



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Desarrollo de una aplicación IoT basada en FIWARE para
incrementar la eficiencia energética en una Smart City

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Delgado Cavero, Marcos

Tutor/a: Bonastre Pina, Alberto Miguel

CURSO ACADÉMICO: 2023/2024

Resumen

En la actualidad, cada vez son más las ciudades que deciden dar el paso para convertirse en Smart Cities debido a las ventajas que conlleva esta transformación.

El avance de las tecnologías a lo largo de los años permite un gran abanico de posibilidades a la hora de conectar los recursos disponibles, y así sacar el máximo partido a los sistemas basados en dispositivos electrónicos que nos rodean y que usamos continuamente en nuestra vida cotidiana.

Una de las aplicaciones más llamativas de este nuevo contexto tecnológico son las Ciudades Inteligentes (Smart Cities). Consisten en aplicar tecnologías como las redes de sensores, Internet de las Cosas (Internet of Things o IoT), Computación y almacenamiento en la nube (Cloud Computing) y Algoritmos de tratamientos de información (Data Science, Inteligencia Ambiental, Inteligencia Artificial) a la gestión pública municipal de espacios urbanos.

Para proporcionar la necesaria integración de todos estos servicios es necesario disponer de las denominadas plataformas de IoT, que soportan la colaboración entre las diferentes tecnologías involucradas, así como la interconexión con otras plataformas y bases de datos abiertas. Como ejemplo de aplicación de estas plataformas IoT, en este proyecto hemos empleado la plataforma Fiware, software de código abierto impulsado por la Unión Europea, con implantación en muchas de las grandes Smart Cities europeas.

En este proyecto utilizaremos la plataforma FIWARE para la implementación de una aplicación Smart City en colaboración con el ayuntamiento de Vilamarxant. Como parte de un futuro sistema integral gestión municipal, en este trabajo nos centraremos en las herramientas que permiten administrar de la forma más eficiente posible el consumo de energía en las dependencias municipales, sin desatender la comodidad de los usuarios, a la par que se garantiza el cumplimiento de los estándares de calidad exigibles.

Palabras clave: Fiware, Smart Cities, Internet of Things, Código abierto, Inteligencia artificial.

Abstract

Nowadays the number of cities that decide to take the step to become Smart Cities is increasing due to the advantages that this transformation brings.

The advance of technologies over the years allows a wide range of possibilities when it comes to connecting available resources, and thus making the most of the systems based on electronic devices that surround us and that we use continuously in our daily lives.

One of the most striking applications of this new technological context is Smart Cities. They consist of applying technologies such as sensor networks, Internet of Things (Internet of Things or IoT), cloud computing and storage (Cloud Computing) and information processing algorithms (Data Science, Ambient Intelligence, Artificial Intelligence) to municipal public management of urban spaces.

To provide the necessary integration of all these services it is necessary to have the so-called IoT platforms, which support collaboration between the different technologies involved, as well as interconnection with other platforms and open databases. As an example of the application of these IoT platforms, in this project we have used the Fiware platform, open-source software promoted by the European Union, with implementation in many of the large European Smart Cities.

In this project we will use the FIWARE platform for the implementation of a Smart City application in collaboration with Vilamarxant town council. As part of a future integral municipal management system, in this work we will focus on the tools that allow the most efficient possible management of energy consumption in municipal buildings, without neglecting the comfort of users, while ensuring compliance with the required quality standards.

Keywords: Fiware, Smart Cities, Internet of Things, Open Source, Artificial Intelligence.

Agradecimientos

A mis padres por ser las columnas de mi templo y apoyarme a lo largo de esta ardua aventura.

A mis amigos que me han acompañado durante este trayecto, y a los cuales guardo para siempre.

A mi tutor Alberto por ayudarme durante el TFG y motivarme a seguir adelante.

Tabla de contenido

1	Introducción	10
1.1	Entorno del proyecto	10
1.2	Motivación y Objetivos.....	13
1.2.1	Motivación	13
1.2.2	Objetivos.....	13
2	Estado del arte	15
2.1	Propuestas Smart Cities	15
2.2	Plataformas de Internet de las Cosas (IoT).....	19
2.3	Fiware y su arquitectura.....	20
2.3.1	Fiware	20
2.3.2	Arquitectura Fiware	21
3	Especificaciones técnicas	23
3.1	Visión general	23
3.2	Características principales de la solución	23
3.3	Actores	24
3.3.1	Administrador	24
3.3.2	Político	24
3.3.3	Técnico.....	24
3.3.4	Instalador.....	24
3.3.5	Ciudadano	24
3.4	Requisitos del producto.....	25
3.4.1	Requisitos de adquisición	25
3.4.2	Requisitos de actuación.....	25
3.4.3	Requisitos de control.....	26
3.4.4	Requisitos de supervisión	26
3.4.5	Requisitos de interconexión.....	27
3.4.6	Requisitos no funcionales	27
3.5	Metodología	28
3.6	Cronograma.....	29
3.7	Estimación de costes	32
3.8	Análisis de riesgos	33
4	Diseño.....	38
4.1	Casos de uso.....	38
5	Implementación e implantación	49
5.1	Introducción	49
5.2	Rutinas de configuración de código.....	49
5.2.1	Creación del fichero docker-compose.yml	49
5.2.2	Lanzamiento los servicios.....	54



5.2.3	Añadir los sensores y sus localizaciones	56
5.3	Rutinas de configuración gráfica	58
5.3.1	Autenticación y conexión a la aplicación	58
5.3.2	Instalación y configuración del módulo Wirecloud.....	60
6	Pruebas	67
7	Conclusiones y futuras ampliaciones	72
8	Bibliografía.....	74
9	ANEXOS.....	76
9.1	Anexo I: Manual del usuario Administrador	76
9.2	Anexo II: Manual del instalador	79
9.3	Anexo III: Objetivos de Desarrollo Sostenible (ODS)	88
9.3.1	Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).....	88
9.3.2	Reflexión sobre la relación del TFG con los ODS	89

Tabla de ilustraciones

Ilustración 1 - Indicadores Smart City	18
Ilustración 2 - Arquitectura Fiware	¡Error! Marcador no definido.
Ilustración 3 - Etapas del proyecto.....	29
Ilustración 4 - Actividades de Inicio	30
Ilustración 5 - Actividades de Planificación	30
Ilustración 6 - Actividades de Ejecución	31
Ilustración 7 - Actividades de Seguimiento	31
Ilustración 8 - Actividades de Cierre	32
Ilustración 9 - Definición BBDD no relacional Mongo DB	50
Ilustración 10 - Definición BBDD relacional Postgres.....	50
Ilustración 11 - Definición BBDD relacional Mysql.....	51
Ilustración 12 - Definición interfaz Wirecloud	52
Ilustración 13 - Definición módulo de autenticación Keyrock	53
Ilustración 14 - Definición context broker Orion.....	53
Ilustración 15 - Lanzamiento de los servicios necesarios para la aplicación.....	54
Ilustración 16 - Creación e inicio de los servicios	55
Ilustración 17 - Confirmación de levantamiento correcto de los servicios.....	55
Ilustración 18 - JSON definición de los edificios municipals.....	56
Ilustración 19 - Creación de los lugares en BBDD	57
Ilustración 20 - Verificación de los lugares en Orion	57
Ilustración 21 - Pantalla de carga Wirecloud	58
Ilustración 22 - Pantalla de autenticación Keyrock	59
Ilustración 23 - Pantalla de inicio Wirecloud.....	59
Ilustración 24 - Creación de un entorno de trabajo Wirecloud.....	60
Ilustración 25 - Añadir componente NGSI browser para Sensores y Actuadores	61
Ilustración 26 - Añadir componente mapa para las localizaciones	61
Ilustración 27 - Mapa de la relación entre componentes de la aplicación	62
Ilustración 28 - Configuración del componente Sensor.....	63
Ilustración 29 - Configuración del componente sensor	64
Ilustración 30 - Configuración del componente de los edificios municipales	65
Ilustración 31 - Configuración del componente del mapa	66
Ilustración 32 - POST de las localizaciones de los edificios municipals.....	67
Ilustración 33 - Post de los sensores de la aplicación	68
Ilustración 34 - Post de los actuadores de la aplicación.....	68
Ilustración 35 - Definición de los sensores	69
Ilustración 36 - Definición de los actuadores.....	70
Ilustración 37 - Aplicación Wirecloud con los datos enviados por los sensores y los estados de los actuadores.....	71
Ilustración 38 - Pantalla de carga de Wirecloud	76
Ilustración 39 - Registro de un nuevo usuario en Keyrock para la aplicación.....	77
Ilustración 40 - Inserción de datos del nuevo usuario.....	77
Ilustración 41 - Pantalla de entornos de trabajo de Wirecloud	78
Ilustración 42 - Terminal con la descarga de las imágenes necesarias para la aplicación.....	79
Ilustración 43 - Terminal con las respuestas de los estados HTTP de los módulos cargados	80

Ilustración 44 - Pantalla de inicio Wirecloud.....	80
Ilustración 45 - Añadir nuevos widgets en Wirecloud.....	81
Ilustración 46 - Pantalla de widgets instalados en Wirecloud	81
Ilustración 47 - Descripción pantalla de inicio para unir los componentes instalados	82
Ilustración 48 - Pantalla de unión entre los components	83
Ilustración 49 - Configuración de los componentes sensores y actuadores de la aplicación.....	83
Ilustración 50 - Configuración sensors	84
Ilustración 51 - Configuración actuadores	84
Ilustración 52 - Configuración de los componentes del mapa	85
Ilustración 53 - Configuración de las localizaciones	85
Ilustración 54 - Configuración de los atributos del mapa	86
Ilustración 55 - Configuración de las coordenadas del mapa	86
Ilustración 56 - Dashboard de la aplicación Wirecloud después de la configuración de todos los componentes.....	87

1 Introducción

En este primer capítulo de la memoria presentamos el entorno en el cual se ha desarrollado este proyecto, la motivación que ha llevado a cabo su realización, así como los objetivos que se pretenden conseguir.

1.1 Entorno del proyecto

El proyecto se desarrolla en colaboración con el M.I Ayuntamiento de Vilamarxant, en el cual encontramos una problemática relacionada con la gestión energética que se viene llevando a cabo.

Este proyecto se centra en optimizar el consumo energético de las instalaciones municipales, que actualmente se considera excesivo. Según un diagnóstico aproximado, la automatización de los sistemas eléctricos y energéticos podría reducir significativamente este consumo. Actualmente, estos sistemas son gestionados de forma manual por el personal de las instalaciones, en base a las solicitudes que se reciben en ese momento (aumentar la temperatura del recinto debido a la bajada de la temperatura exterior, bajar la temperatura debido al aumento del aforo del recinto, etc...).

Por las características de los equipamientos y la utilización registrada de los mismos, se parte de la hipótesis de que es posible realizar un seguimiento más detallado de la información en tiempo real proporcionada por diferentes sensores con los cuales se han dotado varias de las instalaciones tales como el Ayuntamiento o los recintos deportivos entre otros, y utilizar esta información de forma inteligente para obtener el objetivo deseado. Todo ello, además, en un contexto de elevados precios en la energía como consecuencia del contexto internacional.

Una de las características de la población consiste en las variaciones térmicas, por el hecho de que el pueblo se encuentra alejado de la costa, por lo que la humedad es inferior a la existente en zonas costeras. Así, sin el efecto de suavizado de temperaturas causado por el mar, las diferencias de temperaturas, tanto entre verano e invierno, como día y noche, son más acusadas.

Este proyecto, además, se enmarca en un problema mucho más importante a nivel mundial, que es calentamiento global que sufrimos debido al cambio climático. Pese a los intentos de control que se llevan a cabo, es innegable un aumento progresivo de las temperaturas en todas las partes del mundo, llegando a una previsión científica que nos indica que en 2100 algunas zonas en Asia serán inhabitables¹.

Cabe entonces resaltar la importancia de la implementación de sistemas capaces de paliar (o mitigar) los problemas actuales relacionados con el consumo energético y el incremento de emisiones de CO₂. Dichos problemas se encuentran actualmente en un punto de inflexión dado el control que se trata de ejercer sobre los mismos, pero no siempre con la eficacia deseada.

Hoy en día son cada vez más los estamentos, tanto públicos como privados, que ponen su enfoque en este problema energético. Desde organizaciones no gubernamentales que buscan limitar, a través de pactos, las emisiones de CO₂ producidas por los países y empresas para

¹ (@NatGeoES, 2017, p. 1)

controlar el cambio climático, hasta grandes corporaciones que implementan, ya sea por beneficio fiscal o por propia iniciativa, diferentes objetivos de desarrollo sostenible enfocados en este ámbito con el fin de limitar también el gasto y uso energético, el impacto que tienen en el ecosistema e incluso su impacto en la sostenibilidad a nivel global.

La propia Comisión Europea ha puesto en marcha el denominado pacto “Green Deal”² que busca tres objetivos claros:

- Garantizar un suministro de energía seguro y asequible para la UE.
- Desarrollar un mercado de la energía de la UE completamente integrado, digitalizado e interconectado.
- Priorizar la eficiencia energética, mejorar el rendimiento energético de nuestros edificios y desarrollar un sector energético basado en gran medida en fuentes renovables.

Esto se une a las diferentes líneas energéticas³ que la UE ha ido marcando como objetivos (por ejemplo, los marcados para 2020) que de nuevo buscan poder resolver el problema energético desarrollado.

Este proyecto propone la utilización de los avances tecnológicos presentes con la finalidad de controlar y optimizar ciertas tareas, hasta ahora ejecutadas de forma manual, de tal forma que se permita su escalabilidad y siempre manteniendo el respeto por la sostenibilidad.

En un mundo en el que la reducción del tamaño y el coste de los microprocesadores nos permite día a día crear dispositivos más potentes, adaptativos y en espacios mucho más reducidos, se ha producido también el perfeccionamiento de las tecnologías de comunicaciones, tanto inalámbricas como el 5G, Sigfox, Lora como cableadas, sirviendo como ejemplo la fibra óptica, que permiten dar un paso adelante a la hora de incorporar los avances de dicho hardware a la vida cotidiana.

Una parte de esta extensión de las capacidades de proceso y comunicaciones a los objetos cotidianos es la interconexión entre diferentes aparatos que no se habían planteado antes, como por ejemplo los electrodomésticos o los sensores integrados en lugares como nuestra casa.

La aplicación de estas técnicas a estos sensores ha dado lugar a las Redes de Sensores y Actuadores (WSANs), que son sistemas distribuidos compuestos por una red de sensores que recopilan datos y actuadores que responden a esos datos. Estas WSANs se utilizan en una variedad de aplicaciones, desde la monitorización ambiental y la agricultura de precisión hasta la automatización industrial y la gestión de edificios inteligentes. Al permitir la recopilación de datos en tiempo real y la toma de decisiones basada en esos datos, las WSANs están transformando la forma en que interactuamos con el entorno y mejorando la eficiencia en una amplia gama de campos.

Este avance tecnológico está allanando el camino para un mundo más conectado y automatizado, donde la información y la acción se entrelazan de manera inteligente para mejorar nuestra calidad de vida y optimizar procesos en diversos sectores.

Sin embargo, la tecnología ya permite un paso más, hasta el punto de crearse un nuevo paradigma en el cual los objetos se convierten en dispositivos inteligentes. Es el denominado Internet de las Cosas. El Internet de las Cosas (IoT) representa un paradigma tecnológico

² (Energy and the Green Deal, s. f., p. 2)

³ (¿Cuáles son las fuentes de energía de la UE?, s. f., p. 1)

fascinante en el que la interconexión de objetos cotidianos y su transformación en dispositivos inteligentes están revolucionando la forma en que interactuamos con el mundo que nos rodea. En el marco del IoT, una amplia gama de objetos, desde relojes y termostatos hasta automóviles y ciudades enteras, se conectan a la red y adquieren la capacidad de comunicarse, recopilar datos y tomar decisiones informadas. Esto no solo se limita a dispositivos que usamos en nuestro día a día, sino que también se extiende a áreas como la salud, la agricultura, la logística y la industria.

Un aspecto esencial del IoT es la capacidad de recopilar datos de sensores incorporados en estos objetos, lo que permite una comprensión más profunda y detallada de nuestro entorno y de cómo interactuamos con él. Por ejemplo, sensores de temperatura en una ciudad pueden proporcionar datos en tiempo real para optimizar la gestión del tráfico y reducir la congestión, o sensores de salud pueden monitorear constantemente los signos vitales de un paciente y alertar a los profesionales médicos en caso de anomalías.

Además, el IoT no solo se trata de recopilar datos, sino también de actuar sobre ellos de manera inteligente. Los dispositivos inteligentes pueden realizar tareas automatizadas, como ajustar la temperatura de una habitación o encender las luces cuando detectan que alguien ha entrado. También pueden tomar decisiones basadas en datos en tiempo real, como el riego de cultivos en función de las condiciones climáticas actuales.

El IoT tiene un enorme potencial para mejorar la eficiencia, la seguridad y la calidad de vida. Sin embargo, también plantea desafíos importantes en términos de privacidad y seguridad de datos, así como cuestiones éticas relacionadas con la recopilación y el uso de información personal. A medida que el IoT continúa evolucionando, es esencial abordar estos desafíos de manera responsable para aprovechar al máximo su potencial y garantizar que los beneficios superen a los posibles riesgos.

Todos esos avances se han aplicado, a lo largo de este Trabajo Fin de Grado, en la consecución de los objetivos que se detallan a continuación, junto a la motivación que los ha impulsado.

1.2 Motivación y Objetivos

1.2.1 Motivación

La motivación que ha llevado a la elaboración de este trabajo y estudio no es otra que la capacidad de aportar una solución real y de calidad al problema existente en cuanto a eficiencia energética se refiere.

Dicho problema tiene un origen más antiguo, que data prácticamente del inicio de la revolución industrial, época en la cual las prácticas empleadas para desarrollar las diferentes actividades que han dado lugar inicialmente a los productos y servicios que tenemos hoy en día. Estas prácticas se han ido manteniendo a la vez que se han ido añadiendo algunas nuevas, fruto de la refinación de los servicios y productos cada vez más modernos de los cuales disponemos. En conjunto, han ido dañando poco a poco nuestro planeta.

Por este motivo buscaremos poder solucionar el problema a una escala local (debido los medios de los que disponemos) en el ayuntamiento de Vilamarxant, donde la falta de control y desarrollo de los sistemas medidores y controladores de la eficiencia energética del pueblo nos han llevado a poder sugerir una nueva forma de gestionar esta problemática a través de una solución que emplee las tecnologías existentes que se adapten lo mejor posible a la infraestructura que ya ha sido instalada en toda la localidad.

De esta forma trataremos de aportar nuestro pequeño grano de arena en cuanto a control del gasto energético se refiere.

1.2.2 Objetivos

El objetivo principal del proyecto es dotar al Ayuntamiento de la herramienta para supervisar y controlar el consumo energético de los edificios municipales.

Para la supervisión proporcionaremos un panel de control que muestre en tiempo real el consumo energético de cada instalación, y los parámetros de uso que se correlacionan con el mismo (temperatura exterior e interior, aforo actual, uso actual y previsto – por ejemplo, encender una calefacción una hora antes de utilizar una sala de conciertos –, etc)

Llevaremos esto a cabo en una infraestructura informática basada en tecnologías IoT para soportar una colección de aplicaciones orientadas al campo del consumo energético. Dicha infraestructura se monta a través de una plataforma de código abierto llamada Fiware⁴.

Nuestros objetivos secundarios serán varios.

Primero, detallaremos la arquitectura utilizada para la implementación de la aplicación desarrollada en Fiware.

En segundo lugar, propondremos soluciones tecnológicas que permitan automatizar y controlar diferentes acciones relacionadas con el consumo energético de las dependencias municipales, actualmente efectuadas de forma manual por operarios o bien mal controladas administrativamente, de tal forma que se pueda reducir al máximo el consumo energético, así como el gasto que se incurre en él pasando por los sensores y actuadores mencionados previamente. Esto se implementará como panel visual interactivo en la aplicación Fiware.

⁴ («Fiware», 2022, p. 1)

Tercero, para el control, implementaremos un conjunto de actuadores que permitan actuar sobre el consumo energético cuando sea necesario (apagar calefacciones, ventilación, control de luminarias, etc.) y ser gestionados manualmente por los administradores del sistema desde la consola visual ofrecida por el sistema.

En cuarto lugar, guardaremos un histórico de los datos de medición en una nueva base de datos que permita controlar el consumo real para garantizar una facturación correcta en todo momento. Guardaremos de la misma forma un histórico de las acciones de los actuadores. En definitiva, almacenaremos todo dato recogido por el sistema en bases de datos que pondremos a disposición del ayuntamiento de Vilamarxant.

Por último, en quinto lugar, interconectaremos el resto de aplicaciones (mencionadas e implementadas en el resto de TFGs relacionados) con el sistema presentado a través de una aplicación gestionada por el personal cualificado y autorizado del ayuntamiento del municipio, con el fin de poder sacar el máximo partido a todos los datos recogidos por los diferentes sistemas y garantizar de esta forma un entorno funcional y una toma de decisiones mucho más ágil y determinada, resultando en un ahorro económico y energético.

2 Estado del arte

2.1 Propuestas Smart Cities

Varias ciudades son las que han sido capaces de implementar con éxito ya proyectos que les han permitido convertirse en SmartCities.⁵

En primer lugar, podemos citar el caso de Barcelona. Premiada como la mejor ciudad mundial inteligente por el instituto Juniper Research en 2015, Barcelona se posiciona como un referente en soluciones de conectividad aplicadas a ciudades.⁶

Este reconocimiento ha sido obtenido a partir de la implementación de diferentes proyectos, tales como:

- El sistema de transporte público ha puesto en marcha el uso de autobuses híbridos, placas solares en las marquesinas y una red de rutas diagonales, permitiendo así que los autobuses sean más rápidos y frecuentes.

- La gestión de residuos de la ciudad a partir de “contenedores inteligentes”, el cual utiliza un sistema de vacío para aspirar los residuos bajo tierra. Esto resulta en la reducción de olores y contaminación acústica causada por los camiones de basura.

- El alumbrado público de bajo consumo basado en el uso de LEDs con una red de sensores. Esto reduce el calor causado por las lámparas antiguas. Igualmente, los sensores reciben la información de la temperatura, humedad... permitiendo así ajustar las luces en consecuencia.

- Las calles cuentan con una red de sensores instalados debajo de las carreteras, lo que permite llevar un conteo de los aparcamientos públicos. De esta forma, cada vez que se libera una plaza de aparcamiento, una aplicación avisa a los conductores de su localización. Se han reducido así tanto la contaminación como los tiempos de búsqueda de sitio en la calle.

Estas medidas han permitido que Barcelona ocupe esta primera posición en el ranking de ciudades inteligentes de España.

En segundo lugar, está Madrid, la capital del país. Varios proyectos tienen como objetivo posicionar la ciudad como uno de los referentes de SmartCity en España.⁷

El principal, el proyecto MiNT iniciado en 2014, busca poder permitir a los ciudadanos y revisores de Madrid enviar información a tiempo real acerca de las incidencias que están ocurriendo desde su propio smartphone. Una fotografía con su correspondiente localización geográfica permitirá a la herramienta de análisis INSA poder recibir los datos y ponerlos a

⁵ (Smart City, 2017, p. 1)

⁶ (Barcelona smart city o ciudad inteligente, 2019, p. 1)

⁷ (La nueva capital smart city de la mano de Madrid Nuevo Norte, 2021, p. 1)

disposición del ayuntamiento, con el objetivo de cubrir esa incidencia lo antes posible y prevenir futuras incidencias similares.⁸

Igualmente, a nivel urbanístico, se está llevando a cabo el proyecto “Madrid Nuevo Norte”, el cual incorporará desde el inicio de la edificación el uso de domótica y el desarrollo en pro de la “urbe 2.0”. Se pretende de esta forma poner al ciudadano en el centro de la toma de decisiones para poder mejorar la calidad de vida desde el punto de vista económico, medioambiental y social.

Tal y como se ha implementado en Barcelona, Madrid Nuevo Norte pretende cubrir la zona implicada con sensores en todos los lugares posibles (luces, carreteras, bancos, marquesinas, semáforos...) permitiendo recoger toda la información aportada para poder tomar decisiones en función de estos valores y así adaptarse a la vida de las personas que viven ahí.

Gracias a estas iniciativas, Madrid ha podido situarse en el segundo lugar en cuanto a Smart Cities se refiere.

En tercer lugar, se encuentra Valencia, la cual fue la primera de todas las ciudades en poner en marcha su plataforma de Smart City, llamada VLCi.

Esta plataforma ofrece una visión general de su gestión urbana y sus servicios, la cual al ser de código abierto hace que sea considerada una de las ciudades más transparentes a nivel de gestión ciudadana.

En 2017, se puso en marcha la plataforma VLCi, que ayuda a poner a tiempo real toda la información sobre 60 servicios municipales que participan en esta red en línea, 600 indicadores para medir la ciudad y una aplicación para poder organizar toda la información extraída para que sea visible online y en directo. Además, gracias a la plataforma VLCi se han ahorrado unos 17 millones en la gestión, tal y como se puede comprobar en su web.⁹

De nuevo la implementación llevada a cabo reside en la puesta en marcha de varios servicios inteligentes:

- Instalación de 2000 luminarias en pueblos norte y sur de la ciudad (añadido a las que ya han sido instaladas en el centro de la ciudad). De esta forma se dispone de más información a tiempo real de los distintos tramos de alumbrado de la ciudad y se puede así reducir el consumo y mejorar la sostenibilidad medioambiental.
- “Wifi València¹⁰”, permite disponer de conectividad WiFi en los espacios públicos más transitados tales como bibliotecas, centros culturales, museos, centros municipales...
- Proyecto Geoportal¹¹, el cual devuelve información acerca de planos sobre mapas de la ciudad, mapas históricos de Valencia, jardines, mercados... así como un uso administrativo como por ejemplo apartamentos turísticos, incidencias en la vía pública...
- Instalación de más de 40 dispositivos en la red de autobuses EMT permitiendo así el control de la calidad del aire, humedad y temperatura externa para poder controlar así la temperatura interna en los autobuses.

⁸ (Proyecto mint, 2014, p. 1)

⁹ («Proyectos de transformación hacia la Ciudad Inteligente», s. f., p. 1)

¹⁰ («València se integra en la red europea WiFi4EU y ofrece internet gratuito de alta velocidad», s. f.)

¹¹ («Geoportal», s. f., p. 1)

- Aparcamientos inteligentes, indicando a tiempo real los aparcamientos disponibles para personas de movilidad reducida, carga y descarga, paradas de taxi y zonas de aparcamiento azul.

Estas prácticas han situado a Valencia como la tercera ciudad más importante en cuanto a Smart Cities se refiere.

