

Máster Universitario en Software de Sistemas Distribuidos y Empotrados

*Optimización del consumo energético en las aulas de la
ETSISI mediante la inteligencia colectiva:
Aplicación móvil y registro seguro de los datos mediante
Blockchain*

PROYECTO FIN DE MÁSTER

José Luis Doménech Martínez

Julio de 2022

Máster Universitario en Software de Sistemas Distribuidos y Empotrados

*Optimización del consumo energético en las aulas de
la ETSISI mediante la inteligencia colectiva:*

*Aplicación móvil y registro seguro de los datos
mediante Blockchain*

PROYECTO FIN DE MÁSTER

Autor: José Luis Doménech Martínez
Director: Javier García Martín

Julio de 2022

Dedicatoria

Se lo dedico a mis padres: José Luis y Paqui.

Agradecimientos

Quiero agradecer a la UPM por incluir este proyecto entre los merecedores de una de las Becas de Sostenibilidad de los Campus UPM 2022 dirigidas a la consecución de los Objetivos de Desarrollo Sostenible.

Quiero agradecer al grupo de trabajo SYST el ofrecerme la oportunidad de trabajar en un proyecto que plantea retos tan interesantes y me permite realizar mi pequeña aportación a mejorar el mundo.

A mi familia y amigos, por estar ahí.

A mis compañeros de trabajo en Kumux, en especial Adriá y Javier.

A mis compañeros del máster, por poder contar con ellos.

A Ana-María Chacón Villaseñor le quiero agradecer ser una excelente compañera durante toda mi andadura en este máster y en especial durante la realización de este trabajo y de su TFM: **Optimización del consumo energético en las aulas de la ETSISI mediante inteligencia colectiva: Sensorización de las aulas, análisis de datos y decisiones de actuación.**

A los profesores del máster les tengo que agradecer todo el esfuerzo que han puesto. Siempre se ha podido contar con ellos para solucionar cualquier duda o imprevisto durante el curso.

En especial quiero agradecer a Javier García Martín por recomendarme este máster y ofrecerme la realización de un proyecto de TFM tan interesante y, por supuesto, por su guía y paciencia durante la realización del mismo.

Índice

Dedicatoria.....	3
Agradecimientos.....	4
Índice.....	5
Índice de Figuras.....	7
Índice de tablas.....	8
Glosario.....	9
Resumen.....	10
Palabras claves:.....	11
Abstract.....	12
Keywords:.....	13
Capítulo 1 : Introducción.....	1
1.1. Descripción del problema.....	1
1.2. Relevancia del problema tratado.....	1
1.3. Estado del arte.....	3
1.4. Objetivo del sistema.....	4
1.5. Metodología empleada.....	5
1.6. Particularidades del Sistema.....	6
1.7. Estructura de este documento.....	7
Capítulo 2 : Contexto.....	8
2.1. Internet of things (IoT).....	8
2.2. Crowdsensing.....	9
2.3. Inteligencia colectiva.....	9
2.4. Objetivos de Desarrollo Sostenible.....	9
2.5. Arquitecturas en la Nube[11].....	10
2.6. Plataformas de análisis de datos IoT.....	11
Capítulo 3 : Sistema desarrollado.....	15
3.1. Módulo de captura de datos.....	15
3.2. Módulo de procesado de datos.....	17
3.3. Módulo de almacenamiento.....	19
3.4. Módulo de análisis.....	20
3.5. Módulos de Visualización.....	23
Capítulo 4 : Resultados obtenidos.....	26
4.1. Primera prueba.....	26
4.2. Segunda prueba.....	26
4.3. Tercera prueba.....	26
4.4. Cuarta prueba.....	27
4.5. Quinta prueba.....	28

Capítulo 5 : Discusión del resultado obtenido.....	30
Capítulo 6 : Impacto social y medioambiental.....	31
Capítulo 7 : Planificación y presupuesto.....	32
Capítulo 8 : Conclusiones.....	34
Referencias.....	35
Anexos.....	36
Anexo A – Requisitos del sistema desarrollado.....	36
Anexo B – Variables de configuración.....	38
Anexo C – Fichero de configuración.....	39

Índice de Figuras

<i>Figura 1: Índice de temperatura Global en Tierra y Océanos.....</i>	<i>2</i>
<i>Figura 2: Porcentaje de energía primaria proveniente de renovables [Mundo, UE, UE27 y España]</i>	<i>2</i>
<i>Figura 3: Arquitectura de un proyecto usando Kafka y Spark.....</i>	<i>3</i>
<i>Figura 4: Arquitectura básica de procesamiento de un sistema IoT.....</i>	<i>8</i>
<i>Figura 5: Flujo básico de análisis de datos con AWS IoT Analytics.....</i>	<i>12</i>
<i>Figura 6: Esquema de análisis de datos de arquitectura híbrida que con Stream Analytics de Azure</i>	<i>13</i>
<i>Figura 7: Análisis de datos IoT con Kaa.....</i>	<i>14</i>
<i>Figura 8: Flujo de datos en el sistema.....</i>	<i>15</i>
<i>Figura 9: Ejemplo de datos capturados.....</i>	<i>16</i>
<i>Figura 10: Carga de datos.....</i>	<i>17</i>
<i>Figura 11: Proceso de datos.....</i>	<i>18</i>
<i>Figura 12: Datos de los sensores tras el procesado agrupados en intervalos de una hora.....</i>	<i>19</i>
<i>Figura 13: Decodificación de la información almacenada mediante cachier.....</i>	<i>20</i>
<i>Figura 14: Formato de almacenamiento usado por cachier.....</i>	<i>20</i>
<i>Figura 15: Variables independientes: se toman los valores medios de cada medición.....</i>	<i>21</i>
<i>Figura 16: Variables dependientes: se toman las valoraciones medias y se redondean.....</i>	<i>22</i>
<i>Figura 17: Web con la documentación de la API.....</i>	<i>24</i>
<i>Figura 18: Web de pruebas de la API.....</i>	<i>25</i>
<i>Figura 19: Datos de 1 mes completo agrupados de forma diaria y cargados en la dashboard.....</i>	<i>27</i>
<i>Figura 20: Predicción de sensación de temperatura, con agrupación diaria.....</i>	<i>28</i>
<i>Figura 21: Datos de 1 mes completo agrupados por horas y cargados en la dashboard.....</i>	<i>28</i>
<i>Figura 22: Predicción de sensación de temperatura, con agrupación por horas.....</i>	<i>29</i>
<i>Figura 23: Gráfica Gnatt del desarrollo.....</i>	<i>33</i>

Índice de tablas

Tabla 1: Acceso a la API para un conjunto vacío.....	26
Tabla 2: Acceso a la API de los datos correspondientes a un único día.....	26
Tabla 3: Acceso a la API de los datos correspondientes a un mes.....	26
Tabla 4: Medidas del modelo generado para agrupaciones de 1D y 1M con los datos de un mes completo.....	27
Tabla 5: Coeficientes de la regresión lineal para la temperatura de los datos de 1 mes agrupados por horas.....	29
Tabla 6: Presupuesto.....	32

Glosario

- **Accuracy**: Porcentaje de ejemplos correctamente clasificados.
- **API**: Application programming interface
- **AWS**: Amazon Web Services
- **batch**: referido al procesado por lotes
- **Crowdsensing**: consiste en el análisis de procesos a partir de la información extraída de los dispositivos móviles/personales de un grupo amplio de personas [1].
 - **edge**: se trata de una arquitectura en la que los datos se procesan lo más cerca posible al punto en el que se obtuvieron.
 - **ELM**: Extreme Learning Machine
 - **ETSISI**: Escuela Técnica Superior de Ingeniería de Sistemas Informáticos
 - **IoT**: Internet of Things
 - **NSA**: Nodo de sensores y/o actuadores
 - **MSE**: Mean Squared Error
 - **ODS**: Objetivos de Desarrollo Sostenible
 - **ONU**: Organización de las Naciones Unidas
 - **Participatory crowdsensing**: consiste en el análisis de procesos a partir de la información explícitamente compartida de los usuarios de dispositivos móviles/personales[1].
 - **PCA**: Principal Component Analysis
 - **REST**: Representational state transfer
 - **SIoTCOM**: Sistemas IoT Conscientes de la Sostenibilidad y Dirigidos por Comunidades Sociales del Centro de Investigación en Tecnologías del Software y Sistemas Multimedia para la Sostenibilidad de la Universidad Politécnica de Madrid (Referencia: PID2020-118969RB-I00 del “Proyectos I+D+i” dentro de los Programas Estatales “Generación de Conocimiento y Fortalecimiento Científico y Tecnológico del Sistema I+D+i” y “I+D+i Orientada a los retos de la Sociedad”).
- **TFG**: Trabajo Fin de Grado
- **TFM**: Trabajo Fin de Máster
- **TTL**: Time to live
- **UE**: Unión Europea
- **UE27**: Unión Europea 27
- **UPM**: Universidad Politécnica de Madrid
- **URL**: Uniform Resource Locator

Resumen

La sobreexplotación de los recursos del planeta pone en peligro el progreso económico y social y degrada el medioambiente natural. Las comunidades y los entornos por los que nos movemos deben limitar y racionalizar el uso de los recursos.

El objetivo de este proyecto es mejorar la eficiencia en el uso de la energía empleada en los sistemas de regulación ambientales (calefacción, aire acondicionado, iluminación) en las aulas de la ETSISI en el Campus Sur.

El proyecto recoge información ambiental proveniente las redes de nodos sensores, actuadores, concentradores y controladores instalados en los espacios de la UPM y las sensaciones de la comunidad universitaria relacionadas con las mediciones ambientales, usando un sistema de participatory crowdsensing.

Hemos desarrollado un sistema capaz de relacionar y analizar los datos de ambos orígenes, y proporcionado una API REST y una dashboard para consultar esta información.

Con los resultados que proporciona la aplicación se pueden tomar decisiones que mejoren la eficiencia en el uso de energía y otros recursos

Palabras claves:

UPM, ETSISI, SYST, IoT, crowdsensing, análisis de datos, sensorización, blockchain

Abstract

Overexploitation of the planet's resources jeopardizes economic and social progress and degrades the natural environment. The communities and the environments in which we move must limit and rationalize the use of resources.

The objective of this project is to improve the efficiency in the use of energy employed in the environmental regulation systems (heating, air conditioning, lighting) in the classrooms of the ETSISI in the Campus Sur.

The project collects environmental information from networks of sensor, and actuators nodes, concentrators and controllers installed in the spaces of the UPM and the feelings of the university community related to environmental measurements, using a participatory crowdsensing system.

We have developed a system capable of relating and analyzing data from both sources, and provided a REST API and a dashboard to access the information.

With the results provided by the application, decisions can be made to improve the efficiency in the use of the resources related with the measured environmental properties.

Keywords:

UPM, ETSISI, SYST, IoT, crowdsensing, data analysis, sensoring, blockchain

Capítulo 1 :

Introducción

1.1. Descripción del problema

La sobreexplotación de los recursos del planeta pone en peligro el progreso económico y social y degrada el medioambiente natural. Las comunidades y los entornos por los que nos movemos deben limitar y racionalizar su uso.

