

# RVDynDB: Repositorio abierto para el intercambio de datos dinámicos de vehículos ferroviarios

*B. Suarez, J. Felez, J.D. Sanz y V. Fernandez*

En este artículo se describe el proceso de diseño e implementación de la base de datos RVDynDB (Rail Vehicle Dynamic parameters DataBase), que pretende ser un extenso repositorio de los modelos de dominio público empleados en la simulación dinámica de vehículos ferroviarios en todo el mundo. Atendiendo a sus características de flexibilidad, extensibilidad e independencia de la plataforma, se ha escogido un modelo de datos XML, que facilita el almacenamiento de datos de procedencia muy heterogénea, al tiempo que permite compartir el contenido de la base de datos con otros usuarios a través de internet. Se ha presentado también el lenguaje RVDynML (Rail Vehicle Dynamic parameters Markup Language), que define la estructura de la información almacenada en la base de datos. Al ser un lenguaje basado en XML, con el tiempo podría llegar a convertirse en un estándar para el intercambio de datos sobre los principales parámetros constructivos que definen el comportamiento dinámico de los vehículos. Se han seleccionado 173 referencias bibliográficas, cuyos datos se han utilizado para construir la base de datos, constituida por un total de 957 registros. Finalmente, se ha desarrollado una aplicación específica con MATLAB para gestionar las búsquedas en la base de datos. Para ello se ha empleado una API de Java que proporciona una interfaz para el DOM, que permite acceder, modificar, insertar o eliminar los elementos y atributos que componen un documento XML.

*Esta es una versión electrónica de un artículo publicado en las Actas del IV Congreso Nacional de Innovación Ferroviaria, Sevilla, España, 3-5 Diciembre, 2008.*

***Para citar este artículo, hacer referencia al artículo publicado:***

***B. Suarez, J. Felez, J.D. Sanz y V. Fernandez, RVDynDB: Repositorio abierto para el intercambio de datos dinámicos de vehículos ferroviarios. Actas del IV Congreso Nacional de Innovación Ferroviaria, Sevilla, España, 3-5 Diciembre, 2008.***

## **RVDynDB: Repositorio abierto para el intercambio de datos dinámicos de vehículos ferroviarios**

*Berta Suárez, Jesús Félez, Juan de Dios Sanz, Vicente Fernández  
CITEF, Universidad Politécnica de Madrid*

### **RESUMEN**

En este artículo se describe el proceso de diseño e implementación de la base de datos RVDynDB (Rail Vehicle Dynamic parameters DataBase), que pretende ser un extenso repositorio de los modelos de dominio público empleados en la simulación dinámica de vehículos ferroviarios en todo el mundo.

Atendiendo a sus características de flexibilidad, extensibilidad e independencia de la plataforma, se ha escogido un modelo de datos XML, que facilita el almacenamiento de datos de procedencia muy heterogénea, al tiempo que permite compartir el contenido de la base de datos con otros usuarios a través de internet. Se ha presentado también el lenguaje RVDynML (Rail Vehicle Dynamic parameters Markup Language), que define la estructura de la información almacenada en la base de datos. Al ser un lenguaje basado en XML, con el tiempo podría llegar a convertirse en un estándar para el intercambio de datos sobre los principales parámetros constructivos que definen el comportamiento dinámico de los vehículos.

Se han seleccionado 173 referencias bibliográficas, cuyos datos se han utilizado para construir la base de datos, constituida por un total de 957 registros. Finalmente, se ha desarrollado una aplicación específica con MATLAB para gestionar las búsquedas en la base de datos. Para ello se ha empleado una API de Java que proporciona una interfaz para el DOM, que permite acceder, modificar, insertar o eliminar los elementos y atributos que componen un documento XML.

### **1. MOTIVACIÓN Y OBJETIVOS**

Para poder llevar a cabo estudios de sensibilidad que permitan determinar la influencia de los principales parámetros constructivos de un vehículo ferroviario en el comportamiento dinámico global del mismo, es recomendable conocer con anterioridad los rangos de variación de estos parámetros. De este modo es posible trabajar con valores realistas, propios de cualquier tipo de vehículo ferroviario.