Además, Smartcity Málaga, presenta un proyecto diseñado para cumplir con las directivas europeas sobre energía, fomento de la eficiencia, uso de energías renovables y redes eléctricas avanzadas con capacidad de almacenamiento. Con un presupuesto de 31 millones de euros, contó con el apoyo económico del CDTI (Centro de Desarrollo Tecnológico e Industrial). Los 4 años que el proyecto estuvo vigente han derivado en las siguientes formas de ahorro:

- el ahorro de más del 25 % en el consumo de energía en su área de implementación mediante el uso de sistemas de eficiencia energética que monitorean, controlan y administran de manera proactiva las demandas de los usuarios industriales y residenciales.
- El 42% de las trabajadoras del hogar han reducido su consumo energético en más de un 10% mediante el uso de kits de eficiencia energética en el hogar que les permiten gestionar sus gastos desde cualquier parte del mundo a través de sus teléfonos inteligentes.
- Se han evitado 4.500 toneladas de emisiones de CO₂ de al año.¹²

Además, el proyecto ha sido reconocido como uno de los más importantes en el campo de las Smart Grids¹³ (red eléctrica inteligente), debido a su gran envergadura en la extensión de las diferentes áreas de trabajo implicadas.

Estas son solo 4 de las ciudades más desarrolladas en el ámbito de las Smart Cities, aunque existen muchas más en el territorio español, y cada día son más las ciudades que poco a poco optan por adoptar medidas inteligentes para el control de la urbe, buscando regular y adaptarse a las facilidades y mejoras que las Smart Cities ofrecen.

Por último, cabe resaltar que existe un índice que nos permite medir el grado de avance de las Smart Cities en España. Este índice se llama “Índice Smart”¹⁴, el cual se basa en 5 dimensiones a su vez divididas en 15 categorías en total:

¹² (SmartCity Málaga, s. f., p. 1)

¹³ (*Smart Grids*, s. f.)

¹⁴ (455df16a1a, s. f.)

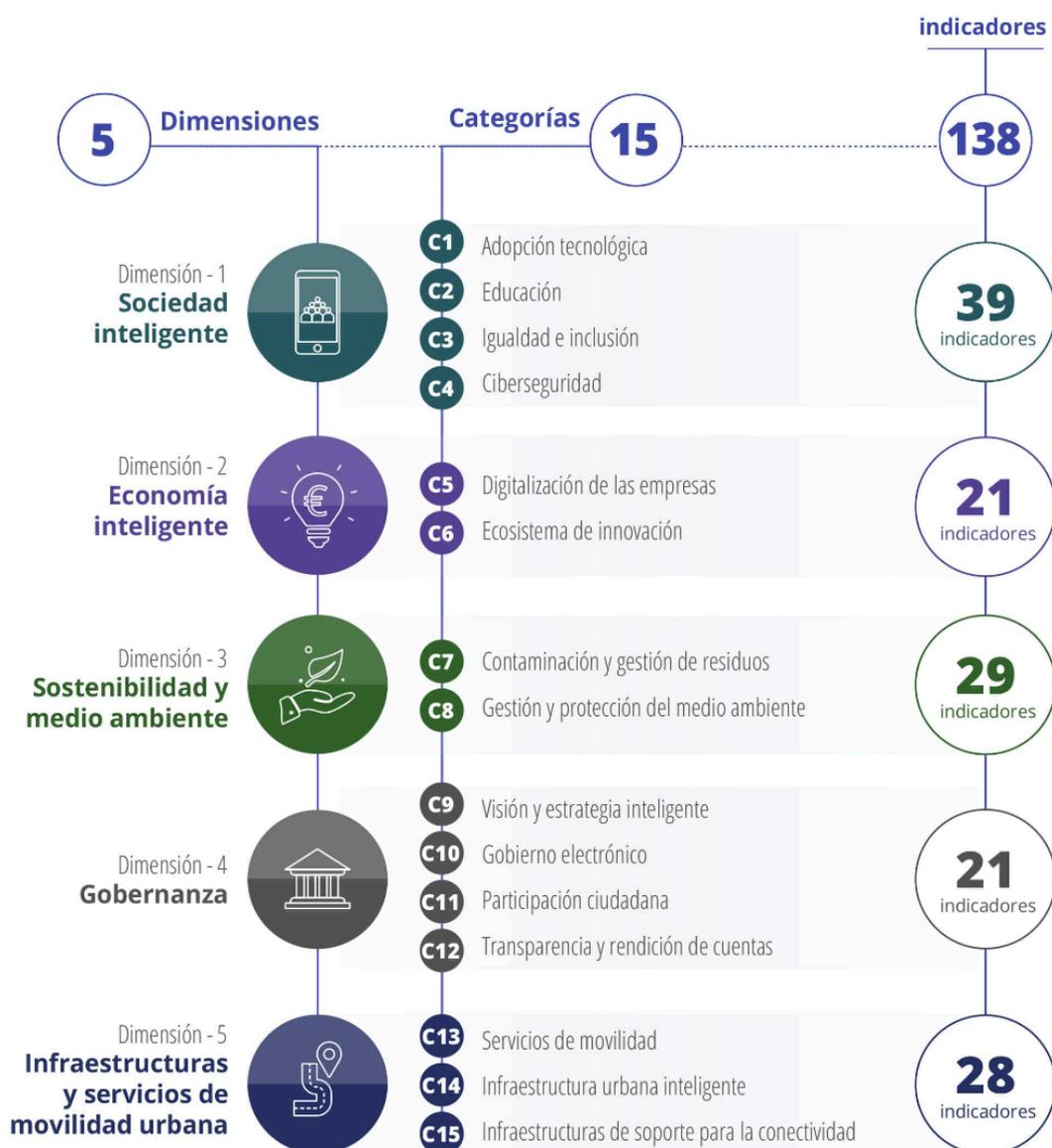


Ilustración 1 - Indicadores Smart City

De nuevo, para cada categoría se han definido una serie de indicadores que nos permiten establecer el porcentaje “Smart” de la ciudad estudiada. Dichos indicadores toman en cuenta cualidades como la inclusión digital de la población mayor (cuántas personas de 65 años o más disponen de ordenador o tablet o se han conectado a Internet en los últimos tres meses), o los recursos de comunicación y recursos digitales universitarios accesibles, entre otros.

Basándonos en esta información, los datos que hemos podido extraer de la web ¹⁵ para 2023 son los siguientes:

El rendimiento promedio del Índice Smart Cities de las ciudades en España es de un 48,32%. Encabezan el listado ciudades como Barcelona (79%), seguida de Madrid (74,1%), Valencia (66,7%) y San Sebastián (66,6%), siendo aquellas con el promedio más alto.

Sin embargo, encontramos aún un margen de mejora dado que ninguna de las ciudades se encuentra al 100% (ni siquiera sobrepasa el 90% en su índice), por lo que a lo largo de los

¹⁵ (ESMARTCITY, 2023, p. 2)

siguientes 10 años se calcula un aumento exponencial de la orientación socio-geográfica de las ciudades hacia la visión Smart City.

2.2 Plataformas de Internet de las Cosas (IoT)

A medida que nos proyectamos hacia el futuro, los avances en las ciudades inteligentes de España se ven impulsados por una evolución constante de las plataformas IoT. Estas plataformas actúan como el tejido conectivo que une los diversos elementos urbanos, permitiendo la recopilación y análisis de datos en tiempo real para una toma de decisiones más informada y eficiente.

La integración de sensores más avanzados y la mejora en la capacidad de procesamiento de datos son aspectos clave que caracterizarán la próxima fase de la revolución digital en el ámbito urbano.

Uno de los horizontes más emocionantes es la convergencia de las plataformas IoT con la inteligencia artificial (IA) y la analítica de datos. Esta sinergia permitirá a las ciudades inteligentes no solo recopilar información en tiempo real, sino también analizar patrones complejos y prever tendencias. Por ejemplo, en Barcelona, la combinación de datos de sensores IoT con algoritmos de aprendizaje automático podría mejorar la predicción de congestiones de tráfico y permitir una gestión más proactiva de la movilidad urbana.

La implementación generalizada de la tecnología 5G será un catalizador clave para acelerar el potencial de las ciudades inteligentes en España. La conectividad ultrarrápida permitirá una comunicación instantánea entre dispositivos, mejorando la eficiencia y la capacidad de respuesta de los sistemas urbanos. Esta mejora en la velocidad de transmisión de datos impulsará la expansión de aplicaciones innovadoras, desde el monitoreo de la salud de la infraestructura hasta la gestión en tiempo real de servicios esenciales.

Si bien las ciudades inteligentes actuales ya han logrado avances notables en sostenibilidad, el futuro promete una expansión significativa en esta área. La optimización de recursos se verá potenciada por algoritmos más avanzados y la integración de energías renovables en la red eléctrica urbana.

Se espera que las ciudades inteligentes de España lideren iniciativas de sostenibilidad a nivel global, adoptando prácticas más ecológicas y reduciendo aún más la huella ambiental.

La evolución de las ciudades inteligentes se traducirá en una experiencia ciudadana más enriquecedora y personalizada. La recopilación y análisis de datos permitirán servicios urbanos adaptados a las necesidades individuales de los ciudadanos. Por ejemplo, en Madrid, la combinación de datos de movilidad, preferencias personales y hábitos de consumo podría dar lugar a recomendaciones personalizadas de transporte, promoviendo un estilo de vida más eficiente y centrado en el usuario.

A medida que avanzamos hacia esta era futurista de ciudades inteligentes en España, surgen nuevos desafíos éticos y humanos. La privacidad de los datos, la equidad en el acceso a la tecnología y la participación ciudadana en la toma de decisiones serán aspectos críticos que requerirán una atención especial.

La tecnología debe ser un medio para mejorar la calidad de vida y no una barrera que excluya a segmentos de la sociedad.

Estos avances tecnológicos en las ciudades inteligentes pueden alcanzar su máximo potencial mediante la integración efectiva con plataformas como Fiware. La flexibilidad y

escalabilidad de Fiware permiten una gestión unificada de datos, facilitando la interoperabilidad entre diversas tecnologías y garantizando una implementación armoniosa de soluciones avanzadas.

La fusión de las plataformas IoT, la inteligencia artificial y la conectividad 5G con Fiware no solo fortalece la base tecnológica de las ciudades inteligentes, sino que también abre nuevas oportunidades para la creación de servicios y aplicaciones más eficientes y centrados en las necesidades de la ciudadanía.

Esta integración estratégica con Fiware se convierte así en un elemento clave para alcanzar una sinergia completa entre los avances tecnológicos y el desarrollo continuo de las ciudades inteligentes en España.

2.3 Fiware y su arquitectura

Dada la amplia información que tenemos disponible acerca de la plataforma y sus diferentes usos, la separaremos en dos partes. Describiremos primero de la plataforma Fiware en sí en primera instancia, y detallaremos en segunda instancia la arquitectura que hemos usado en este proyecto.

2.3.1 Fiware

Fiware se erige como una potente plataforma de código abierto¹⁶, diseñada para impulsar la innovación en el ámbito de las ciudades inteligentes y la Internet de las Cosas (IoT). En su esencia, se presenta como un ecosistema tecnológico que proporciona a desarrolladores, empresas y entidades gubernamentales las herramientas necesarias para crear soluciones inteligentes y conectadas.

La visión de Fiware es la de superar barreras tecnológicas y facilitar el desarrollo de aplicaciones que mejoren la calidad de vida en las ciudades y optimicen procesos en diversos sectores. Al adoptar un enfoque de código abierto, Fiware fomenta la colaboración y la creación de soluciones accesibles y eficientes.

Fiware nos presenta numerosas características.

Primero, se presenta como un entorno flexible capaz de adaptarse a una amplia gama de aplicaciones. Desde la gestión de servicios urbanos hasta la monitorización industrial, la plataforma proporciona un conjunto de herramientas versátiles que se pueden aplicar a diferentes contextos.

En segundo lugar, al adoptar estándares abiertos, la plataforma facilita la integración con una variedad de sistemas y tecnologías, permitiendo a los usuarios construir soluciones que no están limitadas por las barreras de la compatibilidad.

Tercero, está diseñado para escalar, permitiendo su implementación tanto en proyectos a pequeña escala como en despliegues a nivel de ciudad o industria. Esta capacidad de escalabilidad brinda a los usuarios la confianza de que la plataforma puede crecer junto con las necesidades de sus proyectos.

Por último, se distingue por su enfoque amigable para los desarrolladores. Proporciona herramientas y APIs intuitivas que aceleran el proceso de desarrollo y permiten un despliegue

¹⁶ (*Build Your Own IoT Platform with FIWARE Enablers – FIWARE*, 2015, p. 1)

rápido de aplicaciones. Esto es crucial para proyectos que buscan resultados ágiles y eficientes.

2.3.2 Arquitectura Fiware

La arquitectura de Fiware se compone de varios elementos que trabajan de manera conjunta para ofrecer una solución integral. Esta arquitectura se refleja en la siguiente figura:

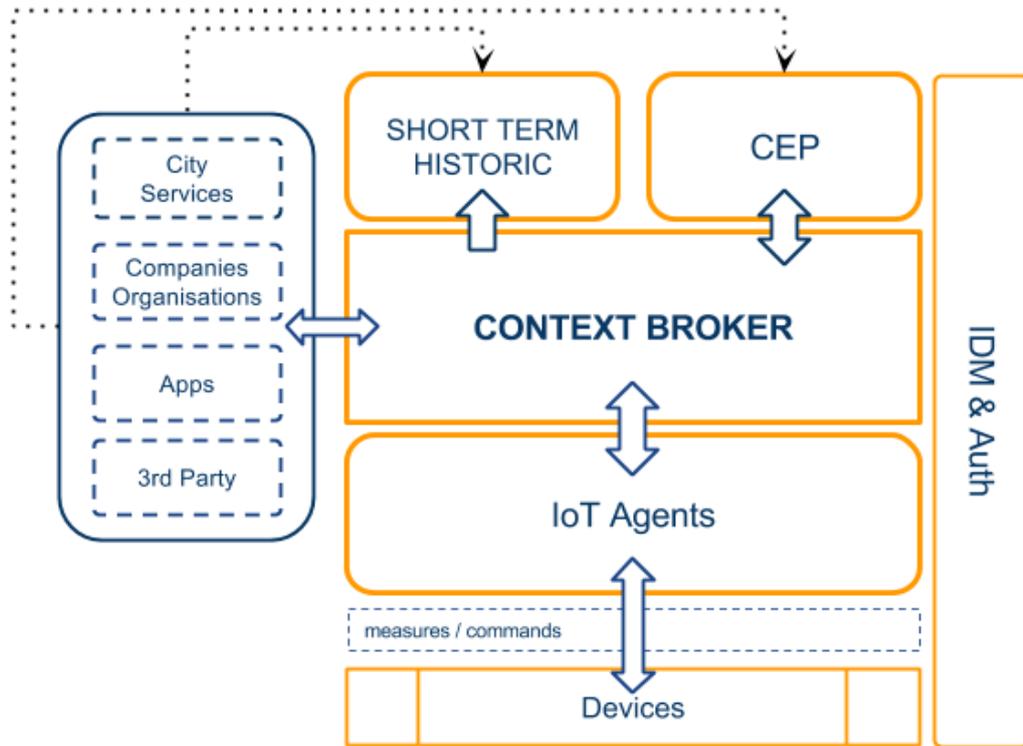


Ilustración 2 - Arquitectura Fiware

Los elementos más sobresalientes de esta arquitectura son los siguientes;

1. Orion Context Broker

El Orion Context Broker, como componente central, se comunica con las aplicaciones y dispositivos a través de la API NGSI. Esta interfaz proporciona operaciones estándar para consultar, actualizar y suscribirse a los datos contextuales. La comunicación se realiza principalmente mediante solicitudes HTTP y el intercambio de datos utiliza formatos como JSON.

2. NGSI v2

NGSI v2 establece la especificación para la estructura y gestión de datos en Fiware. Proporciona una interfaz uniforme que permite a las aplicaciones interactuar de manera coherente con el Orion Context Broker y otros componentes. Esta uniformidad facilita la creación de aplicaciones que pueden adaptarse y escalar fácilmente.

3. Bases de Datos (MongoDB y PostgreSQL)

Fiware admite tanto bases de datos NoSQL como MongoDB, conocida por su flexibilidad en esquemas de datos, como bases de datos relacionales como PostgreSQL, que ofrecen estructuras más rígidas y relaciones complejas. La elección entre estas bases de datos depende

de los requisitos específicos de la aplicación, proporcionando a los desarrolladores la libertad de seleccionar la tecnología que mejor se adapte a sus necesidades.

4. Keyrock: Gestión de Identidades y Accesos

Keyrock desempeña un papel crítico en la arquitectura, gestionando la autenticación y autorización de usuarios y aplicaciones. Utiliza estándares como OAuth 2.0 para proporcionar un control de acceso seguro y escalable. La integración con Keyrock asegura que solo las entidades autorizadas accedan a los recursos protegidos en la plataforma.

5. Integración con Wirecloud

Wirecloud se conecta a Fiware para ofrecer capacidades de interfaz visual avanzada. Permite a los desarrolladores diseñar paneles de control, visualizaciones de datos y aplicaciones personalizadas de manera intuitiva. La comunicación entre Wirecloud y Fiware se establece mediante las APIs proporcionadas por Fiware, facilitando la creación de experiencias de usuario atractivas e interactivas.

Podemos concluir que no solo se destaca por su arquitectura técnica sólida, sino también por su enfoque en la estandarización y la interoperabilidad. Al proporcionar herramientas y estándares abiertos, Fiware empodera a desarrolladores, ciudades e industrias para crear soluciones innovadoras en el ámbito de las ciudades inteligentes y la IoT.

La combinación de flexibilidad en las bases de datos, la gestión de identidades con Keyrock, y la capacidad de integración con herramientas visuales como Wirecloud, posiciona a Fiware como una plataforma completa para la creación de aplicaciones inteligentes y conectadas en la era digital.

3 Especificaciones técnicas

3.1 Visión general

En este proyecto se propone un sistema de supervisión y control del consumo energético de diversas instalaciones del Ayuntamiento de Vilamarxant, como son las instalaciones deportivas, oficinas, servicios de atención al público, etc.

El consumo de estos edificios no está supervisado automáticamente, sino que se controla manualmente. Se considera que mediante un sistema automático de medición y control de consumo se conseguiría reducir sustancialmente el gasto energético. Asimismo, la climatización de dichas instalaciones sigue el mismo patrón: no hay un control automatizado de los sistemas y por lo tanto se sospecha – ya que no se dispone de información precisa – que se produce un consumo energético mayor al necesario.

Para resolver este problema, en este proyecto se lleva a cabo el análisis, desarrollo e implementación de una aplicación de control energético sobre la plataforma Fiware.

La arquitectura de este sistema se basa en la utilización de unos sensores, que monitorizarán diversos parámetros de consumo y ambientales. La información recogida será transmitida a la aplicación, que la analiza para racionalizar el uso de los recursos energéticos en función de la carga necesaria con el objetivo de garantizar un consumo adecuado, respetando siempre el margen necesario para garantizar los mejores servicios a los usuarios de las instalaciones.

Detallaremos a continuación las especificaciones técnicas seguidas para resolver esta problemática.

3.2 Características principales de la solución

El nuevo sistema incluirá/permitirá lo siguiente:

- El control del consumo energético de los edificios públicos a través de la monitorización de los sensores instalados en ellos.
- El seguimiento a tiempo real de la climatización con el fin de poder regular la calefacción de forma automática.
- Puesta en marcha de una serie de actuadores que permitan a los usuarios encargados de gestionar el sistema de llevar a cabo la regulación del mismo gracias a las informaciones extraídas por los sensores.
- Un panel de control adaptado al usuario identificado en ese momento que muestre los datos de interés en base a su capacidad de toma de decisiones.
- Una base de datos capaz de almacenar el histórico de datos de consumo.
- Una base de datos que permita el almacenamiento de los usuarios del sistema, así como la totalidad de los sensores y actuadores.

3.3 Actores

3.3.1 Administrador

El administrador es responsable de llevar a cabo las tareas de gestión más importantes del sistema. Tiene permisos totales como súper usuario (CRUD): creación, borrado, actualización y modificación de perfiles, sensores, etc...

De la misma forma, tiene una visión completa del panel y de la información que recogen. Es capaz de determinar la información que se visualiza para el resto de perfiles existentes.

3.3.2 Político

El político es capaz de consultar la información relevante agregada y/o anonimizada para la toma de decisiones relacionadas con la gestión del consumo energético. De esta forma puede identificar las desviaciones en el consumo esperado y adoptar medidas para corregirlas.

No tiene los permisos necesarios para realizar acciones CRUD de ningún tipo.

3.3.3 Técnico

El técnico puede consultar toda la información relacionada con la gestión y monitorización de los dispositivos físicos, sensores y actuadores. De esta forma es capaz de diagnosticar cuáles requieren una revisión o cambio en caso de que no funcionen como se espera o que devuelvan datos visiblemente incorrectos.

Solamente puede realizar acciones CRUD relacionadas con los sensores que aparecen en el panel, siendo el responsable de agregar sensores y actuadores al sistema.

3.3.4 Instalador

El instalador es capaz de obtener la misma información en el panel que el Técnico para su supervisión, pero no se le atribuyen los permisos necesarios para realizar acciones CRUD sobre el panel.

3.3.5 Ciudadano

El ciudadano no puede realizar ningún tipo de acciones CRUD. Solo tiene acceso visible a la información pública determinada por el administrador. Dicha información podrá incluir, entre otras, la temperatura de los recintos, la temperatura exterior, las emisiones de CO₂ y la estimación del consumo energético ahorrado en tiempo real...

3.4 Requisitos del producto

3.4.1 Requisitos de adquisición

Los sensores permiten obtener información en tiempo real sobre las magnitudes a medir: Consumo energético, temperatura, humedad, etc. Desde el punto de vista de la aplicación, ésta deberá cumplir los requisitos principales siguientes:

- a1 Poder añadir sensores al sistema permitiendo así la incorporación de nuevas zonas públicas a medida se vayan añadiendo sensores nuevos en dichas zonas. De esta forma se podrá tener un sistema cada vez más controlado y automatizado.
- a2 Poder eliminar sensores que hayan dejado de utilizarse o se hayan cambiado de ubicación.
- a3 Leer los datos de los sensores a tiempo real a través de la automatización de la recogida de datos con un tiempo de refresco determinado por el usuario autorizado a ello.
- a4 Probar sensores nuevos que se hayan añadido, sensores que se hayan repuesto o sensores que no se estén utilizando todavía. Se busca poder testarlos sin impactar en el sistema ya puesto en producción.
- a5 Almacenamiento de los datos adquiridos en bases de datos para su posterior tratamiento y/o consulta histórica.

3.4.2 Requisitos de actuación

Un actuador es un elemento que permite realizar acciones físicas en función de las decisiones tomadas por la aplicación del algoritmo sobre los datos de entrada proporcionados por los sensores. En este caso, se contemplan, entre otros, la actuación sobre los equipos de climatización y ventilación: encendido, apagado, diagnóstico, referencias, etc.

Respecto a los actuadores, la aplicación deberá cumplir los requisitos siguientes:

- b1 Poder añadir actuadores que permitan la ejecución de la acción de control prevista por el algoritmo controlador o personal técnico.
- b2 Eliminar los actuadores que ya no se vayan a utilizar, ya sea porque la zona de control ya no está sensorizada, fallo del actuador, cambio por otro mecanismo, etc...
- b3 Listar los estados de los actuadores para poder evaluar su desgaste y poder confirmar su correcto funcionamiento.
- b4 Probar nuevos actuadores que se vayan a instalar antes de que sean puestos directamente en un entorno productivo. De esta forma podremos evaluar en un entorno de prueba si están listos para ser incorporados.

3.4.3 Requisitos de control

La aplicación deberá permitir al usuario, debidamente autenticado:

- c1 Especificación, diseño e implementación del algoritmo de control para:
 - La gestión de la temperatura de los edificios municipales.
 - La gestión de la energía eléctrica de los mismos.
 - Estimación y control de la cantidad de CO₂ emitida por las instalaciones para su posterior información.
- c2 Visualizar las acciones realizadas por los algoritmos de control a través del histórico almacenado en base de datos.
- c3 Implementar adecuadamente y con la seguridad necesaria la base de datos en la cual se lleva a cabo el seguimiento histórico de medidas y actuaciones para evitar ataques externos (conjunto con resto de TFGs relacionados).

3.4.4 Requisitos de supervisión

La aplicación deberá proporcionar, a los usuarios convenientemente autorizados:

- d1 Acceso rápido a la información adquirida y acciones efectuadas en forma de panel de control.
- d2 Capacidad de añadir usuarios al sistema.
- d3 Capacidad de eliminar usuarios ya existentes del sistema.
- d4 Capacidad de modificar los usuarios con permisos de acceso al sistema, así como sus roles dentro del mismo.
- d5 Capacidad de filtrado de la información en función de diversos parámetros
 - Ver todas: se visualizará toda la información existente en el sistema (solamente para el administrador).
 - Ver por encima de umbral de consumo: permitirá visualizar la información en base a parámetros de control establecidos con el fin de agilizar la toma de decisiones sobre los actuadores correspondientes.
 - Ver detalles de una zona concreta: se visualizará un edificio o zona determinada para focalizar la atención sobre dicho lugar.
- d6 Posibilidad de refrescar la información con consultas instantáneas (suponiendo que la sensorización permita el refresco, en caso contrario se determinará una frecuencia de envío de mensajes para obtener la información correspondiente).

- d7 Generación de alertas visuales (bajo cambio de color) en la plataforma para que el usuario sea capaz de identificar los diferentes datos recogidos.

- d8 Resaltar mediante indicadores visuales en el panel de control, las siguientes condiciones de alerta:
 - Alerta en caso de fallo de uno de los sensores
 - Alerta cuando el sistema determine que la temperatura de la calefacción es demasiado alta/baja en comparación con la cantidad de gente dentro del edificio y en relación a la temperatura exterior.
 - Alerta cuando el consumo energético en uno o varios edificios sobrepase o esté por debajo del rango determinado.

3.4.5 Requisitos de interconexión

- e1 Compartir datos con otras aplicaciones con el fin de facilitar la toma de decisiones relacionada y permitirles actualizar su histórico en base a esta información compartida. En conjunto con resto de TFGs relacionados.

- e2 Adquirir datos de otras aplicaciones con el fin de agilizar la toma de decisiones propia y actualizar igualmente el histórico existente gracias a dicha información compartida. En conjunto con resto de TFGs relacionados.

3.4.6 Requisitos no funcionales

- f1 Privacidad de los datos (acceso en función del perfil de usuario). Cada perfil debe de tener restringida el resto de información correspondiente en base a los permisos que se le han asignado. (conjunto con resto de TFGs relacionados)

- f2 Registro de incidencias:
 - La herramienta de control debe garantizar que el consumo llevado a cabo sea el correcto en todo momento para poder automatizar las tareas antes realizadas por un operario.
 - Se debe crear un registro en todo momento de la variación de los parámetros para su posterior medición y facturación.

3.5 Metodología

La gestión de proyectos en cascada es un enfoque clásico y secuencial que se ha utilizado durante décadas para planificar, ejecutar y completar proyectos de manera ordenada. En el contexto del desarrollo de este proyecto, este enfoque proporciona una estructura sólida para abordar las complejidades del proyecto y asegurar que se cumplan los requisitos específicos del cliente.