El Campus Sur con más de 2000 alumnos, 300 profesores y su personal es un buen ejemplo de área en el que aplicar medidas de consumo sostenible.

Frente al reto que supone frenar el cambio climático, y en el marco del Plan Estratégico de Sostenibilidad Ambiental, la UPM ha puesto en marcha las Becas de Sostenibilidad de los Campus UPM para el que este TFM ha sido aceptado.

Este TFM forma parte del proyecto **Sistemas IoT Conscientes de la Sostenibilidad y Dirigidos por Comunidades Sociales** (SioTCOM.1_Ambiente) del Centro de Investigación en Tecnologías del Software y Sistemas Multimedia para la Sostenibilidad de la Universidad Politécnica de Madrid, que tiene **referencia: PID2020-118969RB-I00** en las ayudas “**Proyectos I+D+i**” del 2020 enmarcadas dentro de los Programas Estatales de “**Generación de Conocimiento y Fortalecimiento Científico y Tecnológico del Sistema I+D+i**” y “**I+D+i Orientada a los retos de la Sociedad**”

En esta primera fase del proyecto se va a analizar el bienestar de la comunidad universitaria y el ambiente en las aulas de la ETSISI. Su objetivo es contribuir a alcanzar los ODS propuestos por la ONU proporcionando soluciones para un consumo energético más eficiente.

1.2. Relevancia del problema tratado

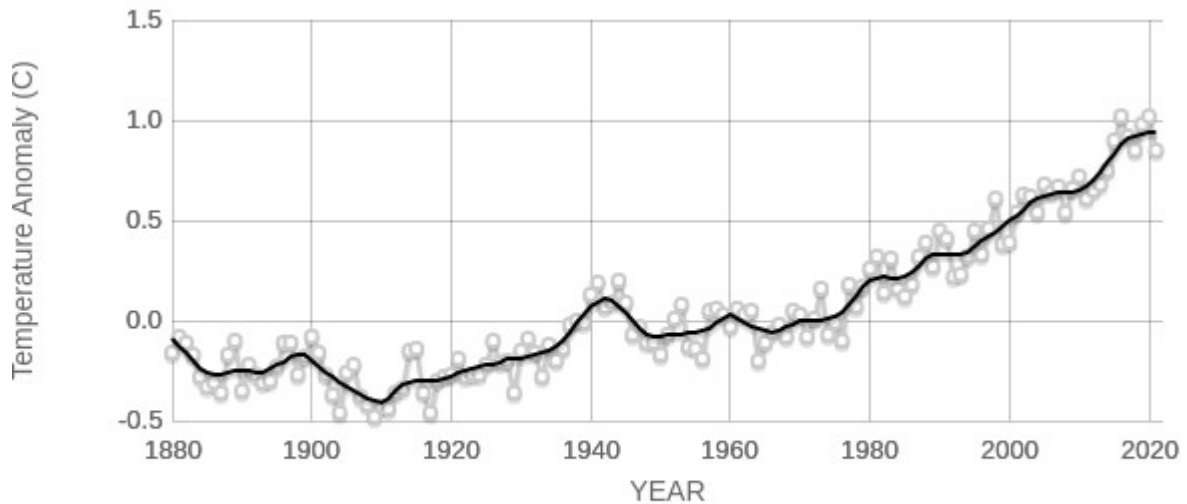
La sobreexplotación de los recursos naturales y el cambio climático son dos problemas relacionados con la utilización de energía de origen fósil [2].

Los ODS son adoptados por todos los miembros de la ONU en 2015, como parte de la Agenda 2030 para el Desarrollo Sostenible¹, sus objetivos son acabar con la pobreza, proteger el planeta y mejorar las vidas y perspectivas de las personas de todo el mundo.

Entre estos objetivos, el ODS 12 es el de garantizar modalidades de consumo y producción sostenibles para favorecer un uso del medio natural y de los recursos del planeta racionalmente. Entre sus retos se quiere minimizar la utilización de combustibles fósiles porque en la generación de este tipo de energía se producen gases de efecto invernadero que contribuyen al calentamiento global y a la contaminación del aire reduciendo el bienestar y la salud en las áreas afectadas [2].

El cambio climático, causado por la expulsión a la atmósfera de gases de efecto invernadero, ha hecho aumentar la temperatura media global del planeta en 0,85 °C [3] como se ve en la figura 1.

¹ Resolución de la ONU aprobando la Agenda 2030 para el Desarrollo sostenible en: https://www.un.org/ga/search/view_doc.asp?symbol=A/RES/70/1&Lang=S



Source: climate.nasa.gov

Figura 1: Índice de temperatura Global en Tierra y Océanos [3].

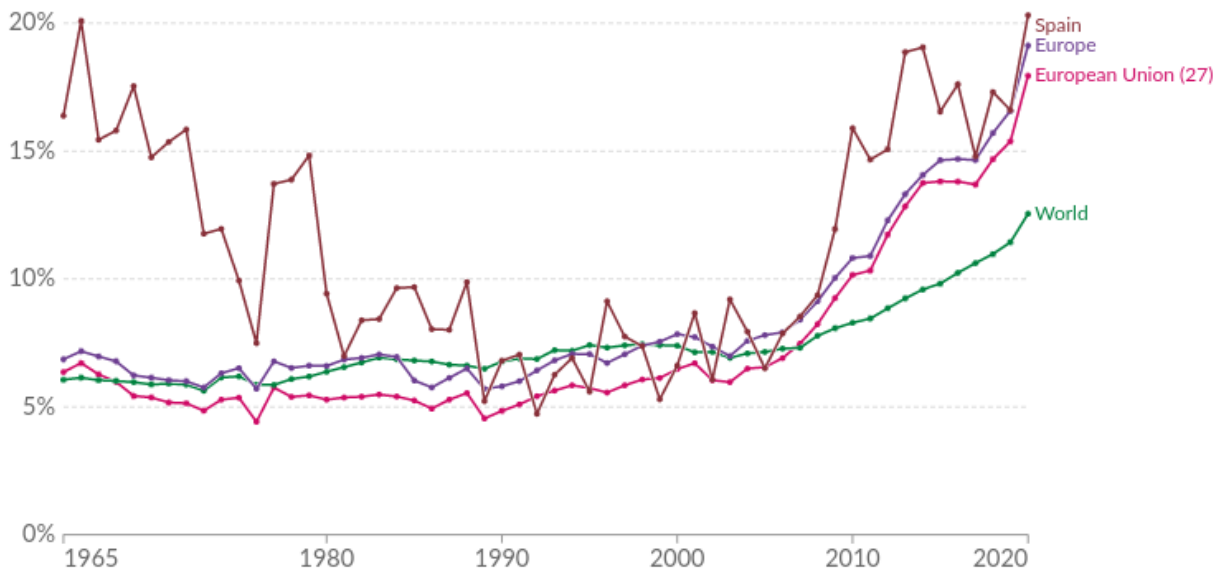
Como se puede comprobar en la figura 2, a pesar de los avances en la reducción del uso de energías fósiles, estas siguen siendo las principales fuentes de energía, por lo que la optimización en el uso de energía es necesaria para alcanzar los ODS.

Share of primary energy from renewable sources

Renewable energy sources include hydropower, solar, wind, geothermal, bioenergy, wave, and tidal. They don't include traditional biofuels, which can be a key energy source, especially in lower-income settings.

Our World in Data

+ Add country



Source: Our World in Data based on BP Statistical Review of World Energy (2021)

Note: Primary energy is calculated using the 'substitution method' which takes account of the inefficiencies energy production from fossil fuels.

OurWorldInData.org/energy • CC BY

Figura 2: Porcentaje de energía primaria proveniente de renovables [Mundo, UE, UE27 y España] [13]

Las soluciones tendrán una mejor implantación y mayor aceptación si los usuarios son parte activa de los procesos que llevan al desarrollo, implantación y aplicación de estas soluciones. Así, el participatory crowdsensing consigue involucrar a los propios usuarios y beneficiarios, que pasan de ser sujetos pasivos a activos. Además, esta actividad permite recopilar una valiosa información que se debe añadir al análisis y la toma de decisiones, ya que enriquece la obtenida de otros medios.

1.3. Estado del arte

1.3.1. Decision Based Model for Real-Time IoT Analysis Using Big Data and Machine Learning [4].

Propone, como ejemplo, la predicción meteorológica analizando los datos recogidos por un sistema de IoT de estaciones meteorológicas.

Como se ve en la figura 3 de la arquitectura propuesta, los datos provenientes de la red IoT alimentan a un Productor de datos IoT que es un único broker de Kafka. Este productor genera un flujo de datos que es enviado a un clúster Spark.

Se usa SparkSQL para obtener Dataframes que serán almacenados en Cassandra y usados para ser analizados.

El análisis consiste en una regresión logística empleando SparkML entrenando un modelo que puede predecir futuras salidas a partir de los datos del sistema IoT.

Por último, los resultados son visualizados en un notebook Zeppelin.

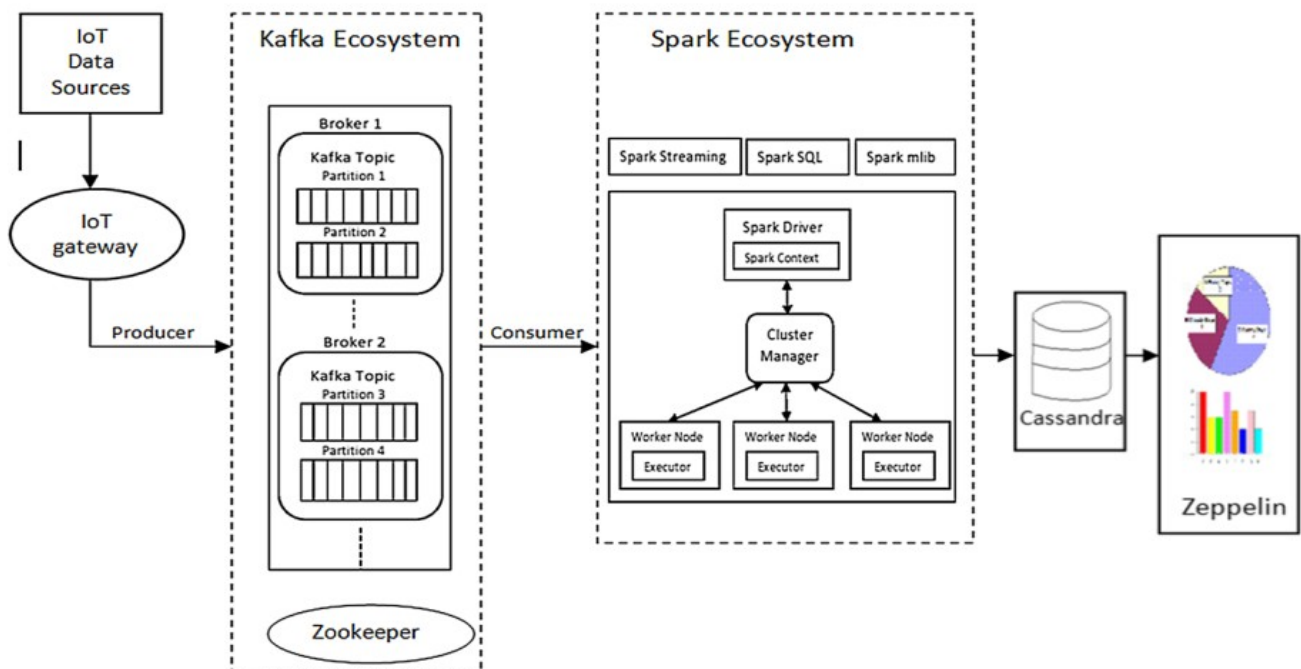


Figura 3: Arquitectura de un proyecto usando Kafka y Spark [4].

