

PROYECTO FIN DE GRADO

TÍTULO: Diseño de Blockchain basado en un modelo de negocio para Smart Campus

AUTOR/A: Leticia Marcos Avendaño

TITULACIÓN: Ingeniería Electrónica de Comunicaciones

TUTOR/A: Margarita Martínez Núñez

DEPARTAMENTO: U.D. de Ingeniería de Organización, Administración de Empresas y Estadística

VºBº TUTOR/A

Miembros del Tribunal Calificador:

PRESIDENTE/A: Carlos Felipe Rueda Frías

TUTOR/A: Margarita Martínez Núñez

SECRETARIO/A: Margarita Millán Valenzuela

Fecha de lectura: 17/07/2024

Calificación:

El Secretario/La Secretaria,

Resumen

A veces, residimos en espacios que no cumplen una serie de condiciones saludables, ya sea por tener una contaminación del aire alta, la elevada temperatura o una mala acústica. Esto supone un riesgo para la salud de las personas.

Para abordar este problema, tenemos que asegurar que las condiciones del espacio son buenas realizando un seguimiento a través de sensores. Una forma efectiva de realizar todo este proceso será utilizando la combinación de sensores y la tecnología Blockchain.

Blockchain es una tecnología que se basa en una cadena de bloques descentralizada, que hace que no sea necesario depender de intermediarios, en la que la información es inmutable y segura, y que está almacenada en todos los nodos que conforman la red de manera que cualquier modificación de la información puede detectarse fácilmente. En sus inicios estaba vinculada a las criptomonedas, como Bitcoin, pero actualmente se ha extendido a diferentes sectores de calidad y sostenibilidad, siendo uno de ellos la creación de Smart Campus.

La obtención de datos se realizará a través de sensores en el que tomarán muchas muestras y procesarán la información en cuestión de minutos.

El objetivo principal de este trabajo de fin de titulación se basa el desarrollo de una Blockchain basado en un modelo de negocio para Smart Campus. El modelo de negocio se centrará en monitorizar las condiciones ambientales de los espacios más concurridos para llevar un seguimiento y poder garantizar y demostrar la seguridad dentro de los Smart Campus. Así mismo, gracias a la Blockchain se podrá tener un historial de los datos y posteriormente utilizarlos para solicitar las certificaciones que acrediten el cumplimiento del Reglamento de Instalaciones Térmicas en Edificios (RITE).

Los sensores estarán colocados en puntos estratégicos del aula y registrarán la información en la cadena de bloques. De esta forma, se podrá conocer de forma precisa y transparente la procedencia de los datos tomados y así evitar la falsificación de ellos.

Este documento pretende detallar la creación de un modelo de negocio consistente para Smart Campus, en el que su finalidad será plasmar cuáles son las condiciones idóneas de un aula para mejorar la experiencia educativa y la colaboración entre las diversas partes que forman parte de la comunidad de un campus universitario.

Abstract

Sometimes, we live in spaces that do not meet several healthy conditions, either because of high air pollution, high temperatures or poor acoustics. This poses a risk to people's health.

To solve this problem, we need to ensure that the conditions in the space are good by monitoring through sensors. An effective way to do this will be to use a combination of sensors and blockchain technology.

Blockchain is a technology based on a decentralized blockchain, which makes it unnecessary to depend on intermediaries, in which the information is immutable and secure, and which is stored in all the nodes that make up the network so that any modification of the information can be easily detected. In its beginnings it was linked to cryptocurrencies, such as Bitcoin, but it has now spread to different quality and sustainability sectors, one of them being the creation of Smart Campus.

Data collection will be conducted through sensors that will take many samples and process the information within minutes.

The main objective of this final degree project is the development of a Blockchain-based business model for Smart Campuses. The business model will focus on monitoring the environmental conditions of the most frequented spaces to keep track and be able to guarantee and demonstrate safety within Smart Campuses. Additionally, thanks to Blockchain, a history of the data can be kept and later used to request certifications that attest to compliance with the Regulations for Thermal Installations in Buildings.

The sensors will be placed at strategic points in the classroom and will record the information on the blockchain. In this way, the origin of the data collected can be precisely and transparently known, preventing their falsification.

This document aims to detail the creation of a consistent business model for Smart Campuses, with the purpose of outlining the ideal conditions of a classroom to improve the educational experience and collaboration among the various parts that make up a university campus community.

Índice de figuras

Figura 1: Bloque de una Blockchain	6
Figura 2: Funcionamiento de Proof of Work [6]	7
Figura 3: Ejemplo de Smart Campus con IoT [10]	8
Figura 4: Venta directa [elaboración propia]	10
Figura 5: Suscripción [elaboración propia]	10
Figura 6: Marketplace [elaboración propia]	10
Figura 7: Licencia de software [elaboración propia]	10
Figura 8: Blockchain [elaboración propia]	11
Figura 9: Ejemplo de Plan Cruncher [40]	11
Figura 10: Metodología Lean Startup [41]	12
Figura 11: Plantilla Canvas	13
Figura 12: Segmentos de clientes [36]	14
Figura 13: Propuesta de valor [37]	15
Figura 14: Canales [38]	16
Figura 15: Relación con los clientes [34]	17
Figura 16: Fuente de ingresos	18
Figura 17: Recursos clave [39]	19
Figura 18: Actividades clave	19
Figura 19: Esquemático del sensor	25
Figura 20: Sensores	25
Figura 21: Columnas del csv	26
Figura 22: Java [28]	27
Figura 23: Eclipse [29]	27
Figura 24: GitHub [44]	28
Figura 25: httpie [42]	28
Figura 26: Spring framework [44]	29
Figura 27: Diagrama de red	29
Figura 28: Generación del token	31
Figura 29: Ejemplo de petición GET	32
Figura 30: Petición GET - Cabeceras	33
Figura 31: Petición GET - Autenticación	33
Figura 32: Respuesta de la petición GET	34
Figura 33: Ejemplo petición GET	34
Figura 34: Prueba petición GET - Contenido del repositorio	35
Figura 35: Petición GET - Obtención URL datos	35
Figura 36: Petición GET - Visualización de los datos	36
Figura 37: Paquete de GitHub	36
Figura 38: Solicitud HTTP	37
Figura 39: Obtención del repositorio y user	37
Figura 40: Obtención URLs de descarga	38

Figura 41: Paquete y clase de DataSensors	38
Figura 42: Método para la media	39
Figura 43: Paquete de las notificaciones	39
Figura 44: Clase Notification	40
Figura 45: Constantes RITE	40
Figura 46: Generación de las notificaciones	40
Figura 47: Servicio de notificaciones	41
Figura 48: Paquete y clases de la Blockchain	41
Figura 49: Método applySha256	42
Figura 50: Método calculateHash	42
Figura 51: Método difficultyBlock	43
Figura 52: Clase Blockchain	44
Figura 53: Método verificación de cadena	44
Figura 54: Paquete monitorización	44
Figura 55: Clase principal	45
Figura 56: Página de login	45
Figura 57: Ruta para acceder a los datos	46
Figura 58: Fichero pom.xml	47
Figura 59: Organización de los recursos estáticos	47
Figura 60: Modelo de negocio	48
Figura 61: Asociaciones clave	54
Figura 62: Acceso	56
Figura 63: Inicio de la aplicación	56
Figura 64: Gráficos de las medidas	57
Figura 65: Representación de la Blockchain	57
Figura 66: Notificaciones	58
Figura 67: Eclipse Maketplace	71
Figura 68: Spring Tools	72
Figura 69: Instalación Spring	72
Figura 70: Confiamos en los desarrolladores	73
Figura 71: Thymeleaf	73
Figura 72: Importación proyecto	74
Figura 73: Selección del proyecto	75

Índice de tablas

Tabla 1: Diferencias estructura de costes	21
Tabla 2: Diferencia tokens	31

Lista de acrónimos

Abreviatura	Significado
API	Application Programming Interfaces
BMC	Business Model Canvas
BREEAM	Building Research Establishment Environmental Assessment Method
Co2	Dióxido de carbono
csv	Comma separated values
ETSISI	Escuela Técnica Superior de Ingeniería de Sistemas Informáticos
HTTP	Hypertext Transfer Protocol
IDE	Entorno de Desarrollo Integrado
I+D+i	Investigación, Desarrollo e Innovación
IoT	Internet of Things
LEED	Leadership in Energy and Environmental Design
Nonce	Number that can be only used once
PoW	Proof-of-Work
ppm	Partes por millón
REST	Representational State Transfer
RH	Humedad relativa
SlotCom	Sistemas IoT Conscientes de la Sostenibilidad y Dirigidos por Comunidades Sociales
SPI	Serial Peripheral Interface
UPM	Universidad Politécnica de Madrid

Índice de contenidos

Resumen	i
Abstract	iii
Índice de figuras	iv
Índice de tablas	vi
Lista de acrónimos.....	vii
1. Introducción	1
1.1 Contexto.....	1
1.2 Objetivos.....	2
1.3 Estructura del resto de la memoria	2
2. Marco tecnológico.....	5
2.1 Blockchain	5
2.1.1 Elementos de la Blockchain.....	5
2.1.2 Smart Campus	7
2.2 Modelo de negocio	9
2.2.1 Tipos de modelo de negocio y ejemplos.....	9
2.2.2 Metodologías.....	11
3. Especificaciones y restricciones de diseño	23
3.1 Sensores.....	23
3.1.1 Especificaciones	23
3.1.2 Datos	25
3.2 Herramientas utilizadas	27
3.2.1 Java.....	27
3.2.2 Eclipse.....	27
3.2.3 GitHub.....	27
3.2.4 HTTPie.....	28
3.2.5 Spring.....	28
3.3 Diagrama de red	29
4. Solución tecnológica.....	31
4.1 Llamadas a la API de GitHub y pruebas	31
4.2 Desarrollo del código	36
4.3 Implementación de Spring.....	44
5. Solución modelo de negocio	48
5.1 Segmentos de clientes	48
5.2 Propuesta de valor.....	49
5.3 Canales.....	51
5.4 Relaciones con los clientes	51
5.5 Fuente de ingresos.....	52
5.6 Recursos clave.....	52

5.7	Actividades clave	53
5.8	Asociaciones clave	53
5.9	Estructura de costes	54
5.10	Gobernanza de la Blockchain	55
6.	Resultados	55
7.	Presupuesto	59
8.	Impacto del proyecto	61
8.1	Salud	61
8.2	Ambiental	61
8.3	Social y tecnológico	61
8.4	Económico	62
9.	Conclusiones	63
9.1	Conclusiones.....	63
9.1	Trabajos futuros	63
10.	Referencias	65
	Manual de usuario	71
A.1	Descargar Eclipse.....	71
A.2	Instalación de Spring dentro de Eclipse	71
A.3	Instalación de Thymeleaf	73
A.4	Importación del proyecto.....	74

1. Introducción

En esta sección se tratará de dar una breve introducción sobre las tecnologías usadas a lo largo del proyecto, así como los objetivos que se han propuesto realizar.

Además, se dará una pequeña sinopsis de los contenidos que se verán en el trabajo de fin de grado.

1.1 Contexto

En los últimos años, ha aumentado la conciencia de proteger el medio ambiente y esto ha llevado a un aumento en la monitorización y medición de datos ambientales en diferentes entornos, como edificios, ciudades e incluso en aulas. La monitorización en las aulas universitarias mediante sensores implica el comienzo hacia la transformación de los campus tradicionales a los llamados Smart Campus. La recopilación de grandes cantidades de datos presenta un papel fundamental para adoptar medidas de sostenibilidad y reducir el cambio climático, así como para garantizar las buenas condiciones ambientales.

Sin embargo, con la toma de estos datos, ha surgido preocupación por la veracidad de la información, ya que se pueden manipular para ocultar la gravedad de la situación y cumplir con estándares. El incumplimiento de estos estándares o reglamentos y la falsificación de la información puede suponer grandes multas ya que es considerado un delito.

Este proyecto se centrará en las aulas universitarias para avanzar hacia la transformación en un Smart Campus, donde la manipulación de datos podría alterar mediciones de calidad del aire interior, humedad o temperatura, únicamente para cumplir con normativas, lo que comprometería tanto la salud de los estudiantes como la protección del medio ambiente.

La solución propuesta en este proyecto para afrontar la falsificación de los datos es la utilización de la tecnología Blockchain. Esta tecnología permite almacenar información de forma inmutable y transparente, ya que cuando los datos se registran en la cadena de bloques, no podrán ser alterados sin dejar rastro evidente. También se propondrá un sistema de monitorización en el que será posible visualizar los datos tomados por los sensores.

Es importante destacar que este proyecto fin de grado está apoyado en el proyecto llamado 'Sistemas IoT Conscientes de la Sostenibilidad y Dirigidos por Comunidades Sociales' (SlOTCom) incorporado en el Plan Nacional de proyectos de Investigación, Desarrollo e Innovación (I+D+i). SlOTCom forma parte de un experimento en el que se está haciendo uso de los sensores incorporados en las aulas de la Escuela Técnica Superior de Ingeniería de Sistemas Informáticos (ETSISI).

1.2 Objetivos

Los objetivos principales que se pretenden conseguir en este trabajo fin de grado son:

- Estudio e implementación de la tecnología Blockchain y sus aplicaciones. Para ello, se desarrollará una Blockchain que integrará los datos provenientes de sensores ubicados en las aulas de la ETSISI.
- Análisis de los modelos de negocio orientados hacia Blockchain. Se estudiarán casos de estudio relevantes y se identificarán las mejores prácticas en la implementación de Blockchain.
- Por último, incorporación de la Blockchain al modelo de negocio. Se diseñará un plan detallado para la implementación de la Blockchain y se evaluarán los posibles impactos y mejoras en el modelo de negocio.

Y, por último, el objetivo general que se pretende conseguir con este proyecto es solventar aquellos problemas que puedan estar producidos por malas condiciones ambientales dentro de las aulas. Para ello, el sistema de monitorización nos facilitará detectar cualquier de estos problemas, así como evitar la falsificación de los datos obtenidos por los sensores. Por lo que podremos decir, que lo que se espera es mejorar la experiencia educativa y la colaboración entre las diversas partes que forman parte de la comunidad de este campus.

1.3 Estructura del resto de la memoria

Para la obtención de los objetivos mencionados en el apartado anterior, este proyecto se ha estructurado en un total de ocho capítulos. A continuación, se refleja una breve explicación de cada uno de ellos:

El primer capítulo está desarrollado en la sección anterior, donde se realiza una introducción de la idea desarrollar junto con los objetivos propuestos para este proyecto. En el capítulo 2, se describirá el marco tecnológico que estará compuesto por la tecnología Blockchain y sus componentes principales y el concepto de Smart Campus. También se describirá el concepto de modelo de negocio y las metodologías que se utilizan para su creación haciendo más hincapié en la metodología Canvas.

El capítulo 3 describe las especificaciones y restricciones de diseño, donde se explicará la toma de medidas que realizan los sensores y la obtención de los datos y las herramientas que se utilizarán para la creación de este proyecto.

En el cuarto capítulo se propondrá la solución tecnológica propuesta que se ha llevado a cabo para la elaboración de este proyecto. Iniciando el capítulo en la creación de las peticiones al repositorio, siguiendo por la creación de la Blockchain y el sistema de monitorización. En esta misma línea, en el capítulo cinco se ha desarrollado la propuesta planteada para el modelo

de negocio incluyendo un elemento clave como la tecnología Blockchain. Y será en el capítulo seis dónde mostrará como ha quedado la solución tecnológica propuesta, añadiendo capturas de las partes más importantes de nuestro sistemas de monitorización.

En el capítulo siete se abordará el presupuesto que se estima para la creación de este proyecto, abarcando desde los sensores hasta la creación de la Blockchain.

La identificación de los aspectos económicos, sociales y ambientales más relevantes tras la realización de este proyecto se llevará a cabo en el capítulo ocho.

Y para finalizar con la estructura principal, en el capítulo nueve se expondrá la conclusión tras la realización de este proyecto y las posibles líneas futuras que se pueden abarcar.

También, al final de esta memoria se añade un manual de usuario enfocado a dónde se describe cómo implementar el sistema de monitorización en otro portátil.

2. Marco tecnológico

En este capítulo se define el concepto de Blockchain, cuáles son sus características más importantes, cuáles son los elementos que constituyen una cadena de bloques concluyendo con el concepto de Smart Campus. Por otro lado, se definirá el concepto de modelo de negocio, la importancia de aplicarlos, los modelos más importantes que existen y las metodologías que hay disponibles para la creación.

2.1 Blockchain

La Blockchain es una tecnología que permite almacenar información de una manera que es casi imposible de falsificar. Se puede asemejar a un libro de contabilidad o a un libro electrónico público en el que se puede compartir entre usuarios y que crea un registro inmutable de sus transacciones. [47]

Hay una serie de características principales que nos ayudarán a entender de una forma más sencilla esta tecnología:

- **Red descentralizada:** los usuarios no dependen de ninguna autoridad que controle la información que se transmite y se recibe. En Blockchain los datos se distribuyen por la red, en la cual tendremos unos nodos que realizarán las transacciones y almacenarán la información. [48]
- **Cadena de bloques:** es una base de datos que permite almacenar información de forma inmutable y ordenada. Para que esa información se añada a la cadena de bloques debe de ser validada por todas las partes que constituyen la Blockchain para que pueda llegar al siguiente nodo. En cada bloque de la cadena se añaden nuevas transacciones que se han realizado desde el bloque anterior, conteniendo su hash y el hash del anterior bloque y así, sucesivamente.
- **Consensuada:** todos los nodos operan en perfecta sincronización, lo que significa que, si uno de ellos se desincroniza, el sistema lo detecta automáticamente. Esto permite identificar cualquier intento de manipulación de la información. Existen varios algoritmos de consenso disponibles que pueden ser elegidos en función de los objetivos que se quieran reunir en la red.
- **Inmutabilidad:** una transacción requiere confirmación por las diversas partes, por lo que no se puede modificar ni eliminar sin dejar un rastro evidente.

2.1.1 Elementos de la Blockchain

Los principales elementos que forman una Blockchain son: [49]

- **Bloque:** está constituido por un conjunto de transacciones. Cada uno forma parte de la cadena de bloques y a su vez, cada bloque está formado por:
 - **Hash:** es una cadena codificada de letras y números que se vincula a cada bloque individual de forma que cada bloque tendrá su propio hash. También se añade el hash del bloque anterior, por lo que, todos los bloques quedarán encadenados de forma secuencial.
 - **Información:** conjunto de datos que se almacenarán en el nodo junto con una marca de tiempo (*Timestamp*).
 - **Number that can be only used once (Nonce):** es un número arbitrario que funciona en combinación con el hash como elemento de control para evitar la manipulación de la información de los bloques. Esto garantiza que los hashes no se puedan volver a utilizar.

A continuación, se muestra un esquema de básico de lo que consta un bloque:

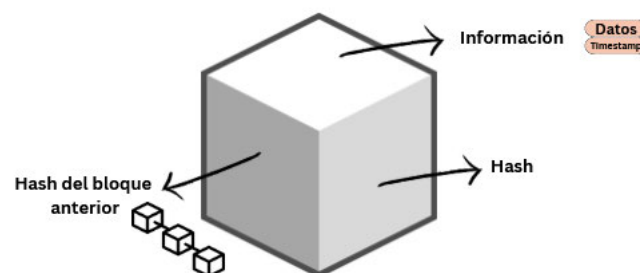


Figura 1: Bloque de una Blockchain

- **Nodo:** son todos los dispositivos que forman parte de la red de Blockchain. Se dedican a actualizar y distribuir una copia actualizada con la información más reciente que se agrega a la cadena de bloques. Por lo tanto, cada vez que un bloque se valida, éste se añade a la cadena de bloques y gracias a esto, los nodos aumentan la seguridad y la integración de la información.
- **Mineros:** son los encargados de validar las transacciones que se dan en la red y generar los nuevos bloques que son añadidos a la Blockchain. Los bloques son validados por el minero y el procedimiento que utilizan es aplicando un algoritmo de consenso. Una de las maneras más conocidas de llevar a cabo este proceso es el *Proof of Work (PoW)*. Este mecanismo requiere una computación muy significativa en los dispositivos que forman la red dado que deben de resolver un número hexadecimal cifrado. Esta computación aumenta cuanto más mineros se unan a la red.

Este procedimiento es utilizado por la criptomoneda Bitcoin, la cual establece un consenso entre los participantes sin necesidad de una autoridad central que sea responsable de su emisión y registro de sus movimientos. Esto hace que los usuarios que poseen esta moneda virtual tengan transacciones rápidas y seguras, ya que hace imposible la falsificación. Una de las ventajas que presenta Bitcoin es su transparencia del valor, es decir, el precio de Bitcoin siempre está publicado en tiempo real y basado en la oferta y demanda y es considerada una divisa mundial. Algunos países como El Salvador adoptaron el Bitcoin como moneda oficial de su país.