Fases de la Metodología en Cascada:

1. Inicio: En esta fase inicial, se trabajará estrechamente con las partes influyentes e interesadas en el proyecto, representantes del Ayuntamiento de Vilamarxant, para definir el alcance del proyecto y establecer los objetivos específicos. Se identificarán las necesidades clave y se establecerán las bases para el desarrollo de la aplicación Fiware.
2. Planificación: Con los objetivos claros, se procederá a la planificación detallada del proyecto. Esto implica la elaboración de un cronograma, la asignación de recursos, la identificación de posibles riesgos y la definición de los entregables específicos para cada etapa del desarrollo.
3. Ejecución: Una vez que la planificación está completa y se han establecido los fundamentos, se inicia la fase de ejecución. En este punto, se desarrollará la aplicación de acuerdo con las especificaciones y el diseño previamente establecidos.
4. Seguimiento: Durante la fase de seguimiento, se llevará a cabo un monitoreo constante del progreso del proyecto. Se realizarán revisiones periódicas para evaluar el avance en comparación con el cronograma planificado, identificar posibles desviaciones y abordar cualquier problema emergente. Además, se realizarán pruebas continuas para garantizar la calidad del producto en desarrollo.
5. Cierre: La fase de cierre implica la finalización del proyecto y la entrega de la aplicación al Ayuntamiento de Vilamarxant. Se llevará a cabo una revisión final para asegurarse de que todos los requisitos se hayan cumplido satisfactoriamente. Además, se proporcionará capacitación si es necesario, y se documentarán los procesos y el código para facilitar futuras actualizaciones y mantenimiento.

Existen otras metodologías a la hora de llevar a cabo un proyecto de desarrollo software, tales como las metodologías ágiles. Aun así, hemos decidido utilizar la metodología en cascada ya que presenta numerosas ventajas:

- Estructura Clara: La metodología en cascada proporciona una estructura clara y fácilmente comprensible, lo que facilita la planificación y ejecución del proyecto. Cada fase tiene objetivos y resultados específicos, lo que ayuda a mantener el enfoque y la organización.

- **Visibilidad del Progreso:** Debido a su naturaleza secuencial, la metodología en cascada permite una clara visibilidad del progreso proyecto en cada etapa. Esto facilita la evaluación continua y la identificación temprana de posibles desviaciones o problemas.
- **Documentación Exhaustiva:** Cada fase de la metodología en cascada requiere documentación detallada. Esto no solo garantiza la comprensión clara de los requisitos y el diseño, sino que también facilita la transferencia de conocimientos entre equipos y la futura mantenibilidad de la aplicación desarrollada.
- **Control de Cambios:** Dado que cada fase debe completarse antes de pasar a la siguiente, la metodología en cascada facilita el control de cambios. Los cambios en los requisitos o el diseño pueden evaluarse y gestionarse de manera más efectiva antes de que se avance demasiado en el proceso.

La elección de la metodología de gestión de proyectos en cascada para el desarrollo de la aplicación informática de Fiware destinada al Ayuntamiento de Vilamarxant se basa en la necesidad de establecer una estructura sólida y secuencial que se alinee de manera efectiva con los requisitos específicos de este proyecto.

Aunque las metodologías ágiles podrían adaptarse bien a entornos en constante cambio, la elección de la metodología en cascada refleja la importancia de una planificación meticulosa y una ejecución precisa en un proyecto con requisitos específicos y objetivos claramente definidos, como es el caso del desarrollo de la aplicación de Fiware para este proyecto.

3.6 Cronograma

Para la implementación del cronograma nos hemos basado en las etapas de todo proyecto de implantación software. Según la metodología anteriormente descrita.

Para ello hemos tomado las cinco etapas descritas en el punto anterior, las cuales han sido detalladas a su vez en diferentes actividades y hemos computado por horas por persona, ya que el proyecto no se ha medido por una temporalidad fija, es decir, por una fecha de inicio y fin definidas por contrato.

En las imágenes a continuación podremos explicar el cronograma que hemos definido, actividad por actividad.

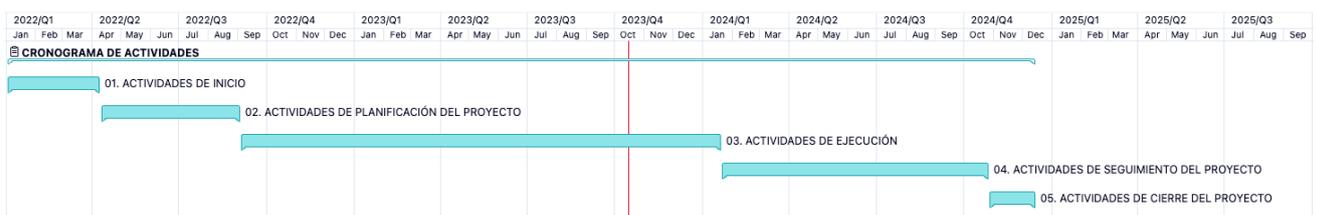


Ilustración 3 - Etapas del proyecto

Primero, detallaremos las actividades de Inicio.



Ilustración 4 - Actividades de Inicio

En las actividades de inicio encontramos las tareas relacionadas con la definición y firma de los documentos relativos al proyecto tales como el acta de constitución, el acta de comunicaciones, la matriz de riesgos, ...

Se toman los requisitos necesarios para la implementación del sistema y se revisan con el cliente, en este caso el ayuntamiento de Vilamarxant.

Una vez revisados los documentos, se firman para validar el comienzo del proyecto.

Resaltan por duración dos tareas:

La primera la tarea de reuniones de revisión del sistema y análisis de los puntos de mejora. Dicha actividad será recurrente durante todo el proyecto ya que estaremos en contacto constante con la parte funcional del cliente para poder confirmar de manera ágil que las implementaciones testeadas corresponden con los requisitos firmados inicialmente.

La segunda tarea que resalta es la toma de requisitos en base a las mejoras solicitadas. En dicha tarea llevaremos a cabo diversas reuniones con el cliente para plasmar sobre el papel los requisitos que el cliente espera satisfacer a la hora de la entrega del sistema.

En segundo lugar, veremos las actividades de Planificación.

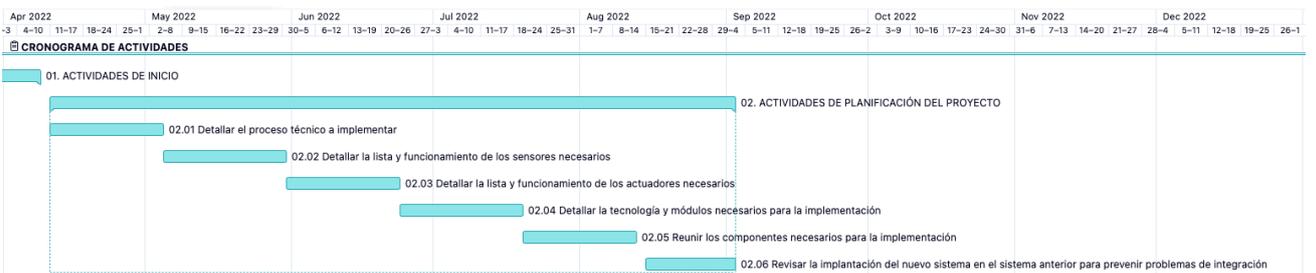


Ilustración 5 - Actividades de Planificación

Dichas tareas constan de detallar a nivel técnico el documento que ilustra el proyecto que se va a llevar a cabo, el cual será el que se usaremos como guía para el desarrollo de la aplicación, así como detallar los elementos necesarios para poner en marcha el sistema correspondiente (en este caso los actuadores, los sensores y componentes del sistema) y revisar que el antiguo sistema contiene los datos que posteriormente serán migrados al nuevo sistema, y por último confirmar que la nueva tecnología permite la adaptación esperada.

Nuestra tercera fase corresponde a las actividades de ejecución.



Ilustración 6 - Actividades de Ejecución

Éstas muestran las tareas ligadas al desarrollo propio del sistema sobre la plataforma Fiware, así como la configuración de los sensores y actuadores que han sido requeridos.

Estas tareas han sido diferenciadas en diseño del nuevo sistema, configuración de sensores y actuadores, y la implementación del nuevo sistema.

En esta primera parte de diseño buscaremos detallar técnicamente cómo usaremos el código abierto proporcionado por Fiware para poder adaptarlo al sistema en el que lo vamos a integrar.

Durante la configuración nos centraremos directamente en la puesta a punto de los sensores y actuadores para que puedan conectarse y funcionar en base a los parámetros deseados. Esta parte la hemos dividido en dos (sensores y actuadores) para poder focalizar la entregas y revisiones con el cliente, agilizando estas dos actividades lo máximo posible.

En la parte de implementación pondremos en marcha la redacción del código necesario. Aprovecharemos igualmente esta tarea para poder migrar progresivamente los datos del sistema antiguo al nuevo.

En cuarto lugar, detallaremos las actividades de seguimiento.

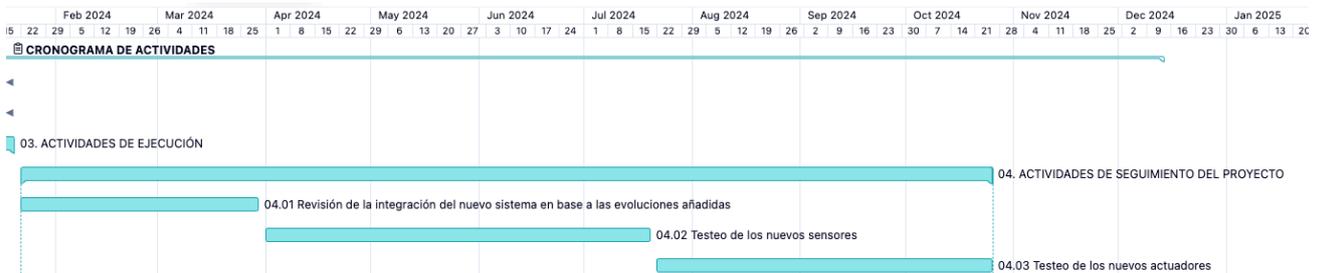


Ilustración 7 - Actividades de Seguimiento

Hemos representado esta actividad después de las actividades de ejecución puesto que será en este momento cuando haremos la revisión completa del sistema y revisaremos/testearemos con el cliente el sistema entregado. Sin embargo, desarrollaremos estas actividades de seguimiento de forma iterativa durante las actividades de ejecución (siguiendo la metodología SCRUM) permitiendo así un desarrollo ágil del proyecto.

Dichas actividades de seguimiento constan de la revisión del sistema nuevo y del testeo de los nuevos sensores y actuadores que han sido añadidos al sistema.

Por último, llegamos a las actividades de cierre.



Ilustración 8 - Actividades de Cierre

En éstas se revisa con el cliente el cumplimiento de los requisitos establecidos en el acta de inicio del proyecto, así como el funcionamiento del nuevo sistema que hemos implantado.

En caso de que todo esté de acuerdo con lo solicitado, se entrega el proyecto. En esta última etapa se lleva a cabo la firma del acta de cierre, validando la entrega final con el cliente y confirmando el cierre del proyecto.

3.7 Estimación de costes

Una vez realizada la estimación temporal, podremos realizar una estimación de costes que nos permita reflejar el análisis económico del proyecto, sobre el cual reposará la viabilidad económica. Detallaremos la estimación en 2 categorías diferentes: Hardware y Software.

Categoría	Concepto	Coste	Total
Hardware	Sensores de humedad x20	26,70€	1 068€
	Sensores de temperatura x20	26,70€	1 068€
	Actuadores para la regulación x2	45€	90€
	Soporte instalación para sensores x40	5€	200€
	Soporte instalación para actuadores x2	10€	20€
Software	Entorno web para pruebas y desarrollo	200€	200€
	Entorno de base de datos	200€/mes	200€/mes

Coste total	2 846€
-------------	--------

El precio del hardware y software es de 2 646€ inicialmente, teniendo en cuenta un coste de 200€ mensuales por el mantenimiento de la base de datos.

Una vez estimados los costes fijos, pasamos a detallar los costes de la mano de obra. Partimos del hecho que la contratación se hace mediante un contrato de oferta pública por lo que los costes de la seguridad social y derivados corren a cargo de la empresa o freelance que lleva a cabo el proyecto. En este caso no los detallaremos aquí.

Etapa del proyecto	Duración en semanas	Coste total
Actividades de inicio	15	7 800€
Actividades de planificación	21	10 920€
Actividades de ejecución	72	37 440€
Actividades de seguimiento	39	20 280€
Actividades de cierre	6	3 120€
TOTAL	153	79 560 €

En este caso no tomamos en cuenta el tiempo de redacción del TFG ni la defensa ante tribunal ya que estamos detallando el coste del proyecto en sí. Dichos hitos (redacción y defensa) no se facturan ni impactan el desarrollo del proyecto de cara al cliente final.

En la tabla previamente mostrada, detallamos los costes de la mano de obra restando 14 semanas, que corresponden al tiempo estimado para la redacción y defensa del TFG.

El coste total por las 153 semanas es de 79 560€. Esto resulta en un coste de 13€/hora.

3.8 Análisis de riesgos

A continuación, detallaremos el análisis de riesgos del proyecto. Es necesario identificar y evaluar cuantitativa y cualitativamente los riesgos de tal forma que podamos reducir el impacto de los mismos en caso de que pudieran llegar a suceder.

Para ello, los analizaremos mediante tablas básicas con la descripción del riesgo e identificación, la probabilidad, la gravedad y el plan de contingencia a aplicar en caso de que suceda.

ID: R01	Riesgo: Falta de control sobre la implementación pedida
Descripción	Una vez definidas las especificaciones de la aplicación, el cliente solicita nuevos arreglos o correcciones que están fuera del alcance del proyecto
Probabilidad	Muy alta
Gravedad	Baja a grave (dependiendo de la solicitud de cambio)
Plan de contingencia	Prevención: implementación mediante una metodología ágil que se base en revisiones continuas con el cliente para confirmar que la implementación es la esperada. Corrección: revisión con el cliente de las especificaciones pactadas a la hora de las nuevas solicitudes no contempladas para analizar el impacto. Si el impacto es mayor de lo contemplado en el margen permitido, descartaremos la

	solicitud o estimaremos el coste temporal de su implementación después de haberla presupuestado.
--	--

ID: R02	Riesgo: Mala planificación o subestimación de los tiempos
Descripción	Los tiempos estimados inicialmente son más bajos de lo necesario y se deben ampliar los recursos temporales/personales/materiales
Probabilidad	Media
Gravedad	Baja - Media
Plan de contingencia	Prevenición: confirmación de las implementaciones a efectuar para que no se pasen los tiempos estimados. Reuniones semanales de revisión de la planificación para confirmar que los tiempos se siguen respetando. Corrección: usar el tiempo de contingencia que ya se había previsto y sumarlo a los tiempos iniciales. En caso de que el tiempo necesario sea mayor, revisaremos con el cliente el tiempo necesario a añadir y reestimaremos los costes / recursos humanos implicados.

ID: R03	Riesgo: Falta de disponibilidad del cliente para implementación del proyecto
Descripción	El cliente no puede asistir a las reuniones regulares o la parte de recursos humanos asignados por el cliente no pueden dedicar el tiempo necesario
Probabilidad	Media
Gravedad	Alta
Plan de contingencia	Prevenición: al inicio del proyecto, detallaremos las horas de recursos humanos necesarios por parte del cliente para pre-reservar el tiempo necesario y poder acceder a los recursos durante el proyecto sin impedimentos. Corrección: establecer reuniones con mayor margen para poder bloquear ese espacio de tiempo y asegurar la disponibilidad.

ID: R04	Riesgo: Incompatibilidad de versiones
Descripción	Las versiones de los sistemas operativos en el entorno de desarrollo inicial y los entornos de test y producción del cliente son diferentes, lo que puede provocar incompatibilidades.
Probabilidad	Media
Gravedad	Media
Plan de contingencia	Prevenición: preparar inicialmente el sistema de desarrollo con las versiones de destino que tendrá el sistema de desarrollo y producción del cliente. Corrección: la actualización del sistema puede causar fallos, incluso habiendo previsto que las versiones actuales sean las mismas. Una copia de seguridad se hará con cada instalación para asegurar que se pueda volver a una versión estable que funcione correctamente.

ID: R05	Riesgo: Problemas de despliegue de la infraestructura
Descripción	Después de validar las versiones y el entorno, es posible que el sistema no permita la instalación por problemas de restricciones de seguridad internas o compatibilidad (que no sea causada por la versión del sistema).
Probabilidad	Baja - Media

Gravedad	Alta
Plan de contingencia	<p>Prevención: haremos una preinstalación del sistema en el entorno de desarrollo con los mismos parámetros que el entorno productivo para confirmar que funciona todo correctamente. Migraremos el sistema al entorno productivo una vez validado.</p> <p>Corrección: involucraremos al departamento de informática y soporte en caso de necesidad para acomodar los entornos de desarrollo / producción, con el fin de poder eliminar las restricciones de seguridad que puedan impedir el correcto funcionamiento.</p>

ID: R06	Riesgo: Errores o imprecisiones en las especificaciones
Descripción	En las especificaciones definidas en el acta de inicio hay huecos sin cubrir o se encuentran imprecisiones. Esto nos puede llevar a nuevas solicitudes de implementación o frustración por parte del cliente.
Probabilidad	Media
Gravedad	Media
Plan de contingencia	<p>Prevención: validaremos con el cliente todas las especificaciones iniciales en el acta de forma conjunta. Validaremos igualmente a lo largo de los entregables que corresponden con lo esperado por parte del cliente.</p> <p>Corrección: Asumimos que puede haber ciertos puntos a clarificar durante la implementación (los cuales cubrimos con los tiempos y costes dados como margen).</p>

ID: R07	Riesgo: Las pruebas realizadas no son suficientes
Descripción	El producto se entrega sin haber realizado todas las pruebas necesarias y se encuentran fallos a la hora del uso en el entorno productivo
Probabilidad	Baja
Gravedad	Alta
Plan de contingencia	<p>Prevención: realizaremos por nuestro lado una batería de pruebas automatizadas y pruebas de usuario en el entorno de desarrollo. El cliente realizará también todas las pruebas necesarias antes del paso al entorno de producción.</p> <p>Corrección: Resolveremos el problema encontrado lo antes posible y haremos la importación en el entorno de desarrollo para confirmar que el parche no afecta a las demás funcionalidades actuales. Una vez validado, actualizaremos el entorno productivo con el parche.</p>

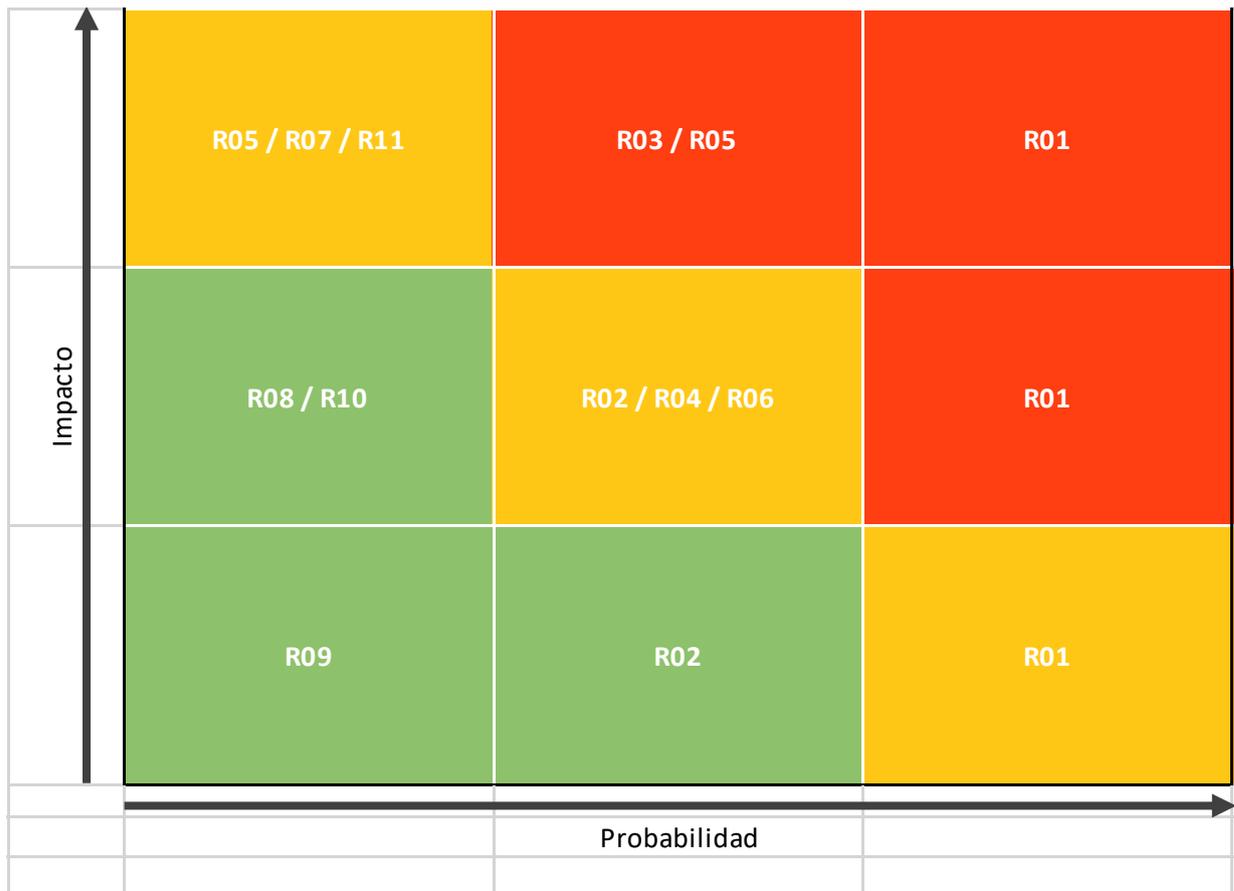
ID: R08	Riesgo: Cambio en la directiva del proyecto / ayuntamiento
Descripción	La directiva encargada de validar el proyecto cambia por otra diferente.
Probabilidad	Baja
Gravedad	Media
Plan de contingencia	<p>Prevención: el proyecto se lleva a cabo durante el mandato de la directiva que lo ha aprobado. Igualmente, al tratarse de una oferta pública, se supone que el proyecto sigue adelante, aunque la directiva no sea la misma.</p> <p>Corrección: en caso de tener que cerrar el proyecto sin haberlo acabado, facturaremos los costes en los que se ha incurrido.</p>

ID: R09	Riesgo: Problemas varios ligados con las inclemencias temporales
Descripción	Debido a las lluvias, nevadas u otros problemas temporales, los sensores, actuadores o sistemas informáticos se ven afectados y dejan de funcionar.
Probabilidad	Baja
Gravedad	Baja
Plan de contingencia	Prevenición: los sensores, actuadores y sistemas informáticos están asegurados físicamente (sensores y actuadores están guardados en un lugar físico que les permita dar un uso correcto sin exponerse al exterior. Los sistemas informáticos están asegurados y los establecimientos públicos responden a las normas ISO respectivas). Corrección: Se cambiarán los sensores/actuadores que puedan fundirse por problemas eléctricos y crearemos copias de seguridad de los sistemas informáticos que permitan reponerlos en otro lugar rápidamente.

ID: R10	Riesgo: La seguridad informática del sistema se ve comprometida
Descripción	El nivel de seguridad de la aplicación es insuficiente o se ve comprometida por un ataque externo profesional.
Probabilidad	Baja
Gravedad	Media
Plan de contingencia	Prevenición: cumplir con todos los estándares de seguridad requeridos por el estamento público donde se hace la implantación y llevar al día las actualizaciones de seguridad necesarias. Corrección: Identificar rápidamente la información comprometida y contratar una empresa externa que permita solucionar la brecha de seguridad de la forma más rápida posible.

ID: R11	Riesgo: El proyecto no es rentable
Descripción	Después de haber estudiado la viabilidad del proyecto y haber aceptado una oferta inicial, el incremento de los costes y el tiempo del proyecto lo hacen no rentable.
Probabilidad	Baja
Gravedad	Alta
Plan de contingencia	Prevenición: con cada aumento del tiempo o coste establecido, revisaremos con el cliente que seguimos cumpliendo el presupuesto y/o tiempo establecido(s). Corrección: Aquí tenemos dos opciones: <ul style="list-style-type: none"> - entregar el proyecto con lo facturado en el estado en el que se encuentra en ese momento. - Reestimar en la medida de lo posible el tiempo o el coste, reduciendo funcionalidades y permitiendo la finalización del proyecto.

La matriz de riesgos es la siguiente:



En la matriz de riesgos anterior podemos ver la disposición de los riesgos que hemos identificado, los cuales plasmamos en función de su impacto y de la probabilidad de que ocurran.

Como podemos apreciar, los cuadrantes con mayores riesgos son los que se sitúan con un impacto grande y poca probabilidad (riesgos R05, R07 y R11) y los que se sitúan con un impacto medio y probabilidad media (R02, R04 y R06).

Esto nos indica que hemos identificado y balanceado correctamente los posibles riesgos del proyecto.

4 Diseño

4.1 Casos de uso

Describiremos a partir de los siguientes casos de uso las actividades llevadas a cabo por los diferentes actores que tienen acceso a la aplicación desarrollada y detallaremos así el proceso seguido por cada uno de ellos.

1. Clase general para Usuario

Existe una superclase usuario que es genérica para todos los actores. Personalizaremos para cada uno de los actores los puntos necesarios que se tengan que adaptar a sus requisitos.

1.1. Caso de uso: Inicio de sesión en la aplicación

- Nombre: Login
- Actor principal: Usuario / Administrador
- Precondiciones: Ninguna
- Cumple requisitos: f1.

A) Escenario principal:

- a) El usuario hace clic sobre el botón “Login” de la pantalla de la aplicación
- b) El usuario introduce sus credenciales (usuario y contraseña) en el panel.
- c) El sistema accede a la base de datos para comprobar que las credenciales existen y son correctos.
- d) El usuario accede al sistema identificado como usuario / administrador.
- e) El caso de uso de inicio de sesión finaliza con éxito.

B) Escenarios alternativos:

- a) El usuario introduce de forma incorrecta sus credenciales.
 - 1) El sistema devuelve un mensaje de error notificando el fallo al acceder.
 - 2) El sistema solicita al usuario sus credenciales de nuevo, hasta un máximo de 2 veces más.
 - 3) El sistema comprueba en la nueva inserción que las credenciales son correctas. En caso afirmativo, devuelve al usuario al paso A-d.
 - 4) En caso contrario, el caso de uso finaliza sin éxito y bloquea el usuario temporalmente.

C) Escenarios de excepción:

- a) El sistema no puede acceder a la base de datos
 - 1) El sistema devuelve un aviso indicando que no ha podido acceder a la base de datos del servidor para comprobar las credenciales.
 - 2) El caso de uso finaliza sin éxito.

2. Caso de uso: Modificar los sensores activos

- Nombre: Modificar sensores
- Actor principal: Administrador
- Precondiciones: el usuario debe tener permisos de administrador
- Cumple requisitos: a1, a4, c1, f2.

A) Escenario principal:

- a) El usuario accede a la terminal que comunica con la aplicación.
- b) El usuario envía por línea de comando el cuerpo de código correspondiente a la modificación de los datos del sensor.
- c) El usuario lleva a cabo las modificaciones necesarias sobre los parámetros de medición o información devuelta por los sensores.
- d) El sistema almacena las modificaciones llevadas a cabo por el usuario en el histórico de la base de datos.
- e) El sistema devuelve una respuesta positiva a las modificaciones aplicadas.
- f) El caso de uso finaliza con éxito.

