ifcopenshell.open('Relaciones_Muro_Cortina.ifc')
curtnwlls = model.by_type('IfcCurtainWall')
prtInfo = []
for d in curtnwlls[0].IsDecomposedBy:
    if d.is_a() == 'IfcRelAggregates':
        for aa in d.RelatedObjects:
            bb = aa.is_a(), aa.Name, aa.GlobalId, curtnwlls[0].Name
            prtInfo.append(bb)
df = pd.DataFrame(prtInfo)
df.columns = ['Clase IFC', 'Nombre de subcomponente', 'GUID', 'Muro cortina']
display(df)

```

Id	Clase IFC	Nombre de subcomponente	GUID	Muro cortina
0	IfcPlate	Panel de sistema: Acristalado...	0CwwTjvHL6\$A Msolw0uHSI	Muro cortina:Muro cortina - simple:167453
1	IfcMember	Montante angular cuadrilátero: Montante cuadril...	0CwwTjvHL6\$A Msolw0uHV5	Muro cortina:Muro cortina - simple:167453
2	IfcMember	Montante angular cuadrilátero: Montante cuadril...	0CwwTjvHL6\$A Msolw0uHV6	Muro cortina:Muro cortina - simple:167453
3	IfcMember	Montante angular cuadrilátero: Montante cuadril...	0CwwTjvHL6\$A Msolw0uHV7	Muro cortina:Muro cortina - simple:167453

Tabla 3.5.29. Subelementos que componen un muro cortina

Otros elementos que pueden estar contenidos en un muro cortina son las puertas y las ventanas, pudiendo ser simples o bien estar compuestas por subelementos como se muestra en la siguiente figura:

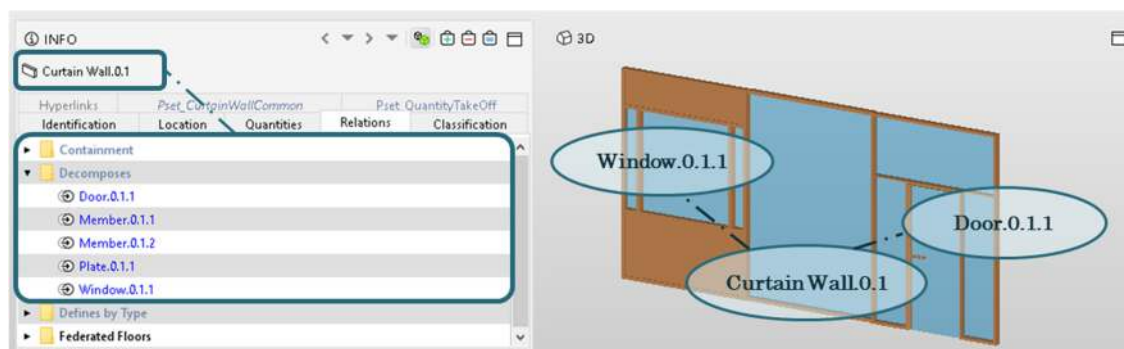


Figura 3.5.51. Puertas y ventanas en muro cortina

Para obtener una tabla que solo contengan puertas y ventanas incluidas dentro de un muro cortina, se define en el código un IF que solo extraiga las clases de IfcDoor e IfcWindow.

```

for cw in curtain_walls:
    for rel in cw.IsDecomposedBy:
        for obj in rel.RelatedObjects:
            if obj.is_a("IfcDoor") or obj.is_a("IfcWindow"):
                elements.append({"GUID": obj.GlobalId,
                                "Clase IFC": obj.is_a(), "Nombre": obj.Name})
display(pd.DataFrame(elements))

```

Id	GUID	Clase IFC	Nombre
0	0CwwTjvHL6\$AMsolw0uHSI	IfcDoor	Panel con puerta simple en vidrio (1):Puerta 1...
1	3tRt8B2sX62xVN3tytOoum	IfcWindow	Panel con ventana:Panel con ventana:168012

Tabla 3.5.30. Subelementos que componen un muro cortina

3.5.8.3. Huecos, puertas y ventanas

Un muro puede relacionarse también con otros elementos como puede ser una instancia de hueco o abertura, de la clase *IfcOpeningElement*. La relación entre ambas clases viene definida por *IfcRelVoidsElement*. Por lo tanto, cualquier muro que tenga una relación de este tipo tendrá una abertura dentro de su perfil, pudiendo ser un hueco vacío o bien tener elementos incrustados, como puertas o ventanas, teniendo en este último caso una relación del tipo *IfcRelFillsElement* que se establece entre el hueco y el elemento incrustado.

///Relación de muro con hueco

```
#4608=IFCRELVOIDSELEMENT('18Ti_lEaTC7gsK1sXIg2JD',#20,$,$,#124,#4607);
```

///Muro al que se le define el hueco

```
↳ [#124=IFCWALL('18Ti_lEaTC7gsK',#20,'Muro:167380',$,'MuroE1:ParticF3n',#108,#123,'167380')]
```

///Hueco definido en el muro

```
↳ [#4607=IFCOPENINGELEMENT('1XvY5npw',#20,'Puerta:80x210cm:1',$,'Ope,#460,#46','169358')]
```

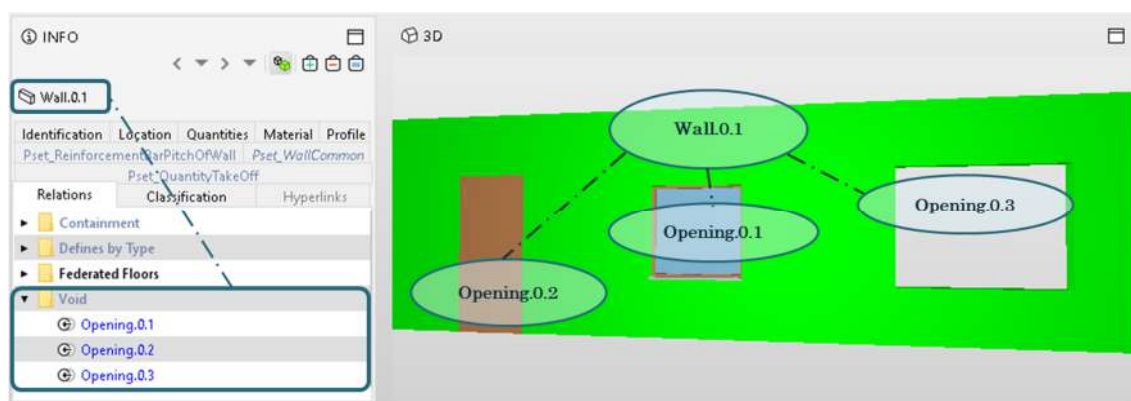


Figura 3.5.52. Tipos de relaciones de muros con huecos

En el caso de acceder a las relaciones contenidas dentro de la instancia del hueco *Opening.0.2*, se muestran dos tipos de relaciones: la que rellena el hueco con la puerta y, por otro lado, el vacío que genera el hueco dentro del muro.

///Relación de hueco con puerta

```
#4609=IFCRELFILLSELEMENT('2D7x3Yg_rflGun3vA1lvrc',#20,$,$,#4607,#1924);
```

///Hueco en el muro que se rellena con la puerta

```
↳ [#4607=IFCOPENINGELEMENT('1XvY5npw',#20,'Puerta:80x210cm:1',$,'Ope,#460,#46','169358')]
```

///Puerta que rellena el hueco de muro

```
↳ [#1924=IFCDOOR('18Ti_lEaTC7gsK1t',#20,'Puerta',$,'1hoja:80x210',#4611,#1919,'169358')]
```

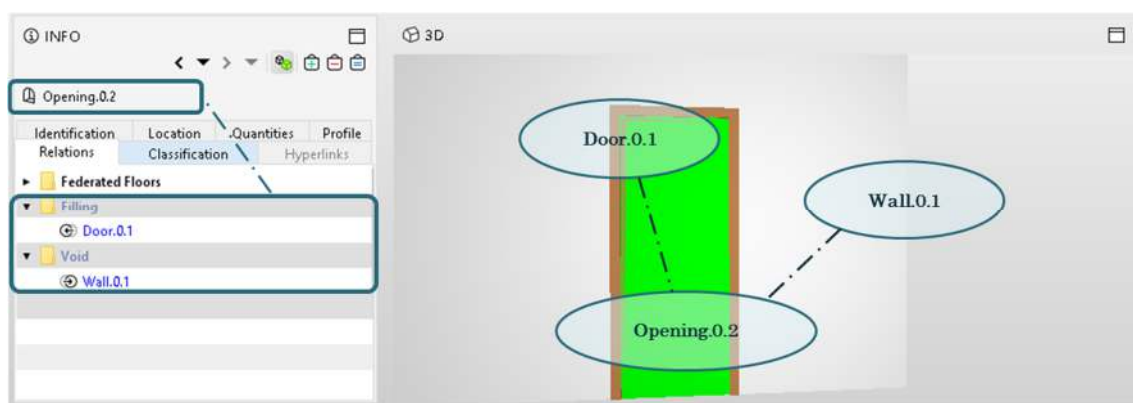


Figura 3.5.53. Relación de muro y puerta con hueco

De esta manera, existe una relación entre la puerta con el hueco y otra entre el muro y el hueco. Gracias a ello, se podrá saber qué puertas están incrustadas en qué muros. En el siguiente esquema se puede observar estas relaciones y también la relación de los elementos con el nivel en el que están contenidos. Cada conexión entre la relación y el elemento tiene dos definiciones que permiten identificar ambos elementos, una del primero al segundo y la otra del segundo al primero (*INV*). Por ejemplo, se puede saber si el hueco tiene o no uno o varios rellenos a través de la definición *HasFilling*, mientras que con *RelatedOpeningElement*, la relación identifica el hueco en el que se encuentra la puerta.²¹²

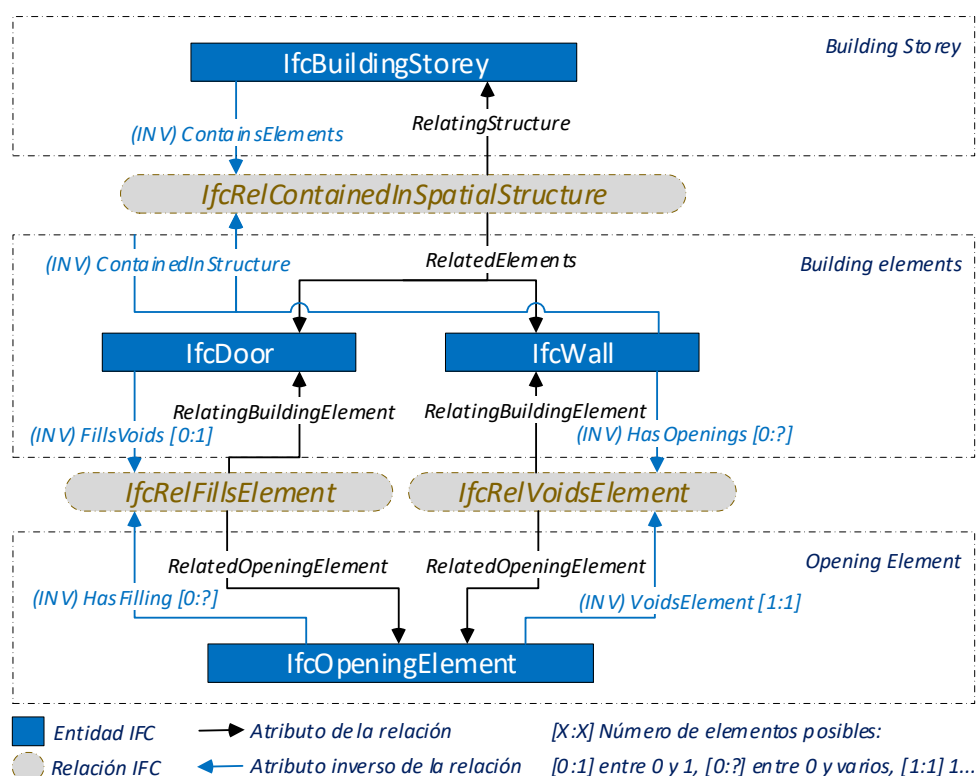


Figura 3.5.54. Relación de muro, puerta y hueco. Fuente* (ISO 16739)

²¹² *Generic BIM queries based on the IFC object model using graph theory*. E. Tauscher, H. Bargstädt, K. Smarsly. 2016.

Es importante destacar que no siempre una relación puede albergar múltiples objetos, por ello, cada relación contiene un rango de números específicos. Por ejemplo, en el caso de la relación del vacío con el muro, este puede tener entre 0 y varios huecos, mientras que la definición de relación del hueco con el propio hueco debe siempre ser una por cada hueco que contenga el muro.

Siguiendo el esquema anterior, es posible obtener todas las carpinterías del proyecto indicando el elemento en el que se hospedan. Para ello, es necesario coleccionar todas las puertas y ventanas del proyecto, acceder a los huecos con los que están relacionadas y finalmente obtener el elemento IFC al que se encuentran asociadas.

```

model = ifcopenshell.open('Muros_Ventanas_y_Puertas.ifc')
carpt = []
elem = model.by_type('IfcElement')
for e in elem:
    if e.is_a() in ('IfcDoor', 'IfcWindow'):
        dta = e.is_a(), e.Name, e.GlobalId, e.FillsVoids[0].RelatingOpeningElement.
        VoidsElements[0].RelatingBuildingElement.Name, e.FillsVoids[0].Relating
        OpeningElement.VoidsElements[0].RelatingBuildingElement.GlobalId
        carpt.append(dta)
df3 = pd.DataFrame(carpt)
df3.columns = ['Clase Ifc', 'Nombre', 'GUID', 'Elemento contenedor', 'GUID contenedor']
display(df3)

```

Id	Clase Ifc	Nombre	GUID	Elemento contenedor
0	IfcDoor	Puerta de 1 hoja:80 x 210 cm:163553	0pmjikAkL7yAETkf4eS6Z5	Muro básico:Partición con capa de yeso:162782
1	IfcDoor	Puerta de 1 hoja:80 x 210 cm:163556	0pmjikAkL7yAETkf4eS6Z0	Muro básico:Partición con capa de yeso:162798
2	IfcDoor	Puerta de 1 hoja:80 x 210 cm:163558	0pmjikAkL7yAETkf4eS6Z2	Muro básico:Fachada ventilada con aislamiento ...
3	IfcWindow	Ventana simple con contraventana:80 x 230 cm:	0pmjikAkL7yAETkf4eS6ZH	Muro básico:Por defecto - 30 cm:162941
4	IfcWindow	Ventana simple con contraventana:80 x 230 cm:	0pmjikAkL7yAETkf4eS6ZS	Muro básico:Por defecto - 30 cm:162941
5	IfcWindow	Ventana simple con contraventana:80 x 230 cm:	0pmjikAkL7yAETkf4eS6ab	Muro básico:Muro de ladrillo visto:162942
6	IfcWindow	Ventana simple con contraventana:80 x 230 cm:	0pmjikAkL7yAETkf4eS6aW	Muro básico:Partición con capa de yeso:162824
7	IfcWindow	Ventana simple con contraventana:80 x 230 cm:	0pmjikAkL7yAETkf4eS6am	Muro básico:Partición con capa de yeso:162824
8	IfcWindow	Ventana simple con contraventana:80 x 230 cm:	0pmjikAkL7yAETkf4eS6a7	Muro básico:Partición con capa de yeso:162782

Tabla 3.5.31. Extracto de ejemplares de puertas y ventanas

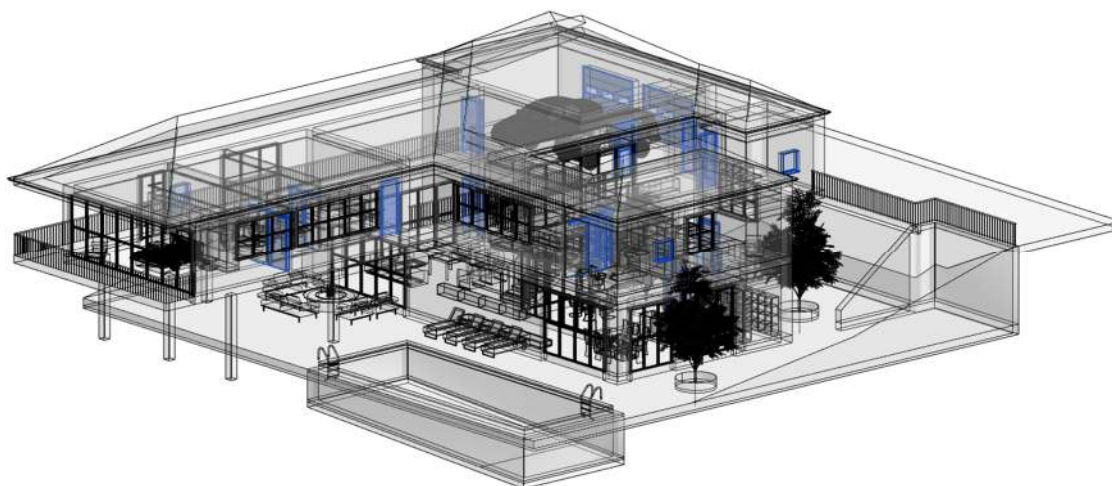


Figura 3.5.55. Muros y ventanas existentes en la vivienda unifamiliar

En el caso específico de las puertas, denominadas *IfcDoor* dentro del esquema IFC, se definen como elementos constructivos que se utilizan principalmente para proporcionar un acceso controlado a personas, mercancías, animales y vehículos. Incluye elementos móviles con operaciones articuladas, pivotantes, deslizantes, giratorias y plegables.

Puede tratarse de un elemento independiente, por ejemplo, si se trata de una puerta de acceso dentro del perímetro de una parcela, de un elemento incrustado en otra clase de elemento como un *IfcWall* o ser un subcomponente de un elemento compuesto, como un *IfcCurtainWall*. Por lo general, se trata de elementos que separan verticalmente diferentes espacios interiores y exteriores, aunque también pueden ser aberturas horizontales, como en el caso de las escotillas, las cuales se incrustan en clases de objetos horizontales o inclinados, como *IfcRoof* o *IfcSlab*.



Figura 3.5.56. Relación de puertas IFC con anfitriones

Al igual que ocurre con los *IfcWall*, las puertas pertenecen a los elementos de construcción definidos como *IfcBuiltElement* (anteriormente *IfcBuildingElement*) y también vienen definidas por sus propias características de tipo *IfcDoorType*. Este tipo especifica parámetros que son comunes a todas las instancias de dicho tipo, como el tipo de operación de apertura o las propiedades de sus subelementos, mientras que

IfcDoor contienen otras propiedades como el alto, el ancho y la dirección de apertura de la puerta. En cualquier caso, cada instancia de puerta contendrá las propiedades tanto específicas de la clase, como las del tipo que tenga asignado.

A partir de la versión IFC4, la clase *IfcDoor* contiene las definiciones de *PredefinedType* y *OperationType*. Además, en esa misma versión se sustituye la definición *IfcDoorStyle* por la de *IfcDoorType*. Por este motivo, al extraer las propiedades de diferentes versiones IFC, aparecerán o no los tipos predefinidos para las puertas. En el caso de la versión IFC3x2, se añade el tipo predefinido al tipo de objeto para que cada instancia tenga la definición que le corresponde.

```
ifcfile = ifcopenshell.open('Puertas_Tipos_2x3.ifc')
print(ifcfile.schema)
drs = ifcfile.by_type('IfcDoor')
drsprps = []
for d in drs:
    aa = [d.is_a(), d.Name, d.GlobalId, d.ObjectType, round(d.OverallHeight,2),
round(d.OverallWidth,2)]
    drsprps.append(aa)
df = pd.DataFrame(drsprps)
df.columns = ['IfcClass', 'Name', 'GlobalId', 'ObjectType', 'Height', 'Width']
display(df)
```

Id	IfcClass	Name	GlobalId	ObjectType	Height	Width
0	IfcDoor	Puerta_Acceso Cubierta...	3SnZVVdO59Jf2 YLHOGlutF	TRAPDOOR	1.1	1.1
1	IfcDoor	NBS_Puerta Entrada_Crb...	3SnZVVdO59Jf2 YLHOGI7iU	GATE	1.21	1.2
2	IfcDoor	Balconera practicable, 2 hojas:1600 x 2200mm:4...	3SnZVVdO59Jf2 YLHOGIBgT	DOOR	2.2	1.6

Tabla 3.5.32. Obtención de tipos de puertas en IFC2x3

En el caso de la versión IFC4 o versiones posteriores, ya es posible encontrar el tipo predefinido dentro del conjunto de atributos de cada instancia de puerta:

```
ifcfile = ifcopenshell.open('Puertas_Tipos_4.ifc')
print(ifcfile.schema)
drs = ifcfile.by_type('IfcDoor')
drsprps = []
for d in drs:
    aa = [d.is_a(), d.Name, d.GlobalId, d.PredefinedType, round(d.OverallHeight,2),
round(d.OverallWidth,2)]
    drsprps.append(aa)
df = pd.DataFrame(drsprps)
df.columns = ['IfcClass', 'Name', 'GlobalId', 'PredefinedType', 'Height', 'Width']
display(df)
```

Id	IfcClass	Name	GlobalId	PredefinedType	Height	Width
0	IfcDoor	Puerta_Acceso Cubierta...	3SnZVVdO59Jf2 YLHOGlutF	TRAPDOOR	1.1	1.1
1	IfcDoor	NBS_Puerta Entrada_Crb...	3SnZVVdO59Jf2 YLHOGl7iU	GATE	1.21	1.2
2	IfcDoor	Balconera practicable, 2 hojas:1600 x 2200mm:4...	3SnZVVdO59Jf2 YLHOGlBgT	DOOR	2.2	1.6

Tabla 3.5.33. Obtención de tipos de puertas en IFC4

Adicionalmente, a partir de la versión IFC4.3, aparecen dos nuevos tipos predefinidos, quedando actualmente los que se indican a continuación:

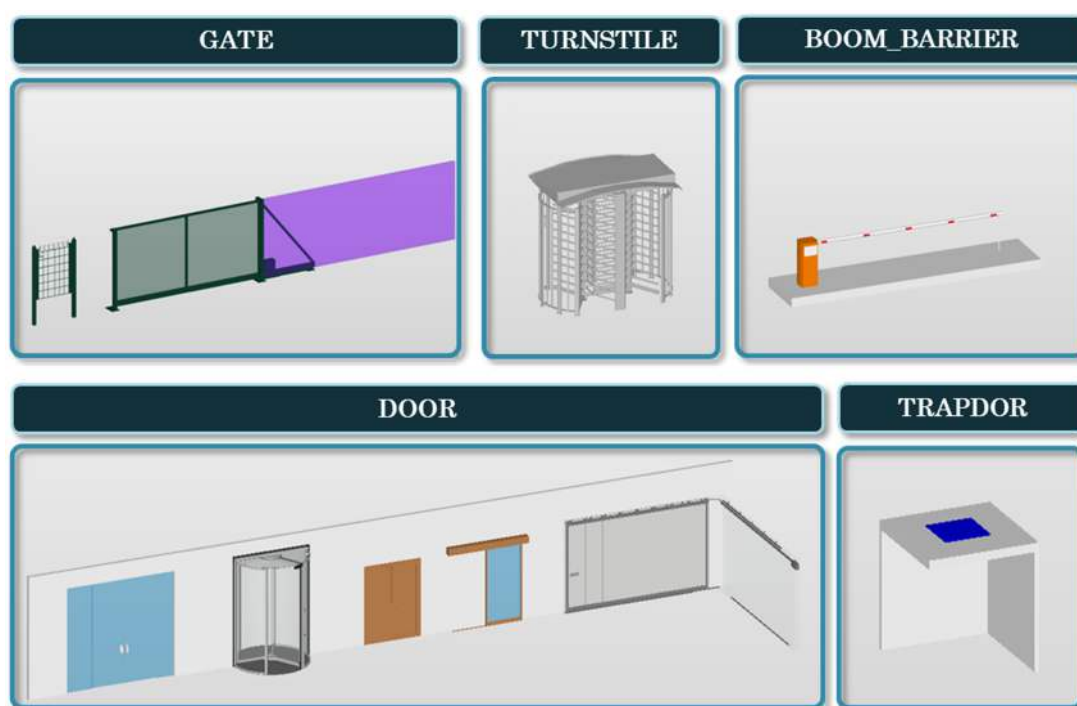


Figura 3.5.57. Tipos predefinidos de puertas IFC

Por lo tanto, cuando se genera el modelo en la versión IFC 4.3, se obtendrán los nuevos tipos predefinidos que se han incorporado en el esquema:

```

ifcfile = ifcopenshell.open('Puertas_Tipos_4.3.ifc')
drs = ifcfile.by_type('IfcDoor')
drsprps = []
for d in drs:
    aa = [d.is_a(), d.Name, d.GlobalId, d.PredefinedType, round(d.OverallHeight,2),
    round(d.OverallWidth,2)]
    drsprps.append(aa)
df = pd.DataFrame(drsprps)
df.columns = ['IfcClass', 'Name', 'GlobalId', 'PredefinedType', 'Height', 'Width']
display(df)

```

Id	IfcClass	Name	GlobalId	PredefinedType	Height	Width
0	IfcDoor	BL229:6:469152	3SnZVVdO59Jf2 YLHOGlutU	BOOM_BARRIER	1.4	5.44
1	IfcDoor	Puerta_Acceso Giratoria...	3SnZVVdO59Jf2 YLHOG17iT	TURNSTILE	1.18	0.85

Tabla 3.5.34. Obtención de tipos de puertas en IFC4.3

Hay dos representaciones principales para las instancias de puertas, Por un lado, las entidades *IfcDoor* que tienen definida una representación de forma de perfil de rectángulo 3D. Este perfil se puede utilizar para generar paramétricamente la geometría de una puerta. Si no se proporciona, el perfil de *IfcOpeningElement* se puede utilizar si la puerta llena una abertura. Los parámetros se especifican en el *IfcDoorType* relacionado que hace referencia a *IfcDoorLiningProperties* e *IfcDoorPanelProperties* para cada panel de la puerta. Por otro lado, las entidades *IfcDoor* que no se generan paramétricamente y solo tienen geometría 'Brep' o 'SurfaceModel'. Para obtener las propiedades del panel de la puerta, se pueden extraer los datos del archivo IFC y generar una tabla con la información:

```

model = ifcopenshell.open("Proj_2_ARC_IFC4.ifc")
doors = model.by_type("IfcDoor")
results = []
for door in doors:
    door_type = None
    for rel in door.IsDefinedBy:
        if rel.is_a("IfcRelDefinesByType"):
            door_type = rel.RelatingType
            break
    if door_type and hasattr(door_type, "HasPropertySets"):
        for prop in door_type.HasPropertySets:
            if prop.is_a("IfcDoorPanelProperties"):
                results.append({
                    "GUID": door.GlobalId,
                    "PanelDepth": getattr(prop, "PanelDepth", None),
                    "PanelWidth": getattr(prop, "PanelWidth", None),
                    "PanelPosition": getattr(prop, "PanelPosition", None),
                    "OperationType": getattr(prop, "OperationType", None),
                    "PanelOperation": getattr(prop, "PanelOperation", None)})
df = pd.DataFrame(results)
print(df)

```

GUID	PanelDepth	PanelWidth	PanelPosition	PanelOperation
3SnZVVdO59Jf2 YLHOGlutU	0.04	0.9	LEFT	SWINGING
3SnZVVdO59Jf2 YLHOG17iT	0.045	0.85	RIGHT	SLIDING

Tabla 3.5.35. Obtención de tipos en IFC4.3

El procedimiento desarrollado para las *IfcDoor*, puede ser replicado para las *IfcWindow*, con el objetivo de estructurar y analizar sus propiedades básicas. En este caso, se accede al modelo IFC y se extraen las instancias clasificadas como ventanas. Para cada una de ellas, se definen atributos fundamentales como el GUID, la clase IFC, el nombre, el tipo predefinido y las dimensiones. El resultado incluye tipologías tales como *SKYLIGHT*, *WINDOW* y *LIGHTDOME*, los cuales son tipos predefinidos de la clase.



Figura 3.5.58. Tipos predefinidos de ventanas IFC

```
model = ifcopenshell.open("Proj_2_ARC_IFC4.ifc")
windows = model.by_type("IfcWindow")
data = []
for i, window in enumerate(windows):
    data.append({
        "Id": i,
        "IfcClass": window.is_a(),
        "Name": getattr(window, "Name", ""),
        "GlobalId": window.GlobalId,
        "ObjectType": getattr(window, "ObjectType", ""),
        "Height": getattr(window, "OverallHeight", None),
        "Width": getattr(window, "OverallWidth", None),})
df = pd.DataFrame(data)
print(df)
```

Id	IfcClass	Name	GlobalId	PredefinedType	Height	Width
0	IfcWindow	Ventana techo claraboya	1AbCdEfGh YLHOGlutP	SKYLIGHT	1.2	1
1	IfcWindow	Ventana estándar salón	2IjKlMnOp YLHOGlutJ	WINDOW	1.5	1.5
2	IfcWindow	Cúpula translúcida sobre cubierta	3QrStUvWx YLHOGlutH	LIGHTDOME	0.8	0.8

Tabla 3.5.36. Obtención de tipos de ventanas en IFC4

3.5.8.4. Escaleras

La entidad *IfcStair* en el estándar IFC representa un conjunto de componentes que conforman una escalera completa dentro de un modelo de edificación. Esta entidad puede estar compuesta por subelementos, como tramos *IfcStairFlight*, descansillos *IfcLanding* y barandillas *IfcRailing*, que se relacionan con la escalera principal mediante una relación de agregación definida como *IfcRelAggregates*.

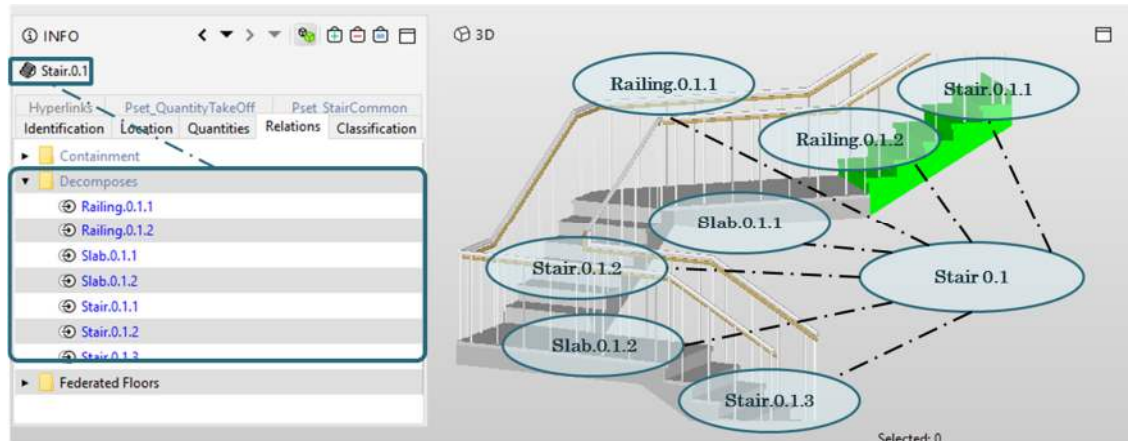


Figura 3.5.59. Relación entre escalera y sus subelementos

Para identificar y listar todas las subentidades que conforman cada instancia de *IfcStair* en el modelo, se desarrolla un nuevo código con *IfcOpenShell*. La rutina va recorriendo cada escalera presente en el modelo y se inspeccionan sus relaciones de descomposición para acceder a sus subelementos. Por cada subcomponente identificado, se extraen los atributos relevantes, como su clase IFC, GUID, nombre y tipo.

```

model = ifcopenshell.open('Relaciones_Escaleras.ifc')
# Obtener todas las escaleras del modelo
stairs = model.by_type("IfcStair")
rows = []
for stair in stairs:
    # Buscar relaciones de agregación
    for rel in stair.IsDecomposedBy:
        if rel.is_a("IfcRelAggregates"):
            for sub in rel.RelatedObjects:
                rows.append({
                    "StairGlobalId": stair.GlobalId,
                    "StairName": getattr(stair, "Name", ""),
                    "StairType": getattr(stair, "ObjectType", ""),
                    "SubcomponentClass": sub.is_a(),
                    "SubcomponentGlobalId": sub.GlobalId,
                    "SubcomponentName": getattr(sub, "Name", ""),
                    "SubcomponentType": getattr(sub, "ObjectType", "")
                })
df = pd.DataFrame(rows)
display(df)

```

Id	Stair GlobalId	Stair Name	Subcomp. Class	Subcomp. GlobalId	Subcomp. Name
0	1KH8\$45U18 OkbDQkdfq6n	Escalera moldeada in situ:167371	IfcStair Flight	1KH8\$45U18 OkbDQkdfq6i	Escalera: 167371 Run 1
1	1KH8\$45U18 OkbDQkdfq6n	Escalera moldeada in situ:167371	IfcStairFlight	1KH8\$45U18 OkbDQkdfq65	Escalera: 167371 Run 2
2	1KH8\$45U18 OkbDQkdfq6n	Escalera moldeada in situ:167371	IfcStairFlight	1KH8\$45U18 OkbDQkdfq9B	Escalera: 167371 Run 3
3	1KH8\$45U18 OkbDQkdfq6n	Escalera moldeada in situ: 167371	IfcSlab	1KH8\$45U18V OkbDQkdfq9x	Escalera: 167371 Land...
4	1KH8\$45U18 OkbDQkdfq6n	Escalera moldeada in situ:167371	IfcSlab	1KH8\$45U18V OkbDQkdfq99	Escalera: 167371 Land...
5	1KH8\$45U18 OkbDQkdfq6n	Escalera moldeada in situ:167371	IfcRailing	1KH8\$45U18V OkbDQkdfq96	Barandilla:Barrotes redondos:167484
6	1KH8\$45U18 OkbDQkdfq6n	Escalera moldeada in situ:167371	IfcRailing	1KH8\$45U18V OkbDQkdfq8w	Barandilla:Barrotes redondos:167488

Tabla 3.5.37. Obtención de subelementos de escaleras

Otro procedimiento consiste en extraer la información contenida en el conjunto de propiedades *Pset_StairFlightCommon* para todos los tramos de escaleras del proyecto. Para ello, se recurre a la entidad *IfcStairFlight*, que representa cada uno de los tramos individuales. Para cada tramo, se accede al conjunto de propiedades, el cual incluye datos relevantes como el número de contrahuellas, la altura de la contrahuella o la longitud de la huella. Junto con estas propiedades se registran también el GUID y el nombre del tramo.

```
# Obtener todos los tramos de escaleras
flights = model.by_type("IfcStairFlight")
rows = []
for flight in flights:
    # Obtener el pset de propiedades comunes del tramo
    pset = ifcopenshell.util.element.get_pset(flight, "Pset_StairFlightCommon")
    if pset:
        pset["GlobalId"] = flight.GlobalId
        pset["Name"] = getattr(flight, "Name", "")
        rows.append(pset)
display(pd.DataFrame(rows))
```

GlobalId	Name	Number OfRiser	Number OfTreads	Riser Height	Tread Length	Nosing Length
1KH8\$45U18V OkbDQkdfq6i	Escalera:167371 Run 1	5	4	0.16667	0.3	0.03
1KH8\$45U18V OkbDQkdfq65	Escalera:167371 Run 2	6	5	0.16667	0.3	0.03
1KH8\$45U18V OkbDQkdfq9B	Escalera:167371 Run 3	6	5	0.16667	0.3	0.03

Tabla 3.5.38. Propiedades de tramos de escaleras IFC

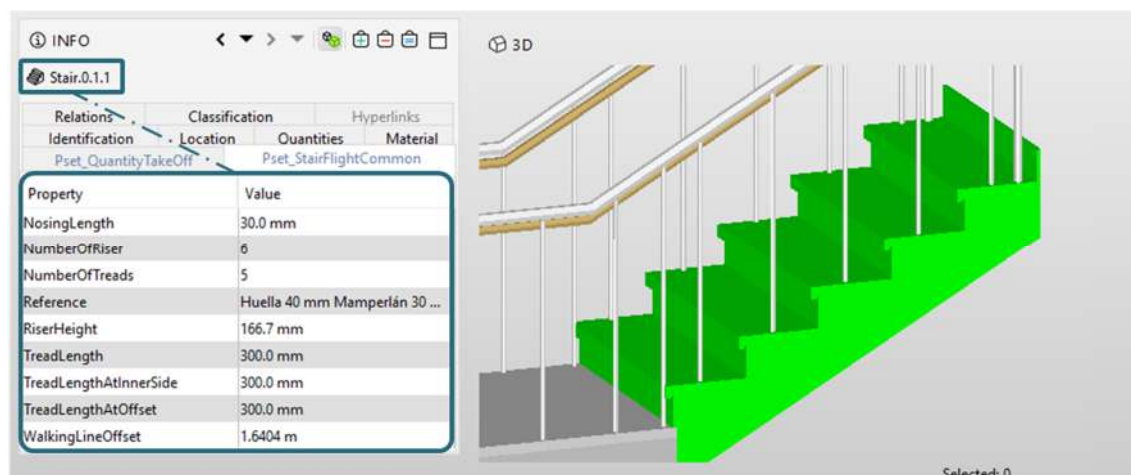


Figura 3.5.60. Propiedades de tramo de escalera

La entidad *IfcStair* ha evolucionado a partir de la versión IFC2x3 hacia una representación más estructurada y detallada en las nuevas versiones. En IFC2x3, aunque se podía clasificar el tipo de escalera, muchas propiedades dependían de implementaciones específicas del software y no estaban estandarizadas. Con IFC4, se amplía la clasificación mediante *PredefinedType* y se estandarizan conjuntos de propiedades como *Pset_StairCommon* y *Pset_StairFlightCommon*, lo que permite describir con mayor precisión aspectos geométricos y funcionales de las escaleras. En cuanto a los subcomponentes, como *IfcStairFlight* y *IfcLanding*, se conservan en todas las versiones, pero en IFC4 y 4.3 se introducen definiciones de tipo y propiedades adicionales.

3.5.8.5. Armazones estructurales

La definición *IfcBeam* hace referencia a un elemento estructural lineal, diseñado principalmente para resistir cargas mediante flexión. En la práctica, se refiere a las vigas, componentes fundamentales en la construcción de edificios. Como parte de la jerarquía de *IfcBuiltElement*, esta entidad permite describir tanto las características geométricas y materiales del elemento como sus relaciones estructurales y funcionales dentro del modelo IFC.

Para facilitar la clasificación semántica de las vigas, el atributo *PredefinedType* de *IfcBeam* especifica diferentes subtipos con propósitos particulares. Entre ellos, algunos que hacen referencia a edificación, son *BEAM*, que se refiere a la viga estándar, *JOIST*, que representa una vigueta secundaria de soporte,

HOLLOWCORE, una viga pretensada con núcleo hueco común en elementos prefabricados; y *LINTEL*, que describe un dintel ubicado sobre aberturas como puertas o ventanas. Otros tipos menos conocidos son *SPANDREL*, una viga alta usada en fachadas, y *T_BEAM*, una viga que actúa conjuntamente con la losa, típica de construcciones de hormigón. A partir del modelo estructural del edificio de viviendas, todos estos armazones pueden ser extraídos junto con la información de cada instancia.

```

model = ifcopenshell.open("Proj_3_STR_IFC4.ifc")
#Obtener todas las instancias de IfcBeam
beams = model.by_type("IfcBeam")
data = []
#Recorrer todas las vigas
for beam in beams:
    psets = ifcopenshell.util.element.get_psets(beam)
    beam_data = {
        "GlobalId": beam.GlobalId, "Name": beam.Name,
        "PredefinedType": getattr(beam, "PredefinedType", None)}
    #Extraer propiedades del Pset_BeamCommon
    pset_common = psets.get("Pset_BeamCommon", {})
    for key in ["Reference", "LoadBearing", "FireRating", "IsExternal"]:
        beam_data[key] = pset_common.get(key)
    data.append(beam_data)
df = pd.DataFrame(data)
pd.set_option('display.max_rows', None)
display(df)

```

GlobalId	Name	Predefined Type	Reference	Load Bearing	Fire Rating	Is External
008JNpu7v5 Bu3gbcTmqx	Hormigón-Viga rectangular:250 x 300:361748	BEAM	250 x 300	True	R120	True
008JNpu7v5 Bu3gbcTmqv	Vigueta y casetón con repisas:150 x 500 Forma ...	T_BEAM	150 x 500 Forma en T	True	R120	True
3WKIOY3kb eUrebMfYwX	IPE - Secciones en I con alas paralelas:IPE 24...	SPANDREL	IPE 240	True	R120	True
2ltsebu89AR Q8smfGZtH	Hormigón-Viga rectangular:100 X 250 mm:367416	JOIST	100 X 250 mm	True	R120	False

Tabla 3.5.39. Propiedades de armazones estructurales

Varias de las propiedades mostradas en la tabla anterior, se encuentran en el conjunto general de propiedades de las vigas, denominado *Pset_BeamCommon*, que contiene atributos como la referencia de la instancia, la definición de exterior o interior, de si es portante o no, y datos de protección contra fuego.

Al igual que con las cantidades de los muros, las vigas también contienen definiciones de longitudes, áreas, volúmenes y pesos que pueden ser extraídas siempre y cuando se hayan configurado correctamente los archivos IFC, en el siguiente caso, con la versión IFC2x3.