En la siguiente figura se resumen el funcionamiento de este algoritmo:



Figura 2: Funcionamiento de Proof of Work [6]

2.1.2 Smart Campus

El Smart Campus es un concepto emergente habilitado por las oportunidades de transformación digital en la educación superior. Los Smart Campus sirven como laboratorios vivos para la investigación, desarrollo y adopción de tecnología inteligente, junto con sus funciones tradicionales de enseñanza, aprendizaje e investigación. [8]

Hoy en día, las instituciones de educación superior están explorando esta tecnología para transformar las experiencias de enseñanza, incluyendo la implementación de nuevos métodos de apoyo y colaboración entre estudiante y personal universitario.

Para poder realizar esta transformación hacia un “campus inteligente” podemos hacer uso de varias tecnologías:

- **Internet of Things (IoT):** El “Internet de las cosas” es la interconexión y agrupación de dispositivos a través de una red. Esto permite recopilar datos que pueden ser transmitidos a una plataforma central donde se puede analizar y utilizar para tomar decisiones o realizar acciones específicas. Hay varias razones por la cual esta tecnología es esencial para la transformación de un campus convencional en un Smart Campus, entre ellas:
 - **Conectividad e integración:** permite la conectividad de una gran variedad de dispositivos, como sensores o cámaras, que facilitan la integración de los datos.
 - **Recopilación de datos en tiempo real:** estos dispositivos pueden recopilar datos en tiempo real como la calidad del aire, la temperatura... que pueden ayudar a responder a las necesidades del campus.
 - **Seguridad y sostenibilidad:** la tecnología IoT puede ayudar a mejorar la seguridad en el campus mediante la monitorización de gestión de accesos y la notificación de incidentes. Así mismo, la gestión de la energía puede contribuir a la sostenibilidad y a la reducción del impacto ambiental.
 - **Mejora de la experiencia del usuario:** puede otorgar a los estudiantes y personal universitario una experiencia mejorada, ya que pueden tener información en tiempo real de los recursos disponibles.

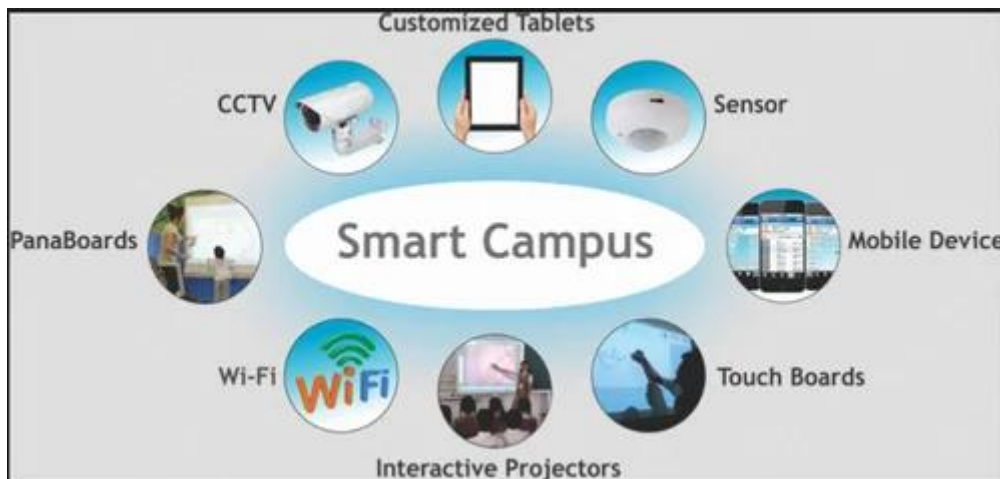


Figura 3: Ejemplo de Smart Campus con IoT [10]

- **Blockchain:** como se comentó anteriormente destaca por ser una tecnología descentralizada, transparente, inmutable y segura y esto aplicado en los Smart Campus nos permite aumentar la trazabilidad y seguridad en los datos almacenados. Gracias a esto, sería posible proteger los datos académicos, información de los estudiantes o datos confidenciales de investigación. Así mismo, se podrían automatizar contratos

inteligentes que se pueden ejecutar en la Blockchain cuando se cumplan ciertas condiciones en el Smart Campus.

También es importante añadir que la transformación de los campus universitarios tradicionales a los Smart Campus no es una tarea sencilla, ya que cada campus tiene sus propias necesidades y limitaciones y no hace que sea un proceso genérico para todos. Una de las ventajas que tiene esta transformación es que abre las puertas a la creación de nuevos modelos de negocio innovadores, como por ejemplo la gestión de datos, la creación de aplicaciones educativas que mejoren la experiencia del alumnado y profesorado o modelos enfocados a la seguridad y sostenibilidad del campus.

En el siguiente apartado se explicará el concepto de modelo de negocio, los tipos que existen y las metodologías que se utilizan para su creación. Basándonos en esta sección, creamos nuestro modelo de negocio para Smart Campus, en el que la monitorización de las aulas mediante los sensores y la incorporación de la Blockchain serán los principales componentes.

2.2 Modelo de negocio

El concepto de modelo de negocio se define por cómo una empresa identifica, crea y ofrece valor a sus clientes, es decir, se centra en cómo una empresa logra satisfacer rentablemente las necesidades de los clientes.

El planteamiento de un modelo de negocio busca obtener un análisis estratégico y visión global de un proyecto empresarial, y para ello se debe plantear cómo la futura empresa generará ingresos y obtendrá ganancias.

Para la creación de un modelo de negocio, se pueden responder a los siguientes preguntas que nos ayudarán a definir un modelo más consistente:

- ¿Quién es el cliente?
- ¿Qué valora el cliente?
- ¿Cómo ganamos dinero en este negocio?
- ¿Cómo podemos ofrecer valor a los clientes a un costo adecuado?

Una vez que se hayan respondido a estas preguntas, podremos empezar a decidir qué tipo de modelo de negocio se adapta más a lo que necesitamos.

2.2.1 Tipos de modelo de negocio y ejemplos

Los tipos de modelo de negocio más importantes que hay en la actualidad son: [53]

-
- **Venta directa:** los consumidores compran productos directamente de los fabricantes o minoristas, es decir, sin intermediarios, como, por ejemplo, Amazon.



Figura 4: Venta directa [elaboración propia]

- **Suscripción:** a través de pagar una suscripción periódica, se accede a un producto específico. Como, por ejemplo, Netflix o Spotify.



Figura 5: Suscripción [elaboración propia]

- **Marketplace:** es un mercado en línea que conectan a compradores y vendedores, es decir, actúan como intermediarios. Como, por ejemplo, Booking o Airbnb.



Figura 6: Marketplace [elaboración propia]

- **Licencia de software:** son empresas que desarrollan software y proporcionan licencias para usar sus productos, como, por ejemplo, Microsoft.



Figura 7: Licencia de software [elaboración propia]

- **Blockchain:** consta de la creación y gestión de plataformas que permiten transacciones seguras y descentralizadas a través de tecnología Blockchain.



Figura 8: Blockchain [elaboración propia]

2.2.2 Metodologías

Existen diversas herramientas que facilitan la creación de modelos de negocio. Las más conocidas son:

2.2.2.1 Plan Cruncher

Con esta herramienta se puede crear un resumen ejecutivo sobre la idea de negocio. Se deben responder a las preguntas típicas y las respuestas se muestran a través de iconos para hacerlo más visible. [40]

<p>Idea</p>  <p>It is a high-risk idea. We have a new idea. This is going to be the best thing since sliced bread. Better, even, than Web 2.0. Sure, everyone says that, but this time it's really true.</p>	<p>Team</p>  <p>We have a business network. We are not going to build it. We have an entrepreneur. We have years of experience in the Dutch 'broodvervangers' industry; our relentless CEO will leverage our existing network of contacts.</p>
<p>Product</p>  <p>We have a working demo. We own the IP. We have samples of our new product. It's crunchy. We also own the rights to our revolutionary new recipe, which is fully patent-free.</p>	<p>Revenue</p>  <p>We will monetise later. We have a revenue forecast. We will start by getting users. In the first phase we will create our market with a freemium breakfast model, followed by direct retail sales and recipe licensing deals.</p>

Figura 9: Ejemplo de Plan Cruncher [40]

2.2.2.2 Lean Startup

Esta metodología es destacable por verificar cada cierto tiempo la hipótesis antes de tener el producto final. El objetivo es definir los ciclos de desarrollo y proyectar diversas ideas obteniendo feedback de los clientes o usuarios. [41]

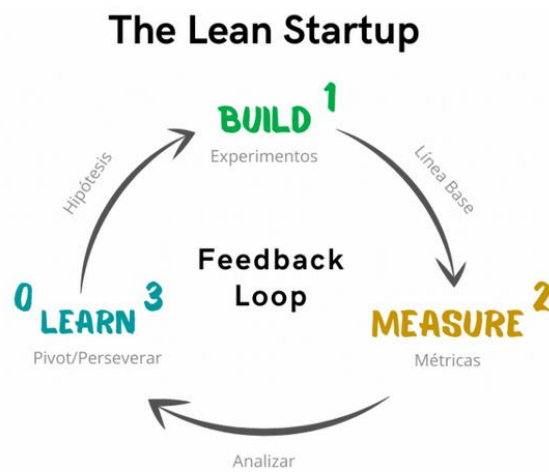


Figura 10: Metodología Lean Startup [41]

2.2.2.3 Modelo Canvas

El Modelo Canvas es una de las metodologías más aplicadas en la creación de negocios innovadores y es la que utilizaremos para nuestro modelo de negocio.

Esta metodología fue desarrollada por Alexandre Osterwalder y Yves Pigneur que permite visualizar, desarrollar, evaluar y modificar modelos de negocio nuevos o existentes y la plasmaron en el libro "Business Model Generation". Es conocido como BMC y está dividido en nueve módulos que cubren las áreas principales para un negocio: oferta, infraestructura, viabilidad económica y clientes. [11]

Existe una plantilla creada con los nueve módulos para que resulte más fácil la implementación del modelo de negocio.

MODELO CANVAS

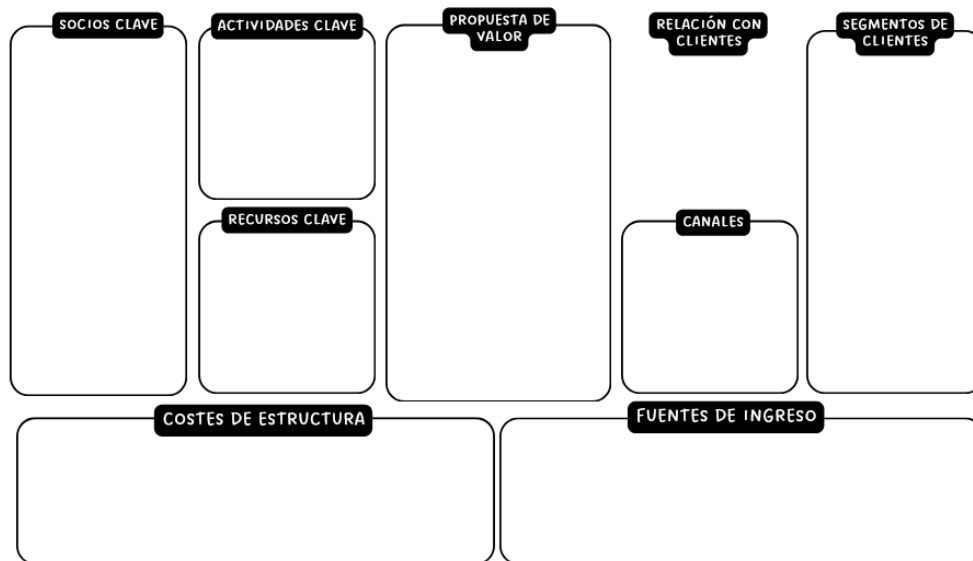


Figura 11: Plantilla Canvas

A continuación, se explicará cada uno de estos módulos para tener un conocimiento previo antes de mostrarlos en la solución final el modelo de negocio propuesto.

1. Segmentos de clientes

Los segmentos de clientes son los diferentes grupos de personas o entidades a los que una empresa quiere hacer llegar su servicio o producto, es decir, es como dividir el mercado de clientes en diferentes tipos pero que comparten características comunes. Por ejemplo, un restaurante tiene varios menús para ofrecer, pero con platos diferentes según el comensal, como puede ser vegano o gourmet.

La segmentación de clientes es un punto muy importante para tener en cuenta cuando se quiere crear un modelo de negocio, porque será la base para definir la propuesta de valor. Existen varios tipos de segmentos de clientes y entre los más destacables encontramos estos cuatro tipos: [60]

- **Mercado masivo:** está centrado en una gran cantidad de clientes sin hacer diferenciación entre ellos, esto quiere decir, que el producto o servicio que ofrecen está para un amplio grupo de personas. Un ejemplo muy simple sería una cadena de supermercados.
- **Nicho de mercado:** este tipo de segmentación es completamente opuesta a la descrita anteriormente y es que se centra en un grupo muy específico de clientes con unas necesidades muy particulares. Está más enfocado a la personalización del producto y, por tanto, la propuesta de valor, los canales

y las relaciones con los clientes están muy enfocados en los clientes. Por ejemplo, una tienda de ropa para ciclistas se dirige a personas que buscan ropa especializada en ciclismo.

- **Segmentado:** el mercado se divide en grupos más pequeños en los que los problemas y las necesidades son distintas pero que están relacionados entre sí. Por ejemplo, una tienda de cosméticos vende productos para diferentes tipos de piel.
- **Diversificado:** este segmento es similar al anterior, pero con la particularidad de que utiliza propuestas de valor completamente diferentes para cada cliente, como puede ser Amazon cuya plataforma proporciona productos para los consumidores generales, pero a su vez, sirve de plataforma para que las pequeñas y medianas empresas vendan sus productos.



Figura 12: Segmentos de clientes [36]

2. Propuesta de valor

La propuesta de valor es el paso más importante a la hora de crear un modelo de negocio porque es la propuesta que presentas al cliente para cubrir sus necesidades y aprovechar los beneficios que tu servicio o producto ofrece. La propuesta debe ser diferenciadora respecto a las empresas competidoras. Se podría decir que es como el corazón del negocio y de ahí su colocación en el centro de la plantilla que se mostró anteriormente. [24].

Para proponer una buena propuesta de valor a los clientes podemos hacerlo en pasos para que sea más fácil crearla:

- El **primer paso** podría ser conocer al segmento de clientes al que queremos enfocar nuestra propuesta. Y esto es algo que se complementa con el primer módulo del modelo de negocio, por lo que, una vez que tenemos los

segmentos de clientes definidos, podremos investigar y recopilar datos sobre sus necesidades, problemas y características que presentan.

- El **segundo paso** que se puede plantear para establecer la propuesta de valor es tener claro cuál será la personalización que le ofreceremos al cliente para cumplir con las expectativas y resaltar los beneficios que obtendrán si adquieren nuestro producto o servicio. Se podrá plantear la siguiente pregunta: ¿Qué problemas resuelve al cliente nuestro servicio que no ofrece una empresa competidora?
- El **tercer paso** lo enfocaremos en obtener una comunicación efectiva hacia el cliente, es decir, nos aseguraremos de que la propuesta se transmita de manera clara y efectiva.
- Y como **último paso**, la propuesta de valor debe estar en constante evaluación, esto quiere decir, que se debe estar dispuesto a ajustarse y mejorar basándose en la opinión del cliente, o en cambios que se produzcan en el mercado o en la innovación.

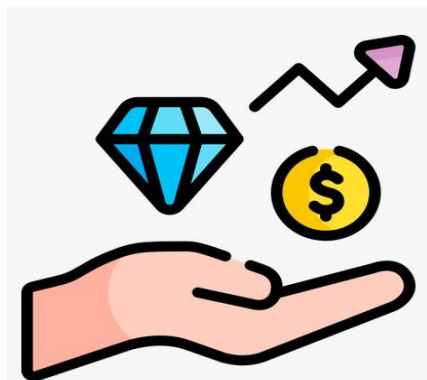


Figura 13: Propuesta de valor [37]

3. Canales

Los canales son los medios a través de los cuáles conectarás a tu empresa con los clientes incluyendo cualquier vía que utilice el cliente para acceder a tu servicio o producto. [38] Los canales presentan un papel fundamental porque en ellos se define la comunicación que quieres tener entre la empresa y cliente. Hay muchos tipos de canales, como pueden ser, tiendas, páginas web, redes sociales o incluso aplicaciones entre otras.

En esta sección nos podemos plantear una cuestión de forma general:

¿Qué canal o canales son los adecuados para mi propuesta de valor?

La respuesta tenemos que enfocarla al cliente para que nos ayude a seleccionar los canales en los que se reconozcan nuestros productos o servicios mediante marketing o publicidad. Una vez que se ha captado la atención del cliente tendremos que enfocar nuestra propuesta de valor de forma que puedan evaluarla, es decir, mostrarles el producto en uso para que puedan formar una opinión al respecto. Esto se puede conseguir, por ejemplo, mediante videos explicativos del producto o servicio, exposición de situaciones reales dónde se está utilizando... etc.

Para que el proceso de compra llegue satisfactoriamente se debe de disponer de un canal que facilite todo este trámite, ya sea mediante reuniones presenciales e intercambio de correos con los contratos y ofertas del producto. Una vez que se ha hecho la venta se debe definir el canal mediante el cual el producto llegará a cliente y cómo brindarás la atención y soporte después de la venta. [77]



Figura 14: Canales [38]

4. Relaciones con clientes

Las relaciones con los clientes nos ayudarán a establecer relaciones duraderas. Esto se debe centrar en cómo interactuarás con el cliente en cada etapa de la obtención del producto o servicio que proporcionas. Establecer una buena relación con los clientes que ya poseen el producto o servicio representa un papel muy importante porque puede conllevar a atraer a nuevos clientes gracias a la recomendación de nuestro producto, a recibir feedback constante para mejorar los servicios... [78]

Dependiendo de la propuesta de valor que tengas se deberá de elegir el tipo de relación que más se ajuste a tu servicio o producto. Existen varios tipos de relaciones:

- **Automatizadas:** este tipo de relación está basado, como su nombre indica, en sistemas automatizados con el que el cliente deberá de interactuar sin interacción humana. Un ejemplo sería un *chatbot*, que son chats en línea con respuestas predefinidas.
- **Co-creativas:** los clientes participan en el desarrollo de productos o servicios proporcionando ideas, ayudando a realizar pruebas... como, por ejemplo, la empresa LEGO dispone de una plataforma donde los clientes pueden presentar sus propios diseños para ser considerados para la producción.

-
- **Relaciones personales:** se establece una conexión directa con el cliente, es decir, se trabaja la relación con el cliente mediante llamadas telefónicas, correos personalizados, reuniones constantes... Este tipo de relaciones es el más utilizado para los negocios donde la confianza y la personalización del servicio o producto son muy importantes.



Figura 15: Relación con los clientes [34]

5. Fuente de ingresos

La fuente de ingresos es uno de los bloques fundamentales que se deben definir para entender cómo un negocio va a generar ingresos. Las fuentes de ingresos vendrán a través de nuestra propuesta de valor, que es el núcleo de lo que ofrecemos a nuestros clientes y que justifica el valor monetario que estarán dispuestos a pagar.

Existen varios tipos para definir la fuente de ingresos y entre los más destacados tenemos:

- **Ingresos por ventas de productos o servicios:** esta fuente de ingresos viene definida por la venta de productos directamente al cliente, es decir, no existen intermediarios. Esta fuente es uno de los más utilizados y abarca desde la venta en tiendas físicas, tiendas en línea, ferias... Una de las ventajas más significativas es que facilita una relación más cercana con los clientes.
- **Ingresos por uso:** son los ingresos generados cada vez que un cliente utiliza un servicio. Como, por ejemplo, servicio de alquiler de coches eléctricos.
- **Ingresos por alquiler:** este tipo de fuente de ingresos son generados por el alquiler de los activos físicos o digitales. Un ejemplo muy simple es el alquiler de oficinas.

-
- **Ingresos por intermediación:** estos ingresos son los obtenidos por actuar de intermediarios con otras empresas, como, por ejemplo, Airbnb.

Conociendo estos tipos de fuente de ingresos es importante tener en cuenta que debemos de comprender el valor que el servicio o producto ofrece a los clientes y que es lo que están dispuestos a pagar por él. Por esta razón es muy importante no tener una única fuente de ingresos porque puede impactar gravemente en el negocio, ya que las variaciones en el mercado o la demanda afecta directamente.