1.3.2. Machine learning based distributed big data analysis framework for next generation web in iot [5]

Se propone una solución usando una nube híbrida utilizando computación en el edge para obtener la escalabilidad necesaria para el procesamiento de los datos.

El análisis se realiza en la nube en un único centro de datos.

Los datos, obtenidos de los dispositivos IoT, son procesados en el edge, donde aplican los algoritmos de PCA, para la extracción de característica, y K-medias para el escalado.

En la nube se emplea un algoritmo ELM para clasificar y analizar los datos.

1.3.3. A heterogeneous IoT data analysis framework with collaboration of edge-cloud computing: Focusing on indoor PM10 and PM2.5 status prediction [6]

Propone la predicción de partículas PM10 y PM2.5 en interiores usando un sensor por cada habitación y otro en el exterior para recoger diferentes mediciones ambientales.

Este proyecto expone una arquitectura de nube híbrida, en la que la nube es la encargada de entrenar varios modelos, seleccionar los más adecuados para cada dispositivo en el edge (según los sensores con los que cuenta el dispositivo) y distribuir estos modelos.

En el edge se utilizan los modelos recibidos de la nube para inferir el estado ambiental futuro a partir de los datos en tiempo real que capturan.

1.4. Objetivo del sistema

El objetivo de este proyecto es mejorar la eficiencia del uso de la energía empleada en los sistemas de regulación ambientales (calefacción, aire acondicionado, iluminación) en las aulas de la ETSISI en el Campus Sur.

Como objetivos más específicos se tienen los siguientes:

- Obtener información de redes de sensores IoT instaladas en aulas de la ETSISI, para lo que se contará con el respaldo del grupo de investigación SYST y la Dirección de la ETSISI.
- Utilizar Inteligencia colectiva (participatory crowdsensing) para recoger la experiencia de los alumnos en las aulas.
- Almacenar en la nube la información recogida para que pueda ser analizada.
- Realizar un análisis de los datos recogidos y obtener recomendaciones de mejora del rendimiento energético de los sistemas de regulación ambiental.
- El proyecto además almacenará los valores de los sensores recibidos relativos a la confortabilidad en las aulas, los resultados y recomendaciones generadas del análisis de los datos y las acciones siguiendo estas recomendaciones en una Blockchain para su consulta por los actores interesados.
- Visualizar los datos de los sistemas monitorizados.

1.5. Metodología empleada

1.5.1. Primera fase:

Se nos presentó el proyecto SioTCOM.Ambiente1.

- a) Accedimos a la información del trabajo realizado en el proyecto.
 - La red de sensores/actuadores/concentradores y los controladores.
 - El proyecto Green Smart Campus [7], de Jaime Vidal Coleman que recopila la información de crowdsensing.
- b) Obtuvimos los requisitos del SioTCOM.Ambiente1.
- c) Analizamos los requisitos y los sistemas existentes.
- d) Elegimos el alcance que va a tener el trabajo.
- e) Dividimos los objetivos entre los dos miembros del equipo.

Hito 1: División del trabajo que corresponde a este proyecto: El proceso, análisis y visualización de los resultados y su almacenamiento en blockchain.

1.5.2. Segunda fase:

En el proyecto SioTCOM se utiliza como fuente externa de datos Firestore para el almacenamiento de los datos enviados por los usuarios mediante la aplicación móvil. Además, el nuevo sistema debe recoger los datos de sensorización de una base de datos Mongo desarrollada en el TFM **Optimización del consumo energético en las aulas de la ETSISI mediante inteligencia colectiva: Sensorización de las aulas, análisis de datos y decisiones de actuación** que complementa a este TFM.

- a) Elegimos el tipo de análisis final y su visualización.
- b) Diseñamos una arquitectura para el sistema, descartando su dependencia de servicios en la nube.

Hito 2.1: Decidimos el diseño del sistema y los módulos que lo compondrán: uno por cada fase del tratamiento del flujo de datos.

- c) Investigamos y estudiamos las tecnologías existentes para adoptar las más adecuadas.

Hito 2.2: Nos decidimos por un stack en python, para aprovechar las populares librerías de manipulación de datos y machine learning y su interoperatividad con todo tipo de sistemas y frameworks.

1.5.3. Tercera fase:

Se inicia la implementación del sistema.

Hito 3.1: Completamos el módulo de adquisición de datos.

Hito 3.2: Completamos el módulo de procesamiento de datos.

Hito 3.3.1: Probamos la computación distribuida de la carga y el procesamiento de los datos.

Se realizaron varios prototipos:

- utilizando un framework de programación reactiva.
- utilizando el framework dask, con procesos del SO como workers.
- utilizando el framework dask, con máquinas virtuales como workers.
- utilizando pandas (numpy)
- utilizando la librería modin (que usa dask y pandas) con procesos del SO como workers.
- utilizando joblib para usar procesos del SO.

Hito 3.4: Completamos la API del servidor de datos.

Hito 3.5: Implementación de un sistema de caché de los resultados en una base de datos no relacional.

Hito 3.6: Completamos el módulo de análisis de datos.

Hito 3.7: Completamos el módulo de visualización de datos.

Hito 4: Testeo e Integración de los módulos.

Se queda fuera el almacenamiento en blockchain.

1.6. Particularidades del Sistema

Este proyecto sigue la arquitectura general para el análisis que proponen los proyectos estudiados, pero a diferencia de estos no tiene ninguna dependencia con una nube pública o privada.

Comparando con el resto de proyectos, este proyecto presenta las siguientes particularidades:

- **Obtener los datos capturados del sistema IoT**: Aunque en el sistema [4] se utilizan también todos los datos usando Dataframes distribuidos, este proyecto no utiliza una capa de streaming para la captura de datos, pero utiliza directamente los streams de datos que proporcionan cada fuente.
- **Procesar los datos de IoT**: A diferencia de los sistemas [5] y [6] en este trabajo el proceso no se realiza en el edge, sino que como en [4] se realiza en un servicio centralizado. Las ventajas de ejecutar la computación en servidores dedicados es la alta disponibilidad de recursos de computación, con lo que se aumenta el rendimiento y se reducen los costes y el poder acceder al conjunto de datos completo.
- **Almacenar los datos procesados**: Al igual que en [4] se almacenan los datos procesados y estructurados como un Dataframe, pero al contrario que en los proyectos analizados su almacenamiento lo utilizamos como una caché que servirá para agilizar cargas posteriores. Con esta forma de almacenamiento, sin definir ningún TTL, se pueden recuperar datos que sirvan de histórico de operaciones.

1.7. Estructura de este documento

El segundo capítulo de este documento lo dedicamos a describir las fases de las que consta el procesado y análisis de datos para sistemas IoT, los conceptos básicos sobre crowdsensing e inteligencia colectiva, las diferentes arquitecturas de computación en la nube y a resumir las plataformas que ofrecen servicios específicos para el análisis de datos generados desde sistemas IoT.

A continuación, en el tercer capítulo explicamos el sistema desarrollado, detallando los módulos que lo componen y resaltando algunos detalles de su funcionamiento.

Describimos los resultados obtenidos en el cuarto capítulo y comentamos dichos resultados en el capítulo siguiente (quinto). En el sexto capítulo indicamos el impacto social y medioambiental resultado del desarrollo y la aplicación de este proyecto. En el séptimo capítulo mostramos la planificación y presupuesto. Por último en el octavo capítulo exponemos las conclusiones.

Capítulo 2 : Contexto

2.1. Internet of things (IoT)

“Internet de las cosas (IoT) describe la red de objetos físicos ("cosas") que llevan incorporados sensores, software y otras tecnologías con el fin de conectarse e intercambiar datos con otros dispositivos y sistemas a través de Internet. Estos dispositivos van desde objetos domésticos comunes hasta herramientas industriales sofisticadas. Con más de 7 mil millones de dispositivos IoT conectados en la actualidad, los expertos prevén que este número aumentará a 10 mil millones para el 2020 y 22 mil millones para el 2025.” [8]

El tratamiento completo de datos provenientes de IoT se divide en 3 fases, representadas en la figura 4:

- procesado,
- fusión,
- y el análisis de los datos:

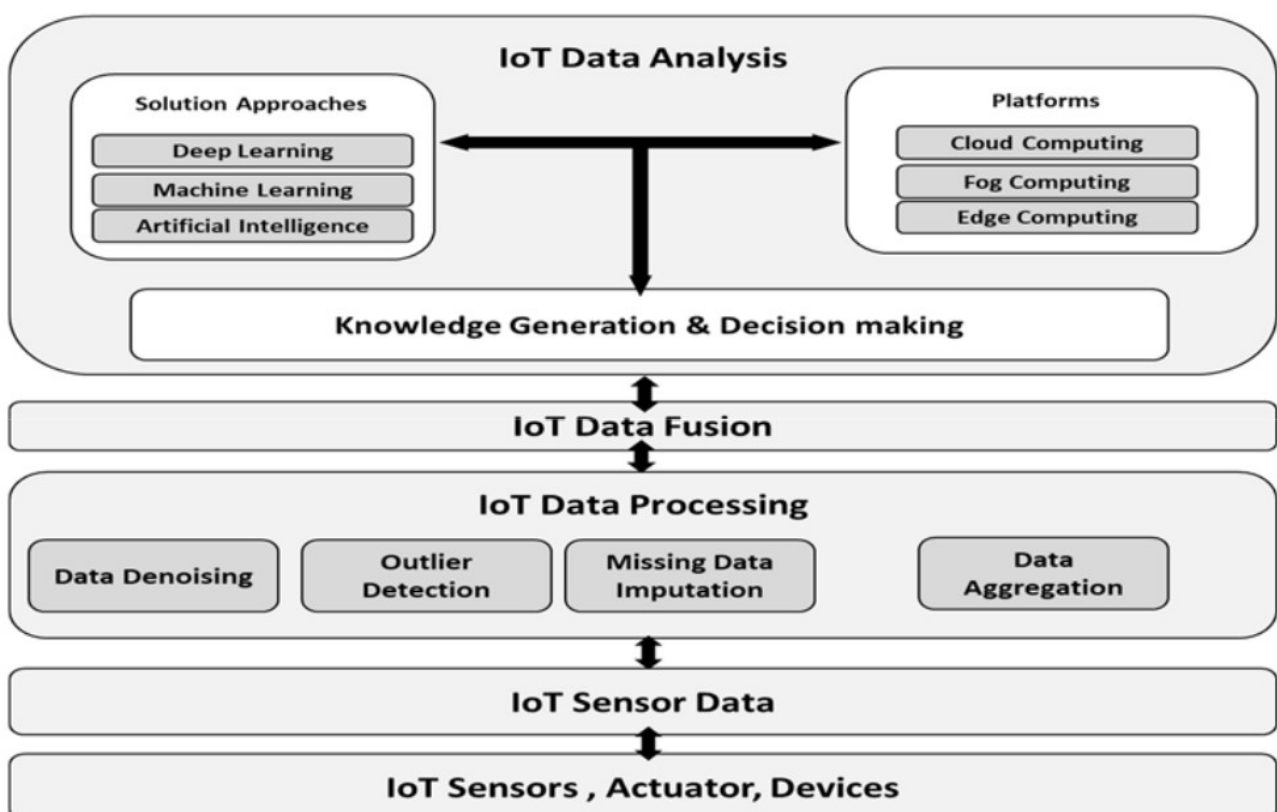


Figura 4: Arquitectura básica de procesamiento de un sistema IoT [8]