Con esta intención, se ha considerado conveniente llevar a cabo un tratamiento estadístico sobre este tipo de datos. Para ello se ha recurrido a la utilización de información de dominio público, utilizando los datos publicados por diferentes autores en relación a los valores utilizados para construir modelos matemáticos para la simulación dinámica de vehículos ferroviarios. Tras consultar numerosas fuentes bibliográficas

especializadas, se han localizado más de 150 referencias bibliográficas, en las que se recoge este tipo de información.

Dada la gran cantidad de material disponible, para facilitar su posterior manipulación, se ha construido una base de datos, a la que se ha denominado RVDynDB (Rail Vehicle Dynamic parameters DataBase). El primer paso para la creación de la base de datos es llevar a cabo un diseño conceptual de la misma para, a continuación, materializarlo con alguna herramienta específica, que utilice un modelo de datos y un lenguaje propios. Durante este proceso se han seguido los siguientes pasos:

- Recopilación de datos
  - Búsqueda bibliográfica
  - Selección de documentos
- Diseño de la base de datos
  - Definición del esquema de datos conceptual
  - Elección del modelo de datos → XML
  - Definición del esquema de la base de datos (XML Schema)
- Implementación de la base de datos
  - Elección del software de gestión de la base de datos (XMLSpy, Matlab)
  - Entrada de datos
- Definición de consultas a la base de datos
  - Recuperación de parámetros directos
  - Estimación de parámetros indirectos
- Intercambio de datos con otros usuarios
  - Publicación de la base de datos en internet
  - Intercambio de datos con otros usuarios

En los siguientes apartados se describirá cada etapa con mayor grado de detalle.

## **2. RECOPIACIÓN DE DATOS**

El proceso de recopilación de datos se ha iniciado con una exhaustiva búsqueda bibliográfica. Entre los documentos consultados, se han escogido para construir la base de datos aquellos que incluyen valores numéricos para caracterizar los parámetros geométricos y mecánicos (masas, inercias, rigideces, amortiguamientos,...) de algún vehículo ferroviario en particular, habiéndose seleccionado finalmente un total de 173 documentos.

## **3. DEFINICIÓN DEL ESQUEMA CONCEPTUAL**

El primer paso para diseñar la base de datos es construir el esquema conceptual, que es una descripción abstracta de la parte del mundo real que el contenido de la base de datos pretende representar. La definición del esquema conceptual comienza con la selección de los datos que se desea almacenar, asegurándose de que se recogen todos aquellos datos necesarios para satisfacer el propósito por el que se crea la base de datos. Para construir la base de datos RVDynDB, se ha procedido a fragmentar la información en grandes bloques de datos, habiéndose identificado los grupos representados en la siguiente figura.