Figura 16: Fuente de ingresos

6. Recursos clave

Los recursos clave son todos aquellos elementos que una empresa necesita para crear, ofrecer y entregar la propuesta de valor a los clientes. Se podría decir que son los ingredientes necesarios para que el modelo de negocio planteado tenga éxito, ya que, si falta uno de ellos, sería imposible llevar a cabo el desarrollo del producto o servicio. Existen varios tipos de recursos clave: [61]

- **Físicos:** son todos los activos tangibles, es decir, los edificios, vehículos, maquinaria, instalaciones... Por ejemplo, un restaurante necesita un local para poder producir, por lo que el local se considera un recurso clave.
- **Intelectuales:** los recursos clave del tipo intelectual hacen referencia a las marcas registradas, patentes, derechos de autor... como, por ejemplo, Coca-Cola.
- **Humanos:** este tipo se refiere al personal que llevará a cabo el desarrollo del producto o servicio.
- **Financieros:** engloba los recursos relacionados con el ámbito financiero, es decir, todo lo relacionado con el dinero y la gestión financiera de la empresa. Estos recursos son fundamentales porque permiten financiar las operaciones diarias, las inversiones estratégicas y los proyectos de crecimiento e innovación.

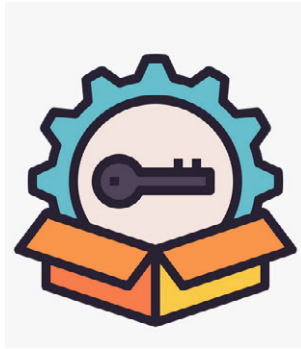


Figura 17: Recursos clave [39]

7. Actividades clave

Las actividades clave son aquellas acciones que una empresa debe realizar para cumplir con la propuesta de valor y operar de manera efectiva. Estas actividades no solo tienen por qué ser operativas, si no, que también pueden incluir aspectos estratégicos como pueden ser: diseñar campañas de marketing, plantear alianzas con proveedores, formación...

Existen diferentes tipos de actividades clave y la elección de éstas dependerá de la propuesta de valor que tengas definida. Entre las más destacables tenemos:

- La **producción** es la más típica y la que más empresas abarcan ya que incluye todas las operaciones en las que se fabrica o desarrolla un producto o servicio.
- También existen las actividades que solucionan los problemas específicos que el cliente tiene o proporcionan una mejora en la experiencia del usuario, son las llamadas **actividades de resolución de problemas**.
- Y, por último, las **actividades de plataforma** son aquellas que permiten la creación y mantenimiento de una plataforma que soporta el modelo de negocio.

Es importante añadir, que tener una actividad clave no te excluye de poder tener otras. Existen muchos negocios que tienen varias actividades clave entre sí y esto dependerá de la propuesta de valor que quieras proporcionar al cliente.



Figura 18: Actividades clave

8. Asociaciones clave

Las asociaciones clave permiten a las empresas acceder a recursos, conocimientos y capacidad que no poseen internamente, lo que ayuda a desarrollar el producto o servicio de formas más efectiva y entregarle valor. Plantear asociaciones puede ser parte del éxito de tu negocio, ya que, se pueden ofrecer soluciones más completas o innovadoras que los competidores, acceder a recursos que son difíciles de conseguir o incluso reducir los riesgos que puede provocar crear algo nuevo.

Existen varios tipos y las más comunes son:

- **Asociaciones con proveedores:** esta asociación se lleva a cabo cuando una empresa se asocia con proveedores que aseguran el suministro de materias primas, como puede ser, proveedores que distribuyen componentes electrónicos para empresas de fabricación de sensores, o fabricantes de automóviles necesita asociarse con proveedores de neumáticos...
- Las **alianzas estratégicas** son colaboraciones entre dos empresas que buscan beneficios mutuos. Por ejemplo, una empresa de software podría establecer una alianza con otra empresa con servicios en la nube para ofrecer soluciones integradas a sus clientes.
- **Cooperaciones con canales:** implica trabajar con intermediarios para llegar al mercado de manera más efectiva y directa. Un ejemplo sencillo podría ser Amazon, que permite a muchísimos vendedores de minoristas vender sus productos en su plataforma.

9. Estructura de costes

La estructura de costes es fundamental para entender cómo una empresa gestiona sus recursos para operar y obtener valor. Es importante mentalizarse de que este bloque del modelo Canvas no son solo números y presupuestos, si no de entender cómo cada gasto contribuye al funcionamiento y crecimiento del negocio. [62]

Existen dos tipos diferentes de estructura de costes: impulsadas por los costes e impulsada por el valor. La primera estructura mencionada se enfoca en minimizar los costes siempre que sea posible con mucha automatización. Por otro lado, la segunda estructura está más enfocada en la creación de valor y por la personalización del servicio. Para entender bien las diferencias, se ha creado una tabla con los puntos más importantes y un ejemplo:

ASPECTO	IMPULSADA POR COSTES	IMPULSADA POR VALOR
Enfoque principal	Minimización de costes	Creación de valor
Estructura de Costes	Simplificación y minimización	Flexibilidad y adaptabilidad, a veces con costos más altos
Automatización	Máxima automatización	Menor dependencia de automatización, mayor personalización
Servicio al cliente	Menos énfasis en el servicio personalizado	Alto grado de servicio personalizado
Ejemplos	Ryanair	Mercedes-Benz

Tabla 1: Diferencias estructura de costes

Con este último bloque terminaríamos de explicar los conceptos más importantes de todos los módulos que forman parte del Modelo Canvas. Para dar un poco de contexto, nuestro modelo de negocio se enfocará en la desarrollo e implementación de una Blockchain que irá incorporada en un sistema de monitorización. Por lo que, para poder introducir correctamente el concepto de Blockchain en nuestro modelo de negocio, se añadirá un nuevo bloque denominado ‘Gobernanza de la Blockchain’ donde se explicará en el apartado de ‘Solución del modelo de negocio’ qué es lo que nos aportará esta tecnología tan innovadora.

3. Especificaciones y restricciones de diseño

En este capítulo explicaremos las especificaciones y restricciones del diseño, es decir, de los sensores, cómo obtenemos las medidas y cómo son representadas, así como plantearemos las herramientas que se han utilizado para el desarrollo de la Blockchain y se mostrará un diagrama de red de la arquitectura.

3.1 Sensores

3.1.1 Especificaciones

Los sensores utilizados para la monitorización de las aulas han sido diseñados y montados exclusivamente por el profesorado de la ETSISI. El sistema completo está formado por un despliegue de redes inalámbricas de sensores que permitirá diferentes configuraciones, de forma que se puedan atender las necesidades específicas de cada tipo de aula, es decir, la configuración dependerá de la orientación del aula, de su tamaño, del número de ventanas...

No se podrá hacer una explicación exhaustiva de las especificaciones de los sensores, ya que no disponemos de toda la información, pero hemos añadiremos algunas imágenes, prototipos y sensores utilizados que nos ha proporcionado la ETSISI.

Para el diseño de los sensores han utilizado el sensor de temperatura, humedad y Co2 SCD3 diseñado por Sensirion. Las características que más podemos destacar de este sensor es la precisión que tienen cuando toman las medidas, en este caso son: [57]

- Para la **medición de la temperatura** tiene un rango de medición entre $-40\text{ }^{\circ}\text{C}$ a $70\text{ }^{\circ}\text{C}$ y una precisión de: $\pm (0.4^{\circ}\text{C} + 0.023 \times (T [^{\circ}\text{C}] - 25^{\circ}\text{C}))$.
Por ejemplo, supongamos que la temperatura medida por el sensor es de $30\text{ }^{\circ}\text{C}$, para calcular la temperatura real aplicaríamos la fórmula y obtendríamos lo siguiente:

$$\text{Precisión} = \pm(0.4^{\circ}\text{C} + 0.023^{\circ}\text{C} \times (30^{\circ}\text{C} - 25^{\circ}\text{C})) = \pm 0.515^{\circ}\text{C}$$

Por lo tanto, la temperatura real podría estar comprendida entre 29.48°C o 30.51°C .

- Para la **medición de Co2** el rango está comprendido entre los 0 ppm a $40,000\text{ ppm}$ y una precisión de $\pm 30\text{ ppm} + 3\%$ de la lectura. Esto quiere decir que si, por ejemplo, nuestro sensor mide una concentración de 1000 ppm , el cálculo de la precisión sería el siguiente:

$$\begin{aligned} \text{Precisión} &= \pm 30\text{ ppm} + 3\% \text{ de la lectura} \\ \text{Precisión} &= \pm 30\text{ ppm} + 0.03 \times 1000\text{ ppm} = \pm 60\text{ ppm} \end{aligned}$$

La precisión real estaría en el rango 940 ppm y 1060 ppm .

-
- Para la **medición de humedad** su rango es de 0% a 100% RH y la precisión es de $\pm 3\%$ RH para el rango 20% a 80% RH de la humedad leída por el sensor. Esto quiere decir, que, si tenemos una humedad relativa del 50% RH, obtendríamos:

$$\text{Precisión} = 50\%RH \pm 0.03 \times 50\%RH = 50\%RH \pm 1.5\%RH$$

Esto quiere decir que nuestra precisión real de la humedad estaría entre 48.5% RH y el 51.5% RH.

Para la interconexión de los sensores con la red inalámbrica han utilizado el módulo MRF24J40ME, diseñado por Microchip Technology. Este módulo dispone de librerías de software propias para facilitar el desarrollo de aplicaciones. De las características que podemos destacar:

- Compatible con el estándar IEEE 802.15.4¹.
- Rango de comunicación de hasta 400 metros.
- Interfaz *Serial Peripheral Interface* (SPI) para la comunicación con microcontroladores y otros dispositivos de control.

De esta misma empresa electrónica han hecho uso del microcontrolador dsPIC33E1512GM604. Este microcontrolador se utiliza para el procesamiento digital y control en tiempo real. De las características más destacables tenemos:

- Frecuencia de operación hasta 60 millones de instrucciones por segundo.
- Amplia gama de periféricos como pueden ser comparadores analógicos, conversores analógico-digital, entre otros.
- Módulos de comunicación con otros componentes.

En la siguiente imagen se puede ver el esquemático que realizaron para el diseño de los sensores. Los componentes VEML-7700, MKD-1365 no tienen relevancia para este proyecto y, por lo tanto, no se explicarán.

¹ Estándar que permite que dispositivos portátiles puedan comunicarse entre sí mediante redes inalámbricas de área personal con bajas tasas de transmisión de datos. [59]

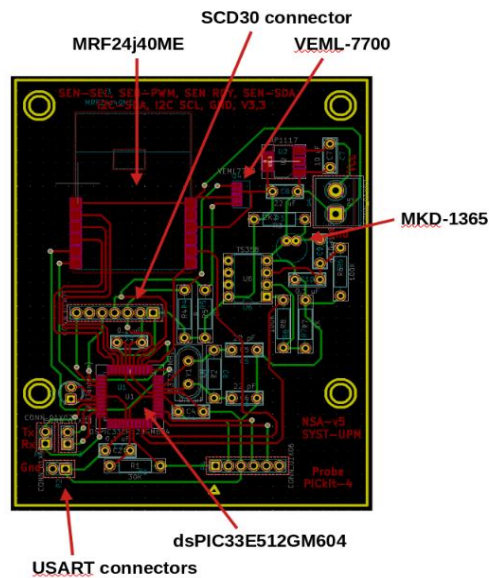


Figura 19: Esquemático del sensor

Y para concluir con este apartado, las siguientes imágenes nos mostrarán como quedaron finalmente los sensores una vez hicieron el montaje completo. La caja sobre la que van montados los sensores también lo realizó la ETSISI mediante impresión 3D.

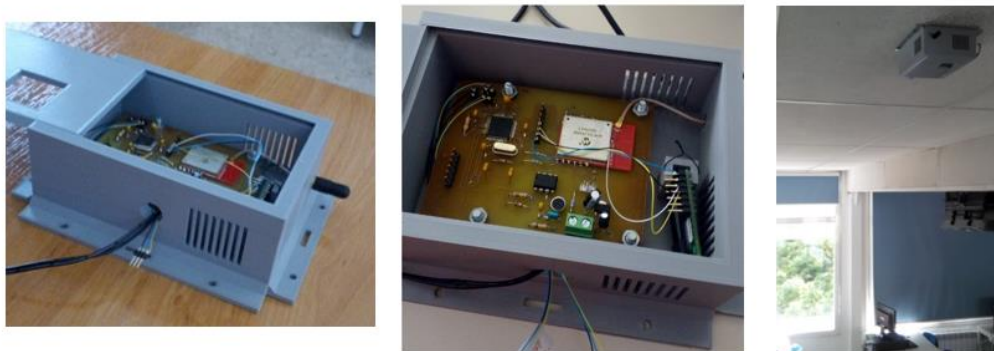


Figura 20: Sensores

3.1.2 Datos

Para la obtención de datos, los sensores de temperatura y humedad toman medidas cada 5 minutos, mientras que el sensor de dióxido de carbono captura cada 15 minutos. También se toman otras medidas como luminosidad, picos de ruido o temperatura en la superficie de la clase, pero para este proyecto no se tomarán en cuenta.

Los datos se obtienen mediante llamadas a la API del servidor. En nuestro caso, el servidor será simulado por un repositorio en GitHub donde se almacenarán los datos en formato Excel.

Este archivo contiene mediciones realizadas cada hora, lo que nos proporciona 12 mediciones por hora de temperatura y humedad y 4 de dióxido de carbono, según los intervalos de adquisición de datos de los sensores mencionados en el apartado anterior.

Cada medida está definida por `id_transferencia` que es un número único que nos permite identificar a cada medida. Las columnas de medición más importantes para este proyecto vienen definidas como:

- `data.room_temp`: que representa la temperatura de la clase.
- `data.co2`: que representa el dióxido de carbono.
- `data.humidity`: que representa la humedad.

También se guarda la fecha y hora de cada medida realizada en el formato ISO 8601, que consiste en una representación de `yyyy-MM-ddTHH:mm:ss.sss`, siendo:

- `yyyy`: corresponde al año.
- `MM`: mes.
- `dd`: día.
- `T`: separador entre fecha y hora.
- `HH`: hora.
- `mm`: minutos.
- `ss`: segundos.
- `sss`: milisegundos.

El horario de la ETSISI comprende desde las 8:00 hasta las 21:00, por lo que se generarán 13 documentos en formato csv con el formato mostrado en la Figura 11.

Hay que destacar que, dependiendo de la franja horaria, habrá medidas disponibles, por la ocupación que puedan tener las diversas clases.

A	B	C	D	E	F	G	H
<code>_id_transferencias</code>	<code>class</code>	<code>data.co2</code>	<code>data.humidity</code>	<code>data.room_temp</code>	<code>data.surf_temp</code>	<code>node</code>	<code>time</code>

Figura 21: Columnas del csv

Como se visualiza en la Figura 11, también existen otras columnas como:

- `class`: corresponde a la clase dónde se están realizando las medidas.
- `data_room_temp`: es la temperatura de la superficie de la clase, es decir, se mide la temperatura que tienen las paredes.
- `node`: corresponde al nodo que está tomando las medidas.

3.2 Herramientas utilizadas

3.2.1 Java

En nuestro proyecto, el lenguaje principal que se utilizará será Java. Todo el sistema de monitorización estará programado en este lenguaje, ya que se considera uno de los lenguajes más utilizados para crear software compatible con una gran diversidad entre sistemas operativos. [27]

Nuestro código estará estructurado en paquetes con sus correspondientes clases para que el proyecto sea fácil de gestionar y manejar.



Figura 22: Java [28]

3.2.2 Eclipse

Eclipse proporciona una serie de herramientas para la gestión de espacios de trabajo, depurador de código y una extensa colección de plug-ins y por lo que será el entorno de desarrollo integrado (IDE) que utilizamos para el desarrollo de nuestro sistema de monitorización.



Figura 23: Eclipse [29]

3.2.3 GitHub

Para nuestro proyecto, GitHub representa un papel fundamental ya que, simulará el servidor dónde se almacenan los datos de los sensores. Para la obtención de estos datos tendremos que usar la API (Application Programming Interfaces) REST (Representational State Transfer) para realizar peticiones a nuestro repositorio. Y tras estas peticiones seremos capaces de poder visualizar y leer los datos de los sensores.



Figura 24: GitHub [44]

3.2.4 HTTPie

Esta herramienta será la que utilizemos para realizar las pruebas de comunicación con el repositorio en GitHub generando solicitudes API. Está basada en inteligencia artificial y está diseñada para que funcione como un cliente HTTP de línea de comandos.

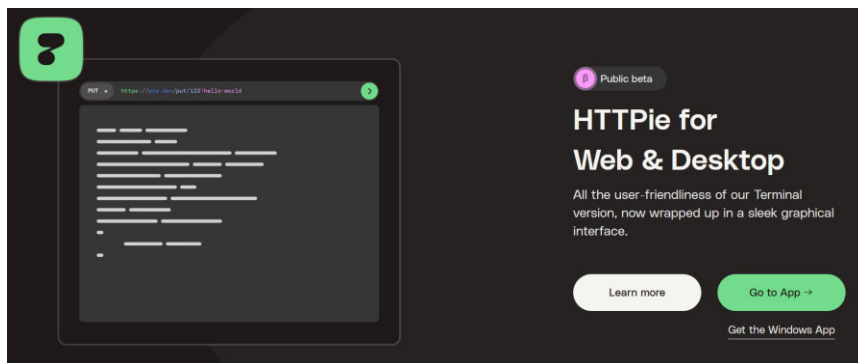


Figura 25: httpie [42]

3.2.5 Spring

Para este proyecto, Spring se utilizará para crear la aplicación del sistema de monitorización que nos permitirá poder visualizar las medidas tomadas, y la formación de la Blockchain.

Este framework es muy utilizado por los desarrolladores Java y es un plug-in que se puede instalar en Eclipse y por eso fue seleccionado para desarrollar todo el sistema de monitorización. Spring está configurado por defecto para ejecutarse en el *localhost*² en el puerto 8080.

² Localhost: Es el nombre comúnmente utilizado para designar el equipo que estamos utilizando. [51]

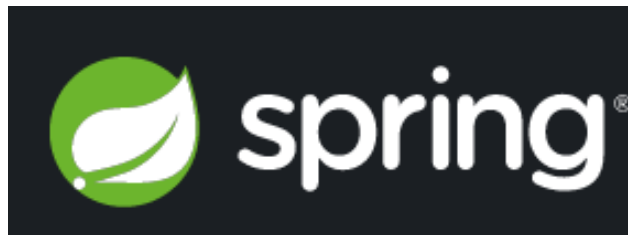


Figura 26: Spring framework [44]

3.3 Diagrama de red

El diagrama de red es la representación visual de una red de ordenadores o telecomunicaciones mostrando los componentes que lo forman. [26]

En nuestro proyecto, el diagrama de red constará de cuatro bloques fundamentales para el entendimiento de este.

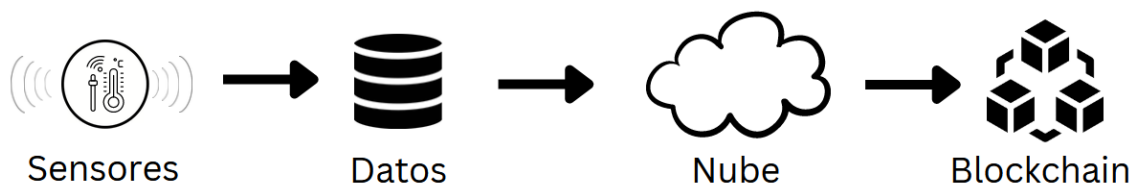


Figura 27: Diagrama de red

Según se muestra en la figura 8, el primer bloque lo formarán los sensores ubicados en las aulas de la ETSI de Sistemas Informáticos de la UPM.

Los datos de temperatura, humedad y dióxido de carbono obtenidos de dichos sensores serán enviados a la nube. Se almacenarán en un documento Excel y posteriormente se subirán a un servidor que, en nuestro caso, será simulado por GitHub.

Una vez que los datos están en GitHub, se realizarán llamadas a la API para poder leer los datos e introducirlos en la Blockchain de forma que se vayan generando bloques encadenados con las medidas obtenidas por los sensores.

4. Solución tecnológica

En este apartado se explicará la solución tecnológica propuesta final. Se mostrará todo el proceso de desarrollo de la Blockchain basada en las mediciones de los sensores.

4.1 Llamadas a la API de GitHub y pruebas

Para simular que nuestro repositorio en GitHub es el servidor y poder obtener los datos, es necesario configurarlo de forma que podamos obtener un token para acceder a él.

En GitHub los repositorios pueden ser públicos o privados. El repositorio para este proyecto es público, ya que, en este mismo, se almacenará todo el código de la Blockchain para que cualquier persona pueda visualizarlo.