El **procesado** de los datos intenta eliminar posibles errores e inconsistencias, utilizando una o más de las siguientes funciones:

- *Filtrar el ruido*: parte de los datos recibidos en un sistema IoT son erróneos. Consiste en eliminar los datos que son considerados como ruido transformando la señal imponiendo un umbral.
- *Completar datos perdidos*: los algoritmos que analizan los datos esperan un conjunto de datos completo.
- *Detección de anomalías*, producidas por factores externos en las redes de sensores, para detectarlas se utilizan sistemas de votación, clasificadores o PCA.
- *Data aggregation*, que generará resúmenes de los datos para poder ser empleados en el análisis estadístico.

En la fase de **fusionado** las lecturas de varios sensores son combinadas para tener una mayor precisión, mejor fiabilidad, poder medir estados inmensurables o incrementar el área medida.

El mayor reto para obtener un **análisis** correcto es conseguir la escalabilidad necesaria para el tratamiento del volumen de datos manejado y conseguir unos datos fiables de las fases previas.

Tras el **análisis** se espera obtener el conocimiento necesario para la toma de decisiones. En esta fase se emplean técnicas de extracción del conocimiento como lo son las de *machine learning* y *deep learning*.

2.2. Crowdsensing

En la actualidad hay más de 6.000 millones de usuarios (contratos) de dispositivos móviles y se espera que en 2024 se llegue a los 7.000 millones [9].

El crowdsensing es una técnica en la que un gran grupo de individuos con dispositivos móviles capaces de detectar y computar (como smartphones, tabletas, wearables) comparten colectivamente datos y extraen información para medir, mapear, analizar, estimar o inferir (predecir) cualquier proceso de interés común [1].

El crowdsensing tiene dos modalidades:

- **Participatory crowdsensing**, en el que los usuarios participan voluntariamente contribuyendo información.
- **Opportunistic crowdsensing**, en el que la información es recogida y enviada automáticamente sin la intervención del usuario, a veces sin su consentimiento[1].

2.3. Inteligencia colectiva

La inteligencia colectiva es una forma de inteligencia que surge a partir de la colaboración de diversos individuos. En el mundo actual, altamente interconectado, aparece junto con las nuevas maneras de relacionarse. Un ejemplo son los proyectos o eventos de edición colaborativa [10].

2.4. Objetivos de Desarrollo Sostenible

Son 17 objetivos globales, aprobados por todos los miembros de la ONU en 2015, como parte de la Agenda 2030 para el Desarrollo Sostenible², sus objetivos son acabar con la pobreza, proteger el planeta y mejorar las vidas y perspectivas de las personas de todo el mundo.

² Resolución de la ONU aprobando la Agenda 2030 para el Desarrollo sostenible en: https://www.un.org/ga/search/view_doc.asp?symbol=A/RES/70/1&Lang=S

Los objetivos propuestos son:

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

Cada objetivo está compuesto por unas metas que se deben cumplir y mantener para el 2030.

2.5. Arquitecturas en la Nube[11]

2.5.1. Public Cloud.

En este tipo de nube, los recursos que contiene (como servidores y almacenamiento) son propiedad del negocio o la organización y están operados por un proveedor de servicios de la nube externo que los ofrece a través de internet.

Sus ventajas son:

1. Menor coste (no se necesita adquirir ningún equipamiento).
2. Sin mantenimiento, el proveedor realiza el mantenimiento.
3. Escalabilidad casi ilimitada.
4. Alta fiabilidad.

2.5.2. Private Cloud.

Consiste en servicios en la nube usados en exclusiva por un negocio u organización. El equipamiento necesario puede estar situado en el data-center de la organización o estar hospedado por un servicio externo.

Sus ventajas son:

1. Mayor adaptabilidad a los requisitos del usuario.
2. Al no tener que compartir sus recursos se puede conseguir mayores niveles de control y privacidad.
3. Mejor escalabilidad que las infraestructuras basadas en on-premise.
4. Acerca la computación a los datos.

2.5.3. Hybrid Cloud.

En este tipo se combinan las infraestructuras de on-premise y de nube pública. Este tipo de arquitectura permite el movimiento de aplicaciones entre los dos entornos.

Sus ventajas son, en resumen, las de tener una nube pública y una infraestructura on-premise:

1. Control, se puede utilizar la infraestructura privada para los elementos más delicados o sobre los volúmenes de trabajo que requieran de menor latencia.
2. Flexibilidad, al poder contar para adaptarse a las cargas de trabajo con los recursos extra ofrecidos por la nube pública.

2.5.4. On-Premise.

No es una arquitectura en la nube, se refiere al uso de servidores y del resto de infraestructura informática pertenecientes al negocio u organización.

Sus ventajas son:

1. Control del entorno, reduciendo el nivel de latencia y asegurando que los contenidos más delicados permanecen dentro de la compañía.

2.6. Plataformas de análisis de datos IoT

El reto del análisis de los datos adquiridos desde sistemas IoT proviene de las características de los propios datos. Se requiere el procesado de un gran volumen de información heterogénea, normalmente no estructurada, de conjuntos de datos no siempre exactos, válidos o completos.

Es por ello que su análisis no es óptimo con las soluciones tradicionales de análisis e inteligencia empresarial que operan sobre datos estructurados y se utilizan técnicas de Big Data mediante servicios en la nube para soportar la infraestructura requerida.

A continuación se exponen algunas plataformas que ofrecen análisis específico de datos provenientes de sistemas IoT.

2.6.1. AWS IoT Analytics de Amazon

Es un servicio en nube pública ofrecido por Amazon, que integra AWS IoT Core para la obtención de datos o recibirlos en batch de cualquier fuente por medio de HTTP.

El proceso permite la limpieza y filtrado, la transformación y el enriquecimiento de los datos recibidos. AWS IoT Analytics detecta datos faltantes disparando eventos AWS IoT Events que pueden hacer ejecutar código para rellenar los datos.

Las series temporales resultado del proceso y los datos en crudos que se reciben son almacenados en la nube.

El análisis se puede realizar mediante consultas SQL, machine learning, análisis de series temporales o clasificación estadística.

El flujo de los datos básico que proponen se puede consultar en la imagen 5.

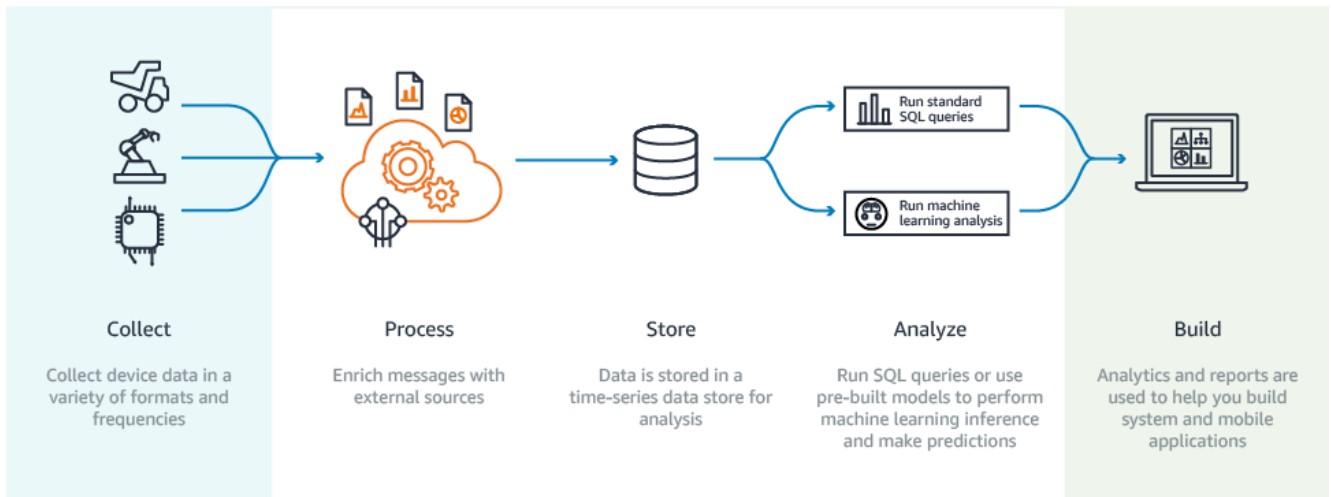


Figura 5: Flujo básico de análisis de datos con AWS IoT Analytics [15].

En el ejemplo proporcionado por AWS el servicio para procesar 10 Mb/Mes de información por 1000 dispositivos en 4 flujos costaría 3,436 € al mes.

2.6.2. IoT Edge Analytics de Crosser

Se ofrece como un motor de procesamiento de datos que se puede instalar donde se necesite, habilitando el procesamiento y análisis en el edge.

La infraestructura necesaria para soportar los flujos de datos requeridos por los motores desplegados puede ser proveída por una nube pública, privada o híbrida.

2.6.3. Azure de Microsoft

Es un servicio en nube pública ofrecido por Microsoft.

Ofrece los servicios IoT Central y IoT Hub para la comunicación de datos con los dispositivos, usando el resto de servicios de Azure para realizar el almacenamiento y procesamiento de los datos. Por ejemplo Azure Stream Analytics puede ser empleado para el procesamiento de flujos de datos o utilizar el análisis en una nube híbrida con Azure IoT Edge, como se muestra en la imagen 6.