```

model = ifcopenshell.open("Proj_3_STR_IFC2x3.ifc")
#Filtrar todas las vigas
beams = model.by_type("IfcBeam")
data = []
#Recorrer cada viga y extraer sus cantidades
for beam in beams:
    qtos = ifcopenshell.util.element.get_psets(beam, qtos_only=True)
    base_qto = qtos.get("Qto_BeamBaseQuantities", {})
    if base_qto:
        row = {
            "GlobalId": beam.GlobalId, "Name": beam.Name,}
        #Añadir cada quantity como una columna
        row.update(base_qto)
        data.append(row)
# Crear el DataFrame
df = pd.DataFrame(data)
display(df)

```

GlobalId	Name	Length	Cross Section Area	Outer Surface Area	Gross Volume	Net Volume
008JNpu7v5N Bu3gbcTmqxB	Hormigón-Viga rectangular:250 x 300:361748	1.94	0.075	2.147	0.146	0.14
008JNpu7v5N Bu3gbcTmqv6	Vigueta y casetón con repisas:150 x 500 Forma ...	6.87	1.672	3.324	0.022	0.184
3Y6HjZxwT0t A86zM0Fh7TE	Hormigón-Viga rectangular:250 x 300:371263	1.87	0.075	2.057	0.140	0.144
008JNpu7v5N Bu3gbcTmq6p	IPE - Secciones en I con alas paralelas:IPE 24...	5.77	0.003	5.324	0.022	0.022

Tabla 3.5.40. Cantidades de armazones estructurales

En cuanto a su evolución, *IfcBeam* ha estado presente desde las primeras versiones del esquema con una definición funcional, pero las posibilidades de clasificación y descripción eran más limitadas. Las propiedades se agregaban a través de extensiones o configuraciones específicas de cada software de autoría. Con la llegada de IFC4, se amplió la taxonomía de tipos predefinidos y se estandarizó el uso de conjuntos de propiedades, especialmente *Pset_BeamCommon*, que incluye datos como capacidad portante, clasificación de resistencia al fuego, y parámetros de referencia. En IFC4.3, aunque la estructura de *IfcBeam* no cambia, se mejora su aplicación para proyectos de infraestructura.

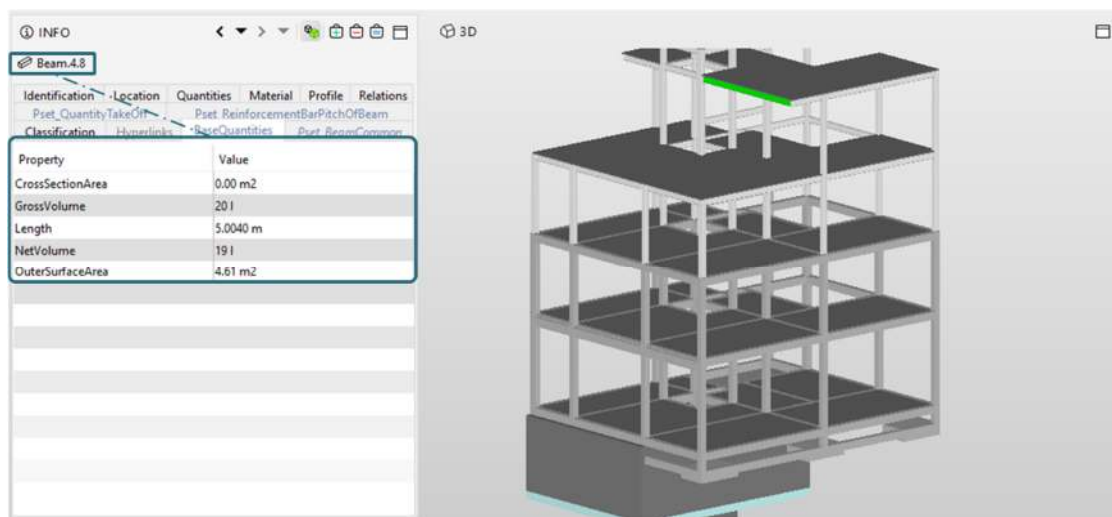


Figura 3.5.61. Cantidades de instancia de armazón estructural

3.5.8.6. Sistemas, aparatos y segmentos de instalaciones

Dentro de las clases relacionadas con instalaciones, los *IfcSystem* representan un conjunto lógico de elementos que trabajan juntos para una función común dentro de un edificio. No describe componentes físicos, sino agrupados funcionalmente. Un caso especial es el *IfcDistributionSystem*, subtipo de *IfcSystem*, que se utiliza para describir sistemas de instalaciones como HVAC, eléctricos, sanitarios y redes de telecomunicaciones, entre otros.

Este tipo puede especificar su función concreta mediante el atributo *PredefinedType*, por ejemplo, *AIRCONDITIONING*, *VENTILATION*, *ELECTRICAL* o *DRAINAGE*. Los elementos que forman parte de un sistema están relacionados mediante la entidad *IfcRelAssignsToGroup*.

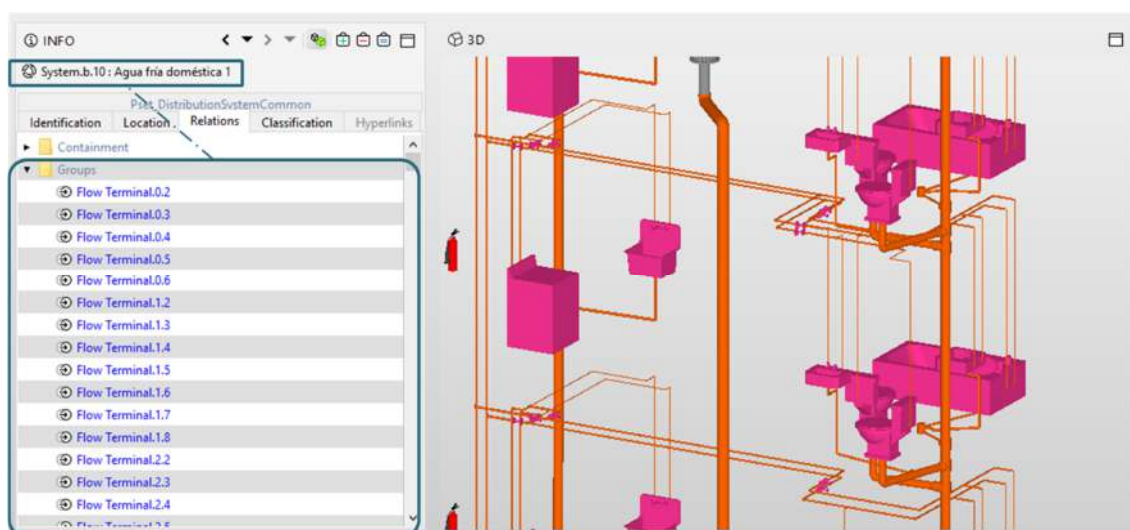


Figura 3.5.62. Relaciones entre instancias de objetos y el sistema

El siguiente código busca todas las entidades del tipo *IfcSystem* y sus derivados. Para cada sistema encontrado, recopila su nombre, clase, tipo predefinido y su GUID. Luego, se obtienen todos los objetos asignados a cada sistema, calculando el número

total de elementos que lo conforman y desglosando cuántos pertenecen a cada clase IFC. Finalmente, compila toda esta información en un DataFrame.

```

model = ifcopenshell.open("Proj_3_MEP.ifc")
systems = model.by_type("IfcSystem")
data = []
#Recorrer cada sistema
for system in systems:
    system_class = system.is_a()
    predefined_type = getattr(system, "PredefinedType", None) if system_class ==
    "IfcDistributionSystem" else None
    #Obtener objetos asignados al sistema
    related = model.get_inverse(system)
    elements = []
    for rel in related:
        if rel.is_a("IfcRelAssignsToGroup"):
            elements.extend(rel.RelatedObjects)
    #Contar objetos por clase IFC
    type_counts = {}
    for obj in elements:
        cls = obj.is_a()
        type_counts[cls] = type_counts.get(cls, 0) + 1
    #Construir filas
    row = {
        "Nombre": system.Name,
        "Tipo de sistema": str(predefined_type),
        "GUID": system.GlobalId,
        "Objetos totales": len(elements),
        "Recuento por tipos": type_counts}
    data.append(row)
display(pd.DataFrame(data))

```

Nombre	Tipo de sistema	GUID	Objetos totales	Recuento por tipos
Agua fría doméstica 1	DOMESTIC COLD WATER	2sNILs8cDDEf6eQh0rVxiq	989	{'IfcFlowTerminal': 24, 'IfcPipeSegment': 139...
Sanitario 2	SEWAGE	2N6MZfnF954ehuWMvevNe7	515	{'IfcFlowTerminal': 16, 'IfcPipeSegment': 76, ...
Agua caliente doméstica 1	DOMESTIC HOT WATER	0QCki30KXFBf0K8P9M2JmT	875	{'IfcFlowTerminal': 18, 'IfcPipeSegment': 120...
Ventilación 1	VENT	1kocamJ181ASlxOxhvK	5	{'IfcPipeFitting': 1, 'IfcPipeSegment': 1, 'If...
Suministro hidrónico 2	HEATING	1kxiYI2X91fBRqbSUi3\$B6	3	{'IfcPipeSegment': 1, 'IfcDistributionPort': 2}
Pluviales 1	SEWAGE	1kxiYI2X91fBRqbSUi3\$fs	77	{'IfcBuildingElementProxy': 3, 'IfcPipeSegment...

Tabla 3.5.41. Extracto de sistemas de instalaciones

IfcSanitaryTerminal es una entidad del esquema que representa un tipo específico de terminal de flujo, denominado *IfcFlowTerminal*, utilizado en sistemas de instalaciones sanitarias. Se refiere a dispositivos destinados al uso para funciones higiénicas. Estos objetos suelen encontrarse en instalaciones de fontanería y saneamiento y forman parte de un sistema funcional dentro del modelo. El atributo *PredefinedType* permite clasificar aún más estos terminales en tipos estandarizados como *SINK*, *TOILETPAN*, *BATH*, *BIDET*, etc.

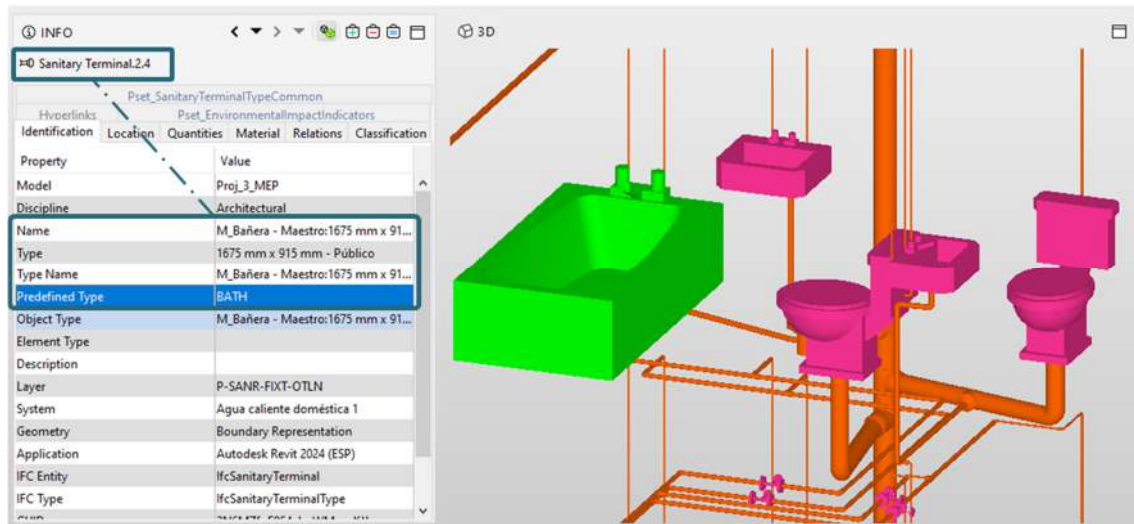


Figura 3.5.63. Atributos de aparatos sanitarios

Para obtener estos datos de los aparatos sanitarios, se genera un código que extrae todos los objetos de la clase. Para cada uno de estos aparatos o terminales, recopila información como su GUID, nombre, tipo predefinido y su clase IFC. Estos datos se almacenan en un DataFrame, que permite visualizarlos, analizarlos o exportarlos de forma estructurada.

```
model = ifcopenshell.open("Proj_3_MEP.ifc")
# Obtener todos los IfcSanitaryTerminal
sanitary_terminals = model.by_type("IfcSanitaryTerminal")
data = []
for st in sanitary_terminals:
    row = {
        "GUID": st.GlobalId,
        "Nombre": st.Name,
        "Tipo predefinido": getattr(st,
        "PredefinedType", None),
        "Tipo": st.is_a(),}
    data.append(row)
df = pd.DataFrame(data)
# Mostrar todas las filas
pd.set_option('display.max_rows', None)
display(df)
```

GUID	Nombre	Tipo predefinido	Clase
2N6MZfnF954 ehuWMvevKI	M_Sanitario - Cisterna:Público - Vaciado mayor...	WCSEAT	IfcSanitaryTerminal
2N6MZfnF954 ehuWMvevKJ	M_Bañera - Maestro:1675 mm x 915 mm – Público...	BATH	IfcSanitaryTerminal
1x_wdW6OXB bQYP1OzE4ff	M_Lavabo - Montado en muro:485 mm x 355 mm – P...	WASH HANDBASIN	IfcSanitaryTerminal
1x_wdW6OXB QYP1OzE4fV	M_Compartimento de ducha - Esquina:865 mm x 81...	SHOWER	IfcSanitaryTerminal
2Roa_JYlf6L8 Kfy9MklC_	M_Fregadero - Servicio:560 mm x 485 mm - Ofici...	SINK	IfcSanitaryTerminal

Tabla 3.5.42. Extracto de instancias de apartatos sanitarios

La definición *IfcPipeSegment* es utilizada para representar tramos físicos de tubería dentro de un sistema de instalaciones. Estos segmentos pueden ser rectos, curvos o flexibles, y generalmente están definidos por su geometría, material y conexiones. El atributo *PredefinedType* permite clasificar el segmento según su forma o comportamiento, con opciones como *RIGIDSEGMENT*, *FLEXIBLESEGMENT* o *SPOOL*. Esta clase es define el recorrido y el contenido de los sistemas de tuberías del modelo, siendo parte del sistema de objetos conectados.

Con el siguiente código se recopilan todos los objetos del tipo *IfcPipeSegment*, se extraen sus tipos predefinidos, clase IFC y, además, se accede al conjunto de cantidades estándar *Qto_PipeSegmentBaseQuantities* para obtener la longitud del tramo de tubería, si está disponible en el modelo.

```

model = ifcopenshell.open("Proj_3_MEP.ifc")
# Obtener todos los IfcPipeSegment
segments = model.by_type("IfcPipeSegment")
data = []
for seg in segments:
    # Obtener cantidades base (si existen)
    qtos = util.get_pset(seg, "Qto_PipeSegmentBaseQuantities")
    length = qtos.get("Length") if qtos else None
    data.append({
        "GlobalId": seg.GlobalId,
        "Name": seg.Name,
        "PredefinedType": getattr(seg, "PredefinedType", None),
        "Length": length,
        "Type": seg.is_a()})
df = pd.DataFrame(data)
display(df)

```


GUID	Nombre	Tipo predefinido	Longitud	Clase IFC
0kChS_DJnD5wGm8gWmiCRN	Tipos de tubería:Norma:658273	SPOOL	0.543539	IfcPipeSegment
0kChS_DJnD5wGm8gWmiCpt	Tipos de tubería:Norma:658369	SPOOL	0.302424	IfcPipeSegment
0kChS_DJnD5wGm8gWmiC5i	Tipos de tubería:Norma:658650	FLEXIBLE SEGMENT	0.745705	IfcPipeSegment
0kChS_DJnD5wGm8gWmiC5O	Tipos de tubería:Norma:658670	RIGID SEGMENT	2.663181	IfcPipeSegment
0kChS_DJnD5wGm8gWmiC3L	Tipos de tubería:Norma:658787	RIGID SEGMENT	0.389988	IfcPipeSegment
0kChS_DJnD5wGm8gWmiC1J	Tipos de tubería:Norma:658917	RIGID SEGMENT	0.06117	IfcPipeSegment

Tabla 3.5.43. Extracto de instancias de segmentos de tuberías

A lo largo de las últimas versiones de IFC, las entidades *IfcSystem*, *IfcSanitaryTerminal* e *IfcPipeSegment* han ido cambiando en precisión y capacidad de modelado. Un punto importante es que en IFC2x3, *IfcSystem* se usaba como agrupador funcional sin clasificaciones específicas, mientras que en IFC4 se amplió con subtipos y atributos como *PredefinedType*. *IfcSanitaryTerminal*, también amplió en IFC4 sus tipos predefinidos, mejorando también la integración con sistemas de instalaciones. En cuanto a *IfcPipeSegment*, pasó de ser un componente básico en IFC2x3 a incluir conjuntos de cantidades y mayor conectividad funcional en IFC4.

3.5.8.7. Espacios

IfcSpace es la entidad que representa un espacio físico delimitado dentro de un edificio, como una habitación, vestíbulo o zona técnica. Su función principal es definir áreas funcionales para usos específicos, siendo clave para el análisis de superficies, cálculos energéticos y planificación arquitectónica. Puede estar asociado a zonas, o niveles, y admite propiedades como volumen, uso previsto o condiciones ambientales.

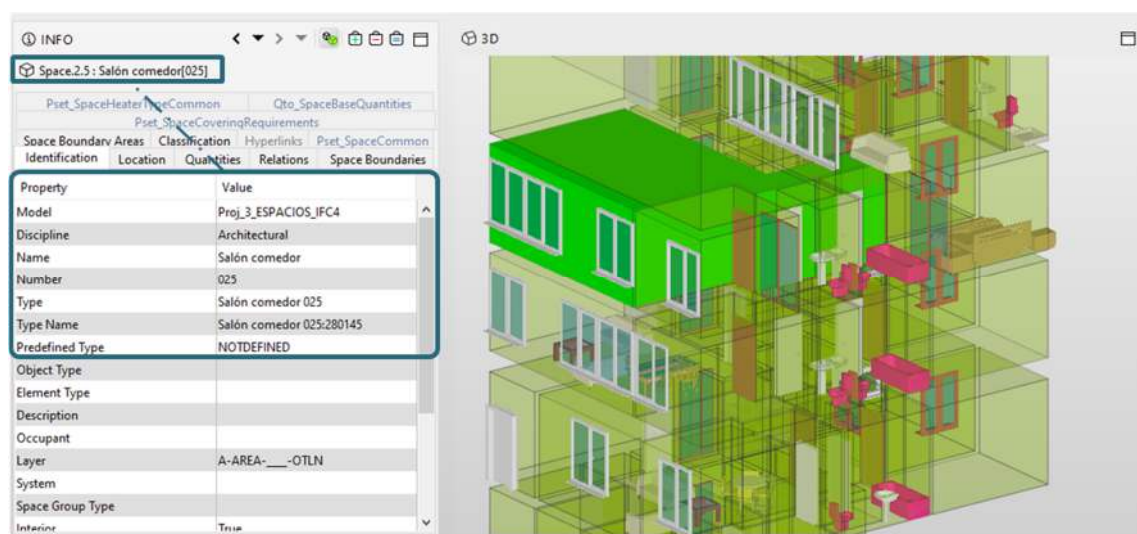


Figura 3.5.64. Atributos generales de espacios

Con el objetivo de extraer los datos de los espacios, se genera un nuevo código realiza colecta todas las entidades *IfcSpace* y, para cada instancia, se recuperan atributos relevantes como el GUID, el número, el nombre, el tipo predefinido y la altura. Además, se determina el nivel al que pertenece cada espacio mediante el análisis de relaciones jerárquicas con entidades *IfcBuildingStorey*.

```

modelo = ifcopenshell.open("Proj_3_ESPACIOS_IFC4.ifc")
#Obtener todos los espacios
espacios = modelo.by_type("IfcSpace")
def obtener_nivel(espacio):
    for rel in espacio.Decomposes:
        if rel.RelatingObject.is_a("IfcBuildingStorey"):
            return rel.RelatingObject.Name
    for rel in getattr(espacio, "ContainedInStructure", []):
        if rel.RelatingStructure.is_a("IfcBuildingStorey"):
            return rel.RelatingStructure.Name
    return None
#Función para obtener la altura
def obtener_altura_superior(espacio):
    for rel in espacio.HasBoundaries:
        if hasattr(rel, "RelatingSpace") and rel.RelatingSpace == espacio:
            if hasattr(rel, "ConnectionGeometry") and rel.ConnectionGeometry:
                geom = rel.ConnectionGeometry
                if hasattr(geom, "Height"):
                    return geom.Height
    return None
return None
#Recopilar datos
datos = []
for espacio in espacios:
    datos.append({
        "GUID": espacio.GlobalId, "Nombre": espacio.Name,
        "Nombre Largo": getattrs(espacio, "LongName", None),
        "Tipo Predefinido": getattrs(espacio, "PredefinedType", None),
        "Altura Superior": obtener_altura_superior(espacio),
        "Nivel": obtener_nivel(espacio)})
df = pd.DataFrame(datos)
display(df)

```

GUID	Número	Nombre	Tipo Predefinido	Altura	Nivel
1NQVyFx0X01BkUZlqio85l	8	Salón comedor	INTERNAL	2.8	Nivel 1
1NQVyFx0X01BkUZlqio85g	9	Ascensor	INTERNAL	15.32	Nivel 1
1NQVyFx0X01BkUZlqio85_	12	Dormitorio	INTERNAL	2.5	Nivel 1
1NQVyFx0X01BkUZlqio8I_	39	Patio	EXTERNAL	15.32	Nivel 0

Tabla 3.5.44. Extracto de instancias de espacios del edificio

Cada espacio contiene un conjunto denominado *Pset_SpaceCoveringRequirements*, desde donde se obtienen los acabados definidos para suelo, paredes y techo. Además, en el caso del edificio de viviendas, cada espacio se encuentra dentro de un departamento o vivienda específico.

```
#Obtener todos los espacios
espacios = modelo.by_type("IfcSpace")
#Recopilar datos
datos = []
for espacio in espacios:
    psets = get_psets(espacio)
    # Extraer acabados desde Pset_SpaceCoveringRequirements
    suelo = psets.get("Pset_SpaceCoveringRequirements", {}).get("FloorCovering")
    pared = psets.get("Pset_SpaceCoveringRequirements", {}).get("WallCovering")
    techo = psets.get("Pset_SpaceCoveringRequirements", {}).get("CeilingCovering")
    datos.append({
        "GUID": espacio.GlobalId, "Nombre": espacio.Name,
        "Departamento": getattr(espacio, "ObjectType", None),
        "Acabado Suelo": suelo, "Acabado Pared": pared, "Acabado Techo": techo})
df = pd.DataFrame(datos)
display(df)
```

GUID	Número	Departamento	Acabado Suelo	Acabado Pared	Acabado Techo
1NQVyFx0X01BkUZlqio85l	8	Vivienda 4	Suelo laminado	Pintura vinílica	Plafón de yeso
1NQVyFx0X01BkUZlqio85r	10	Vivienda 4	Loseta cerámica	Pintura vinílica	Plafón de yeso
1NQVyFx0X01BkUZlqio85v	13	Vivienda 4	Loseta antiderrapante	Azulejo	Plafón de yeso

Tabla 3.5.45. Extracto con información de acabados de espacios

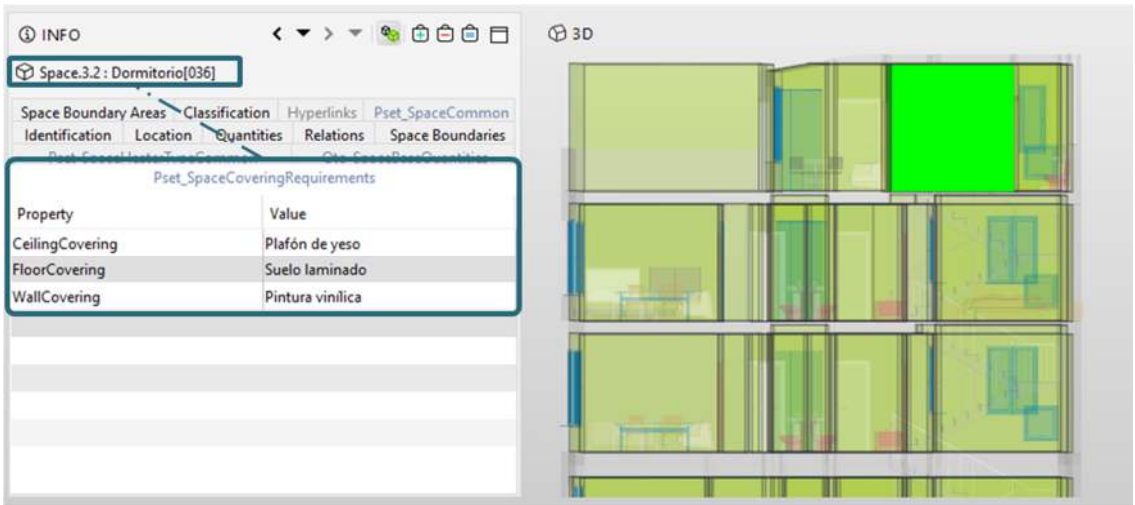


Figura 3.5.65. Atributos de acabados de espacios

3.5.8.8. Información del proyecto

La entidad *IfcProject* representa el proyecto BIM completo dentro del modelo IFC. Esta entidad es el punto de entrada jerárquico del modelo y contiene información general como el nombre del proyecto, su descripción, fase, unidades utilizadas y referencias al historial de propiedad. Desde este historial, denominado *IfcOwnerHistory*, se accede a *IfcPerson*, *IfcOrganization* e *IfcApplication*, que contienen información sobre el autor, la organización y el software que creó el modelo. Además, se accede al atributo *UnitsInContext*, que es una instancia de *IfcUnitAssignment* y agrupa todas las unidades definidas, como *LENGTHUNIT*, *AREAUNIT*, etc., que a su vez son subtipos de *IfcUnit*.

Para obtener toda esta información, se genera una rutina que localiza la entidad *IfcProject* y extrae una serie de atributos relevantes como: identificador global, nombre, descripción, fase del proyecto y unidades. Estas unidades se filtran por tipo y se colocan en filas separadas. Además, se accede a los metadatos de creación del archivo, convirtiendo la fecha *Unix* a un formato legible. Por último, se extrae el tipo de esquema y se compila toda la información en un DataFrame.

```

modelo = ifcopenshell.open("Proj_2_ARC_IFC4.ifc")
proyecto = modelo.by_type("IfcProject")[0]
#Datos generales del proyecto
datos = {
    "ID Global": proyecto.GlobalId, "Nombre": proyecto.Name,
    "Nombre largo": getattr(proyecto, "LongName", None),
    "Descripción": getattr(proyecto, "Description", None),
    "Fase": getattr(proyecto, "Phase", None),}
#Inicializar columnas de unidades
tipos_unidades = ["LENGTHUNIT", "AREAUNIT", "VOLUMEUNIT",
    "MASSUNIT", "TIMEUNIT", "PLANEANGLEUNIT"]
for tipo in tipos_unidades:
    datos[tipo] = None
#Asignar unidades por tipo
if hasattr(proyecto, "UnitsInContext") and proyecto.UnitsInContext:
    for unidad in proyecto.UnitsInContext.Units:
        tipo = getattr(unidad, "UnitType", None)
        nombre = getattr(unidad, "Name", None)
        if tipo in datos:
            datos[tipo] = nombre
#Extraer metadatos del archivo
owner = proyecto.OwnerHistory
person = owner.OwningUser.ThePerson
org = owner.OwningUser.TheOrganization
app = owner.OwningApplication
from datetime import datetime
fecha_legible = datetime.fromtimestamp(owner.CreationDate).strftime("%d/%m/%Y
%H:%M")
datos["Fecha de creación"] = fecha_legible
datos["Autor"] = person.FamilyName + ", " + person.GivenName if person else None
datos["Organización"] = org.Name if org else None

```

```

datos["Software"] = app.ApplicationFullName if app else None
datos["Version IFC"] = modelo.schema_identifier
df = pd.DataFrame([datos]).T
display(df)

```

ID Global	3cFk7muNn3Q8BShZaAPWw1
Nombre	5658255
Nombre largo	Edificio de viviendas
Descripción	Proyecto constructivo de ejecución de edificio de viviendas
Fase	Fase de desarrollo de proyecto
LENGTHUNIT	METRE
AREAUNIT	SQUARE_METRE
VOLUMEUNIT	CUBIC_METRE
MASSUNIT	GRAM
TIMEUNIT	SECOND
PLANEANGLEUNIT	DEGREE
Fecha de creación	01/07/2024 19:35
Autor	samuel.triana.gonzález
Organización	STG
Software	Autodesk Revit 2024 (ESP)
Version IFC	IFC4

Tabla 3.5.46. Datos de proyecto extraídos del modelo IFC

Además de los datos generales del proyecto, el archivo IFC contiene el nivel del *IfcBuilding* en el que también hay información almacenada. El siguiente código extrae información de datos del edificio, y también del emplazamiento, denominado *IfcSite* y de los niveles, *IfcBuildingStorey*. Primero se recolectan los atributos generales y luego se construye una tabla en forma de diccionario con estos datos. A esta tabla se le añaden filas adicionales, una por cada nivel, en las que se registra el nombre del nivel como atributo y su altura como valor.

```

modelo = ifcopenshell.open("Proj_2_ARC_IFC4.ifc")
# Obtener entidades clave
edificio = modelo.by_type("IfcBuilding")[0]
emplazamiento = modelo.by_type("IfcSite")[0]
niveles = modelo.by_type("IfcBuildingStorey")
# Construir datos generales
datos_edificio = {

```



```

"ID Global": edificio.GlobalId,
"Nombre del edificio": edificio.Name,
"Descripción": edificio.Description,
"Nombre del emplazamiento": emplazamiento.Name,
"Altitud": getattr(emplazamiento, "RefElevation", None),
"Número de niveles": len(niveles)}

# Convertir a DataFrame
df_general = pd.DataFrame.from_dict(datos_edificio, orient="index",
columns=["Valor"])

# Añadir una fila por cada nivel (nombre del nivel como clave)
for nivel in niveles:
    df_general.loc[f"Nivel: {nivel.Name}"] = nivel.Elevation
display(df_general)

```

ID Global	3cFk7muNn3Q8BShZaAPWwk
Nombre del edificio	Edificio de viviendas
Descripción	Proyecto de edificación para encargo de ejecución de obra
Nombre del emplazamiento	Superficie:580841
Altitud	508.568
Número de niveles	3
Nivel: 00 Planta baja	0.00
Nivel: 01 Planta primera	3.15

Tabla 3.5.47. Datos del edificio extraídos del modelo IFC

Por último, también es posible obtener el tamaño del archivo IFC directamente desde el sistema operativo, sin necesidad de abrirlo, a partir del módulo OS. De esta manera, se extrae el tamaño en bytes, kilobytes y megabytes, lo que permite conocer el peso del archivo de forma precisa y rápida.

```

import os
# Ruta al archivo IFC
filepath = "Proj_2_ARC_IFC4.ifc"
# Obtener estadísticas del archivo
file_stats = os.stat(filepath)
size_bytes = file_stats.st_size
size_kb = size_bytes / 1024
size_mb = size_bytes / (1024 * 1024)
print(f"Tamaño en megabytes: {round(size_mb, 2)} MB")

```

Tamaño en megabytes: 101.79 MB

3.5.9. Revisiones y chequeos de calidad

Los procesos de calidad se enfocan en asegurar la veracidad y consistencia de la información de la construcción, generando para ello informes en los que se indica la conformidad con respecto a los datos generados en el modelo BIM. Si el informe es desfavorable, se envía al equipo de autoría para su actualización.

Para la aplicación práctica de este caso de uso, los modelos IFC se cargan en entornos como Google Collaboratory o Jupyter Notebook para realizar comprobaciones automatizadas. Principalmente a partir del uso de la librería IfcOpenShell, la cual permite extraer los datos del archivo y compararlos con los requisitos definidos en el contrato. El cumplimiento de estos requisitos se verifica con plantillas previamente extraídas de la base de datos alojada en MariaDB, atendiendo siempre a diferentes casuísticas como la versión de IFC, fases, usos BIM y disciplinas implicadas.

Es importante señalar que, aunque existen muchos tipos de revisiones, las del presente estudio se centran mayormente en la información alfanumérica del archivo IFC, excluyendo comprobaciones de geometría o cumplimiento de normativas técnicas avanzadas. Por tanto, se abordan comprobaciones relacionadas con estructuras de datos, nomenclaturas, relaciones entre entidades, sistemas de clasificación y definición de atributos, propiedades y cantidades. Además, se realizan también otras comprobaciones relacionadas con documentación asociados o fuentes externas de información.

En primer lugar, se abordan las revisiones relacionadas con sistemas de clasificación. Estas comprobaciones garantizan que los objetos tengan rellenos los valores tanto del código como de la descripción en base a la tabla del sistema que haya sido requerida, definidos como *IfcClassification*. Además de esto, debe existir una comprobación que determine si código y descripción se corresponden según dicha tabla. Para realizar este control, un procedimiento semiautomático es a partir de la extracción de los datos de las clasificaciones de los elementos provenientes del archivo IFC a un archivo XLSX en el que, mediante una serie de fórmulas, se verifique si los datos se han definido de la manera correcta.

Tomando como base la tabla de Funciones del SCFclass de Railway Innovation Hub en su versión 2, se aplican las siguientes fórmulas en el modelo de arquitectura del complejo edificatorio del nivel de análisis 4, el cual contiene los códigos y descripciones definidas. En una hoja del archivo, se encuentra la tabla de Funciones, mientras que en la siguiente se añaden los registros de todas las instancias del modelo BIM, con sus datos de clasificación y otros datos relevantes como el nombre y el GUID.

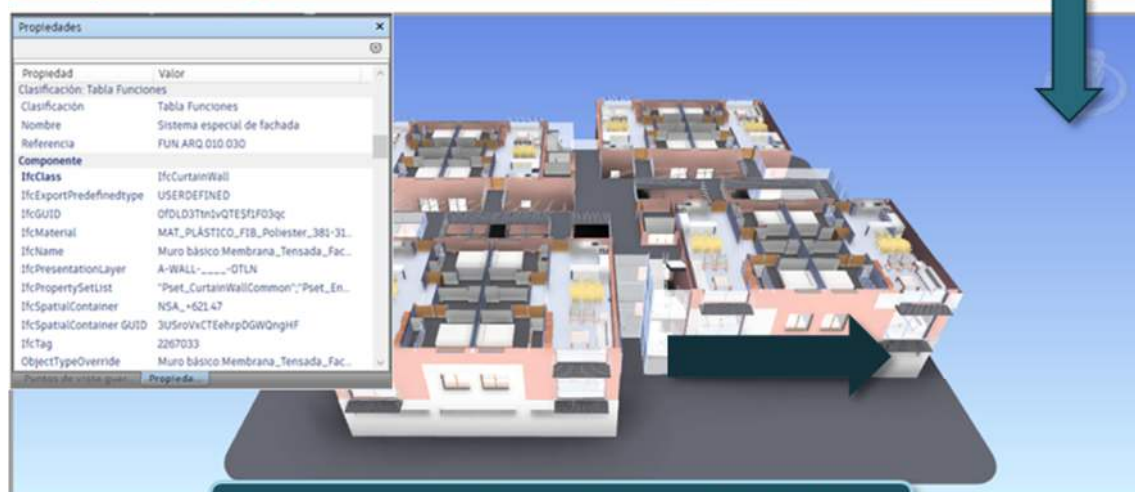
- Descripción: busca si un valor existe en la columna Descripción de la tabla Funciones del sistema de clasificación. Si lo encuentra, devuelve "SI"; si no lo encuentra, o hay un error en la búsqueda, devuelve "NO".

=SI.ERROR(SI(COINCIDIR(E2, FUNCIONES[[#Todo],[DESCRIPCIÓN]],0)>0,"SI","NO"),"NO")

- Código: comprueba si el Código está relleno y si coincide con el que corresponde a la Descripción en la tabla Funciones. Si coincide, devuelve "SI"; si no, o si hay un error, devuelve "NO".

Tabla de Funciones del sistema de clasificación

Tabla de Funciones del sistema de clasificación									
Nombre:									
Versión:									
Descripción:	Tabla estándar para clasificar elementos BIM según la función que desempeñan.								
NIVEL 2	NIVEL 3	NIVEL 4	CÓDIGO	DESCRIPCIÓN	AD MEN	ENTIDAD IFC	TIPO PREDEFINIDO IFC		
			FUN	Tipo de Función	Unidad	No aplica			
ENT			FUN.ENT	Entorno		No aplica			
ENT	010		FUN.ENT.010	Símbolos		No aplica			
ENT	010	010	FUN.ENT.010.010	Origen de coordenadas	ud	IfcCartesianPoint	NO TIENE SUBTIPOS		
ENT	010	020	FUN.ENT.010.020	Red o malla	ud	IfcGrid	USERDEFINED		
ENT	020		FUN.ENT.020	Áreas arboladas		No aplica			
ENT	020	010	FUN.ENT.020.010	Árbol	ud	IfcGeographicElement	VEGETATION		
ENT	020	020	FUN.ENT.020.020	Arbusto	ud	IfcGeographicElement	VEGETATION		



Extracción de datos del modelo IFC

Componente		Componente	Clasificación: Tabla Funciones	Clasificación: Tabla Funciones	Componente	Componente
IfoGUID		IfoName	Referencia	Nombre	IfoClass	IfoExportDefiniedTyp
		2f0f3A2Lnl6k4x4zN330	Barandilla: 60.20.10.80:1100mm:804427	FUN_Arq_020_250	Barandilla	USERDEFINED
		16f30125590nmqN4Zu8Zi	Barandilla: 60.20.10.80:Barandilla-Generico-Barandilla	FUN_Arq_020_250	Barandilla	USERDEFINED
		16f30125590nmqN4Zu8Vq	Barandilla: 60.20.10.80:Barandilla-Generico-Barandilla	FUN_Arq_020_250	Barandilla	USERDEFINED
		16f30125590nmqN4Zu8Y4	Barandilla: 60.20.10.80:Barandilla-Generico-Barandilla	FUN_Arq_020_250	Barandilla	USERDEFINED
		16f30125590nmqN4Zu8YK	Barandilla: 60.20.10.80:Barandilla-Generico-Barandilla	FUN_Arq_020_250	Barandilla	USERDEFINED
		16f30125590nmqN4Zu8Xa	Barandilla: 60.20.10.80:Barandilla-Generico-Barandilla	FUN_Arq_020_250	Barandilla	USERDEFINED
		16f30125590nmqN4Zu8Xq	Barandilla: 60.20.10.80:Barandilla-Generico-Barandilla	FUN_Arq_020_250	Barandilla	USERDEFINED
		16f30125590nmqN4Zu8VW	Barandilla: 60.20.10.80:Barandilla-Generico-Barandilla	FUN_Arq_020_250	Barandilla	USERDEFINED
		16f30125590nmqN4Zu8GU	Barandilla: 60.20.10.80:Barandilla-Generico-Barandilla	FUN_Arq_020_250	Barandilla	USERDEFINED
		2nnTm_CX8785wLPFm1N	Barandilla: 60.20.10.80:Barandilla-Generico-Barandilla	FUN_Arq_020_250	Barandilla	USERDEFINED
		30PmK78LZU8082197FA54	ARQ_MOB_Solardo Diámetro 150 mm:787129	FUN_VIO_060_070	Solardo	USERDEFINED
					IfoBuildingElementProxy	USERDEFINED

Comprobación de clasificaciones IFC

Código Funcion	Descripción Funcion	Entidad IFC	IFC	Descripción de clasificación	Código de clasificación	Clase IFC	Tipo predefinido
FUN.ARQ.020.250	Barandilla	IfcRailing	USERDEFINED	SI	SI	SI	SI
FUN.ARQ.020.050	Protección interior	IfcRailing	NOTDEFINED	SI	SI	SI	NO
FUN.ARQ.020.180	Acabado tramo interior	IfcWall	NOTDEFINED	SI	SI	NO	NO
FUN.ARQ.020.050	Protección interior	IfcRailing	NOTDEFINED	SI	SI	SI	NO
FUN.ARO.010.030	Sistema especial de fachada	IfcCurtainWall	USERDEFINED	SI	SI	SI	SI

Además de los datos del código y la descripción, en muchas clasificaciones existen clases y tipos predefinidos que han sido establecidos para todos los tipos de objetos según el código. Por este motivo, también se verifica si estos datos han sido definidos según corresponda en cada caso. En este sentido, es importante revisar en qué versión de IFC se ha generado el archivo, ya que pueden existir diferencias entre unas y otras. En este caso, el archivo se ha generado como IFC 4, con la Model View Definition de Design Transfer View.

- Clase IFC: comprueba si existe una fila en la tabla FUNCIONES que tenga la misma descripción que la Descripción y la misma clase IFC. Si encuentra esa combinación, devuelve "SI"; si no, "NO".

$=SI(SUMAPRODUCTO((FUNCIONES[DESCRIPCIÓN]=E2)*(FUNCIONES[CLASE IFC]=F2))>0, "SI", "NO")$

- Tipo predefinido IFC: analiza si existe una fila en la tabla FUNCIONES donde la descripción coincida con la Descripción y el tipo predefinido IFC coincida con el tipo correspondiente. Si encuentra esa combinación, devuelve "SI"; si no, "NO".

$=SI(SUMAPRODUCTO((FUNCIONES[DESCRIPCIÓN]=E2)*(FUNCIONES[TIPO PREDEFINIDO IFC]=G2))>0, "SI", "NO")$

Una vez realizada la revisión, los elementos que no están definidos correctamente deben ser modificados por parte del equipo de modelado. Para realizar este proceso de manera automatizada, se crea una rutina de Dynamo que genera una nueva vista 3D en Revit que colorea en rojo los elementos cuyo parámetro *IfcGUID* coincide con la lista de identificadores filtrados del Excel. Para ello, se accede al modelo, se buscan todos los elementos con el parámetro *IfcGUID* de la lista y se crea una vista 3D isométrica con una configuración de ajustes gráficos con color rojo y patrón de relleno sólido para aplicar a los elementos.



Figura 3.5.67. Rutina para generar vista 3D con objetos incorrectos