Antes de poder hacer llamadas a la API, es necesario crear un token para tener acceso al repositorio. En GitHub existen dos tipos de tokens: Fine-grained y Classic. Las principales diferencias entre ellos están resumidas en la siguiente tabla: [58]

Características	Fine-grained	Classic
Acceso a recursos propiedad de un usuario u organización	Sí	No
Acceso a repositorios específicos	Sí	No
Permisos específicos	Sí	No
Fecha de expiración	Sí	Opcional
Requerimiento de aprobación por parte de dueños de organización	Sí	No

Tabla 2: Diferencia tokens

Para nuestro proyecto, se generará un token Fine-grained que se utilizará para las llamadas a la API. Este token sólo es válido para el repositorio que contendrá todo lo relacionado con este proyecto.

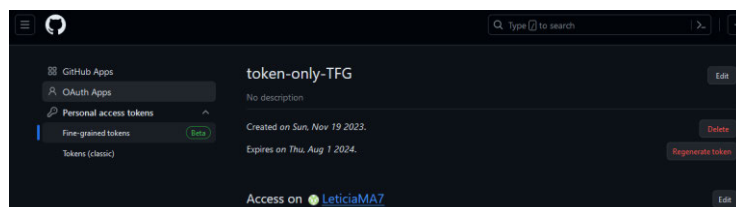


Figura 28: Generación del token

Antes de incorporar las llamadas directas a la API en el código Java, se llevaron a cabo pruebas simuladas para verificar la precisión de las URL y los encabezados requeridos según la información que proporciona GitHub. En todas las llamadas, esperamos un código 200 para

verificar que se han realizado correctamente. Es importante saber que los códigos nos ayudan a identificar como se ha procesado la solicitud, entre lo más comunes podemos encontrar: [52]

- **Código 200:** la llamada ha tenido éxito y se ha procesado correctamente.
- **Código 201:** es código muy parecido al código 200 pero significa que la solicitud se procesó y devolvió un recurso.
- **Código 400:** la llamada ha sido errónea, pero por fallos del cliente, es decir, es el servidor quién no puede procesar la solicitud por diversas razones, como, por ejemplo, una URL no válida o mala sintaxis.
- **Código 401:** la llamada que se ha realizado no está autorizada para acceder al servidor. Puede que las cabeceras de autenticación no sean las correctas, entre otras razones.
- **Código 404:** es uno de los errores más comunes y hace referencia a que la solicitud no es encontrada.

En nuestras pruebas, las llamadas han sido divididas en tres partes:

1. Visualización del repositorio

Se realiza una visualización previa de todos los repositorios a los que podemos acceder mediante el token. En nuestro caso solo podremos acceder a un repositorio y a través de éste, obtendremos el nombre del usuario y el nombre del repositorio. Estos datos son vitales para realizar las siguientes llamadas a la API de GitHub. [30]

GitHub nos proporciona mediante un ejemplo la siguiente información:



```
Request example

GET /user/repos

cURL  JavaScript  GitHub CLI

curl -L \
-H "Accept: application/vnd.github+json" \
-H "Authorization: Bearer <YOUR-TOKEN>" \
-H "X-GitHub-API-Version: 2022-11-28" \
https://api.github.com/user/repos
```

Figura 29: Ejemplo de petición GET

Por lo tanto, teniendo en cuenta esta información, construimos la llamada de esta forma:

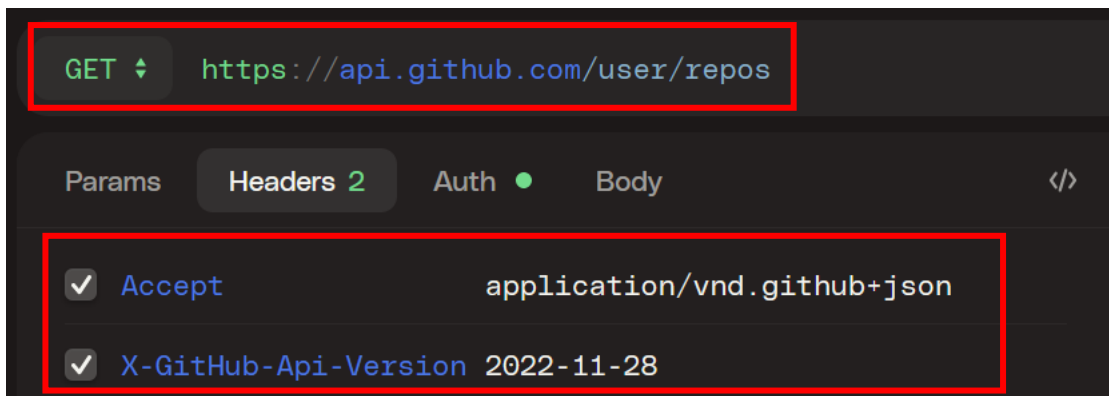


Figura 30: Petición GET - Cabeceras

En la pestaña de Auth, indicaremos que necesitamos una autenticación 'Bearer token' para poder introducir el token.

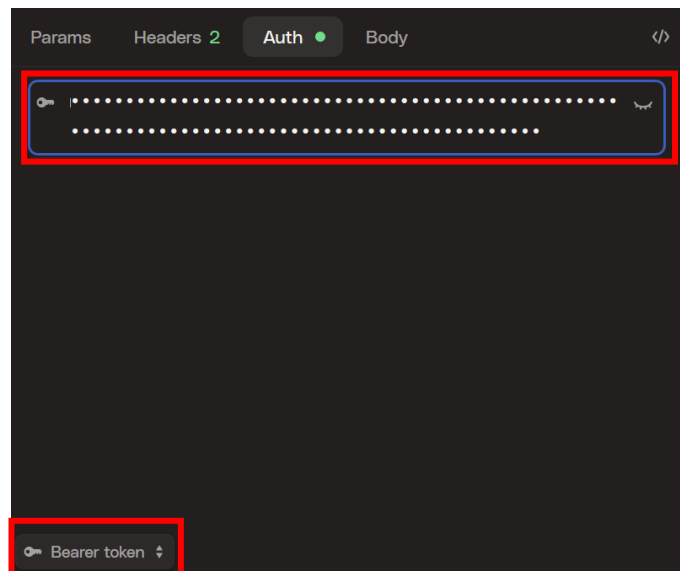


Figura 31: Petición GET - Autenticación

Y tras completar todos los parámetros requeridos, realizamos la petición y obtenemos una respuesta satisfactoria.

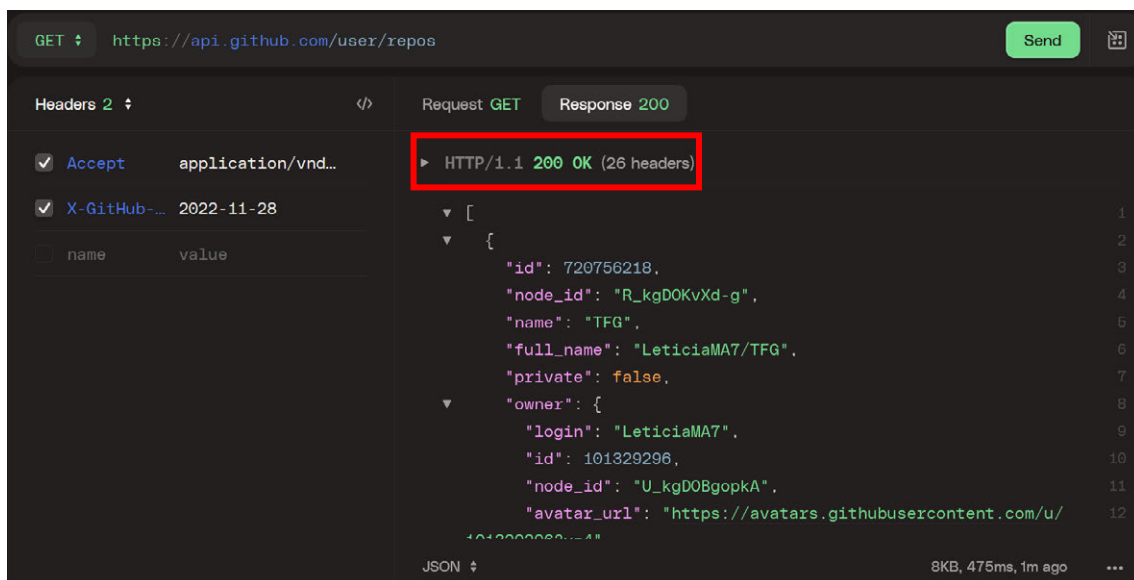


Figura 32: Respuesta de la petición GET

2. Visualización del contenido del repositorio

Una vez que hemos obtenido los repositorios a los que podemos acceder mediante el token, realizaremos otra petición GET para acceder al contenido de nuestro repositorio. Para ello, GitHub nos proporciona este ejemplo:



Figura 33: Ejemplo petición GET

Las cabeceras y la autenticación son idénticas a la petición anterior excepto por la URL. Por lo que, para esta nueva URL deberemos de proporcionarle el usuario y nombre del repositorio que hemos obtenido en la primera petición. De igual forma, hemos obtenido una respuesta correcta.



Figura 34: Prueba petición GET - Contenido del repositorio

3. Lectura de datos del repositorio

Para esta prueba, la lectura de datos se realizó sobre un documento csv. GitHub en su página web informa de que, para poder realizar la lectura de datos, previamente se necesita obtener un enlace de descarga para acceder a los datos. Por lo que, primero se realizará una llamada al fichero y de ahí obtendremos el enlace.

Seguidamente, se realizará la llamada a ese enlace para la lectura de datos.

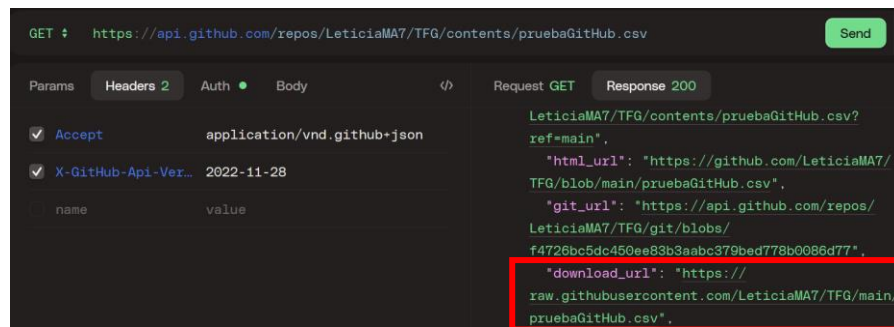


Figura 35: Petición GET - Obtención URL datos

Una vez que obtenemos la URL de descarga, se hace otra petición GET a esa URL para obtener los datos del fichero csv, pero esta vez, no son necesarias las cabeceras únicamente la autenticación.

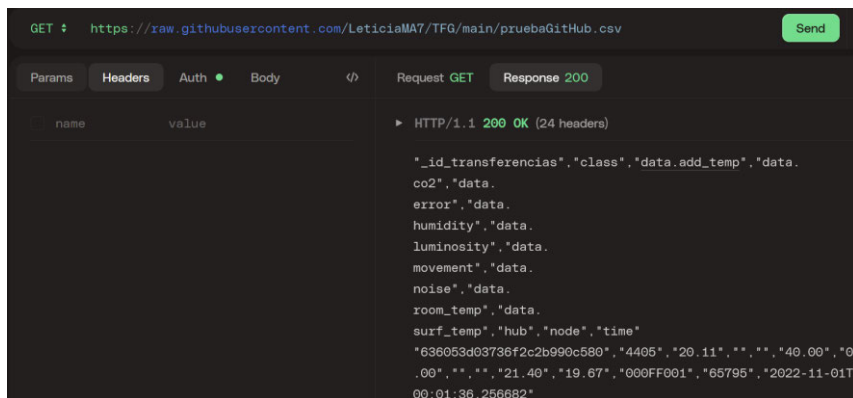


Figura 36: Petición GET - Visualización de los datos

4.2 Desarrollo del código

Para el desarrollo del código, ha sido dividido en cuatro partes para que sea más fácil su seguimiento. Por lo que, primero hemos creado un primer paquete que contendrá todo lo relacionado con las llamadas a la API de GitHub. El segundo paquete tendrá las clases relacionadas con el procesamiento de los datos, el tercer paquete será el encargado de contener las clases que crean las notificaciones y el último paquete lo formará la Blockchain.

- Paquete `tfg.monitorizacion.github`

Una vez realizadas las pruebas de las llamadas a la API de GitHub, se construyen en Java para replicarlas. Se han creado tres clases, una para cada llamada. La estructura del paquete es la siguiente:

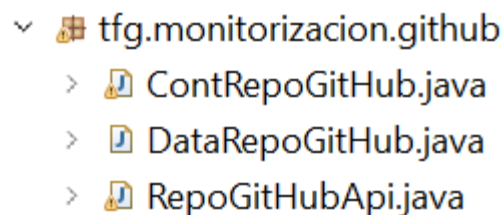


Figura 37: Paquete de GitHub

La clase `RepoGitHub` es la primera clase que necesitamos para empezar con el proceso de las llamadas al repositorio. En esta primera clase, lo más importante es poder obtener el nombre del repositorio y el nombre del usuario, ya que serán datos clave para realizar las siguientes.

Para ello, primero crearemos la solicitud con los parámetros necesarios que hemos visto en la documentación de GitHub.

```

// Crear un cliente HTTP
HttpClient httpClient = HttpClient.newHttpClient();
// Crear una solicitud HTTP GET con los encabezados necesarios
HttpRequest request = HttpRequest.newBuilder()
    .uri(URI.create(URL))
    .header("Accept", "application/vnd.github+json")
    .header("X-GitHub-API-Version", "2022-11-28")
    .header("Authorization", "Bearer " + TOKEN)
    .build();

```

Figura 38: Solicitud HTTP

Una vez, que creamos y enviamos la solicitud tendremos que verificar si el código que nos proporciona la llamada es un código 200, y si es así, obtendremos el nombre del repositorio y el nombre del usuario. Entre las líneas del código, se puede visualizar una función `'System.out.println'` que imprime por la consola de Eclipse comentarios que nos sirven como seguimiento.

```

try {
    // Enviar la solicitud y obtener la respuesta
    HttpResponse<String> response = httpClient.send(request, HttpResponse.BodyHandlers.ofString());

    // Verificar si el estado de la respuesta es 200 (OK)
    if (response.statusCode() == 200) {
        System.out.println("STATUS 200 OK - REPO");
        // Parsear la respuesta JSON
        JsonParser jsonParser = new JsonParser();
        JsonElement jsonElement = jsonParser.parse(response.body());

        // Verificar si la respuesta es un array JSON
        if (jsonElement.isJsonArray()) {
            JsonArray jsonArray = jsonElement.getAsJsonArray();
            // Iterar sobre cada elemento del array
            for (JsonElement e : jsonArray) {
                if (e.isJsonObject()) {
                    JsonObject repo = e.getAsJsonObject();
                    // Obtener el nombre del repositorio
                    this.repoName = repo.get("name").getString();
                    System.out.println("repoName: " + repoName);
                    JsonObject owner = repo.getAsJsonObject("owner");
                    // Obtener el nombre de usuario del propietario del repositorio
                    if (owner != null) {
                        this.user = owner.get("login").getString();
                        System.out.println("user: " + user);
                    }
                }
            }
        }
    }
} else {
    // Imprimir el estado de la respuesta si no es 200
    System.out.println("STATUS: " + response.statusCode());
}
} catch (Exception e) {
    // Manejar excepciones que puedan ocurrir durante la solicitud HTTP
    e.printStackTrace();
}

```

Figura 39: Obtención del repositorio y user

La segunda clase que necesita los datos obtenidos de esta primera solicitud es `ContRepoGitHub` para crear la solicitud que nos proporcionará los enlaces de los ficheros que están en el repositorio para poder acceder a los datos con la última solicitud.

```

try {
    // Enviar la solicitud y obtener la respuesta
    HttpResponse<String> response = httpClient.send(request, HttpResponse.BodyHandlers.ofString());

    // Verificar si el estado de la respuesta es 200 (OK)
    if (response.statusCode() == 200) {
        System.out.println("STATUS 200 OK - CONTENT");
        // Parsear la respuesta JSON
        JsonParser jsonParser = new JsonParser();
        JsonElement jsonElement = jsonParser.parse(response.body());

        // Verificar si la respuesta es un array JSON
        if (jsonElement.isJsonArray()) {
            JsonArray jsonArray = jsonElement.getAsJsonArray();
            // Iterar sobre cada elemento del array
            for (JsonElement e : jsonArray) {
                if (e.isJsonObject()) {
                    JsonObject repo = e.getAsJsonObject();
                    // Obtener el nombre del archivo y la URL de descarga
                    String fileName = repo.get("name").getString();
                    String downloadUrl = repo.get("download_url").getString();

                    // Si el archivo es un CSV, añadir la URL de descarga a la lista
                    if (fileName.endsWith(".csv")) {
                        System.out.println("downloadUrl: " + downloadUrl);
                        LinksFilesDownload.add(downloadUrl);
                    }
                }
            }
        }
    }
} else {
    // Imprimir un mensaje si el estado de la respuesta no es 200
    System.out.println("STATUS NO OK - CONTENT");
}
} catch (Exception e) {
    // Manejar excepciones que puedan ocurrir durante la solicitud HTTP
    e.printStackTrace();
}

```

Figura 40: Obtención URLs de descarga

Y, por último, en la clase DataRepoGitHub se realiza la llamada que nos proporcionará los datos de los ficheros leídos. En esta clase, se realiza una lectura de los nombres de los ficheros, se leen las columnas y se obtiene la media de los datos de temperatura, humedad y Co2 por cada fichero leído.

También, se instancia a otra clase llamada DataSensors, que se explicará en el siguiente apartado de procesamiento de datos.