B) Escenarios de excepción:

- a) El usuario accede lleva a cabo las modificaciones oportunas sobre los sensores, pero el sistema no es capaz de almacenar las modificaciones en histórico de la base de datos.
 - 1) El sistema devuelve una respuesta negativa al usuario advirtiéndolo del fallo a la hora de almacenar.
 - 2) El sistema bloquea el cambio efectuado.
 - 3) El caso de uso finaliza sin éxito.

3. Caso de uso: Añadir un nuevo sensor al sistema

- Nombre: Añadir sensores
- Actor principal: Administrador
- Precondiciones: el usuario debe de tener permisos de administrador
- Cumple requisitos: a1, c1

A) Escenario principal:

- a) El usuario accede a la terminal que comunica con la aplicación.
- b) El usuario envía por línea de comando el cuerpo de código correspondiente a la creación de un nuevo sensor.
- c) El sistema almacena las modificaciones llevadas a cabo por el usuario en el histórico de la base de datos.
- d) El sistema devuelve una respuesta positiva al alta del nuevo sensor.
- e) El caso de uso finaliza con éxito.

B) Escenarios de excepción:

- a) El administrador lleva a cabo el alta del sensor, pero el sistema no es capaz de almacenar las modificaciones en histórico de la base de datos.
 - 1) El sistema devuelve una respuesta negativa advirtiendo del fallo a la hora de almacenar el cambio.
 - 2) El sistema bloquea el cambio efectuado.
 - 3) El caso de uso finaliza sin éxito.
- b) El sistema no reconoce el nuevo sensor añadido.
 - 1) El sistema devuelve una respuesta negativa avisando al usuario que no se ha podido añadir el sensor indicado.
 - 2) No se actualiza el listado del sistema con el nuevo sensor.
 - 3) El caso de uso finaliza sin éxito.

4. Caso de uso: Eliminar un sensor activo del sistema

- Nombre: Eliminar sensores
- Actor principal: Administrador
- Precondiciones: el usuario debe tener permisos de administrador
- Cumple requisitos: a2.

A) Escenario principal:

- a) El usuario accede a la terminal que conecta con la aplicación.
- b) El usuario envía por línea de comando el cuerpo de código correspondiente a la eliminación de un sensor existente.
- c) El sistema elimina dicho sensor de la lista de sensores activos.
- d) El sistema almacena las modificaciones llevadas a cabo por el usuario en el histórico de la base de datos.
- e) El sistema devuelve una respuesta positiva de la supresión del sensor.
- f) El caso de uso finaliza con éxito.

B) Escenarios de excepción:

- a) El administrador que accede lleva a cabo eliminación del sensor (o sensores deseados), pero el sistema no es capaz de almacenar las modificaciones en histórico de la base de datos.
 - 1) El sistema devuelve una respuesta negativa advirtiendo del fallo a la hora de almacenar el cambio.
 - 2) El sistema bloquea el cambio efectuado.
 - 3) El caso de uso finaliza sin éxito.
- b) El administrador intenta eliminar un sensor que no existe
 - 1) El sistema devuelve una respuesta negativa indicando que el sensor que el administrador quiere eliminar no existe.
 - 2) El sistema bloquea el cambio efectuado.
 - 3) El caso de uso finaliza sin éxito.

5. Caso de uso: Listar los sensores activos

- Nombre: Listar sensores
- Actor principal: Administrador, Técnico
- Precondiciones: el usuario debe permisos de administrador o de técnico.
- Cumple requisitos: a4, f2.

A) Escenario principal:

- a) El usuario accede a la terminal que conecta con la aplicación.
- b) El usuario envía por línea de comando el cuerpo de código correspondiente a la obtención del listado de los sensores existentes.
- c) El sistema devuelve la lista completa con los identificadores, localización y el estado de todos los sensores que se encuentran activos.
- d) El caso de uso finaliza con éxito.

B) Escenario de excepción:

- a) El sistema no puede conectar con la base de información de actuadores.
 - 1) El sistema devuelve una respuesta negativa alertando del fallo de conexión.
 - 2) El caso de uso finaliza sin éxito.

6. Caso de uso: Acceder a la pantalla de sensores

- Nombre: Pantalla de sensores
- Actor principal: Administrador, Político, Técnico e Instalador
- Precondiciones: el usuario debe haber hecho Login como administrador.
- Cumple requisitos: a5, c1, c2, f2.

A) Escenario principal:

- a) El usuario accede a la vista del sistema destinada a los sensores.
- b) El sistema muestra la información relativa a los sensores en cuanto al estado de funcionamiento, fallos, datos extraídos por cada uno de ellos, modificación y eliminación de los sensores, dependiendo del tipo de login efectuado.
- c) El usuario recupera la información deseada, dependiendo del tipo de login efectuado.
- d) El usuario vuelve a la pantalla de inicio o cierra la sesión.
- e) El caso de uso de pantalla de sensores finaliza con éxito.

7. Caso de uso: Modificar los actuadores activos

- Nombre: Modificar actuadores
- Actor principal: Administrador
- Precondiciones: el usuario debe tener permisos de administrador
- Cumple requisitos: b4, c1, f2.

A) Escenario principal:

- a) El usuario accede a la terminal que comunica con la aplicación.
- b) El usuario envía por línea de comando el cuerpo de código correspondiente a la modificación de los datos del actuador.
- c) El usuario lleva a cabo las modificaciones necesarias sobre los parámetros de medición o información devuelta por los actuadores.
- d) El sistema almacena las modificaciones llevadas a cabo por el usuario en el histórico de la base de datos.
- e) El sistema devuelve una respuesta positiva a las modificaciones aplicadas.
- f) El caso de uso finaliza con éxito.

B) Escenarios de excepción:

- a) El usuario accede lleva a cabo las modificaciones oportunas sobre los actuadores, pero el sistema no es capaz de almacenar las modificaciones en histórico de la base de datos.
 - 1) El sistema devuelve una respuesta negativa al usuario advirtiendo del fallo a la hora de almacenar el actuador.
 - 2) El sistema bloquea el cambio efectuado.
 - 3) El caso de uso finaliza sin éxito.

8. Caso de uso: Añadir un nuevo actuador al sistema

- Nombre: Añadir actuadores
- Actor principal: Administrador
- Precondiciones: el usuario debe permisos de administrador
- Cumple requisitos: b1, c1.

A) Escenario principal:

- a) El usuario accede a la terminal que comunica con la aplicación.
- b) El usuario envía por línea de comando el cuerpo de código correspondiente a la creación de un nuevo actuador.
- c) El sistema almacena las modificaciones llevadas a cabo por el usuario en el histórico de la base de datos.
- d) El sistema devuelve una respuesta positiva al alta del nuevo actuador.
- e) El caso de uso finaliza con éxito.

B) Escenarios de excepción:

- a) El administrador lleva a cabo el alta del actuador, pero el sistema no es capaz de almacenar las modificaciones en histórico de la base de datos.
 - 4) El sistema devuelve una respuesta negativa advirtiendo del fallo a la hora de almacenar el cambio.
 - 5) El sistema bloquea el cambio efectuado.
 - 6) El caso de uso finaliza sin éxito.

- b) El sistema no reconoce el nuevo actuador añadido.
 - 4) El sistema devuelve una respuesta negativa avisando al usuario que no se ha podido añadir el actuador indicado.
 - 5) No se actualiza el listado del sistema con el nuevo sensor.
 - 6) El caso de uso finaliza sin éxito.

9. Caso de uso: Eliminar un actuador activo del sistema

- Nombre: Eliminar actuadores
- Actor principal: Administrador
- Precondiciones: el usuario debe tener permisos de administrador
- Cumple requisitos: b2.

A) Escenario principal:

- a) El usuario accede a la terminal que conecta con la aplicación.
- b) El usuario envía por línea de comando el cuerpo de código correspondiente a la eliminación de un actuador existente.
- c) El sistema elimina dicho actuador de la lista de sensores activos.
- d) El sistema almacena las modificaciones llevadas a cabo por el usuario en el histórico de la base de datos.
- e) El sistema devuelve una respuesta positiva de la supresión del actuador.
- f) El caso de uso finaliza con éxito.

B) Escenarios de excepción:

- a) El administrador que accede lleva a cabo eliminación del actuador (o actuadores deseados), pero el sistema no es capaz de almacenar las modificaciones en histórico de la base de datos.
 - 1) El sistema devuelve una respuesta negativa advirtiendo del fallo a la hora de almacenar el cambio.
 - 2) El sistema bloquea el cambio efectuado.
 - 3) El caso de uso finaliza sin éxito.
- b) El administrador intenta eliminar un actuador que no existe
 - 1) El sistema devuelve una respuesta negativa indicando que el actuador que el administrador quiere eliminar no existe.
 - 2) El sistema bloquea el cambio efectuado.
 - 3) El caso de uso finaliza sin éxito.

10. Caso de uso: Listar los actuadores activos

- Nombre: Listar actuadores
- Actor principal: Administrador, Técnico
- Precondiciones: el usuario debe de haber tener permisos de administrador o de técnico.
- Cumple requisitos: b3, f2.



- A) Escenario principal:
- a) El usuario accede a la terminal que conecta con la aplicación.
 - b) El usuario envía por línea de comando el cuerpo de código correspondiente a la obtención del listado de los actuadores existentes.
 - c) El sistema devuelve la lista completa con los identificadores, localización y el estado de todos los actuadores que se encuentran activos.
 - d) El caso de uso finaliza con éxito.
- B) Escenario de excepción:
- b) El sistema no puede conectar con la base de información de actuadores.
 - 1) El sistema devuelve un mensaje de error alertando del fallo de conexión.
 - 2) El caso de uso finaliza sin éxito.

11. Caso de uso: Acceder a la pantalla de actuadores

- Nombre: Pantalla de actuadores
- Actor principal: Administrador, Técnico
- Precondiciones: el usuario debe permisos de administrador o de técnico.
- Cumple requisitos: b5, c1, c2, f2.

- A) Escenario principal:
- a) El usuario accede a la vista del sistema destinada a los actuadores.
 - b) El sistema muestra la información relativa a los actuadores en cuanto al estado de funcionamiento y fallos.
 - c) El usuario realiza las tareas de mantenimiento oportunas dependiendo de sus permisos.
 - d) El sistema manda la orden de ejecución a los actuadores correspondientes y almacena el nuevo estado y las órdenes enviadas en el histórico de acciones de la base de datos.
 - e) El usuario vuelve a la pantalla de inicio o cierra sesión.
 - f) El caso de uso finaliza con éxito.
- B) Escenario de excepción:
- a) El usuario accede a la pantalla destinada a los actuadores.
 - b) Después de listar la información relativa a los actuadores disponibles, el actuador no puede recibir la acción a ejecutar debido a una caída de la red o fallo en la comunicación con el actuador.
 - c) El sistema bloquea la acción ejecutada por el usuario y devuelve un mensaje de error por pantalla y avisa al personal correspondiente para la solución del problema existente.
 - d) El caso de uso finaliza sin éxito.

12. Caso de uso: Listar los usuarios conectados

- Nombre: Listar usuarios conectados
- Actor principal: Administrador
- Precondiciones: el usuario debe tener permisos de administrador.
- Cumple requisitos: d1, d5, d6, f1.

A) Escenario principal:

- a) El usuario accede a la base de datos de usuarios.
- b) El usuario consulta con el sistema el listado de usuarios conectados a tiempo real.
- c) El sistema devuelve un listado con todos los usuarios conectados en ese momento, independientemente del tipo de credenciales con el que dichos usuarios existentes se hayan conectado.
- d) El caso de uso finaliza con éxito.

B) Escenario de excepción:

- a) El sistema no puede conectar con la base de información de usuarios.
 - 1) El sistema devuelve un mensaje de error alertando del fallo de conexión.
 - 2) El caso de uso finaliza sin éxito.

13. Caso de uso: Eliminar un usuario existente

- Nombre: Eliminar usuario
- Actor principal: Administrador
- Precondiciones: el usuario debe tener permisos de administrador.
- Cumple requisitos: d3, f1.

A) Escenario principal:

- a) El usuario accede a la base de datos de usuarios.
- b) El usuario busca mediante una consulta a base de datos el ID del usuario a eliminar.
- c) El usuario lanza la supresión del usuario en la base de datos.
- d) El sistema devuelve un mensaje de confirmación.
- e) El caso de uso finaliza con éxito.

B) Escenario de excepción:

- a) El usuario introducido no existe.
 - 1) El sistema devuelve un mensaje de error advirtiendo que no existe el usuario con ID introducido.
 - 2) El caso de uso finaliza sin éxito.
- b) El sistema no puede conectar con la base de datos para eliminar el usuario con ID indicado.
 - 1) El sistema devuelve un mensaje de error indicando el problema de conexión.



- 2) El sistema bloquea la acción de eliminar.
 - 3) El caso de uso finaliza sin éxito.
- c) El usuario conectado introduce su mismo ID para ser eliminado.
- 1) El sistema comprueba que el ID que se desea eliminar es el mismo que está activo en la sesión de gestión de usuarios.
 - 2) El sistema devuelve un mensaje de error indicando que el propio usuario no se puede eliminar a sí mismo.
 - 3) El sistema bloquea la acción de eliminación.
 - 4) El usuario vuelve a la pantalla de gestión de usuarios.
 - 5) El caso de uso finaliza sin éxito.

14. Caso de uso: Añadir un nuevo usuario

- Nombre: Añadir usuario
- Actor principal: Administrador
- Precondiciones: el usuario debe tener credenciales de administrador.
- Cumple requisitos: d2, f1.

A) Escenario principal:

- a) El usuario accede a la base de datos de usuarios.
- b) El usuario envía la consulta a base de datos con el ID del usuario que desea añadir.
- c) El usuario introduce el rol del usuario nuevo que se va a crear.
- d) El sistema añade de la base de datos de usuarios activos el usuario con el ID introducida junto con su rol.
- e) El sistema devuelve un mensaje de confirmación.
- f) El caso de uso finaliza con éxito.

B) Escenario de excepción:

- a) El usuario introducido existe.
 - 1) El sistema devuelve un mensaje de error advirtiéndole que ya existe el usuario con ID introducido.
 - 2) El caso de uso finaliza sin éxito.
- b) El sistema no puede conectar con la base de datos para añadir el usuario con ID indicado.
 - 1) El sistema devuelve un mensaje de error indicando el problema de conexión.
 - 2) El sistema bloquea la acción de añadir.
 - 3) El usuario vuelve a la pantalla de gestión de usuarios.
 - 4) El caso de uso finaliza sin éxito.

15. Caso de uso: Actualizar un usuario existente

- Nombre: Actualizar usuario
- Actor principal: Administrador
- Precondiciones: el usuario debe tener credenciales de administrador.
- Cumple requisitos: d4, f1.

A) Escenario principal:

- a) El usuario accede a la base de datos de usuarios.
- b) El usuario consulta en base de datos el ID del usuario que desea actualizar.
- c) El usuario actualiza por consulta el rol del usuario con el ID deseado.
- d) El sistema actualiza en la base de datos de usuarios activos el usuario con el ID introducida junto con su nuevo rol.
- e) El sistema devuelve un mensaje de confirmación.
- f) El caso de uso finaliza con éxito.

B) Escenario de excepción:

- a) El usuario introducido no existe.
 - 1) El sistema devuelve un mensaje de error advirtiéndole que no existe el usuario con ID introducido.
 - 2) El caso de uso finaliza sin éxito.
- b) El sistema no puede conectar con la base de datos para actualizar el usuario con ID indicado.
 - 1) El sistema devuelve un mensaje de error indicando el problema de conexión.
 - 2) El sistema bloquea la acción de actualizar.
 - 3) El caso de uso finaliza sin éxito.

16. Caso de uso: Acceder a la pantalla de gestión de usuarios

- Nombre: Pantalla de gestión de usuarios
- Actor principal: Administrador
- Precondiciones: el usuario debe tener credenciales de administrador.
- Cumple requisitos: d5, d6, f1.

A) Escenario principal:

- a) El usuario accede a la pantalla destinada a la gestión de usuarios.
- b) El sistema muestra la información relativa a los usuarios presentes en el sistema, junto con sus roles y su actividad reciente.
- c) El usuario realiza las tareas de mantenimiento oportunas.
- d) El sistema persiste la orden correspondiente y almacena los cambios efectuados en el histórico de acciones de la base de datos.
- e) El usuario vuelve a la pantalla de inicio o cierra sesión.
- f) El caso de uso finaliza con éxito.



B) Escenario de excepción:

- a) El usuario accede a la pantalla destinada a la gestión de usuarios.
 - 1) Después de listar la información relativa a los usuarios disponibles, el sistema no puede realizar la acción a ejecutar debido a una caída de la red.
 - 2) El sistema bloquea la acción ejecutada por el usuario y devuelve un mensaje de error por pantalla y avisa al personal correspondiente para la solución del problema existente.
 - 3) El caso de uso finaliza sin éxito.

17. Caso de uso: Acceder a la pantalla de información genérica

- Nombre: Pantalla genérica de información
- Actor principal: Ciudadano
- Precondiciones:
- Cumple requisitos: f2.

A) Escenario principal:

- a) El usuario accede a la pantalla genérica de información.
- b) El sistema muestra la información relativa a la temperatura actual en los diferentes recintos en los cuales se dispone de sensores, así como el aforo de las instalaciones correspondientes.
- c) El usuario vuelve a la pantalla de inicio o cierra sesión.
- d) El caso de uso finaliza con éxito.

B) Escenario de excepción:

- a) El usuario no puede acceder a la pantalla destinada a la información genérica debido a un problema de caída de la conexión.
 - 1) Se devuelve un mensaje de error de conexión a la página con la información correspondiente.
 - 2) Se envía un aviso al personal indicado para la posterior resolución del problema.
 - 3) El caso de uso finaliza sin éxito.

5 Implementación e implantación

5.1 Introducción

En el desarrollo del proyecto, nos enfrentamos al desafío de replicar un escenario real complejo que involucraba 40 sensores y 2 actuadores en dos ubicaciones distintas de Vilamarxant. Sin embargo, debido a restricciones presupuestarias y la falta de información completa, decidimos implementar un entorno de prueba más manejable utilizando solo 4 sensores y 4 actuadores, mientras manteníamos las dos ubicaciones clave.

La decisión de optar por un entorno de prueba reducido se basó en varios factores cruciales. En primer lugar, la implementación completa del escenario real exclusivamente para este proyecto queda fuera del coste estimado en términos de recursos financieros y tiempo. Además, dado que nuestro proyecto se centraba en una implementación para un ayuntamiento real, no teníamos acceso a toda la información necesaria para replicar fielmente el entorno completo.

El entorno de prueba que diseñamos se centró en capturar la esencia del escenario real al mantener las dos ubicaciones y, al mismo tiempo, garantizar una implementación manejable y eficiente. Los 4 sensores y 4 actuadores seleccionados fueron cuidadosamente elegidos para representar una muestra representativa de la funcionalidad del sistema, permitiendo evaluar y validar las características principales de la aplicación.

Aunque el entorno de prueba era más pequeño en escala, nos aseguramos de que la aplicación pudiera interactuar de manera significativa con los sensores y actuadores disponibles. Esto nos permitió realizar pruebas exhaustivas de las funcionalidades críticas y validar el rendimiento de la aplicación en condiciones controladas.

A pesar de las limitaciones en el tamaño del entorno de prueba, nuestra metodología nos permitió avanzar de manera eficiente en el desarrollo de la aplicación, identificar posibles problemas y realizar ajustes necesarios. En última instancia, esta aproximación nos brindó la flexibilidad necesaria para trabajar dentro de las restricciones del proyecto ficticio mientras manteníamos un enfoque pragmático y efectivo hacia la implementación del sistema.

5.2 Rutinas de configuración de código

A continuación, ilustraremos en primera instancia las rutinas relacionadas con el código que hemos creado para poder lanzar y levantar los servicios que necesitaremos en nuestra aplicación.

5.2.1 Creación del fichero `docker-compose.yml`

En primer lugar, deberemos definir la estructura de la aplicación que implementaremos. De esto se encargará el fichero `docker-compose.yml`, el cual contendrá toda la información e implementación de los contenedores con las imágenes necesarias para el correcto funcionamiento de dicha aplicación.

Estas imágenes pueden separarse por grupos (los cuales retomaremos a continuación). Estos grupos incluyen la parte de bases de datos, front-end, autenticación y orion entre otros.

Primero, dispondremos de tres bases de datos diferentes las cuales nos permitirán almacenar las distintas informaciones generadas y necesitadas por la aplicación. Es por esto por lo que distinguiremos las bases de datos relacionales de las no relacionales.

Tendremos una base de datos no relacional Mongo DB. Dicha base de datos nos permitirá almacenar los sensores ya que éstos son creados en formato JSON para poder indicar los diferentes atributos que los componen tal como la temperatura, unidad de medida de la temperatura, nombre y localización del sensor, etc...

```
mongo-db:
  image: mongo:latest
  hostname: mongo-db
  container_name: db-mongo
  ports:
  - "27017:27017" # localhost:27017
  networks:
  - my-network
  volumes:
  - mongo-db:/data
  healthcheck:
    test: |
      host=`hostname --ip-address || echo '127.0.0.1'`;
      mongo --quiet $host/test --eval 'quit(db.runCommand({ ping: 1 }).ok ? 0 : 2)' && echo 0 || echo 1
    interval: 5s
```

Ilustración 9 - Definición BBDD no relacional Mongo DB

En este bloque cabe destacar el puerto 27017 que nos permitirá escuchar la información enviada con las localizaciones y los sensores que se necesitarán. Igualmente, la red utilizada “network” que corresponde en este caso a una red local.

Después, utilizaremos dos bases de datos relacionales. La primera una base de datos Postgres, la cual contendrá la información requerida por la parte del front-end (Wirecloud), tal como la URL en la cual se conectarán los sensores para transmitir los datos recogidos a tiempo real, los parámetros de los widgets que mostrarán la información necesaria, etc...

```
postgres-db:
  image: postgres
  hostname: postgres-db
  container_name: db-postgres
  restart: always
  ports:
  - "5432:5432"
  networks:
  - my-network
  environment:
  - POSTGRES_PASSWORD=wirepass
  volumes:
  - postgres-db:/var/lib/postgresql/data
```

Ilustración 10 - Definición BBDD relacional Postgres

De nuevo destacaremos el puerto de conexión 5432 para la escucha, así como la red de conexión que será la misma red local utilizada por Mongo DB, y la variable “hostname” que nos permitirá más adelante conectarnos a esta base de datos.

La segunda base de datos relacional será una base de datos MySQL.

Utilizaremos esta base de datos para el almacenamiento de las credenciales de los usuarios dados de alta en la aplicación (email de registro, nombre de usuario y contraseña). Dicha base de datos se utilizará por el módulo Keyrock para poder gestionar la autenticación de dichos usuarios.

```
mysql-db:
  restart: always
  image: mysql:latest
  hostname: mysql-db
  container_name: db-mysql
  expose:
    - "3306"
  ports:
    - "3306:3306" # localhost:3306
  networks:
    - my-network
  environment:
    - "MYSQL_ROOT_PASSWORD_FILE=/run/secrets/my_secret_data"
    - "MYSQL_ROOT_HOST=172.18.1.5" # Allow Keyrock to access this database
  volumes:
    - mysql-db:/var/lib/mysql
    - ./mysql-data:/docker-entrypoint-initdb.d/:ro # Preload Keyrock Users
  secrets:
    - my_secret_data
```

Ilustración 11 - Definición BBDD relacional Mysql

En esta ocasión, varios parámetros a resaltar: el puerto de escucha 3306 y la misma red local que la utilizada anteriormente. Pero aquí cabe resaltar los atributos de la variable “environment” la cual nos permitirá definir el lugar en el que se encuentra la clave para poder desbloquear la base de datos (MYSQL_ROOT_PASSWORD_FILE), así como la URL a la cual se conectará al módulo Keyrock para poder retomar los datos de los usuarios (MYSQL_ROOT_HOST).

En segundo lugar, detallaremos la parte del front-end. Esta viene definida por el módulo Wirecloud. Wirecloud permite crear una interfaz adaptada al usuario que, a través de varios widgets específicos (graficos, mapas, ...), permitirán mostrar la información requerida por la aplicación y enviada por los sensores que se hayan creado.

```
wirecloud:
  image: fiware/wirecloud:latest
  container_name: fiware-wirecloud
  hostname: wirecloud
  ports:
    - "8000:8000" # localhost:8000
  networks:
    - my-network
  restart: always
  depends_on:
    - keyrock
    - elasticsearch
    - memcached
    - postgres-db
  environment:
    - DEBUG=True
    - DEFAULT_THEME=wirecloud.defaulttheme
    - DB_HOST=postgres-db
    - DB_PASSWORD=wirepass
    - FORWARDED_ALLOW_IPS=*
    - ELASTICSEARCH2_URL=http://elasticsearch:9200/
    - MEMCACHED_LOCATION=memcached:11211
    - FIWARE_IDM_PUBLIC_URL=http://localhost:3005
    - FIWARE_IDM_SERVER=http://172.18.1.5:3005
    - SOCIAL_AUTH_FIWARE_KEY=wirecloud-dckr-site-0000-000000000000
    - SOCIAL_AUTH_FIWARE_SECRET=wirecloud-docker-000000-clientsecret
  volumes:
    - wirecloud-data:/opt/wirecloud_instance/data
    - wirecloud-static:/var/www/static
```

Ilustración 12 - Definición interfaz Wirecloud

En este caso, además del puerto 8000 y de la misma red local utilizada en los módulos anteriores, resaltaremos de nuevo los atributos de la variable “environment”. Cuatro atributos son especialmente importantes aquí: `FIWARE_IDM_PUBLIC_URL` que nos permitirá definir la URL pública en la cual se llevará a cabo la conexión con el módulo Keyrock para la autenticación de los datos introducidos en la pantalla de login de Wirecloud, y `FIWARE_IDM_SERVER` que nos permitirá determinar la URL del servidor de autenticación al cual gestionará dicha autenticación. De la misma forma, en la variable `DB_HOST` indicaremos el host de la base de datos Postgres que hemos creado y detallado anteriormente.

En tercer lugar, tenemos el módulo Keyrock. Keyrock es el componente encargado de gestionar la parte de la autenticación de la aplicación, a partir de los usuarios que se hayan creado y de los que hagan el login en la pantalla de inicio del front-end. En función del usuario con el que nos autentiquemos en la aplicación, tendremos acceso a unos datos y a unos permisos u otros.

```

keyrock:
  image: fiware/idm:latest
  container_name: fiware-keyrock
  hostname: keyrock
  networks:
    - my-network
  depends_on:
    - mysql-db
  ports:
    - "3005:3005" # localhost:3005
    - "3443:3443" # localhost:3443
  environment:
    - DEBUG=idm:*
    - IDM_DB_HOST=mysql-db
    - IDM_DB_PASS_FILE=/run/secrets/my_secret_data
    - IDM_DB_USER=root
    - IDM_HOST=http://localhost:3005
    - IDM_PORT=3005
    - IDM_ADMIN_USER=alice
    - IDM_ADMIN_EMAIL=alice-the-admin@test.com
    - IDM_ADMIN_PASS=test
  secrets:
    - my_secret_data
  healthcheck:
    interval: 5s

```

Ilustración 13 - Definición módulo de autenticación Keyrock

Aquí indicaremos a través del atributo “IDM_HOST” la URL pública a la cual se conectará Keyrock para poder llevar a cabo a autenticación. Será a esta URL a la cual será redirigido el usuario una vez introducidas las credenciales para que puedan ser validadas por el módulo. Indicamos de la misma forma a través de los atributos IDM_ADMIN_USER, IDM_ADMIN_EMAIL y IDM_ADMIN_PASS las credenciales del administrador de la plataforma. La red utilizada es la red local que hemos definido anteriormente, y en este caso Keyrock escuchará tanto en el puerto 3005 como en el 3443.