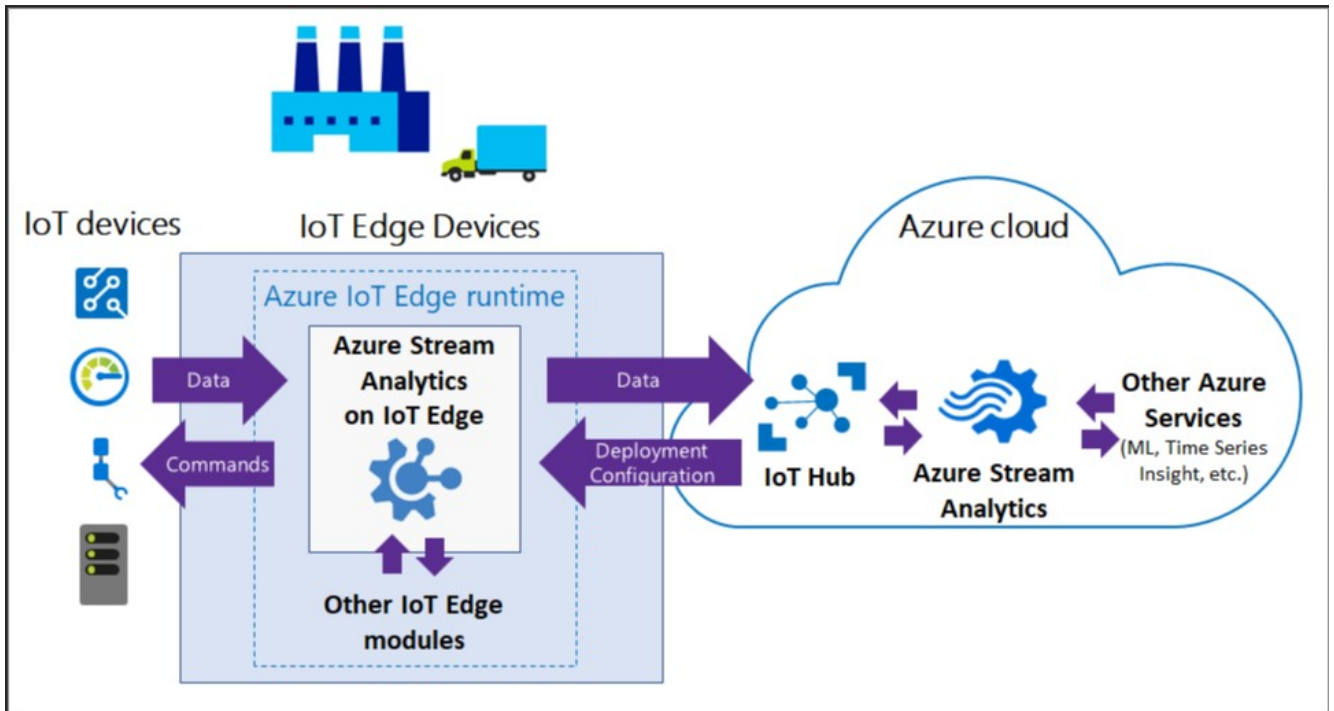


Figura 6: Esquema de análisis de datos de arquitectura híbrida que con Stream Analytics de Azure [14].

2.6.4. Kaa [12]

Ofrece una plataforma IoT, integrada con Open Distro Elasticsearch. Un flujo para el análisis es mostrado en la figura 7:

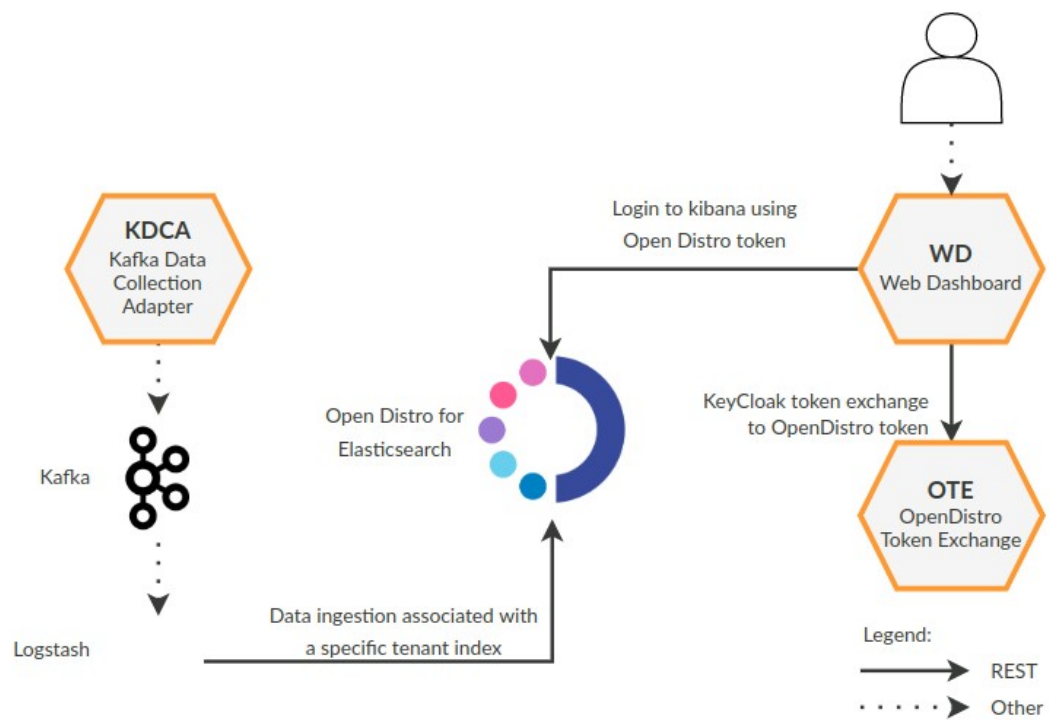


Figura 7: Análisis de datos IoT con Kaa [12]

El precio es por dispositivo conectado, costando 1000 dispositivos en el plan más asequible 250,0 € al mes.

Capítulo 3 : Sistema desarrollado

El trabajo recoge información ambiental proveniente las redes de nodos sensores, actuadores, concentradores y controladores instalados en los espacios de la UPM y, mediante un proceso de participatory crowdsensing, las sensaciones de la comunidad universitaria relacionadas con las mediciones ambientales.

Se plantea el problema de obtener la relación entre las mediciones reales y la percepción que los usuarios tienen de las características ambientales en los espacios sensorizados. Para ello se debe analizar y visualizar los datos generados por el sistema de IoT de manera que los resultados obtenidos sirvan para decidir nuevas estrategias en el consumo energético.

A partir de los requisitos desarrollados en el proyecto SioTCOM, de los que contábamos con una primera versión, hemos seleccionado un subconjunto de ellos y diseñado e implementado un sistema modular, configurable y escalable que permite:

- Mostrar el valor de diferentes tipos de mediciones (temperatura, humedad, luminosidad, nivel de co2, ruido y movimiento) junto a la valoración de los usuarios de cada sensación.
- Mostrar mediciones y sensaciones en cualquier rango de tiempo y agrupar los eventos en periodos de tiempo arbitrarios³.
- Las reglas que asignan sensores y sensaciones a tipos de medición puede ser editada en un fichero de configuración.
- Devolver, tras el análisis de los datos, un modelo que puede ser utilizado en el control de reguladores ambientales.

El sistema está estructurado de varios módulos que se corresponden con cada una de las etapas del flujo de datos mostrado en la figura 8.



Figura 8: Flujo de datos en el sistema

3.1. Módulo de captura de datos

Es el módulo encargado de acceder a los servicios externos para recuperar la información.

³ Admite cualquier alias válido a un offset de pandas que represente un periodo.
https://pandas.pydata.org/pandas-docs/stable/user_guide/timeseries.html

Los datos a los que accede provienen de:

- Un sistema de sensorización, una red inalámbrica compuesta por sensores/actuadores y concentradores y servidores que recogen y almacenan la información referente a las medidas físicas. Los datos son los recogidos y almacenados usando el proyecto de TFM de Ana María Chacón Villaseñor **Optimización del consumo energético en las aulas de la ETSISI mediante inteligencia colectiva: Sensorización de las aulas, análisis de datos y decisiones de actuación.**
- Y un sistema para la recogida y almacenamiento de las sensaciones que los usuarios han compartido usando la aplicación móvil del proyecto **Green Smart Campus** [7].

Estos sistemas devuelven conjuntos de datos sin ninguna normalización. Un ejemplo de como se estructuran se puede ver en la figura 9.

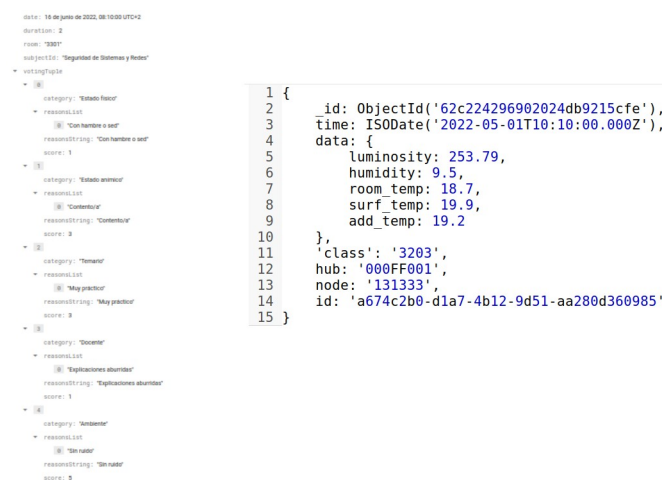


Figura 9: Ejemplo de datos capturados.

La carga de datos, esquematizada en la figura 10, es igual para todos los orígenes de datos y comienza con una llamada desde la API para leer y aplanar los datos originales y que cada registro contenga solo una entrada de medición o valoración.

Detalles de implementación:

En este paso, el más costoso del sistema, hemos aplicado:

- **Paralelización:** la adquisición de datos se descompone por rangos de tiempo. A cada rango se le asigna una tarea de carga que se ejecutará de forma paralela.
- **Uso de caché:** Se comprueba, comparando los parámetros de cada tarea con la caché almacenada. Si existen en caché se devuelve el contenido de la caché, si no se lanza la tarea.

La paralelización se realiza asignando un proceso del SO a cada tarea de carga. Para el sistema de caché utilizamos memoria persistente usando una base de datos Mongo. De esta forma la información puede ser compartida entre los procesos de SO.

Los datos recibidos desde cada tarea son fusionados con el resto y aplanados para ser devueltos como un *DataFrame* de *pandas*.