- Paquete tfg.monitorizacion.data

El procesamiento de los datos es realizado por la clase DataSensors y por lo que, el paquete quedará de esta forma:

```

v tfg.monitorizacion.data
  > DataSensors.java

```

Figura 41: Paquete y clase de DataSensors

En esta clase, el método más importante es el que realiza la media de los datos que se obtiene de cada fichero leído.

```

public double calculateAndPrintMedian(List<String> data) {
    Collections.sort(data);
    int size = data.size();
    double median;

    if (size == 0) {
        System.out.println("No hay datos");
        return 0;
    }

    int nonEmptyCount = 0;
    double[] values = new double[size];

    // Llena el array 'values' solo con los valores no vacíos
    for (int i = 0; i < size; i++) {
        String stringValue = data.get(i);
        if (!stringValue.isEmpty()) {
            values[nonEmptyCount] = Double.parseDouble(stringValue);
            nonEmptyCount++;
        }
    }

    // Calcula la mediana usando solo los valores no vacíos
    int effectiveSize = nonEmptyCount;
    if (effectiveSize % 2 == 0) {
        // La lista tiene un número par de elementos
        double middle1 = values[effectiveSize / 2 - 1];
        double middle2 = values[effectiveSize / 2];
        median = (middle1 + middle2) / 2;
    } else {
        // La lista tiene un número impar de elementos
        median = values[effectiveSize / 2];
    }

    return median;
}

```

Figura 42: Método para la media

- Paquete tfg.monitorizacion.notifications

Este paquete está formado por tres clases que se relacionan entre sí:

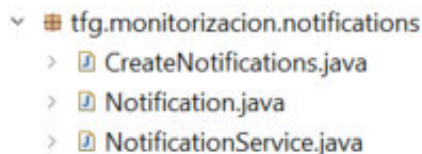


Figura 43: Paquete de las notificaciones

En la clase Notification.java se encuentran los parámetros que contendrá cada una de las notificaciones, que en nuestro caso serán:

- Parámetro que puede tener valor de: temperatura, humedad o Co2.
- Valor que obtendrá el valor de temperatura, humedad o Co2.
- Descripción donde se explicará brevemente cuál es la notificación.

```

public class Notification {
    private String parameter;
    private double value;
    private String description;
    // Parámetro asociado a la notificación.
    // Valor del parámetro al momento de la notificación.
    // Descripción detallada de la notificación.

    public Notification(String parameter, double value, String description) {
        this.parameter = parameter;
        this.value = value;
        this.description = description;
    }

    // Método para obtener el parámetro de la notificación.
    public String getParameter() {
        return parameter;
    }

    // Método para obtener el valor del parámetro al momento de la notificación.
    public double getValue() {
        return value;
    }

    // Método para obtener la descripción detallada de la notificación.
    public String getDescription() {
        return description;
    }
}

```

Figura 44: Clase Notification

En la siguiente clase llamada CreateNotification.java es donde se crea la notificación con los valores mencionados anteriormente. Para esta clase, se han introducido como constantes los valores que establece el RITE tanto para los meses de verano como los meses de invierno. Se ha considerado meses de verano de marzo a septiembre (periodo de primavera y verano) y el resto de los meses para invierno (periodo de otoño e invierno).

```

public class CreateNotifications {

    // Constantes para los rangos según el RITE
    private static final double TEMP_VERANO_MIN = 23.0;
    private static final double TEMP_VERANO_MAX = 25.0;
    private static final double HUM_VERANO_MIN = 40.0;
    private static final double HUM_VERANO_MAX = 60.0;

    private static final double TEMP_INVIERNO_MIN = 21.0;
    private static final double TEMP_INVIERNO_MAX = 23.0;
    private static final double HUM_INVIERNO_MIN = 40.0;
    private static final double HUM_INVIERNO_MAX = 50.0;

    private static final double CO2 = 800.0;
}

```

Figura 45: Constantes RITE

```

// Método para verificar y generar notificaciones
public static List<Notification> checkAndGenerateNotifications(double temperature, double humidity, double co2) {
    List<Notification> notifications = new ArrayList<>();
    notifications.clear();
    // Obtener el mes actual
    int currentMonth = java.time.LocalDate.now().getMonthValue();

    // Verificar los valores según la temporada (verano o invierno)
    if (isSummer(currentMonth)) {
        if (temperature < TEMP_VERANO_MIN || temperature > TEMP_VERANO_MAX) {
            String description = "La temperatura en la clase 4005 debería de encontrarse entre " + TEMP_VERANO_MIN + " y " + TEMP_VERANO_MAX + ".";
            Notification notification = new Notification("Temperatura", temperature, description);
            notifications.add(notification);
        }

        if (humidity < HUM_VERANO_MIN || humidity > HUM_VERANO_MAX) {
            String description = "La humedad en la clase 4005 debería de encontrarse entre " + HUM_VERANO_MIN + " y " + HUM_VERANO_MAX + ".";
            Notification notification = new Notification("Humedad", humidity, description);
            notifications.add(notification);
        }
    }
    else { // Es invierno
        if (temperature < TEMP_INVIERNO_MIN || temperature > TEMP_INVIERNO_MAX) {
            String description = "La temperatura en la clase 4005 debería de encontrarse entre " + TEMP_INVIERNO_MIN + " y " + TEMP_INVIERNO_MAX + ".";
            Notification notification = new Notification("Temperatura", temperature, description);
            notifications.add(notification);
        }

        if (humidity < HUM_INVIERNO_MIN || humidity > HUM_INVIERNO_MAX) {
            String description = "La humedad en la clase 4005 debería de encontrarse entre " + HUM_INVIERNO_MIN + " y " + HUM_INVIERNO_MAX + ".";
            Notification notification = new Notification("Humedad", humidity, description);
            notifications.add(notification);
        }
    }
}
}

```

Figura 46: Generación de las notificaciones

La última clase será la encargada de gestionar las notificaciones para luego poder generarlas y visualizarlas. Se trata de un servicio que actúa de intermediario entre la lectura de los datos y la generación de las notificaciones.

```
import org.springframework.stereotype.Service;

// Clase que gestiona las notificaciones en un servicio.
@Service
public class NotificationService {

    private List<Notification> notifications = new ArrayList<>();

    // Método para agregar una nueva notificación a la lista.
    public void addNotification(Notification notification) {
        notifications.add(notification);
    }

    // Método para obtener todas las notificaciones almacenadas.
    public List<Notification> getNotifications() {
        return notifications;
    }
}
```

Figura 47: Servicio de notificaciones

- Paquete **tfg.monitorizacion.blockchain**

La creación de la Blockchain está dividida en tres clases importantes:

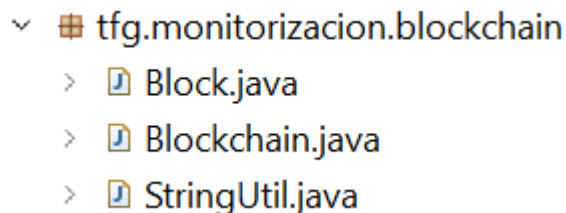


Figura 48: Paquete y clases de la Blockchain

En la clase StringUtils se desarrolla un método que aplica el algoritmo SHA-256, que es un algoritmo criptográfico que produce un hash con el objetivo de crear un fragmento de datos único con una longitud fija de 256 bits.

```

public static String applySha256(String input) {
    try {
        // Obtener una instancia de MessageDigest para SHA-256
        MessageDigest summary = MessageDigest.getInstance("SHA-256");

        // Convertir la cadena de entrada a un array de bytes usando UTF-8
        byte[] hash = summary.digest(input.getBytes("UTF-8"));

        // Crear un StringBuffer para construir la cadena hexadecimal del hash
        StringBuffer hexString = new StringBuffer();

        // Convertir cada byte del hash a una cadena hexadecimal
        for (int i = 0; i < hash.length; i++) {
            String hex = Integer.toHexString(0xff & hash[i]);
            // Asegurarse de que cada hexadecimal tenga dos caracteres
            if (hex.length() == 1) hexString.append('0');
            hexString.append(hex);
        }

        // Devolver la cadena hexadecimal completa
        return hexString.toString();
    } catch (Exception e) {
        // Lanzar una RuntimeException si ocurre algún error
        throw new RuntimeException(e);
    }
}

```

Figura 49: Método applySha256

Una vez hemos programado esta clase, se debe crear una clase que pueda realizar un bloque por cada fichero leído con las medidas obtenidas de los sensores. Para crear un bloque, se ha programado el constructor de la clase Block con los siguientes parámetros: medida de la temperatura, medida de la humedad, medida del Co2 y el hash del bloque anterior.

En esta clase, se crean dos métodos muy importantes, el primero es la creación de hash combinando varios datos como el previousHash pero que para la primera cadena tendrá un valor de cero, el timeStamp que corresponde a la fecha y hora en la que se crea la cadena, el nonce y los datos de los sensores.

```

/**
 * Calcula el hash del bloque combinando los valores de previousHash, timeStamp, nonce,
 * temperature, humidity y co2.
 *
 * @return El hash calculado como una cadena hexadecimal.
 */
public String calculateHash() {
    String calculatedHash = null;
    try {
        calculatedHash = StringUtil.applySha256(
            previousHash + timeStamp + Integer.toString(nonce) +
            Double.toString(temperature) + Double.toString(humidity) + Double.toString(co2)
        );
    } catch (Exception e) {
        e.printStackTrace();
    }
    return calculatedHash;
}

```

Figura 50: Método calculateHash

El segundo método importante es el que crea la dificultad de los bloques. Este método es configurable introduciéndole el número de dificultad que queramos. Es el encargado de realizar la prueba de trabajo.

```
/**
 * Realiza la prueba de trabajo (proof of work) ajustando el nonce hasta que el hash del bloque
 * comience con un número de ceros igual a la dificultad especificada.
 *
 * @param difficulty El nivel de dificultad (número de ceros iniciales requeridos en el hash).
 */
public void difficultyBlock(int difficulty) {
    String target = new String(new char[difficulty]).replace('\0', '0');
    while (!hash.substring(0, difficulty).equals(target)) {
        nonce++;
        hash = calculateHash();
    }
}
```

Figura 51: Método difficultyBlock

Y, por último, la clase Blockchain es la encargada de crear la cadena con todos los bloques creados anteriormente. Cabe destacar que, para nuestro proyecto, tendremos un total de 13 bloques, uno por cada fichero leído en su correspondiente hora. Y esos bloques, formarán la Blockchain completa.

Por defecto, se ha creado un Blockchain con un grado de dificultad de 4 pero este valor, como hemos visto anteriormente es configurable. También, se ha creado un primer bloque que consta como inicial, al que se le irán añadiendo el resto de los bloques. Éste es llamado: 'bloque génesis'.

```
// Lista para almacenar los bloques en la cadena
private List<Block> blockchain;

// Nivel de dificultad para la prueba de trabajo (proof of work)
private int difficulty;

// Constructor por defecto con nivel de dificultad predeterminado
public Blockchain() {
    this(4); // Nivel de dificultad predeterminado
}

// Constructor con parámetro de nivel de dificultad
public Blockchain(int difficulty) {

    // Inicializar la lista de bloques y el nivel de dificultad
    this.blockchain = new ArrayList<>();
    this.difficulty = difficulty;

    // Crear y agregar el bloque génesis
    Block genesisBlock = createGenesisBlock();
    genesisBlock.difficultyBlock(difficulty); // Aplicar la prueba de trabajo al bloque génesis
    blockchain.add(genesisBlock);
}

// Método privado para crear el bloque génesis
private Block createGenesisBlock() {
    return new Block(0, 0, 0, "0");
}

// Método para agregar un nuevo bloque a la cadena
public void addBlock(Block newBlock) {
    newBlock.difficultyBlock(difficulty); // Aplicar la prueba de trabajo al nuevo bloque
    blockchain.add(newBlock); // Agregar el nuevo bloque a la cadena
}

// Método getter para obtener la lista de bloques en la cadena
public List<Block> getBlockchain() {
    return blockchain;
}
```

Figura 52: Clase Blockchain

Adicionalmente, como método de seguridad se ha creado un método que comprueba los hashes y así poder verificar la cadena de bloques. Este método se llama: `isBlockchainValid` y su implementación es la siguiente:

```
// Método para verificar la validez de la cadena de bloques
public boolean isBlockchainValid() {
    Block currentBlock;
    Block previousBlock;

    // Iterar sobre cada bloque en la cadena (excepto el bloque génesis)
    for (int i = 1; i < blockchain.size(); i++) {
        currentBlock = blockchain.get(i);
        previousBlock = blockchain.get(i - 1);

        // Verificar que el hash del bloque actual sea válido
        if (!currentBlock.hash.equals(currentBlock.calculateHash())) {
            return false;
        }

        // Verificar que el hash anterior en el bloque actual coincida con el hash del bloque anterior
        if (!currentBlock.previousHash.equals(previousBlock.hash)) {
            return false;
        }
    }
    return true; // La cadena es válida si todas las comprobaciones pasan
}
```

Figura 53: Método verificación de cadena

Una vez vista esta última clase, solo queda la implementación de Spring en todas estas clases.

4.3 Implementación de Spring

Como se ha mencionado anteriormente durante este proyecto, se ha hecho uso de Spring, donde gracias a este framework³, podremos ser capaces de visualizar los datos de forma gráfica y así mismo, obtener una forma representada de la Blockchain formada.

Para ello, se añadirán 3 clases Java a este proyecto, así como, varias plantillas HTML, algo de código JavaScript y estilos en css. Es importante destacar que no nos adentraremos en explicaciones detalladas sobre el código de las plantillas, el JavaScript empleado ni de los estilos aplicados.

Las nuevas clases, se han organizado en un paquete:





- ▼  tfg.monitorizacion
 - >  ControllerBasic.java
 - >  MonitorizacionApplication.java
 - >  SecurityConfig.java

Figura 54: Paquete monitorización

³ Entorno para desarrollar software

La clase MonitorizacionApplication será la que se ejecute y cree la aplicación.

```
@SpringBootApplication
public class MonitorizacionApplication {

    public static void main(String[] args) {
        SpringApplication.run(MonitorizacionApplication.class, args);
    }

}
```

Figura 55: Clase principal

En la clase SecurityConfig se configura la página de inicio al sistema de monitorización en el cual se deberá de ingresar el usuario y contraseña para acceder a la página principal y una vez hayamos accedidos, también tendremos que indicar cuál es la siguiente página a la que debe redirigirnos. En nuestro caso, será la página principal de monitorización.

```
public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
    http
        .authorizeRequests(authorizeRequests ->
            authorizeRequests
                // Permitir acceso sin autenticación a las rutas específicas
                .requestMatchers("/login", "/resources/**", "/static/**", "/css/**", "/js/**").permitAll()
                // Requerir autenticación para cualquier otra solicitud
                .anyRequest().authenticated()
            )
        .formLogin(formLogin ->
            formLogin
                // Configurar la página de inicio de sesión personalizada
                .loginPage("/login")
                // Configurar la URL de éxito después del inicio de sesión
                .defaultSuccessUrl("/monitorizacion", true)
                .permitAll()
            )
        .logout(logout ->
            logout
                // Configurar la URL de cierre de sesión
                .logoutUrl("/logout")
                // Configurar la URL de éxito después del cierre de sesión
                .logoutSuccessUrl("/login?logout")
                .permitAll()
            )
        // Deshabilitar CSRF para simplificar el desarrollo
        .csrf(csrf -> csrf.disable());

    // Construir el SecurityFilterChain
    return http.build();
}
```

Figura 56: Página de login

El usuario y contraseña tienen que estar definidas en el fichero llamado applications.properties dentro de nuestro proyecto de Spring. En este fichero, se pueden definir todos los valores que tomemos como constantes, en este caso, ahí también se definirá el token con el que accedemos al repositorio de GitHub.

La clase más importante de este paquete es la llamada ControllerBasic. Aquí se definirán todas las rutas a las que accederemos mientras navegamos en la aplicación. A través de una de estas

rutas, accederemos a los datos para poder mostrarlos. En nuestro caso, será la ruta definida como: /repo-content-and-median-data.

```
@GetMapping("/repo-content-and-median-data")
@ResponseBody
public Map<String, Object> getCombinedRepoContentAndMedianData() {
    // Paso 1: Obtener el repositorio y contenido del repositorio
    repoGitHubApi.getRequestRepo(token);
    String user = repoGitHubApi.getUser();
    String repoName = repoGitHubApi.getRepoName();
    contRepoGitHub.getContentRepo(user, repoName, token);
    List<String> urlsDownload = contRepoGitHub.getUrlsDownload();
    dataRepoGitHub.getData(token, urlsDownload);

    // Paso 2: Obtener datos procesados por archivo
    Map<String, DataSensors> fileDataMap = dataRepoGitHub.getFileDataMap();

    // Paso 3: Crear y añadir bloques a la Blockchain
    for (Map.Entry<String, DataSensors> entry : fileDataMap.entrySet()) {

        DataSensors dataSensors = entry.getValue();
        double medianTemperature = dataSensors.getMedianTemperature();
        double medianHumidity = dataSensors.getMedianHumidity();
        double medianCo2 = dataSensors.getMedianCo2();

        // Generar notificaciones y añadirlas al servicio
        List<Notification> notifications = CreateNotifications.checkAndGenerateNotifications(medianTemperature, medianHumidity, medianCo2);
        notifications.forEach(notificationService::addNotification);

        // Obtener el hash previo
        String previousHash = blockchain.getBlockchain().get(blockchain.getBlockchain().size() - 1).hash;
        // Crear un nuevo bloque y añadirlo a la blockchain
        Block newBlock = new Block(medianTemperature, medianHumidity, medianCo2, previousHash);
        blockchain.addBlock(newBlock);
    }

    //Paso 4: Preparar la respuesta combinada
    Map<String, Object> combinedResponse = new HashMap<>();
    combinedResponse.put("RepoName", repoGitHubApi.getRepoName());
    combinedResponse.put("User", repoGitHubApi.getUser());
    combinedResponse.put("UrlsDownload", urlsDownload);
    combinedResponse.put("FileData", fileDataMap);
    combinedResponse.put("Blockchain", blockchain.getBlockchain());

    return combinedResponse;
}
```

Figura 57: Ruta para acceder a los datos

Es importante destacar que, para llevar a cabo la realización de esta aplicación, ha sido necesaria la implementación de las siguientes dependencias en el fichero pom.xml de nuestro proyecto.

```

<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <scope>runtime</scope>
    <optional>true</optional>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
  </dependency>
  <dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpclient</artifactId>
    <version>4.5.13</version>
  </dependency>
  <dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.8.9</version>
  </dependency>
  <dependency>
    <groupId>com.opencsv</groupId>
    <artifactId>opencsv</artifactId>
    <version>5.9</version>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-core</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-config</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-web</artifactId>
  </dependency>
</dependencies>

```

Figura 58: Fichero pom.xml

Como se ha mencionado anteriormente también se ha hecho uso de plantillas HTML, JavaScript incluido dentro de las plantillas HTML y estilos css. La organización de estos ficheros queda organizada de la siguiente manera en nuestro proyecto:

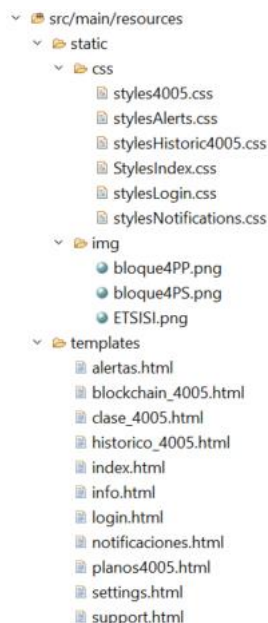


Figura 59: Organización de los recursos estáticos

Y con esto, terminaría este capítulo donde se ha mostrado la solución tecnológica propuesta.

5. Solución modelo de negocio

En esta sección desarrollaremos en profundidad nuestro modelo de negocio planteado para el sistema de monitorización basado en sensores. Se hará uso de la metodología *Business Model Canvas* que anteriormente hemos explicado y que ahora aplicaremos a nuestro modelo. También se mencionó la existencia de una plantilla que facilitaba mostrar el resumen de nuestro modelo. Nuestra modelo de negocio resumido en la plantilla es la siguiente:

MODELO CANVAS

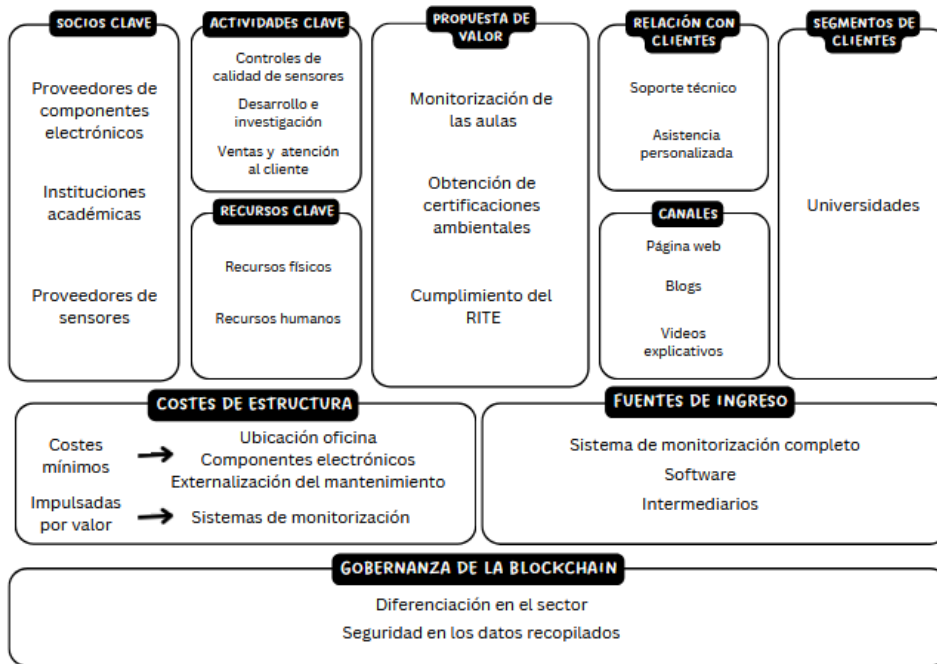


Figura 60: Modelo de negocio

5.1 Segmentos de clientes

Para definir el modelo de negocio de este proyecto se planteará un segmento en el que los clientes tengan instalaciones transitadas con frecuencia y por una cantidad considerable de personas. En otras palabras, nos referimos a espacios que requieran unas condiciones ambientales específicas para la habitabilidad.

Según las necesidades que pretendamos resolver en el cliente tendremos unos segmentos de clientes u otros. Basándonos en nuestro sistema de monitorización que medirá las condiciones ambientales de las aulas podremos enfocarlo en varios sectores según el tipo de mejora o beneficio que esperan obtener de nuestro servicio.

El segmento principal de clientes se enfoca en instituciones educativas, tanto del ámbito público como privado que estén dispuestos a transformar su campus universitario tradicional a un Smart Campus.

Para poder determinar estos segmentos de clientes podremos hacernos algunas preguntas, como, por ejemplo: **¿Cuáles son los problemas relacionados con la temperatura, la humedad y el CO2 a los que se enfrentan los campus universitarios en sus aulas? ¿Qué beneficios esperan obtener al implementar el sistema de monitorización?** A la respuesta de estas preguntas hace obtengamos dos subsegmentos de clientes muy relacionados entre sí.

El primer subsegmento de clientes estará enfocado a las instituciones educativas que buscan mejorar la calidad del ambiente de sus aulas para que no suponga un riesgo para la salud del alumnado y profesorado, así como obtener un mejor rendimiento educativo.