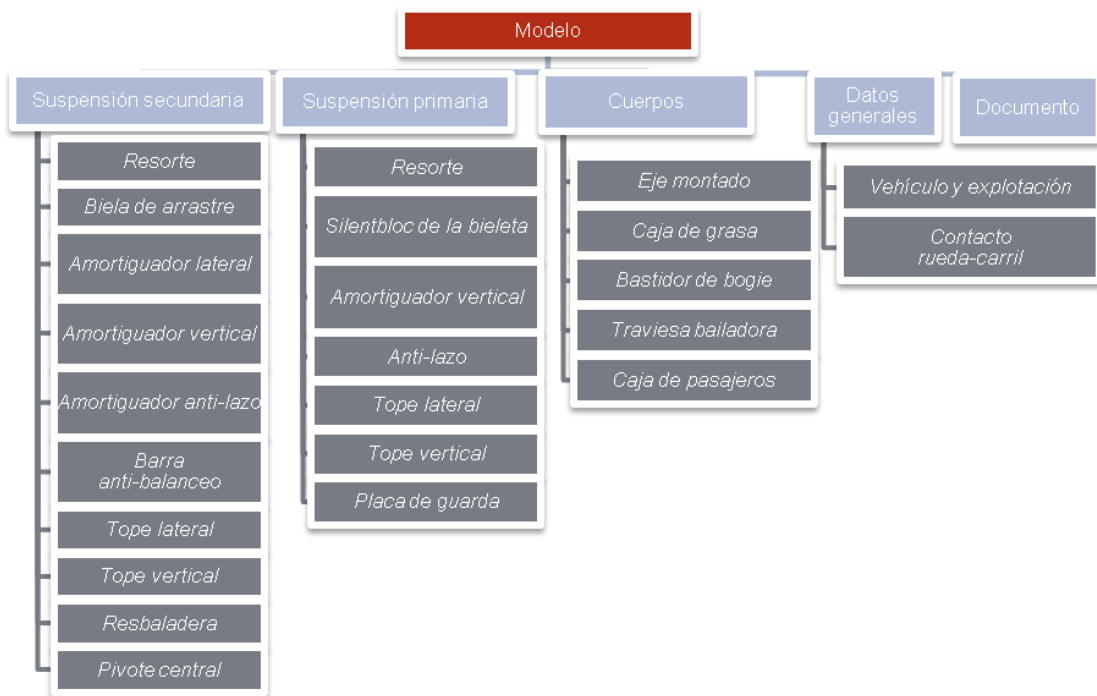


Figura 1 Grupos de datos para un modelo de vehículo ferroviario

A partir de este esquema general, la información necesaria para caracterizar a cada componente se ha desglosado a su vez en pedazos de información más pequeños. A modo de ejemplo, la información correspondiente a los muelles de la suspensión podría esquematizarse de forma estructurada, según se indica en la siguiente figura.

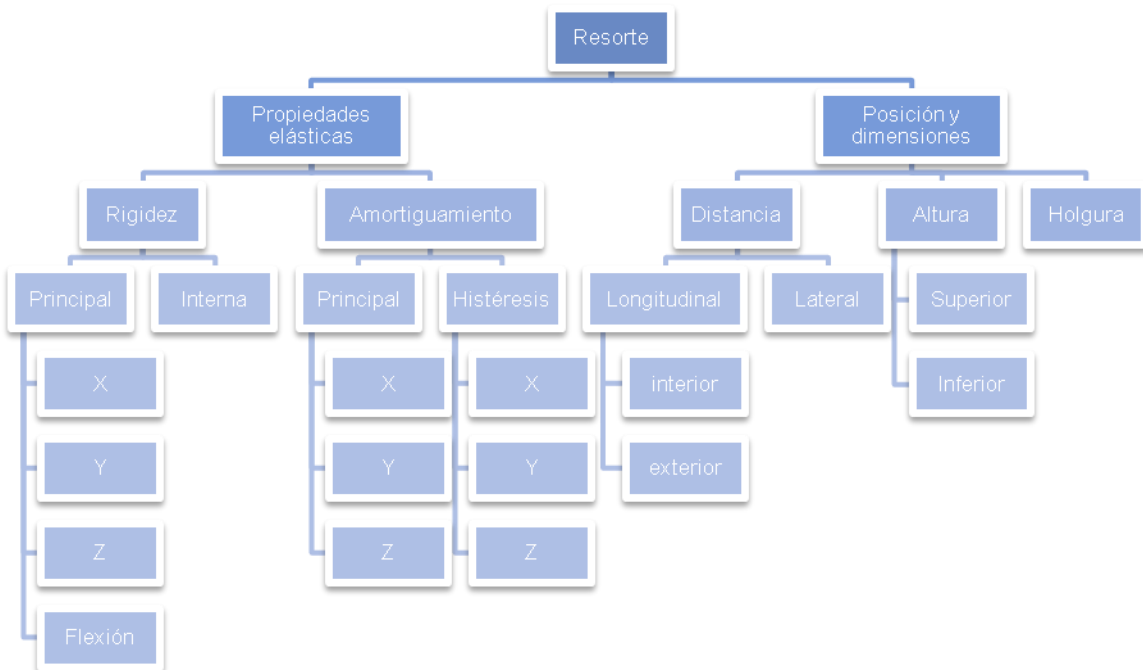


Figura 2 Desglose de datos del resorte de la suspensión primaria

#### 4. ELECCIÓN DEL MODELO DE DATOS

El siguiente paso en el proceso de diseño de la base de datos sería definir el esquema lógico de la base de datos, a partir del esquema conceptual. Sin embargo, el esquema lógico depende de un modelo de datos en particular, por lo que antes de continuar es preciso elegir un modelo de datos adecuado, entendiendo por modelo de datos la forma de organizar el esquema de la base de datos o, lo que es lo mismo, de modelar la estructura de la base de datos.