Por último, resaltaremos uno de los componentes más importantes llamado Orion. Orion es un context bróker que nos permite interconectar los diferentes módulos que hemos definido anteriormente, para que éstos puedan trabajar de forma conjunta. Es el responsable de tener una aplicación funcional ya que sino los módulos no serían capaces de procesar la información entre ellos correctamente.

```

orion:
  image: fiware/orion:latest
  container_name: fiware-orion
  depends_on:
    - mongo-db
  networks:
    - my-network
  expose:
    - "1026"
  ports:
    - "1026:1026" # localhost:1026
  command: -dbhost mongo-db -logLevel DEBUG
  healthcheck:
    test: curl --fail -s http://orion:1026/version || exit 1
    interval: 5s

```

Ilustración 14 - Definición context broker Orion

Orion escuchará en el puerto 1026 y se conectará a la red local utilizada en los módulos anteriores también.

Una vez el fichero docker-compose.yml está creado, podremos pasar a la siguiente rutina que nos permitirá crear los contenedores con las imágenes correspondientes y levantar el servicio con la aplicación.

5.2.2 Lanzamiento los servicios

Para poder lanzar los servicios necesarios, nos situaremos en el terminal en la carpeta contenedora del proyecto e introduciremos el comando “./services start”.

Este comando lanzará el fichero “services” que se encargará de, a partir del fichero docker-compose.yml definido anteriormente, atribuir los puertos correspondientes a las variables declaradas, inicializar las bases de datos con los parámetros correspondientes y mostrar el proceso por pantalla para mayor facilidad para el usuario.

Detallaremos dicho proceso a continuación:

```
(base) MBP-M:AppSMvilla marc$ ./services start
Removing tutorial networks
00fa0fb11fb3
Starting containers: Orion, IoT-Agent, Keyrock, Wirecloud
Along with the following databases: MongoDB, PostGres and MySQL
and microservices: ElasticSearch, MemCache and NGSI Proxy

- Orion is the context broker
- IoT-Agent is configured for the UltraLight Protocol
- Keyrock is an Identity Management Front-End
- Wirecloud is a Dashboard Utility
- NGSI Proxy creates a server that is capable of redirecting Orion notifications to web pages

WARN[0000] The "host" variable is not set. Defaulting to a blank string.
[+] Running 0/42
.: mongo-db Pulling
.: elasticsearch Pulling
.: ngsi-proxy Pulling
.: 6ae821421a7d Waiting
.: f0d1c6739de6 Waiting
.: bc4bc29a22 Waiting
.: 8c582ea70b4e Waiting
.: 0b3591965f5d Waiting
.: mysql-db Pulling
.: wirecloud Pulling
.: 9b99af5931b3 Pulling fs layer
.: b6013b3e77fe Pulling fs layer
.: bbced17b6899 Pulling fs layer
[+] Running 0/608 Waiting
.: mongo-db Pulling
.: elasticsearch Pulling
.: 05d1a5232b46 Waiting
```

Ilustración 15 - Lanzamiento de los servicios necesarios para la aplicación

La primera vez que se lance el comando, se llevará a cabo la descarga de las imágenes contenidas en el fichero docker-compose antes de poder hacer el pull de las mismas. Por ende, en el primer lanzamiento, el tiempo de que tomará la finalización del proceso será mayor que el resto de los lanzamientos una vez las imágenes estén descargadas ya.

Una vez descargadas todas, se crean los contenedores e inicializan los volúmenes con dichas imágenes:

```
[+] Running 18/18
  :: Network fiware_default          Created
  :: Volume "fiware_elasticsearch-data" Created
  :: Volume "fiware_mysql-db"       Created
  :: Volume "fiware_wirecloud-data"  Created
  :: Volume "fiware_wirecloud-static" Created
  :: Volume "fiware_postgres-db"     Created
  :: Volume "fiware_mongo-db"        Created
  :: Container wc-elasticsearch       Started
  :: Container wc-memcached           Started
  :: Container db-mongo               Started
  :: Container wc-ngsi-proxy          Started
  :: Container db-mysql               Started
  :: Container db-postgres            Started
  :: Container fiware-keyrock         Started
  :: Container fiware-orion           Started
  :: Container fiware-wirecloud       Started
  :: Container fiware-iot-agent       Started
  :: Container fiware-tutorial        Started
```

Ilustración 16 - Creación e inicio de los servicios

Del lado de dockers, la instalación está completa ya. Ahora deberemos comprobar que los servicios se han levantado correctamente, y para ello el fichero “services” detalla el estado HTTP de los diferentes servicios:

```
⌚ Waiting for MongoDB to be available
Adding appropriate MongoDB indexes for Orion ... done
Adding appropriate MongoDB indexes for IoT-Agent ... done
⌚ Waiting for Orion to be available
Context Broker HTTP state: 200 (waiting for 200)
Unable to find image 'curlimages/curl:latest' locally
latest: Pulling from curlimages/curl
df9b9388f04a: Pull complete
ffaca4cc7d5a: Pull complete
17b4f9600428: Pull complete
b07eadfb9305: Pull complete
4ad5b64acddc: Pull complete
b6a21241ef17: Pull complete
e02c9f2ce612: Pull complete
8e26c9329104: Pull complete
1000da985f9b: Pull complete
a4bd3c08ea7a: Pull complete
a32f5307f213: Pull complete
Digest: sha256:9fab1b73f45e06df9506d947616062d7e8319009257d3a05d970b0de80a41ec5
Status: Downloaded newer image for curlimages/curl:latest
⌚ Loading context data done
⌚ Waiting for IoT-Agent to be available
⌚ Provisioning IoT devices done
⌚ Waiting for Wirecloud to be available
Wirecloud HTTP state: 000 (waiting for 200)
done
⌚ Waiting for Keyrock to be available
done
You can now start the application App VillamarXant
```

Ilustración 17 - Confirmación de levantamiento correcto de los servicios

Una vez las respuestas son correctas (como se muestra en la imagen anterior), los servicios están levantados y los propios mensajes por pantalla nos indica que podemos empezar a utilizar la aplicación.

5.2.3 Añadir los sensores y sus localizaciones

El siguiente paso que iniciaremos será la incorporación de los sensores que van a llevar a cabo las mediciones correspondientes, las cuales serán recogidas y analizadas más adelante a través de la interfaz gráfica.

Para ello, utilizaremos el envío de un mensaje JSON a través de un POST que recogerá los atributos necesarios para poder identificar cada uno de los sensores.

El POST se hará a la url `http://localhost:1026/v2/entities` que corresponde a la url de Orion. Dicho módulo se encargará de reenviar la información a la base de datos MongoDB, donde retomaremos los datos.

El cuerpo del mensaje JSON contendrá el ID que identifica el sensor, un atributo Integer que recoge la temperatura, un Float que recoge la humedad y finalmente un String que identifica el lugar del sensor.

Enviaremos los lugares de cada uno de los sensores en un nuevo POST de un mensaje JSON a la misma url definida anteriormente.

Este mensaje JSON contendrá un ID que identificará el lugar, un atributo Address que contendrá la calle, la localidad, la región y el código postal del sensor correspondiente, así como las coordenadas precisas del lugar.

De esta forma podremos relacionar cada uno de los sensores existentes con su lugar.

El cuerpo del JSON que enviaremos será como el que se muestra a continuación, realizando las modificaciones necesarias para adaptar las coordenadas, el ID y el nombre de la dirección:

```
{
  "id": "ID_lugar_1",
  "type": "Location",
  "address": {
    "type": "PostalAddress",
    "value": {
      "streetAddress": "Carrer Don Lluís Santàngel",
      "addressRegion": "Vilamarxant",
      "addressLocality": "Valencia",
      "postalCode": "46191"
    }
  },
  "location": {
    "type": "geo:json",
    "value": {
      "type": "Point",
      "coordinates": [39.57085, -0.62089]
    }
  },
  "name": {
    "type": "Text",
    "value": "Localización del sensor1"
  }
}
```

Ilustración 18 - JSON definición de los edificios municipales

De la misma forma, comprobaremos que los datos de los sensores han sido cargados correctamente comprobando el resultado del POST tanto en la base de datos:

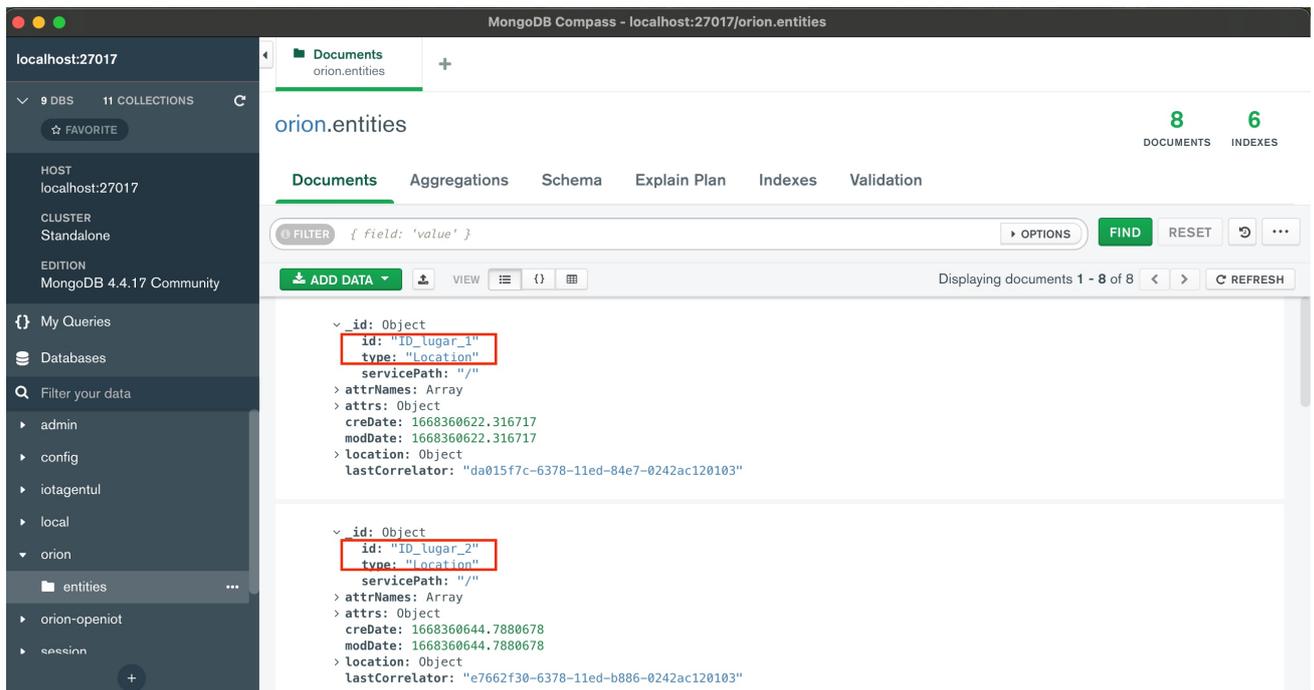


Ilustración 19 - Creación de los lugares en BBDD

Como en la url de Orion “http://localhost:1026/v2/entities”:

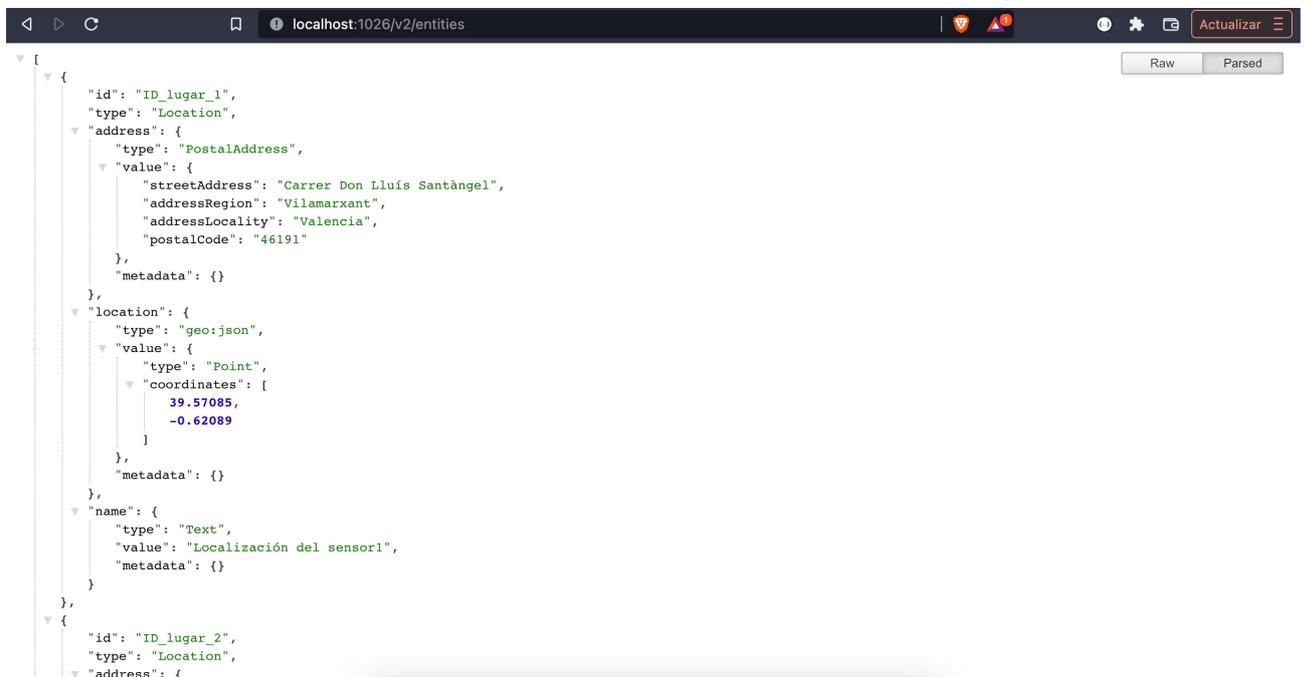


Ilustración 20 - Verificación de los lugares en Orion

De esta forma confirmaremos que, tanto la base de datos como Orion han recibido correctamente la información que hemos transmitido a través del POST de los mensajes JSON.

5.3 Rutinas de configuración gráfica

En el apartado anterior hemos detallado el código necesario para poder en marcha la aplicación. En este apartado, detallaremos los pasos a seguir para poder terminar la configuración de la interfaz gráfica, que será a la cual el usuario final tendrá acceso.

5.3.1 Autenticación y conexión a la aplicación

Una vez los módulos han sido correctamente descargados y lanzados podemos lanzar el módulo de Wirecloud que cargará la interfaz gráfica de la aplicación.

La primera página que veremos será la que nos permita iniciar sesión con las credenciales correspondientes. Ésta corresponde a la “landing page”:

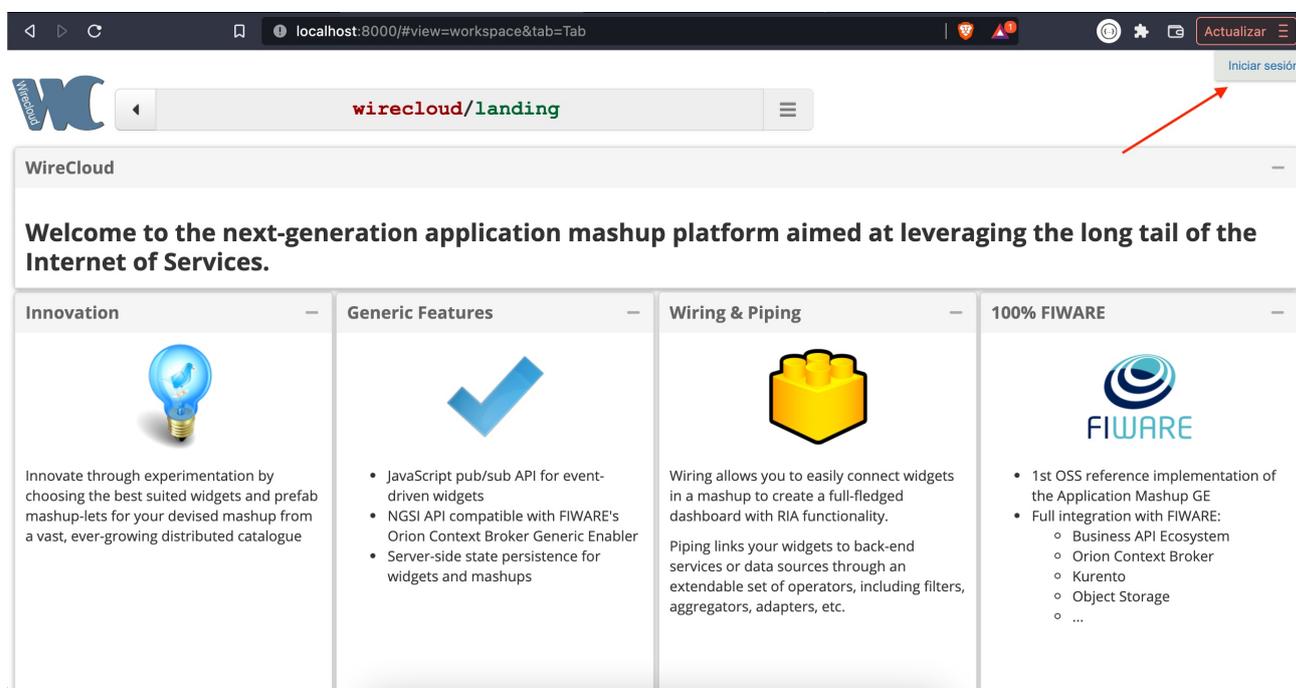


Ilustración 21 - Pantalla de carga Wirecloud

Haciendo click en el icono de inicio de sesión, seremos redirigidos a la interfaz de autenticación. Dicha interfaz conecta el módulo Wirecloud con Keyrock y valida contra la base de datos de usuarios las credenciales introducidas:

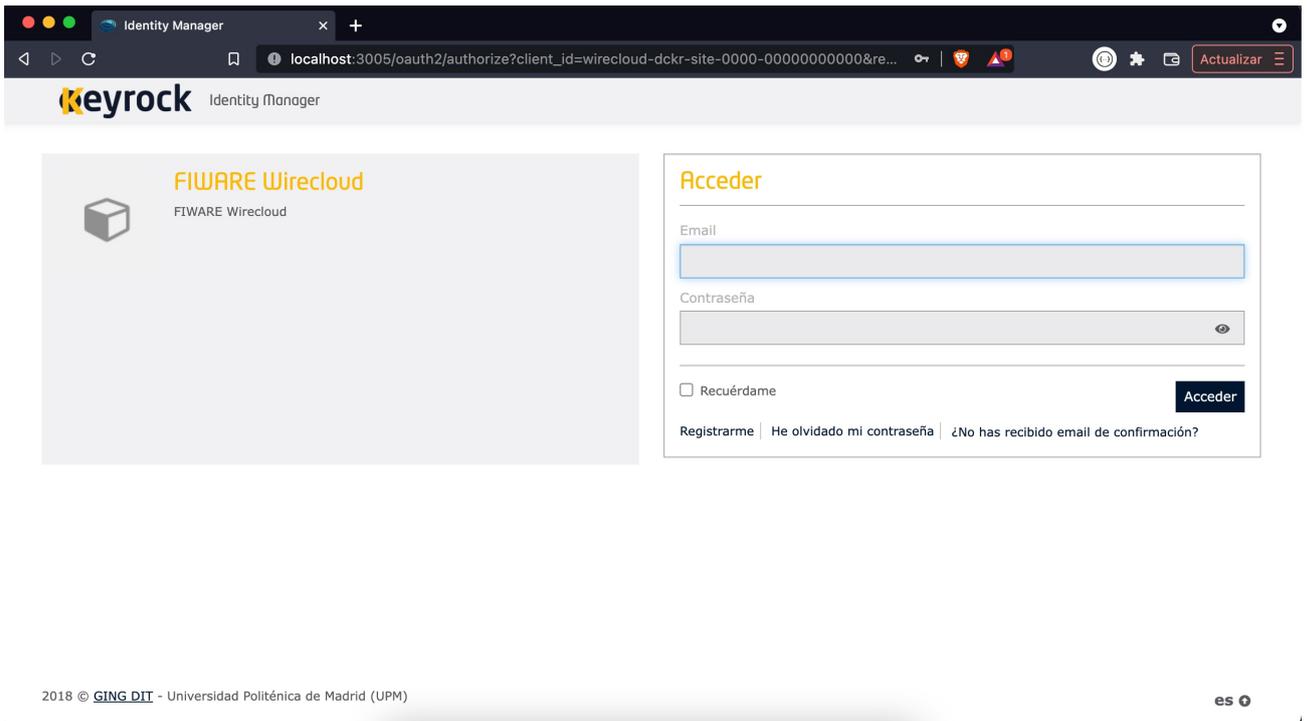


Ilustración 22 - Pantalla de autenticación Keyrock

En cuanto se hayan introducido credenciales válidas, el usuario será redirigido a la pantalla de inicio de la aplicación.

Dicha pantalla se distribuye de la siguiente manera:

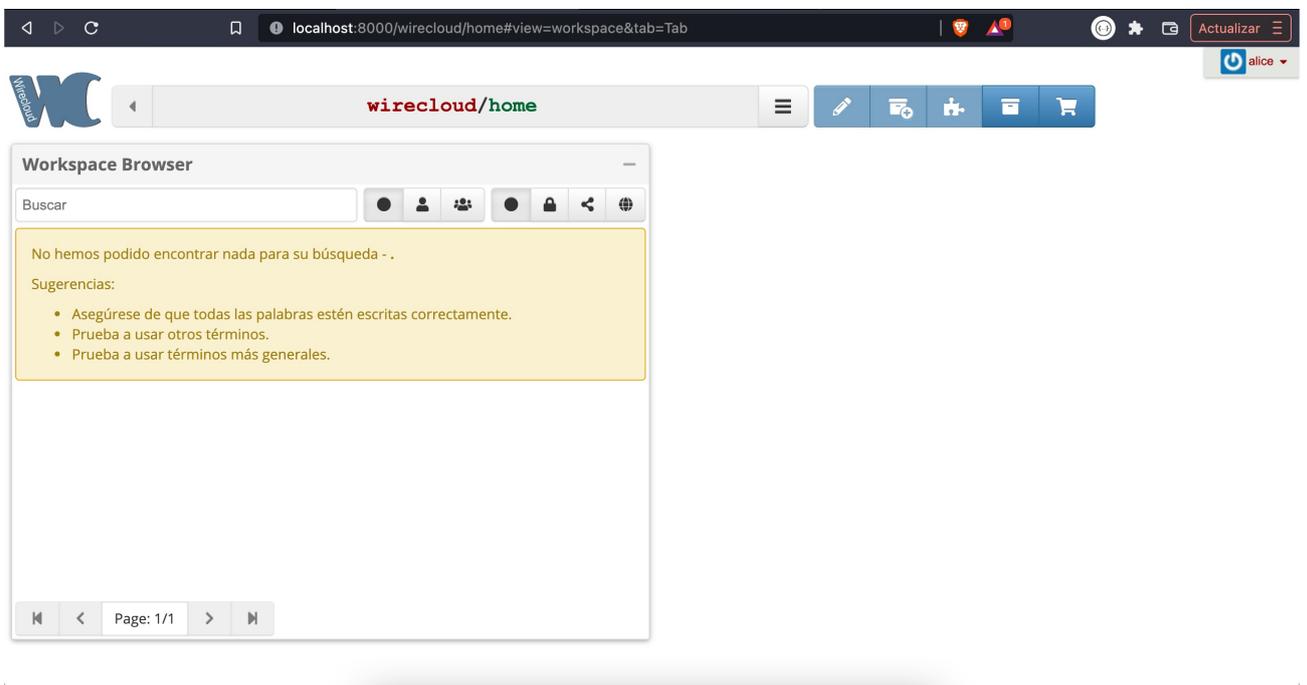


Ilustración 23 - Pantalla de inicio Wirecloud

Esta pantalla de inicio nos mostrará los espacios de trabajo existentes (de momento estará vacío ya que no existe ninguno), así como los diferentes iconos que nos permitirán gestionar

los widgets que instalaremos para la gestión de nuestra aplicación. Trataremos el proceso de configuración en el apartado a continuación.

5.3.2 Instalación y configuración del módulo Wirecloud

Ya hemos lanzado y levantado todos los servicios necesarios y nos hemos autenticado en la aplicación. Ahora, añadiremos los componentes que necesitaremos para poder visualizar tanto los sensores que existen como los valores que recogen.

Para ello, crearemos primero un nuevo entorno de trabajo:



Ilustración 24 - Creación de un entorno de trabajo Wirecloud

Llamaremos a este nuevo entorno “App Villamarxant”.

En segundo lugar, configuraremos los componentes que vamos a necesitar en nuestra aplicación para poder ilustrar gráficamente tanto los sensores que hemos creado como los valores que han sido recogidos por los mismos.