```
import clr
clr.AddReference('RevitServices')
clr.AddReference('RevitAPI')
clr.AddReference('RevitNodes')
from RevitServices.Persistence import DocumentManager
from Autodesk.Revit.DB import (
    FilteredElementCollector, View3D, OverrideGraphicSettings,
    ElementId, ViewFamilyType, ViewFamily, Transaction,
    Color, BuiltInParameter, Parameter, FillPatternElement)
#Obtener el modelo BIM actual
```



```

doc = DocumentManager.Instance.CurrentDBDocument
uidoc = DocumentManager.Instance.CurrentUIApplication.ActiveUIDocument
#Entrada: lista de IfcGUIDs desde Excel
ifc_guids = [str(g).strip() for g in IN[0]]
#Función para generar nombre único para vista
def nombre_vista_unico(base):
    i = 1
    nombre = base
    nombres_existentes = [v.Name for v in
FilteredElementCollector(doc).OfClass(View3D)]
    while nombre in nombres_existentes:
        nombre = f"{base} {i}"
        i += 1
    return nombre
#Crear vista 3D nueva
collector = FilteredElementCollector(doc).OfClass(ViewFamilyType)
view3DType = None
for v in collector:
    if v.ViewFamily == ViewFamily.ThreeDimensional:
        view3DType = v
        break
newView = None
if view3DType:
    t = Transaction(doc, "Crear vista 3D con color")
    t.Start()
    newView = View3D.CreateIsometric(doc, view3DType.Id)
    newView.Name = nombre_vista_unico("Vista 3D IFC")
    t.Commit()
#Buscar elementos con IfcGUID coincidente
matched_elements = []
all_elements = FilteredElementCollector(doc).
WhereElementIsNotElementType().ToElements()
for el in all_elements:
    param = el.LookupParameter("IfcGUID")
    if param and param.HasValue:
        valor = param.AsString()
        if valor and valor.strip() in ifc_guids:
            matched_elements.append(el)
#Modificación de gráficos
color = Color(255, 0, 0)
ogs = OverrideGraphicSettings()
ogs.SetProjectionLineColor(color)
ogs.SetSurfaceForegroundPatternColor(color)
#Buscar patrón de relleno sólido
solid_fill = None
for fp in FilteredElementCollector(doc).OfClass(FillElement).ToElements():
    if fp.GetFillPattern().IsSolidFill:
        solid_fill = fp
        break

```



```

if solid_fill:
    ogs.SetSurfaceForegroundPatternId(solid_fill.Id)
#Aplicar gráficos a los elementos en la nueva vista
if newView:
    t = Transaction(doc, "Colorear elementos IFC")
    t.Start()
    for el in matched_elements:
        newView.SetElementOverrides(el.Id, ogs)
    t.Commit()
#Salida
OUT = [matched_elements, newView]

```

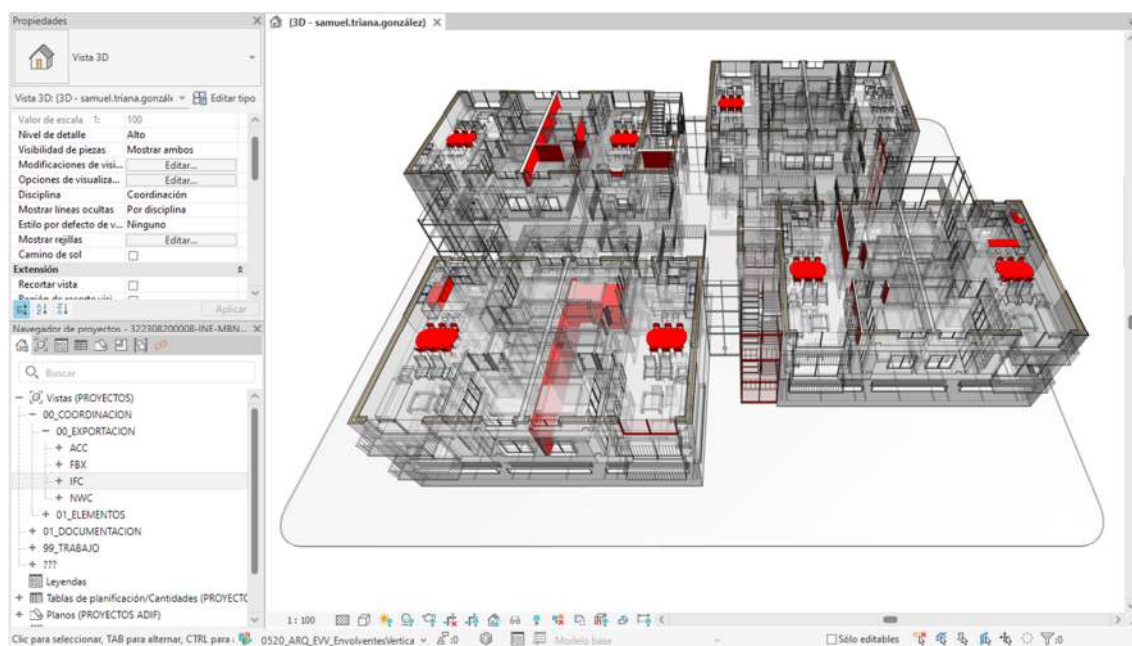


Figura 3.5.68. Nueva vista 3D con objetos incorrectos en rojo

Las revisiones de calidad de modelos BIM también pueden ser abordadas a partir de la ejecución de archivos IDS que contengan diferentes especificaciones para el cumplimiento de los requisitos del proyecto. Existen actualmente diferentes herramientas y aplicaciones como BIMCollab o BIMVision que permiten trabajar directamente con estas especificaciones. No obstante, también es posible el trabajo avanzado para la creación y gestión del archivo para así poder conectarlo con los datos provenientes de la base relacional.

Partiendo de los datos obtenidos de MariaDB con todas las definiciones de los requisitos de datos, se genera un código que implementa un flujo automatizado para generar el IDS basado en las propiedades que han sido definidas en una hoja de cálculo. El desarrollo, en este caso, comienza leyendo una serie de propiedades asociadas a la entidad IfcColumn para seguidamente realizar un proceso de traducción desde datos tabulares hacia a un formato estructurado y legible por máquina.

Una vez que los datos son cargados en memoria como un DataFrame, el código agrupa las filas por la entidad IFC a la que se refieren. En este caso solo se definen

especificaciones para pilares, pero con el código se podrían añadir otras definiciones diferentes. Para dicha entidad, se construye una instancia del objeto *Specification*, que encapsula al requisito. A nivel de esquema, este método sigue una estrategia de instanciación basada en clases y propiedades, donde cada especificación contiene su lógica de aplicabilidad y restricciones.

A continuación, cada propiedad se transforma en un objeto *Property* que contiene tanto atributos semánticos, como nombre, tipo de dato, conjunto de propiedades, como operacionales, como instrucciones, ocurrencias mínimas y máximas. Estos objetos son agregados a las especificaciones respectivas, lo que permite que el IDS final contenga reglas que son directamente vinculables a un modelo IFC, generando una validación automatizada mediante herramientas de verificación. Al finalizar la definición de las especificaciones, el IDS completo se exporta como un archivo *.ids* estructurado conforme al esquema oficial de buildingSMART.

```
import pandas as pd
import ids
from reporter import Console
# Leer archivo Excel con encabezados en español
df = pd.read_excel("propiedades_ifc.xlsx")
# Crear IDS
mi_ids = ids.Ids(
    title="Requisitos IFC por entidad",
    description="IDS de validación de propiedades específicas según la entidad IFC,
    leído desde la base de datos de la organización.",
    copyright="STG_Investigación",
    version="1.0",
    author="samueltrianagonzalez",
    date="2024-05-28",
    purpose="coordinación",
    milestone="Diseño esquemático")
# Agrupar por entidad IFC
for entidad, grupo in df.groupby("entidad IFC"):
    especificacion = ids.Specification(
        name=f"Requisitos para {entidad}",
        identifier=f"REQ-{entidad}",
        ifcVersion="IFC4X3_ADD2",
        description=f"Propiedades requeridas para la clase {entidad}",
        instructions=f"Estas propiedades deben completarse para todas las instancias
    de {entidad}.")
    especificacion.applicability.append(ids.Entity(name=entidad))
# Agregar propiedades
for _, row in grupo.iterrows():
    propiedad = ids.Property(
        name=row["nombre propiedad"],
        propertySet=row["conjunto propiedades"],
        datatype=row["tipo de dato IFC"],
        value=row["valor esperado"],
        uri=row["URI de la propiedad"],
```

```

instructions=row["instrucciones"],
minOccurs=1,
maxOccurs="unbounded")
especificacion.requirements.append(propiedad)
mi_ids.specifications.append(especificacion)
# Exportar a XML
mi_ids.to_xml("IDS_desde_datos_BBDD.ids")

```

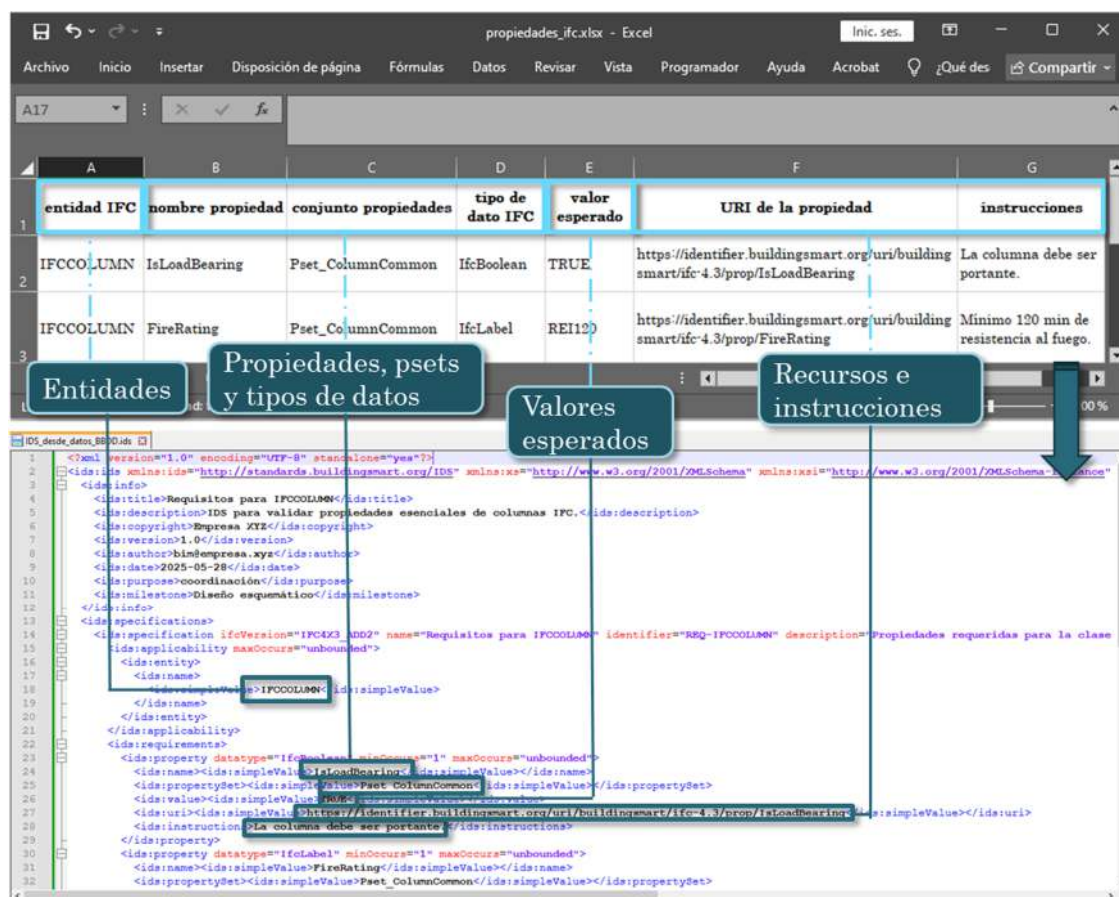


Figura 3.5.69. Creación de especificaciones IDS

Es importante destacar que, para análisis más complejos en casos de proyectos reales, esta aplicación requeriría de un enfoque más pormenorizado, ya que es posible que los requisitos contengan múltiples tipos de datos que incluyen relaciones entre objetos, materiales específicos, sistemas de clasificación, jerarquías de elementos y flujos de trabajo complejos que también dependen de las versiones IFC. Además, una misma especificación dentro del IDS puede contener varias reglas o requisitos simultáneos. Por ejemplo, no solo se puede exigir que una columna tenga una propiedad de resistencia al fuego, sino también que esté correctamente relacionada con una estructura de soporte, que pertenezca a un sistema de clasificación particular y que cumpla con determinadas reglas de nomenclatura o etiquetado. Esto hace que la lógica de validación tenga que contemplar escenarios múltiples, validando desde valores estáticos hasta reglas dinámicas que involucren patrones y relaciones.

Asimismo, en muchos casos, los valores dentro de las especificaciones no son fijos, sino que pueden definirse mediante patrones o comodines, dependiendo esto también del software o herramienta en la que se importen. Estos patrones permiten validar reglas como la longitud máxima o mínima de un texto, la estructura de un código alfanumérico, la inclusión de símbolos específicos o la conformidad con expresiones regulares para asegurar la consistencia y calidad de los datos.

Por último, la generación de un IDS no solo se limita a definir qué propiedades o atributos deben estar presentes, sino también a describir con claridad el propósito de cada requisito, quién es responsable de proporcionarlo, en qué etapa del proyecto se debe cumplir y cómo debe verificarse. Esto hace que el IDS sea una herramienta para validación automática y, a la vez, sirva para la comunicación efectiva entre todos los participantes, asegurando que el modelo BIM sea útil, preciso y acorde a las expectativas del proyecto.

En cualquier caso, la plantilla generada a partir del código puede ser ejecutada a partir de una nueva función, utilizando la definición *ids.validate(Proj_03_ARQ.ifc)* o bien puede ser importada dentro de aplicaciones como BIMCollab Zoom a partir de su plataforma web, para luego ser añadidos y ejecutados en el proyecto.

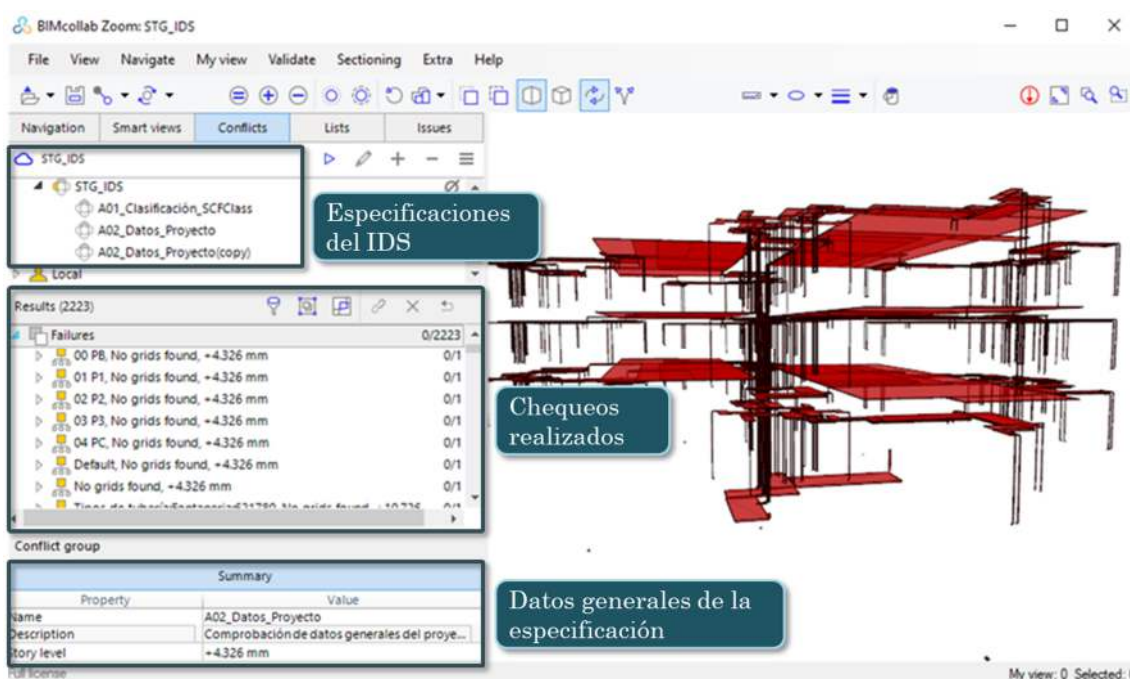


Figura 3.5.70. Ejecución de chequeos de especificaciones en BIMCollab

Si bien las estructuras XML e IDS no tienen a priori limitaciones en cuanto al número máximo de valores permitidos, existe una limitación en múltiples softwares que imposibilitan la definición de, por ejemplo, todos los códigos de una clasificación, ya que existe un número máximo y no hay posibilidad de referenciar a tablas o fuentes externas. Por este motivo, sumado a que es muy complicado establecer asociaciones de relación dependientes entre valores predefinidos de diferentes propiedades y también a las restricciones relacionadas con definir reglas de fórmulas condicionales avanzadas, se puede recurrir a la generación de otras comprobaciones complementarias al IDS a partir de automatizaciones y macros para revisar los datos de los modelos en referencia a tablas de datos con valores predefinidos.

Otra forma de automatizar los procesos de validación y control de los datos de los modelos BIM, es a partir de automatizaciones personalizadas que toman los valores desde archivos estructurados y comprueban dentro del IFC si la información solicitada está correctamente definida. Un caso de aplicación de este proceso es el uso de archivo JSON, el cual puede ser extraído directamente desde la BBDD con la información de la especificación y un código Python que revise la información del modelo IFC, en este caso específico, conjuntos de cantidades de las instancias.

```
#Datos de entrada IFC 4 y JSON
jsonfile =
pd.json_normalize(pd.read_json("Test_Properties.json")['Revisiones']).values.tolist()
filenm = 'Proj_04_MEP.ifc'
ifcfile = ifcopenshell.open(filenm)
lst, lst1, lst2, vls1, vls2, vls3, elems = [], [], [], [], [], [], []
for i in jsonfile:
    vls1.append(i[0])
    vls2.append(i[1])
    vls3.append(i[2])
print("\033[1m" + 'Revisión del modelo ' + filenm + '!\n')
#Instancias que contienen Qtos
for i, u in zip(vls1, vls3):
    elems = ifcfile.by_type(i)
    for e in elems:
        lst1.append(e.Name)
        inst = []
        atribs = [e.GlobalId, e.PredefinedType, e.ObjectType, e.Name]
        inst.extend(atribs)
        for relps in e.IsDefinedBy:
            if relps.is_a() == 'IfcRelDefinesByProperties':
                pset = relps.RelatingPropertyDefinition
                if pset.is_a() == vls2[0]:
                    lst2.append(e.Name)
                    for q in pset.Quantities:
                        dt = []
                        if q.Name in u:
                            vl = round(q[3],2)
                            dt.append(vl)
                        else:
                            dt.append('null')
                    inst.append(dt[0])
        lst.append(inst)
#Dataframe con instancias que no contienen Qtos
wrng = list(set(lst1).symmetric_difference(set(lst2)))
print("\033[1m" + i + '-----\n\n'
      + f'El total de instancias de ' + i + ' es ' + str(len(lst1)) + '!\n')
if len(wrng) == 0:
    print("\033[1m" + 'Todas las instancias de ' + i + ' contienen el pset ' + vls2[0] +
    ':')
else:
```



```

    print('\033[1m' + 'Se han encontrado ' + str(len(wrng)) + ' instancias de ' + i + '
que no contienen el pset ' + vls2[0] + ':')
    df1 = pd.DataFrame(wrng)
    df1.columns = ['Name']
    display(df1)
    #Dataframe con cantidades
    if len(lst) == 0:
        print('\033[1m' + 'No se han encontrado datos de ' + i + ' relativos al pset ' +
vls2[0] + ':')
    else:
        print('\033[1m' + 'Se han encontrado ' + str(len(lst)) + ' instancias de ' + i + '
que contienen datos relativos a ' + vls2[0] + ':')
        df2 = pd.DataFrame(lst)
        u[:0] = ['GUID', 'PredefinedType', 'ObjectType', 'Name']
        df2.columns = u
        df2.set_index('GUID', inplace=True)
        df2.index.name = None
        df2.sort_values(['PredefinedType', 'ObjectType'], ascending=[True, False],
inplace=True)
        display(df2)
        #Dataframe con agrupaciones de cantidades
        print('\033[1m' + 'Los valores agrupados que se obtienen por tipos son:')
        df3 = df2.drop(columns=['Name', 'Width'])
        df3.reset_index(drop=True, inplace=True)
        dfx = df3.groupby(['PredefinedType', 'ObjectType'], as_index=False).value_counts()
        dfx2 = dfx.groupby(['PredefinedType', 'ObjectType'], as_index=False).sum()
        dfx2.loc["Total: "] = dfx2.sum(numeric_only=True, axis = 0)
        display(dfx2)
        b_lsts = [elems, inst, atribs, dt, lst, lst1, lst2]
        for b in b_lsts:
            b.clear()

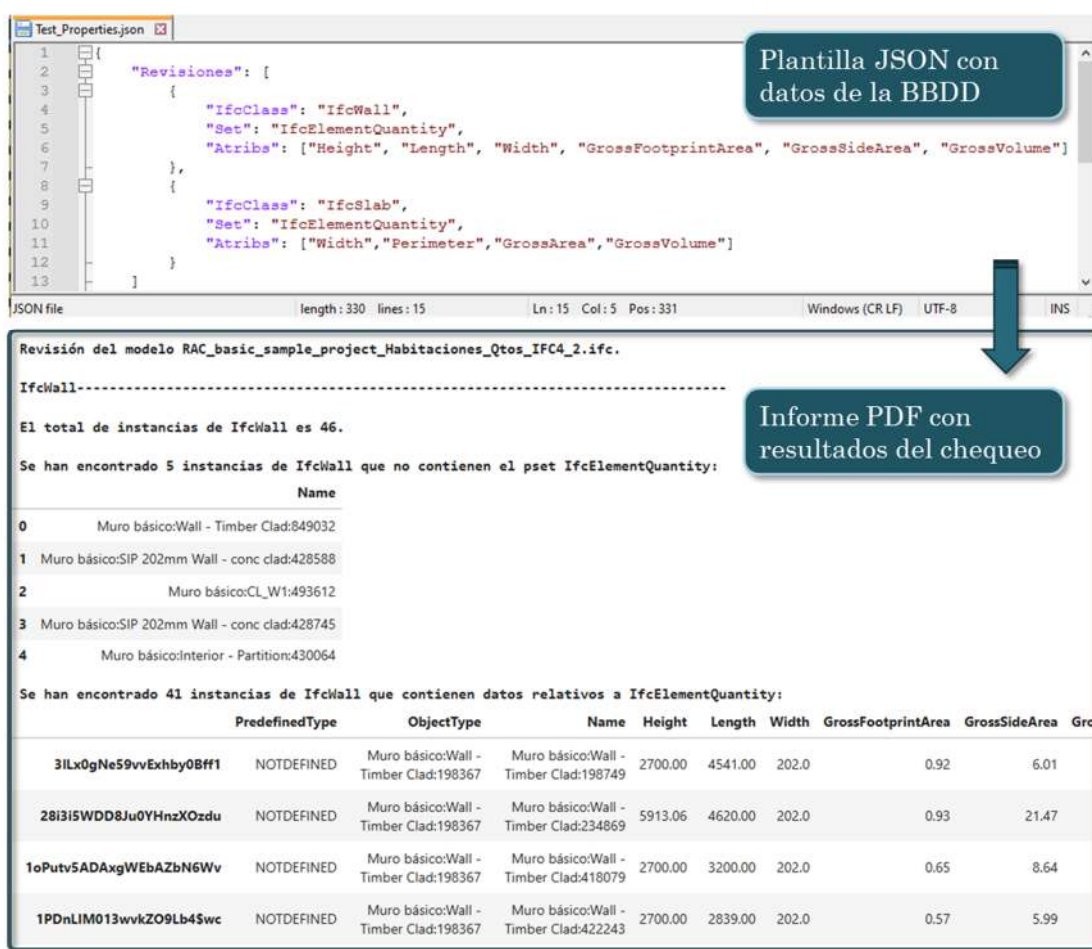
```

Una vez que finaliza el chequeo de comprobación, también puede exportarse el archivo a diferentes formatos, como PDF para enviarlo a los equipos que corresponda y que así puedan realizar las modificaciones dentro de los modelos nativos.

```

#Exportación de resultados
def convert_notebook_to_pdf(notebook_path, output_pdf_path):
    with open(notebook_path, 'r', encoding='utf-8') as f:
        nb = nbformat.read(f, as_version=4)
    #Configurar el exportador PDF y asignar el preprocesador
    pdf_exporter = PDFExporter()
    pdf_exporter.exclude_input = True
    pdf_exporter.preprocessors = [tag_remove_preprocessor]
    # Convertir y guardar a PDF
    pdf_data, _ = pdf_exporter.from_notebook_node(nb)
    with open(output_pdf_path, 'wb') as f:
        f.write(pdf_data)
    #Usar la función para convertir el cuaderno actual
    convert_notebook_to_pdf('STG_Especificaciones.ipynb', 'STG_Informe.pdf')

```



Plantilla JSON con datos de la BBDD

```

1 {
2   "Revisiones": [
3     {
4       "IfcClass": "IfcWall",
5       "Set": "IfcElementQuantity",
6       "Atribs": ["Height", "Length", "Width", "GrossFootprintArea", "GrossSideArea", "GrossVolume"]
7     },
8     {
9       "IfcClass": "IfcSlab",
10      "Set": "IfcElementQuantity",
11      "Atribs": ["Width", "Perimeter", "GrossArea", "GrossVolume"]
12    }
13  ]
14 }
  
```

Informe PDF con resultados del chequeo

Revisión del modelo RAC_basico_sample_project_Habitaciones_Qtos_IFC4_2.ifc.

IfcWall-----

El total de instancias de IfcWall es 46.

Se han encontrado 5 instancias de IfcWall que no contienen el pset IfcElementQuantity:

	Name
0	Muro básico:Wall - Timber Clad:849032
1	Muro básico:SIP 202mm Wall - conc clad:428588
2	Muro básico:CL_W1:493612
3	Muro básico:SIP 202mm Wall - conc clad:428745
4	Muro básico:Interior - Partition:430064

Se han encontrado 41 instancias de IfcWall que contienen datos relativos a IfcElementQuantity:

	PredefinedType	ObjectType	Name	Height	Length	Width	GrossFootprintArea	GrossSideArea	GrossVolume
3ILx0gNe59vvExhby0Bff1	NOTDEFINED	Muro básico:Wall - Timber Clad:198367	Muro básico:Wall - Timber Clad:198749	2700.00	4541.00	202.0	0.92	6.01	
28i3i5WDD8Ju0YHnzXOzdu	NOTDEFINED	Muro básico:Wall - Timber Clad:198367	Muro básico:Wall - Timber Clad:234869	5913.06	4620.00	202.0	0.93	21.47	
1oPutv5ADAgWEbAZbN6Wv	NOTDEFINED	Muro básico:Wall - Timber Clad:198367	Muro básico:Wall - Timber Clad:418079	2700.00	3200.00	202.0	0.65	8.64	
1PDnLIM013wvkZO9Lb4\$wc	NOTDEFINED	Muro básico:Wall - Timber Clad:198367	Muro básico:Wall - Timber Clad:422243	2700.00	2839.00	202.0	0.57	5.99	

Figura 3.5.71. Exportación de informe PDF con revisiones BIM

Aunque el registro por medio de documentos tipo PDF es importante, también es de gran utilidad el uso y trabajo de archivos BCF que permitan ser luego importados en los softwares de autoría y que rápidamente los técnicos y modeladores puedan ver las instancias que no cumplen con cada especificación. Para ello, se pueden generar nuevas incidencias dentro del archivo de comunicación en las que se definan los GUIDS de los elementos afectados. A continuación, se muestra un código de ejemplo:

```

from bcf import bcfxml
from bcf.bcfxml.topic import BcfTopic
from bcf.bcfxml.comment import BcfComment
from bcf.bcfxml.comment import BcfComponent
from datetime import datetime, timedelta
import uuid

#BCF
archivo_bcf = bcfxml.load("STG_Proj_04.bcf")
#Modificar proyecto
archivo_bcf.project.name = "Coordinación elementos de construcción"
archivo_bcf.edit_project()
#Definir responsables y prioridades simuladas
responsables = ["usuario01", "usuario02", "usuario03"]
prioridades = ["Low", "Medium", "High"]
  
```

```

#Lista de dos incidencias simuladas
incidencias = [
    {
        "titulo": "Conflicto entre bandejas eléctricas y conductos ", "descripcion": (
            "Se ha detectado una colisión entre bandejas portacables y conductos"
            "en el pasillo técnico del nivel 1. Requiere coordinación entre disciplinas.")
        "guids_objetos": [str(uuid.uuid4()), str(uuid.uuid4())] },
    {
        "titulo": "Cota de ventana fuera de especificación", "descripcion": (
            "La altura de instalación de una ventana en fachada no cumple requisitos "
            "de accesibilidad. Es necesario revisar los planos de carpintería.")}]

#Crear y agregar las dos incidencias simuladas
for i, incidencia in enumerate(incidencias, start=1):
    tema = BcfTopic(
        title=incidencia["titulo"],
        priority=random.choice(prioridades),
        index=i,
        assigned_to=random.choice(responsables),
        description=incidencia["descripcion"],
        status="Open")

# Asignar componentes (objetos) al tema
for guid_objeto in incidencia["guids_objetos"]:
    componente = BcfComponent(
        ifc_guid=guid_objeto,
        originating_system="STG",
        selection=True )
    tema.add_component(componente)

#Fecha de creación del comentario
fecha_base = datetime.now()

#Agregar el tema al archivo BCF
archivo_bcf.add_topic(tema)

#Guardar el archivo resultante
archivo_bcf.save("STG_incidencias.bcf")

```

Además, gracias a otras librerías y definiciones, como OCC y JupyterRenderer, es posible leer las geometrías de los objetos del informe y abrirlos en un visor online en entornos como Jupyter o Google Colab. Desde estos visores se genera una captura para incorporarla en el archivo de revisión. Para el desarrollo del caso, se genera un código que abre un archivo IFC para procesar y visualizar únicamente aquellos muros que no cumplen con una determinada especificación relacionada con su espesor. Se recorren todos los elementos de la clase muro en el modelo, y para cada uno se verifica si dispone de una propiedad que indica su espesor. Aquellos cuyo valor es inferior al umbral establecido, se seleccionan para su representación gráfica en un visor 3D, donde se destacan en color rojo para facilitar su identificación visual. Además, se implementa una funcionalidad interactiva que permite mostrar las propiedades del muro seleccionado en pantalla, proporcionando información adicional sobre el elemento mediante una interfaz de texto.

```

import ifcopenshell
import ifcopenshell.geom
from OCC.Display.WebGl.jupyter_renderer import JupyterRenderer

```

```

import ipywidgets as widgets
from IPython.display import display
#Abrir archivo IFC
ifc_file = ifcopenshell.open('Proj_04_STR.ifc')
#Inicializar visor 3D
renderer = JupyterRenderer(size=(800, 700))
settings = ifcopenshell.geom.settings()
settings.set(settings.USE_PYTHON_OPENCASCADE, True)
#Mapa de relación entre forma y producto IFC
shape_to_product = {}
#Mostrar solo muros que incumplen espesor mínimo de 200 mm
for wall in ifc_file.by_type("IfcWall"):
    if not wall.Representation:
        continue
    #Buscar propiedad de espesor
    espesor = None
    for rel in wall.IsDefinedBy:
        if rel.RelatingPropertyDefinition.is_a("IfcPropertySet"):
            for prop in rel.RelatingPropertyDefinition.HasProperties:
                if prop.is_a("IfcPropertySingleValue") and "espesor" in prop.Name.lower():
                    espesor = prop.NominalValue.wrappedValue
                    break
    #Requisito de información
    if espesor is not None and espesor < 200:
        shape = ifcopenshell.geom.create_shape(settings, wall)
        shape_id = renderer.DisplayShape(shape.geometry, color=(1.0, 0.0, 0.0),
        transparency=False) # Rojo
        shape_to_product[shape_id] = wall
#Crear widget para mostrar propiedades al seleccionar
info_box = widgets.Textarea(
    value="",
    placeholder='Selecciona un objeto...',
    description='Propiedades:',
    layout={'width': '100%', 'height': '200px'})
display(info_box)
#Formatear propiedades IFC
def format_ifc_properties(product):
    props = []
    for rel in product.IsDefinedBy:
        if rel.RelatingPropertyDefinition.is_a("IfcPropertySet"):
            for prop in rel.RelatingPropertyDefinition.HasProperties:
                if prop.is_a("IfcPropertySingleValue"):
                    props.append(f'{prop.Name}: {prop.NominalValue.wrappedValue}')
    return "\n".join(props)
#Llamada para selección de objetos
def on_select(event):
    shape_id = event['shape']
    product = shape_to_product.get(shape_id)
    if product:

```

```

info_box.value = format_ifc_properties(product)
else:
    info_box.value = "Sin información disponible."
renderer.register_select_callback(on_select)
#Mostrar visor 3D
display(renderer)

```



Figura 3.5.72. Visor de geometría desde Jupyter Notebook

Finalmente, utilizando los archivos IDS o bien generando automatizaciones de revisión, el BCF puede ser visualizado en programas como BIMCollab en donde puede más tarde llevarse un control y seguimiento de todos los asuntos que vayan siendo generados en los diferentes hitos de entrega del proyecto.

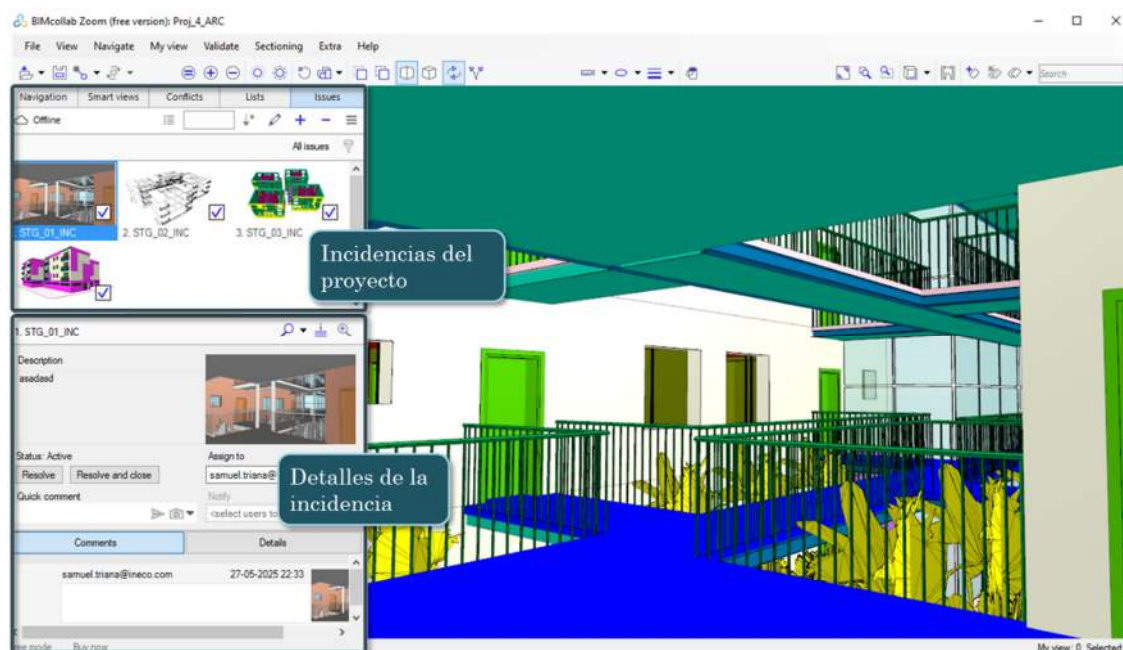


Figura 3.5.73. Gestión BCF dentro de BIMCollab Zoom

Otras comprobaciones IFC más avanzadas que relacionen los datos con otras fuentes externas pueden ser desarrolladas desde Dynamo. En este entorno los archivos openBIM pueden ser cargados y su información puede ser extraída con diferentes nodos y fórmulas de gestión de datos alfanuméricos.

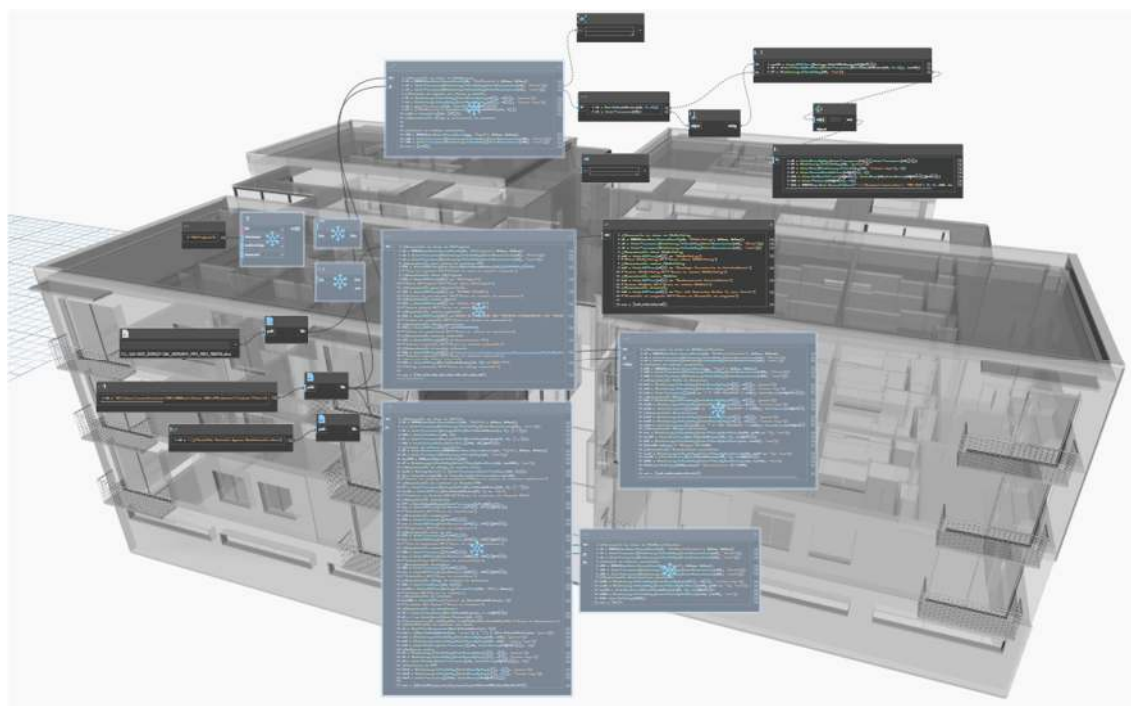


Figura 3.5.74. Automatización en la validación IFC con Dynamo

En este caso, se desarrolla una nueva rutina compuesta por diferentes grupos que verifican la información de diferentes clases y propiedades IFC. En primer lugar, se realiza una lectura estructurada de los datos provenientes de una hoja Excel que contiene información de elementos IFC, separando encabezados y registros, y permitiendo su manipulación como listas. A través de operaciones de agrupamiento y conteo, se genera una tabla resumen que cuantifica la presencia de elementos por clase IFC, y calcula el total global de elementos, lo que permite detectar desviaciones o vacíos de modelado. Simultáneamente, el script incorpora una segunda fuente de datos desde otra hoja de Excel, que actúa como conjunto de referencia de valores válidos o permitidos para propiedades clave como códigos o descripciones de producto. Esta estructura modular y extensible permite establecer una base para procesos de validación automática entre datos modelados y catálogos de referencia, reforzando así la trazabilidad y calidad BIM.