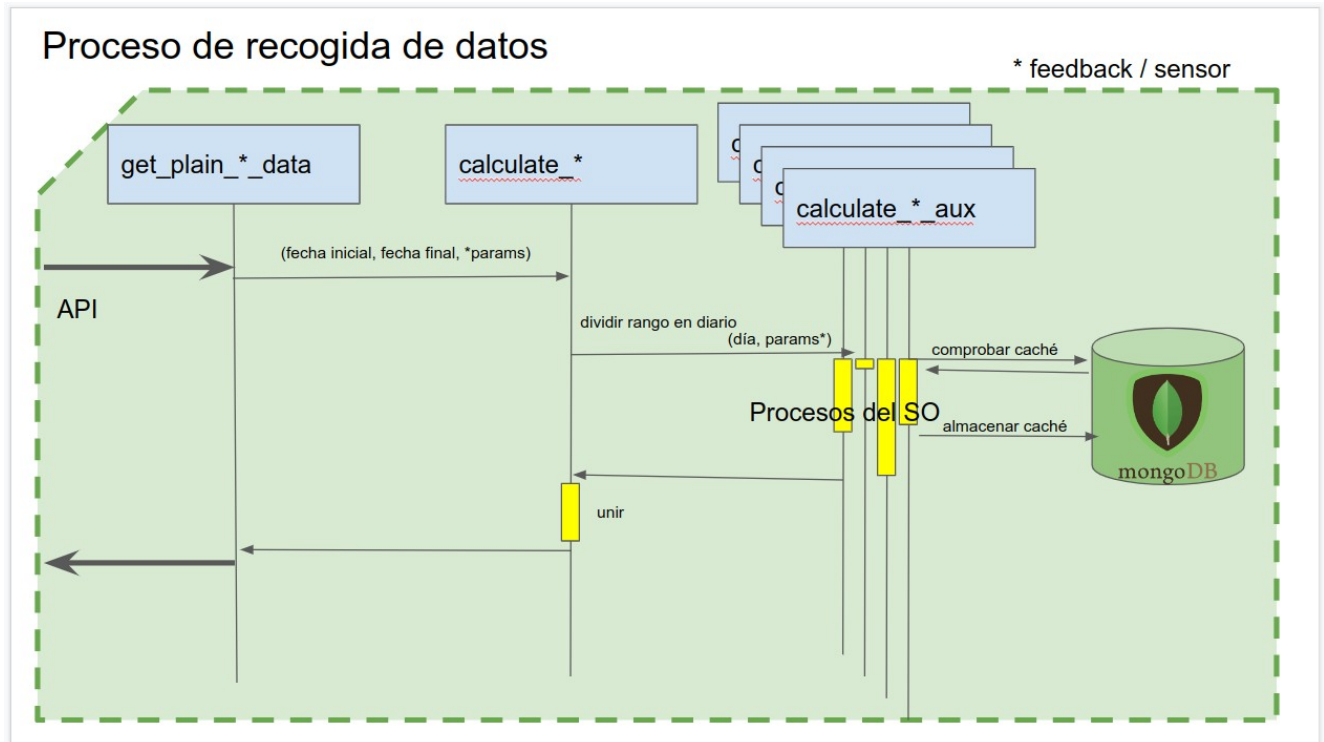


Figura 10: Carga de datos

3.2. Módulo de procesamiento de datos

Este módulo realiza las operaciones de filtrado, reestructuración y agrupación de la manera que se muestra en la figura 11.

A partir de la estructura obtenida de la carga de datos:

- Filtramos la información, eliminando entradas erróneas o innecesarias.
- Agrupamos la información en intervalos de tiempo, combinando la información proveniente de sensores del mismo tipo y siempre diferenciando el tipo de medición y el espacio de los que procede.
- Calculamos y almacenamos para cada intervalo de tiempo el número muestras, media, máximo, mínimo y desviación estándar que hemos obtenido de su agregación.
- Después, rellenamos los datos necesarios para obtener múltiples series temporales completas, al menos una por tipo de medición y espacio, con un formato como el que aparece en la figura 12.

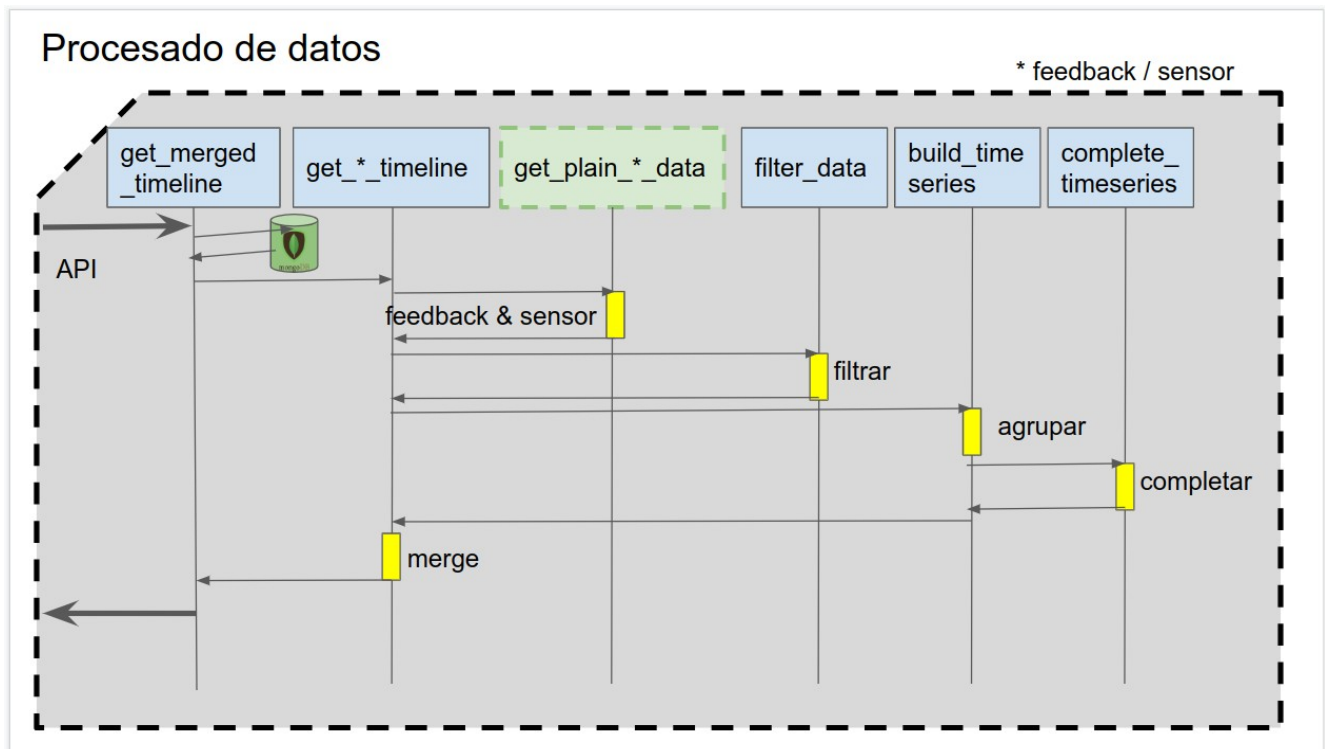


Figura 11: Proceso de datos

Detalles de implementación:

Se completan los datos faltantes con la media de los datos que corresponden al mismo espacio y medición o valoración.

Las series que se obtienen tienen como rango temporal el mínimo y máximo encontrados dentro del rango pedido. Es decir, no se generan (completan) datos fuera del rango de tiempo encontrado.

Antes de calcular las series se comprueba si se encuentran en caché, para devolver su contenido o si lo se encuentran, calcularlas y almacenarlas.

Durante todo el proceso se utilizan las operaciones que proporciona la librería *pandas*. Hemos evitado usar bucles externos en las operaciones con los datos para intentar aprovechar todas las optimizaciones posibles.

	dt	measure	room	value_min	value_mean	value_max	value_std	value_count
0	2022-07-07 17:00:00	co2	4405	1113.00	1113.000000	1113.00	10.803711	1.0
1	2022-07-07 17:00:00	humidity	4405	1.00	20.426667	26.30	9.285149	12.0
2	2022-07-07 17:00:00	luminosity	4405	35.83	86.286364	138.33	43.579147	11.0
3	2022-07-07 17:00:00	movement	4405	0.00	0.000000	0.00	0.000000	10.0
4	2022-07-07 17:00:00	temperature	4405	23.90	24.508824	29.20	0.886989	34.0
5	2022-07-07 18:00:00	humidity	4405	1.00	1.000000	1.00	10.803711	1.0
6	2022-07-07 18:00:00	luminosity	4405	142.50	142.500000	142.50	10.803711	1.0
7	2022-07-07 18:00:00	temperature	4405	24.10	24.406667	24.59	0.267270	3.0
8	2022-07-07 18:00:00	co2	4405	1113.00	1113.000000	1113.00	10.803711	1.0
9	2022-07-07 18:00:00	movement	4405	0.00	0.000000	0.00	0.000000	10.0

Figura 12: Datos de los sensores tras el procesamiento agrupados en intervalos de una hora.

3.3. Módulo de almacenamiento

Este módulo almacena los datos procesados de forma persistente. Utilizamos la información como caché para agilizar el acceso a los datos.

Detalles de implementación:

Las operaciones previas son lo suficientemente costosas como para que el uso de almacenamiento persistente en una base de datos (Mongo), en lugar de en memoria (Redis), esté justificado.

El servicio encargado de almacenar la caché puede ser configurado para ser independiente del usado como origen de datos.

Utilizamos dos cachés: una que guarda los datos procesados y otra que guarda los datos de las series temporales.

El formato de almacenamiento sigue el que ofrece la librería *cachier*. Se puede ver un ejemplo en la figura 14. Los parámetros de la función y la salida obtenida se almacenan como la codificación en base64 del pickle (marshalling) de los parámetros de entrada. Un proceso para extraer información de esta codificación se muestra en la figura 13.


```
>>> d = base64.decodestring(b'gASVtAAAAAAAAAAojAhjYXRlZ29yeZSMCEftYmlbnRlIaUjAxLmRfZGF0ZXRpbnWUjAhkYXRldGltZSMCGRhVG0aW1lJ0UQwoH5gMCFz
s7D0I/LIWUpSGlWkZ3JvdXBfdHlwZSMEdyY3VwX2tpbmRfc2Vuc29yIaUjAxpbnRfZGF0ZXRpbnWUaAZDcGfmAwIAAAAAACUhzRSIAUjAdtZWfzdXJlLE6GIwEcm9vbZROhp
iR0lC4=')
>>> d
b'\x00\x04\x95\xb4\x00\x00\x00\x00\x00\x00(\x8c\x08category\x94\x8c\x08Ambiente\x94\x86\x94\x8c\x0cend_datetime\x94\x8c\x08datetime\x94\x
8c\x08datetime\x94\x93\x94C\n\x07\xe6\x03\x02\x17;\x0fB7\x94\x85\x94R\x94\x86\x94\x8c\x0cngroup_type\x94\x8c\x11group_kind_sensor\x94\x86\x94
\x8c\x0cini_datetime\x94h\x06C\n\x07\xe6\x03\x02\x00\x00\x00\x00\x00\x94\x85\x94R\x94\x86\x94\x8c\x07measure\x94N\x86\x94\x8c\x04room\x9
4N\x86\x94t\x94.'
>>> pickle.loads(d)
(('category', 'Ambiente'), ('end_datetime', datetime.datetime(2022, 3, 2, 23, 59, 59, 999999)), ('group_type', 'group_kind_sensor'), ('ini_d
atetime', datetime.datetime(2022, 3, 2, 0, 0)), ('measure', None), ('room', None))
```

Figura 13: Decodificación de la información almacenada mediante cachier

```
{
  _id: ObjectId('62ccb0c5a47021b4c9bbf8bb'),
  func: '.analysis.process.fetcher.calculate_feedback_aux',
  key: BinData(0, 'gASVtAAAAAAAAAAojAhjYXRlZ29yeZSMCEftYmlbnRlIaUjAxLmRfZGF0ZXRpbnWUjAhkYXRldGltZSMCGRhVG0aW1lJ0UQwoH5gMCFz
s7D0I/LIWUpSGlWkZ3JvdXBfdHlwZSMEdyY3VwX2tpbmRfc2Vuc29yIaUjAxpbnRfZGF0ZXRpbnWUaAZDcGfmAwIAAAAAACUhzRSIAUjAdtZWfzdXJlLE6GIwEcm9vbZROhp
iR0lC4='),
  being_calculated: false,
  stale: false,
  time: ISODate('2022-07-12T01:22:45.533Z'),
  value: BinData(0, 'gASVkaMAAAAAACMEXBhbmRhcy5jb3JlLmZyYW1lIiwJRGF0YUZYyYw1lJ0UkYGUfZ0ojARfbWdyIiwecGFuZGFzLmNvcuUuaW50ZXJyYw
'),
}
```

Figura 14: Formato de almacenamiento usado por cachier

3.4. Módulo de análisis

El objetivo es obtener un clasificador que tome las mediciones en un momento dado y determine las sensaciones que los usuarios tendrán.