La información disponible se caracteriza por tener una procedencia muy heterogénea, por lo que los modelos de vehículo muestran grandes variaciones. Esto se refleja tanto en el número de componentes utilizado para describir cada vehículo, como en el grado de detalle con que se describe cada uno de estos componentes. Esta característica condiciona la estructura de la base de datos, que debe ser lo suficiente flexible para admitir un número variable de componentes, y lo suficientemente compleja para adaptarse a todas las posibles caracterizaciones de un determinado componente. También debe tener una estructura fácilmente ampliable o extensible, que permita incorporar nuevos campos durante el proceso de introducción de datos, a fin de recoger aquella información que no se haya contemplado en la fase inicial de definición del esquema.

Atendiendo a estos requisitos, se ha optado por utilizar un modelo de datos XML, que permite definir una estructuración jerárquica, extensible y capaz de adaptarse a cualquier grado de complejidad. Otra ventaja de usar un modelo de datos XML es el fácil acceso a software especializado, incluso de difusión gratuita. Además, es independiente del sistema operativo utilizado, por lo que facilita el intercambio de información con otros usuarios, con independencia de su localización geográfica.

Un requisito importante que ha condicionado la elección de este modelo es su extensibilidad. Esta característica resulta de gran interés para compartir la información con otros usuarios a través de internet, ofreciéndoles la posibilidad de adaptar el esquema de la base de datos a sus necesidades.

Finalmente, para la construcción de la base de datos RVDynBD se ha optado por una base de datos XML nativa, en la que se almacenen directamente los ficheros XML con la descripción de los vehículos ferroviarios.

## **5. DEFINICIÓN DEL ESQUEMA DE LA BASE DE DATOS**

Una vez construido el esquema conceptual se procede a la construcción del esquema lógico global del sistema, también llamado esquema de la base de datos. Mientras que el esquema conceptual representa a la realidad, el esquema lógico se corresponde con la descripción de los datos de la base de datos propiamente dicha. El paso de un esquema a otro es un proceso directo, dado que la única información nueva es la definición de índices y consultas que deben implementarse en la base de datos.

Al haber escogido un modelo de datos XML para la creación de la base de datos RVDynDB, se ha definido un nuevo lenguaje estándar, basado en XML, que permita estructurar la información referente a los parámetros geométricos y mecánicos que caracterizan la dinámica de los vehículos ferroviarios. Este lenguaje, al que se ha denominado RVDynML (Rail Vehicle Dynamic parameters Markup Language), se define en un documento XML Schema, en el que se especifica la estructura de la información que debe incluirse en los ficheros de datos XML.

El lenguaje RVDynML se define en un documento XML Schema, en el que se especifica la información que debe incluirse en los ficheros de datos con las características geométricas y mecánicas de los vehículos ferroviarios, así como la forma en que debe estructurarse esta información. Los elementos que lo componen se organizan en una estructura jerárquica en forma de árbol.

En las siguientes figuras se presentan algunos fragmentos del esquema XML correspondientes a algunos de los principales componentes del vehículo. Aunque XML tiene formato de texto, para facilitar la interpretación del esquema, se ha utilizado una simbología gráfica. Los elementos se representan por un rectángulo; las secuencias ordenadas de elementos se representan con el símbolo y las opciones de selección entre varios elementos con el símbolo ; el símbolo situado detrás de un elemento indica que este contiene otros elementos anidados; finalmente, los tipos complejos, que se repiten en distintas partes del esquema, se encierran dentro de un recuadro amarillo.

El elemento raíz del esquema XML es el modelo de vehículo, del que dependen los restantes elementos. En el último nivel del árbol se encuentran los parámetros físicos a los que debe asignarse algún valor numérico, tales como dimensiones, masas, rigideces, amortiguamientos, etc.

En la siguiente figura se muestra el esquema general, correspondiente al modelo de vehículo. El modelo de vehículo contiene cinco elementos anidados: fuente bibliográfica, datos generales, cuerpos, suspensiones y acoplamientos. En el primero se definen las características de identificación del documento fuente del que se han extraído los datos del modelo de vehículo; en el segundo las particularidades del vehículo y la explotación, así como del contacto rueda-carril; en el tercero las características de los distintos cuerpos que componen el vehículo; en el cuarto las de los componentes de los sistemas de suspensión del vehículo; y en el quinto las de los sistemas de unión entre coches, como el sistema de enganche o los topes.