```
//Extracción de datos de IfcElements
m1 = DSOOffice.Data.ImportExcel(m0, "IfcElements", false, false);
n1 = List.Transpose(Dictionary.ValueAtKey(List.Deconstruct(m1), "first"));
n2 = List.Transpose(Dictionary.ValueAtKey(List.Deconstruct(m1), "rest"));
//Recuento de objetos por clases y totales
t3 = Dictionary.ValueAtKey(List.GroupByKey(n2[3], n2[3]), "groups");
t4 = Dictionary.ValueAtKey(List.GroupByKey(n2[3], n2[3]), "unique keys");
t6 = List.Transpose([t4, List.Count(t3@L2<1>)]);
t7 = ["IfcElements", List.Count(Get.ValueAtIndex(n2, 3))];
sal1 = List.Join([t6, t7]);
//Comprobación código y descripción de producto
//Listado de valores permitidos
m00 = DSOOffice.Data.ImportExcel(yy, "Hoja2", false, false);
n10 = List.Transpose(Dictionary.ValueAtKey(List.Deconstruct(m00), "first"));
```

```
n20 = List.Transpose(Dictionary.ValueAtKey(List.Deconstruct(m00), "rest"));
out = [sal1];
```

El siguiente fragmento de código en DesignScript realiza una verificación de los datos contenidos en la entidad *IfcProject* del modelo. La finalidad es asegurar que los parámetros críticos del proyecto cumplan con estándares predefinidos, como peso de archivo, sistema de coordenadas, identificación del proyecto, y metadatos administrativos. Se parte de la lectura de la hoja *IfcProject* del Excel de referencia mediante la función *DSOffice.Data.ImportExcel*, descomponiendo sus contenidos en y transponiéndolos para trabajar por columnas. A partir de ahí, el código verifica uno por uno los criterios relevantes del modelo, como que el tamaño de los archivos IFC no exceda los 250 MB, que todos los elementos pertenezcan a la clase *IfcProject*, que el sistema de coordenadas sea ETRS89 UTM-30N, que el nombre global del proyecto y el nombre de actuación sean exactos, que el código de expediente y el entorno estén correctamente definidos, y que la URL de ubicación esté correctamente establecida. Cada comprobación devuelve un mensaje que indica si la condición ha sido satisfecha o si existe un error, agrupando los resultados en una lista final de salida.

```
//Extracción de datos de IfcProject
m1 = DSOffice.Data.ImportExcel(m0, "IfcProject", false, false);
n1 = List.Transpose(Dictionary.ValueAtKey(List.Deconstruct(m1), "first"));
n2 = List.Transpose(Dictionary.ValueAtKey(List.Deconstruct(m1), "rest"));
//Comprobación peso máximo de archivos IFC
bl1 = List.AllFalse(Math.Round(Get.ValueAtIndex(n2,2)/1024000,2)>250)
?"Peso de archivos OK":"Límite de peso de archivo superado";
//Comprobación clase IfcProject
bl2 = List.AllTrue(n2[3] == "IfcProject")
?"Clase IfcProject OK":"Error clase IfcProject";
//Comprobación georreferenciación
bl3 = List.AllTrue(n2[4] == "ETRS89 UTM-30N")
?"Sistema de coordenadas OK":"Error de sistema de coordenadas";
//Comprobación nombre global
bl4 = List.AllTrue(n2[5] == "Atocha Digital")
?"Nombre global OK":"Error en nombre global";
//Comprobación nombre actuación
bl5 = List.AllTrue(n2[6] == "OBRAS DE EJECUCIÓN DEL PROYECTO")
?"Nombre de actuación OK":"Error en Nombre de actuación";
//Comprobación código expediente
bl6 = List.AllTrue(n2[7] == 322308200008)
?"Número de expediente OK":"Error en número de expediente";
//Comprobación entorno actuación
bl7 = List.AllTrue(n2[8] == "Aparcamiento P3")
?"Entorno actuación OK":"Error en entorno actuación";
//Comprobación dirección ubicación
bl8 = List.AllTrue(n2[9] == "https://acc.autodesk.eu/docs/files/projects/b7be7")
?"Entorno actuación OK":"Error en entorno actuación";
//Comprobación código actuación
bl9 = List.AllTrue(Get.ValueAtIndex(n2, 10) == "100-PP3")
?"Código actuación OK":"Error en código actuación";
```

```
out = [bl1,bl2,bl3,bl4,bl5,bl6,bl7,bl8,bl9];
```

A continuación, el siguiente fragmento automatiza la verificación de propiedades del *IfcBuilding*. El proceso importa los datos de la hoja correspondiente, descomponiéndolos y reorganizándolos para facilitar su lectura por índice. Posteriormente, se realizan comprobaciones lógicas para validar que ciertos atributos del modelo IFC, como el nombre del edificio, el emplazamiento y la dirección del proyecto, coincidan con los valores esperados.

```
//Extracción de datos de IfcBuilding
m1 = DSOOffice.Data.ImportExcel(m0, "IfcBuilding", false, false);
n1 = List.Transpose(Dictionary.ValueAtKey(List.Deconstruct(m1), "first"));
n2 = List.Transpose(Dictionary.ValueAtKey(List.Deconstruct(m1), "rest"));
//Comprobación clase IfcBuilding
bb2 = List.AllTrue(n2[5] == "IfcBuilding")
?"Clase IfcBuilding OK":"Error clase IfcBuilding";
//Comprobación nombre IfcBuilding
bb3 = List.AllTrue(n2[0] == "Complejo Ferroviario de Madrid-Atocha")
?"Nombre IfcBuilding OK":"Error en nombre IfcBuilding";
//Comprobación nombre IfcSite
bb4 = List.AllTrue(n2[2] == "Emplazamiento Madrid-Atocha")
?"Nombre IfcSite OK":"Error en nombre IfcSite";
//Comprobación dirección
bb5 = List.AllTrue(n2[3] == "Pza. del Emperador Carlos V, s/n, Madrid")
?"Dirección de proyecto OK":"Error en Dirección de proyecto";
out = [bb2,bb3,bb4,bb5];
```

Otro bloque de código establece una validación de las *IfcClassification* del modelo IFC, comparando los datos exportados desde el modelo con un listado de valores permitidos definidos en otra hoja de Excel. Primero, se importan y transponen los datos para facilitar el acceso a los campos como tabla de funciones, códigos y descripciones. Más tarde, se agrupan estos valores para generar resúmenes que muestren frecuencias de aparición y se detectan entradas no rellenadas. Además, se realiza una validación cruzada entre los códigos y descripciones del modelo y una tabla de valores permitidos, señalando los elementos no conformes. Finalmente, la salida incluye los recuentos por valor único en cada campo, así como listas de códigos o descripciones que no coinciden con los permitidos.

```
//Extracción de datos de IfcClassification
m1 = DSOOffice.Data.ImportExcel(m0, "IfcClassification", false, false);
n1 = List.Transpose(Dictionary.ValueAtKey(List.Deconstruct(m1), "first"));
n2 = List.Transpose(Dictionary.ValueAtKey(List.Deconstruct(m1), "rest"));
//Listado de valores permitidos
m00 = DSOOffice.Data.ImportExcel(yy, "Hoja2", false, false);
n10 = List.Transpose(Dictionary.ValueAtKey(List.Deconstruct(m00), "first"));
n20 = List.Transpose(Dictionary.ValueAtKey(List.Deconstruct(m00), "rest"));
//Comprobación Tabla de funciones
tt5 = Dictionary.ValueAtKey(List.GroupByKey(n2[0], n2[0]), "groups");
tt6 = Dictionary.ValueAtKey(List.GroupByKey(n2[0], n2[0]), "unique keys");
```

```

tt8 = List.Transpose([(tt6 == "" ? "SIN RELLENAR" : tt6),
List.Count(tt5@L2<1>)]);
//Comprobación Códigos
tt9 = Dictionary.ValueAtKey(List.GroupByKey(n2[2], n2[2]), "groups");
tt10 = Dictionary.ValueAtKey(List.GroupByKey(n2[2], n2[2]), "unique keys");
tt11 = List.Transpose([(tt10 == "" ? "SIN RELLENAR" : tt10),
List.Count(tt9@L2<1>)]);
//Comprobación Descripciones
tt12 = Dictionary.ValueAtKey(List.GroupByKey(n2[1], n2[1]), "groups");
tt13 = Dictionary.ValueAtKey(List.GroupByKey(n2[1], n2[1]), "unique keys");
tt14 = List.Transpose([(tt13 == "" ? "SIN RELLENAR" : tt13),
List.Count(tt12@L2<1>)]);
//Comprobación Códigos permitidos
asd = Dictionary.ValueAtKey(List.FilterByBoolMask(tt10, tt10 == ""), "out");
bool2 = List.Contains(Get.ValueAtIndex(n20, 0), asd@L1<1>);
t10 = Dictionary.ValueAtKey(List.FilterByBoolMask(asd, bool2), "out");
fi1=t10==Empty ? "Códigos OK":t10;
//Comprobación Descripciones permitidas
asd1 = Dictionary.ValueAtKey(List.FilterByBoolMask(tt13, tt13 == ""), "out");
bool3 = List.Contains(Get.ValueAtIndex(n20, 1), asd1@L1<1>);
t101 = Dictionary.ValueAtKey(List.FilterByBoolMask(asd1, bool3), "out");
fi2=List.IsEmpty(t101)==true? "Descripciones OK":t101;
out = [tt8,tt11,tt14,fi1,fi2];

```

El último bloque de código se diseña para extraer y validar datos del propio archivo IFC. Inicialmente importa los datos desde un archivo Excel, organiza y transfiere la información para analizar nombres y características de los archivos IFC, descomponiéndolos en elementos individuales para identificar y verificar valores únicos, evitando duplicados. Se valida que todos los archivos tengan extensión .ifc y que los códigos relacionados cumplan con los valores permitidos extraídos también de otra hoja Excel. Además, comprueba que las unidades sean metros y verifica la corrección del sistema de coordenadas del proyecto. Se evalúan las fechas de actualización de los archivos, agrupándolas por mes y año, así como la versión del software nativo utilizado y las versiones IFC presentes. Finalmente, genera un resumen con mensajes de conformidad o error para cada validación, facilitando la revisión y auditoría integral.

```

//Extracción de datos de IfcFile
m1 = DSOOffice.Data.ImportExcel(m0, "IfcFile", false, false);
n = List.Transpose(Dictionary.ValueAtKey(List.Deconstruct(m1), "rest"));
t6 = List.Transpose(String.Split(Get.ValueAtIndex(n, 0), ["-"]));
t10 = List.TakeItems(t6, 5);
t9 = List.Transpose(String.Split(Get.ValueAtIndex(t7, 0), ["_"]));
t12 = List.UniqueItems(List.Join([t10, t9])@L2<1>);
//Listado de valores permitidos
p3 = List.Deconstruct(DSOOffice.Data.ImportExcel(p1, "Hoja1", false, false));
p6 = List.Transpose(Dictionary.ValueAtKey(p3, "rest"));
bool00 = DSCore.Object.IsNull(p6);
p7 = Dictionary.ValueAtKey(List.FilterByBoolMask(p6, bool00), "out");

```



```

//Comprobación de código de archivo único
str0 = String.Join("-", [List.Transpose(List.TakeItems(t6, 6))]);
t2 = List.Count(str0) == List.Count(List.UniqueItems(str0))
?"Nombre de archivos sin duplicar":"Error de nombre de archivos duplicados";
//Comprobación de archivos en formato IFC
t7 = List.Transpose(String.Split(Get.ValueAtIndex(t6, 6), ["."]));
bool1 = List.AllTrue(Get.ValueAtIndex(t7, 1) == "ifc")
?"Archivos en formato IFC OK":"Error de archivos con formato IFC";
//Comprobación expediente
t77 = Get.ValueAtIndex(List.Transpose([p7,t12]),0);
exp = List.AllTrue(List.Contains(t77[0], t77[1]@L1<1>))
?"Expediente OK":"Error de expediente";
//Comprobación originador
t78 = List.Transpose([p7,t12])[1];
ori = List.AllTrue(List.Contains(t78[0], t78[1]@L1<1>))
?"Originador OK":"Error de originador";
//Comprobación tipo de doc
t79 = List.Transpose([p7,t12])[2];
tip = List.AllTrue(List.Contains(t79[0], t79[1]@L1<1>))
?"Tipo de documento OK":"Error de tipo de documento";
//Comprobación especialidad
t80 = List.Transpose([p7,t12])[3];
esp = List.AllTrue(List.Contains(t80[0], t80[1]@L1<1>))
?"Especialidad OK":"Error de especialidad";
//Comprobación disciplina
t81 = List.Transpose([p7,t12])[4];
dis = List.AllTrue(List.Contains(t81[0], t81[1]@L1<1>))
?"Disciplina OK":"Error de disciplina";
//Comprobación código de subtipo de documento
t11 = List.TakeItems(t6, 6)[5];
bool4 = List.AllTrue(String.StartsWith(t11, "522", false))
?"Subtipo OK":"Error de subtipo";
//Comprobación de unidades
bool01 = List.AllTrue("Metros" == Get.ValueAtIndex(n, 1))
?"Unidades OK: Metros":"Error de Unidades";
//Comprobación de coordenadas
c3 = List.UniqueItems(Get.ValueAtIndex(n, 2..4)@L2<1>);
c5 = 3 == List.Count(List.Flatten(c3, -1))
?"Coordenadas OK: Z=622,497m | Y=4473159,541m | X=441232,878m":"Error de
Coordenadas";
//Actualización de archivos por fecha
dt = DateTime.Components(Get.ValueAtIndex(n, 5));
dt2 = ((Get.ValueAtIndex(dt, "month") + "/" + (Get.ValueAtIndex(dt, "year")));
dt4 = Dictionary.ValueAtKey(List.GroupByKey(dt2, dt2), "groups");
dt5 = Dictionary.ValueAtKey(List.GroupByKey(dt2, dt2), "unique keys");
dt8 = List.Flatten(List.Transpose([dt5, List.Count(dt4@L2<1>)]), -1);
//Software nativo
v4 = Dictionary.ValueAtKey(List.GroupByKey(n[6], n[6]), "groups");
v5 = Dictionary.ValueAtKey(List.GroupByKey(n[6], n[6]), "unique keys");

```



```

v8 = List.Flatten(List.Transpose([v5, List.Count(v4@L2<1>)]), -1);
//Versiones de IFC
ifc4 = Dictionary.ValueAtKey(List.GroupByKey(n[7], n[7]), "groups");
ifc5 = Dictionary.ValueAtKey(List.GroupByKey(n[7], n[7]), "unique keys");
ifc7 = List.Transpose([ifc5, List.Count(ifc4@L2<1>)]);
out = [t2,bool1,exp,ori,tip,esp,dis,bool4,bool01,c5,dt8,v8,ifc7];

```

Por otro lado, dentro del entorno de los nativos también pueden realizarse comprobaciones de cumplimiento de requisitos de información. En este contexto, la creación de una plantilla en ModelChecker de Revit puede automatizar la verificación del modelo a través de criterios preestablecidos. El proceso se inicia con la utilización del complementario de configuración, que facilita la estructuración de verificaciones agrupadas por temáticas generales. Este entorno de configuración permite desarrollar plantillas personalizadas adaptadas a las necesidades específicas de cada proyecto o especificaciones de la propia organización.

Durante la configuración de la plantilla, se definen distintos grupos temáticos que organizan las reglas de verificación en categorías lógicas, como datos generales del modelo, aspectos relacionados con la georreferenciación y las coordenadas, atributos específicos de los elementos, así como errores y advertencias originados en el entorno de modelado. Cada uno de estos grupos funciona como una carpeta contenedora de verificaciones, que a su vez puede incluir subgrupos que organizan criterios más específicos dentro del mismo ámbito temático.

Para cada regla definida dentro de la plantilla, se establece un conjunto de parámetros técnicos que incluyen la categoría de elementos del modelo a verificar, los filtros que identifican las condiciones de revisión, y los criterios lógicos que determinan si un ítem cumple o no con lo esperado. Estas condiciones pueden abordar, por ejemplo, la existencia de información, su adecuación formal o su coincidencia con estándares de nomenclatura. Además, se asigna un nivel de severidad al incumplimiento de cada regla, permitiendo clasificar los resultados en advertencias, errores o simples observaciones informativas. Al aplicar la plantilla mediante el complemento, se ejecuta una revisión automatizada que presenta los resultados de forma jerárquica, agrupados según las categorías y subcategorías definidas previamente.

```

import pandas as pd
import xml.etree.ElementTree as ET
import uuid
def generate_id():
    return str(uuid.uuid4())
def create_filter(row):
    if pd.isna(row["FilterCategory"]):
        return None
    return ET.Element("Filter", {
        "ID": generate_id(),
        "Operator": str(row.get("FilterOperator", "And")),
        "Category": str(row.get("FilterCategory", "")),
        "Property": str(row.get("FilterProperty", "")),
        "Condition": str(row.get("FilterCondition", "")),
    })

```

```

    "Value": str(row.get("FilterValue", "")),
    "CaseInsensitive": str(row.get("FilterCaseInsensitive", False)),
    "Unit": str(row.get("FilterUnit", "None")),
    "UnitClass": str(row.get("FilterUnitClass", "None")),
    "FieldTitle": str(row.get("FilterFieldTitle", "")),
    "UserDefined": str(row.get("FilterUserDefined", False)),
    "Validation": str(row.get("FilterValidation", "None"))})
def create_check_element(check_rows):
    base = check_rows.iloc[0]
    check = ET.Element("Check", {
        "ID": generate_id(),
        "CheckName": base["Check Name"],
        "Description": str(base.get("Description", "")),
        "FailureMessage": str(base.get("FailureMessage", "")),
        "ResultCondition": "FailNoElements" if base["CheckType"] == "Custom" else
        "FailMatchingElements",
        "CheckType": base["CheckType"],
        "IsRequired": str(base.get("IsRequired", False)),
        "IsChecked": "True"})
    for _, row in check_rows.iterrows():
        f = create_filter(row)
        if f is not None:
            check.append(f)
    #Opciones de configuración
    if pd.notna(base.get("Prebuilt_FileNamePattern")):
        po = ET.SubElement(check, "PrebuiltOptions")
        po.set("FileNamePattern", str(base["Prebuilt_FileNamePattern"]))
    return check
#Función para generar el archivo
def generate_xml_from_excel(filepath, output_xml_path):
    df = pd.read_excel(filepath)
    root = ET.Element("MCSettings", {
        "AllowRequired": "True",
        "Name": "Manual de entrega de Información requerida (MEI) - v01",
        "Author": "samuel triana gonzalez",
        "Description": "Plantilla de revisión de requisitos comunes de información.",
        "Image": "",
        "LastModified": "133615245848266488"})
    for (heading, section), group in df.groupby(["Heading", "Section"]):
        heading_el = ET.Element("Heading", {
            "ID": generate_id(),
            "HeadingText": heading,
            "Description": "",
            "IsChecked": "True"})
        section_el = ET.SubElement(heading_el, "Section", {
            "ID": generate_id(),
            "SectionName": section,
            "Title": "",
            "IsChecked": "True",

```

```
"Description": ""})  
for check_name, check_group in group.groupby("Check Name"):  
    section_el.append(create_check_element(check_group))  
root.append(heading_el)  
tree = ET.ElementTree(root)  
ET.indent(tree, space=" ", level=0)  
tree.write(output_xml_path, encoding="utf-8", xml_declaration=True)  
generate_xml_from_excel("STG_datos_BBDD.xlsx", "plantilla_STG_MC.xml")
```

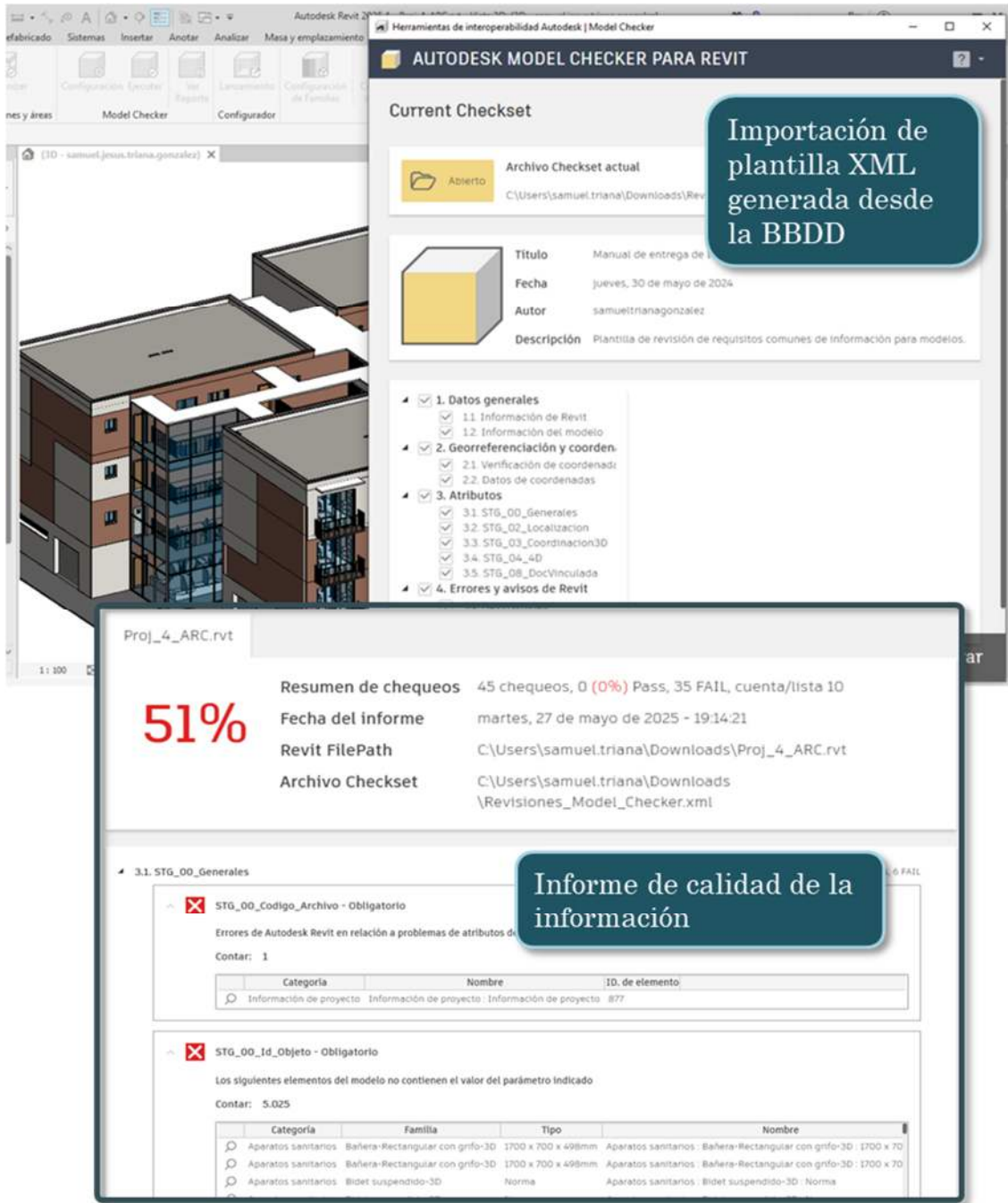


Figura 3.5.75. Informe de calidad desde Model Checker de Revit

Todos estos flujos de trabajo permiten establecer una metodología sistemática y reproducible para la validación de modelos BIM en función de requisitos previamente definidos y estructurados. La integración de formatos interoperables como IDS, junto con herramientas de chequeo automatizado y gestión colaborativa de incidencias, garantiza una verificación eficiente, coherente y trazable de los proyectos de edificación.

Además, las automatizaciones desarrolladas reducen considerablemente el margen de error humano y también agilizan la detección de desviaciones, facilitando la generación de informes de revisión claros y sustentados. En conjunto, el proceso contribuye significativamente a mejorar la calidad del modelo y fortalecer el cumplimiento de los estándares y objetivos del proyecto.

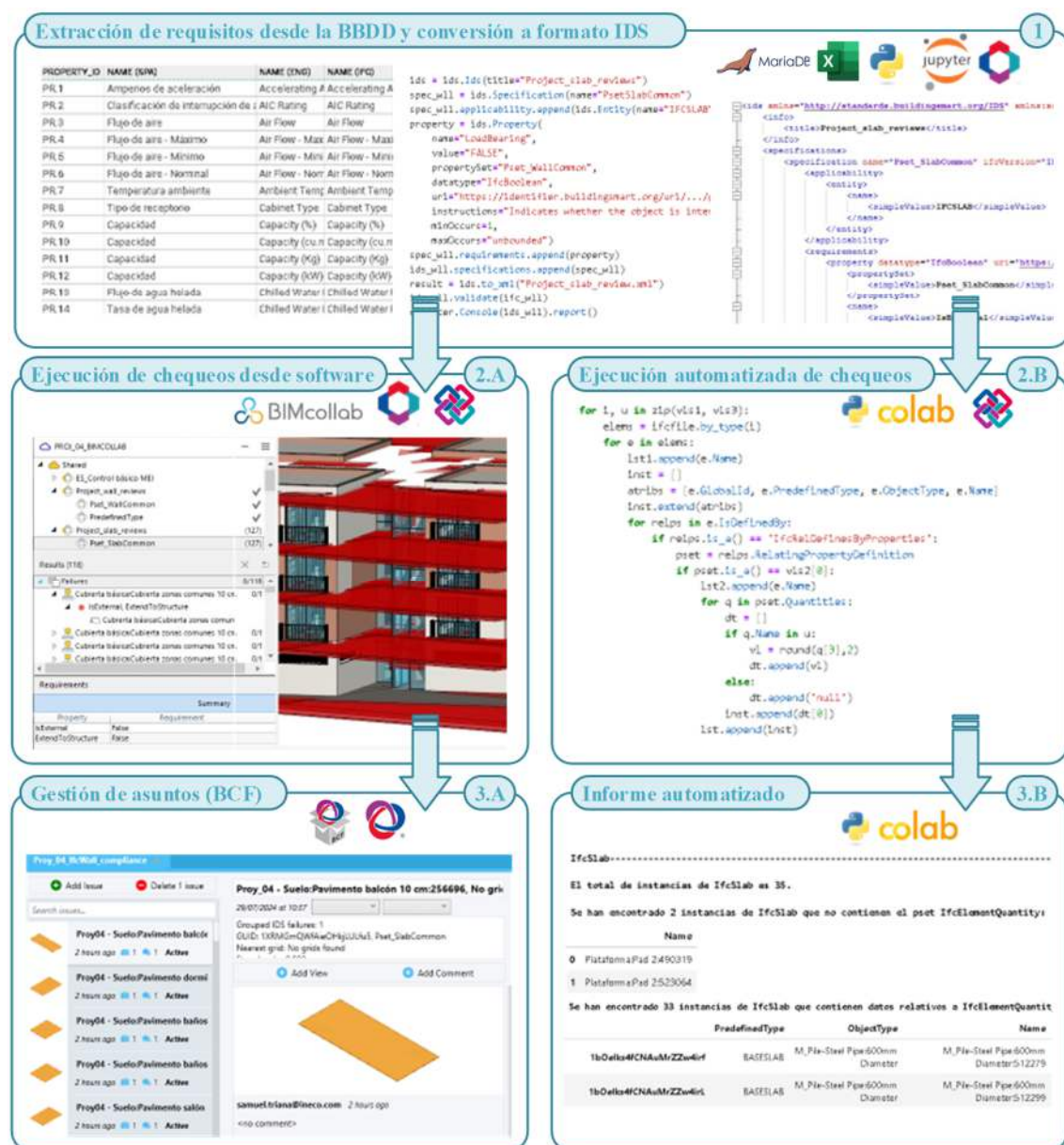


Figura 3.5.76. Procesos de control de calidad de los datos BIM

3.5.10. Consolidación y analítica de datos

Para desarrollar un ejemplo práctico del proceso de gestión del modelo federado de coordinación y su posterior integración en la base de datos de la organización, se desarrollan nuevos flujos de trabajo con Navisworks Manage, BIMCollab Zoom y Bexel Manager. Junto a estos programas de coordinación, se realizan diferentes automatizaciones que permiten conectar de manera semi automática la información extraída de la base de datos con diferentes definiciones de estos softwares, como conjuntos de búsquedas, tablas de datos o listados de clasificación.

Desde el entorno de Autodesk Navisworks, se comienza con la generación de conjuntos de búsqueda de entidades IFC, así como las definiciones de datos de la sección de búsqueda de elementos. Ambas definiciones se extraen de los datos exportados desde MariaDB y se generan en el software mediante automatizaciones con Jupyter Notebook. En los dos casos, se utilizan plantillas con formato XML, generadas a partir de datos tabulares de la BBDD. Desde el entorno de programación, y con los paquetes pandas, xml e IfcOpenShell de Python, se crean las estructuras de archivos y la información relacionada con las reglas de búsqueda y las definiciones en cada caso.

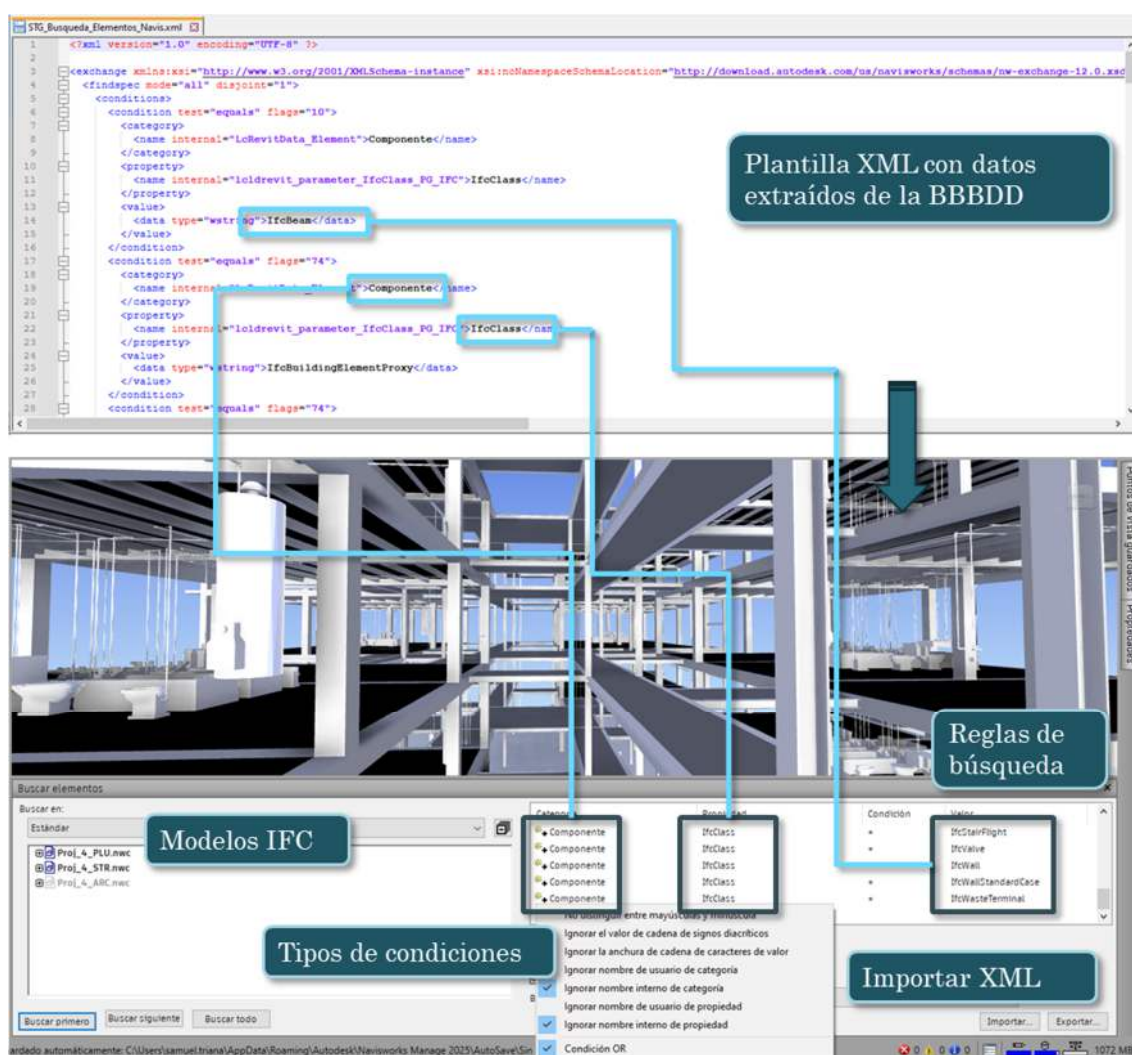


Figura 3.5.77. Importación XML de búsquedas en Navisworks

Dentro de las definiciones del XML de búsquedas es posible filtrar elementos según su nombre, categoría, nivel, propiedades personalizadas o valores IFC, entre otros. Se pueden utilizar operadores lógicos como Y, O y NO, combinaciones de condiciones y comodines para afinar los resultados. Esta flexibilidad permite, por ejemplo, seleccionar todos los elementos estructurales de una determinada fase del proyecto o localizar elementos con propiedades vacías o inconsistentes. Es importante tener en cuenta que los modelos provenientes de diferentes programas pueden tener estructuras de propiedades distintas, lo cual obliga a adaptar los criterios de búsqueda a cada caso.

Una vez definidas las búsquedas, se agrupan los resultados en lo que se denomina conjuntos de búsqueda. Estas agrupaciones son dinámicas por lo que, si el modelo se actualiza, los elementos que cumplen las condiciones del conjunto se actualizan automáticamente. Estos conjuntos son de una alta relevancia en el trabajo de coordinación debido a que pueden emplearse en múltiples contextos, como la vinculación con reglas de detección de interferencias, la generación de animaciones, la planificación 4D o la exportación de datos. Para que las configuraciones de estos conjuntos provengan de datos fiables, se ha generado un Addin en C# que ejecuta la creación de conjuntos desde un archivo de datos estructurado a partir del uso de la API de Navisworks y el entorno de programación de Visual Studio.

```
using Autodesk.Navisworks.Api;
using Autodesk.Navisworks.Api.Plugins;
using Autodesk.Navisworks.Api.DocumentParts;
using System;
using System.Windows.Forms;
using ExcelDataReader;
namespace NavisworksSearchSetPlugin
{
    [Plugin("NavisworksSearchSetPlugin", "Samuel", DisplayName = "STG_Conjuntos")]
    public class CreadorDeConjuntoDeBusqueda : AddInPlugin
    {
        public override int Execute(params string[] parameters)
        {
            try
            {
                string excelFilePath = ShowOpenFileDialog();
                if (string.IsNullOrEmpty(excelFilePath))
                {
                    MessageBox.Show("No se seleccionó ningún archivo.");
                    return 0;
                }
                // Leer códigos desde el Excel
                var codigos = LeerCodigosDesdeExcel(excelFilePath);
                if (codigos.Count == 0)
                {
                    MessageBox.Show("No se encontraron códigos en el archivo de Excel.");
                    return 0;
                }
                Document doc = Autodesk.Navisworks.Api.Application.ActiveDocument;
                foreach (var codigo in codigos)
                {
                    var search = new Search();
                    search.Selection.SelectAll();
                    search.SearchConditions.Add(
                        SearchCondition.HasPropertyByDisplayName("Propiedad", "Valor")
                        .EqualValue(VariantData.FromDisplayString(codigo)));
                    var items = search.FindAll(doc);
                }
            }
            catch { }
        }
    }
}
```

```

    if (items.Count > 0)
    {
        SearchSet searchSet = new SearchSet
        {
            DisplayName = $"Conjunto_{codigo}",
            Search = search;
        };
        doc.SearchSets.AddCopy(searchSet);
    }
    MessageBox.Show("Conjuntos de búsqueda creados correctamente.");
    return 0;
}
catch (Exception ex)
{
    MessageBox.Show("Error: " + ex.Message);
    return 1;
}

private string ShowOpenFileDialog()
{
    OpenFileDialog dialog = new OpenFileDialog();
    dialog.Filter = "Archivos de Excel (*.xlsx) | *.xlsx";
    dialog.Title = "Seleccionar archivo de Excel con códigos";
    return dialog.ShowDialog() == DialogResult.OK ? dialog.FileName : null;
}

private List<string> LeerCodigosDesdeExcel(string filePath)
{
    Encoding.RegisterProvider(CodePagesEncodingProvider.Instance);
    List<string> codigos = new List<string>();
    using (var stream = File.Open(filePath, FileMode.Open, FileAccess.Read))
    using (var reader = ExcelReaderFactory.CreateReader(stream))
    {
        var result = reader.AsDataSet();
        var table = result.Tables[0];
        for (int i = 1; i < table.Rows.Count; i++)
        {
            var valor = table.Rows[i][0].ToString();
            if (!string.IsNullOrEmpty(valor))
                codigos.Add(valor.Trim());
        }
    }
    return codigos;
}

```

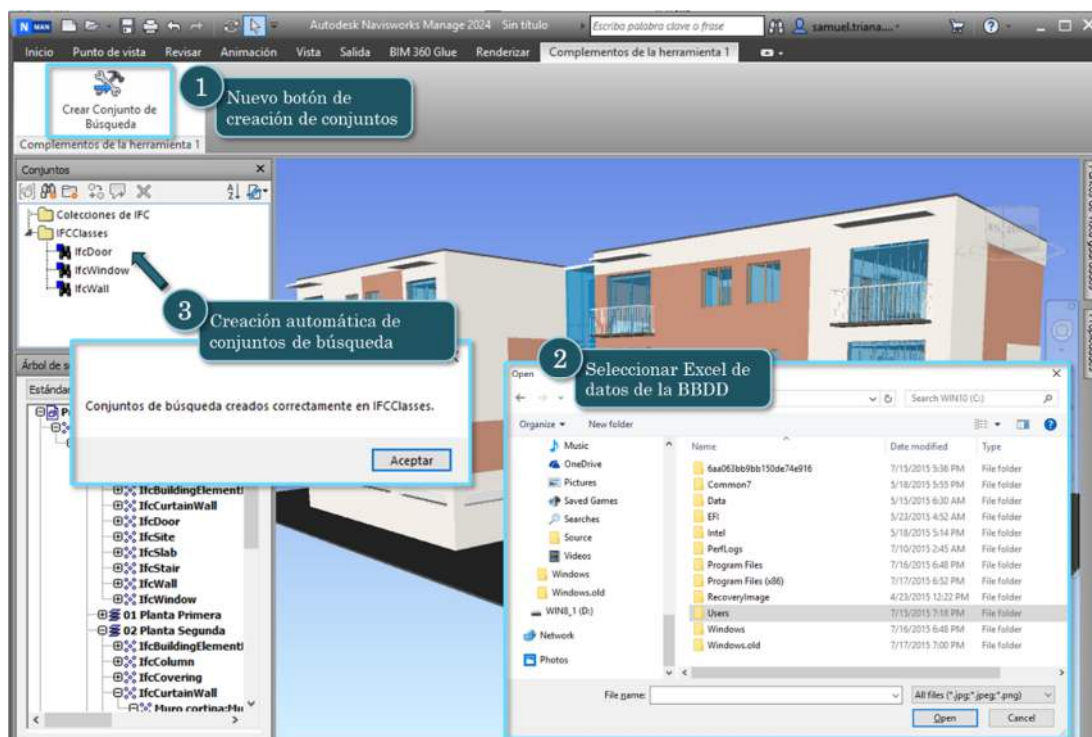


Figura 3.5.78. Addin de Navisworks para creación de conjuntos

Posteriormente, desde la herramienta del Inspector de selección, se pueden crear propiedades rápidas, funcionalidad que permite visualizar de forma inmediata información de los elementos. Los valores mostrados pueden proceder de propiedades propias del modelo o ser generados a partir de combinaciones de propiedades mediante la herramienta avanzadas, lo que permite enriquecer la información visual sin alterar el modelo original. Esta capacidad es útil para revisiones rápidas o para facilitar la navegación de usuarios menos técnicos que necesitan interpretar los datos del modelo sin necesidad de inspeccionar todas sus propiedades manualmente. Una vez más, se realiza el proceso de conversión de los datos XLSX que han sido exportados de la BBDD a un archivo XML que pueda ser importado en el programa.

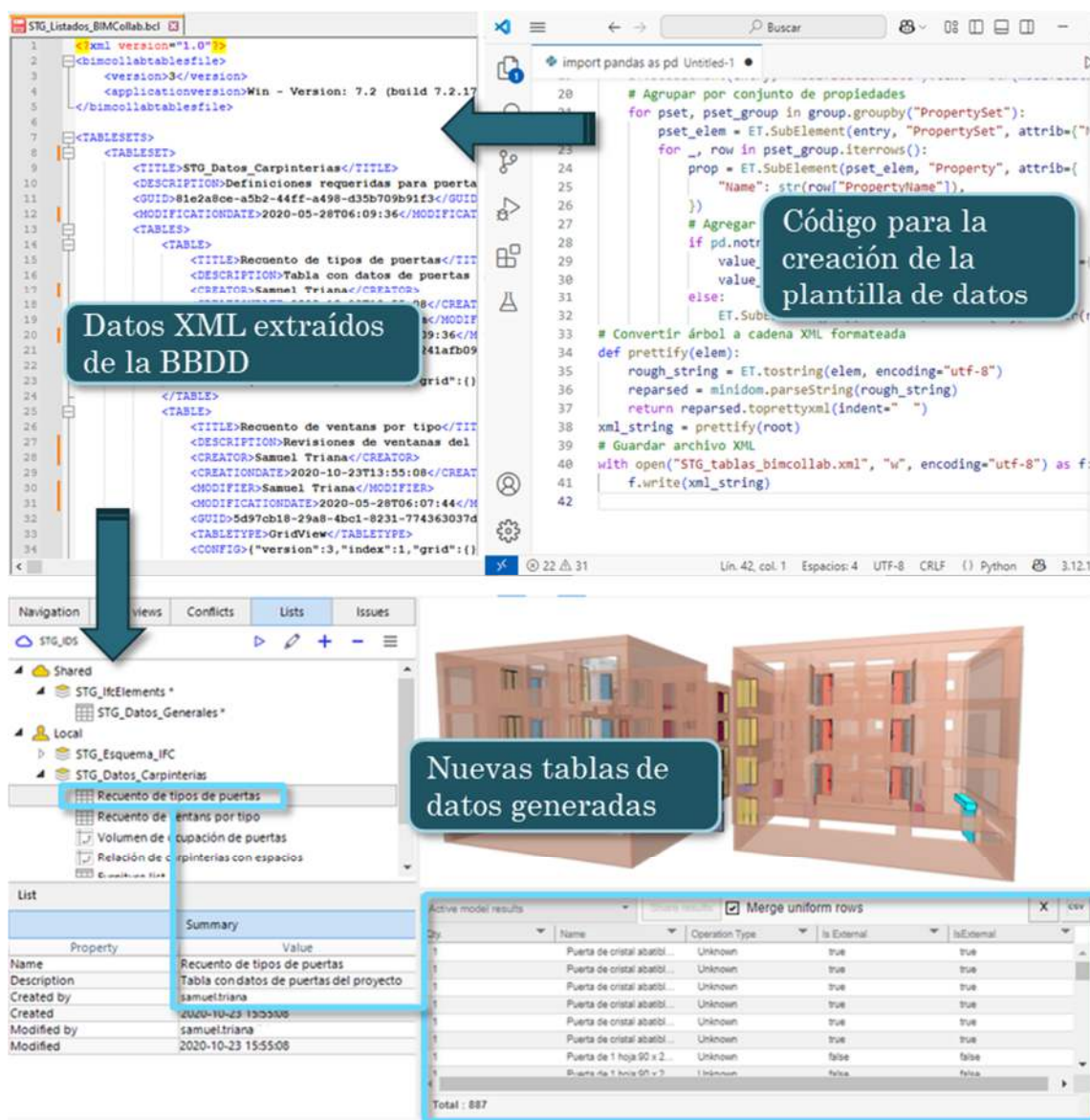
```
import pandas as pd
import xml.etree.ElementTree as ET
df = pd.read_excel('datos_BBDD.xlsx')
# Crear la raíz del XML
exchange = ET.Element('exchange', {
    'xmlns:xsi': "http://www.w3.org/2001/XMLSchema-instance",
    'xsi:noNamespaceSchemaLocation':
    "http://download.autodesk.com/us/navisworks/schemas/nw-exchange-12.0.xsd"})
# Estructura fija
optionset_root = ET.SubElement(exchange, 'optionset', {'name': ''})
interface_set = ET.SubElement(optionset_root, 'optionset', {'name': 'interface'})
smart_tags_set = ET.SubElement(interface_set, 'optionset', {'name': 'smart_tags'})
def add_option(parent, name, value, flags=None):
    attribs = {'name': name}
    if flags:
        attribs['flags'] = flags
    option = ET.SubElement(parent, 'option', attribs)
    data = ET.SubElement(option, 'data', {'type': 'bool'})
    data.text = 'true' if value else 'false'
add_option(smart_tags_set, 'enabled', True)
add_option(smart_tags_set, 'hide_category', False, flags="32")
optionarray = ET.SubElement(smart_tags_set, 'optionarray', {'name': 'definitions'})
# Agregar cada fila del Excel como un nuevo set
for _, row in df.iterrows():
    item_set = ET.SubElement(optionarray, 'optionset', {'name': ''})
    category = ET.SubElement(item_set, 'option', {'name': 'category'})
    category_data = ET.SubElement(category, 'data', {'type': 'name'})
    category_name = ET.SubElement(category_data, 'name', {'internal':
    'LcRevitData_Element'})
    category_name.text = 'Componente'
    property_ = ET.SubElement(item_set, 'option', {'name': 'property'})
    property_data = ET.SubElement(property_, 'data', {'type': 'name'})
    property_name = ET.SubElement(property_data, 'name', {'internal':
    row['internal_name']})
    property_name.text = row['property_name']
# Crear el árbol XML y guardarlo
tree = ET.ElementTree(exchange)
tree.write('salida.xml', encoding='utf-8', xml_declaration=True)
```



Datos XML extraídos de la BBDD

Estas tablas también pueden ser generadas a partir del uso de plantillas XML. Para automatizar esta tarea a partir de un archivo Excel, se puede diseñar un proceso que lea los datos estructurados y los transforme en el formato requerido, utilizando código Python desde el entorno de programación de Visual Studio Code. El primer paso consiste en definir la organización del archivo que requiere la aplicación. Luego se establece una correspondencia entre las columnas del Excel y los nodos del XML, para que cada fila del Excel represente un conjunto de datos que será convertido en un fragmento del XML. El proceso automatizado lee cada fila del Excel, extrae los valores correspondientes, los inserta en las etiquetas adecuadas de acuerdo con la plantilla definida, y repite la operación por cada fila del documento. Finalmente, todo el contenido generado se une en un solo archivo XML completo, listo para su validación e importación.

Durante la construcción del código, es importante considerar cuidadosamente los encabezados que representan las propiedades, los conjuntos de propiedades y los tipos de datos, ya que estos encabezados son los que se traducen directamente en nodos del XML. En el caso desarrollado, cada columna del Excel actúa como una definición del contenido que irá en una propiedad XML, incluyendo atributos como *Name*, *IsExternal* y *OperationType*, agrupados bajo conjuntos como *Summary* o *Pset_DoorCommon*. Además, deben tenerse en cuenta las condiciones asociadas a cada propiedad, como el tipo de dato y si tienen valores vacíos o constantes que afectan la generación de etiquetas vacías o etiquetas con contenido predeterminado. Este modelo también incorpora metainformación como GUIDs y fechas de creación y modificación, por lo que estos campos también deben estar presentes en el Excel de forma estructurada.



The figure illustrates the workflow for generating data tables from a BIMCollab Zoom project. It shows the XML structure extracted from the database, the Python code used to process this data, and the resulting data tables within the application's interface.

Datos XML extraídos de la BBDD

Código para la creación de la plantilla de datos