Para la obtención de este modelo utilizamos regresión logística. Necesitamos preparar los datos antes de aplicar el algoritmo, para obtener unas estructuras como los que se muestran en las figuras 15 y 16: Tenemos varias series temporales independientes con la información de sensores y valoraciones y debemos obtener series temporales homogéneas donde cada muestra debe contener un valor para todas las variables dependientes e independientes que vamos a usar.

	dt	room	co2	humidity	luminosity	noise	temperature
0	2022-03-01	1001	0.000000	0.000000	0.000000	0.0	0.000000
1	2022-03-01	3104	0.000000	0.000000	0.000000	0.0	0.000000
2	2022-03-01	3203	6.096491	4.838596	244.515614	0.0	15.808187
3	2022-03-01	3301	0.000000	0.000000	0.000000	0.0	0.000000
4	2022-03-01	3302	0.000000	0.000000	0.000000	0.0	0.000000
...
79	2022-03-12	3203	1.164706	3.750980	290.845294	0.0	-7.200000
80	2022-03-12	3301	0.000000	0.000000	0.000000	0.0	0.000000
81	2022-03-12	3302	0.000000	0.000000	0.000000	0.0	0.000000
82	2022-03-12	CIC-3	0.000000	0.000000	0.000000	0.0	0.000000
83	2022-03-12	CIC-4	0.000000	0.000000	0.000000	0.0	0.000000

84 rows × 7 columns

Figura 15: Variables independientes: se toman los valores medios de cada medición.

	dt	room	co2	humidity	luminosity	noise	temperature
0	2022-03-01	1001	4	3	1	3	2
1	2022-03-01	3104	3	3	4	2	2
2	2022-03-01	3203	3	3	3	3	2
3	2022-03-01	3301	4	3	3	4	2
4	2022-03-01	3302	2	3	3	4	2
...
79	2022-03-12	3203	2	3	3	3	2
80	2022-03-12	3301	4	3	4	4	2
81	2022-03-12	3302	3	3	3	3	2
82	2022-03-12	CIC-3	4	3	3	3	1
83	2022-03-12	CIC-4	3	3	4	2	2

84 rows × 7 columns

Figura 16: Variables dependientes: se toman las valoraciones medias y se redondean.

Empezamos fusionando los datos para obtener las correspondencias temporales y espaciales entre las mediciones de los sensores y las valoraciones de los usuarios.

A continuación moldeamos la estructura para obtener una variable independiente por cada tipo de medición y una variable dependiente de salida por cada tipo de valoración.

Debemos agrupar los valores medios de las valoraciones en sus niveles originales, ya que la regresión logística requiere que sus variables dependientes sean ordinales discretos. Filtramos las variables dependientes antes de realizar esta agrupación para que sus valores estén dentro de un rango. También debemos completar los datos faltantes puesto que, aunque las series temporales obtenidas tras el procesado son completas aún puede faltar alguna serie. Escalamos los datos.

Dividimos el conjunto de datos que hemos preparado en un conjunto de entrenamiento y otro de test.

Aplicamos el algoritmo sobre estos conjuntos varias veces: tantas como tipos de valoración estamos procesando. En cada iteración cambiamos la variable dependiente por uno de esos tipos.

El resultado obtenido al unir todas las iteraciones sobre el algoritmo es:

Varios modelos, uno por cada tipo de valoración, capaces de determinar la sensación que experimentarán los usuarios a partir de una entrada de datos con mediciones ambientales.

Junto a cada modelo devolvemos la media de la precisión y el error cuadrático medio obtenidos durante el entrenamiento.

Detalles de implementación:

Utilizamos la librería *scikit-learn* para usar su algoritmo de regresión logística.

Se puede modificar la ejecución del algoritmo desde la API eligiendo algunos parámetros para incrementar el aprendizaje o reducir el overfitting:

- la proporción de datos de entrenamiento y de test,
- el tipo de penalización usado para ajustar los coeficientes: L1 valor absoluto de los coeficientes, L2 cuadrado de los coeficientes o una combinación de ambos.
- C, inversa de la regularización, la proporción con la que se aplicará la penalización.
- El ratio para L1 cuando se usa la combinación de los dos tipos de penalización. Si L1 es 0 la penalización es equivalente a L2.

Cada modelo obtenido determina un nivel de valoración. Esto se consigue obteniendo un clasificador binario por cada nivel de valoración. Es decir en realidad obtenemos: $n.^{\circ}$ tipos valoración \times $n.^{\circ}$ niveles de valoración = $n.^{\circ}$ modelos

3.5. Módulos de Visualización

Proporcionamos dos módulos que sirven como interfaz para la consulta del sistema:

Una API REST con la que se pueden obtener las series temporales y los resultados de los análisis de forma que sirve de interfaz con otras aplicaciones.

Una dashboard para la visualización y comparación de los datos contenidos en el sistema.

Las dos interfaces son accesibles a través de sus propios servicios web.

3.5.1. API REST

Proporciona una interfaz web con la que se puede consultar el sistema.

Junto al servicio API ofrecemos su especificación OpenAPI (accesible en la URL ["/openapi.json"](#) del servicio), su documentación en un formato legible (accesible en ["/"](#), un ejemplo se ve en la figura 17) y un sistema de prueba y acceso (en la URL ["/test"](#) se encuentra una interfaz interactiva como la de la figura 18).

Los puntos de acceso de la API permiten:

Consultar del estado del sistema:

- La versión actual del sistema.

- La configuración actual del sistema.

Obtener el resumen de los datos mantenidos por el sistema:

- Habitaciones de las que se tiene información de medidas o sensaciones.
- Tipos de medidas configuradas de las que se puede realizar el análisis.

Conseguir los resultados de las fases de proceso:

- Series temporales de las sensaciones aportadas.
- Series temporales de las mediciones de los sensores.

Obtener los resultados de los análisis:

- Series temporales con las mediciones y las sensaciones relacionadas.
- Correlación entre los distintos tipos de medición y valoración.
- Modelos obtenidos mediante regresión logística.

Devolvemos el resultado del análisis como un grupo de modelos en un formato portable para que puedan implementarse en otros sistemas. Por ejemplo, pueden ser usados para actualizar condiciones de actuación en la red de sensores y actuadores. Junto a los modelos enviamos información adicional sobre el rendimiento obtenido durante su entrenamiento.

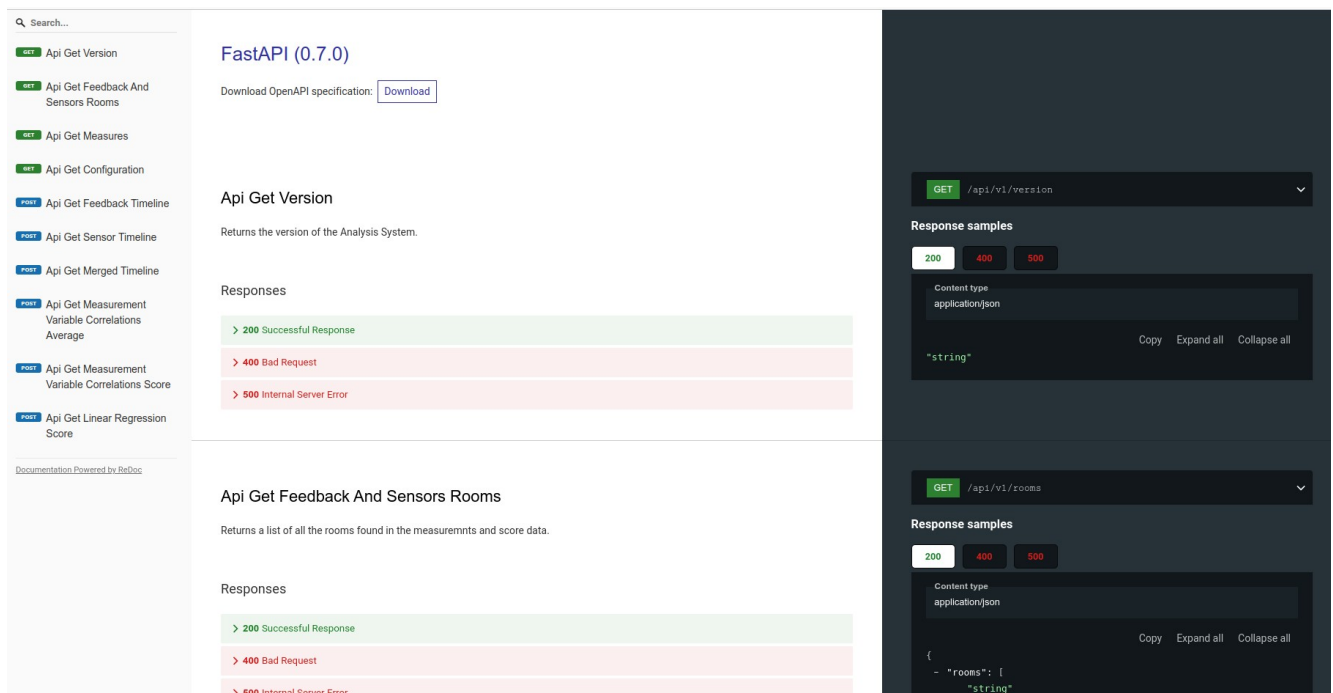


Figura 17: Web con la documentación de la API

3.5.2. Dashboard

El dashboard ofrece la posibilidad de consultar al sistema y visualizar las respuestas obtenidas de forma más amigable.

Utiliza la API REST para obtener los datos a visualizar, realizando la consulta con las preferencias que el usuario haya introducido. Por ejemplo, el usuario puede elegir el periodo de agrupación de los datos, pero solo son ofrecidas las opciones: horaria, diaria o mensual. Las opciones ofrecidas corresponde a los requisitos del proyecto SloTCOM y son también un compromiso con los datos reales. No tendría sentido ofrecer agrupaciones por debajo de la resolución de captura de los sensores o de la aportación de los usuarios.