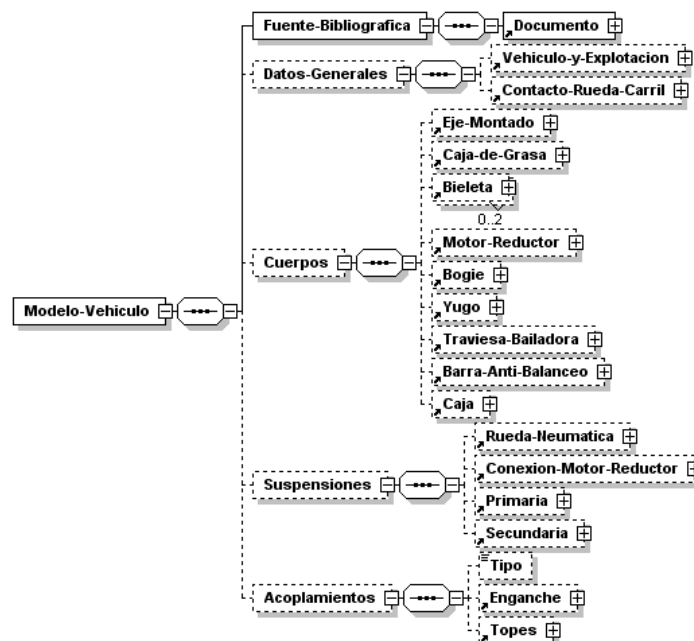


Figura 3 Esquema XML: Modelo-Vehículo

A modo de ejemplo, la siguiente figura muestra la parte del esquema correspondiente a los componentes de la suspensión primaria: resorte; bieleta; amortiguador vertical; antilazo; unión entre ejes; anti-balanceo; topes longitudinal, lateral y vertical, y placa de guarda. Las propiedades de cada componente de la suspensión se han agrupado en propiedades elásticas y posición-y-dimensiones. La primera categoría engloba las rigideces, amortiguamientos, coeficientes de rozamiento y holguras, mientras que la segunda hace referencia, fundamentalmente, a la posición que ocupa cada componente dentro del vehículo.