```

# Agrupar por conjunto de propiedades
for pset, pset_group in group.groupby("PropertySet"):
    pset_elem = ET.SubElement(entry, "PropertySet", attrib={"Name": str(pset_group)})
    for _, row in pset_group.iterrows():
        prop = ET.SubElement(pset_elem, "Property", attrib={"Name": str(row["PropertyName"])})
        # Agregar valor
        if pd.notna(row["value"]):
            prop.text = row["value"]
        else:
            prop.text = ""
    ET.SubElement(entry, "Table", attrib={"Name": str(pset_group)})
    # Convertir árbol a cadena XML formateada
    def prettify(elem):
        rough_string = ET.tostring(elem, encoding="utf-8")
        reparsed = minidom.parseString(rough_string)
        return reparsed.toprettyxml(indent="  ")
    xml_string = prettify(root)
    # Guardar archivo XML
    with open("STG_tablas_bimcollab.xml", "w", encoding="utf-8") as f:
        f.write(xml_string)
  
```

Nuevas tablas de datos generadas

Property	Value
Name	Recuento de tipos de puertas
Description	Tabla con datos de puertas del proyecto
Created by	samuel triana
Created	2020-10-23 15:55:08
Modified by	samuel triana
Modified	2020-10-23 15:55:08

Name	Operation Type	Is External
Puerta de cristal abatible...	Unknown	true
Puerta de cristal abatible...	Unknown	true
Puerta de cristal abatible...	Unknown	true
Puerta de cristal abatible...	Unknown	true
Puerta de cristal abatible...	Unknown	true
Puerta de 1 hoja 90 x 2...	Unknown	false
Puerta de 1 hoja 90 x 2...	Unknown	false

Total : 887

Figura 3.5.80. Tabla de datos desde plantilla XML en BIMCollab Zoom

Otra de las herramientas del programa que pueden interactuar con los datos de la BBDD son las SmartViews, las cuales son similares a los conjuntos de búsqueda de

Navisworks, pero adicionalmente, permiten desde la configuración de cada vista establecer modificaciones de visibilidad y gráficos a los elementos que cumplan con las reglas de búsqueda.

Para este segundo flujo en la aplicación también se utiliza un código generado con el objetivo de automatizar la creación de la plantilla, en este caso un archivo *.bscv*, a partir de datos de la BBDD. La estructura del código se basa en dos componentes principales. Primero, se utiliza pandas para leer los datos de cada SmartView y sus reglas asociadas. Luego, se emplea la librería *xml.etree.ElementTree* para construir la jerarquía XML esperada por el formato BIMCollab, empezando por los metadatos del archivo y continuando con los nodos de vistas, reglas y acciones. Al final, el código guarda el archivo XML combinando las secciones de encabezado y contenido.

```
import pandas as pd
import xml.etree.ElementTree as ET
from datetime import datetime
def create_bscv_from_excel(excel_path, output_path):
    df = pd.read_excel(excel_path)
    # Crear estructura XML base
    root = ET.Element("bimcollabsmartviewfile")
    ET.SubElement(root, "version").text = "6"
    ET.SubElement(root, "applicationversion").text = "Generated by script"
    smartviewsets = ET.Element("SMARTVIEWSETS")
    sv_set = ET.SubElement(smartviewsets, "SMARTVIEWSET")
    ET.SubElement(sv_set, "TITLE").text = "SmartView Set Auto"
    ET.SubElement(sv_set, "DESCRIPTION").text = ""
    ET.SubElement(sv_set, "GUID").text = "auto-generated-guid"
    ET.SubElement(sv_set, "MODIFICATIONDATE").text =
datetime.now().isoformat()
    views_grouped = df.groupby("View Name")
    smartviews = ET.SubElement(sv_set, "SMARTVIEWS")
    for view_name, group in views_grouped:
        view = ET.SubElement(smartviews, "SMARTVIEW")
        ET.SubElement(view, "TITLE").text = view_name
        now = datetime.now().isoformat()
        for tag in ["CREATOR", "MODIFIER"]:
            ET.SubElement(view, tag).text = "autogen"
        for tag in ["CREATIONDATE", "MODIFICATIONDATE"]:
            ET.SubElement(view, tag).text = now
        ET.SubElement(view, "GUID").text = "autogen-" + view_name.replace(" ", "_")
        rules = ET.SubElement(view, "RULES")
        for _, row in group.iterrows():
            rule = ET.SubElement(rules, "RULE")
            ET.SubElement(rule, "IFCTYPE").text = row["IFC Type"]
            prop = ET.SubElement(rule, "PROPERTY")
            ET.SubElement(prop, "NAME").text = str(row["Property Name"])
            ET.SubElement(prop, "PROPERTYSETNAME").
            text = str(row["Property Set"])
            ET.SubElement(prop, "TYPE").text = str(row["Property Set"])
```

```

ET.SubElement(prop, "VALUETYPE").text = str(row["Value Type"])
ET.SubElement(prop, "UNIT").text = str(row["Unit"])
condition = ET.SubElement(rule, "CONDITION")
ET.SubElement(condition, "TYPE").text = row["Condition Type"]
ET.SubElement(condition, "VALUE").
text = str(row["Condition Value"]) if pd.notna(row["Condition Value"]) else ""
action = ET.SubElement(rule, "ACTION")
ET.SubElement(action, "TYPE").text = row["Action Type"]
if "R" in row and not pd.isna(row["R"]):
    ET.SubElement(action, "R").text = str(int(row["R"]))
    ET.SubElement(action, "G").text = str(int(row["G"]))
    ET.SubElement(action, "B").text = str(int(row["B"]))
info = ET.SubElement(view, "INFORMATIONTAKEOFF")
ET.SubElement(info, "PROPERTYSETNAME").text = "None"
ET.SubElement(info, "PROPERTYNAME").text = "None"
ET.SubElement(info, "OPERATION").text = "0"

#Guardar archivo
tree1 = ET.ElementTree(root)
tree2 = ET.ElementTree(smartviewsets)
with open(output_path, "wb") as f:
    f.write(b'<?xml version="1.0"?>\n')
    tree1.write(f, encoding="utf-8", xml_declaration=False)
    f.write(b'\n')
    tree2.write(f, encoding="utf-8", xml_declaration=False)
create_bscv_from_excel("plantilla_BBDD.xlsx", "STG_plantilla_SmartView.bscv")

```

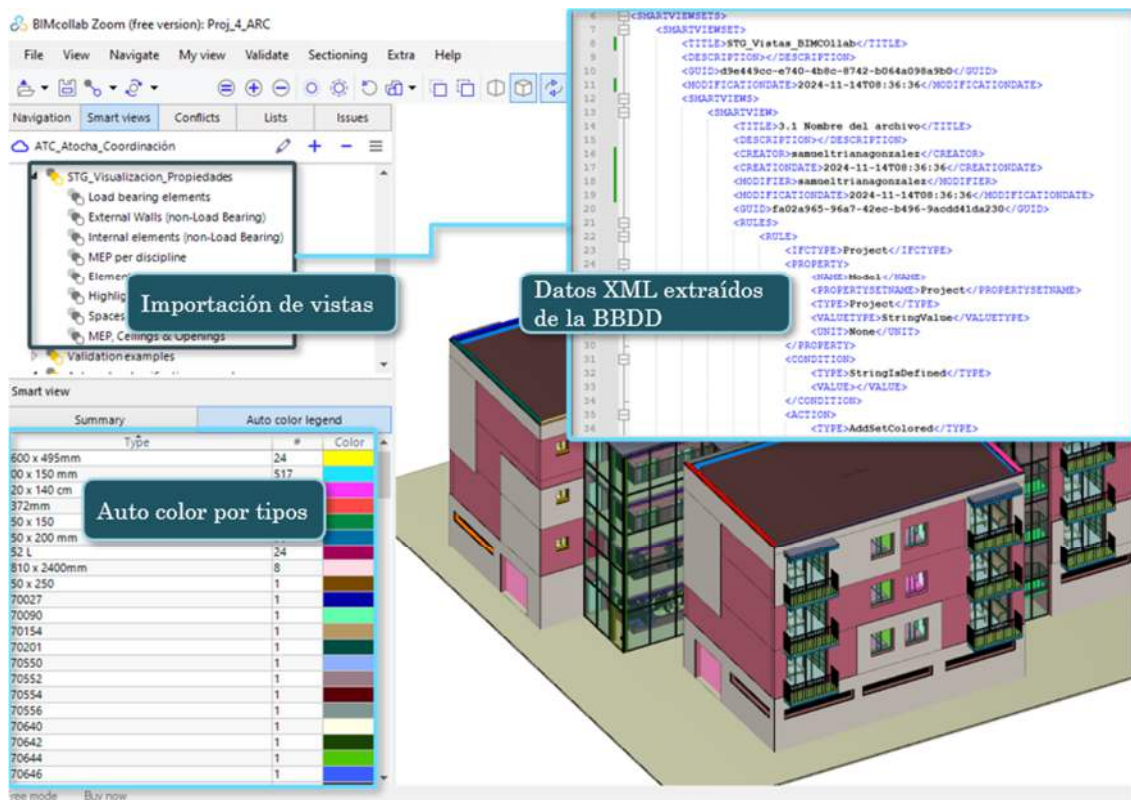


Figura 3.5.81. SmartViews desde plantilla XML en BIMCollab Zoom

Bexel Manager es otro de los programas de coordinación que cuenta con la posibilidad de importación y exportación de plantillas preconfiguradas de datos para el trabajo con diferentes herramientas de análisis y control BIM. Algunas de estas plantillas se definen con código binario, por lo que no es posible el tratamiento de los datos fuera del entorno del programa. Sin embargo, en otros casos existen herramientas que utilizan archivos XLSX para el intercambio de datos, los cuales son muy fáciles de manejar y preconfigurar.

El código Python desarrollo para esta plantilla parte de un proceso automatizado de extracción, análisis y transformación de datos. Partiendo de una tabla en la que las expresiones de búsqueda se encuentran distribuidas en columnas y formuladas en un formato funcional, se aplican técnicas de análisis mediante expresiones regulares para identificar y descomponer los patrones sintácticos. Este procedimiento estructura la información en una forma tabular estandarizada que segmenta las componentes semánticas clave: la condición lógica, la categoría del elemento IFC y el valor objetivo de búsqueda, dando como resultado final la plantilla de Bexel.

```
import pandas as pd
import re
df = pd.read_excel("datos_BBDD.xlsx")
# Aquí uso 'Category' y 'Name', cambia a los nombres reales que tengas
df['Query'] = df.apply(lambda row: f'match(['row['Category']'],
'^({0}}{row['Name']}.*)$', axis=1)
# Función para extraer datos
def parse_query(query):
    condition_match = re.search(r'(\w+)\s*(\(', query)
    condition = condition_match.group(1) if condition_match else ""
    category_match = re.search(r"\[(.+?)\]", query)
    category = category_match.group(1) if category_match else ""
    value_match = re.search(r'\(\. \{0\} (.+?) \. \*\)', query)
    value = value_match.group(1) if value_match else ""
    return pd.Series([condition, category, value])
# Aplicar función parse_query
df[['Condition', 'Category_Parsed', 'Value']] = df['Query'].apply(parse_query)
# Agregar columnas fijas para el XML
df['CaseInsensitive'] = 'False'
df['Unit'] = 'None'
df['UnitClass'] = 'None'
df['FieldTitle'] = ""
df['UserDefined'] = 'False'
df['Validation'] = 'None'
# Elegir columnas finales
columns_finales = [
    'Folder', 'Name', 'Condition', 'Category_Parsed', 'Value',
    'CaseInsensitive', 'Unit', 'UnitClass', 'FieldTitle', 'UserDefined', 'Validation']
df_final = df[columns_finales]
# Guardar resultado
df_final.to_excel("STG_Visualizacion_Conjuntos.xlsx", index=False)
```

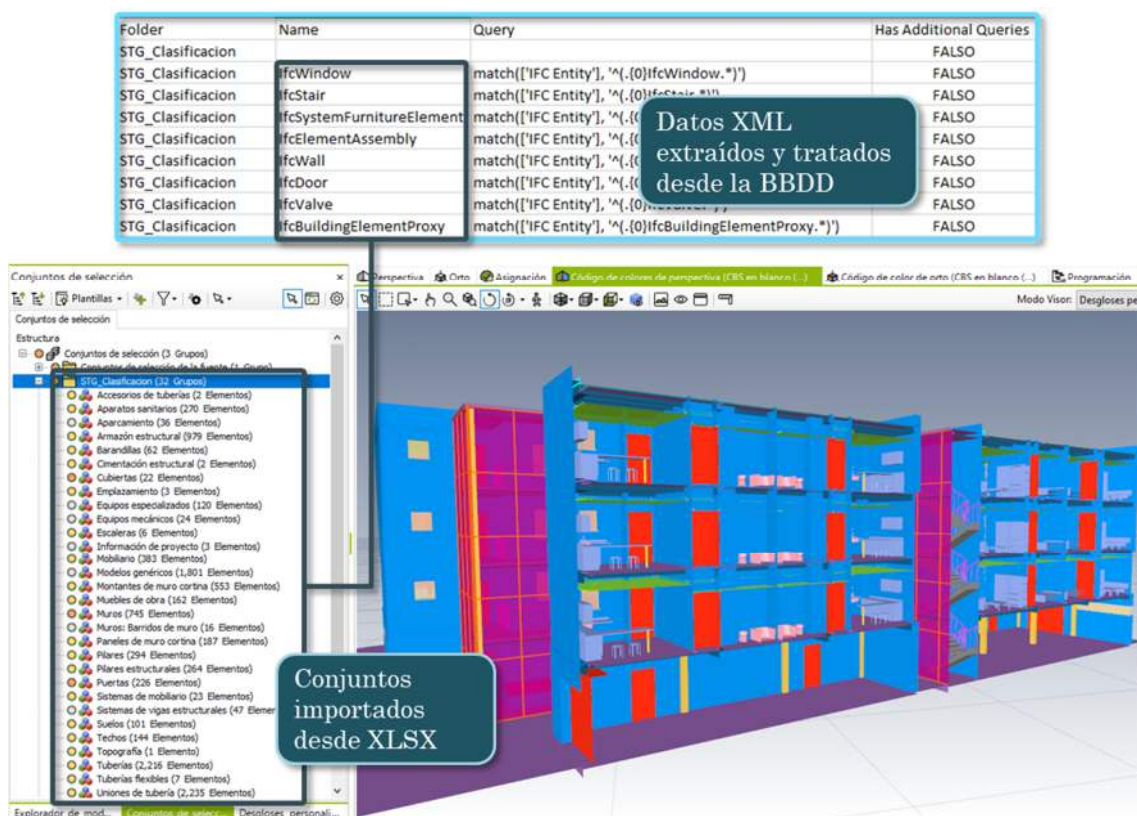


Figura 3.5.82. Plantilla de conjuntos de objetos XLSX en Bexel

Esta aplicación de coordinación también permite la importación de plantillas de clasificaciones para el trabajo con definiciones de planificación y también de mediciones y presupuestos. El formato de igual manera es XLSX por lo que el tratamiento de los datos de la BBDD es bastante simple, ya que solo hay que adaptar algunas celdas para que puedan ser entendidas durante el proceso de importación.

Al tratarse de una clasificación, es importante definir los niveles de cada ítem, estableciendo también para cada uno al menos un número de identificación, un nombre y una regla o query que permita conectar la línea de clasificación con los elementos de los modelos IFC. Esta query parte de datos de columnas independientes que han sido extraídos desde MariaDB y que luego se operan para conectarlos dentro de un mismo conjunto de información para cada registro de la tabla.

```
import pandas as pd
#Leer el archivo Excel de entrada
df = pd.read_excel("Datos_BBDD.xlsx")
#Función para construir la expresión match
def construir_query(row):
    categoria = row['Categoría']
    valor = row['Valor']
    return f'match(['{categoria}'], '{valor}'))'
#Aplicar la función a cada fila para crear la columna 'Element query'
df['Element query'] = df.apply(construir_query, axis=1)
#Guardar el DataFrame resultante en un nuevo archivo Excel
df.to_excel("STG_Clasificaciones_Bexel.xlsx", index=False)
```




Figura 3.5.83. Plantilla de clasificaciones XLSX importadas en Bexel

Cabe destacar que, dependiendo del uso BIM que se desarrolle, existen otros datos que deben ser definidos en la plantilla, como información relacionada con mediciones y costes asociadas a unidades, unidades de coste, recursos enlazados, etc. También es importante en estos casos la definición de identificadores para cada ítem de la clasificación, ya que este es el dato clave que permite mantener la trazabilidad de la información, permitiendo en el futuro mantener la coherencia de los datos cuando se realicen procesos de actualización o modificación del archivo.

Una vez que se finalizan los procesos de coordinación y validación de la información de los archivos IFC dentro del modelo federado, se procede a extraer la información en formatos tabulares y de manera estructurada, en base a la organización planteada para los datos de proyectos dentro de MariaDB. La realización del flujo de trabajo desde Navisworks se realiza desde el Inspector de selección, a través de las opciones de exportación y la configuración de las propiedades rápidas, incluyendo incluso nueva información que se haya integrado desde fuentes externas a partir de la herramienta DataTools de conexión de datos.

Esta extracción se basa en la composición de diferentes conjuntos de datos que atienden a información de todas las entidades IFC relevantes, tanto clases espaciales y de proyecto, como específicas de elementos constructivos y sistemas compuestos de elementos. Para cada conjunto, se va combinando la selección de los objetos a partir

de los conjuntos de búsqueda junto con la definición de las propiedades rápidas que definen las propiedades que van apareciendo en los conjuntos de datos. En este proceso es muy importante gestionar correctamente los niveles exactos de selección de entidades de los esquemas IFC, ya que dentro de Navisworks es posible seleccionar las instancias IFC, pero también sus subcomponentes y geometrías y sus tipos y clases relacionadas. Para ello, hay que asegurar que las búsquedas que han generado los conjuntos hayan tenido en cuenta todas las casuísticas posibles, incluyendo condiciones OR para incluir clases que hayan cambiado entre versiones del esquema IFC.



Figura 3.5.84. Tabla de datos a exportar desde Navisworks

Este proceso de exportación se puede realizar de la misma manera dentro del entorno de BIMCollab Zoom, a partir de los listados de datos que generan tablas de cada entidad IFC con la información relacionada en cada caso. Por otro lado, Bexel Manager no tiene directamente una ventana dentro de la aplicación que muestre una tabla de datos, pero sí que pueden seleccionarse las instancias de cada entidad, personalizar una ventana con las propiedades que debe contener los encabezados de la tabla y, por último, exportar todos los datos a un archivo XLSX.

En el presente caso de estudio se contempla tanto la exportación de tablas con conjuntos de propiedades de instancias como tablas con conjuntos de tipos de objetos. Una vez que se tienen todos los archivos exportados, se realiza un nuevo proceso de importación a MariaDB que se automatiza mediante código Python desde Jupyter, extrayendo los datos de los archivos tabulares, sean estos CVS o XLSX.

Este proceso se realiza a partir de un script de Python, en el que se integran los datos IFC de la hoja Excel en la base de datos de la organización. Este flujo utiliza bibliotecas como *pandas* y *SQLAlchemy* para poder realizar los códigos. En primer lugar, se establece una conexión con la base de datos MySQL, que permite una interacción con el sistema de gestión de bases de datos relacional. Posteriormente, se procede a la lectura de todas las hojas contenidas en el archivo Excel.

Cada hoja del archivo es procesada individualmente, donde se realiza una limpieza y normalización de los nombres de las columnas para garantizar la coherencia y evitar posibles conflictos durante la inserción en la BBDD. Se eliminan las filas que no contienen información relevante y se reemplazan los valores nulos por *None* para asegurar la compatibilidad con MariaDB. Finalmente, los datos procesados de cada hoja se insertan en la base como tablas independientes, utilizando el nombre de la hoja como referencia para el nombre de la tabla correspondiente.

```
import pandas as pd
from sqlalchemy import create_engine
import os

#Configuración de la conexión a la base de datos
DB_USER = 'XXXX'
DB_PASSWORD = 'XXXX'
DB_HOST = 'localhost'
DB_PORT = '3306'
DB_NAME = 'exp_0001_edif_proy_a'

#Crear la conexión
engine = create_engine(f'mysql+mysqlconnector://{
DB_USER}:{DB_PASSWORD}@{DB_HOST}:{DB_PORT}/{DB_NAME}')