Para ello, añadiremos primero un componente de tipo NGSI Browser, el cual nos permitirá obtener una lista con los sensores que hemos creado y añadido a la base de datos:

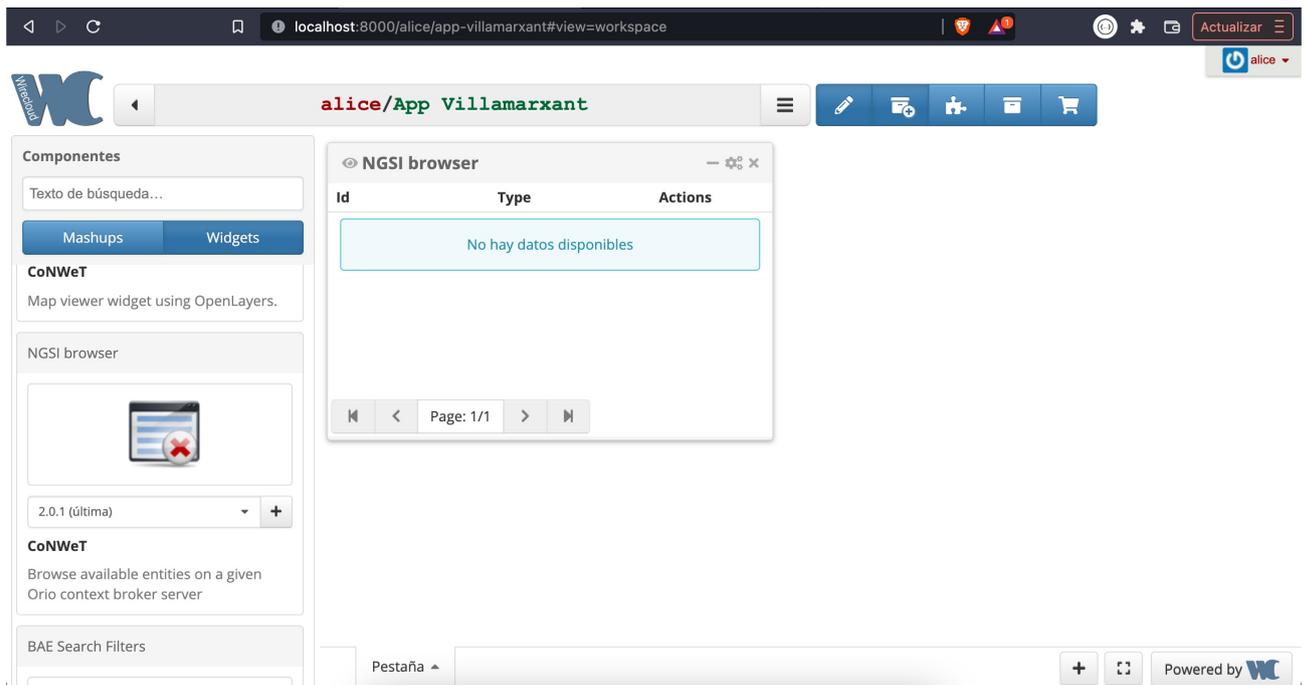


Ilustración 25 - Añadir componente NGSI browser para Sensores y Actuadores

Después, añadiremos el componente Mapa necesario para poder visualizar los puntos de los sensores en el mapa:

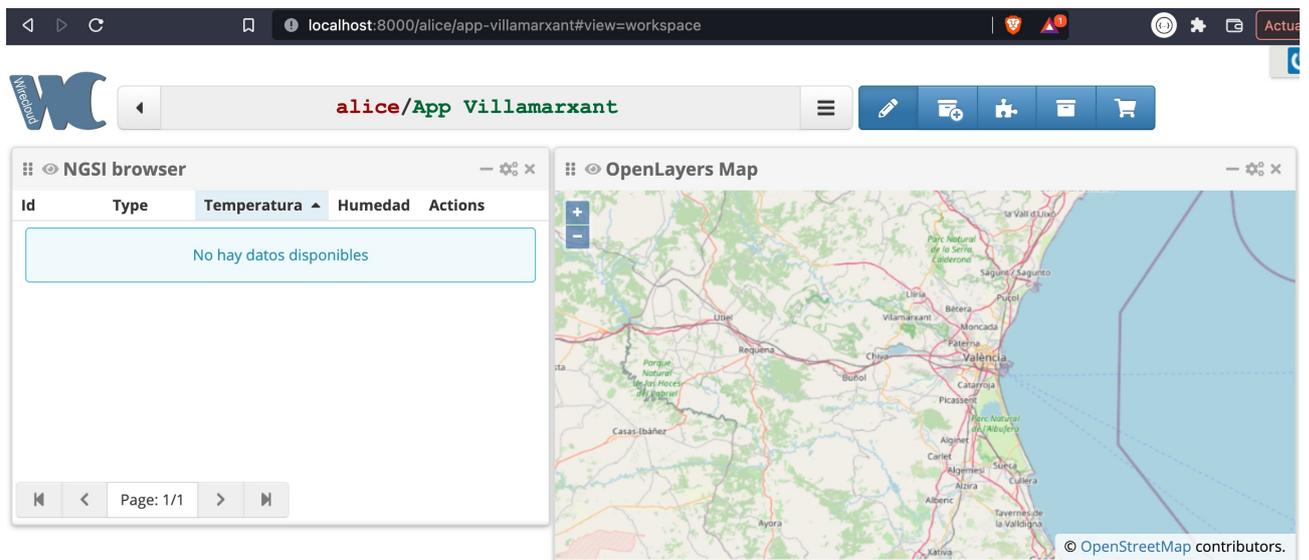


Ilustración 26 - Añadir componente mapa para las localizaciones

En cuarto lugar, conectaremos todos los componentes para que puedan mostrar los valores correctos que hemos enviado a la base de datos. Esto se hace desde el apartado de Wiring de la interfaz gráfica, y permite definir la configuración de cada uno de los componentes, así como la relación que presentan entre cada uno de ellos.

La conexión que hemos implementado entre los componentes cargados en la aplicación es la siguiente:

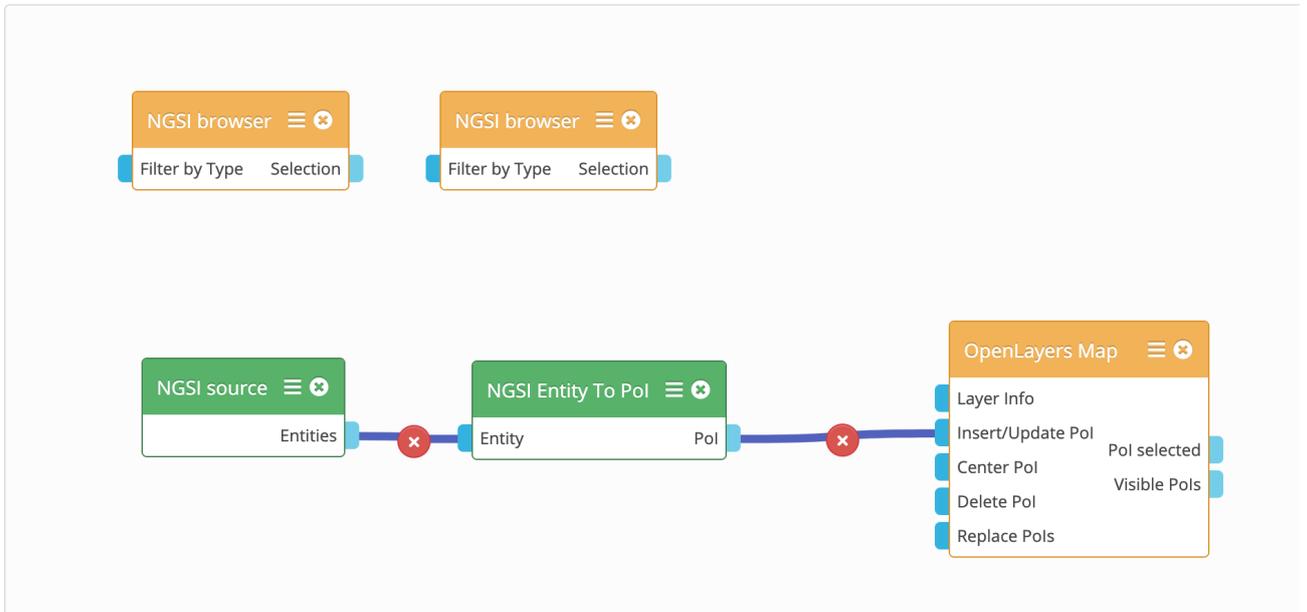


Ilustración 27 - Mapa de la relación entre componentes de la aplicación

Una vez definida la relación entre componentes, configuraremos los parámetros de dichos componentes con el fin de poder recoger los datos que hemos dado de alta en el sistema, en este caso primero los diferentes sensores y actuadores existentes:

Configurar ✕

NGSI server URL

Use the FIWARE credentials of the user

NGSI tenant/service

NGSI scope

NGSI entity types

Id pattern

Display Entity Type

Allow Edit

Allow Delete

Run button

Extra Attributes

Ilustración 28 - Configuración del componente Sensor

Dicha conexión se basa en la conexión con la URL `http://orion:1026/`, que es la URL en la cual se pueden encontrar los sensores que hemos dado de alta inicialmente.

Como atributos añadiremos los que hemos definido en el momento del post. Éstos corresponden al estado, humedad y temperatura.

Repetiremos este paso con los actuadores, basados también en un componente NGSI Browser con la diferencia que en este caso nuestro campo NGSI entity types será Actuador. De la misma forma, añadiremos los atributos correspondientes, que en este caso es solo el estado del actuador:

Configurar [X]

NGSI server URL

Use the FIWARE credentials of the user

NGSI tenant/service

NGSI scope

NGSI entity types

Id pattern

Display Entity Type

Allow Edit

Allow Delete

Run button

Extra Attributes

Ilustración 29 - Configuración del componente sensor

Por último, configuraremos los componentes del mapa.

Empezaremos configurando la fuente donde hemos añadido las localizaciones de los edificios municipales donde hemos realizado la instalación:

Configuración del operador ×

NGSI server URL	<input type="text" value="http://orion:1026/"/>
NGSI proxy URL	<input type="text" value="http://ngsi-proxy:8100"/>
Use the FIWARE credentials of the user	<input type="checkbox"/>
Use the FIWARE credentials of the workspace owner	<input type="checkbox"/>
FIWARE-Service	<input type="text"/>
FIWARE-ServicePath	<input type="text" value="/"/>
NGSI entity types	<input type="text" value="Store"/>
Id pattern	<input type="text"/>
Query	<input type="text"/>
Monitored NGSI Attributes	<input type="text"/>

Ilustración 30 - Configuración del componente de los edificios municipales

Una vez añadidos los edificios municipales, solo nos queda ajustar las coordenadas del mapa para que las localizaciones puedan ser cargadas por defecto, con el objetivo de poder limitar el mapa a la localidad de Vilamarxant:

Configurar ✕

Initial Location

Initial Zoom Level

Min Zoom

Pol Zoom

Layers Widget

Use Clustering

Ilustración 31 - Configuración del componente del mapa

Con esto, la aplicación ya está lista para mostrar la información correspondiente y empezar a funcionar.

6 Pruebas

Una vez hemos seguido los pasos necesarios para la puesta en marcha completa del sistema, iniciaremos las pruebas con el fin de asegurar el correcto funcionamiento de la aplicación.

Para ello, empezaremos definiendo un conjunto de localizaciones que nos permitirán situar los sensores y actuadores de test:

```
curl -s -o /dev/null -X POST \
  'http://orion:1026/v2/op/update' \
  -H 'Content-Type: application/json' \
  -d '{
    "actionType": "append",
    "entities": [
      {
        "id": "urn:ngsi-ld:Store:001", "type": "Store",
        "address": { "type": "PostalAddress", "value": { "streetAddress": "Av. Poliesportiu", "addressRegion": "Valencia", "addressLocality": "Vilamarxant", "postalCode": "46191" } },
        "location": { "type": "geo:json", "value": { "type": "Point", "coordinates": [-0.6243872, 39.5659149] } },
        "name": { "type": "Text", "value": "Polideportivo" }
      },
      {
        "id": "urn:ngsi-ld:Store:002", "type": "Store",
        "address": { "type": "PostalAddress", "value": { "streetAddress": "Calle de Luis Santágel", "addressRegion": "Valencia", "addressLocality": "Vilamarxant", "postalCode": "46191" } },
        "location": { "type": "geo:json", "value": { "type": "Point", "coordinates": [-0.6207604, 39.5713202] } },
        "name": { "type": "Text", "value": "Pabellón" }
      }
    ]
  }'
```

Ilustración 32 - POST de las localizaciones de los edificios municipals

Definimos en este caso dos ubicaciones: el pabellón y el polideportivo de Vilamarxant. Estas ubicaciones se determinan por calle, región, localidad y código postal.

Añadimos igualmente las coordenadas de dichas ubicaciones de manera que se puedan leer directamente por el componente de mapa de la aplicación.

En segundo lugar, definiremos un conjunto de sensores y actuadores de test.

Los sensores y actuadores de la primera ubicación son los siguientes:

```
curl -s -o /dev/null -X POST \
  "http://iot-agent:$IOTA_NORTH_PORT/iot/devices" \
  -H 'Content-Type: application/json' \
  -H 'fiware-service: openiot' \
  -H 'fiware-servicepath: /' \
  -d '{
  "devices": [
    {
      "device_id": "sensor001",
      "entity_name": "Sensor:001",
      "entity_type": "Sensor",
      "protocol": "PDI-IoTA-UltraLight",
      "transport": "HTTP",
      "endpoint": "\"http://iot-sensors:$TUTORIAL_DUMMY_DEVICE_PORT/iot/sensor001\"",
      "commands": [
        {"name": "on", "type": "command"},
        {"name": "off", "type": "command"}
      ],
      "attributes": [
        {"object_id": "s", "name": "estado", "type": "Text"},
        {"object_id": "h", "name": "humedad", "type": "Integer"},
        {"object_id": "t", "name": "temperatura", "type": "Integer"}
      ],
      "static_attributes": [
        {"name": "category", "type": "Text", "value": ["actuator", "sensor"]},
        {"name": "controlledProperty", "type": "Text", "value": "light"},
        {"name": "function", "type": "Text", "value": ["onOff", "sensing"]},
        {"name": "supportedProtocol", "type": "Text", "value": ["ul20"]},
        {"name": "supportedUnits", "type": "Text", "value": "CDL"},
        {"name": "refStore", "type": "Relationship", "value": "urn:ngsi-ld:Store:001"}
      ]
    }
  ],
  }
```

Ilustración 33 - Post de los sensores de la aplicación

```
{
  "device_id": "actuador001",
  "entity_name": "Actuador:001",
  "entity_type": "Actuador",
  "protocol": "PDI-IoTA-UltraLight",
  "transport": "HTTP",
  "endpoint": "\"http://iot-sensors:$TUTORIAL_DUMMY_DEVICE_PORT/iot/actuador001\"",
  "commands": [
    {
      "name": "encender",
      "type": "command"
    }
  ],
  "attributes": [
    {"object_id": "s", "name": "estado", "type": "Text"}
  ],
  "static_attributes": [
    {"name": "category", "type": "Text", "value": ["actuator"]},
    {"name": "controlledProperty", "type": "Text", "value": "noiseLevel"},
    {"name": "function", "type": "Text", "value": ["onOff"]},
    {"name": "supportedProtocol", "type": "Text", "value": ["ul20"]},
    {"name": "refStore", "type": "Relationship", "value": "urn:ngsi-ld:Store:001"}
  ]
}
}
```

Ilustración 34 - Post de los actuadores de la aplicación

Enviaremos estos parámetros a través del curl de un fichero JSON que contiene los valores mostrados anteriormente.

Resaltaremos primero el comando “encender” que encontraremos tanto en los sensores como en los actuadores. Este comando nos permitirá controlar su estado en función de los datos recogidos.

Por otro lado, para los sensores, resaltaremos los atributos “estado”, “humedad” y “temperatura” que nos informarán de la situación atmosférica de cada una de las ubicaciones, así como el estatus de los sensores, y nos permitirán tomar las decisiones necesarias en base a dichas informaciones.

De la misma forma, encontramos en los actuadores el atributo de estado, que nos permitirá saber si el actuador se encuentra activo o no.

A continuación, añadiremos los sensores y actuadores necesarios en la ubicación 2 como sigue:

```
curl -s -o /dev/null -X POST \
  "http://iot-agent:$IOTA_NORTH_PORT/iot/devices" \
  -H 'Content-Type: application/json' \
  -H 'fiware-service: openiot' \
  -H 'fiware-servicepath: /' \
  -d '{
  "devices": [
    {
      "device_id": "sensor002",
      "entity_name": "Sensor:002",
      "entity_type": "Sensor",
      "protocol": "PDI-IoTA-UltraLight",
      "transport": "HTTP",
      "endpoint": "\"http://iot-sensors:$TUTORIAL_DUMMY_DEVICE_PORT/iot/sensor002\"",
      "commands": [
        {"name": "on", "type": "command"},
        {"name": "off", "type": "command"}
      ],
      "attributes": [
        {"object_id": "s", "name": "estado", "type": "Text"},
        {"object_id": "h", "name": "humedad", "type": "Integer"},
        {"object_id": "t", "name": "temperatura", "type": "Integer"}
      ],
      "static_attributes": [
        {"name": "category", "type": "Text", "value": ["actuator", "sensor"]},
        {"name": "controlledProperty", "type": "Text", "value": "light"},
        {"name": "function", "type": "Text", "value": ["onOff", "sensing"]},
        {"name": "supportedProtocol", "type": "Text", "value": ["ul20"]},
        {"name": "supportedUnits", "type": "Text", "value": "CDL"},
        {"name": "refStore", "type": "Relationship", "value": "urn:ngsi-ld:Store:002"}
      ]
    }
  ],
}
```

Ilustración 35 - Definición de los sensores

```
{
  "device_id": "actuador001",
  "entity_name": "Actuador:001",
  "entity_type": "Actuador",
  "protocol": "PDI-IoTA-UltraLight",
  "transport": "HTTP",
  "endpoint": "'http://iot-sensors:$TUTORIAL_DUMMY_DEVICE_PORT/iot/actuador001'",
  "commands": [
    {
      "name": "encender",
      "type": "command"
    }
  ],
  "attributes": [
    {"object_id": "s", "name": "estado", "type": "Text"}
  ],
  "static_attributes": [
    {"name": "category", "type": "Text", "value": ["actuador"]},
    {"name": "controlledProperty", "type": "Text", "value": "noiseLevel"},
    {"name": "function", "type": "Text", "value": ["onOff"]},
    {"name": "supportedProtocol", "type": "Text", "value": ["ul20"]},
    {"name": "refStore", "type": "Relationship", "value": "urn:ngsi-ld:Store:001"}
  ]
}
]
```

Ilustración 36 - Definición de los actuadores

Añadiremos en el mismo post un sensor y un actuador con el fin de limitar las llamadas. No obstante, cada sensor y actuador corresponden a una localización específica, no son dependientes los unos de los otros.

Por último, repetiremos los pasos para los sensores y actuadores 3 y 4, situados en las ubicaciones 1 y 2 respectivamente.

Estos conjuntos de sensores y localizaciones no corresponden a los conjuntos reales y finales, en esta ocasión los daremos de alta como objetos de pruebas con el fin de garantizar la usabilidad de la aplicación.

Aun así, los datos que introduciremos intentarán ser lo más realistas posibles con el objetivo de asemejar la información de test introducida a la realidad lo máximo posible.

Dichos datos serán generados de forma aleatoria en base a un rango realista de humedad y temperatura, los cuales insertaremos en la base de datos Mongo DB que gestiona los datos de dichos componentes.

La pantalla de Wirecloud que visualizaremos una vez generado los datos será la siguiente:

alice/Vilamarxant app

NGSI browser

Id	Type	estado	humedad	temperatura	Actions
Sensor:001	Sensor	on	70	25	
Sensor:002	Sensor	on	68	25	
Sensor:003	Sensor	on	69	25	
Sensor:004	Sensor	on	68	26	

Page: 1/1

NGSI browser

Id	Type	estado	Actions
Actuador:001	Actuador	on	[Edit] [Delete]
Actuador:002	Actuador	on	[Edit] [Delete]
Actuador:003	Actuador	on	[Edit] [Delete]

OpenLayers Map

© OpenStreetMap contributors.

Pestaña

Powered by

Ilustración 37 - Aplicación Wirecloud con los datos enviados por los sensores y los estados de los actuadores

7 Conclusiones y futuras ampliaciones

A lo largo de este TFG, hemos podido llevar a cabo el proyecto que nos ha permitido detallar y explicar el desarrollo y funcionamiento de una aplicación Fiware que permitirá al ayuntamiento de Vilamarxant poder incrementar su eficiencia energética.

Tras haber recopilado toda la información que se ha especificado, podemos asegurar que se han cumplido los objetivos y sub objetivos detallados a lo largo del mismo:

Hemos definido una aplicación web Fiware con una interfaz visual al utilizador que permite la gestión y recopilación de la información recogida.

En dicha aplicación hemos implementado las herramientas que permitan automatizar y controlar diferentes acciones relacionadas con el consumo energético de las dependencias municipales.

Hemos ligado estas herramientas a actuadores que se instalan físicamente permitiendo la gestión del consumo energético dependiendo de los datos recogidos en la aplicación Fiware.

Podemos revisar gracias a diferentes bases de datos (tanto relacionales como no relacionales) todas las informaciones que hemos añadido en la aplicación (cada base de datos almacenará toda acción efectuada en la aplicación con el fin de facilitar el control de la misma por parte de los administradores).

Podemos, a través de varios usuarios, llevar a cabo un control de la plataforma

La arquitectura que hemos puesto en marcha corresponde con la que se ha detallado a partir de un código abierto basado en Fiware, siguiendo con la implementación por defecto de la misma.

El conjunto de sensores existentes nos permite poder seguir a tiempo real las informaciones tomadas en cuanto a temperatura y humedad.

Es posible retomar la información emitida por los sensores con el fin de poder identificar el consumo que se ha tenido en las instalaciones y así poder determinar a posteriori la facturación que se tendrá.

Estos puntos han sido recogidos inicialmente en el acta de inicio del proyecto que firmamos ambas partes, por lo que hemos cumplido con los objetivos pactados inicialmente, a expensas de poder añadir las ampliaciones que detallamos a continuación.

En cuanto a las ampliaciones que podremos llevar a cabo en el futuro, podemos basarnos en los siguientes puntos:

- Añadiremos la relación con otros trabajos que se centran en otros puntos de las Smart cities, permitiendo así tener una vista de 360 grados sobre los datos que se recogen en la ciudad correspondiente (en el pueblo de Vilamarxant en este caso).
- Profundizaremos en el sistema de alertas gestionado por la aplicación con el fin de poder mantener un nivel de alerta mayor sobre los resultados de los valores tomados por los sensores.

- Implementaremos una red de actuadores capaces de tomar las decisiones necesarias a partir de una IA cuyas reglas estarán definidas por los usuarios correspondientes.
- Nos centraremos en la mejora de los resultados obtenidos basándonos en los picos de emisión CO₂, permitiendo poder reducirlos al máximo a partir de un estudio de las horas de mayores emisiones.
- Mejoraremos la sensorización existente con el fin de poder tener una serie de informaciones más precisas, mejorando así también la toma de decisiones.
- Añadiremos nuevos componentes y funcionalidades a la aplicación permitiendo al usuario tener una mejora visual y funcional.

Dichas implementaciones a futuro serán previamente reevaluadas por ambas partes para poder establecer tanto el coste temporal como el alcance tácito de cada uno de los puntos.

Esto nos permitirá tener una visión clara de nuevo proceso y subproyecto que llevaremos a cabo.

Cabe resaltar que el último sub objetivo destacado al inicio del documento no se ha cumplido ya que el resto de TFGs relacionados a esta aplicación Fiware no han sido presentados en el plazo esperado. Esto no nos ha permitido llevar a cabo la interconexión deseada con el resto de trabajos.

8 Bibliografía

Energy and the Green Deal. (s. f.). [Text]. Comisión Europea – European Commission. Recuperado 29 de marzo de 2022, de https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal/energy-and-green-deal_es

@NatGeoES. (2017, agosto 3). *El calor extremo podría hacer que algunas zonas de Asia sean inhabitables en 2100*. National Geographic. <https://www.nationalgeographic.es/medio-ambiente/2017/08/el-calor-extremo-podria-hacer-que-algunas-zonas-de-asia-sean-inhabitables-en-2100>

Fiware. (2022). En *Wikipedia, la enciclopedia libre*. <https://es.wikipedia.org/w/index.php?title=FIWARE&oldid=144630232>

Smart City: Las 7 Ciudades más Inteligentes de España. (2017, noviembre 10). Inarquia. <https://inarquia.es/smart-city-ciudades-inteligentes-espana/>

Barcelona smart city o ciudad inteligente. (2019, mayo 2). Bcn Advisors. <https://www.bcn-advisors.com/barcelona-una-ciudad-inteligente-lider>

Proyecto mint: Madrid, la ciudad más inteligente de europa. (2014, octubre 17). Inarquia. <https://inarquia.es/el-proyecto-mint-convertira-a-madrid-en-la-ciudad-mas-inteligente-de-europa/>

La nueva capital smart city de la mano de Madrid Nuevo Norte. (2021, enero 20). <https://www.madridesnoticia.es/2021/01/capital-smart-city-madrid-nuevo-norte/>

Proyectos de transformación hacia la Ciudad Inteligente. (s. f.). *València Ciudad Inteligente*. Recuperado 15 de mayo de 2022, de <https://smartcity.valencia.es/proyectos/>

Geoportal. (s. f.). *València Ciudad Inteligente*. Recuperado 16 de noviembre de 2023, de <https://smartcity.valencia.es/vlci/geoportal/>

València se integra en la red europea WiFi4EU y ofrece internet gratuito de alta velocidad. (s. f.). *València Ciudad Inteligente*. Recuperado 16 de noviembre de 2023, de <https://smartcity.valencia.es/vlci/wifi4eu/>

Smart grids. (s. f.). IEA. Recuperado 16 de noviembre de 2023, de <https://www.iea.org/energy-system/electricity/smart-grids>

SmartCity Málaga. (s. f.). Recuperado 15 de mayo de 2022, de <https://malagasmart.malaga.eu/es/habitat-sostenible-y-seguro/energia/smartcity-malaga/#:~:text=Smartcity%20M%C3%A1laga%20fue%20un%20proyecto,avanzad as%20con%20capacidad%20de%20almacenamiento.>

ESMARTCITY. (2023, marzo 29). *El Índice Smart Cities España de 2023 analiza la accesibilidad de las ciudades*. ESMARTCITY. <https://www.esmartcity.es/2023/03/29/indice-smart-cities-espana-2023-analiza-accesibilidad-ciudades>

455df16a1a. (s. f.). *Indice Smart 2023*. Indice Smart. Recuperado 7 de noviembre de 2023, de <https://www.idencityconsulting.com/indicesmart/>
Build your own IoT platform with FIWARE enablers – FIWARE. (2015, marzo 27).
<https://www.fiware.org/2015/03/27/build-your-own-iot-platform-with-fiware-enablers/>

9 ANEXOS

9.1 Anexo I: Manual del usuario Administrador

Recogeremos en el manual del usuario la guía paso a paso que permitirá a un individuo acceder a la aplicación para poder realizar las tareas necesarias.

Como prerrequisitos, supondremos que los servicios han sido lanzados y levantados previamente.

De la misma forma, las credenciales del usuario habrán sido creadas anteriormente por el administrador al cargo y validadas para la conexión.

Una vez los prerrequisitos han sido llevados a cabo, los pasos de conexión del usuario son los siguientes.