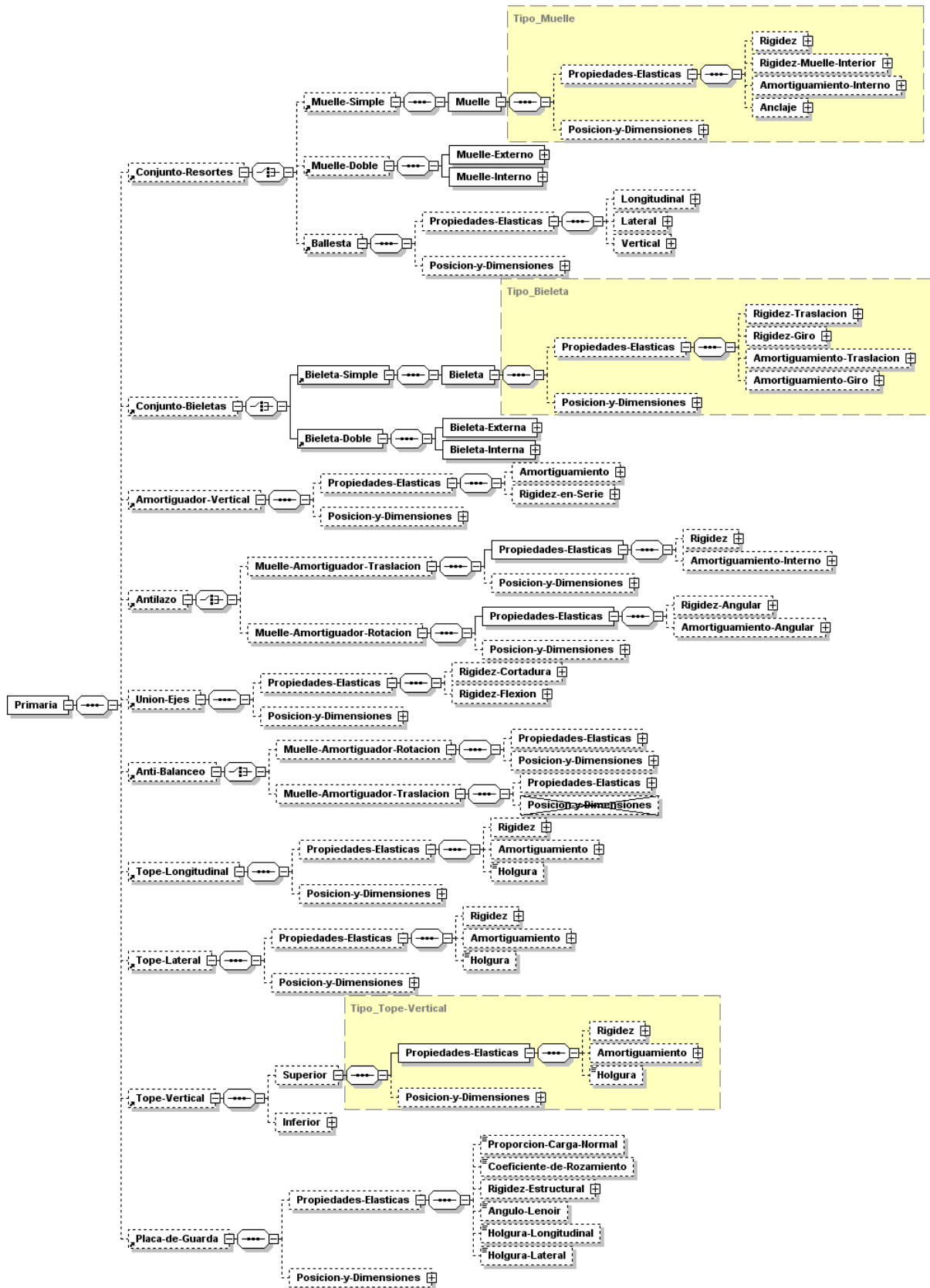


Figura 4 Esquema XML: Modelo-Vehículo/Suspensiones/Primaria

## 6. ELECCIÓN DEL SISTEMA DE GESTIÓN DE LA BASE DE DATOS



El sistema de gestión de bases de datos (SGBD, en inglés DBMS, Database Management System) es el conjunto de programas empleados para definir el esquema de la base de datos, almacenar los datos y realizar consultas y actualizaciones. Al haberse optado por una base de datos XML nativa, el SGBD deberá ser compatible con este tipo de bases de datos.

Atendiendo a la gran cantidad de información que se desea almacenar en la base de datos, y a la heterogeneidad de los documentos fuente, conviene facilitar al máximo la entrada de datos. En este sentido, interesa una entrada de datos amigable, que facilite la selección del tipo de datos a introducir en cada momento, permitiendo que los datos puedan introducirse en el mismo orden en que aparezcan en cada documento fuente, independientemente de cómo esté estructurado el esquema de la base de datos. Una buena alternativa es el uso de editores XML.

Se ha escogido el editor XMLSpy tanto para construir el esquema XML como para generar cada uno de los documento XML que forman parte de la base de datos. Este programa incluye una parte gráfica para el diseño de esquemas. Ofrece también una representación concisa de los documentos XML, que permite ver y manipular de forma gráfica los elementos contenidos en ellos, como si se tratase de los datos reales que almacenan. Además, permite adaptar el contenido de los archivos XML al esquema de forma interactiva, activando los campos a rellenar conforme se definen los elementos situados en un nivel jerárquico superior, lo que facilita la edición del documento y reduce el riesgo de errores.

Una vez completada la implementación de la base de datos, es preciso disponer también de una herramienta que permita realizar búsquedas en la base de datos. Con esta finalidad, se ha desarrollado una función específica en MATLAB que se encarga de gestionar las búsquedas en la base de datos. Para ello se ha utilizado una API de Java para el procesado de documentos XML (API for XML Processing).

## **7. IMPLEMENTACIÓN DE LA BASE DE DATOS**

Una vez definido el esquema de la base de datos se procede a introducir los datos, utilizando el SGBD elegido. Los datos incluidos en cada documento fuente se han introducido en un archivo XML independiente, siempre de acuerdo con la estructura definida por el esquema propio del lenguaje RVDynML, frente al que se ha validado cada fichero de datos. En este proceso se ha utilizado el programa XMLSpy, que permite ir incorporando de forma gráfica los elementos del esquema XML a medida que se desean incluir. De este modo, se consigue construir un archivo XML válido de forma semiautomática, restando tan sólo introducir los valores numéricos apropiados en las casillas correspondientes.

A modo ilustrativo, la siguiente figura muestra una vista de la ventana gráfica del editor XMLSpy, correspondiente a un registro típico asociado a un catálogo de fabricante.