#Ruta al archivo Excel
excel_file_path = 'C:\04 Casos de uso\EXCEL2MYSQL\001_edif_proy_a_STR.xlsx'
#Leer todas las hojas del archivo Excel
xls = pd.ExcelFile(excel_file_path, engine='openpyxl')
sheet_names = xls.sheet_names
for sheet in sheet_names:
    #Leer cada hoja en un DataFrame
    df = xls.parse(sheet)
    #Limpiar nombres de columnas
    df.columns = [col.strip().lower().replace(' ', '_') for col in df.columns]
    #Eliminar filas completamente vacías
    df.dropna(how='all', inplace=True)
    #Reemplazar valores NaN por None
    df = df.where(pd.notnull(df), None)
    #Definir el nombre de la tabla en la base de datos
    table_name = sheet.strip().lower().replace(' ', '_')
    #Insertar los datos en la base de datos
    try:
        df.to_sql(name=table_name, con=engine, if_exists='replace', index=False)
        print(f'Datos de la hoja "{sheet}" insertados en la tabla "{table_name}".')
    except Exception as e:
        print(f'Error al insertar datos de la hoja "{sheet}": {e}')
```

Este enfoque sistemático y automatizado mejora en gran medida la eficiencia en la gestión de grandes volúmenes de datos provenientes de modelos openBIM y a la vez asegura la integridad y coherencia de la información almacenada. Cada vez que los datos son volcados a la base, también se asegura que los datos obsoletos de anteriores entregar son borrados para que no existan problemas de duplicados. Una vez finalizado el proceso, los datos pueden ser analizados desde phpMyAdmin.

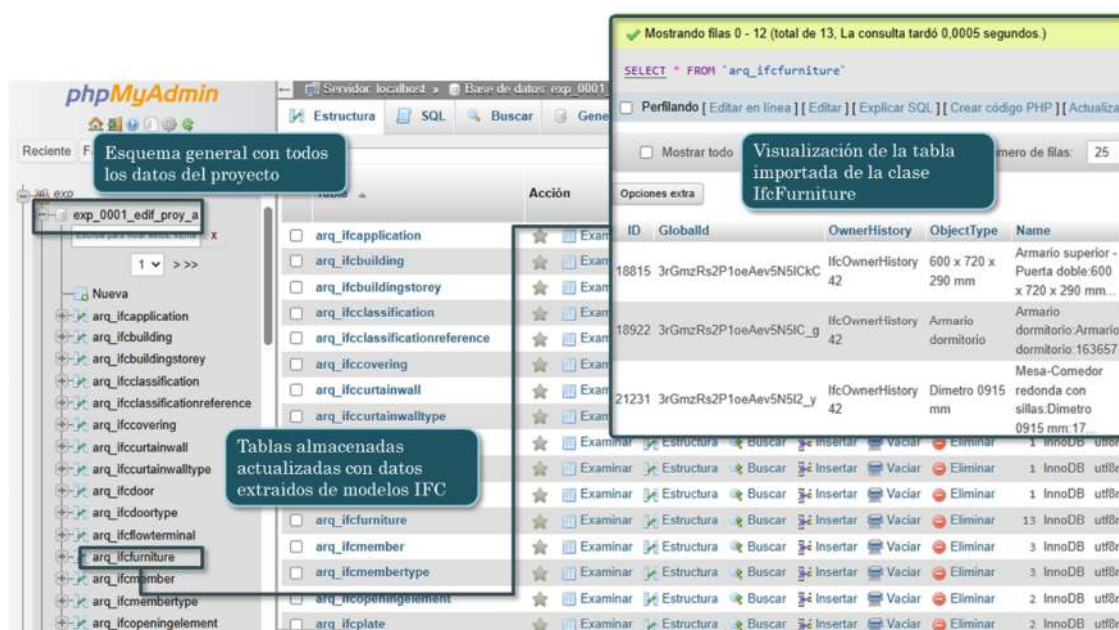


Figura 3.5.85. Integración de los datos del proyecto en MariaDB

Teniendo en cuenta todo lo expuesto con anterioridad, los nuevos flujos planteados posibilitan la gestión de información consistente y la posibilidad de generación de informes detallados y personalizados. Sumado a estos flujos, la sistematización también propone la definición de procesos para el trabajo de los datos desde herramientas de visualización y análisis a partir de paneles interactivos o cuadros de mando que engloben de una manera holística toda la información, tanto a nivel de proyecto como también a nivel de organización. Con ello se busca un mayor control en la toma de decisiones y en la gestión de proyectos, analizando la información clave de los activos de edificación, la cual es establecida a partir de una conectividad entre MariaDB y PowerBI.

De esta manera se desarrollan diferentes enfoques para la organización y generación de tableros visuales de datos que permitan llevar un control y seguimiento de los datos durante todas las fases del ciclo de vida del edificio. Para generarlos a partir de la BBDD, es necesario establecer una conexión directa entre ambos entornos. Este proceso requiere ingresar el nombre del servidor, la BBDD específica y las credenciales de autenticación para realizar la conexión. Una vez establecida, PowerBI permite seleccionar las tablas o vistas necesarias, con opciones de cargar los datos directamente o transformarlos mediante PowerQuery antes de su incorporación al modelo de datos.

En el proceso de diseño de los tableros, se utilizan temas personalizados para mantener una coherencia visual y facilitar la interpretación de la información. Estos temas definen paletas de colores, tipografías y estilos de visualización. Además, durante este proceso se crean medidas y columnas calculadas mediante DAX para más tarde poder realizar análisis más profundos y personalizados, pudiendo ser adaptados a las necesidades de cada proyecto.

Para facilitar la reutilización de estos tableros en múltiples proyectos, se generan también de plantillas PBIT. Estos archivos conservan la estructura del modelo de datos, las visualizaciones y las configuraciones de formato, pero permiten redefinir

la fuente de datos al abrirse. En este caso, al tratarse de un proceso sistematizado en el que cada proyecto repite los mismos flujos, este uso de plantillas es de gran utilidad, pudiendo reutilizar las configuraciones preestablecidas y optimizando así el tiempo de desarrollo de los cuadros de mando.

Por un lado, se desarrollan cuadros que extraen información no BIM del proyecto, relacionados con el seguimiento de incidencias, estado de los trabajos y entregables generados por cada hito de proyecto, datos siempre que provienen de la base relacional. Por otro lado, se generan también tableros que muestran la información contenida en los datos IFC provenientes del modelo federado multidisciplinar. En este caso, se muestran datos relacionados con recuentos y definiciones de entidades IFC, archivos generados, sistemas de clasificación, definiciones de valores predefinidos que han sido rellenados en propiedades, etc.

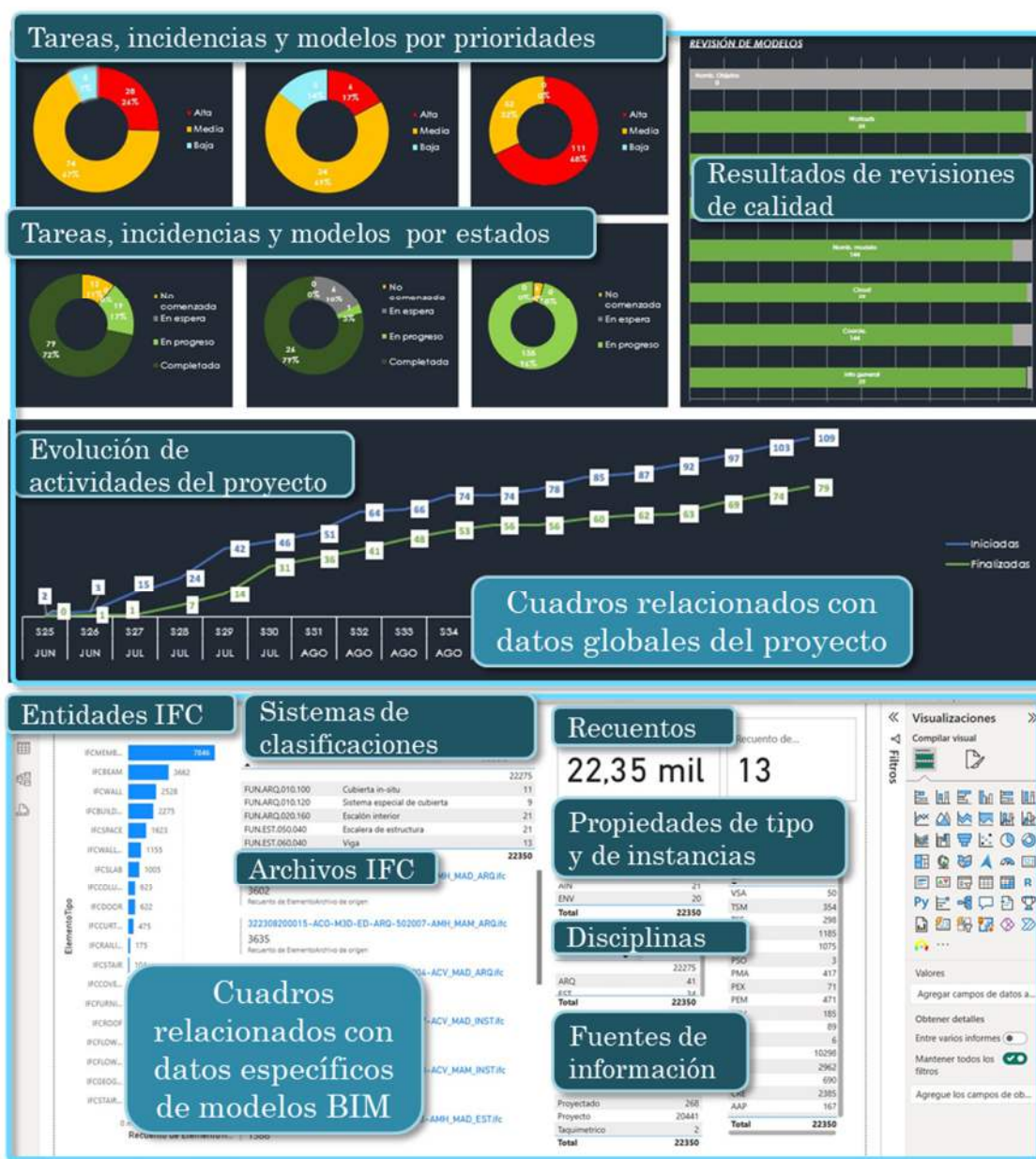


Figura 3.5.86. Generación de cuadros de mando del proyecto

Es importante también la configuración de actualizaciones automáticas para mantener los tableros actualizados con los datos más recientes. Esto se logra mediante la programación de actualizaciones en el servicio de PowerBI, asegurando que los usuarios siempre accedan a información actualizada sin intervención manual. Asimismo, la implementación de controles de acceso y la gestión de permisos garantizan que la información sensible esté protegida y solo sea accesible para usuarios autorizados.

Unificando los flujos desarrollados en este caso de uso, se consigue una parte muy importante de la sistematización que permite integrar la totalidad de la información de la organización dentro de una base de datos global, accesible y actualizable. En el caso concreto de la siguiente imagen del esquema, se muestra un tablero desarrollado con la información combinada de diferentes proyectos de la organización, integrando datos de diferentes esquemas de la base relacional.

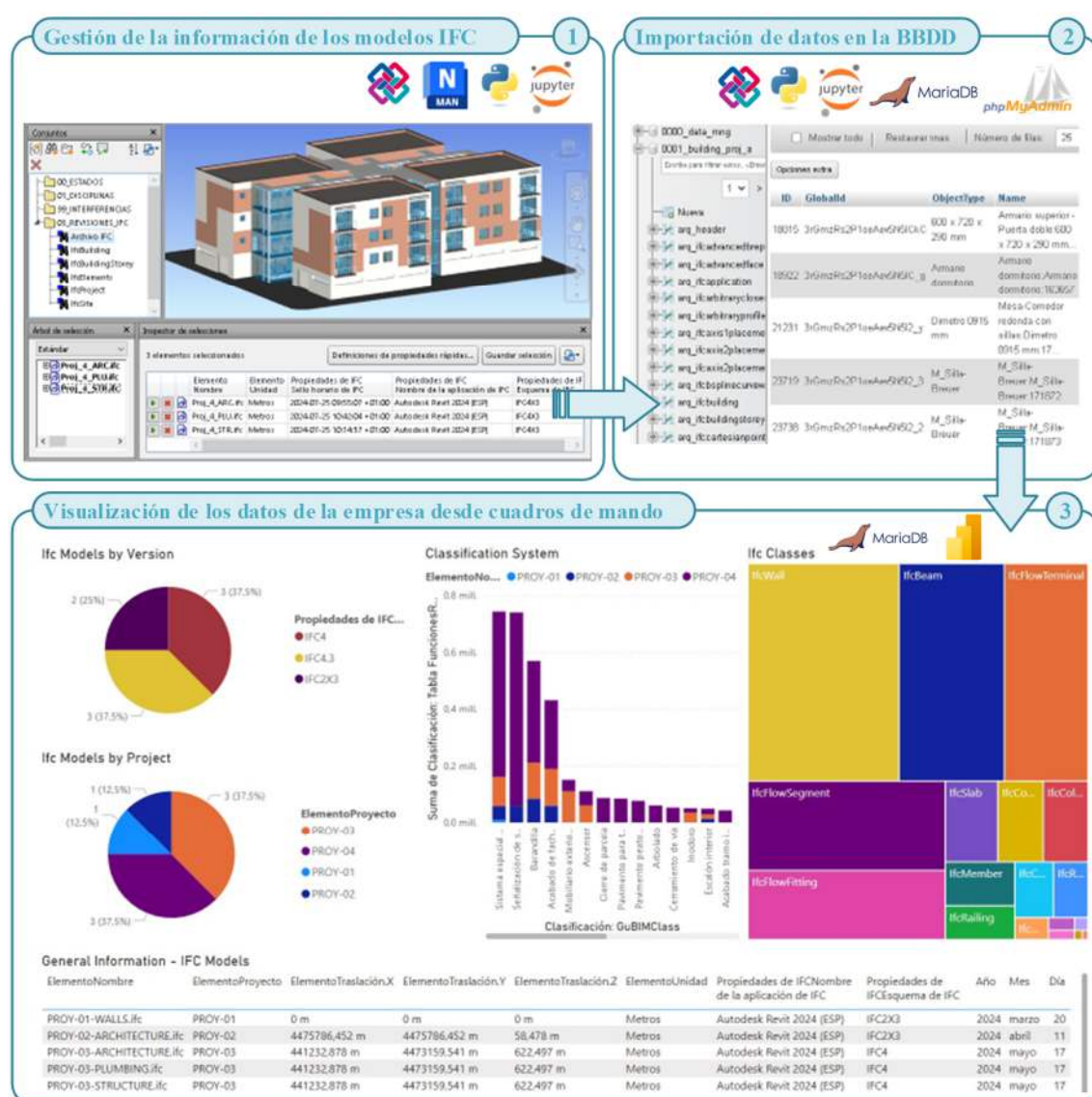


Figura 3.5.87. Proceso de consolidación y analítica de datos

CAPÍTULO 4: ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

4.1. Resultados obtenidos

Tras realizar todos los casos de uso para la aplicación práctica de la sistematización basada en los modelos utilizados en el presente estudio, se recopilaban diversos resultados para evaluar la mejora y la optimización que ofrece la solución propuesta en comparación con los métodos tradicionales de creación y gestión de activos, ya que estos métodos suelen generar falta de coordinación entre disciplinas y errores humanos en la definición de los datos.

El primer proceso de la sistematización es el de recolección de diferentes fuentes y recursos de información, el cual permitió definir una serie de conjuntos de datos de partida para todo el desarrollo de la investigación. Son datos que desde el inicio se van transfiriendo y exportando entre plataformas y aplicaciones de tal manera que se garantiza que la definición, el tipo de dato y su composición sea inalterable en cualquiera de los entornos digitales de trabajo.

Categoría	BIMFORUM	IFC	REVIT	NBS	SCFCClass
Disciplinas	-	7	3	6	24
Subdisciplinas	32	-	8	9	60
Entidades	487	95	45	-	95
Tipos de Entidades	-	289	-	-	2450
Grupos de Propiedades	51	356	32	15	-
Parámetros	1328	1258	1109	1589	-
Tipos de Datos	8	15	19	7	-

Tabla 4.1.1. Información de partida para la BBDD

Junto a estos datos se importaron otros relacionados con los datos de la organización, definiendo por ejemplo usos BIM, fases del proyecto y disciplinas propias de la empresa, además de datos específicos de listados y estructuras de software, como son categorías y familias de sistema de Revit. Los conjuntos de datos fueron definidos de manera clara con una nomenclatura específica y con una tipología de datos permitidos específicos para cada caso.

Todos estos datos, después de un tratamiento de extracción y transformación para llegar a un nivel de estandarización óptimo y subdividir en tablas más pequeñas la información para conseguir conjuntos normalizados, fueron cargados en la base de datos global. Previamente la base fue configurada a partir de la gestión de permisos, usuarios, esquemas y relaciones para poder realizar las operaciones de intercambio de datos de manera correcta.

Durante el proceso de integración de los datos en la base relacional, fueron detectados diferentes errores en la entrada de datos debido a preconfiguraciones que no habían sido definidas correctamente en los datos tabulares de partida. Al realizar la importación, las configuraciones de las tablas de la BBDD ya tenían restricciones por lo que solo permitían integrar datos que cumplieran con lo requerido según lo

establecido en cada caso. Es gracias a ello que se detectaron diferentes inconsistencias dentro de las tablas. Las más destacadas se enumeran a continuación:

- Inconsistencias en el formato de datos: Se detectaron diferencias entre el tipo de dato definido en la base de datos y el tipo de dato contenido en los archivos de origen. Esto ocasionó errores de validación automática durante el proceso.
- Duplicidad de parámetros: Algunos parámetros aparecían repetidos con ligeras variaciones en el nombre o en la capitalización, lo que impedía su correcta normalización e integración como atributos únicos.
- Ausencia de claves primarias o identificadores únicos: Varias entradas no contenían identificadores únicos consistentes, lo cual impedía su vinculación con otras tablas o la verificación de relaciones entre entidades.
- Problemas con caracteres especiales: Durante el proceso de importación se identificaron errores relacionados con caracteres especiales propios del idioma español, como tildes y la letra ñ. En varios casos, estos caracteres fueron reemplazados por símbolos ilegibles o eliminados, afectando la legibilidad y precisión semántica de los datos. Este problema fue resuelto evitando este tipo de caracteres, retirando las tildes y cambiando las ñ por n.
- Asignación errónea de clasificaciones: Se observaron errores en la correspondencia entre entidades y sus respectivas clasificaciones tipológicas, lo que afectó la semántica de los datos.
- Campos obligatorios incompletos: Algunas columnas definidas como obligatorias *NOT NULL* contenían valores vacíos o nulos en los archivos de entrada, lo que bloqueó su inserción.
- Desajustes terminológicos entre fuentes: Se evidenció una falta de correspondencia terminológica entre distintas fuentes, que generó conflictos, pero que fue resuelto mediante el mapeo de definiciones entre sistemas.
- Errores de codificación de caracteres: En ciertos registros, especialmente provenientes de archivos internacionales o en otros idiomas, se produjeron errores por incompatibilidades de codificación, como *UTF-8 vs ANSI*, afectando la correcta lectura de nombres o descripciones.

Aunque son numerosos los errores y avisos que fueron detectados durante esta etapa inicial, el aseguramiento de la correcta definición y validación de toda esta información de partida representa un valor significativo para el desarrollo del proyecto. Esta labor minuciosa no solo permite identificar y corregir posibles inconsistencias o deficiencias en una fase temprana, sino que también establece una base sólida y confiable sobre la cual se construye el resto del trabajo. Al contar con datos bien definidos y estructurados desde el inicio, se minimizan los riesgos de retrabajos y se optimizan los recursos, lo que en última instancia se traduce en un considerable ahorro de tiempo y esfuerzo en las fases de desarrollo de los modelos de información. Además, este enfoque preventivo contribuye a una mayor eficiencia global y a una mejora en la calidad del producto final.

Para poner en práctica el desarrollo de los procesos de la sistematización, se ejecutaron los flujos de trabajo en los cuatro proyectos de investigación definidos al inicio del estudio. El alcance de los datos de estos proyectos se define a continuación:

Tipo de Proyecto	Modelos BIM	Clases	Instancias	Psets	Propiedades	Datos totales
Categoría de objeto individual	1	1	8	10	65	524
Disciplina de proyecto	1	21	758	55	140	2984
Proyecto multidisciplinar	3	38	2078	92	328	31602
Proyecto multidisciplinar (todos)	8	47	14554	165	459	172423

Tabla 4.1.2. Recuento de definiciones de proyectos de investigación

El siguiente gráfico ilustra de manera clara cómo la complejidad y la cantidad de información en un proyecto crecen de forma exponencial a medida que aumenta su escala y requisitos. Mientras que una categoría de objeto individual maneja menos del 1% del total de los datos de los cuatro proyectos, el proyecto multidisciplinar complejo alcanza más del 82%. Por lo tanto, esta progresión no es lineal, ya que cada nivel adicional de integración implica un incremento drástico en la cantidad de clases, instancias y propiedades a gestionar. Esto resalta la importancia de una gestión de información eficiente y escalable en el desarrollo de proyectos BIM.

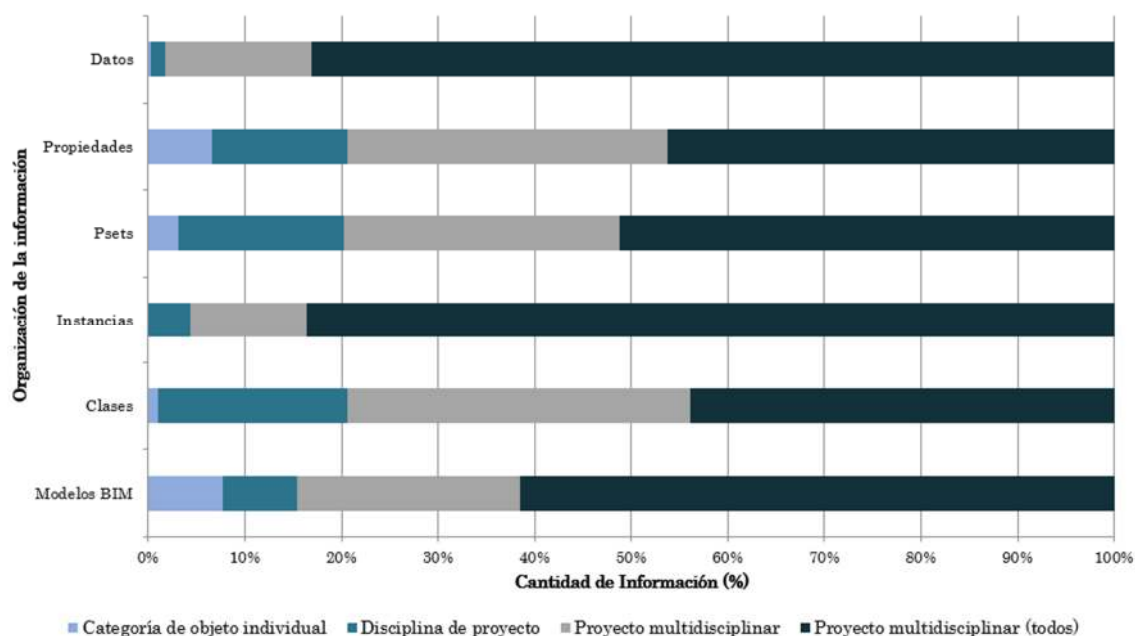


Figura 4.1.1. Incremento exponencial en la complejidad del proyecto

Esta complejidad también se ve reflejada durante el desarrollo del siguiente proceso de la sistematización, en el que se realizan los procesos de extracción de información de la base de datos según los requisitos concretos exigidos por el proyecto de

edificación. Mientras que para la definición del modelo de muros el trabajo es fácil y ágil, las configuraciones y programaciones necesarias para el resto de proyectos aumentan de manera significativa, ya que hay que realizar un mayor número de bucles e iteraciones de datos para poder recopilar y conectar todas las clases y propiedades en cada caso. Esto también es debido a que los datos están separados en tablas independientes pero que, gracias a las vistas preconfiguradas, pueden ser extraídos de un modo más automático.

Una vez se han extraído los datos, comienza el proceso de trabajo para la creación de las propiedades dentro de los modelos de información BIM. En el caso concreto de la creación del archivo de parámetros compartidos en Autodesk Revit, se detectó que no siempre es posible realizar la automatización con Dynamo, ya que depende de la versión de Python y también de Revit que se utilice. El script que se desarrolló se ejecuta correctamente solo si se usa CPython3 y se utilizan versiones de Revit posteriores a 2023. Por otro lado, existe una limitación importante dentro del software relacionada con los grupos de parámetros, no siendo posible la personalización de estos. Es por ello que, tanto con los parámetros compartidos como con la configuración de exportación de los archivos IFC, es necesario realizar un mapeo o conversión de los grupos internos del programa a los que realmente requiere el proyecto de edificación.

En los casos de generación de archivos tipo XML para el trabajo de plantillas dentro de Autodesk Revit, se identificaron diferentes errores durante la generación del archivo derivados de falta de información de identificadores y de algunas conversiones de tipos de datos. Para evitar estos fallos, es importante definir en el código o bien en un archivo auxiliar todas las casuísticas y datos que puedan ser necesarios, como los IDs de las categorías de Revit, los tipos de parámetros permitidos y las agrupaciones de parámetros disponibles en Revit.

En referencia a los scripts de creación de parámetros, se observaron resultados consistentes en la correcta asignación de parámetros a las categorías indicadas, gracias al mecanismo de normalización implementado. Este comportamiento redujo significativamente errores comunes asociados a discrepancias tipográficas, mejorando la robustez del proceso.

Entre los principales hallazgos, destaca la importancia de validar previamente la existencia y nomenclatura exacta de las categorías dentro del proyecto de Revit, ya que la ausencia o mal nombramiento de alguna categoría impedía la vinculación correcta del parámetro, generando advertencias en el flujo de trabajo. Asimismo, la necesidad de manejar correctamente las transacciones en Revit fue fundamental para mantener la integridad del modelo durante las modificaciones paramétricas. Se identificaron algunos puntos críticos a considerar como la dependencia del archivo externo de parámetros compartidos, que debe estar correctamente estructurado y accesible para evitar errores de lectura y la necesidad de garantizar que los nombres de parámetros no se dupliquen, para evitar conflictos en la base de datos del proyecto.

Al agregar propiedades personalizadas a entidades IFC desde Blender mediante scripts, se presentan diversos aspectos críticos que deben ser considerados. En primer lugar, la correcta ejecución del proceso depende en gran medida de la compatibilidad y estabilidad de Bonsai y de la librería IfcOpenShell, ya que cualquier

actualización o cambio en estas herramientas puede afectar la funcionalidad del script. Además, es fundamental mantener la integridad estructural del modelo IFC, ya que modificaciones inadecuadas o propiedades mal definidas pueden generar inconsistencias que dificulten la interoperabilidad con otros sistemas y visores IFC.

La asignación precisa de las clases IFC es otro aspecto crucial, dado que errores o imprecisiones en la identificación de las entidades provocan la falta de datos o asignaciones incorrectas. Asimismo, la creación de PropertySets debe manejarse con cuidado para evitar conflictos en nombres o duplicidades que puedan causar confusión en el procesamiento posterior de la información.

En proyectos de gran escala, se detecta que el rendimiento puede verse comprometido si no se implementan estrategias adecuadas para procesar grandes cantidades de entidades de forma eficiente. Otro punto relevante es la correcta gestión de los tipos de datos, ya que IFC requiere una estricta definición para enteros, reales, cadenas y booleanos, entre otros. Las conversiones incorrectas provocan errores e incompatibilidades. Finalmente, la validación del archivo IFC tras la modificación y exportación es indispensable para asegurar que las propiedades se han integrado correctamente y que el modelo mantiene su consistencia. A pesar de estos retos, la automatización en la incorporación de propiedades personalizadas aporta beneficios significativos, ya que mejora la eficiencia en la gestión de la información y adapta los modelos openBIM a requerimientos específicos del proyecto.

Con respecto a los modelos seleccionados para el estudio, se observan mejoras significativas en la consistencia, coherencia y calidad de los datos. Esto garantiza que todos los modelos sigan los mismos procedimientos y formatos, manteniendo la integridad del diseño y la uniformidad de los datos. Además, el proceso propuesto mejora el cumplimiento de los requisitos, ya que los valores de las propiedades se generan desde la base de datos, a diferencia de los métodos tradicionales BIM, donde la mayoría de los valores son introducidos manualmente por los usuarios.

En el caso de la utilización de bibliotecas preconfiguradas con las propiedades generales de la organización, o bien con las específicas del proyecto, se ha garantizado la exactitud y consistencia de las propiedades, incorporando en cada modelo solo las propiedades que corresponden, dependiendo de los usos, disciplinas y fases especificadas en los requisitos de información para el proyecto. No obstante, se han identificado diferentes restricciones o errores durante los procesos de integración de datos en los modelos BIM, los cuales se enumeran a continuación:

- Errores con celdas combinadas en la plantilla. Al tener celdas combinadas, los valores se distribuyen mal al importar, causando desalineación con los elementos o parámetros.
- Celdas vacías, nulas o con espacios invisibles. Celdas aparentemente vacías pueden contener espacios o saltos que los scripts leen como valores inválidos.
- Saltos de línea dentro de celdas. Texto con saltos de línea generan errores o malinterpretación del contenido.

- Uso de comillas dobles (“ ” o ""). Dependiendo de las comillas tipográficas o múltiples comillas, se puede romper la cadena de texto o impedir su asignación.
- Texto demasiado largo. Algunos parámetros, especialmente tipo Texto, tienen límite de caracteres, generalmente 255. Si se supera este límite, el valor no se asigna y en muchos casos no aparece un aviso del error.
- Codificación de la plantilla. Archivos guardados con codificación incorrecta, como ANSI, pueden generar errores con acentos, ñ o caracteres especiales.
- Fórmulas o celdas con errores. Celdas que contienen fórmulas en programas como Excel, pueden generar null o errores si no son evaluadas correctamente antes de la importación.
- Parámetros con restricciones. Algunos parámetros están bloqueados o restringidos y software como Dynamo no pueden escribir en ellos, aunque el valor sea válido.

Por otro lado, a partir de los subprocesos relacionados con el trabajo en software de autoría BIM, se ha realizado un estudio de resultados obtenidos desde Autodesk Revit y Blender, realizando una comparativa en la que se realiza una estimación en base a los trabajos realizados en la que se define el nivel de complejidad del trabajo manual frente al automatizado que presenta la sistematización desarrollada. En todos los casos, tanto en la configuración inicial, como en el desarrollo y entrega de los modelos BIM, se concluye con una simplificación bastante importante en los trabajos relacionados con datos y gestión de la información dentro de los archivos nativos. Esto se debe al control tan minucioso de la información y a las nuevas automatizaciones desarrolladas a partir de códigos de programación. Entre los resultados obtenidos, destaca notablemente el aporte del nuevo proceso en la configuración y gestión del IFC dentro de Revit durante la generación de los entregables, ya que la exportación se simplifica en gran medida y las configuraciones se realizan de manera automática para realizar los mapeos de datos entre el formato nativo y el abierto.

	Configuración inicial		Desarrollo del trabajo		Generación del entregable final	
	(M)	(A)	(M)	(A)	(M)	(A)
Definición de atributos						
Revit + Dynamo	<i>Medio</i>	<i>Bajo</i>	<i>Alto</i>	<i>Muy Bajo</i>	<i>Medio</i>	<i>Muy Bajo</i>
Blender + Bonsai + Programación	<i>Alto</i>	<i>Medio</i>	<i>Alto</i>	<i>Bajo</i>	<i>Medio</i>	<i>Bajo</i>
Creación de propiedades y cantidades						
Revit + Dynamo	<i>Medio</i>	<i>Bajo</i>	<i>Alto</i>	<i>Bajo</i>	<i>Medio</i>	<i>Muy Bajo</i>
Blender + Bonsai + Programación	<i>Alto</i>	<i>Medio</i>	<i>Alto</i>	<i>Bajo</i>	<i>Medio</i>	<i>Bajo</i>
Definición de clasificaciones						
Revit + Dynamo	<i>Medio</i>	<i>Bajo</i>	<i>Alto</i>	<i>Bajo</i>	<i>Medio</i>	<i>Muy Bajo</i>

	Configuración inicial		Desarrollo del trabajo		Generación del entregable final	
	(M)	(A)	(M)	(A)	(M)	(A)
Blender + Bonsai + Programación	<i>Alto</i>	<i>Medio</i>	<i>Alto</i>	<i>Bajo</i>	<i>Medio</i>	<i>Bajo</i>
Valores de propiedades						
Revit + Dynamo	<i>Medio</i>	<i>Bajo</i>	<i>Alto</i>	<i>Bajo</i>	<i>Medio</i>	<i>Muy Bajo</i>
Blender + Bonsai + Programación	<i>Alto</i>	<i>Medio</i>	<i>Alto</i>	<i>Bajo</i>	<i>Medio</i>	<i>Bajo</i>
Valores de cantidades						
Revit + Dynamo	<i>Medio</i>	<i>Bajo</i>	<i>Alto</i>	<i>Bajo</i>	<i>Medio</i>	<i>Muy Bajo</i>
Blender + Bonsai + Programación	<i>Alto</i>	<i>Medio</i>	<i>Alto</i>	<i>Bajo</i>	<i>Medio</i>	<i>Bajo</i>
Configuración IFC						
Revit + Dynamo	<i>Alto</i>	<i>Medio</i>	<i>Alto</i>	<i>Bajo</i>	<i>Muy Alto</i>	<i>Muy Bajo</i>
Blender + Bonsai + Programación	<i>Medio</i>	<i>Bajo</i>	<i>Medio</i>	<i>Bajo</i>	<i>Medio</i>	<i>Muy Bajo</i>

(M) Manual (A) Automatizado

Tabla 4.1.3. Complejidad operativa dentro de modelos BIM

Otro resultado de la investigación ofrece una visión comparativa sobre las funcionalidades clave de las tres aplicaciones utilizadas: Navisworks, BIMCollab y Bexel. Este análisis revela no solo la existencia o ausencia de ciertas utilidades en cada software, sino también aspectos como la compatibilidad para interoperar con los datos, los formatos soportados, los tipos de archivos utilizados, el número de elementos requeridos y el nivel de complejidad de la implementación de los procesos de automatización.

La carga operativa que representa la generación manual de configuraciones y definiciones de datos dentro de estas aplicaciones es muy alta y conlleva a errores, demoras y falta de estandarización. Sin embargo, automatizar la generación de estas plantillas para las tareas rutinarias no solo permite ahorrar tiempo, sino que también ofrece beneficios para el control y la trazabilidad de la información del proyecto. Estas plantillas, diseñadas en formatos estructurados como XML, XLSX u otros derivados como BCL, pueden ser reutilizadas y compartidas entre proyectos y equipos, fortaleciendo la interoperabilidad y la calidad de los datos.

Utilidad	Software	Existe	Importar/ Exportar	Formato	Tipo de archivo	Items necesarios	Complejidad de código
Búsquedas de objetos	Navis	<i>Sí</i>	<i>Sí</i>	<i>XML</i>	<i>E</i>	<i>3</i>	<i>Baja</i>
	BIMCollab	<i>Sí</i>	<i>No</i>	-	-	-	-
	Bexel	<i>Sí</i>	<i>No</i>	-	-	-	-
Conjuntos de objetos	Navis	<i>Sí</i>	<i>Sí</i>	<i>XML</i>	<i>E</i>	<i>5</i>	<i>Baja</i>
	BIMCollab	<i>No</i>	<i>No</i>	-	-	-	-
	Bexel	<i>Sí</i>	<i>Sí</i>	<i>XLSX</i>	<i>E</i>	<i>5</i>	<i>Muy baja</i>
Tablas de datos	Navis	<i>Sí</i>	<i>Sí</i>	<i>XML</i>	<i>E</i>	<i>3</i>	<i>Baja</i>
	BIMCollab	<i>Sí</i>	<i>Sí</i>	<i>BCL</i>	<i>E</i>	<i>8</i>	<i>Alta</i>

Utilidad	Software	Existe	Importar/ Exportar	Formato	Tipo de archivo	Items necesarios	Complejidad de código
	Bexel	No	No	-	-	-	-
Vistas filtradas	Navis	Sí	Sí	XML	E	6	Alta
	BIMCollab	Sí	Sí	BCSV	E	8	Alta
	Bexel	Sí	Sí	CBSTEM	B	-	-
Incluir datos externos	Navis	Sí	Sí	XLSX	E	>1	Bajo
	BIMCollab	No	No	-	-	-	-
	Bexel	Sí	Sí	XLSX	E	>1	Bajo
Chequeos de datos	Navis	No	No	-	-	-	-
	BIMCollab	Sí	Sí	IDS	E	10	Alta
	Bexel	No	Sí	PMTEM	B	-	-
Test de conflictos	Navis	Sí	Sí	XML	E	6	Alta
	BIMCollab	Sí	Sí	BCR	E	6	Alta
	Bexel	Sí	Sí	XLSX	E	6	Baja
Sistemas 4D y/o 5D	Navis	Sí	Sí	CSV	E	3	Media
	BIMCollab	No	No	-	-	-	-
	Bexel	Sí	Sí	XLSX	E	10	Alta

(E) Estructurado (B) Binario

Tabla 4.1.4. Resultados de conectividad entre BBDD y software

Entre las principales mejoras que aporta la automatización de estas plantillas se han identificado principalmente las que se indican a continuación:

- Estandarización y consistencia en la estructura de datos y procesos entre diferentes proyectos y equipos.
- Reducción significativa de errores humanos, al evitar configuraciones manuales repetitivas.
- Aceleración de tareas operativas, como la generación de conjuntos o filtros, que pueden automatizarse en segundos.
- Facilidad de integración multisoftware, al emplear formatos compatibles con diversas plataformas BIM.
- Control de versiones y trazabilidad, al registrar sistemáticamente los cambios realizados en plantillas y configuraciones.
- Inclusión estructurada de datos externos, permitiendo enriquecer los modelos con información adicional desde otras fuentes.

La trazabilidad de los datos cobra especial importancia en entornos colaborativos donde múltiples disciplinas interactúan. Automatizar la creación de plantillas para tareas como chequeos de datos, pruebas de conflictos o generación de vistas filtradas ha asegurado que cada acción quede documentada y vinculada a parámetros concretos. Además, aunque ciertas funciones muestran una complejidad técnica alta, como la gestión de vistas filtradas o el uso de formatos específicos como IDS o BCSV, otras pueden ser desarrolladas con scripts de bajo nivel de procesamiento. Esto

democratiza el acceso a la automatización, permitiendo que perfiles intermedios en programación puedan generar soluciones prácticas sin depender exclusivamente de especialistas avanzados.

En conjunto, la aplicación de la sistematización, gracias a la suma de los cuatro procesos que define, logra un alto nivel de detalle en comparación con otros proyectos ejecutados durante el mismo tiempo y con metodologías tradicionales. Los datos evolucionan desde las primeras fases del proyecto y siempre se obtienen de las mismas fuentes, por lo que existe una seguridad y confiabilidad muy alta. Además, se facilita la coordinación de agentes, se acorta el tiempo necesario para el desarrollo de la definición de información y se reducen las incoherencias y omisiones relacionadas con trabajos manuales del usuario.

Asimismo, la extracción de información permite visualizar, analizar y combinar los datos del activo de construcción desde diferentes plataformas y entornos, lo cual permite validar datos o realizar cálculos fuera del entorno BIM, a partir del uso de otras herramientas de administración o gestión de proyectos o activos. Cuanto mayor es el proyecto y más son los usos y/o fases requeridas, la gestión de la información más se complica de manera exponencial. A través del uso de los nuevos procesos, el estudio de los datos se simplifica y se gestiona de una manera más ágil.

Si bien las soluciones tradicionales basadas en modelos OpenBIM ofrecen una alta interoperabilidad, existen importantes barreras que dificultan el uso completo de la información contenida en archivos IFC. La nueva propuesta de sistematización permite el flujo de datos entre entornos y sistemas de información a través de tablas relacionales, lo que mejora el análisis de datos tanto a nivel de proyecto como de organización. Además, la creación e intercambio de estándares abiertos como BCF e IDS facilita una mayor interoperabilidad y flexibilidad en la selección de herramientas BIM.

Para verificar la permeabilidad y escalabilidad de la solución, se realizaron comprobaciones aleatorias en cada caso de uso para evaluar la viabilidad del proceso en diversas condiciones, como el filtrado de datos y las nuevas fuentes de conexión. Los hallazgos confirmaron el potencial de crecimiento de la sistematización, destacando la importancia de comprender las nuevas herramientas y su integración con el entorno digital de la organización, a la vez que destacan la facilidad de intercambio de información de la base de datos relacionales con otros sistemas.

Los niveles de automatización alcanzados cumplen con las premisas del estudio, lo que permite la transferencia de datos entre entornos y sistemas, especialmente en los controles de calidad, donde las revisiones se optimizan mediante soluciones automatizadas. Existe potencial de mejora adicional mediante la integración más avanzada de APIs de software o la sincronización automática con la base de datos relacional.

Por otro lado, una vez completados todos los casos de uso, el estudio detectó un ahorro de tiempo significativo en todos los procesos analizados gracias a la optimización de los flujos de trabajo y a la facilidad para definir la información en cada etapa del proyecto. Esto incluye ahorros durante las actualizaciones y modificaciones del proyecto, especialmente en la etapa de integración de datos, donde se minimiza la repetición de trabajos y se reutilizan los procesos. Al establecer procesos de ejecución

claros y sencillos, los usuarios solo necesitan seleccionar los datos necesarios para cada uso, fase y disciplina. Además, el nuevo proceso de control de calidad simplifica enormemente el trabajo del usuario, ya que está totalmente automatizado.

	Proyecto_01		Proyecto_02		Proyecto_03		Proyecto_04	
	(M)	(A)	(M)	(A)	(M)	(A)	(M)	(A)
1. Requisitos de información								
– Definición de tablas manualmente	2 h	-	18 h	-	26 h	-	60 h	-
– Filtrado y agrupación dese la BBDD	-	0.5 h	-	3 h	-	7 h	-	17 h
– Generación de vistas y extracción de datos	-	1 h	-	3 h	-	7 h	-	17 h
– Transformación de datos según software	-	0.4 h	-	1.5 h	-	4 h	-	18 h
2. Integración en modelos BIM								
– Creación de atributos	1 h	0.8 h	1.5 h	1 h	5 h	3 h	40 h	10 h
– Asignación de valores de atributos	1 h	2 h	1.5 h	7 h	5 h	16 h	65 h	50 h
–Configuraciones de exportación IFC	0.5 h	0.5 h	8 h	6 h	20 h	10 h	40 h	20 h
3. Controles de calidad								
– Chequeos de estructura y esquema	1 h	0.9 h	3 h	2 h	10 h	6 h	30 h	17 h
– Chequeos de información de objetos	2 h	1 h	5 h	4 h	15 h	12 h	60 h	30 h
– Generación de informes de revisión	1 h	0.9 h	2 h	1.5 h	5 h	3 h	20 h	13 h
4. Coordinación y consolidación BIM								
– Conjuntos, tablas, vistas de datos, etc.	0.3 h	0.3 h	1 h	0.5 h	4 h	2 h	40 h	13 h
– Actualizaciones y ajustes finales	1.2 h	0.4 h	4 h	1.5 h	31 h	8 h	69 h	32 h
– Integración de datos en la BBDD	-	0.2 h	-	1 h	-	2 h	-	4 h
Total general por proyecto	10 h	8.9	44 h	32	121 h	80	424 h	241h
Ahorro estimado (%)		21%		24%		31%		43%
	(M) Manual		(A) Automatizado					

Tabla 4.1.5. Estimación de ahorro en horas de trabajo

Gestión y control de los datos BIM a lo largo del Ciclo de Vida en Proyectos de Edificación

Los resultados muestran que las automatizaciones y plantillas desarrolladas en la sistematización, junto al uso de bases de datos en estos procesos BIM, permiten una reducción de tiempo significativa, con ahorros del 21 % al 43 % según la complejidad del proyecto. Las mayores mejoras se observan en tareas repetitivas como asignación de valores o trabajos de coordinación y chequeo de la información.

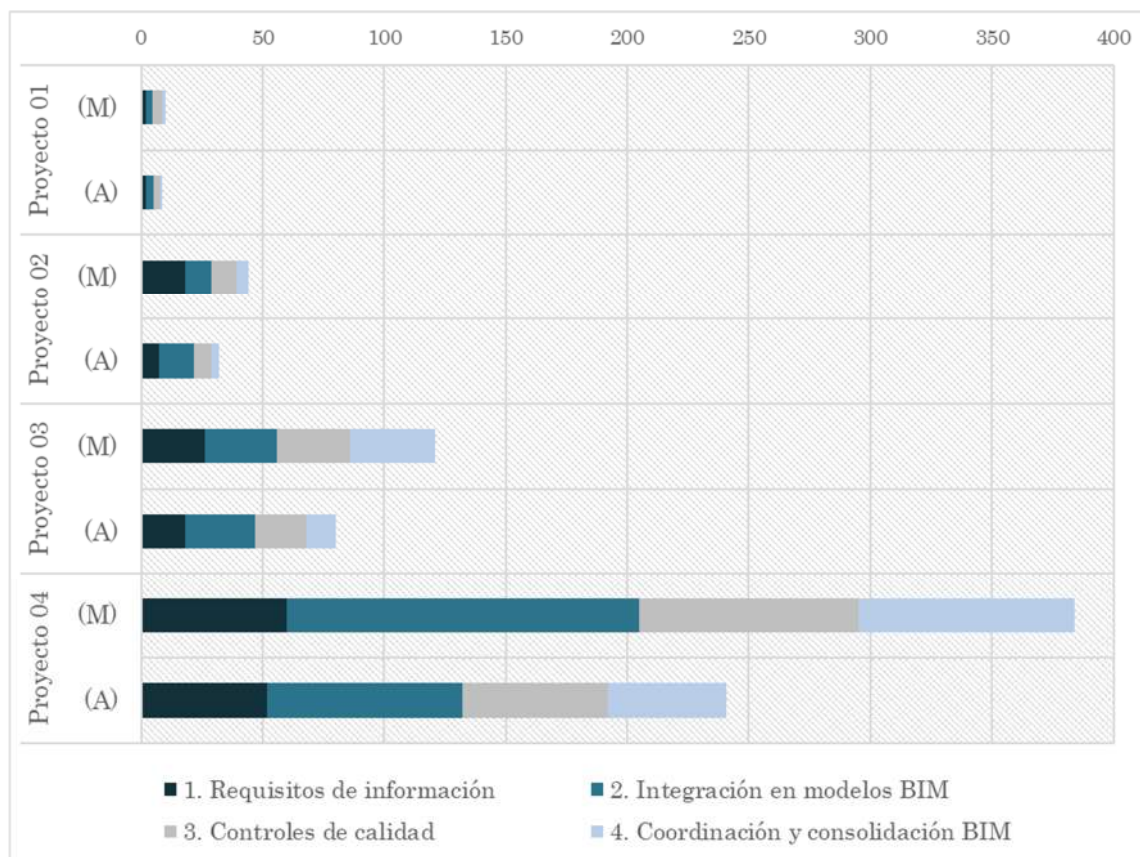


Figura 4.1.2. Ahorro de horas estimadas por proceso y proyecto

Además del cálculo de los tiempos, se realiza un análisis de detección de errores tanto en procesos manuales como en los nuevos procesos automatizados. Estos análisis se basan en los tipos de revisiones de calidad definidos en el anexo E del presente estudio, concretamente en la tabla E.9, denominada *Revisiones de calidad de los modelos BIM*.

En realización a este análisis, se destaca el caso del proyecto 4 con más de 75 mil errores manuales ya que, a medida que los proyectos aumentan en complejidad, también lo hace la cantidad de errores que van siendo detectados en el método tradicional BIM. Si bien los métodos automatizados aún representan una porción de detección que no es total, comienzan a tener un papel más óptimo en los proyectos de mayor escala, especialmente en aspectos estructurados y repetitivos.

Estos hallazgos evidencian la necesidad de fortalecer el uso de herramientas sistematizadas para mejorar la eficiencia del control de calidad, reducir la carga manual y minimizar errores humanos. La implementación temprana de procesos automatizados y una mayor estandarización de los modelos permitirían no solo optimizar los flujos de trabajo, sino también garantizar una mejor calidad informacional BIM.

	Proyecto_01		Proyecto_02		Proyecto_03		Proyecto_04	
	(M)	(A)	(M)	(A)	(M)	(A)	(M)	(A)
Datos generales	0	0	1	0	0	0	5	0
Segregación	0	0	0	0	0	0	0	0
Estructura general	0	0	0	0	3	0	14	0
Elementos de referencia	0	0	0	0	0	0	1	0
Atributos del edificio	0	0	0	0	1	0	12	0
Georreferenciación	0	0	0	0	0	0	0	0
Nomenclatura de instancias	0	0	3	0	125	5	1420	58
Atributos de espacios	-	-	0	0	0	0	2	0
Atributos de objetos	0	0	2	0	2	0	12	0
Grupos y montajes	-	-	0	0	69	3	189	12
Sistemas y subsistemas	-	-	-	-	3	0	8	0
Asociación a niveles	0	0	6	1	55	12	258	84
Organización por clases	0	0	0	0	2	0	5	0
Tipos predefinidos y personalizados	1	0	12	0	37	0	43	0
Subcomponentes	3	0	25	0	301	14	631	79
Materiales y capas	0	0	13	0	47	7	128	16
Contenido y nombrado de conjuntos/psets	-	-	17	0	23	0	29	0
Contenido y nombrado de atributos	0	0	855	25	5874	104	72580	570
Sistemas de clasificación	0	0	0	0	10	0	26	0
Identificadores únicos	0	0	0	0	0	0	0	0
Recursos vinculados	0	0	0	0	0	0	0	0
Total general por proyecto	4	0	934	26	6552	145	75363	819

Tabla 4.1.6. Análisis comparativo de detección de errores

En cuanto al nivel de complejidad que supone incorporar la nueva sistematización a los procesos de la organización, se detecta que el mayor esfuerzo se concentra al inicio del trabajo, cuando deben desarrollarse todas las plantillas, conexiones y automatizaciones de cada etapa. Sin embargo, una vez que se inicia un nuevo proyecto, las tareas se simplifican a actualizaciones y revisiones de mantenimiento, ya que los procesos utilizan los mismos recursos y solo requieren modificaciones menores para integrar nuevos conjuntos de datos.

Todos estos resultados comparativos entre los métodos de trabajo tradicionales BIM y la nueva propuesta de sistematización demuestran mejoras sustanciales en todos los indicadores evaluados. En particular, la propuesta presenta un nivel de

automatización notablemente superior, con mejores calificaciones en todas las unidades funcionales, frente a niveles deficientes o ausentes en los métodos no sistematizados. Estos procesos incrementan la eficiencia, reduciendo los tiempos de ejecución y mejorando la facilidad de uso, especialmente en los procesos de integración de datos y controles de calidad. Además, se evidencia un avance significativo en interoperabilidad y escalabilidad, aspectos clave para proyectos de gran complejidad.

	[U1]		[U2]		[U3]		[U4]	
	Creación de BBDD		Integración de datos		Controles de calidad		Consolidación	
	(Tr)	(Pr)	(Tr)	(Pr)	(Tr)	(Pr)	(Tr)	(Pr)
Nivel de complejidad	●	●●	●	●●	●	●●	●	●●
Nivel de automatización	(-)	●●●	●●	●●●	(-)	●●●●	●●	●●●
Tiempo de ejecución	●●	●●●	●	●●●	●	●●●	●	●●●
Facilidad de uso	●	●●●	●●●	●●●	●	●●●●	●●	●●●
Interoperabilidad	●●	●●●●	●	●●●●	●●	●●●●	●●	●●●●
Escalabilidad	●	●●●●	(-)	●●●	(-)	●●●	(-)	●●●●

(Tr) - Métodos tradicionales BIM (Pr) - Propuesta de sistematización

(-) Sin evidencia ● Deficiente ●● Aceptable ●●● Bueno ●●●● Excelente

Tabla 4.1.7. Análisis comparativo de la propuesta de sistematización

En conjunto, estos resultados validan el impacto positivo de la sistematización propuesta en la gestión de datos BIM, consolidando su aplicabilidad en entornos colaborativos y de alta demanda técnica.

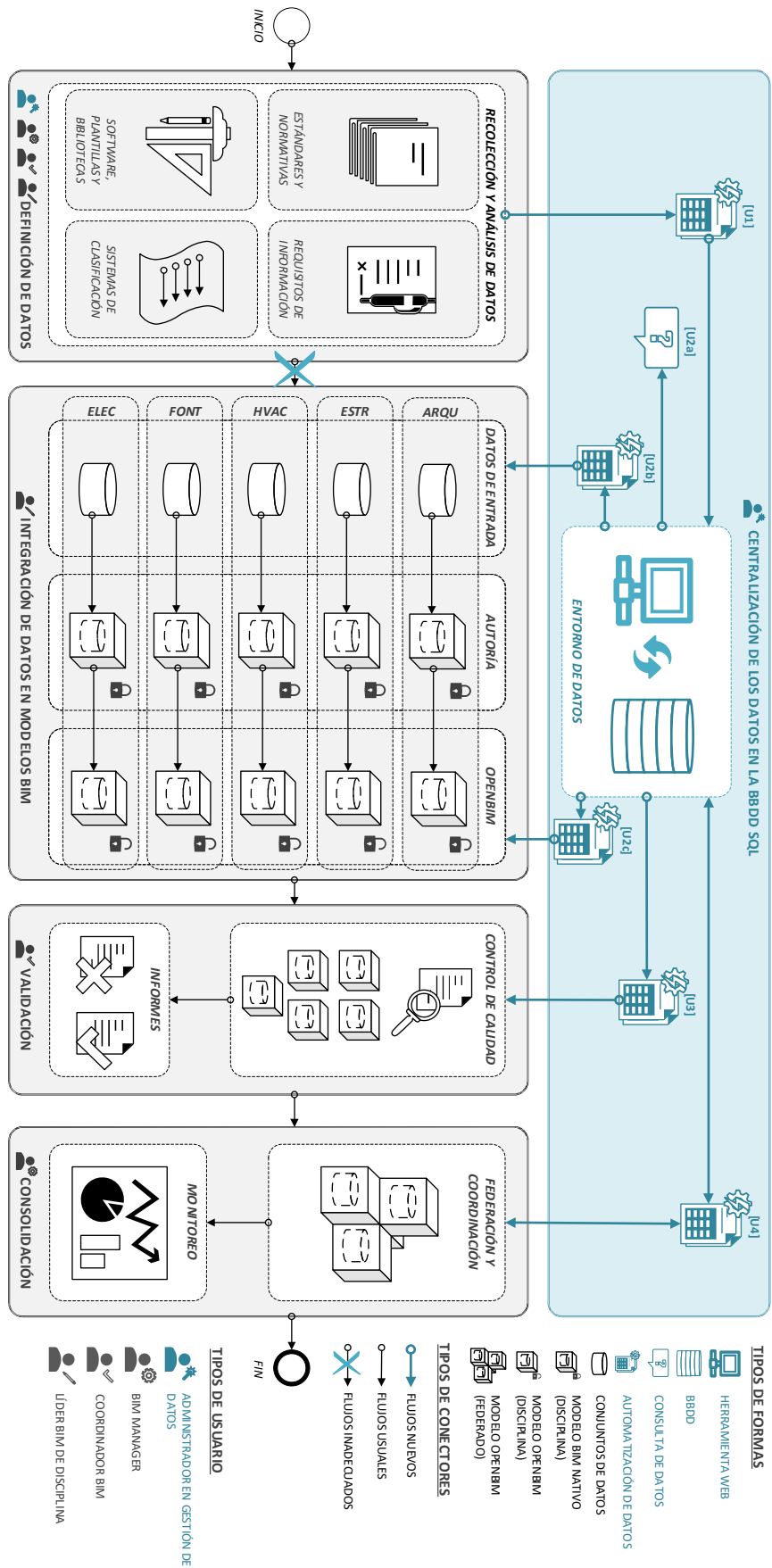


Figura 4.1.3. Propuesta de sistematización para la gestión de activos

4.2. Limitaciones de la investigación

Esta investigación ha estudiado los procedimientos de gestión de información de activos de edificación desde diferentes perspectivas, entornos y fases. Sin embargo, es necesario señalar algunas limitaciones:

- Aunque cada vez existe una mayor interoperabilidad entre los programas y entornos BIM, siguen siendo muchas las limitaciones que tienen la gran mayoría de ellos, no permitiendo intercambiar la totalidad de la información en muchos casos, principalmente a la hora de exportar o importar archivos con formato IFC.
- Esta sistematización no se ha probado exhaustivamente en proyectos reales, lo que supone una limitación parcial del estudio. Si bien se ha desarrollado un marco teórico y un enfoque práctico, la aplicación del sistema en los complejos entornos de los proyectos de construcción reales aún no se ha verificado. Estudios adicionales son esenciales para evaluar su utilidad y recopilar datos consistentes. Además, la retroalimentación de profesionales de diferentes disciplinas proporcionaría información valiosa para el perfeccionamiento, garantizando que la sistematización satisfaga eficazmente las necesidades reales de los usuarios.
- Si bien esta investigación describe las funciones básicas para la gestión y coordinación de proyectos, es necesario un análisis detallado de los niveles de acceso y la manipulación de la información en bases de datos y entornos BIM para definir nuevos tipos de usuarios y permisos avanzados dependiendo del contexto y las necesidades de cada proyecto.