Primero, deberá conectarse a la aplicación a través de la URL `http://localhost:8000`:

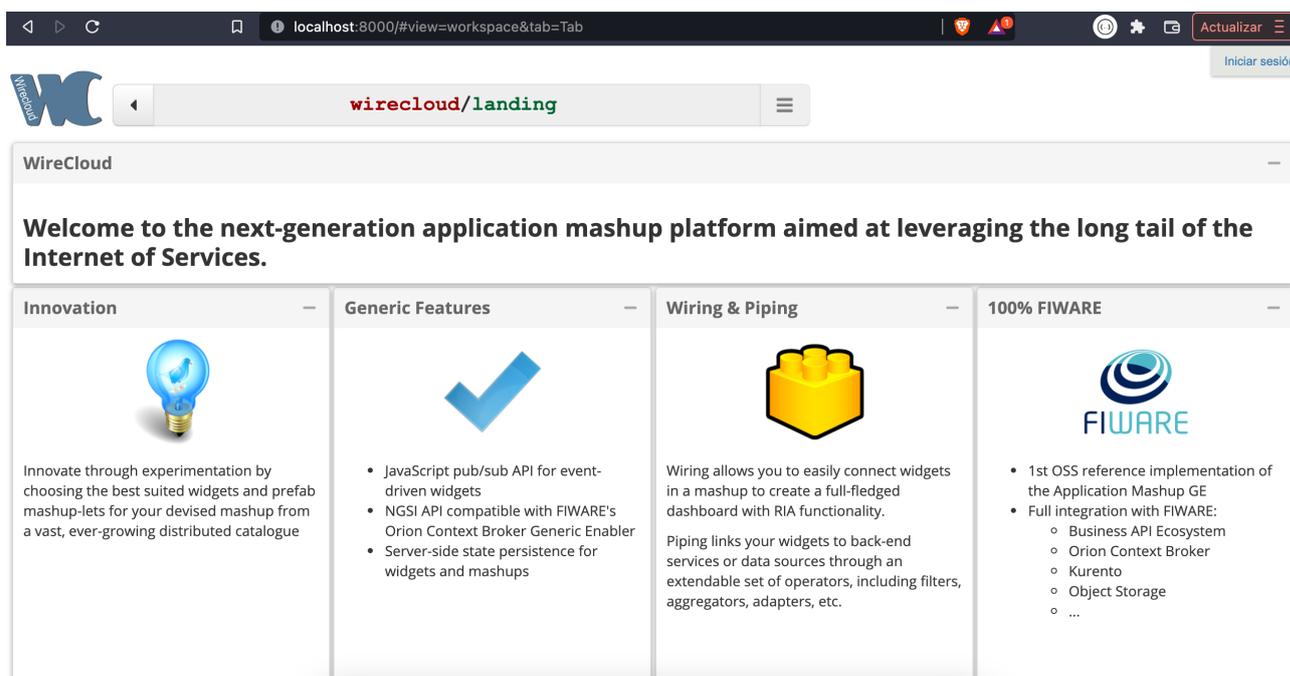
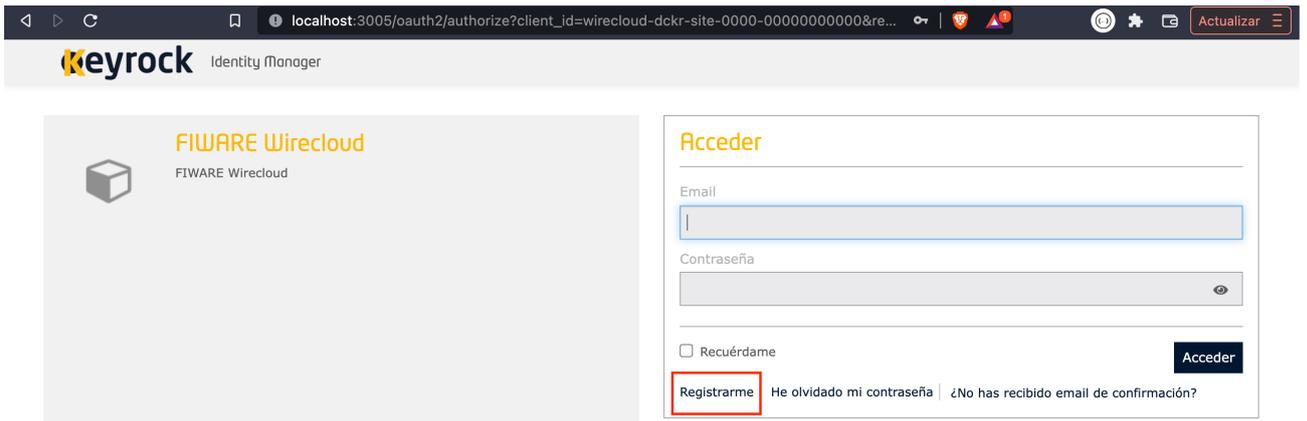


Ilustración 38 - Pantalla de carga de Wirecloud

Una vez conectado, el usuario deberá introducir las credenciales que le han sido asignadas para entrar en la página inicial (como hemos detallado previamente en el punto 5.2.1). En caso de tratarse de la primera conexión para un nuevo usuario, deberá registrarse primero:



2018 © GING DIT - Universidad Politécnica de Madrid (UPM)

es

Ilustración 39 - Registro de un nuevo usuario en Keyrock para la aplicación

El botón “Registrarme” permitirá al nuevo usuario crear una cuenta nueva:

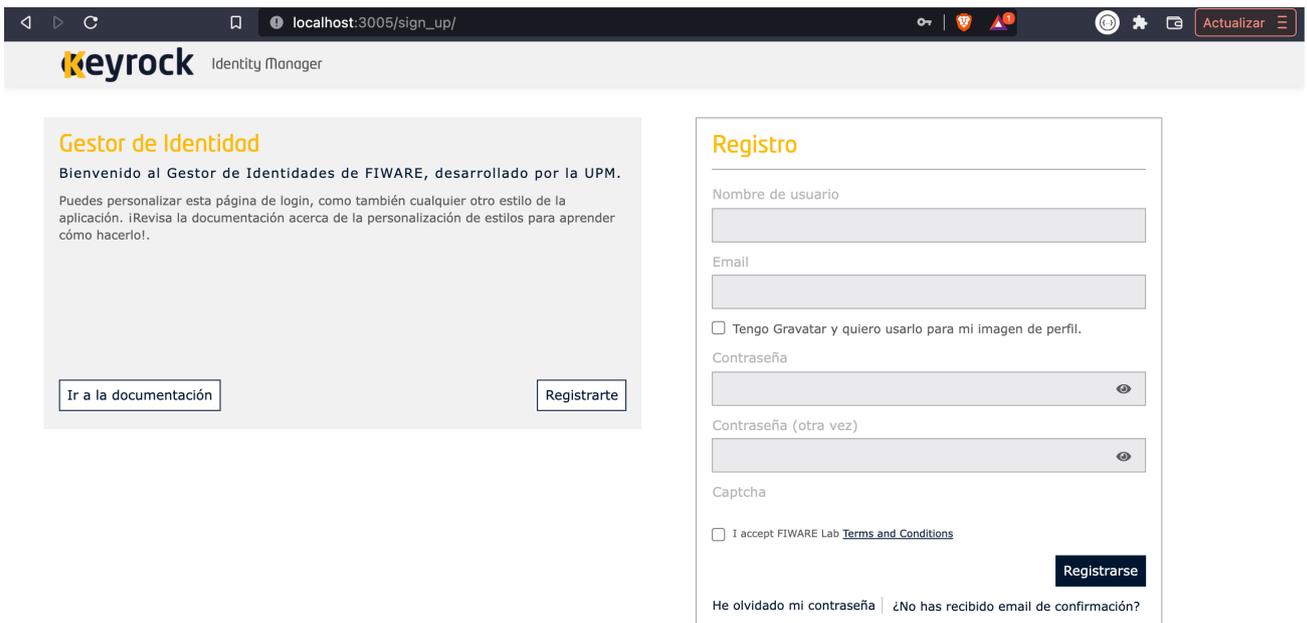


Ilustración 40 - Inserción de datos del nuevo usuario

Con las credenciales creadas y validadas, el usuario deberá seleccionar el entorno de trabajo correspondiente dentro de la aplicación, en este caso el entorno llamado “Vilamarxant”.

Dicho entorno apunta directamente al previamente dado de alta con los widgets necesarios:

Desarrollo de una aplicación IoT basada en Fiware para incrementar la eficiencia energética en una Smart City

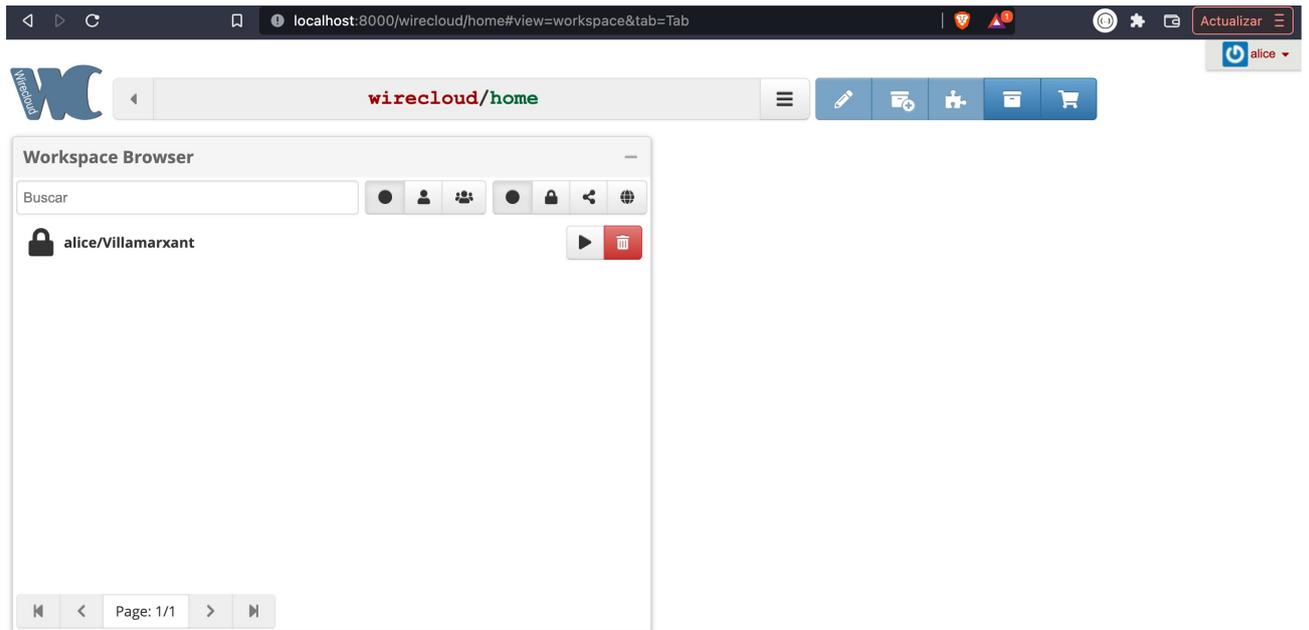


Ilustración 41 - Pantalla de entornos de trabajo de Wirecloud

Llegado a este punto, el usuario es capaz de visualizar la interfaz ya totalmente configurada y cargada, lo que le permitirá poder revisar los valores tomados por los sensores que han sido añadidos también anteriormente en el momento de la instalación y configuración del entorno.

9.2 Anexo II: Manual del instalador

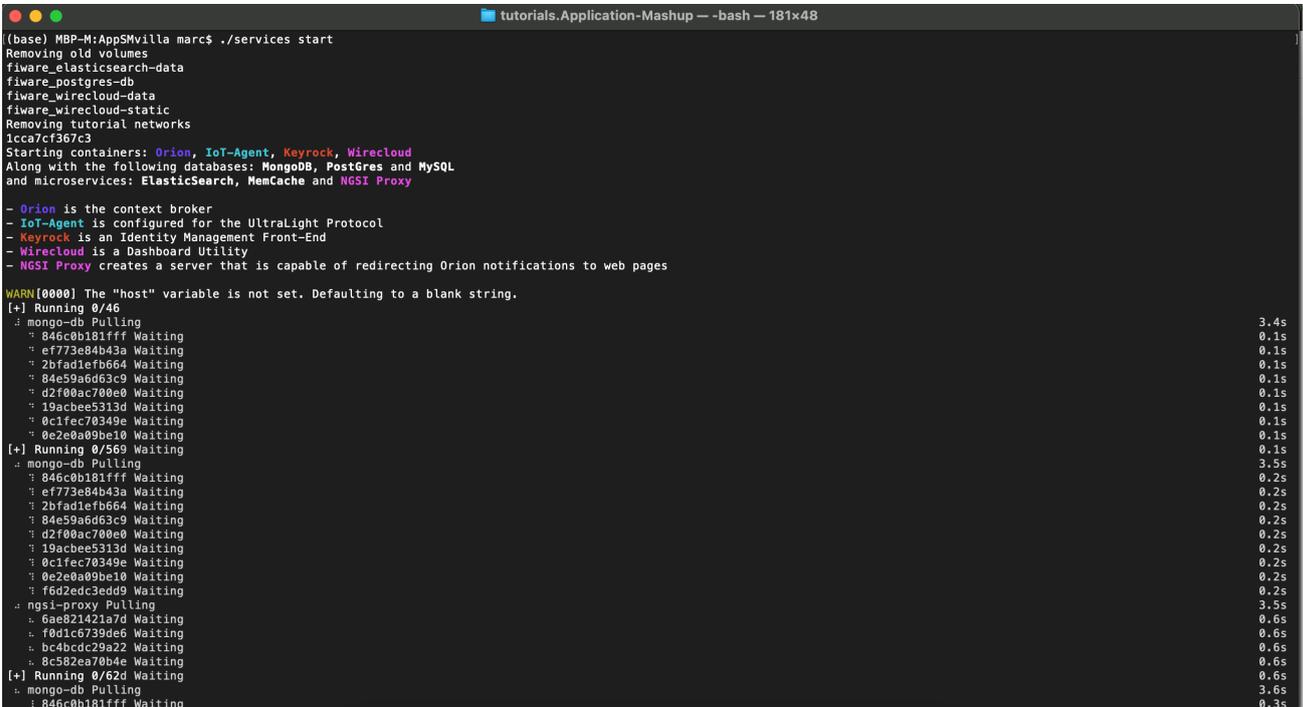
Para poder instalar la aplicación, el instalador deberá descomprimir el fichero .zip transferido al cliente con los ficheros de la aplicación.

Una vez descomprimido dicho fichero, el instalador deberá acceder a la ruta que contiene la carpeta que contiene los ficheros proporcionados (dicha ruta será especificada en el momento de la integración en el sistema del cliente).

El instalador deberá inicialmente ingresar dicha ruta en el editor CMD.

Una vez accedida la ruta, deberá lanzar los servicios gracias al comando “./services start”.

Esto permitirá a la aplicación la descarga de todos los ficheros necesarios, así como la instalación en el sistema de destino (en el cual se está lanzando el comando). De la misma forma se descargarán las imágenes de los componentes requeridos por la aplicación:



```
(base) MBP-M:AppSMVilla marc$ ./services start
Removing old volumes
fiware_elasticsearch-data
fiware_postgres-db
fiware_wirecloud-data
fiware_wirecloud-static
Removing tutorial networks
1ccc7cf367c3
Starting containers: Orion, IoT-Agent, Keyrock, Wirecloud
Along with the following databases: MongoDB, PostGres and MySQL
and microservices: Elasticsearch, MemCache and NGSI Proxy

- Orion is the context broker
- IoT-Agent is configured for the UltraLight Protocol
- Keyrock is an Identity Management Front-End
- Wirecloud is a Dashboard Utility
- NGSI Proxy creates a server that is capable of redirecting Orion notifications to web pages

WARN[0000] The "host" variable is not set. Defaulting to a blank string.
[+] Running 0/46
  ⚙️ mongo-db Pulling
   ⚙️ 846c0b181fff Waiting
   ⚙️ ef773e84b43a Waiting
   ⚙️ 2bfad1e7b664 Waiting
   ⚙️ 84e59a6d63c9 Waiting
   ⚙️ d2f00ac700e0 Waiting
   ⚙️ 19acbee5313d Waiting
   ⚙️ 0c1fec70349e Waiting
   ⚙️ 0e2e0a09be10 Waiting
[+] Running 0/569 Waiting
  ⚙️ mongo-db Pulling
   ⚙️ 846c0b181fff Waiting
   ⚙️ ef773e84b43a Waiting
   ⚙️ 2bfad1e7b664 Waiting
   ⚙️ 84e59a6d63c9 Waiting
   ⚙️ d2f00ac700e0 Waiting
   ⚙️ 19acbee5313d Waiting
   ⚙️ 0c1fec70349e Waiting
   ⚙️ 0e2e0a09be10 Waiting
   ⚙️ f6d2edc3edd9 Waiting
  ⚙️ ngsi-proxy Pulling
   ⚙️ 6ae82142187d Waiting
   ⚙️ f0d1c6739de6 Waiting
   ⚙️ bc4bdc29a22 Waiting
   ⚙️ 8c582ea70b4e Waiting
[+] Running 0/62d Waiting
  ⚙️ mongo-db Pulling
   ⚙️ 846c0b181fff Waiting
```

Ilustración 42 - Terminal con la descarga de las imágenes necesarias para la aplicación

Cuando todas las imágenes hayan sido descargadas, el sistema comprobará a través de la recepción de una respuesta http igual a 200 que todas las URL necesarias son accesibles y funcionan correctamente:

Desarrollo de una aplicación IoT basada en Fiware para incrementar la eficiencia energética en una Smart City

```
Container mongo          Started      6.3s
Container db-postgres   Started      6.4s
Container wc-memcached  Started      5.8s
Container wc-ngsi-proxy Started      8.3s
Container fiware-keyrock Started      8.6s
Container fiware-orion  Started     10.4s
Container fiware-iot-agent Started     10.1s
Container fiware-wirecloud Started     12.3s

Waiting for MongoDB to be available
Adding appropriate MongoDB indexes for Orion ... done
Adding appropriate MongoDB indexes for IoT-Agent ... done
Waiting for Orion to be available
Loading context data done
Waiting for IoT-Agent to be available
IoT Agent HTTP state: 000 (waiting for 200)
IoT Agent HTTP state: 000 (waiting for 200)
IoT Agent HTTP state: 200 (waiting for 200)
Provisioning IoT devices done
Waiting for Wirecloud to be available
Wirecloud HTTP state: 000 (waiting for 200)
done
Waiting for Keyrock to be available
Keyrock HTTP state: 000 (waiting for 200)
done
Now open http://localhost:8000
```

Ilustración 43 - Terminal con las respuestas de los estados HTTP de los módulos cargados

Cuando el estado http recibido sea igual a 200 para todos los módulos instalados, la aplicación estará lista para ser utilizada.

Para poder proceder con la instalación, nos dirigiremos a la dirección `http://localhost:8000`, procediendo a la autenticación para poder conectarnos.

Una vez autenticados, procederemos a añadir los widgets necesarios para visualizar las informaciones de los sensores, tiendas y actuadores.

Para ello, nos dirigiremos en primera instancia al icono de caja situado en la parte superior de la pantalla:

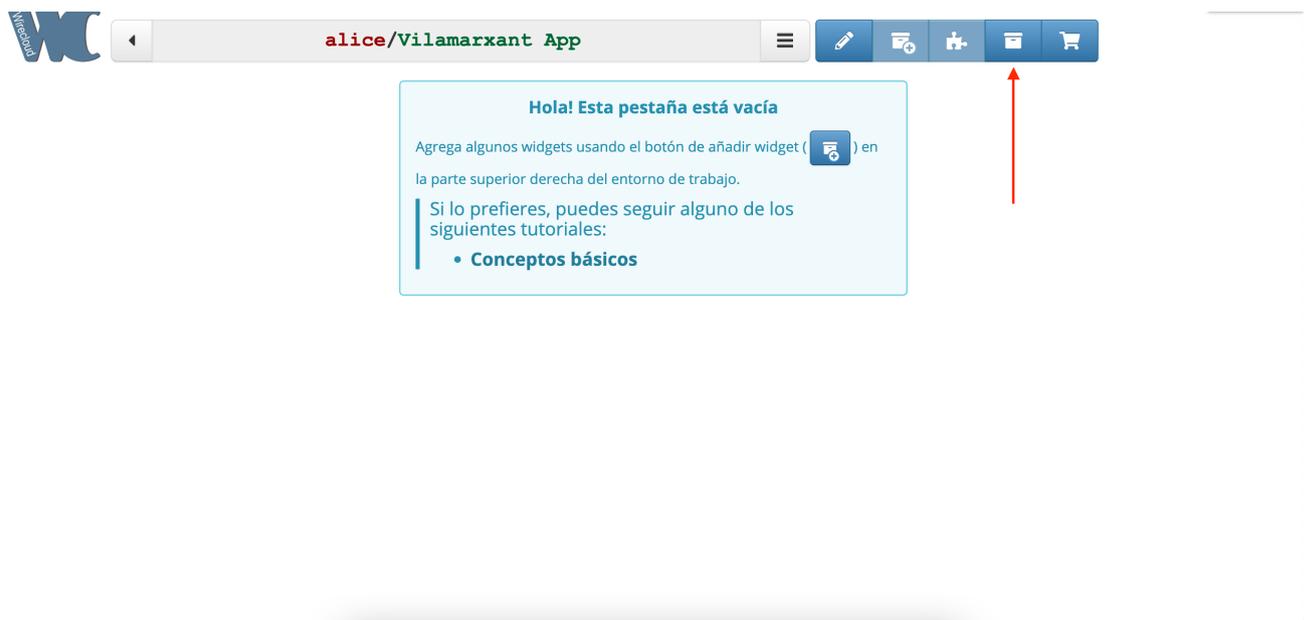


Ilustración 44 - Pantalla de inicio Wirecloud

Esto nos abrirá la siguiente pantalla, que nos permitirá cargar los widgets proporcionados en la carpeta “Widget” contenida en el fichero .zip de la instalación:

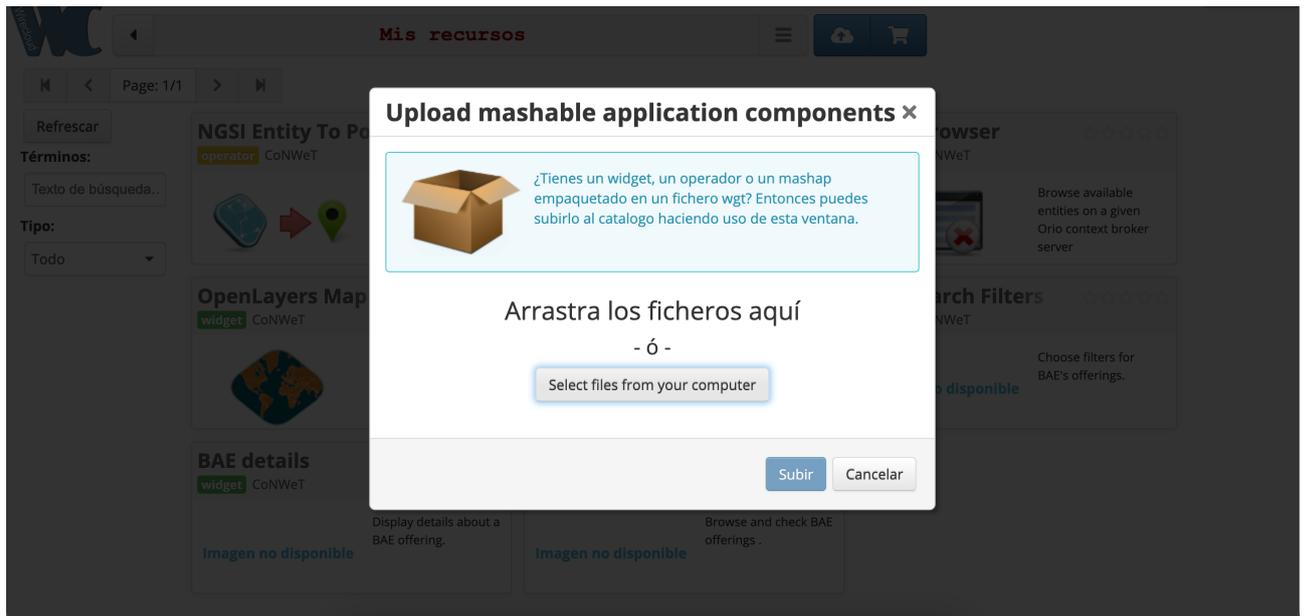


Ilustración 45 - Añadir nuevos widgets en Wirecloud

Una vez hemos añadido todos los widgets, podremos visualizar todos los componentes necesarios para poder proseguir con la instalación:

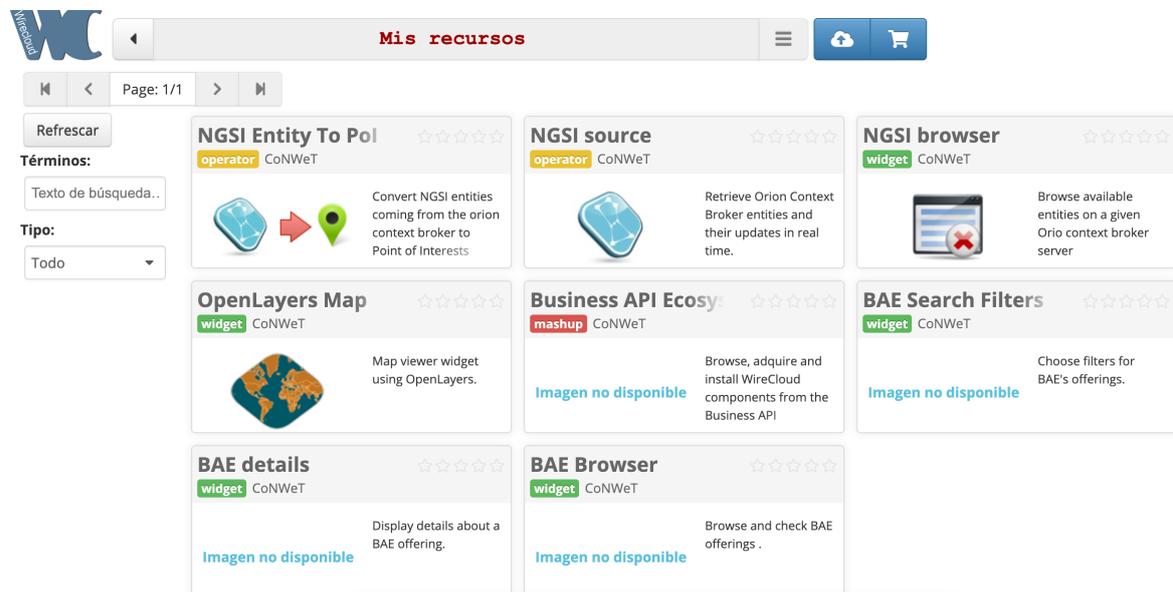


Ilustración 46 - Pantalla de widgets instalados en Wirecloud

Los nuevos widgets añadidos serán “NGSI Entity to PoI”, “NGSI source”, “NGSI browser” y “OpenLayers Map”. Cada componente instalado tiene una función específica:

- NGSI Entity to PoI recibe las entidades NGSI insertadas previamente en Orion y las convierte para que los puntos de interés (PoI) puedan ser leídos por el Context Broker de Orion.

- NGSI source permite actualizar a tiempo real los datos de las entidades insertadas en el Context Broker de Orion.
- NGSI browser permite leer las entidades insertadas en el Context Broker de Orion.
- OpenLayers Map permite visualizar en un mapa las entradas insertadas en Orion, transferidas por el NGSI Entity To PoI.

Una vez hemos añadido todos los componentes, deberemos configurarlos correctamente para que puedan mostrar toda la información.