Modelo-Vehiculo	
Fuente-Bibliografica	Documento
	Archivo: CAF_20050503052433_Civia_A.pdf
	Pagina-Web
	U...: www.caf.net
Datos-Generales	Vehiculo-y-Explotacion
	Tipo-de-Transporte: Cercanias
	Tipo-de-Traccion: EMU
	Caracteristicas-Especiales: Unidad Articulada
	Fabricante: CAF
	Administracion-Ferrovial: Renfe
	Serie-Modelo: Civia A1
	Velocidad
	Valor: Rbc Text 120
Cuerpos	Eje-Montado
	Posicion-y-Dimensiones
	Rueda
	Radio: Rbc Text 0.445
	Bogie
	Bastidor
	Posicion-y-Dimensiones
	Lx-Empate: Rbc Text 1.35
	Caja
	Posicion-y-Dimensiones
	Lx-Empate: Rbc Text 8.7
	Lx-Carga: Rbc Text 22

Figura 5 Registro típico de catálogo de fabricante (vista del editor XMLSpy)

En la siguiente figura se refleja el tamaño medio de los registros de la base de datos asociados a cada una de las carpetas en las que se ha organizado la información, en términos porcentuales.

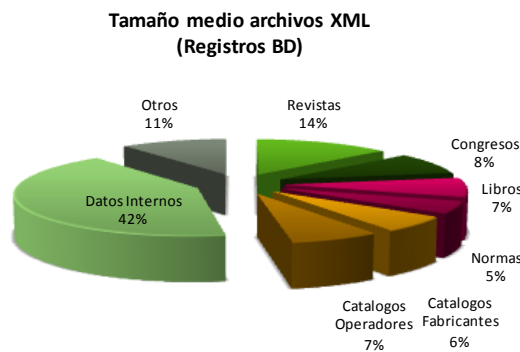


Figura 6 Distribución de registros por tamaño medio de los archivo XML

Otro aspecto interesante a tener en cuenta es el tipo de vehículos que se han almacenado en la base de datos. La siguiente figura presenta las distribuciones en términos porcentuales. Como puede apreciarse, las dos terceras partes de los vehículos son de pasajeros. Dentro de esta categoría, aproximadamente el 40 % corresponden a trenes de largo o medio recorrido, el 20 % a trenes de ámbito urbano, y el 40 % restante son de tipo desconocido.

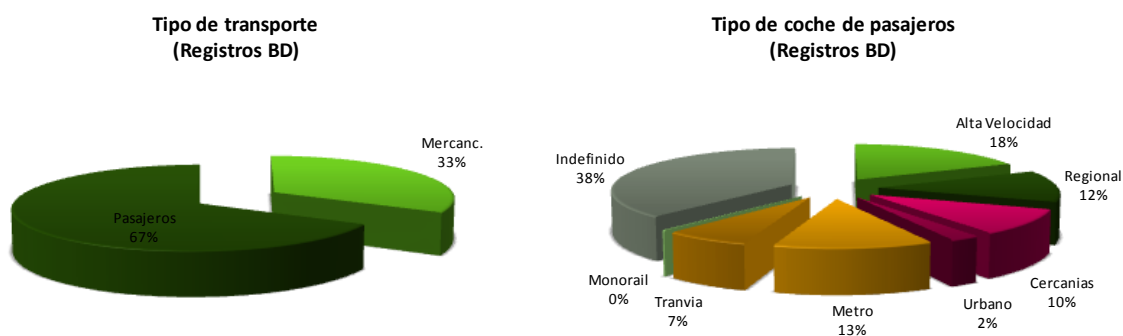


Figura 7 Distribución de registros por tipo de vehículo

## 8. DEFINICIÓN DE LAS CONSULTAS A LA BASE DE DATOS

Se distinguen dos tipos de consultas sobre la base de datos RVDynDB, en función del tratamiento posterior que se pretende realizar con los resultados de las búsquedas: para recuperación de parámetros indirectos y para estimación de parámetros indirectos.

En las primeras, que pueden ser simples o combinadas, se recuperan parámetros directamente almacenados en la base de datos. Los parámetros habituales en este tipo de búsquedas son:

- Carga por eje y carga por longitud.
- Masa, momentos de inercia y altura del c.d.g. del eje, del bogie, y de la caja.
- Empate del bogie y de la caja.
- Rigidez y amortiguamiento de las suspensiones primaria y secundaria.