- El estudio de los objetos de construcción se ha centrado en seleccionar clases relevantes, dejando condicionantes relacionados con otras clases, relaciones o propiedades para futuras investigaciones.

A pesar de estas limitaciones e inconvenientes derivados del estudio realizado, los resultados obtenidos son muy positivos, ya que en la gran mayoría de los casos se observan grandes beneficios a partir de implantaciones parciales de las soluciones planteadas.

4.3. Implicaciones teóricas

La presente investigación representa una nueva contribución al conocimiento desde diferentes aspectos teóricos relacionados con el ámbito científico y el de negocio de la línea de investigación desarrollada:

- Entender el potencial del uso de estos nuevos métodos sistemáticos e incorporarlos gradualmente y de manera coordinada dentro de los procesos tradicionales de trabajo BIM es un factor determinante para el desarrollo y crecimiento eficiente de las empresas de construcción que busquen un alto nivel de madurez digital.
- Gracias a la sistematización propuesta se asientan las bases y se abre la puerta al desarrollo de otras propuestas pormenorizadas acerca de estudios

avanzados relacionados con el gobierno del dato y soluciones “data-driven” conectadas a ecosistemas BIM.

- Las aportaciones de la investigación pueden ser aprovechadas por distintos equipos o agentes del sector para ser reformuladas o adaptadas para diferentes casuísticas de tipologías de proyectos, metodologías o para complementarse con otros usos BIM.

4.4. Implicaciones prácticas

La sistematización propuesta mejora significativamente los procedimientos de trabajo BIM tradicionales al optimizar la coordinación entre equipos y fases del proyecto, garantizando que todos trabajen con información actualizada y reduciendo errores y retrasos. Permite un seguimiento más eficiente del progreso al mantener la continuidad digital de los datos, independientemente del software utilizado, lo que evita la pérdida de información durante las transiciones tecnológicas y aumenta la calidad del proyecto.

Además, la centralización de la información de forma estructurada permite que el sistema se escale e integre con otros sistemas de planificación de recursos, gestión de relaciones con los clientes y cadena de suministro relacionados con los activos de construcción.

4.5. Grado de validez y fiabilidad del estudio

Existe un sesgo en el estudio realizado debido a la exclusión de proyectos de infraestructuras de obra civil, industriales y de energía. Para la incorporación de la sistematización a esta tipología de proyectos, otros estudios deberían ser desarrollados para garantizar la trazabilidad, conectividad y coherencia de relaciones, jerarquías y estructuras de datos.

En cuanto a las entidades, relaciones, propiedades y contenedores de información trabajados, se ha tenido especial cuidado en respetar las estructuras de datos de los diferentes formatos, así como garantizar la correcta transmisión de información entre BBDD y archivos BIM.

Otro tema acerca de la fiabilidad en la implementación de la solución propuesta es que sería conveniente un análisis a partir de su aplicación en proyectos piloto en los que se trabajen los conjuntos de datos en todas las etapas del proyecto de manera coordinada por todas las partes o equipos, potenciando la colaboración y continuidad digital y validando de una manera más realista los resultados obtenidos con los casos de uso. No obstante, tanto los modelos BIM como las bases de datos y soluciones propuestas simulan casos de aplicación que podrían darse en la realidad.

Por último, la validez de la solución se ha comprobado en terminados flujos y programas lo que puede desencadenar ciertas limitaciones a la hora de ser implementado en otros entornos de trabajo dentro o fuera de las plataformas BIM. De igual forma, versiones anteriores o posteriores de los softwares, o incluso, de los formatos utilizados, pueden generar inconsistencias en las soluciones desarrolladas en la investigación, por lo que un control periódico es siempre necesario para garantizar la actualización de todos los procesos y recursos generados.

CAPÍTULO 5: CONCLUSIONES Y FUTURAS INVESTIGACIONES

5.1. Conclusiones parciales

El marco general propuesto para la sistematización de esta investigación se conforma de diferentes subprocesos que establecen las actividades específicas que deben seguirse para el desarrollo de las diferentes tareas relacionadas con la creación, coordinación, gestión y validación de los datos de construcción. Las conclusiones específicas de estos subprocesos de trabajo se detallan a continuación.

Centralización de la información en BBD

La implementación de una base de datos relacional como núcleo central de gestión de la información de los activos constructivos de la organización representa una innovación estructural en la digitalización del entorno construido. Más allá de su función de almacenamiento, esta arquitectura posibilita el modelado del conocimiento técnico vinculado al activo, permitiendo estructurar de forma semántica tanto los atributos como las relaciones funcionales entre elementos. La capacidad de modelar entidades interdependientes en un entorno relacional permite que los datos adquieran contexto, jerarquía y trazabilidad en el tiempo, lo cual es esencial para una gestión orientada al ciclo de vida.

Desde una perspectiva operativa, el uso de bases relacionales permite desacoplar los procesos de generación de datos de su análisis, lo que abre nuevas posibilidades para crear flujos de trabajo asincrónicos, escalables y distribuibles entre diferentes actores y sistemas. Además, al incorporar funciones avanzadas como vistas personalizadas, índices complejos y reglas de integridad referencial, se garantiza una mayor robustez y fiabilidad en el manejo de grandes volúmenes de información.

A nivel estratégico, esta centralización basada en esquemas relacionales facilita el desarrollo de datos conectados. Así, la base de datos se convierte en un sistema vivo de información, que no solo responde al presente del activo, sino que es capaz de registrar su evolución e incluso de mejorar los procesos operativos de la organización.

Extracción e integración de los datos en modelos BIM

Este proceso constituye un componente esencial para garantizar la coherencia, precisión y utilidad de la información digital a partir de plantillas estructuradas y flujos de trabajo automatizados, permitiendo introducir datos tabulares y relaciones de entidades dentro del modelo BIM, manteniendo la integridad semántica y técnica de la información.

El uso de plantillas de integración personalizadas actúa como puente entre la lógica de datos relacionales y las necesidades específicas de los modelos multidisciplinares, pudiendo adaptarse a distintos usos y controlando la correspondencia entre los campos de la base de datos y los atributos del modelo, lo cual se ha demostrado que minimiza errores en la transferencia de información, reduce los tiempos de configuración manual y estandariza el proceso de incorporación de datos a lo largo de diferentes proyectos.

Además, las automatizaciones implementadas en este marco permiten escalar los procesos de carga, inserción y exportación de datos en múltiples modelos simultáneamente, sin intervención humana constante. Asimismo, posibilitan la aplicación reglas predefinidas relacionadas con disciplinas, fases y usos, que mejoran

la eficiencia de las tareas y reducen la carga operativa de los modeladores y coordinadores BIM. Esta capacidad de operar sobre grandes volúmenes de datos de forma automática constituye un avance clave frente a metodologías tradicionales basadas en edición manual y tareas repetitivas.

Por otro lado, los procesos optimizados de integración facilitan la trazabilidad inversa, es decir, la capacidad de verificar que los datos cargados en el modelo se correspondan con los registros originales de la base de datos. Esta trazabilidad bidireccional asegura la fidelidad del modelo respecto al sistema de información central, consolidando el modelo BIM como una representación confiable del activo.

Al construir un sistema en el que los datos fluyen desde una fuente común y se insertan en los modelos bajo criterios uniformes y replicables, se facilita la colaboración entre disciplinas, se mejora la transparencia del proceso constructivo y se potencia el uso estratégico del modelo como herramienta de gestión. Así, se avanza hacia un enfoque verdaderamente basado en datos en el ámbito del modelado y la gestión de información de la construcción.

Controles de calidad de los modelos BIM

La incorporación de mecanismos de control de calidad automatizados, basados en datos extraídos directamente desde la base de datos relacional, representa una evolución significativa en los procesos de validación de modelos BIM. Esta estrategia garantiza la alineación de la información contenida en los modelos con los requisitos definidos previamente por la organización, al mismo tiempo que optimiza tiempos y recursos destinados a tareas de verificación.

Las plantillas de chequeo, estructuradas en función de los parámetros de calidad establecidos en la base de datos, actúan como instrumentos dinámicos que trasladan exigencias específicas a cada modelo o disciplina. Estas plantillas cumplen con diferentes objetivos de validación, como la revisión de nomenclaturas, el control de propiedades obligatorias, la existencia de relaciones espaciales, o la verificación de tipos de objetos y estructuras de clasificación.

Este enfoque también ha permitido desarrollar procesos sistemáticos de verificación que se actualizan en función de los cambios en los criterios de proyecto, sin necesidad de reconfiguración desde cero. Así, los criterios de control evolucionan al mismo ritmo que la base de datos central, manteniendo la coherencia entre los modelos y las decisiones organizativas o regulatorias que les afectan.

Además, al almacenar y gestionar los resultados de las comprobaciones dentro del propio entorno relacional, ha sido posible generar indicadores cuantitativos de calidad informativa, establecer el seguimiento del cumplimiento del modelo, e incluso identificar patrones de error recurrentes asociados a equipos, herramientas o procesos. Esto habilita una retroalimentación continua para la mejora de la calidad informativa a nivel organizacional.

Consolidación de la información en la BBD

Por último, la consolidación de la información en la base de datos relacional, mediante la extracción e importación sistemática de los datos provenientes de los modelos BIM y de otras bases y recursos de construcción, permite establecer un entorno de gestión integral en el que convergen diversas fuentes de información

técnica de la totalidad de los proyectos de la organización. Este enfoque posibilita la agregación estructurada de los datos del modelo y también su combinación con información externa, como cronogramas, presupuestos o datos de mantenimiento, generando una visión transversal y contextualizada del activo constructivo.

El estudio llevado a cabo destaca la importación de la automatización desde los modelos hacia la base de datos, asegurando la sincronización constante entre los entornos de modelado y los sistemas de gestión, evitando desalineaciones entre el diseño digital y la documentación del proyecto. Esta bidireccionalidad en el flujo de datos facilita una supervisión más exhaustiva de los avances, el cumplimiento de requisitos y la evolución de los elementos a lo largo de las etapas del proyecto.

Una vez consolidada, la información se convierte en una fuente clave para su representación visual mediante cuadros de mando interactivos, configurados con indicadores personalizables según los intereses de los distintos agentes implicados.

Desde una perspectiva organizacional, esta consolidación habilita el diseño de modelos de gobernanza informativa, en los que los datos no se gestionan de forma aislada o fragmentada, sino como parte de un ecosistema unificado que respalda la gestión inteligente de los activos. Este nivel de integración y visualización convierte a la base de datos relacional en un componente clave del ecosistema de datos, sirviendo como punto de referencia para consultas, auditorías y reportes periódicos.

5.2. Conclusiones generales

Las conclusiones del presente estudio aportan una visión integral acerca de la necesidad de transformar los procesos tradicionales de gestión de información en proyectos de edificación mediante el uso y aplicación de metodologías digitales avanzadas. A partir de los objetivos formulados al inicio de la investigación, se ha demostrado que la integración entre modelos BIM y bases de datos relacionales representa una vía eficaz para estructurar, estandarizar y controlar los flujos de datos que atraviesan las distintas fases del ciclo de vida de un activo constructivo. Esta aproximación permite optimizar la productividad operativa y también incrementar la calidad y la trazabilidad de la información gestionada.

El entorno actual de la edificación se caracteriza por una elevada fragmentación técnica y por la participación simultánea de múltiples agentes, cada uno con diferentes niveles de conocimiento y adopción tecnológica. Esta situación, sumada a la evolución acelerada de las herramientas digitales y a la proliferación de formatos propietarios y estructuras de datos heterogéneas, plantea serios desafíos para la interoperabilidad y la gestión eficiente de la información. Frente a este contexto, la presente investigación ha establecido un marco metodológico sólido para conectar y armonizar estos datos, independientemente del software de origen, lo cual resulta fundamental para garantizar la coherencia semántica y técnica en entornos de trabajo colaborativos.

El principal logro de la investigación ha sido la definición de una sistematización agnóstica de los procesos, que posibilita una elevada adaptabilidad y escalabilidad de las soluciones propuestas. Al desvincular la gestión de datos de herramientas específicas, se favorece una estrategia digital abierta que puede ser implementada en organizaciones de distinta naturaleza y tamaño. La conectividad entre bases de

datos y modelos openBIM, soportada en estándares como el IFC y en el uso de identificadores únicos, permite una actualización automática y controlada de los datos, minimizando los riesgos asociados a modificaciones no sincronizadas, errores humanos o duplicidades informativas. Además, la flexibilidad de la sistematización permite a las empresas llegar hasta el nivel de complejidad que considere, y solo utilizar las herramientas y soluciones que necesite, dependiendo de las fases en las que se trabaje, las disciplinas y los usos de aplicación BIM.

Asimismo, se constata que la gestión anticipada y estructurada de los datos, incluso antes de la firma de contratos con proveedores o de la elaboración de los modelos, es un factor crítico para el éxito de los proyectos. En este sentido, la estandarización de formatos, nomenclaturas, propiedades y relaciones se presenta como condición fundamental para alcanzar un entorno digital interoperable. Esta planificación temprana facilita también la validación automatizada de requisitos técnicos, la verificación de cumplimiento de estándares y la detección de inconsistencias estructurales desde las fases iniciales del proyecto.

Por otro lado, la centralización de datos desarrollada no implica una unificación de herramientas, sino más bien la creación de un repositorio global, accesible, conectado y en constante actualización, que actúa como núcleo informacional de todos los sistemas. Gracias a esta visión holística, los datos pueden ser reutilizados en diferentes proyectos o fases, generando conocimiento avanzado de los datos de construcción. Además, a través de plantillas preconfiguradas y procedimientos automatizados, los datos pueden ser filtrados, integrados y transformados para cumplir funciones analíticas, logísticas o de seguimiento, sin necesidad de conocimientos avanzados en modelado BIM por parte de los usuarios.

El impacto positivo de estos procesos se refleja no solo en la mejora de la eficiencia operativa, sino también en la calidad de las entregas digitales y en la capacidad de respuesta ante cambios o imprevistos. La posibilidad de que cualquier modificación en una disciplina se propague automáticamente a través del sistema evita descoordinaciones entre equipos, sobrecostos y retrasos, problemáticas recurrentes en la gestión tradicional de proyectos. Esta resiliencia técnica, soportada en una arquitectura de datos conectada, representa una ventaja competitiva clave para las organizaciones del sector AECO.

Adicionalmente, el enfoque desarrollado promueve una progresión gradual en el nivel de madurez digital de los agentes involucrados.²¹³ Con estos nuevos procesos se superan los niveles básicos relacionados únicamente con revisiones visuales y análisis manuales, consiguiendo llegar en muchos casos de uso a un nivel híbrido de madurez, en el que los valores de las propiedades BIM se utilizan activamente para la verificación de códigos específicos relacionados con falta de datos o estructuras requeridas, con la existencia o número de objetos solicitados, relaciones espaciales o geométricas, cálculos o análisis de datos y relaciones BIM. En otros casos, el estudio desarrollado logra alcanzar un nivel automatizado, en el que la totalidad de los datos de los modelos son revisados, incluyendo análisis más avanzados relativos a la consistencia y la exactitud de los datos generados en el conjunto de archivos IFC. Esta evolución metodológica mejora la fiabilidad de los modelos, habilitando su uso

²¹³ e-submission common guidelines for introduce BIM to building process. Technical Report. Masaki Muto. buildingSmart Japan. 2020.

en ámbitos estratégicos como la gestión de proyectos o la evaluación del rendimiento de los activos de edificación.

Finalmente, los resultados obtenidos refuerzan la idea de que la metodología BIM debe consolidarse como una base estructural para el desarrollo de procesos colaborativos, pero también que debe evolucionar mediante la incorporación de nuevas capas de inteligencia de datos, conectividad entre plataformas, y automatización de tareas. Solo mediante una adopción estratégica, progresiva y adaptativa de estas tecnologías será posible enfrentar los desafíos emergentes del sector, mejorar la toma de decisiones basada en datos, y garantizar la eficiencia de los activos construidos a lo largo de su vida útil.

5.3. Futuras líneas de investigación

A partir del presente trabajo, se abren diversas líneas de investigación orientadas a mejorar la interoperabilidad y la persistencia de datos entre bases de datos relacionales y modelos openBIM. La lógica de que la información técnica se origine en una base de datos estructurada, evolucione durante las diferentes fases del proyecto y, gradualmente, se sincronice con los datos del modelo BIM, plantea desafíos tecnológicos, semánticos y metodológicos. Las siguientes líneas proponen avances relevantes para fortalecer esta sincronización durante todo el ciclo de vida del activo de edificación.

Una nueva línea de investigación puede estar centrada en los costos de implementar sistemas de información centralizados como el del presente estudio, incluyendo la inversión inicial, el mantenimiento y el potencial de ahorro derivado de una mayor eficiencia. Además, es crucial examinar la resistencia al cambio de las partes interesadas, ya que los estudios podrían explorar los factores psicológicos y organizativos que la sustentan e identificar estrategias para facilitar la aceptación de nuevas tecnologías. Estos futuros estudios también podrían buscar una mayor exploración de los procedimientos establecidos a partir del uso de otros software y entornos BIM, así como nuevas propuestas para la conectividad de la sistematización con otras tecnologías emergentes.

La conexión con otros sistemas de bases de datos de tipo objeto-relacional, como PostgreSQL, puede generar ventajas sustanciales respecto a las bases del estudio en términos de extensibilidad, modelado de datos complejos y rendimiento para operaciones analíticas. Su soporte para tipos de datos complejos permite almacenar conjuntos de datos semi-estructurados, facilitando la vinculación de entidades openBIM heterogéneas con otros tipos de información. Además, contiene algunos módulos avanzados que permiten la manipulación de datos espaciales y topológicos directamente dentro de la base de datos, lo cual es altamente relevante para modelos IFC con información georreferenciada. Investigaciones futuras podrían centrarse en evaluar el rendimiento de consultas espaciales sobre estructuras IFC importadas a PostgreSQL, establecer esquemas de sincronización o comparar la eficiencia de transacciones frente a motores como MySQL o MariaDB. Asimismo, el mapeo automático de archivos IFC a grafos puede ofrecer una forma más avanzada de representar las complejas relaciones topológicas y jerárquicas de los modelos BIM.

Otra línea futura de investigación surge a partir del IFC5 y su potencial sinergia con tecnologías emergentes como el ECS (Entity Component System) y USD (Universal

Scene Description). Esta línea se centraría en cómo estos paradigmas de modelado y serialización podrían potenciar la sincronización y gestión de datos en todo el ciclo de vida del activo, permitiendo una integración aún más fluida y eficiente entre sistemas relacionales y modelos openBIM. La adopción de ECS, con su estructura modular y granular basada en entidades y componentes, permitiría redefinir la forma en que los objetos BIM se representan y actualizan desde bases de datos SQL, abriendo la puerta a mecanismos más dinámicos y escalables de actualización bidireccional. Asimismo, la incorporación de USD facilitaría la interoperabilidad multidisciplinar y la composición de datos de diversas fuentes, integrando modelos geométricos y semánticos en una estructura más ágil, orientada a flujos colaborativos.

Una integración pormenorizada entre los conjuntos de datos BIM y los sistemas empresariales como ERP (Enterprise Resource Planning), CMS (Construction Management Systems) y MES (Manufacturing Execution Systems) podría abrir nuevas posibilidades para gestionar el ciclo de vida completo del activo. Esta sinergia requeriría el establecimiento de una ontología común o puentes semánticos entre los sistemas, ya que los datos que gestiona un ERP, por ejemplo, no están estructurados de la misma forma que los de un modelo BIM. Los estudios podrían explorar mecanismos de ETL (Extract, Transform, Load) adaptativos para sincronizar datos entre los distintos entornos, buscando enriquecer los modelos IFC con información de costes y pedidos, mantenimiento planificado, inventario de activos, entre otros.

El uso de inteligencia artificial, y en particular el aprendizaje automático, puede aportar mecanismos de análisis predictivo, clasificación y validación automática de datos tanto en modelos BIM como en bases de datos relacionales. Una línea de investigación consiste en el desarrollo de algoritmos que permitan, por ejemplo, detectar incoherencias entre el modelo IFC y los datos almacenados en SQL, generar predicciones de comportamiento del activo durante su operación, o realizar estimaciones automáticas de mantenimiento basadas en datos históricos. Por otro lado, técnicas de procesamiento más avanzadas pueden ayudar a mapear descripciones textuales en el modelo BIM a entidades estructuradas de la base de datos. El desarrollo de estos sistemas requiere definir estructuras de entrenamiento, validación y métricas de precisión aplicables a modelos IFC.

Por último, la conexión con gemelos digitales, donde un modelo BIM se pueda conectar en tiempo real a sensores y sistemas de monitoreo, permitiría que el modelo no solo represente el diseño, sino que también refleje el estado operativo del activo en cada momento. La base de datos SQL puede actuar como repositorio estructurado donde se almacena el histórico de datos IoT vinculado directamente con las entidades BIM. Futuras investigaciones podrían analizar modelos de arquitectura en los que las entidades IFC funcionen como nodos semánticos en los que se agregan datos dinámicos, y cómo estas actualizaciones pueden mantenerse sincronizadas entre el modelo BIM y la base de datos. Asimismo, se pueden estudiar técnicas de visualización en tiempo real y análisis predictivo usando los datos recopilados por sensores durante la operación del activo.

Referencias bibliográficas

- Abanda, F., Balu, B., Adukpo, S., & Akintola, A. (2025). Decoding ISO 19650 Through Process Modelling for Information Management and Stakeholder Communication in BIM. *Buildings*, 15(3)10.3390/buildings15030431
- Aghimien, D., Aigbavboa, C., Oke, A., & Koloko, N. (2018). Digitalisation in Construction Industry: Construction Professionals Perspective. *Proceedings of International Structural Engineering and Construction*, 510.14455/ISEC.res.2018.90
- Ahankoob, A., Manley, K., & Abbasnejad, B. (2019). The Role Of Contractors' Building Information Modelling (BIM) Experience In Realising The Potential Values of BIM. *International Journal of Construction Management*, , 1–12. 10.1080/15623599.2019.1639126
- Ait, L. e. a. (2021). Extending the IFC Standard to Enable Road Operation and Maintenance Management through OpenBIM. *ISPRS International Journal of Geo-Information*, 10, 496. 10.3390/ijgi10080496
- al Omari, R., Sweis, G., Abu-Khader, W., & Sweis, R. (2023). Barriers to the Adoption of Digitalization in the Construction Industry: Perspectives of Owners, Consultants, and Contractors. *Construction Economics and Building*, 23, 87–106. 10.5130/AJCEB.v23i3/4.8636
- Alzraiee, H. (2020). Cost Estimate System Using Structured Query Language in BIM. *International Journal of Construction Management*, 22, 1–13. 10.1080/15623599.2020.1823061
- Annunen, P., Tella, J., Pekki, S., & Haapasalo, H. (2024). Maintenance Capability Creation For Buildings – Concurrent Process With Design And Construction. *Journal of Facilities Management*, 22(3), 479–496. 10.1108/JFM-05-2022-0052
- Atuahene, B., Kanjanabootra, S., & Gajendran, T. (2018). Towards an Integrated Framework Of Big Data Capabilities In The Construction Industry: A Systematic Literature Review. *Conference: 34th Association of Researchers in Construction Management (ARCOM)*.
- Balakrishnan Selvakumaran, S., & Hall, D. M. (2022). From Crowd To Cloud: Simplified Automatic Reconstruction Of Digital Building Assets For Facility Management. *Journal of Facilities Management*, 20(3), 401–436. 10.1108/JFM-02-2021-0017
- Bello, S., Oyedele, L., Akinadé, O., Bilal, M., Davila Delgado, M., Akanbi, L., Anuoluwapo, O., & Owolabi, H. (2020). Cloud Computing In Construction Industry: Use Cases, Benefits And Challenges. *Automation in Construction*, 122, 103441. 10.1016/j.autcon.2020.103441
- Berlo, L. v., Willems, P., & Pauwels, P. (2019). Creating Information Delivery Specifications Using Linked Data. *CIB W78 Conference on Technology*.

- Borrmann, A., Beetz, J., Koch, C., Liebich, T., & Muhic, S. (2018a). Industry Foundation Classes: A Standardized Data Model for the Vendor-Neutral Exchange of Digital Building Models. *Building Information Modeling - Technology Foundations and Industry Practice* (pp. 81–126) 10.1007/978-3-319-92862-3_5
- Borrmann, A., König, M., Koch, C., & Beetz, J. (2018b). Building Information Modeling: Why? What? How?: *Technology Foundations and Industry Practice*. (pp. 1–24)10.1007/978-3-319-92862-3_1
- Bosch-Sijtsema, P., Gluch, P., & Sezer, A. A. (2019). Professional Development of the BIM Actor Role. *Automation in Construction*, 97, 44–51. 10.1016/j.autcon.2018.10.024
- Bosch-Sijtsema, P., Claeson-Jonsson, C., Johansson, M., & Roupe, M. (2021). The Hype Factor Of Digital Technologies in AEC. *Construction Innovation*, 21(4), 899–916. 10.1108/CI-01-2020-0002
- Bryde, D., Broquetas, M., & Volm, J. M. (2013). The Project Benefits of Building Information Modelling (BIM). *International Journal of Project Management*, 31(7), 971–980. 10.1016/j.ijproman.2012.12.001
- buildingSMART International. (2020). The Relation Between IFC and bSDD. buildingSMART International Policy Statement.
- Cepa, J., Muñoz Pavón, R., Alberti, M., & Caramés, P. (2023). Towards BIM-GIS Integration For Road Intelligent Management System. *Journal of Civil Engineering and Management*, 29, 621–638. 10.3846/jcem.2023.19514
- CEROVSEK, T., & Omar, M. (2024). Advancements and Gaps in Information Compliance Checking (ICC) Automation with Information Delivery Specification (IDS). *Proc. of the CIB W78 Conference 2024*.
- Chen, B., Liu, A., & Hua, Y. (2017). An Exploration Of The Interaction Between BIM Technology And The Business Process Of A Construction Organization In BIM Implementation. *WIT Transactions on The Built Environment*. 10.2495/BIM170171
- Chen, S. (2023). Digital Transformation in the Construction Industry: A Systematic Literature Review. *Engineering Project Organization Conference (EPOC)*. 10.1145/3629378.3629431
- Deng, X., Huahui, L., Xu, J., & Zhao, Y. (2020). Generic Language for Partial Model Extraction from an IFC Model Based on Selection Set. *Applied Sciences*, 10, 1968. 10.3390/app10061968
- Du, J., Liu, R., & Issa, R. R. A. (2014). BIM Cloud Score: Benchmarking BIM Performance. *Journal of Construction Engineering and Management*, 140(11)10.1061/(asce)co.1943-7862.0000891
- El Arass, M., & Souissi, N. (2018). Data Lifecycle: From Big Data to Smart Data. *IEEE 5th International Congress on Information Science and Technology (CiSt)*.

- Emunds, C. e. a. (2021). IFCNet: A Benchmark Dataset for IFC Entity Classification. EG-ICE 2021.
- Ershadi, M., & Lijauco, F. (2024). A Framework For Digitalizing Construction Businesses: Drivers, Barriers And Outcomes. Smart And Sustainable Built Environment, 10.1108/SASBE-01-2024-0003
- Figueiredo Muller, M., Garbers, A., Esmanioto, F., Huber, N., Rocha Loures, E., & Cancigliieri Junior, O. (2017). Data Interoperability Assessment Though IFC for BIM in Structural Design – a Five-Year Gap Analysis. Journal of Civil Engineering and Management, 23, 943–954. 10.3846/13923730.2017.1341850
- Forcael, E., Ferrari, I., Opazo-Vega, A., & Pulido Arcas, J. (2020). Construction 4.0: A Literature Review. Sustainability, 12, 9755. 10.3390/su12229755
- Fountain, J., & Langar, S. (2018). Building Information Modeling (BIM) Outsourcing Among General Contractors. Automation In Construction, 95, 107–117. 10.1016/j.autcon.2018.06.009
- Garber, R. (2014). BIM Design: Realising the Creative Potential of Building Information Modelling. 10.1002/9781118914694
- Gigante-Barrera, A., Crunden, M., & Ruikar, K. (2017). LOD Object Content Specification For Manufacturers Within The UK Using The IDM Standard. Journal of Information Technology in Construction, 22, 80–103.
- Godager, B., Mohn, K., Merschbrock, C., Klakegg, O., & Huang, L. (2022). Towards an Improved Framework for Enterprise BIM: the Role of ISO 19650. Journal of Information Technology in Construction, 27, 1075–1103. 10.36680/j.itcon.2022.053
- Golestanizadeh, M., Sarvari, H., Chan, P. D., Banaitiene, N., & Banaitis, A. (2023a). Opportunities In Application Of Business Intelligence In Construction Companies. Journal of Civil Engineering and Management, 29, 487–500. 10.3846/jcem.2023.19533
- Han, Y. Q., & Yu, J. J. (2018). Benefit Evaluation Research of BIM Application. Paper presented at the 867–871. 10.25236/ssehr.2018.174
- Hernán, J. (2021). La Implantación de BIM en España = The Implementation of Bim in Spain. Building & Management, 4, 4. 10.20868/bma.2020.1.4661
- Hewavitharana, F. S. T., & Perera, A. (2018). Sustainability via ERP and BIM Integration. Paper presented at the 44 202–210. 10.1007/978-981-13-9749-3_19
- Hjelseth, E. (2017). BIM Understanding and Activities. WIT Transactions on the Built Environment. 10.2495/BIM170011
- Holzer, D. (2015). BIM for Procurement - Procuring for BIM. Paper presented at the 237–246.
- Hooper, M. (2015). BIM Standardisation Efforts - The Case Of Sweden. Journal of Information Technology in Construction, 20, 332–346.

- Huahui, L., & Deng, X. (2018). Interoperability Analysis of IFC-Based Data Exchange between Heterogeneous BIM software. *Journal of Civil Engineering and Management*, 24, 537–555. 10.3846/jcem.2018.6132
- Hui, S. C. M. (2018). From BIM to VDC: Strategies for Innovation and Transformation. *Joint Symposium 2018: Innovative Technology Fusion for Next Challenging Century*.
- Jaganathan Manonmani, A. (2025). *Data Acquisition Fundamentals And Methodology Data Acquisition Fundamentals And Methodology*.
- Jeon, K., & Lee, G. (2018). Information Delivery Manual (IDM) Configurator: Previous Efforts and Future Work. *18th International Conference on Construction Applications of Virtual Reality*.
- Jiang, S., Jiang, L., Han, Y., Wu, Z., & Wang, N. (2019). OpenBIM: An Enabling Solution for Information Interoperability. *Applied Sciences*, 9(24)10.3390/app9245358
- Kang, T., & Choi, H. (2015). BIM Perspective Definition Metadata For Interworking Facility Management Data. *Advanced Engineering Informatics*, 29(4), 958–970. 10.1016/j.aei.2015.09.004
- Karlapudi, J. (2020). A Methodology To Classify And Determine Data Sharing Requirements Between Openbim Models And Energy Simulation Models. *Master of Science*. 10.13140/RG.2.2.26167.27041/1
- Kim, J. U., Kim, Y. J., Ok, H., & Yang, S. H. (2015). A Study on the Status of Infrastructure BIM and BIM Library Development. Paper presented at the 857–858. 10.1109/csci.2015.52
- Kralj, G. (2018). BIM in Infrastructure. *5th International Conference on Road and Rail Infrastructure (CETRA)*. Paper presented at the 1027–1033. 10.5592/co/cetra.2018.732
- Krijnen, T. & L., T. (2021). *Future of the Industry Foundation Classes: towards IFC 5. Urban Data Science*.
- Lammers, B., & Forth, K. (2022). IFC-Based Variant Analysis Considering Multi-Criterial Sustainability Analysis of Buildings. *Foruma Bauinformatik*.
- Lange, H., Johansen, A., & Kjærgaard, M. (2018). Evaluation of the Opportunities and Limitations of Using IFC Models as Source of Building Metadata. *BuildSys '18: Proceedings of the 5th Conference on Systems for Built Environment*. 10.1145/3276774.3276790
- Lawrence, M., Pottinger, R., Staub-French, S., & Nepal, M. P. (2014). Creating flexible mappings between Building Information Models and cost information. *Automation in Construction*, 45, 107–118. 10.1016/j.autcon.2014.05.006
- Lee, G., & Borrmann, A. (2020). BIM Policy and Management. *Construction Management and Economics*, 38, 413–419. 10.1080/01446193.2020.1726979

- Lee, J., Lee, H., Park, M., & Kim, W. (2016). An Interoperable Coordination Method for Sharing Communication Information Using BCF (BIM Collaboration Format). Construction Research Congress. 10.1061/9780784479827.243
- Leygonie, R., & Motamedi, A. (2021). Quality Management Processes and Automated Tools for FM-BIM Delivery. Proceedings of the 38th ISARC, Dubai, UAE. 10.22260/ISARC2021/0120
- Lino, J., Djukic, A., & Caires, B. (2024). Definição e controlo dos requisitos de informação BIM. UMinho Editora. 10.21814/uminho.ed.164.37
- Liu, L., Hagedorn, P., & König, M. (2023). Definition of a container-based machine-readable IDM integrating level of information needs. Proceedings of the 2023 European Conference on Computing in Construction and the 40th International CIB W78. 10.35490/EC3.2023.221
- Lopez, A. E. (2017). BIM as a Chage Driver in Public Organizations. 2nd International Conference on Building Information Modelling, 169 169–175. 10.2495/bim170161
- Lou, J., Lu, W., & Xue, F. (2020). A Review of BIM Data Exchange Method in BIM Collaboration. Proceedings of the 25th International Symposium on Advancement of Construction Management and Real Estate (CRIOCM2020).
- Lourenzi, F., & Geiger, A. (2022). BCFViewer – BIM Collaboration Format Tool Development. Paper presented at the ForumBauinformatik, 343–350.
- Masaki Muto. (2020). e-Submission Common Guidelines to Introduce BIM to the Building Process. Building Research Institute. bS Japan.
- Matos, B., Cruz, C., & Branc, F. (2024). Digitalization and PROCUREMENT in CONSTRUCTION PROJECTS: An Integrated BIM-Based Approach. Journal of Information Technology in Construction, 29, 400–423. 10.36680/j.itcon.2024.019
- Messner, J., Anumba, C., Dubler, C., Goodman, S., Kasprzak, C., Kreider, R., Leicht, R., Saluja, C., & Zikic, N. (2011). BIM Project Execution Planning Guide. PennState, College of Engineering.
- Mohanaraj, R., Ganeshu, P., & Mahendrarajah, G. (2020). Application of Cloud-Computing for the Internal Stakeholders' Collaborative Involvement in the Construction Industry: A Literature Review. Gari Winter Multidisciplinary Symposium.
- N, H. K., & Padala, S. P. S. (2024). A BIM-Integrated Multi Objective Optimization Model For Sustainable Building Construction Management. Construction Innovation. 10.1108/CI-09-2023-0223
- Nguyen, T. Q., & Nguyen, D. P. (2021). Barriers in BIM Adoption and the Legal Considerations in Vietnam. International Journal of Sustainable Construction Engineering and Technology, 12(1), 283–295. 10.30880/ijscet.2021.12.01.027
- Noardo, F., Ohori, K., Krijnen, T., & Stoter, J. (2021). An Inspection of IFC Models from Practice. Applied Sciences, 11, 2232. 10.3390/app11052232

- Oesterreich, T., & Teuteberg, F. (2016). Understanding The Implications of Digitisation and Automation in the Context of Industry 4.0: A Triangulation Approach and Elements of a Research Agenda for the Construction Industry. *Computers in Industry*, 8310.1016/j.compind.2016.09.006
- Okonta, E., Vukovic, V., & Hayat, E. (2024). Prospective Directions in the Computer Systems Industry Foundation Classes (IFC) for Shaping Data Exchange in the Sustainability and Resilience of Cities. *Electronics*, 13, 2297. 10.3390/electronics13122297
- Olugboyega, O., & Windapo, A. (2019). A Comprehensive BIM Implementation Model For Developing Countries. *Journal of Construction Project Management and Innovation*, 9, 83–104. 10.36615/jcpmi.v9i2.187
- Omotunde, H., & Ahmed, M. (2023). A Comprehensive Review of Security Measures in Database Systems: Assessing Authentication, Access Control, and Beyond. *Mesopotamian Journal of Cyber Security*, 2023, 115–133. 10.58496/MJCSC/2023/016
- Oppong, G., Chan, A., & Dansoh, A. (2017). A Review of Stakeholder Management Performance Attributes in Construction Projects. *International Journal of Project Management*, 35, 1037–1051. 10.1016/j.ijproman.2017.04.015
- Pan, Z., Yu, Y., Xiao, F., & Zhang, J. (2023). Recovering Building Information Model from 2D Drawings for Mechanical, Electrical and Plumbing Systems of Ageing Buildings. *Automation in Construction*, 152, 104914. 10.1016/j.autcon.2023.104914