Por otro lado, el segundo subsegmento de clientes se enfocará a los campus universitarios que quieran adoptar medidas de sostenibilidad para cumplir con normativas y estándares ambientales.

Es importante añadir que la relación entre estos dos subsegmentos está en el cumplimiento de las normativas ambientales que existan. La qué más podemos destacar y que deberá ser común para ambos subsegmentos de clientes es el Reglamento Instalaciones Térmicas en los Edificios (RITE) que garantiza parte de la solución a ambos.

5.2 Propuesta de valor

Para iniciar a desarrollar la propuesta de valor, habrá que analizar los problemas que se intentan resolver para los clientes y cuáles no los abordarán las empresas competidoras. Así, cuando entendamos cuáles son los aspectos que mejorar, se intentará hallar una solución que genere valor a nuestros clientes. Para comenzar a plantear nuestra propuesta de valor podremos responder a unas preguntas ya enfocadas en nuestros segmentos de clientes.

Para el primer subsegmento: **¿Cómo afecta la calidad ambiental del aula al rendimiento académico? ¿Se compromete la salud del alumnado y del profesorado?**

Por otro lado, para el segundo subsegmento podríamos plantearnos la siguiente cuestión: **¿Cómo puede contribuir el sistema de monitorización a obtener certificaciones ambientales?**

Una vez que hemos evaluados los problemas y tenemos respuesta a las preguntas planteadas podremos empezar a crear la propuesta de valor para cada subsegmento, pero con una solución común.

La propuesta de valor común será:

- **Aplicación del RITE**

Este reglamento establece que en los meses de verano la temperatura debe de ser de entre 23°C y 25°C y una humedad relativa entre el 40% y 60%. En cambio,

durante el invierno, se sugiere mantener una temperatura entre 21°C y 23°C, junto con una humedad relativa entre el 40% y el 50%. Para los niveles de Co2 no se debe de superar el umbral de 800-1000 ppm (partes por millón) y si éste es superado, se deberá ventilar el espacio hasta reducir la concentración teniendo cómo referencia que la concentración en el exterior es de 420 ppm. [56]

Para poder cumplimentar con el RITE es necesaria la monitorización en las aulas mediante los sensores en tiempo real. Gracias a éstos podremos detectar cuándo se sobrepasan los umbrales máximos permitidos y así poder actuar de inmediato a través de las notificaciones que el sistema de monitorización proporciona.

La propuesta de valor para el primer subsegmento debe de resolver los problemas ambientales que estén comprometiendo la salud de los estudiantes y profesorado. Existen numerosos estudios que afirman que unas malas condiciones en las aulas afectan negativamente en la salud pudiendo provocar como insolación o asma. Además, estas condiciones también influyen en el desarrollo del aprendizaje influyendo directamente en el estado anímico del estudiante. Por lo tanto, los sensores instalados en las aulas nos ayudarán a solventar este problema. [33]

Para concluir, explicaremos la propuesta planteada para el segundo subsegmento en el que deberemos de poder ayudar a nuestro cliente a obtener las certificaciones ambientales mediante el sistema de monitorización. Para ello, nuestra propuesta de valor estará enfocada en proporcionar datos precisos y en tiempo real de temperatura, humedad y Co2 para la generación de informes detallados y así facilitar la obtención de certificaciones reconocidas, como pueden ser:

- **Certificación LEED** (Leadership in Energy and Environmental Design) que fomenta prácticas de construcción en edificios que mejoran la eficiencia energética, el uso del agua, la calidad del aire y la sostenibilidad. Los edificios que consigan esta certificación se les conoce como edificios verdes.[54]
- **Certificación BREEAM** (Building Research Establishment Environmental Assessment Method): es el método de evaluación y certificación de la sostenibilidad en la edificación técnicamente más avanzado y líder a nivel mundial. Esta certificación evalúa impactos en 10 categorías como, por ejemplo: innovación, energía, salud y bienestar entre otras. [55]

Para ambos subsegmentos se aplicará la seguridad e inmutabilidad a los datos monitorizados por los sensores y esto se puede conseguir gracias a la tecnología Blockchain que se aplicará.

5.3 Canales

En nuestro caso, se establecerá un canal directo. Se basará en la plataforma web donde se expondrán los servicios que la empresa puede aportar. Esta página web dispondrá de un video explicativo del servicio que se vende para que aporte claridad al cliente. Para poder establecer un acercamiento entre ambas empresas estará disponible un formulario de contacto para que las universidades puedan solicitar más información. Además de la plataforma web se planteará la opción de publicar artículos relacionados con los Smart Campus, la innovación, la sostenibilidad y cómo la monitorización de las aulas mediante sensores ayuda a la calidad estudiantil.

Un canal clave para tener acceso a más universidades que quieran iniciar esta transformación hacia los Smart Campus será establecer alianzas estratégicas con empresas tecnológicas que ya trabajen con universidades y así poder atraer a más clientes.

Durante el proceso de venta, se ofrecerán testimonios y demostraciones en vivo visitando otras instituciones educativas que ya dispongan de nuestro sistema de monitorización, y una vez que el proceso esté en venta final se enviará la propuesta comercial adaptada a las necesidades de cliente. Cuando la venta esté cerrada, se procederá a la instalación y configuración del sistema intentando impactar mínimamente en su día a día.

5.4 Relaciones con los clientes

La relación con nuestros clientes universitarios es fundamental para nuestro modelo de negocio. Queremos asegurarnos de que cada campus universitario se sienta acompañado en cada etapa con el sistema de monitorización. Para ello, se le ofrecerá a la universidad una atención personalizada donde se le plantearán reuniones para entender cuáles son sus problemas, así como visitas a las instalaciones para comprender de primera mano las necesidades.

Se implementará un sistema de soporte técnico en tiempo real, dónde se podrán resolver las incidencias que puedan surgir en el sistema de monitorización. Una de las principales prioridades de cara al cliente es asegurar la satisfacción sobre nuestro sistema, pero también que la universidad vea mejoras significativas en las condiciones ambientales en sus aulas.

Para mantener unas buenas relaciones con los clientes, se le brindarán revisiones periódicas para evaluar el desempeño y obtener feedback para futuras mejoras en el sistema de monitorización.

5.5 Fuente de ingresos

La fuente de ingresos principal de nuestro modelo de negocio vendrá dada por la venta de nuestro sistema de monitorización. Es importante añadir que un modelo de negocio que solo tiene una fuente de ingresos puede hacer que sea inestable y poco beneficioso, por lo que partiendo del sistema de monitorización se plantean posibles fuentes de ingresos:

- Se ofrecerá a las instituciones la completa instalación del sistema de monitorización, es decir, se venderá el software y los sensores. Al ser un producto totalmente creado desde cero por nosotros, resultará más fácil ayudar al cliente en su gestión.
- Se le posibilitará al cliente la opción de obtener el software sin la necesidad de instalar los sensores. Esto es algo beneficioso para instituciones que, por ejemplo, tengan instalados sensores ambientales o de ocupación. Esta opción también nos abre las puertas a otras empresas, ya que les ofrecemos la posibilidad de adaptar el software a sus sensores.
- Y como última opción de ingresos será de hacer de intermediario para que la institución pueda acreditar las certificaciones ambientales, encargándonos de todo el proceso hasta la acreditación. Podremos ofrecer al cliente y a la empresa contratante para la acreditación la seguridad y la integridad de los datos, así como su trazabilidad.

5.6 Recursos clave

Para hacer realidad nuestro sistema de monitorización, necesitamos varios recursos clave y todos son igual de importantes para llevar a cabo este negocio. En primer lugar, es fundamental tener un equipo sólido de desarrolladores y soporte técnico. Estos profesionales son esenciales para diseñar, mantener y mejorar la plataforma de monitoreo. También son quienes resolverán cualquier problema que los clientes puedan enfrentar. Sin su conocimiento y experiencia, no podríamos tener un sistema funcional y eficiente.

Por otro lado, como recursos físicos necesitaremos sensores cualificados para medir las condiciones ambientales en las aulas. Aunque se intentará comprar los sensores ya montados, también necesitaremos componentes electrónicos adicionales.

Otro recurso físico clave es contar con una infraestructura de servidores y almacenamiento de datos para garantizar la seguridad y trazabilidad de los datos obtenidos de los sensores para su posterior procesamiento y análisis.

5.7 Actividades clave

Para las actividades clave se tendrá en cuenta el desarrollo e investigación del propio servicio que ofrecemos. Se desarrollarán las necesidades que nuestro software o los sensores no dispongan y, por lo tanto, será necesario poder ser capaces de ofrecer nuevas funcionalidades y adaptarnos a la innovación que se requiera.

Por otro lado, cada cierto tiempo se realizarán controles de calidad en los sensores para verificar que cumple correctamente con los parámetros de su ficha técnica, como por ejemplo voltajes, calibración, margen de error, entre otros.

Cabe destacar también, que dentro de nuestras actividades clave importantes se encuentra las ventas y la atención al cliente. Dentro de este sector podremos establecer relaciones más estrechas con el cliente ofreciéndoles servicio de solución de incidencias. Así mismo, este sector desarrollará las ventas utilizando técnicas como *upselling*⁴ y así como su gestión.

5.8 Asociaciones clave

La principal asociación clave que tendremos en nuestro modelo de negocio será la asociación con distribuidores del ámbito electrónico, ya que nos proporcionarán los sensores que se utilizarán para la monitorización de las aulas, así como componentes de repuesto para resolver cualquier incidencia que ocurra con algún sensor. Se intentará buscar una asociación que nos proporcione una buena calidad-precio. Nuestras asociaciones no se limitarán a un solo distribuidor, si no que, se tendrán varias opciones para adaptarnos a los presupuestos de las universidades. Se pretende asociarnos con empresas electrónicas que sean conocidas mundialmente por la calidad de sus sensores y componentes, como, por ejemplo, Bosch Sensortec, Herter Instruments, Bitmakers, Microchip Technology o Sensirion entre otras.

Otra asociación clave serán las instituciones académicas que requieran de nuestros servicios, ya que gracias a ellos obtendremos la fuente de ingresos. Para poder llevar a cabo esta asociación, se intentará establecer vínculos con intercambios de llamadas, correos y reuniones presenciales para ofrecer una cercanía con el futuro cliente.

⁴ Venta adicional

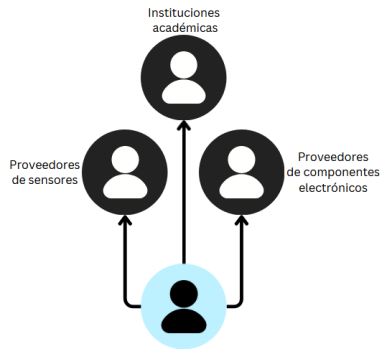


Figura 61: Asociaciones clave

5.9 Estructura de costes

Según la explicación que se ha proporcionado sobre la estructura de costes en el apartado del modelo de Canvas, hemos visto que hay dos tipos de estructuras con poca similitud entre sí; una es impulsada por costes mínimos y la otra impulsada por valor, pero, aunque presenten grandes diferencias no son excluyentes y en nuestro modelo de negocio convivirán las dos.

Se basarán en una estructura de mínimos costes aquellas cosas que no comprometen la calidad del servicio. Para nuestro modelo de negocio basado en el sistema de monitorización esta estructura implicaría:

- Se elegirían ubicaciones compartidas para las oficinas. Actualmente existen muchos edificios que alquilan sus oficinas por plantas.
- Para la selección de componentes electrónicos o sensores, se intentará que las asociaciones elegidas en el ámbito electrónico tengan precios más ajustados para los socios que colaboran con ellos. Por ejemplo, si existe una asociación con Bitmakers cuando se realizan compras al por mayor, se espera que el precio por unidad sea más asequible.
- Se puede considerar la externalización del mantenimiento de los sensores derivándolos directamente a los proveedores mediante acuerdos de soporte.

Como ya se ha comentado, ambas estructuras estarán presentes en nuestro modelo de negocio, y la estructura de costes impulsadas por valor estará enfocado en ofrecer un servicio de alta calidad y personalizado para cada uno de nuestros clientes. Por lo que, para garantizar un sistema de monitorización excelente se invertirá en tecnología innovadora, en actualizaciones o mejoras según la necesidad del cliente y aunque esto suponga un mayor costo aportará un valor significativo.

5.10 Gobernanza de la Blockchain

Añadir un nuevo bloque en la plantilla del modelo Canvas puede suponer un gran riesgo, ya que, si no se explica con determinación, no se entenderá por qué se ha introducido, pero cuándo se trata de una tecnología como la Blockchain podemos encontrar numerosos beneficios y ventajas y marcar un antes y después en el modelo de negocio.

Como se ha explicado anteriormente la tecnología Blockchain se encuentra en auge y nos ofrece una serie de características que la hacen innovadora, pero ahora la centraremos en las novedades que nos puede ofrecer a nuestro modelo negocio:

- **Garantizar la trazabilidad y seguridad:** la Blockchain nos proporciona un registro inmutable y transparente de todas las transacciones realizadas, por lo que, aplicada a nuestro modelo de negocio aseguraríamos que los datos obtenidos no pueden verse manipulados para conseguir acreditaciones ambientales.
- La aplicación de Blockchain también podría proporcionarnos **abrirnos al mercado** en otros ámbitos. Uno de esos ámbitos podría ser, aplicando Smart Contracts⁵ para automatizar nuevos procesos en el sistema de monitorización como podría ser, obtener certificaciones o la creación de tokens. Por lo que, supondría nuevas fuentes de ingresos.

Al incorporar la tecnología Blockchain, no solo hacemos que nuestros servicios sean más confiables, sino que también aseguramos la protección y la precisión de los datos que manejamos para las universidades. Esto nos ayuda a destacar en el campo de las soluciones tecnológicas avanzadas, y nos prepara para las necesidades futuras del sector educativo en entornos más inteligentes y seguros añadiendo valor a nuestra propuesta.

6. Resultados

En esta sección se mostrarán capturas de cómo ha quedado nuestro sistema de monitorización que permitirá a los campus universitarios visualizar las mediciones recogidas por los sensores y resolver los problemas relacionados con las condiciones ambientales.

1. Página de acceso

⁵ Contratos inteligentes: se ejecutan automáticamente a medida que las personas o empresas involucradas en un acuerdo van cumpliendo con las cláusulas de este. [63]

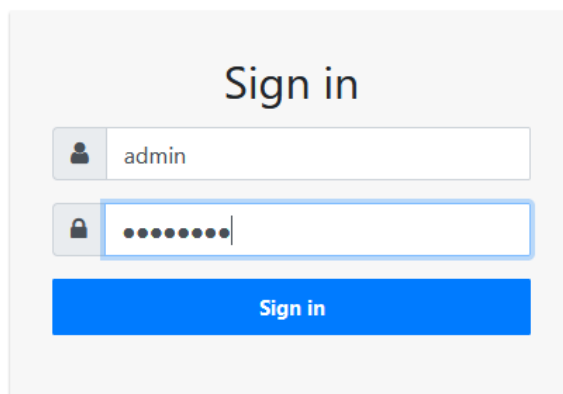


Figura 62: Acceso

2. Página de inicio de la aplicación

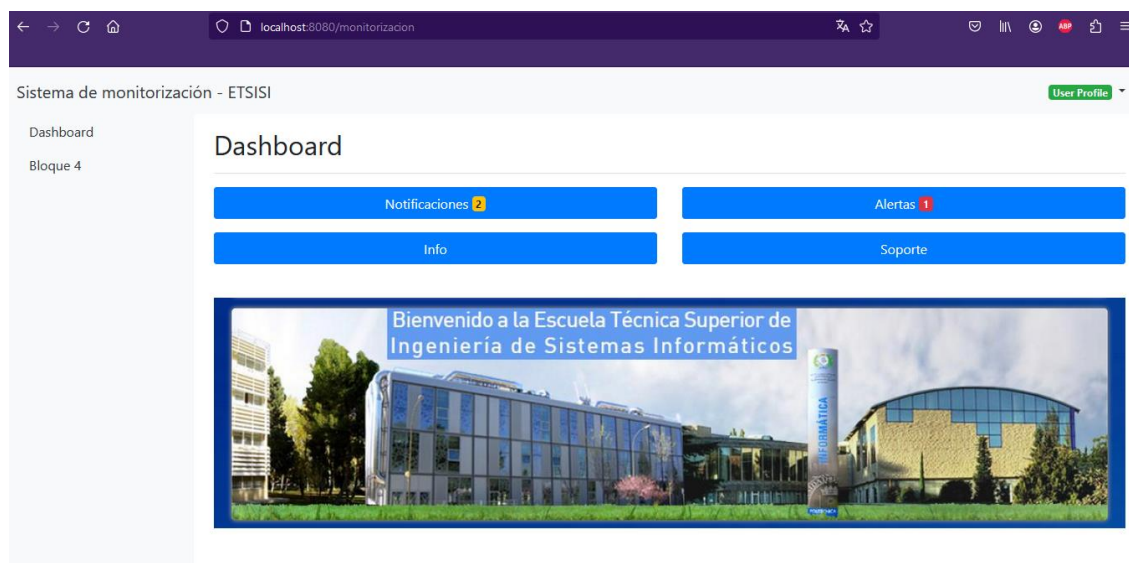


Figura 63: Inicio de la aplicación

3. **En el menú lateral**, si hacemos clic en 'Bloque 4' se abrirán todas las clases monitorizadas que hay disponibles. En este caso, solo tenemos la clase 4005. Y podremos acceder a 3 opciones: visualización de los datos en tiempo real, un histórico, la forma representada de la Blockchain y los planos de la ubicación del aula en la ETSISI.

3.1 Visualización de los datos gráficos

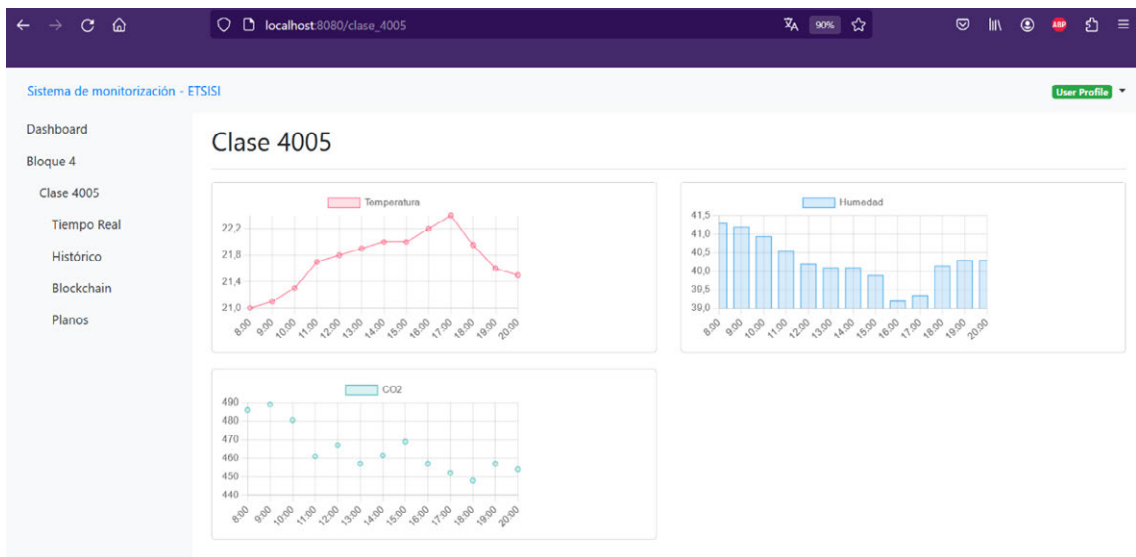


Figura 64: Gráficos de las medidas

3.2 Representación de la Blockchain

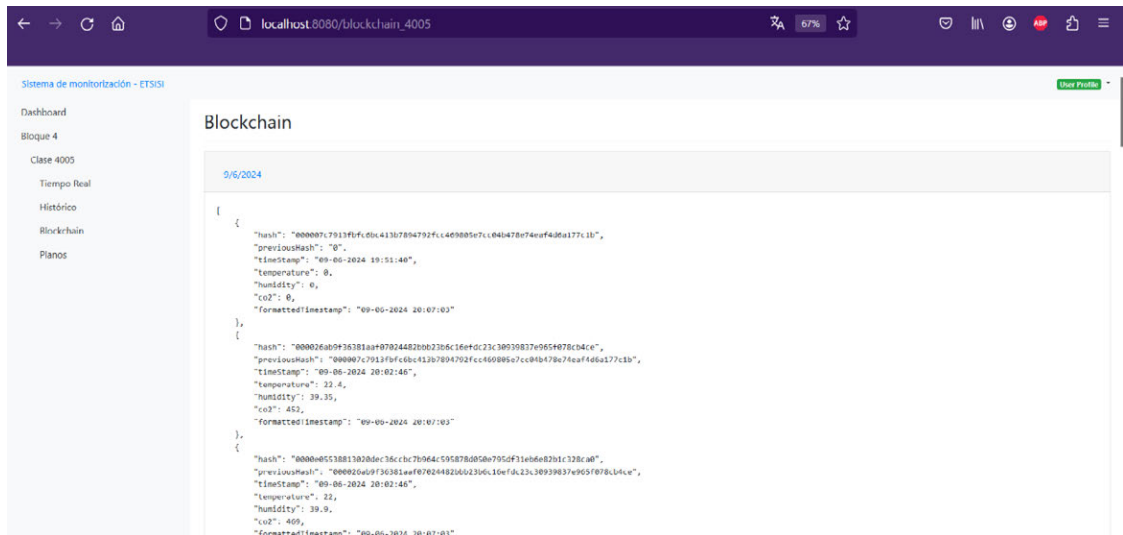


Figura 65: Representación de la Blockchain

3.3 Notificaciones

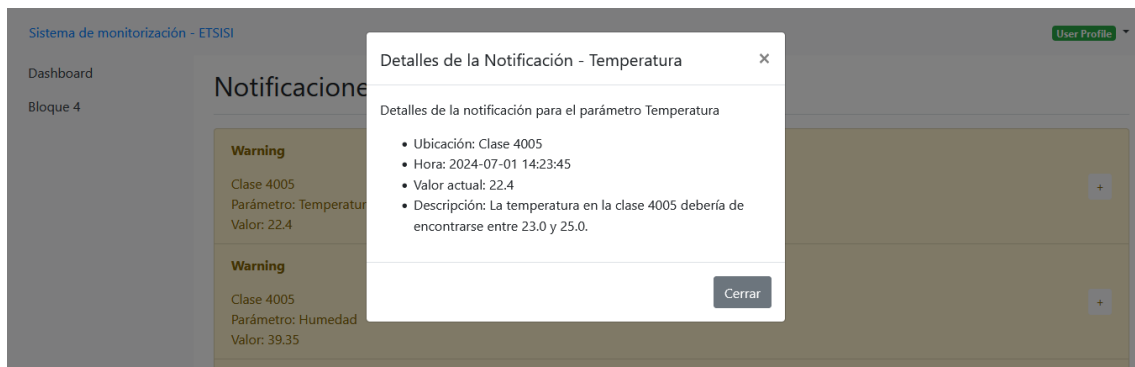


Figura 66: Notificaciones

Estas son las funcionalidades más importantes de esta aplicación, pero también están definidas otras como, por ejemplo: las alertas o como el salir de la aplicación o incluso la visualización de los planos del aula.