En las consultas para la estimación de parámetros indirectos, se localizan diversos parámetros almacenados en la base de datos, y se realiza con ellos una operación matemática para estimar otro parámetro de forma indirecta.

La sintaxis de las consultas de la base de datos se basa en el hecho de que el documento XML tiene estructura de árbol. Dado que es posible acceder a cualquier elemento recorriendo el árbol, la consulta debe especificar una secuencia de nodos, que definan una trayectoria que permita recorrer la estructura del árbol hasta llegar al nodo en el que se encuentra el dato que se está buscando.

## 9. INTERCAMBIO DE DATOS CON OTROS USUARIOS

Una vez construida la base de datos RVDynDB, se pretende que esta constituya un repositorio abierto para intercambiar información sobre modelos de vehículos ferroviarios, de tal modo que otros usuarios puedan acceder a su contenido, al tiempo que contribuyan a enriquecerlo aportando información sobre otros vehículos. Con este propósito, se ha previsto publicar la base de datos en internet, y permitir el libre acceso a su contenido completo a aquellos usuarios debidamente registrados, con excepción de los datos de origen privado, si bien estos no llegan al 4 % del total.

La dinámica ideal para este sistema de intercambio de información sería que estuviese permitida tanto la descarga como la subida de documentos, con el propósito de ampliar la base de datos con las aportaciones de los usuarios registrados procedentes de todo el mundo. En este sentido, resultan de gran utilidad las

características de portabilidad y extensibilidad aportadas por XML. La portabilidad permitirá acceder al contenido con independencia de la plataforma (hardware + sistema operativo). En cuanto a la extensibilidad, posibilitará que los usuarios externos puedan añadir nuevos campos al esquema XML en caso de que lo consideren necesario, contribuyendo con estas aportaciones a enriquecer la estructura de datos. Otra de las ventajas que aporta XML en relación con este uso de la base de datos es el hecho de ser “auto-descriptivo”, facilitando así la interpretación de la estructura de datos a cualquier usuario.

Por último cabe señalar que, en el caso de que llegara a generalizarse el uso de este esquema XML, el lenguaje RVDynML podría llegar a convertirse en un estándar para el intercambio de información sobre los parámetros dinámicos de los vehículos ferroviarios.

## 10. REFERENCIAS

1. Document Object Model (DOM) Level 2 Core Specification. (13 de Noviembre de 2000). Recuperado el 30 de Junio de 2007, de W3C: <http://www.w3.org/TR/DOM-Level-2-Core/>
2. Extensible Markup Language (XML). (s.f.). Recuperado el 30 de Junio de 2007, de W3C: <http://www.w3.org/XML/>
3. Garcia-de-Jalon, J., Vidal, J., & Alvarez, E. (2004). MechML (Mechanism Markup Language). Congresso de Métodos Computacionais em Engenharia. Lisboa.
4. Harold, E. R., & Means, W. S. (2004). XML in a Nutshell. O'Reilly.
5. Korth, H., & Silberschatz. (1994). Fundamentos de bases de datos. Madrid: McGraw.Hill.
6. Muench, S. (2000). Building Oracle XML Applications. O'Reilly.
7. Package org.w3c.dom. (s.f.). Recuperado el 30 de Junio de 2007, de Sun Developer Network (SDN): <http://java.sun.com/j2se/1.4.2/docs/api/org/w3c/dom/package-summary.html>
8. Ray, E. T. (2003). Learning XML. O'Reilly.
9. Riordan, R. M. (2005). Designing Effective Database Systems. Addison Wesley Professional.
10. Rodriguez, J. I., Jimenez, J. M., Funes, F. J., & Garcia-de-Jalon, J. (2003). Dynamic Simulation of Multi-Body Systems on Internet Using CORBA, Java and XML. Multibody System Dynamics , 10, págs. 177-199.
11. The Extensible Stylesheet Language Family (XSL). (s.f.). Recuperado el 30 de Junio de 2007, de W3C: <http://www.w3.org/Style/XSL>
12. XML Schema. (s.f.). Recuperado el 30 de Junio de 2007, de W3C: <http://www.w3.org/XML/Schema>
13. XQuery 1.0: An XML Query Language. (23 de Enero de 2007). Recuperado el 30 de Junio de 2007, de W3C: <http://www.w3.org/TR/xquery/>