- Park, J., & Cai, H. (2017). WBS-Based Dynamic Multi-Dimensional BIM database for Total Construction as-Built Documentation. *Automation in Construction*, 77, 15–23. 10.1016/j.autcon.2017.01.021
- Parusheva, S., & Aleksandrova, Y. (2021). Technologies, Tools, and Resources - Driving Forces in Construction Sector Digitalization. 2021 Tenth International Conference on Intelligent Computing and Information Systems (ICICIS). 10.1109/ICICIS52592.2021.9694177
- Pauwels, P., & Terkaj, W. (2016). EXPRESS to OWL for Construction Industry: Towards a Recommendable and Usable ifcOWL Ontology. *Automation in Construction*, 63, 100–133. 10.1016/j.autcon.2015.12.003
- Pérez-García, A., Martín-Dorta, N., & Aranda, J. Á. (2024). Enhancing BIM Implementation in Spanish Public Procurement: A Framework Approach. *Heliyon*, 10(9)10.1016/j.heliyon.2024.e30650
- Pongoh, D., Kantale, K., Kole, A., Wowor, C., & Wonda, G. (2024). The Influence of Quality Management Systems on the Success of Information Technology. *Journal Syntax Admiration*, 5, 3875–3881. 10.46799/jsa.v5i10.1632
- Preidel, C., Borrmann, A., Mattern, H., König, M., & Schapke, S. (2018). Common Data Environment. *Building Information Modeling - Technology Foundations and Industry Practice*. (pp. 279–291)10.1007/978-3-319-92862-3_15

- Ramaji, I. e. a. (2020). IFC-Based BIM-to-BEM Model Transformation. *Journal of Computing in Civil Engineering*, 3410.1061/(ASCE)CP.1943-5487.0000880
- Ramaji, I.,J., & Memari, A. (2017). Extending the current model view definition standards to support multi-storey modular building projects. *Architectural Engineering and Design Management*, , 1–19. 10.1080/17452007.2017.1386083
- Seiß, S. (2019). Development Of Model Checking Rules For Validation And Content Checking. *International Conference on Building Information Modelling (BIM) in Design, Construction and Operations*. 10.2495/BIM190211
- Serra, J. (2021). Los Datos: ¿Una Amenaza?: Gestión de Documentos en la Era de los Datos. *Tábula*, 43–63. 10.51598/tab.871
- Shin, S., Moon, H., & Shin, J. (2022). BIM-Based Maintenance Data Processing Mechanism through COBie Standard Development for Port Facility. *Applied Sciences*, 12(3)10.3390/app12031304
- Silva, H., Ribeirinho, L., & Henriques, S. (2024). IFC – Desafios de exportação e resolução com Python IfcOpenShell. 10.21814/uminho.ed.164.12
- Smith, P. (2014). BIM Implementation - Global Strategies. Paper presented at the University of Technology Sydney, 85 482–492. 10.1016/j.proeng.2014.10.575
- Sobhkhiz, Soroush and Zhou Yu-Cheng and Lin Jia-Rui and El-Diraby Tamer. (2021). Framing and Evaluating the Best Practices of IFC-Based Automated Rule Checking: A Case Study. *Buildings*, 11, 456. 10.3390/buildings11100456
- Sompolgrunk, A., Banihashemi, S., & Mohandes, S. R. (2021). Building Information Modelling (BIM) and the Return on Investment: a Systematic Analysis. *Construction Innovation*, ahead-of-print. 10.1108/CI-06-2021-0119
- Stepanov, A., Matveeva, M., & Peshkova, E. (2024). Digitalization of the Construction Industry: Prospects and Challenges. *Izvestiya Vuzov.Investitsii. Stroitelstvo. Nedvizhimost*, 14, 356–366. 10.21285/2227-2917-2024-2-356-366
- Sun, J., Paasch, J., Paulsson, J., Tarandi, V., & Harrie, L. (2023). A BIM-Based Approach to Design a Lifecycle 3D Property Formation Process A Swedish case study. *Land use Policy*, 13110.1016/j.landusepol.2023.106712
- Sweta, S. (2015). Database Management System. *Journal of Management Research and Analysis*, 359. 2394-2770
- Thomas Liebich, Modeling Support Group & buildingSMART International. (2009). IFC 2x Edition 3 Model Implementation Guide. buildingSmart.
- Toyin, J., & Mewomo, M. (2021). Barriers to Successful BIM Applications: A Literature Review. *Building Smart, Resilient And Sustainable Infrastructure In Developing Countries*.
- Treldal, N., Parsianfar, H., & Karlshøj, J. (2016). Using BCF as a Mediator for Task Management in Building Design. *International RILEM Conference on Materials, Systems and Structures in Civil Engineering*.

- Tsay, G. S., Staub-French, S., Poirier, E., Zadeh, P., & Pottinger, R. (2023). BIM for FM: Understanding Information Quality Issues in Terms of Compliance with Owner's Building Information Modeling Requirements. *Frontiers in Built Environment*, 9.
- Turk, Ž. (2020). Interoperability in Construction – Mission Impossible? Developments in the Built Environment, 4, 100018. 10.1016/j.dibe.2020.100018
- Utiome, E. (2015). Extending Building Information Models to Construction Specifications. Queensland University of Technology.
- Wahbeh, W. (2017). Building Skins, Parametric Design Tools and BIM Platforms. *Advanced Building Skins*.
- Wang, G. (2014). Building Information Modeling for Quality Management in Infrastructure Construction Projects. The 15th International Conference on Computing in Civil and Building. 10.1061/9780784413616.009
- Wang, Y. (2024). Impact of Management Information Systems on Enterprise Decision Support. *International Journal of Global Economics and Management*, 4, 54–61. 10.62051/ijgem.v4n3.07
- Warren, D. (2019). Parametric Modelling In Construction: Investigating The Quality Of Rule-Based Checking. WIT 2019 BIM International Confrence. 10.2495/BIM190061
- Wu, J. a. Z. J. (2019). New Automated BIM Object Classification Method to Support BIM Interoperability. *Journal of Computing in Civil Engineering*, 33, 04019033. 10.1061/(ASCE)CP.1943-5487.0000858
- Yang, B. e. a. (2021). IFC-Based 4D Construction Management Information Model of Prefabricated Buildings and Its Application in Graph Database. *Applied Sciences*, 11, 7270. 10.3390/app11167270
- Yu, Y., Kim, S., Jeon, H., & Koo, B. (2023). A Systematic Review of the Trends and Advances in IFC Schema Extensions for BIM Interoperability. *Applied Sciences*, 13, 12560. 10.3390/app132312560
- Zadeh, P., Staub-French, S., & Pottinger, R. (2015). Review Of Bim Quality Assessment Approaches For Facility Management. *International Construction Specialty Conference*. 10.13140/RG.2.1.2367.7282
- Zhang, Y., & Ye, Z. (2011). Data Maintenance and Control Strategy of Group Management. *Second International Conference on Innovations in Bio-inspired Computing and Applications*, 141–144. 10.1109/IBICA.2011.39
- Zhao, Q. e. a. (2020). A Graph-Based Method for IFC Data Merging. *Advances in Civil Engineering*, 2020, 1–15. 10.1155/2020/8782740

Anexos

A. Comparativa de aplicaciones IFC

Nombre de aplicación	Gratuito	Visualización	Federación	Detección (múltiples IFCs)	Gestión de interferencias	Gestión BCF (incidencias BIM)	Edición 3D	Edición de propiedades	Vista de clasificaciones	Organización por clasificaciones	Identificación por clasificaciones	Realizar mediciones	Realizar revisiones	Revisión de Psets	Revisión de revisiones	Comparar a partir de IDS	Visualización de versiones del IFC	Exportar/guardar IFC	Importar/guardar IFC	Trabajo con .ifc	Trabajo con .ifc.xml	Trabajo con .ifc.zip	IFC 2x3	IFC 4	IFC 4.3	Categoría	Tipo
Solibri Anywhere	✓	✓	-	-	-	-	-	-	✓	✓	✓	-	✓	-	-	-	-	✓	✓	✓	✓	✓	✓	#	Visor	App	
BIMVision	✓	✓	*	*	*	-	*	*	✓	✓	✓	✓	✓	*	*	-	-	✓	✓	-	-	#	#	-	Visor	App	
usBIM. Viewer+	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	-	#	#	#	Visor	App	
Dalux Viewer	✓	✓	✓	✓	✓	-	-	-	✓	-	-	✓	✓	-	-	-	-	✓	✓	-	-	#	#	-	Visor	web	
BeaVer	✓	✓	-	-	-	✓	-	-	✓	-	-	-	-	-	-	-	✓	✓	✓	-	-	#	#	-	Visor	App	
BIMCollab Zoom	✓	✓	✓	*	✓	-	-	-	✓	✓	✓	✓	✓	-	-	-	-	✓	✓	-	-	#	#	#	Visor	App	
DDS- CAD	✓	✓	-	-	-	-	-	-	✓	-	✓	✓	-	-	-	-	-	✓	✓	✓	✓	✓	#	#	Visor	App	
Trimble Connect Viewer	✓	✓	✓	-	✓	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	#	#	#	Visor	App
Navis Freedom	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	✓	✓	#	#	#	Visor	App	
Autodesk Viewer	✓	✓	✓	-	✓	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	-	-	#	#	-	Visor	web	
OpenIFC Viewer	✓	✓	✓	✓	-	-	-	-	✓	-	-	-	-	-	-	-	✓	✓	✓	-	-	#	#	-	Visor	App	
IFC WebGL	✓	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-	-	#	#	-	Visor	web	
IFC JAVA	✓	✓	-	-	-	-	-	-	✓	-	✓	-	-	-	-	-	✓	✓	✓	-	-	#	#	-	Visor	App	
BIM Collab	-	✓	✓	✓	✓	-	-	-	✓	✓	✓	✓	✓	-	✓	✓	-	✓	✓	-	-	#	#	#	Coordinación	App	
Solibri	-	✓	✓	✓	✓	-	-	-	✓	✓	✓	✓	✓	-	✓	-	-	✓	✓	-	-	✓	✓	#	Coordinación	App	
Navis Manage	-	✓	✓	✓	*	-	✓	-	✓	✓	✓	✓	✓	-	✓	-	✓	✓	✓	✓	✓	#	#	#	Coordinación	App	
Bexel	-	✓	✓	✓	✓	-	-	-	✓	✓	✓	✓	✓	-	-	-	✓	✓	✓	-	-	#	#	#	Coordinación	App/web	
Synchro	-	✓	✓	✓	✓	-	-	-	✓	✓	✓	✓	✓	-	-	-	✓	✓	✓	-	-	#	#	#	Coordinación	App	
usBIM	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	-	-	#	#	#	Coordinación	App/web	
Dalux	-	✓	✓	*	✓	-	-	-	✓	-	✓	✓	✓	-	✓	✓	-	✓	✓	-	-	#	#	#	Coordinación	App/web	
Autodesk Co. Cloud	-	✓	✓	✓	✓	-	-	-	✓	-	✓	✓	✓	-	✓	-	✓	✓	✓	✓	-	#	#	#	Coordinación	web	
Revit	-	✓	✓	✓	✓	-	-	-	✓	✓	✓	✓	✓	-	-	✓	✓	✓	✓	✓	✓	✓	✓	#	Autoría	App	
Revit	-	✓	✓	✓	✓	-	-	-	-	✓	✓	✓	✓	-	-	✓	✓	✓	✓	✓	✓	✓	✓	#	Autoría	App	
AECOsim	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-	-	✓	✓	#	Autoría	App	
Allplan	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-	-	✓	✓	#	Autoría	App	
Edificius	-	✓	-	-	-	✓	✓	-	-	-	-	✓	-	-	-	-	✓	✓	✓	-	-	✓	✓	#	Autoría	App	
Archicad	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-	-	✓	✓	#	Autoría	App	
FreeCAD	✓	✓	✓	-	-	✓	✓	-	-	-	-	-	✓	-	-	-	✓	✓	✓	-	-	✓	✓	#	Autoría	App	
Blender Bonsai	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	#	#	#	Autoría	App	
Vectorworks	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-	-	✓	✓	-	Autoría	App	
Tekla Structures	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	-	-	✓	✓	-	Autoría	App	
Rhinceros 3D	-	✓	-	-	-	✓	✓	-	-	-	-	-	-	-	-	-	✓	✓	✓	-	-	✓	✓	-	Autoría	App	
	App - Aplicación de escritorio web - Navegador web																										
	* No incluido en versión gratuita # Sí, pero no certificado por buildingSmart																										

Tabla A.1. Comparativa de aplicaciones IFC

B. Tipos de datos

Tipo de Dato	Definición	Ejemplo	Formato Común	Bases de Datos	IFC
Datos Caracteres (Strings)	Representan valores alfanuméricos utilizados para nombres o descripciones.	Nombre de una habitación: "Salón-Comedor"	Texto libre	VARCHAR, TEXT	IfcText
Datos Identificadores	Cadenas de caracteres utilizadas para asignar identificadores únicos.	ID de un elemento: "ELEM-00123"	Código alfanumérico	VARCHAR(50)	IfcIdentifier
Datos Etiquetas	Cadenas de caracteres para nombrar elementos de forma legible.	Nombre de un muro: "Muro con capa de yeso"	Texto corto	VARCHAR(100)	IfcLabel
Datos Numéricos Enteros	Representan números sin decimales, utilizados en conteos y valores discretos.	Número de niveles de un edificio: 15	Entero	INT, BIGINT	IfcInteger
Datos Numéricos Decimales	Representan valores con decimales, usados en mediciones y cálculos.	Porcentaje de pilares construidos: 12.75	Número decimal	FLOAT, DECIMAL	IfcReal
Datos Booleanos	Representan valores lógicos binarios, como "verdadero" (true) o "falso" (false).	Estado de construcción de una puerta: true/false	1/0 o true/false	BOOLEAN	IfcBoolean
Datos Lógicos	Similar a los booleanos, aplicados en condiciones dentro de especificaciones.	Material ignífugo: true/false	true/false	BOOLEAN	IfcLogical
Datos de Fecha y Hora	Representan una combinación de fecha y hora.	Momento en que se tomó una medición: "2025-03-17T14:30:00Z"	ISO 8601	DATETIME, TIMESTAMP	IfcDateTime
Datos de Fecha	Representan información relacionada con fechas específicas.	Fecha de inicio de construcción: "2025-06-01"	AAAA-MM-DD	DATE	IfcDate
Datos de Hora	Representan información sobre horas en un día.	Hora de finalización de una tarea: "16:45:00"	HH:MM:SS	TIME	IfcTime
Datos de Duración	Representan un período de tiempo o intervalo.	Duración total del proyecto: "180 días"	Días, horas	INTERVAL	IfcDuration
Datos Espaciales	Contienen información geográfica o de ubicación.	Coordenadas de un punto: "(40.7128° N, 74.0060° W)"	Latitud/Longitud	GEOMETRY, GEOGRAPHY	IfcCartesianPoint
Datos de Medidas de Longitud	Representan medidas de longitud en unidades específicas.	Altura de un muro: 3.5 metros	Número decimal con unidad	DECIMAL(10,2)	IfcLengthMeasure
Datos de Medidas de Área	Representan medidas de superficie.	Área de una habitación: 25.6 m²	Número decimal con unidad	DECIMAL(10,2)	IfcAreaMeasure
Datos de Medidas de Volumen	Representan medidas tridimensionales.	Volumen de una estructura: 120.5 m³	Número decimal con unidad	DECIMAL(10,2)	IfcVolumeMeasure
Datos de Medidas de Ángulo	Representan medidas angulares en grados o radianes.	Inclinación de un techo: 30°	Número decimal con unidad	DECIMAL(10,2)	IfcPlaneAngleMeasure
Datos Enumerados	Representan una selección predefinida de opciones dentro de un conjunto cerrado.	Material de pared: "Concreto, Madera, Acero"	Lista restringida	ENUM	IfcEnumeration
Datos Tabulares	Representan conjuntos de valores tabulados, como relaciones entre dos variables.	Tabla de variación de temperatura y resistencia del material.	Tabla con pares de valores	JSON, TABLE	IfcPropertyTableValue
Datos de Referencia	Representan enlaces a recursos externos o referencias dentro del modelo.	Enlace a una hoja técnica en PDF o una URL con especificaciones.	URL, identificador externo	VARCHAR, LINK	IfcPropertyReferenceValue

Tabla B.2. Tipos de datos

C. Comparativa de Sistemas de Bases de Datos

Característica	Arquitectura	Licencia	Tamaño de archivo	Portabilidad	Escalabilidad	Soporte de usuarios	Seguridad	Variedad de datos	Rendimiento	Soporte de ACID	Instalación	Costo
MySQL	Cliente-servidor	Código abierto (GPL)	~ 600 MB	Alta	Alta	Alto	Alta	Alta	Alto	Alto	Sencilla	Gratis (versión comunitaria)
MariaDB	Cliente-servidor	Código abierto (GPL)	~ 600 MB	Alta	Alta	Alto	Alta	Alta	Alto	Alto	Sencilla	Gratis (versión comunitaria)
PostgreSQL	Cliente-servidor	Código abierto (PostgreSQL)	~ 200 MB	Alta	Alta	Alto	Alta	Alta	Alto	Alto	Sencilla	Gratis (versión comunitaria)
Oracle Database	Cliente-servidor	Comercial (propietario)	~ 5 GB	Alta	Muy alta	Alto	Muy alta	Alta	Alto	Alto	Compleja	Alto (licencia propietaria)
Microsoft SQL Server	Cliente-servidor	Comercial (propietario)	~ 2 GB	Alta	Alta	Alto	Alta	Alta	Alto	Alto	Sencilla	Alto (licencia propietaria)
SQLite	Base de datos integrada	Código abierto (dominio público)	~ 250 KB	Baja	Baja	Bajo	Baja	Media	Bajo	Bajo	Muy sencilla	Gratis (sin costo de licencia)
MongoDB	Cliente-servidor	Código abierto (bases de datos)	~ 600 MB	Alta	Alta	Alto	Alta	Alta	Alto	Alto	Sencilla	Gratis (para bases SQL)
IBM Db2	Cliente-servidor	Comercial (propietario)	~ 3 GB	Alta	Alta	Alto	Alta	Alta	Alto	Alto	Sencilla	Alto (licencia propietaria)
Cassandra	Cliente-servidor	Código abierto (Apache)	~ 2 GB	Alta	Muy alta	Alto	Alta	Alta	Alto	No	Compleja	Gratis (versión comunitaria)
Redis	Cliente-servidor	Código abierto (MIT)	~ 30 MB	Alta	Alta	Alto	Alta	Alta	Alto	No	Sencilla	Gratis (versión comunitaria)
CouchDB	Cliente-servidor	Código abierto (Apache)	~ 500 MB	Alta	Alta	Alto	Alta	Alta	Alto	No	Sencilla	Gratis (versión comunitaria)
ElasticSearch	Cliente-servidor	Código abierto (Apache)	~ 200 MB	Alta	Alta	Alto	Alta	Alta	Alto	Sí	Sencilla	Gratis (versión comunitaria)
Firebase Realtime DDBB	Cliente-servidor	Comercial (propietario)	~ 50 MB	Alta	Alta	Alto	Alta	Alta	Alto	Sí	Sencilla	Pago (en función del uso)

Tabla C.3. Comparativa de Sistemas de Bases de Datos

D. Comparativa de Cantidades IFC

	VER S	Width	NominalWidth	Height	NominalHeight	FinishCeilingHeight	FinishFloorHeight	Depth	Perimeter	GrossPerimeter	NetPerimeter	Length	NominalLength	GrossFootPrintArea	NetFootPrintArea	CrossSectionArea	GrossSideArea	NetSideArea	GrossSideAreaLeft	NetSideAreaLeft	GrossSideAreaRight	NetSideAreaRight	GrossCrossSectionArea	NetCrossSectionArea	Area	GrossArea	NetArea	ProjectedArea	TotalSurfaceArea	OuterSurfaceArea	GrossSurfaceArea	NetSurfaceArea	FootPrintArea	GrossFloorArea	NetFloorArea	GrossWallArea	NetWallArea	GrossCeilingArea	NetCeilingArea	Volume	GrossVolume	NetVolume	Weight	GrossWeight	NetWeight	Count	NumberOfCircuits	NumberOfGangs	
		L	L	L	L	L	L	L	L	L	L	L	L	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	A	V	V	V	W	W	W	C	C	C					
IfcBeam	2x3												X			X													X	X										X	X		X	X				BaseQuantities	
	4										X				X															X	X	X							X	X		X	X				Qto_BeamBase Quantities		
	4.3										X				X															X	X	X							X	X		X	X						
IfcColumn	2x3												X			X													X	X										X	X		X	X				BaseQuantities	
	4										X				X															X	X	X							X	X		X	X				Qto_ColumnBase Quantities		
	4.3										X				X															X	X	X							X	X		X	X						
IfcDoor	2x3																																															BaseQuantities	
	4	X		X				X																X																							Qto_DoorBase Quantities		
	4.3	X		X				X															X																										
IfcWindow	2x3																																															BaseQuantities	
	4	X		X				X																X																							Qto_WindowBase Quantities		
	4.3	X		X				X															X																										
IfcWall	2x3		X		X								X	X	X		X	X	X	X	X	X																		X	X							BaseQuantities	
	4	X		X							X		X	X		X	X																						X	X		X	X				Qto_WallBase Quantities		
	4.3	X		X							X		X	X		X	X																						X	X		X	X						
IfcSlab	2x3		X					X																	X	X														X	X		X	X				BaseQuantities	
	4	X					X	X			X														X	X													X	X		X	X				Qto_SlabBase Quantities		
	4.3	X					X	X			X														X	X													X	X		X	X						
IfcRamp Flight	2x3																																															BaseQuantities	
	4	X									X														X	X													X	X							Qto_RampFlightBase Quantities		
	4.3	X									X														X	X													X	X									
IfcRoof	2x3																											X																				BaseQuantities	
	4																								X	X	X																				Qto_RoofBase Quantities		
	4.3																								X	X	X																						
IfcChimney	2x3																																															BaseQuantities	
	4										X																																				Qto_ChimneyBase Quantities		
	4.3										X																																						
IfcCovering	2x3																																															BaseQuantities	
	4	X																							X	X																					Qto_CoveringBase Quantities		
	4.3	X																							X	X																							

Gestión y control de los datos BIM a lo largo del Ciclo de Vida en Proyectos de Edificación
Samuel J. Triana González

Gestión y control de los datos BIM a lo largo del Ciclo de Vida en Proyectos de Edificación
Samuel J. Triana González

IfcQuantityLength (L)	IfcQuantityArea (A)	IfcQuantityVolume (V)	IfcQuantityWeight (W)	IfcQuantityCount ©
*Excluidos elementos solo con peso, elementos y equipos auxiliares, elementos de infraestructuras, elementos de referencia y emplazamiento				

Gestión y control de los datos BIM a lo largo del Ciclo de Vida en Proyectos de Edificación

E. Revisiones en entornos BIM

Código	Revisión	Tipo	Descripción
A001	Estructura de carpetas	CDE	Define la organización de carpetas para garantizar un almacenamiento estructurado de la información.
A002	Nomenclatura de carpetas	CDE	Establece las reglas para nombrar carpetas de manera uniforme y comprensible.
A003	Permisos de usuarios	CDE	Regula los accesos y permisos de usuarios para proteger la información.

Tabla E.5. Revisiones del Entorno Común de datos

Código	Revisión	Tipo	Descripción
B001	Entrega de archivos según MIDP	DOCS	Define los procedimientos de entrega de archivos en cumplimiento con el plan de ejecución BIM.
B002	Nomenclatura de archivos	DOCS	Establece un sistema estandarizado de nombres para facilitar la identificación de archivos.
B003	Ubicación de archivos	DOCS	Determina dónde se deben almacenar los archivos dentro de la estructura definida.
B004	Formato de archivos	DOCS	Especifica los formatos de archivo aceptados para garantizar compatibilidad y estandarización.
B005	Idioma de archivos	DOCS	Define el idioma oficial a usar en los archivos para evitar confusiones.
B006	Histórico de versiones	DOCS	Registra los cambios y versiones de archivos para mantener trazabilidad.
B007	Metadatos de archivos	DOCS	Define los metadatos obligatorios para categorizar y organizar los archivos.
B008	Registro de actas de reunión	DOCS	Establece el procedimiento para documentar reuniones y acuerdos.
B009	Registros de archivos	DOCS	Controla y documenta el estado y ubicación de archivos clave. Documentos, audiovisuales, modelos, extracciones de modelo y recursos asociados
B010	Registro de informes de transmisión	DOCS	Documenta los envíos y recepciones de archivos para control de versiones.

Código	Revisión	Tipo	Descripción
B011	Registro de revisiones	DOCS	Mantiene un historial de revisiones realizadas a los documentos.
B012	Registro de correspondencia	DOCS	Documenta la comunicación oficial entre las partes del proyecto.
B013	Registro de SDI	DOCS	Controla el intercambio de información dentro del Sistema de Datos de la Infraestructura.

Tabla E.6. Revisiones de documentación BIM

Código	Revisión	Tipo	Descripción
C001	Estructura según requisitos	BEP	Define la organización del modelo según los requisitos del proyecto.
C002	MIDP	BEP	Plan de entrega de información del modelo (Model Information Delivery Plan).
C003	Matriz de atributos	BEP	Define los atributos que deben contener los elementos del modelo.
C004	Matriz de interferencias	BEP	Identifica y clasifica interferencias dentro del modelo BIM.
C005	Procesos BIM	BEP	Establece los procesos a seguir en la implementación de metodología BIM.
C006	Plantillas BIM	BEP	Proporciona formatos y plantillas estandarizadas para el trabajo con modelos BIM.

Tabla E.7. Revisiones del Plan de Ejecución BIM

Código	Revisión	Tipo	Descripción
D001	Metadatos de incidencias	INCIDENCIAS	Define los metadatos a incluir en los reportes de incidencias. BCF: Asignación, Tipo (Diseño, Errores de modelado, información, observación), prioridad
D002	Visual. Geometrías de referencia	INCIDENCIAS	Registra y controla problemas en geometrías de referencia dentro del modelo. Niveles y rejillas sí. Marcadores, bocetos, líneas auxiliares no.
D003	Visual. Geometrías de objetos	INCIDENCIAS	Cumple con LOIN. Identifica inconsistencias en la modelación de objetos, a nivel de detalle, aspecto, etc.

Código	Revisión	Tipo	Descripción
D004	Visual. Cortes con huecos y vacíos	INCIDENCIAS	Detecta errores en cortes y vacíos de los elementos modelados.
D005	Visual. Geometrías de espacios	INCIDENCIAS	Verifica la correcta representación de volumetrías vacías en el modelo. Huecos, espacios, zonas
D006	Visual. Geometría asociada a niveles	INCIDENCIAS	Controla la correcta vinculación de geometrías con niveles.
D007	Visual. Geometría de plano de corte 2D	INCIDENCIAS	Asegura la correcta representación en vistas 2D. Rango de vista, profundidad
D008	Visual. Comparación de versiones de modelo	INCIDENCIAS	Permite comparar versiones del modelo para detectar cambios.
D009	Visual. Clasificación por tipos	INCIDENCIAS	Organiza las incidencias según categorías definidas.
D010	Interferencias. Tests generados	INCIDENCIAS	Pruebas para detectar interferencias en el modelo. Número de test según matriz, conjuntos de selección A y B
D011	Interferencias. Duplicados	INCIDENCIAS	Identifica elementos duplicados en el modelo.
D012	Interferencias. Ocupación de espacios	INCIDENCIAS	Controla la correcta ocupación de espacios en el diseño.
D013	Interferencias. Detecciones dinámicas	INCIDENCIAS	Detecta problemas generados por movimientos dentro del modelo.
D014	Interferencias. Tolerancias y omisiones	INCIDENCIAS	Controla las limitaciones y configuraciones de los tests.
D015	Interferencias. Clasificación por tipos	INCIDENCIAS	Organiza las interferencias en categorías para su gestión.

Tabla E.8. Revisiones de incidencias

Código	Revisión	Tipo	Descripción
E001	Nativos. Datos generales	CALIDAD	Define los parámetros de calidad generales para archivos nativos. Software, autor, peso, versión, idioma, nomenclatura y formato

Código	Revisión	Tipo	Descripción
E002	Nativos. Segregación	CALIDAD	Establece criterios para la segmentación de archivos nativos según su función.
E003	Nativos. Georreferenciación	CALIDAD	Verifica la correcta ubicación geoespacial de los archivos nativos.
E004	IFC. Datos generales	CALIDAD	Especifica la calidad para archivos en formato IFC. Software, autor, versión IFC, MVD, peso, versión, idioma, nomenclatura, unidades y formato
E005	IFC. Segregación	CALIDAD	Define la organización y división de elementos en archivos IFC. Siguiendo requisitos de estados, disciplinas o peso.
E006	IFC. Estructura general	CALIDAD	Establece las reglas de estructuración de modelos en formato IFC. Nombre de proyecto, emplazamiento, edificio
E007	IFC. Elementos de referencia	CALIDAD	Controla la correcta asignación de elementos de referencia en IFC. Niveles y rejillas
E008	IFC. Atributos del edificio	CALIDAD	Define los atributos obligatorios de los modelos IFC.
E009	IFC. Georreferenciación	CALIDAD	Verifica la correcta ubicación geoespacial en archivos IFC.
E010	IFC. Nomenclatura de instancias	CALIDAD	Establece las reglas de nombrado de instancias en archivos IFC. Incluyendo elementos de referencia, objetos, grupos y espacios
E011	IFC. Atributos de espacios	CALIDAD	Define los datos y características necesarias para la definición de espacios.
E012	IFC. Atributos de objetos	CALIDAD	Especifica los atributos que deben tener los objetos en IFC.
E013	IFC. Grupos y montajes de objetos	CALIDAD	Regula la agrupación y ensamblaje de objetos en modelos IFC.
E014	IFC. Sistemas y subsistemas de objetos	CALIDAD	Define la estructura de sistemas y subsistemas en IFC.
E015	IFC. Asociación de objetos a niveles	CALIDAD	Controla la correcta asignación de objetos a niveles dentro del modelo.
E016	IFC. Organización por clases	CALIDAD	Establece criterios para la clasificación de objetos en IFC.
E017	IFC. Tipos predefinidos y personalizados	CALIDAD	Regula el uso de tipos estándar y personalizados en IFC.

Código	Revisión	Tipo	Descripción
E018	IFC. Partes y subcomponentes	CALIDAD	Define los materiales y propiedades físicas de los objetos en IFC.
E019	IFC. Objetos con materiales y capas	CALIDAD	Especifica la información relacionada con materiales: divisiones, aspecto, espesores...
E020	IFC. Contenido y nombrado de conjuntos	CALIDAD	Establece revisiones para: Atributos dentro de conjuntos, conjuntos dentro clases
E021	IFC. Contenido y nombrado de atributos	CALIDAD	Define los estándares para la asignación de atributos y su correcta denominación en modelos IFC. Valores de atributos, nulos, tipos de datos, tolerancias, unidades
E022	IFC. Sistemas de clasificación	CALIDAD	Criterios para la clasificación de elementos en el modelo según normativas y estándares. De objetos, de grupos o sistemas, de espacios y de proyecto
E023	IFC. Identificadores únicos	CALIDAD	Regula la asignación de identificadores únicos a los elementos para garantizar su trazabilidad. Definidos y no duplicados
E024	IFC. Recursos vinculados	CALIDAD	Define los procedimientos para la vinculación de recursos y referencias externas en modelos IFC. Enlaces, documentos, planos etc.
E025	Federado. Datos generales	CALIDAD	Especifica los requisitos y parámetros generales de los modelos federados. Versión, idioma, composición del federado con IFC, nomenclatura, formato, caducidad
E026	Federado. Información gráfica	CALIDAD	Regula los aspectos gráficos del modelo federado para asegurar consistencia visual.
E027	Federado. Información no gráfica	CALIDAD	Define los datos no gráficos esenciales que deben incorporarse en los modelos federados.
E028	Federado. Información vinculada	CALIDAD	Establece los criterios para la integración y conexión de información entre diferentes modelos.

Tabla E.9. Revisiones de calidad de los modelos BIM

Código	Revisión	Tipo	Descripción
F001	Georreferenciación	REF. EXTERNAS	Verifica la correcta ubicación geoespacial del modelo en relación con su entorno real. CAD, taquimétricos, nubes de puntos
F002	Estructura y contenido	REF. EXTERNAS	Define la organización de los archivos de referencia externa para garantizar coherencia y accesibilidad.
F003	Coherencia con modelos	REF. EXTERNAS	Asegura la correcta integración de referencias externas con los modelos principales.

Tabla E.10. Revisiones de referencias externas

Código	Revisión	Tipo	Descripción
G001	Planos BIM con modelos	PLANOS	Especifica los requisitos para la generación de planos a partir de modelos BIM. Coherencia
G002	Planos no BIM con modelos	PLANOS	Define las directrices para la elaboración de planos sin dependencia directa de modelos BIM. Coherencia
G003	Información de planos	PLANOS	Determina los datos esenciales que deben incluirse en los planos para su correcta interpretación. Identificación del modelo al que pertenecen, nomenclatura, fecha, autor, leyendas y anotaciones

Tabla E.11. Revisiones de planos obtenidos de modelos

Código	Revisión	Tipo	Descripción
H001	Estructura y contenido	PRESUPUESTOS	Establece la estructura y organización de la información en los presupuestos generados a partir de modelos. Capítulos, unidades, duplicados...
H002	Mediciones asociadas a modelos	PRESUPUESTOS	Define los criterios para la extracción y cálculo de mediciones desde los modelos. Diferencia entre unidades desde modelos y las que no
H003	Codificación de todos los objetos	PRESUPUESTOS	Regula el uso de códigos estándar para los elementos incluidos en los presupuestos.

Tabla E.12. Revisiones de presupuestos asociados a modelos

I001	Planificación asociada a modelos	PLANIFICACIÓN	Especifica cómo deben integrarse los modelos con los planes de ejecución del proyecto.
I002	Fechas planificadas de inicio y fin	PLANIFICACIÓN	Define las fechas clave para la ejecución del proyecto según la planificación.
I003	Fechas reales de inicio y fin	PLANIFICACIÓN	Registra las fechas efectivas de ejecución comparadas con la planificación.
I004	Codificación de todos los objetos	PLANIFICACIÓN	Asegura que los objetos planificados tengan códigos únicos para su seguimiento.
I005	Organización por tipos de tareas	PLANIFICACIÓN	Clasifica las tareas según categorías para una mejor gestión y control.

Tabla E.13. Revisiones de planificaciones asociadas a modelos

Código	Revisión	Tipo	Descripción
J001	Certificaciones desde modelos	CERTIFICACIONES	Define el procedimiento para la certificación de elementos y procesos a partir de los modelos.
J002	Codificación de todos los objetos	CERTIFICACIONES	Establece códigos estándar para los objetos dentro del proceso de certificación. Fecha de certificación, estado, identificación del código de certificación en cada objeto...
J003	Relación con unidades de obra	CERTIFICACIONES	Vincula los elementos modelados con las unidades de obra certificadas.

Tabla E.14. Revisiones de certificaciones de obra

Código	Revisión	Tipo	Descripción
K001	Modelos según realidad construida	ASBUILT	Regula la actualización de modelos para reflejar con precisión el estado final de la construcción.
K002	Recursos externos vinculados a modelos	ASBUILT	Define cómo deben integrarse recursos externos en los modelos finales del proyecto.
K003	Coherencia entre planos y modelos	ASBUILT	Verifica la consistencia entre los modelos y los planos generados a partir de ellos.
K004	Definición de COBie	ASBUILT	Especifica los requisitos para la generación y gestión de información COBie en los modelos.

Tabla E.15. Revisiones de modelos As-Built

F. Plantilla cuestionario BIM

Encuesta Gestión Datos BIM - PhD – UPM

Introducción:

El presente cuestionario forma parte del trabajo de investigación "**Gestión y control de los datos BIM a lo largo del ciclo de vida en proyectos de edificación**" desarrollado por [D. Samuel J. Triana González](#) y que está dirigiendo el [Doctor Rafael Lozano Díez](#) en la **Universidad Politécnica de Madrid (UPM)**. Este trabajo aborda el uso de los datos BIM a través formatos abiertos y su interacción con otras metodologías y entornos digitales.

Objetivo:

El objetivo de la encuesta es **conocer el nivel de madurez en la gestión de procesos y datos BIM** desde la perspectiva de los expertos BIM del sector AECO.

Tiempo estimado:

Esta encuesta tiene una **duración aproximada de 5 minutos**.
Muchas gracias por su apoyo y colaboración.

**La información recogida en este cuestionario será exclusivamente utilizada para fines académicos y de investigación. Asimismo, es un modo de colaborar con el resto del sector para definir el grado actual de integración de la metodología y de poder determinar posibles mejoras en procesos y métodos de aplicación.*

Las respuestas y la identidad de los participantes serán confidenciales. Al responder a este cuestionario, el participante consiente el tratamiento de los datos por parte del investigador y, en cualquier caso, se le recuerda que se encuentra bajo el amparo de la **Ley Orgánica de Protección de Datos Personales y Garantía de los Derechos Digitales.*

Perfil y experiencia BIM

1. **Nombre y apellidos:** *(Opcional)*

2. **Empresa / Entidad / Organización en la que actualmente trabaja:** *(Opcional)*

3. **Tipología de empresa:** *(Obligatoria)*

☐ Estudio de arquitectura	☐ Ingeniería
☐ Constructora	☐ Promotora
☐ Centro de enseñanza	☐ Consultora
☐ Otros	
4. **¿Con qué tipo de proyectos suele trabajar o trabaja?** *(Obligatoria)*
Multirrespuesta

☐ Infraestructuras y obras civiles	☐ Edificación
☐ Otros	
5. **¿Cuántos años de experiencia tiene en el uso y manejo de la metodología BIM?** *(Obligatoria)*

☐ Menos de 3 años	☐ Entre 3 y 5 años
☐ Entre 5 y 10 años	☐ Más de 10 años
6. **¿Qué puesto desarrolla en su empresa actualmente?** *(Obligatoria)*

☐ CEO	☐ Director
☐ Project Manager	☐ Ingeniero
☐ Arquitecto	☐ Responsable BIM
☐ Coordinador BIM	☐ Técnico BIM
☐ Investigador / Docente	

[A] Planificación inicial BIM

7. ¿Es usual que en los proyectos en los que trabaja se exija desde las primeras fases de proyecto el cumplimiento del EIR, AIR u otros requisitos de información de construcción? *(Obligatoria)*
- ☐ No
- ☐ Solo en algunos proyectos específicos
- ☐ En la mayoría de proyectos
- ☐ En todos los proyectos que realiza la empresa
8. ¿Se establecen desde fases tempranas los Niveles de información (LOD/ LOIN) de las entidades BIM? *(Obligatoria)*
- ☐ No
- ☐ Solo de las entidades principales
- ☐ Sí, pero solo para la fase de diseño
- ☐ Sí, para la fase de diseño y construcción
- ☐ Sí, para todas las fases de proyecto
9. De los siguientes puntos, ¿Cuál es el nivel importancia que considera para una buena planificación y gestión de la información BIM? *El 1 corresponde a “No importante” mientras que el 5 que se considera “Imprescindible”. Responda una opción por fila (Obligatoria)*

	1	2	3	4	5
Estandarización de la información	☐	☐	☐	☐	☐
Trazabilidad de los entregables	☐	☐	☐	☐	☐
Interoperabilidad entre archivos	☐	☐	☐	☐	☐
Centralización de la información	☐	☐	☐	☐	☐
Automatización de procesos	☐	☐	☐	☐	☐
Implicación de los stakeholders	☐	☐	☐	☐	☐

[B] Trabajo con atributos y propiedades

10. *¿Trabaja con plantillas tipo "Parámetros compartidos" o similares? (Obligatoria)*

- ☐ No
- ☐ Sí, pero solo para propiedades relativas a la fase de diseño
- ☐ Sí, para propiedades relativas a la fase de diseño y construcción
- ☐ Sí, para propiedades relativas a todo el ciclo de vida

11. *¿Trabaja con bibliotecas de objetos BIM personalizadas y adaptadas a los proyectos que realiza su empresa? (Obligatoria)*

- ☐ No
- ☐ Sí, entidades por defecto o descargadas de internet
- ☐ Sí, entidades personalizadas por el equipo interno, pero con un nivel genérico de información
- ☐ Sí, entidades correctamente definidas con la información relevante definida

12. *Por norma general, ¿en el BEP se establecen de forma correcta las propiedades, elementos, unidades, tipos de datos, etc. que requiere el proyecto? (Obligatoria)*

- ☐ No
- ☐ De forma muy genérica
- ☐ Sí, existe una estructura clara de información necesaria para el proyecto

[C] Trabajo con formatos abiertos

13. *En los proyectos que trabaja actualmente, ¿Se exige que los entregables se generen con formatos abiertos tipo IFC? (Obligatoria)*

- ☐ No
☐ En muy pocos proyectos
☐ En la mayoría de proyectos
☐ En todos los proyectos

14. *En cuanto al trabajo con modelos IFC, ¿Se respetan los conjuntos de propiedades, nombres de propiedades y resto de información del estándar? (Obligatoria)*

- ☐ No se trabaja con este tipo de archivo
☐ Solo se especifica el MVD (Model View Definition) y se realizan algunas configuraciones generales del archivo
☐ Se personaliza la creación del archivo en cada proyecto
☐ Existen plantillas y archivos de apoyo en los que se encuentran estandarizados los datos y procedimientos de exportación y creación de los archivos según las especialidades de construcción, etapas de proyecto, etc.

15. *De los siguientes estándares y recursos BIM, indique el nivel de conocimiento que posee. El 1 equivale a “Ningún” conocimiento hasta el 4 que corresponde a “Avanzado”. Responda una opción por fila (Obligatoria)*

	1	2	3	4
IFC (Industry Foundation Classes)	☐	☐	☐	☐
IDM (Information Delivery Manual)	☐	☐	☐	☐
MVD (Model View Definitions)	☐	☐	☐	☐
bSDD (BuildingSmart Dictionary)	☐	☐	☐	☐
BCF (BIM Collaboration Format)	☐	☐	☐	☐
COBIE (Const. Inform. Exchange)	☐	☐	☐	☐
IDS (Inform. Delivery Specification)	☐	☐	☐	☐

[D] Controles de calidad

16. *¿Tiene establecido un sistema de control de calidad de los entregables BIM que se generan? (Obligatoria)*
- ☐ No hay un proceso normalizado de Control de calidad en la empresa
 - ☐ Sí, cada departamento o equipo aplica sus propios criterios
 - ☐ Sí, existe un control de Calidad definido en los distintos Protocolos y Planes de Ejecución BIM de la empresa
17. *¿Se realizan controles de calidad relativos al Nivel de información (LOD / LOIN) de las entidades BIM? (Obligatoria)*
- ☐ No
 - ☐ Sí, son realizados manualmente a partir de plantillas de revisión y control
 - ☐ Sí, se realizan a partir de rutinas y procesos de control automático o semiautomático
18. *¿Se realizan controles de calidad relativos a la nomenclatura de archivos y elementos? (Obligatoria)*
- ☐ No
 - ☐ Sí, son realizados manualmente a partir de plantillas de revisión y control
 - ☐ Sí, se realizan a partir de rutinas y procesos de control automático o semiautomático

[E] Futuro del BIM

19. *¿Considera que la metodología BIM tiende a integrarse y colaborar cada vez más con otras tecnologías / herramientas emergentes? (Obligatoria)*

1 2 3 4 5

Muy en desacuerdo ☐ ☐ ☐ ☐ ☐ Muy de acuerdo

20. *¿Considera útil la interacción entre bases de datos BIM y otras bases de datos externas? (Obligatoria)*

1 2 3 4 5

Muy en desacuerdo ☐ ☐ ☐ ☐ ☐ Muy de acuerdo

21. *¿Considera importante el uso de formatos abiertos como el IFC para una mejor gestión en los proyectos del sector AECO? (Obligatoria)*

1 2 3 4 5

Muy en desacuerdo ☐ ☐ ☐ ☐ ☐ Muy de acuerdo

22. *¿Considera importante la trazabilidad de los datos a lo largo del ciclo de vida de un activo de construcción? (Obligatoria)*

1 2 3 4 5

Muy en desacuerdo ☐ ☐ ☐ ☐ ☐ Muy de acuerdo

23. *¿Considera provechoso el uso de una base centralizada que pueda integrar los datos BIM de los activos construidos con otros datos relacionados con proveedores, contratación, mantenimiento, etc.? (Obligatoria)*

1 2 3 4 5

Muy en desacuerdo ☐ ☐ ☐ ☐ ☐ Muy de acuerdo

24. *¿Algún comentario o sugerencia? (Opcional)*