7. Presupuesto

En este apartado trataremos de realizar un estudio más exhaustivo sobre las horas requeridas para realizar este trabajo. Con este análisis lo que se quiere reflejar son las partes en las que se debe hacer un mayor énfasis a la hora del desarrollo.

Este proyecto lo podemos dividir en tres fases: la primera consta del entendimiento de un modelo de negocio, así como la creación del nuestro, la segunda fase se centra en la comprensión de lo que es una Blockchain y sus posibles aplicaciones y en la tercera fase, se diseña y desarrolla la Blockchain con los datos de los sensores.

La primera fase, es la que más tiempo de comprensión ha llevado, ya que, la creación de un modelo de negocio partiendo de cero puede ser algo difícil de analizar y plantear. La última parte que se menciona, también se lleva una gran inversión importante de tiempo debido a que se debe profundizar en el desarrollo de aplicaciones descentralizadas en Blockchain y cómo codificarlas en Java.

Para el cálculo de horas empleadas para llevar a cabo este proyecto dividiéndolo en las fases mencionadas con su correspondiente tiempo:

- Análisis de modelos de negocio: aproximadamente 10 días, dedicando 2,5 horas.
- Creación del modelo de negocio: unos 26 días, dedicando 3 horas.
- Estudio de Blockchain y sus posible aplicaciones: aproximadamente 12 días, con 2,5 horas.
- Creación de la Blockchain y procesamiento de datos: 40 días, con 2,5 horas al día.
- Realización de la memoria: 30 días, dedicando 2 horas.

Para esta parte de diseño de la Blockchain se ha tenido en cuenta el gasto de recursos materiales que consta el portátil personal y las horas empleadas. Se ha determinado que el precio por hora es de 8€ y el total de horas empleadas en este proyecto es de 293.

Como se ha comentado este trabajo forma parte de otro proyecto experimental en el que se ha hecho uso de los sensores de las aulas de la ETSISI y que, aunque no haya sido un desarrollo nuestro, se hace uso de los sensores y por lo que han sido también incluidos al presupuesto.

- Presupuesto enfocado en el diseño de la Blockchain.

	Horas	Precio/hora	Total
Mano de obra	293	10,00 €	2.930,00 €

	Coste	Vida útil (en años)	Meses de uso	Total
Recursos materiales	1000€	6	7	97,30 €

	Total
Presupuesto total	3.027,30 €

- Presupuesto de los sensores.

Recursos materiales	Unidades	Precio	Total
Sensor SCD30	1	40€	40€
MRF24J40ME	1	19€	19€
Componentes electrónicos	-	40€	40€
dsPIC33E1512GM604	1	9€	9€

Presupuesto total	Total
	108€

- Presupuesto total

	Total
Blockchain	3.027,30 €
Sensores	108€
	Total
Presupuesto total	3.135,30 €

8. Impacto del proyecto

En este capítulo se describirá las implicaciones de salud, ambientales, tecnológicas y económicas que nuestro proyecto aborda.

8.1 Salud

Nuestro modelo de negocio tiene un impacto positivo en cuanto a la salud, ya que está enfocado en el cumplimiento de las condiciones ambientales que establece el RITE. Esto nos asegura que la salud de las personas que estén en la institución no está en peligro. Unas malas condiciones ambientales en las aulas pueden poner en riesgo a alumnos y profesores, alterando sus constantes vitales. Por ejemplo, un alto nivel de Co2 puede afectar negativamente a una persona asmática.

Un punto negativo que podría afectar en la salud sería un fallo en el sistema de monitorización ya que no se estarían tomando mediciones en las aulas y podría suponer un riesgo para las personas.

8.2 Ambiental

El cumplimiento de los parámetros que establece el RITE con las condiciones ambientales nos aseguran un cumplimiento para el uso racional de la energía. Esto es algo muy importante porque gracias a los sensores se puede establecer medidas para la sostenibilidad.

Así mismo, es importante destacar que un sistema de monitorización basado en IoT para instituciones muy grandes conlleva a un aumento de residuos electrónicos y, por lo tanto, un mayor consumo energético. Sin embargo, esto se puede disminuir si se reciclan los residuos electrónicos que se generan cuando la vida útil de los sensores termina para crear nuevos.

Así que, por un lado, tenemos que la aplicación de IoT ayuda a mantener un control mayor sobre el uso de la energía, pero para ello también es necesario gestionar el ciclo de vida de los componentes electrónicos para reducir el impacto ambiental negativo.

8.3 Social y tecnológico

La incorporación de sensores en espacios afecta tecnológicamente y socialmente de forma muy chocante. El IoT altera las formas de trabajo más comunes y que rompe con lo establecido, pero abriendo nuevas oportunidades.

El impacto social que supone este modelo de negocio socialmente es facilitar y mejorar la calidad del trabajo gracias a la monitorización. En nuestro modelo de negocio, la aplicación proporciona un sistema de notificaciones y alertas y que, a través de este, la persona responsable de los sensores en las instalaciones pueda ser más efectivo a la hora de resolver incidencias. Por lo que podríamos decir, que permite una gestión más efectiva.

El impacto tecnológico puede tener puntos de vista opuestos. De forma negativa, el uso de tecnología IoT requiere un mayor conocimiento sobre la herramienta que se está utilizando y por lo que se necesitaría un personal más preparado tecnológicamente y requiere un mantenimiento más exhaustivo. El punto positivo, es que esto se puede ver compensado por las grandes posibilidades que innovación que ofrece. Por ejemplo, en nuestro caso, gracias a la tecnología Blockchain se mejora en la inmutabilidad de los datos.

En conclusión, la introducción de sensores y tecnología Blockchain en campus universitarios, abre la puertas a la transformación a los llamados Smart Campus y esto representa un impacto positivo que nos lleva a una mejora de la calidad estudiantil.

8.4 Económico

Iniciar la transformación de los campus tradicionales hacia los Smart Campus y la incorporación de un sistema de monitorización supone un impacto económico alto para la instituciones académicas. Esta inversión de dinero inicial es significativa pero cuando se inicia un proceso muy innovador, los beneficios se tienen que plantear a largo plazo.

Cuando planteamos que los beneficios se tienen que ver a largo plazo es porque la institución académica notará una mejoría notable en la eficiencia energética gracias a los sensores que monitorizarán en tiempo real. Se podría decir que se trata de una inversión estratégica, pero a la vez, fundamental para el futuro sostenible.

9. Conclusiones

9.1 Conclusiones

Tras la realización de este proyecto de fin de carrera, hemos podido observar cómo la tecnología Blockchain puede aplicarse a casi todos los ámbitos. Es una tecnología que se encuentra en auge y que, gracias a ella, podemos obtener una trazabilidad y una seguridad en los datos.

Partiendo de nuestro proyecto, la innovación de aplicar Blockchain en un sistema de monitorización basado en sensores de temperatura, humedad y Co2 le da valor tecnológico iniciando una transformación de los campus tradicionales hacia los Smart Campus. La creación de este sistema de monitorización basado en tecnología Blockchain y aplicado en un modelo de negocio nos ha dado una serie de beneficios. Entre los más importantes:

- **Seguridad en los datos:** la aplicación de la Blockchain en los datos que obtenemos de los sensores nos proporciona inmutabilidad y seguridad contra manipulaciones, que como ya hemos explicado anteriormente es una de las características de la tecnología Blockchain y que es casi imposible realizar alguna modificación sin que se note en la cadena.
- **Cumplimiento normativo:** gracias a la monitorización de las aulas se podrá asegurar que las instalaciones de las universidades cumplan con las normativas como por ejemplo el RITE, así como, certificaciones ambientales adicionales como LEED.
- **Calidad ambiental en las aulas:** nos aseguramos de que las aulas tengan siempre una buena calidad ambiental manteniendo unos buenos niveles de temperatura, humedad y Co2 creando entornos más saludables para los estudiantes y profesores. Esto también hace que se vea un mejor desempeño en las actividades educativas.

Como conclusión para este proyecto, podemos decir que se ha cumplido con creces el objetivo principal que se puso de manifiesto al principio de esta memoria, que era la creación de la Blockchain con los datos de los sensores enfocado en un modelo de negocio dando como fruto el sistema de monitorización creado, así como solventar cualquier problema en las condiciones ambientales de los aulas.

9.1 Trabajos futuros

Como ya se ha comentado anteriormente este proyecto se ha desarrollado partiendo de sensores que ya se encontraban instalados en la ETSISI y a través del sistema de

monitorización basado en Blockchain que hemos proporcionado se pueden crear varias líneas para trabajos futuros.

Las dos ideas más destacables y que pueden ser continuación de este proyecto son:

- Creación de Smarts Contracts para crear certificaciones digitales para la acreditación directa del Reglamento de Instalaciones Térmicas, así como cualquier normativa de sostenibilidad.
- Creación de tokens personalizados para poder utilizar dentro de las instituciones. Estos tokens serían como una moneda interna para gestionar transacciones dentro del campus, como pagos en cafeterías, en fotocopias... A través de la Blockchain se crearía un registro con todas las transacciones.
- Incorporación de nuevos sensores u otros componentes electrónicos para ampliar la monitorización. Como, por ejemplo, sensores de ocupación, luminosidad, ruido entre otros.

Es importante tener en cuenta que la tecnología Blockchain es bastante completa y requiere un conocimiento más profundo. La implementación de estas líneas futuras podría llevar a un desarrollo más avanzado y detallado.

10. Referencias

- [1] P. Tasca, *La revolución blockchain*. Barcelona: Deusto, 2018. [Online]. Available: https://static0planetadelibroscommx.cdnstatics.com/libros_contenido_extra/35/34781_La_revolucion_blockchain.pdf. [Accessed: Jul. 5, 2024]
- [2] A. Oteng-Pepurah, N. Agyei-Owusu, and M. S. B. Zaidi, "An efficient blockchain-based Internet of Things network for low-power IoT devices", *IET Smart Cities*, vol. 2, no. 1, pp. 1-10, Mar. 2020. [Online]. Available: <https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-smc.2019.0072>. [Accessed: Mar. 12, 2024].
- [3] J. Chen, "Proof of Work (PoW)", *Investopedia*, 2022. [Online]. Available: <https://www.investopedia.com/terms/p/proof-work.asp>. [Accessed: Jul. 5, 2024]
- [4] Capital.com, "Proof of Work (PoW) Definition", 2023. [Online]. Available: <https://capital.com/proof-of-work-pow-definition>. [Accessed: Jul. 5, 2024]
- [5] Y. Yuan, Y. Wu, y F.-Y. Wang, "Towards blockchain-based intelligent transportation systems", *Information Fusion*, vol. 54, pp. 134-146, ene. 2020. [En línea]. Disponible: <https://www.sciencedirect.com/science/article/pii/S2210670720302183?via%3Dihub>. [Accedido: jul. 5, 2024]
- [6] M. S. Hossain, S. N. Khan, F. Alharbi, R. M. Rahman, y A. G. Chowdhury, "Digital Twin-Based Smart Building Management System: Formulation to Implementation", *Buildings*, vol. 13, no. 4, p. 891, 2023. [En línea]. Disponible: <https://www.mdpi.com/2075-5309/13/4/891>. [Accedido: jul. 5, 2024]
- [7] J. Galache, F. J. Ortiz, J. R. Vidal, and J. Santa, "Blockchain para la transparencia en la gestión de la ayuda humanitaria", *Cuadernos de Cooperación*, no. 29, pp. 187-207, 2019. [Online]. Available: <http://hdl.handle.net/10045/109217>. [Accessed: Jul. 5, 2024]
- [8] Deloitte, "Internet of Things (IoT)", Deloitte, 2023. [En línea]. Disponible: <https://www2.deloitte.com/es/es/pages/technology/articles/iot-internet-of-things.html>. [Accedido: jul. 5, 2024]
- [9] Libelium, "Crear campus universitarios con tecnología IoT", Libelium, 2021. [En línea]. Disponible: <https://www.libelium.com/wp-content/uploads/2021/04/Crear-campus-universitarios-con-tecnologia-iot.pdf>. [Accedido: jul. 5, 2024]
- [10] IndiaMART, "Smart Campus Management System", IndiaMART, [En línea]. Disponible: <https://m.indiamart.com/proddetail/smart-campus-management-system-19269320688.html>. [Accedido: jul. 5, 2024]
- [11] UNIR, "Business Model Canvas: la guía definitiva", UNIR, [En línea]. Disponible: <https://www.unir.net/empresa/revista/business-model-canvas/>. [Accedido: jul. 5, 2024]

Referencias

- [12] Martín, "Canvas: Segmentos de Clientes", 2023. [En línea]. Disponible: <https://martin.click/marketing-online/canvas-segmentos-de-clientes/>. [Accedido: jul. 5, 2024]
- [13] HubSpot, "Cómo crear una propuesta de valor: guía paso a paso", HubSpot, [En línea]. Disponible: <https://blog.hubspot.es/marketing/crear-propuesta-de-valor>. [Accedido: jul. 5, 2024]
- [14] IEJPVG Modelo Canvas, "Canales", IEJPVG Modelo Canvas, [En línea]. Disponible: <http://iejpvgmodelocanvas.blogspot.com/p/canales.html>. [Accedido: jul. 5, 2024]
- [15] HubSpot, "Cómo crear un flujo de ingresos para tu negocio: guía completa", HubSpot, [En línea]. Disponible: <https://blog.hubspot.es/sales/flujo-ingresos>. [Accedido: jul. 5, 2024]
- [16] J. D. Fernández, "Fuente de ingresos Canvas", José David Fernández, [En línea]. Disponible: <https://josedavidfernandez.com/fuente-ingresos-canvas/>. [Accedido: jul. 5, 2024]
- [17] Modelo Canvas, "Relación con los clientes", Modelo Canvas, [En línea]. Disponible: <https://modelocanvas.net/relacion-con-los-clientes/>. [Accedido: jul. 5, 2024]
- [18] CA Consultores, "Actividades Clave (Paso Nº 7 Canvas)", Consultoría Consciente, [En línea]. Disponible: <https://www.consultoriaconsciente.net/blog/actividades-clave-paso-n-7-canvas-ca-consultores-consultoria-consciente>. [Accedido: jul. 5, 2024]
- [19] M. Ruiz Barroeta, "¿Qué son las Asociaciones Clave en el Modelo Canvas?", Milagros Ruiz Barroeta, [En línea]. Disponible: <https://milagrosruizbarroeta.com/que-son-las-asociaciones-clave-en-el-modelo-canvas/>. [Accedido: jul. 5, 2024]
- [20] Emprender Fácil, "Estructura de Costes de la Empresa Canvas", Emprender Fácil, [En línea]. Disponible: <https://www.emprender-facil.com/modelo-canvas/el-modelo-de-negocio-canvas/estructura-de-costes-de-la-empresa-canvas/>. [Accedido: jul. 5, 2024]
- [21] HubSpot, "Cómo estructurar los costos de tu negocio: guía completa", HubSpot, [En línea]. Disponible: <https://blog.hubspot.es/sales/estructura-costos>. [Accedido: jul. 5, 2024]
- [22] J. D. Fernández, "Estructura de Costes en el Modelo Canvas", José David Fernández, [En línea]. Disponible: <https://josedavidfernandez.com/estructura-costes-modelo-canvas/>. [Accedido: jul. 5, 2024]
- [23] Ministerio de Industria, Comercio y Turismo, "Estadísticas y publicaciones sobre PYME", Ministerio de Industria, Comercio y Turismo, [En línea]. Disponible: <https://industria.gob.es/es-es/estadisticas/paginas/estadisticas-y-publicaciones-sobre-pyme.aspx>. [Accedido: Jul. 5, 2024]
- [24] Cristina Maqueda Liébanas, "La importancia de la propuesta de valor", Universidad Pontificia Comillas, [En línea]. Disponible:

<https://repositorio.comillas.edu/xmlui/bitstream/handle/11531/4546/TFG001318.pdf?sequence=1&isAllowed=y>. [Accedido: Jul. 5, 2024]

[25] FIWARE, "La eficiencia energética: un problema", FIWARE, [En línea]. Disponible: <https://fiware.space/la-eficiencia-energetica-un-problema/>. [Accedido: Jul. 5, 2024]

[26] Lucidchart, "¿Qué es un diagrama de red?", Lucidchart, [En línea]. Disponible: <https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-red>. [Accedido: Jul. 5, 2024]

[27] HubSpot, "¿Qué es Java?", HubSpot, [En línea]. Disponible: <https://blog.hubspot.es/website/que-es-java>. [Accedido: Jul. 5, 2024]

[28] "Java Logo", 1000 Logos, [En línea]. Disponible en: <https://1000logos.net/java-logo/>. [Accedido: jul. 5, 2024]

[29] Genbeta, "Eclipse IDE", Genbeta, [En línea]. Disponible: <https://www.genbeta.com/desarrollo/eclipse-ide>. [Accedido: Jul. 5, 2024]

[30] GitHub, "Getting started with the REST API", GitHub, [En línea]. Disponible: <https://docs.github.com/es/rest/using-the-rest-api/getting-started-with-the-rest-api?apiVersion=2022-11-28>. [Accedido: Jul. 5, 2024]

[31] EBAC, "¿Qué es GitHub?", EBAC, [En línea]. Disponible: <https://ebac.mx/blog/que-es-github>. [Accedido: Jul. 5, 2024]

[32] "MDPI, Sensors, vol. 22, no. 19, p. 7535, Sep. 2022. [Online]. Available: <https://www.mdpi.com/1424-8220/22/19/7535>. [Accessed: Jul. 5, 2024]

[33] Universidad Autónoma de Madrid, "Las condiciones ambientales en las aulas de Primaria en Iberoamérica y su relación con el desempeño académico", Universidad Autónoma de Madrid, [En línea]. Disponible: <https://repositorio.uam.es/handle/10486/662861>. [Accedido: Jul. 5, 2024]

[34] Modelo Canvas, "Relación con los clientes", Modelo Canvas, [En línea]. Disponible: <https://modelocanvas.net/relacion-con-los-clientes/>. [Accedido: Jul. 5, 2024]

[35] Modelo Canvas, "Recursos clave", Modelo Canvas, [En línea]. Disponible: <https://modelocanvas.net/recursos-clave/>. [Accedido: Jul. 5, 2024]