Para ello, nos dirigiremos a la pantalla principal de nuevo y clicaremos primero en el icono de lápiz, seguido del icono de pieza de puzle que estará activado:



Ilustración 47 - Descripción pantalla de inicio para unir los componentes instalados

Añadiremos y conectaremos todos los componentes como se muestra a continuación:

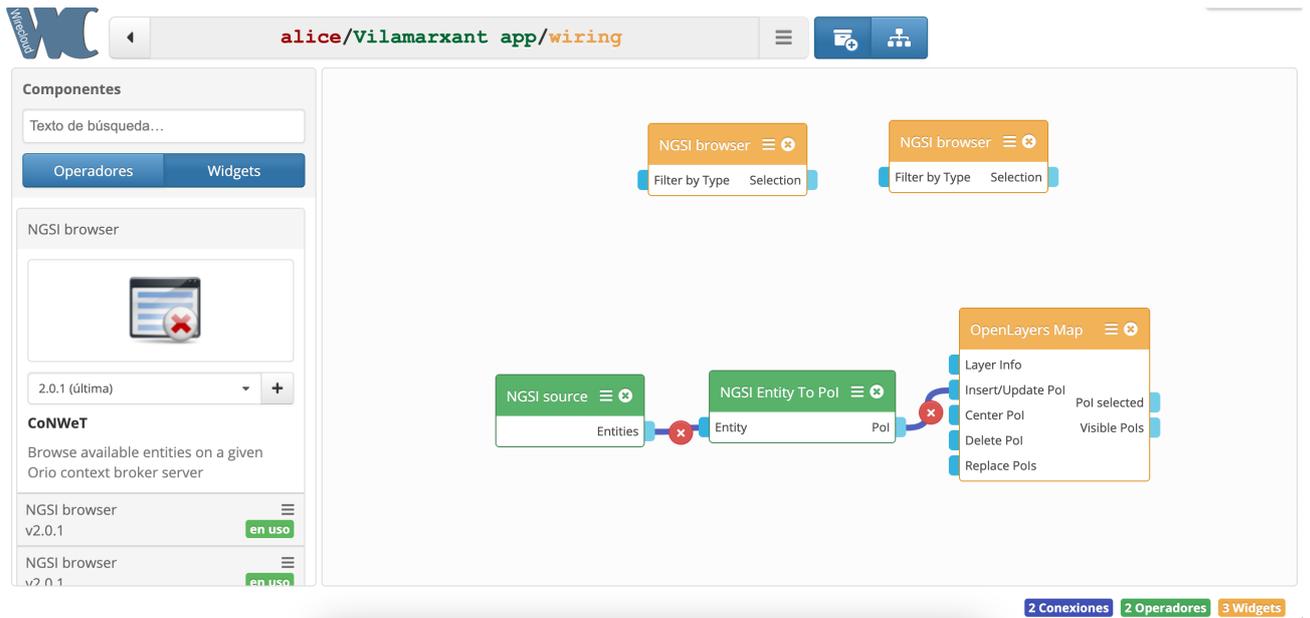


Ilustración 48 - Pantalla de unión entre los componentes

Los dos primeros componentes situados arriba (NGSI Browser) nos mostrarán los sensores y actuadores que han sido añadidos a la base de datos.

Clicaremos sobre los 3 puntos situados en la esquina de cada uno de los componentes y los configuraremos de la siguiente forma.

Empezaremos por los sensores:

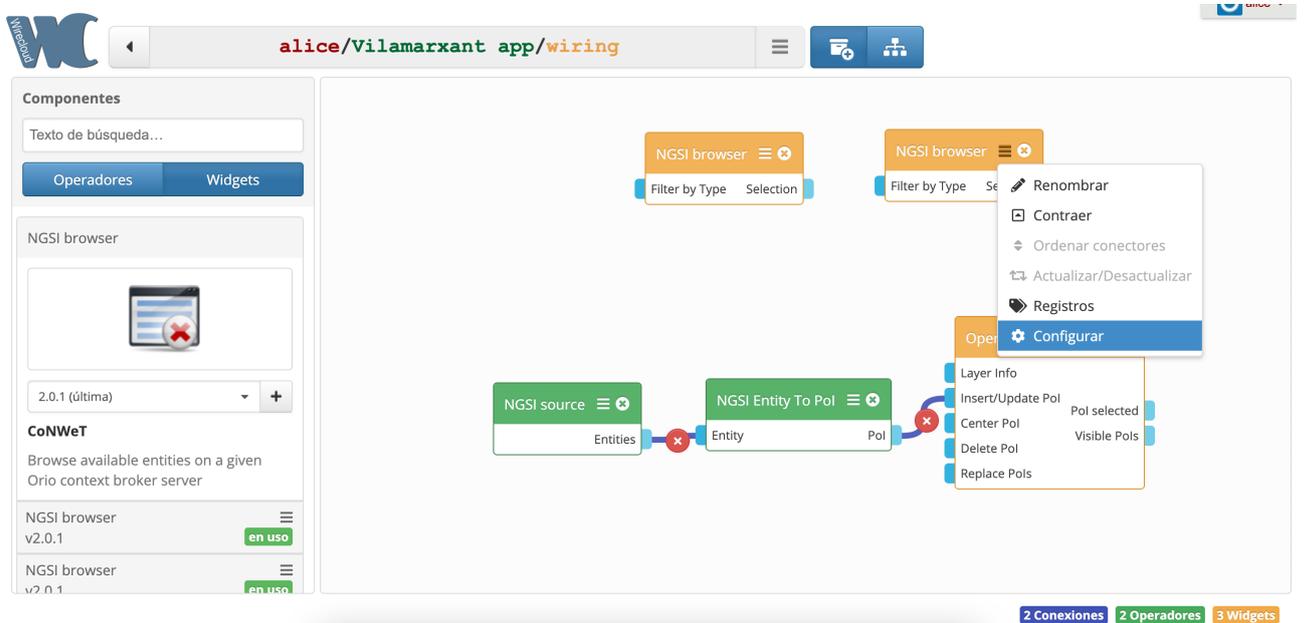


Ilustración 49 - Configuración de los componentes sensores y actuadores de la aplicación

Clicaremos en la opción de configurar, y añadiremos los siguientes valores:

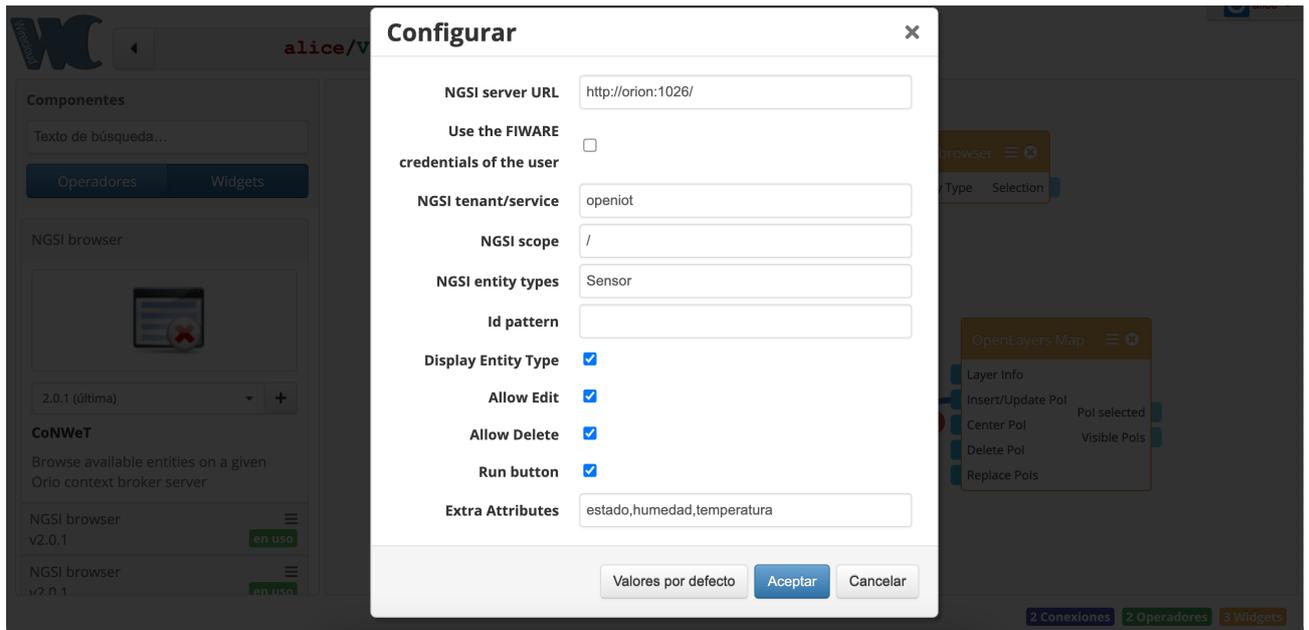


Ilustración 50 - Configuración sensores

Los atributos extra situados al final del formulario los habremos añadido en el fichero JSON de los sensores.

Repetiremos la acción para configurar los actuadores:

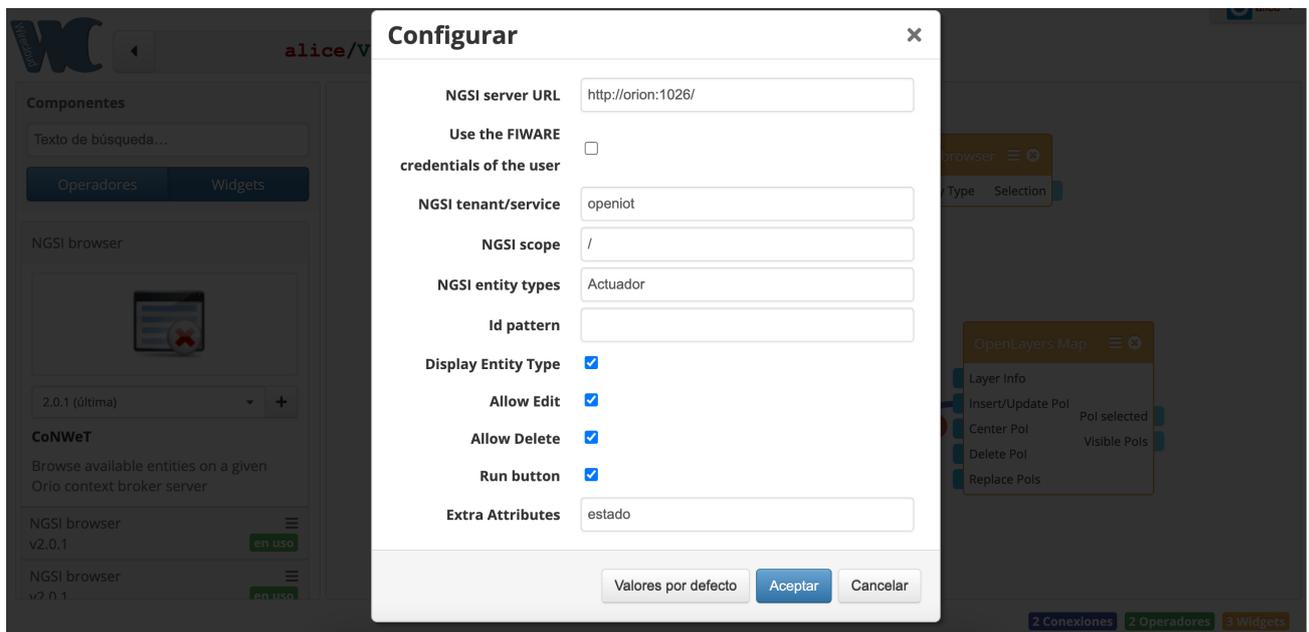


Ilustración 51 - Configuración actuadores

Como hemos indicado anteriormente, esto nos permitirá visualizar la información de los sensores y actuadores añadidos en base de datos.

A continuación, configuraremos las ubicaciones para que puedan ser mostradas en el mapa.

Para ello, empezaremos por el widget “NGSI source”, en el cual clicaremos en los 3 puntos situados en la esquina como anteriormente:

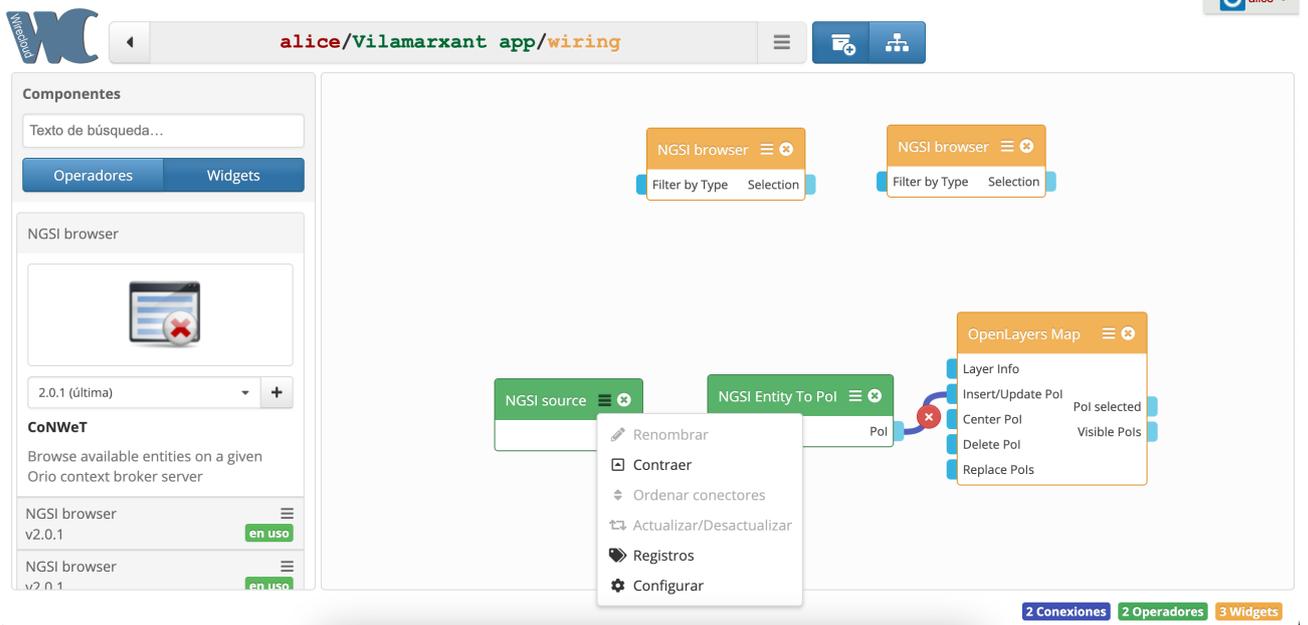


Ilustración 52 - Configuración de los componentes del mapa

Esto nos abrirá el formulario correspondiente, en el cual añadiremos los parámetros siguientes:

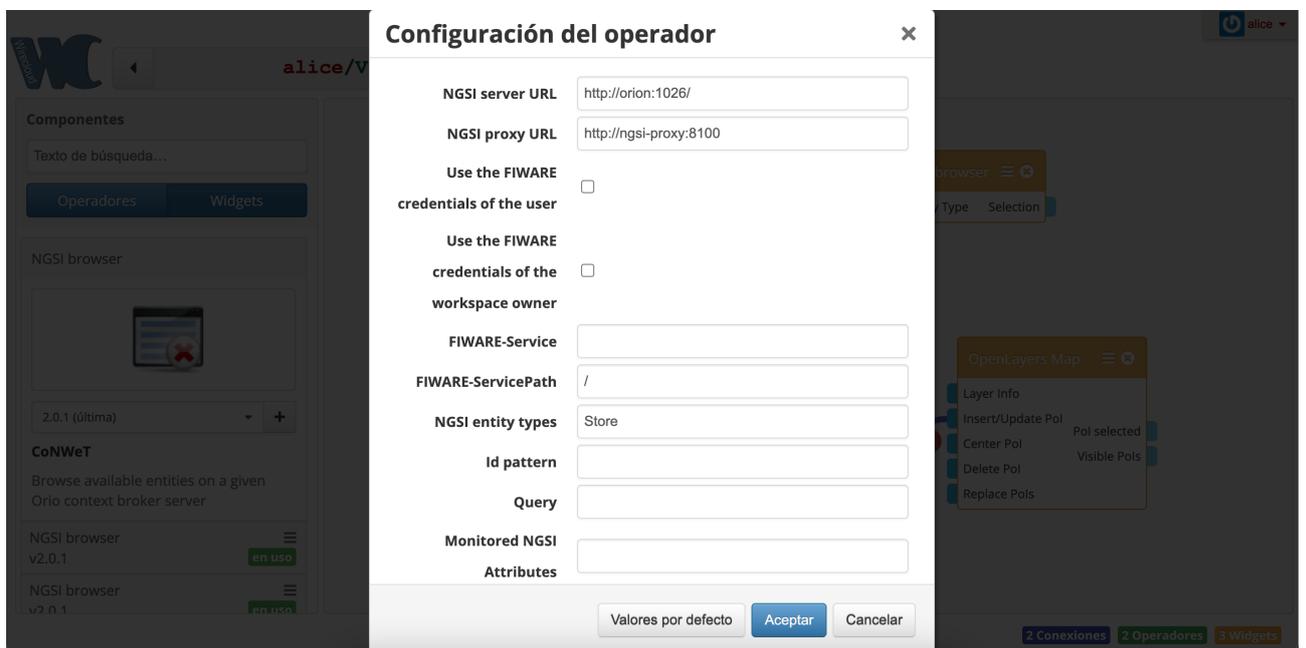


Ilustración 53 - Configuración de las localizaciones

Esto nos permitirá recibir la información de las ubicaciones determinadas en el fichero JSON.

Configuraremos después la localización, gracias al componente “NGSI Entity To Pol”:

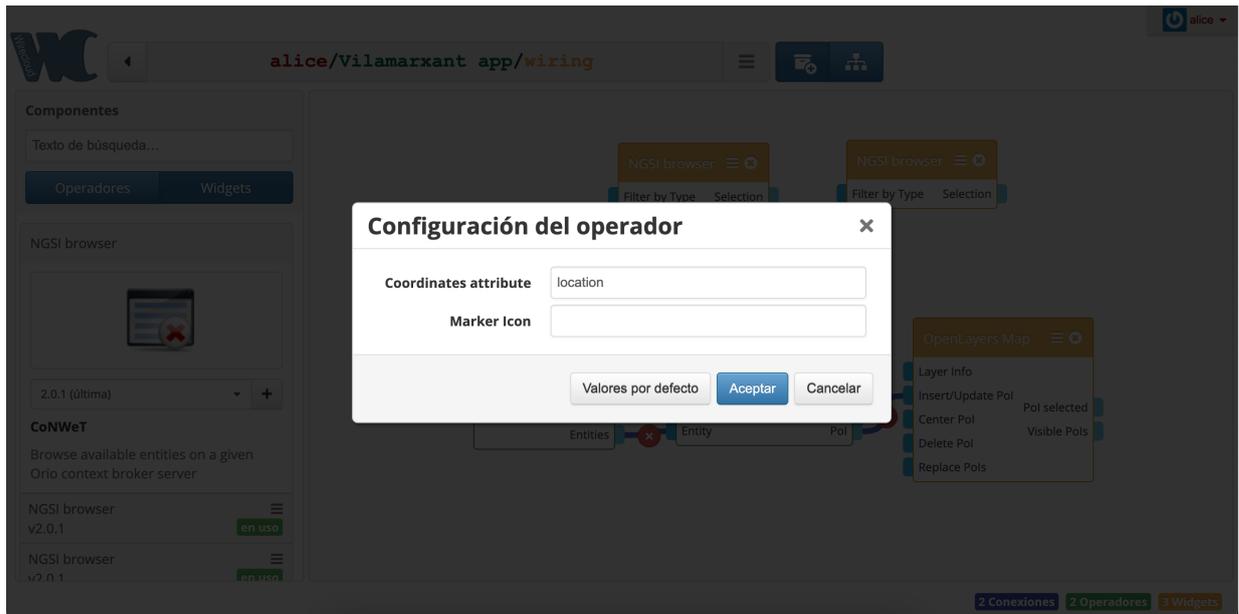


Ilustración 54 - Configuración de los atributos del mapa

Por último, configuraremos el mapa para que muestre por defecto la localización directa donde se encuentran las ubicaciones existentes:

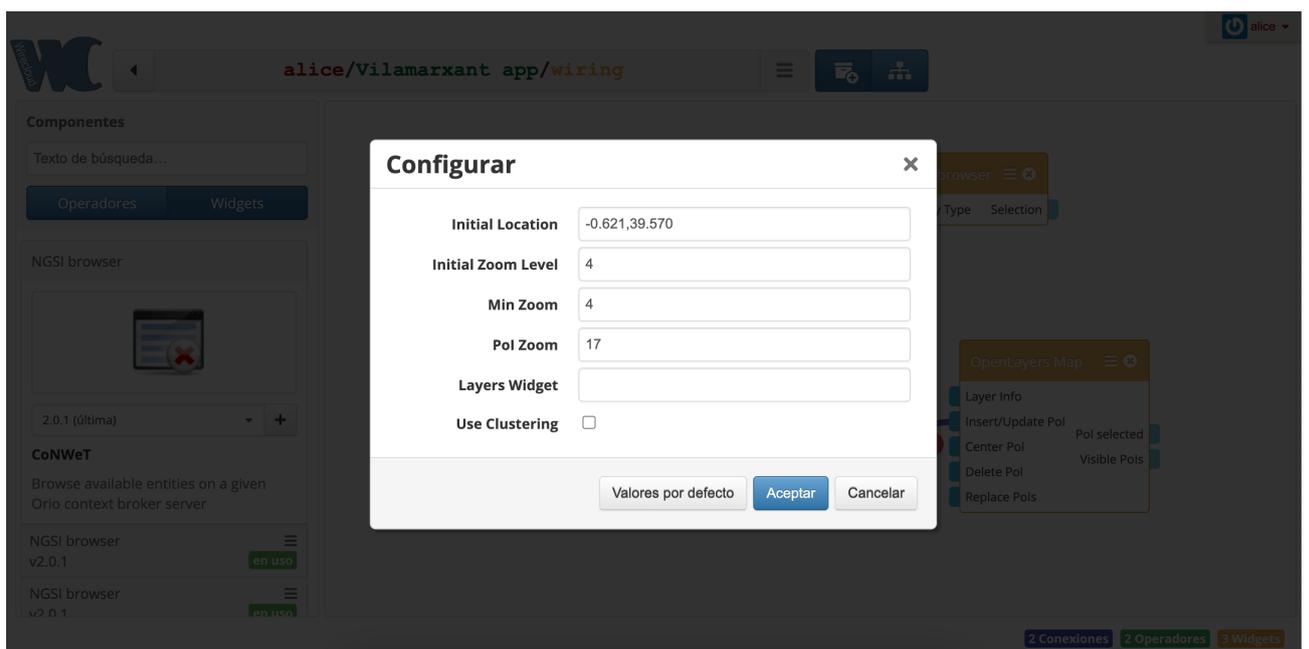


Ilustración 55 - Configuración de las coordenadas del mapa

La localización inicial se tomará a partir de la ubicación de Vilamarxant. Los valores del zoom permitirán acercarse lo suficiente como para que el usuario vea claramente las ubicaciones enviadas a la aplicación.

Una vez el instalador haya configurado toda la aplicación correctamente, la pantalla inicial que veremos será la siguiente:

alice/Vilamarxant app

NGSI browser

Id	Type	estado	humedad	temperatura	Actions
Sensor:001	Sensor	on	70	25	
Sensor:002	Sensor	on	68	25	
Sensor:003	Sensor	on	69	25	
Sensor:004	Sensor	on	68	26	

Page: 1/1

NGSI browser

Id	Type	estado	Actions
Actuador:001	Actuador	on	
Actuador:002	Actuador	on	
Actuador:003	Actuador	on	

OpenLayers Map

© OpenStreetMap contributors.

Pestaña

Powered by Wirecloud

Ilustración 56 - Dashboard de la aplicación Wirecloud después de la configuración de todos los componentes

9.3 Anexo III: Objetivos de Desarrollo Sostenible (ODS)

9.3.1 Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.			X	
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.	X			
ODS 8. Trabajo decente y crecimiento económico.		X		
ODS 9. Industria, innovación e infraestructuras.	X			
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles.	X			
ODS 12. Producción y consumo responsables.	X			
ODS 13. Acción por el clima.		X		
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.				X

9.3.2 Reflexión sobre la relación del TFG con los ODS

El desarrollo de una aplicación Fiware para mejorar la eficiencia energética en el Ayuntamiento de Vilamarxant se presenta como un desafío multifacético que involucra varios Objetivos de Desarrollo Sostenible (ODS). Estos ODS no solo sirven como una guía para abordar cuestiones clave a nivel global, sino que también ofrecen un marco integral para abordar desafíos locales, como la gestión eficiente de recursos y la sostenibilidad. A continuación, se analizará cómo cada ODS se relaciona con el proyecto.

ODS 3 - Salud y bienestar (Bajo): Aunque el ODS 3, centrado en la salud y el bienestar, tiene un grado de implicación bajo en este contexto, no se puede subestimar su importancia. La eficiencia energética contribuye indirectamente a la mejora de la salud al reducir las emisiones contaminantes y crear entornos más sostenibles. La calidad del aire mejorada y la disminución de la contaminación acústica son beneficios secundarios que impactan positivamente en la salud y el bienestar de la comunidad.

ODS 7 - Energía asequible y no contaminante (Alto): El ODS 7 es central para el proyecto, ya que se enfoca en garantizar el acceso a una energía asequible, sostenible y moderna. Desarrollar una aplicación Fiware para aumentar la eficiencia energética en el Ayuntamiento de Vilamarxant contribuirá directamente a este objetivo al optimizar el consumo de energía, promover fuentes de energía renovable y reducir la dependencia de fuentes no sostenibles.

ODS 8 - Trabajo decente y crecimiento económico (Medio): El ODS 8 se relaciona con la promoción del crecimiento económico sostenible y el trabajo decente. La implementación de una aplicación Fiware para mejorar la eficiencia energética generará oportunidades de empleo en el desarrollo, implementación y mantenimiento de la tecnología. Además, al promover la sostenibilidad, se pueden crear empleos especializados en energías renovables y tecnologías limpias, contribuyendo así al desarrollo económico sostenible.

ODS 9 - Industria, innovación e infraestructuras (Alto): La relación con el ODS 9 es significativa, ya que el proyecto implica la introducción de innovaciones tecnológicas para mejorar la eficiencia energética. Desarrollar una aplicación Fiware significa impulsar la infraestructura digital y tecnológica del Ayuntamiento de Vilamarxant, fomentando la innovación y mejorando la gestión de recursos. Esto no solo beneficia a la administración local, sino que también establece un precedente para la adopción de soluciones tecnológicas en otros sectores.

ODS 11 - Ciudades y comunidades sostenibles (Alto): Este objetivo es esencial para el proyecto, ya que se centra en lograr que las ciudades sean inclusivas, seguras, resilientes y sostenibles. Mejorar la eficiencia energética en el Ayuntamiento de Vilamarxant mediante una aplicación Fiware contribuye directamente a la sostenibilidad urbana. Reducir la huella de carbono, optimizar los recursos y promover prácticas sostenibles en la gestión municipal son elementos clave para lograr comunidades más sostenibles.

ODS 12 - Producción y consumo responsables (Alto): La producción y el consumo responsables son fundamentales para abordar la crisis ambiental actual. El desarrollo de una aplicación Fiware para mejorar la eficiencia energética promueve prácticas de consumo responsables al optimizar el uso de recursos energéticos. Además, al implementar tecnologías

más eficientes, se reduce el desperdicio y se fomenta una cultura de producción más sostenible.

ODS 13 - Acción por el clima (Medio): La acción por el clima es un componente crucial del proyecto, aunque su grado de implicación es medio. Al mejorar la eficiencia energética, se contribuye directamente a la reducción de emisiones de gases de efecto invernadero y se mitiga el impacto ambiental. La aplicación Fiware puede facilitar la monitorización en tiempo real del consumo energético, permitiendo una toma de decisiones informada para reducir la huella de carbono del Ayuntamiento.

En resumen, el desarrollo de una aplicación Fiware para aumentar la eficiencia energética en el Ayuntamiento de Vilamarxant está intrínsecamente ligado a varios ODS. Desde la promoción de energía asequible y no contaminante hasta la construcción de ciudades y comunidades sostenibles, el proyecto se alinea con los principios fundamentales de desarrollo sostenible.

Al abordar estos objetivos de manera integral, se puede lograr un impacto significativo en la calidad de vida de la comunidad, fomentando la innovación, el crecimiento económico y la sostenibilidad a largo plazo.