[36] TecnoHotel News, "Segmentación de clientes en la industria hotelera", TecnoHotel News, [En línea]. Disponible: <https://tecnohotelnews.com/2017/12/segmentacion-clientes/>. [Accedido: Jul. 5, 2024]

[37] Flaticon, "Propuesta de Valor - Icono Gratis", Flaticon, [En línea]. Disponible en: https://www.flaticon.es/icono-gratis/propuesta-de-valor_9792936. [Accedido: jul. 5, 2024]

Referencias

- [38] HubSpot, "Modelo de Negocio", HubSpot Blog, [En línea]. Disponible en: <https://blog.hubspot.es/sales/modelo-negocio>. [Accedido: jul. 5, 2024]
- [39] Flaticon, "Recursos - Icono Gratis", Flaticon, [En línea]. Disponible en: https://www.flaticon.es/icono-gratis/recursos_5965929. [Accedido: jul. 5, 2024]
- [40] Advenio, "Tres herramientas para visualizar tu modelo de negocio", Advenio, [En línea]. Disponible en: <https://advenio.es/tres-herramientas-para-visualizar-tu-modelo-de-negocio/>. [Accedido: jul. 5, 2024]
- [41] Startupeable, "Lean Startup", Startupeable, [En línea]. Disponible en: <https://startupeable.com/lean-startup/>. [Accedido: jul. 5, 2024]
- [42] HTTPie, "HTTPie - Command Line Tool for HTTP Clients", HTTPie, [En línea]. Disponible en: <https://httpie.io/docs/cli>. [Accedido: jul. 5, 2024]
- [43] UPM, "Guía Responsabilidad Social y Ambiental", Universidad Politécnica de Madrid, [En línea]. Disponible en: https://oa.upm.es/35542/1/Guia_Responsabilidad_Social_y_Ambiental-V2-1.pdf. [Accedido: jul. 5, 2024]
- [44] Spring Framework, [En línea]. Disponible en: <https://spring.io/>. [Accedido: jul. 5, 2024]
- [45] Telefónica, "Los dispositivos IoT y su impacto en la sostenibilidad", Telefónica Sala de Comunicación, [En línea]. Disponible en: <https://www.telefonica.com/es/sala-comunicacion/blog/los-dispositivos-iot-y-su-impacto-en-la-sostenibilidad/>. [Accedido: jul. 5, 2024]
- [46] Thymeleaf, [En línea]. Disponible en: <https://www.thymeleaf.org/>. [Accedido: jul. 5, 2024]
- [47] Cámara de Comercio de Valencia, "Blockchain: qué es y qué ventajas tiene", TIC Negocios, [En línea]. Disponible en: <https://ticnegocios.camaravalencia.com/servicios/tendencias/blockchain-que-es-y-que-ventajas-tiene/>. [Accedido: jul. 5, 2024]
- [48] Hi-Iberia, "Blockchain: ¿Qué es una red descentralizada?", Green Hi-Iberia, [En línea]. Disponible en: <https://green.hi-iberia.es/blockchain/blockchain-que-es-una-red-descentralizada/>. [Accedido: jul. 5, 2024]
- [49] D. Furlonger y C. Uzureau, "¿Qué es el blockchain?", Gartner, [En línea]. Disponible en: <https://www.gartner.es/es/articulos/que-es-el-blockchain>. [Accedido: jul. 5, 2024]
- [50] Tokio School, "¿Qué es Spring Framework?", Tokio School, [En línea]. Disponible en: <https://www.tokioschool.com/noticias/que-es-spring-framework/>. [Accedido: jul. 5, 2024]
- [51] Cloud Center Andalucía, "localhost: Qué es, conceptos básicos y cómo crearlo", Cloud Center Andalucía Blog, [En línea]. Disponible en:

<https://www.cloudcenterandalucia.es/blog/localhost-que-es-conceptos-basicos-y-como-crearlo/>. [Accedido: jul. 5, 2024]

[52] Mozilla, "HTTP Status", Mozilla Developer Network, [En línea]. Disponible en: <https://developer.mozilla.org/es/docs/Web/HTTP/Status>. [Accedido: jul. 5, 2024]

[53] Universidad Isabel I, "Tipos de Modelos de Negocios", Blog de la Universidad Isabel I, [En línea]. Disponible en: <https://www.ui1.es/blog-ui1/tipos-de-modelos-de-negocios>. [Accedido: jul. 5, 2024]

[54] Certicalia, "¿Qué es la certificación LEED?", Certicalia, [En línea]. Disponible en: <https://www.certicalia.com/certificacion-leed/que-es-la-certificacion-leed>. [Accedido: jul. 5, 2024]

[55] BREEAM España, "BREEAM España", BREEAM España, [En línea]. Disponible en: <https://breeam.es/breeam-espana/>. [Accedido: jul. 5, 2024]

[56] BOE, "Real Decreto 1027/2007, de 20 de julio, por el que se aprueba el Reglamento de Instalaciones Térmicas en los Edificios", Boletín Oficial del Estado, [En línea]. Disponible en: <https://www.boe.es/eli/es/rd/2007/07/20/1027/con>. [Accedido: jul. 5, 2024]

[57] Sensirion AG, "Sensirion CO2 Sensors SCD30 Datasheet", Sensirion, [En línea]. Disponible en: https://sensirion.com/media/documents/4EAF6AF8/61652C3C/Sensirion_CO2_Sensors_SCD30_Datasheet.pdf. [Accedido: jul. 5, 2024]

[58] GitHub, "GitHub REST API v3 - Repositories", GitHub Docs, [En línea]. Disponible en: <https://docs.github.com/es/rest/repos/repos>. [Accedido: jul. 5, 2024]

[59] Archundia, P., "Capítulo 4", Universidad de las Américas Puebla, [En línea]. Disponible en: http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/archundia_p_fm/capitulo4.pdf. [Accedido: jul. 5, 2024]

[60] Live Canvas, "Tipos de segmentos de clientes", Live Canvas, [En línea]. Disponible en: <https://live-canvas.eu/es/learning-area/full/82/3/1/tipos-de-segmentos-de-clientes>. [Accedido: jul. 5, 2024]

[61] Live Canvas, "Recursos clave", Live Canvas, [En línea]. Disponible en: <https://live-canvas.eu/es/learning-area/full/62/4/1/recursos-clave>. [Accedido: jul. 5, 2024]

[62] Live Canvas, "Costes", Live Canvas, [En línea]. Disponible en: <https://live-canvas.eu/es/learning-area/full/68/5/1/costes>. [Accedido: jul. 5, 2024]

[63] Santander, "Smart Contracts", Banco Santander, [En línea]. Disponible en: <https://www.santander.com/es/stories/smart-contracts>. [Accedido: jul. 5, 2024]

Manual de usuario

En este anexo se explicará los pasos que deberá seguir cualquier persona que quiera visualizar el sistema de monitorización en su ordenador. Este manual estará enfocado para que se ejecute dentro de mi repositorio personal, ya que se trata de un repositorio público, en el que se proporciona el token para realizar las llamadas a la API de GitHub en el fichero de 'applications.properties' de nuestro proyecto en Eclipse. El token estará disponible hasta el 20 de junio de 2025; Una vez pasada esta fecha, dejará de estar disponible y se deberá de modificar el código para hacerlo desde un repositorio personal.

A.1 Descargar Eclipse

Lo primero de todo será descargar el programa en el que se está ejecutando toda la aplicación, para ello, buscaremos en la página oficial de 'Eclipse Foundation' [<https://www.eclipse.org/downloads/packages/release/2022-12/r>] el Entorno de desarrollo integrado, IDE, correspondiente a: 'Eclipse IDE for Java Developers'. En nuestro caso, se está utilizando la versión de 2022 – 12.

A.2 Instalación de Spring dentro de Eclipse

Una vez que tenemos Eclipse, lo ejecutamos y una vez que lo tenemos abierto, procedemos a instalar Spring. Los pasos por seguir son los siguientes:

1. Navegaremos hasta 'Help' y seleccionaremos la pestaña de 'Eclipse Marketplace'. Aquí podremos encontrar las diferentes soluciones que proporciona Eclipse para instalar adicionalmente.

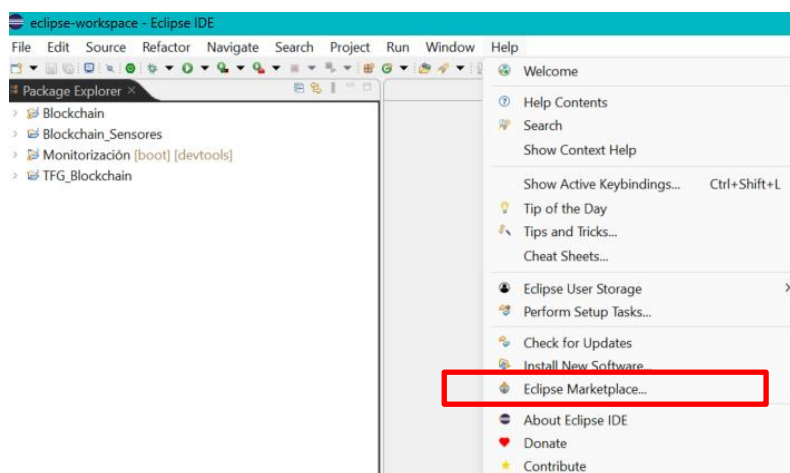


Figura 67: Eclipse Marketplace

2. Una vez dentro, buscaremos 'Spring Tools' e instalaremos la versión 4. (Si se tiene una versión de Eclipse diferente a la mencionada anteriormente se deberá buscar la compatibilidad para la que Spring funcione).

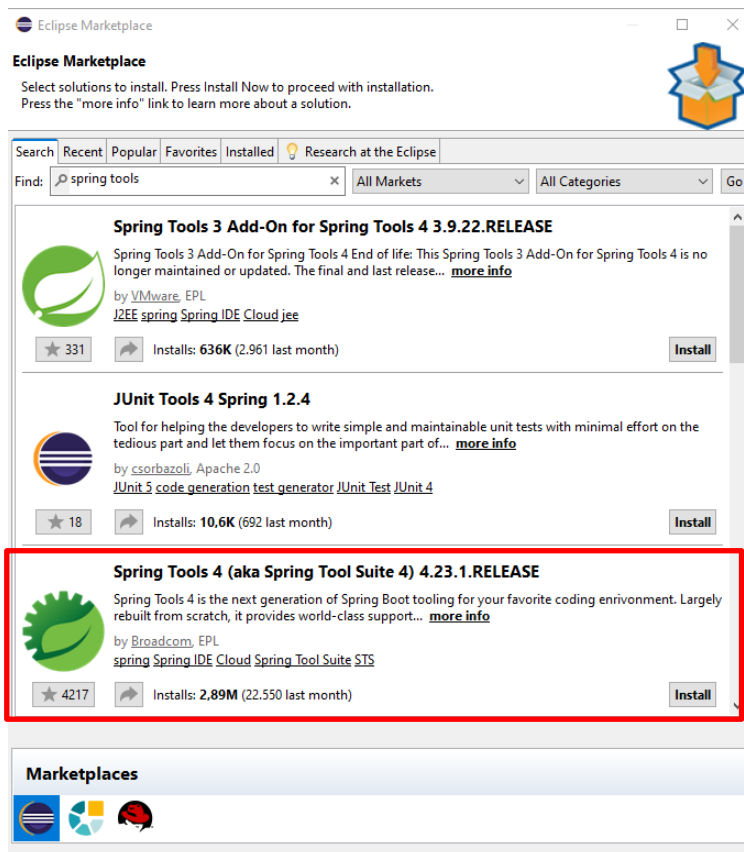


Figura 68: Spring Tools

3. Una vez que hemos dado a 'Instalar' aparecerá una ventana en la que dejaremos seleccionado lo que ya muestra por defecto y le daremos al botón de 'Confirmar'. Después de esto, deberemos de aceptar la licencia para finalizar la instalación.

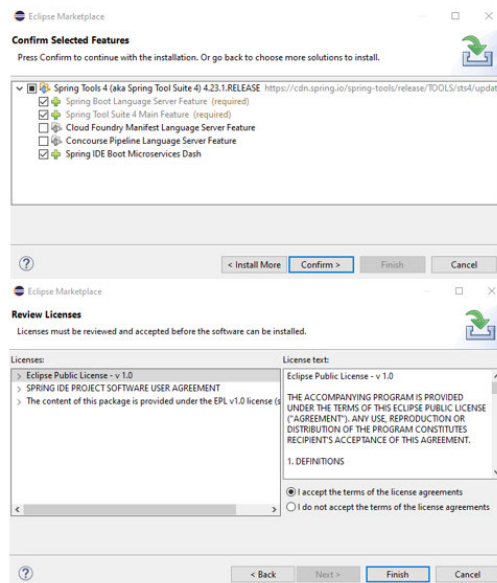


Figura 69: Instalación Spring

- Adicionalmente tendremos que aceptar que ‘Confiamos en los desarrolladores’ para que prosiga con la instalación. Una vez que hemos seguido estos pasos, podremos ver el proceso en la esquina inferior derecha.

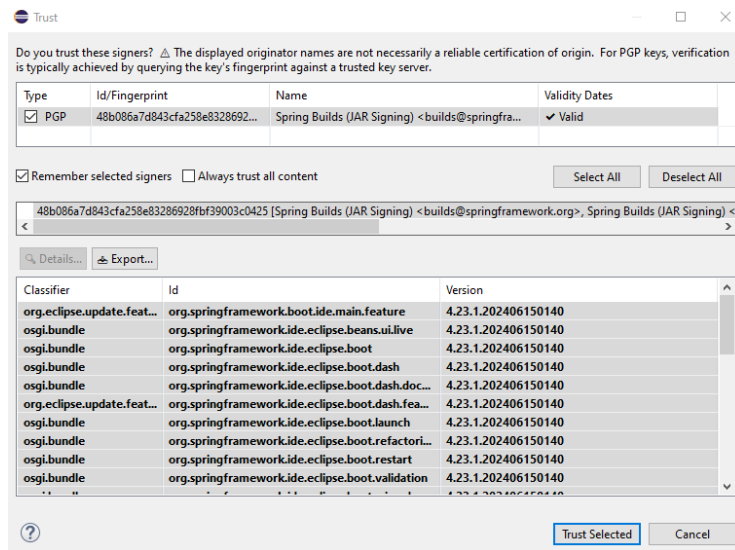


Figura 70: Confiamos en los desarrolladores

- Cuando finalice la instalación del software, nos pedirá reiniciar Eclipse para efectuar los cambios.

A.3 Instalación de Thymeleaf

Una vez que se ha instalado Spring, es necesario instalar Thymeleaf que es un motor de plantillas para Java diseñado para procesar y generar documentos HTML en aplicaciones web [62]. Los pasos que debemos seguir para su instalación son muy similares a los de Spring:

- Navegaremos de nuevo a la pestaña de ‘Help’, seleccionaremos ‘Eclipse Marketplace’ y buscaremos Thymeleaf.

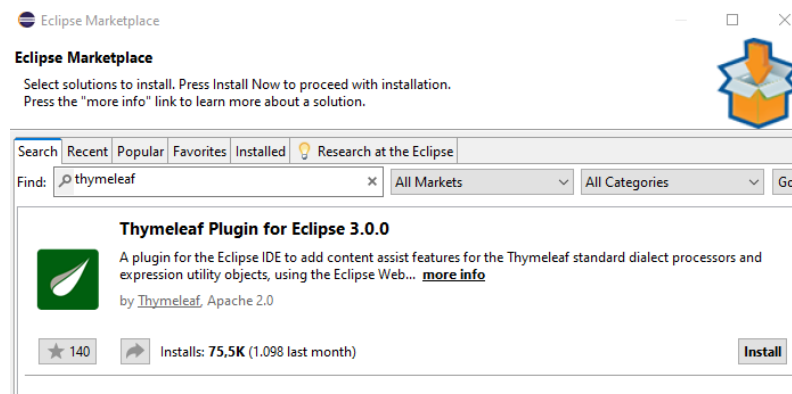


Figura 71: Thymeleaf

2. Los siguientes pasos son iguales a la instalación de Spring, por lo que se podrán seguir basándose en los explicados anteriormente.

A.4 Importación del proyecto

El siguiente paso será importar el proyecto de Monitorización en Eclipse. Para ello, deberemos seguir los siguientes pasos:

1. Descargar el código de Monitorizacion.zip del siguiente repositorio de GitHub <https://github.com/LeticiaMA7/TFG.git> . Una vez que lo tenemos descargado, deberemos descomprimirlo y copiar la carpeta de Monitorización en el 'eclipse-workspace' que tenga definido para su Eclipse.
2. En mi caso, la ruta donde se encuentra es la siguiente: C:\Users\letic\workspace
3. Para poder importar el proyecto a nuestro Eclipse, deberemos de irnos a la pestaña de 'File' y seleccionaremos 'Import'. Seguidamente tendremos que indicar que se trata de un proyecto de Maven.

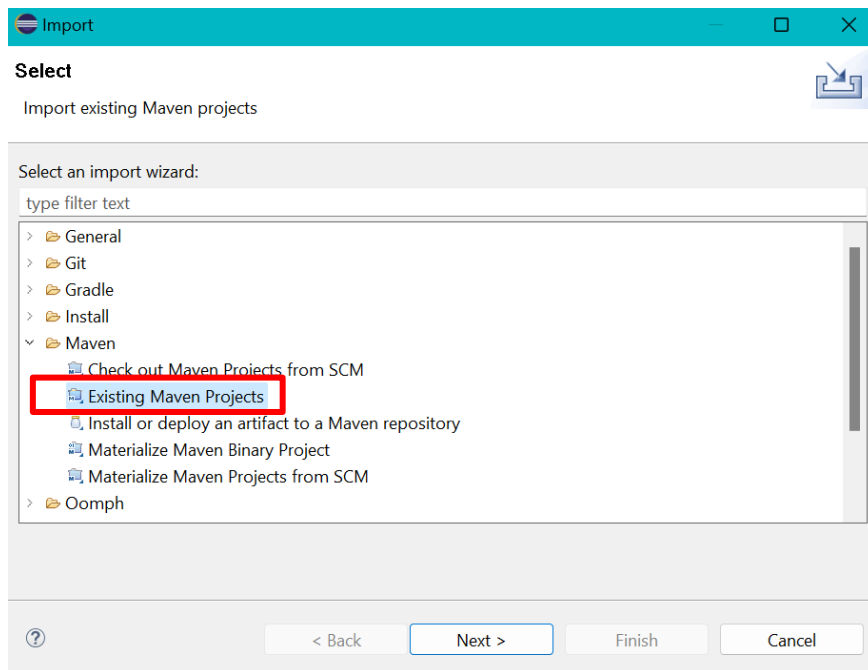


Figura 72: Importación proyecto

4. Y para finalizar la importación, tendremos que seleccionar nuestro proyecto de Monitorización.

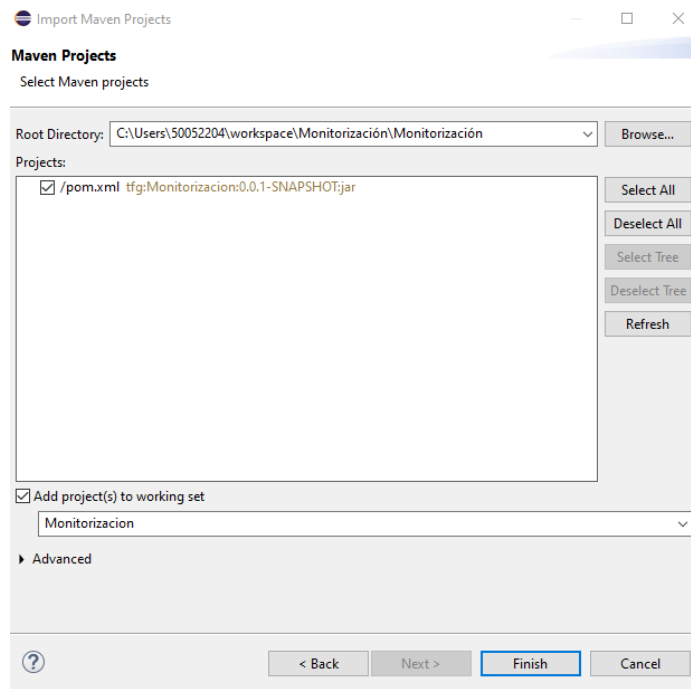


Figura 73: Selección del proyecto

5. Después de terminar la importación, ya solo tendremos que ejecutar nuestro programa y verificar que todo ha ido correctamente. Después, para poder visualizar la aplicación web, en nuestro navegador pondremos la ruta: localhost:8080.