Las gráficas con que el usuario puede interactuar para consultar los datos disponibles son:

- Series temporales de los distintos tipos de mediciones recogidas.
- Series temporales de los distintos tipos de votaciones obtenidas.
- Series temporales de las predicciones del último análisis.
- Diagrama de violín que relaciona la valoración con el valor para cada tipo de valoración.
- Resumen de las características de los modelos del último análisis.

The screenshot displays an API testing tool interface. At the top, a green bar shows the method 'POST' and the endpoint '/api/v1/feedback/timeline' with a description 'Api Get Feedback Timeline'. Below this, another green bar shows 'POST' and '/api/v1/sensorization/timeline' with the description 'Api Get Sensor Timeline'. A note states: 'Returns a timeline of average measurements for each measure and room.' The 'Parameters' section is empty, with 'No parameters' listed. The 'Request body' section is marked 'required' and has a dropdown menu set to 'application/json'. The request body contains a JSON object:

```
{  "ini_date": "2022-03-2",  "end_date": "2022-03-2",  "from": "10"}
```

 Below the request body are 'Execute' and 'Clear' buttons. The 'Responses' section at the bottom shows a 'Curl' command:

```
curl -X 'POST' \  'http://127.0.0.1:8080/api/v1/sensorization/timeline' \  -H 'accept: application/json' \  -H 'Content-Type: application/json' \  -d '{  "ini_date": "2022-03-2",  "end_date": "2022-03-2",  "from": "10"}
```

Figura 18: Web de pruebas de la API

Capítulo 4 : Resultados obtenidos

Hemos generado un conjunto de datos aleatorio, en el que se producen pequeños incrementos o decrementos entre mediciones excepto para la humedad para la que se ha generado un valor constante.

Empleando estos datos simulados hemos realizado tres pruebas sobre la API instalada sobre una máquina con i5-10210U CPU @ 1.60G, RAM de 8G y HD de 450 G y dos pruebas usando la dashboard.

En cada prueba hacemos tres peticiones a la API (para activar la caché). Las pruebas consisten en obtener las series temporales de los Sensores, la de las Valoraciones, su Unión y el resultado de utilizar esta Unión para obtener los modelos.

Con estas pruebas intentamos medir el rendimiento del sistema y las soluciones que propone el análisis.

4.1. Primera prueba

La primera prueba consiste en acceder a un conjunto de datos inexistentes. La respuesta es inmediata para todos los casos (Tabla 1).

Tabla 1: Acceso a la API para un conjunto vacío

Agrupación	Sensores	Valoraciones	Unión	Análisis
1D	0,079 s	0,082 s	0,150 s	0,135 s

4.2. Segunda prueba

Para la segunda prueba accedemos a un solo día (Tabla 2).

Tabla 2: Acceso a la API de los datos correspondientes a un único día

Agrupación	Sensores	Valoraciones	Unión	Análisis
1D	0,150 s	0,140 s	0,155 s	0,276 s

4.3. Tercera prueba

Para la tercera prueba accedemos a todos los datos mensuales (Tabla 3).

Tabla 3: Acceso a la API de los datos correspondientes a un mes

Agrupación	Sensores	Valoraciones	Unión	Análisis
1D	8,13 s	0,870 s	11,20 s	0,643 s

En esta tercera prueba los modelos devueltos tienen las características de Accuracy y MSE que se muestran en la tabla 4:

Tabla 4: Medidas del modelo generado para agrupaciones de 1D y 1M con los datos de un mes completo.

Agrupación	Valoración	Accuracy	MSE
1D	CO ₂	0.2	1.1
1D	Luminosity	0.1	2,3
1D	Noise	0.1	2,3
1D	Temperature	0,5	0,5
1D	Humidity	-	-
1H	CO ₂	0.79	0.45
1H	Luminosity	0.85	0.35
1H	Noise	0.81	0.37
1H	Temperature	0.88	0.11
1H	Humidity	-	-

4.4. Cuarta prueba

En esta prueba usamos la dashboard para cargar y visualizar los datos correspondientes a 1 mes agrupados de forma diaria (Figura 19), generar los modelos a partir de estos datos y obtener una comparativa de los datos usados con los predichos por el modelo.

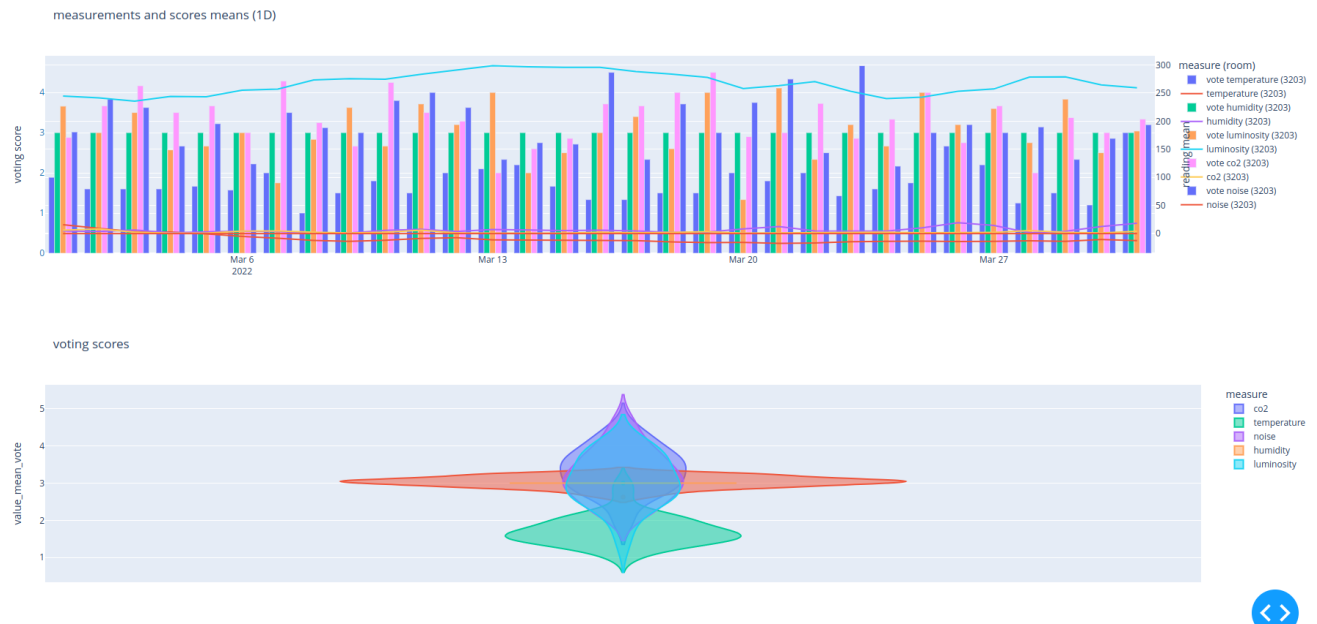


Figura 19: Datos de 1 mes completo agrupados de forma diaria y cargados en la dashboard.

Utilizando la agrupación diaria la precisión alcanzada es del 50% (figura 20).

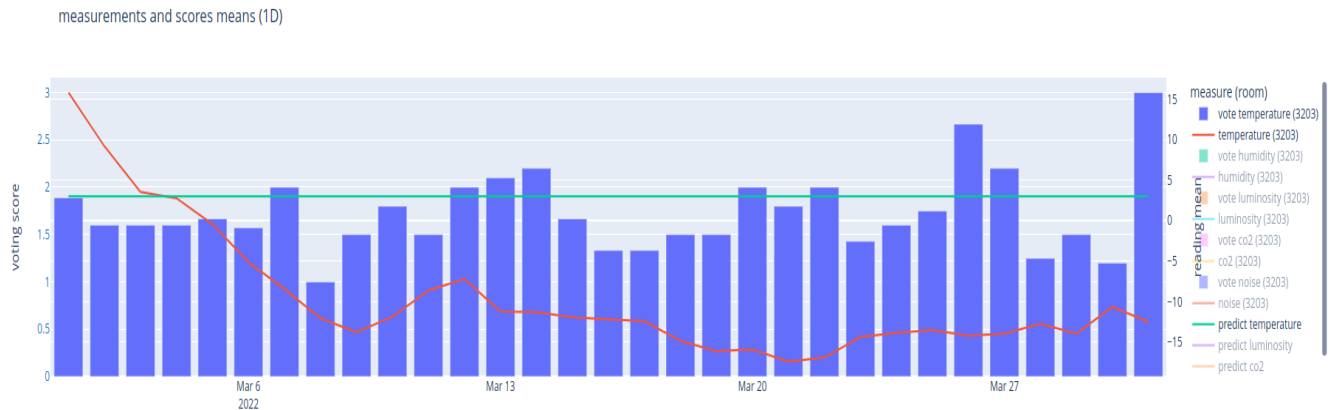


Figura 20: Predicción de sensación de temperatura, con agrupación diaria.

4.5. Quinta prueba

En esta prueba usamos la dashboard para cargar y visualizar los datos correspondientes a 1 mes agrupados por horas (Figura 21), generar los modelos a partir de estos datos y obtener una comparativa de los datos usados con los predichos por el modelo.



Figura 21: Datos de 1 mes completo agrupados por horas y cargados en la dashboard

Utilizando la agrupación por horas la precisión alcanzada es del 88% (figura).

Resultados obtenidos



Figura 22: Predicción de sensación de temperatura, con agrupación por horas.

Para esta última prueba los valores de los coeficientes obtenidos han sido:

Tabla 5: Coeficientes de la regresión lineal para la temperatura de los datos de 1 mes agrupados por horas

Valoración	CO ₂	Luminosity	Noise	Humidity	Temperature	Intercept
1	0.0661601911052862	-0.10991998290852042	-0.07045912929812452	0	-0.017460341505093055	-0.5076584280686977
2	0.0012850332135320786	-0.3084402736352727	-0.059006032484526516	0	-0.11628359233045982	2.250061460470762
3	-0.06744522431881814	0.41836025654379266	0.12946516178265172	0	0.13374393383555283	-1.7424030324020672

Capítulo 5 : Discusión del resultado obtenido

Los resultados obtenidos en los cálculos son los esperados.

La humedad, que se generó como un nivel fijo es correctamente ignorada al realizar la regresión logística.

Al aumentar el número y la precisión de las muestras, de diarias a cada hora, la precisión sube y el MSE disminuye como se espera usando el algoritmo de regresión logística.

Los resultados de acceso mediante la API también son los esperados.

La caché reduce significativamente el tiempo de acceso a los datos, de varios segundos a un tiempo inferior a un segundo.

Con los resultados obtenidos, hemos comprobado que el sistema es capaz de procesar y analizar los datos de forma eficiente, además de mostrar la información requerida.

Capítulo 6 :

Impacto social y medioambiental

La importancia de este proyecto reside en que ofrece la capacidad de analizar los datos recogidos en los espacios de los campus de la UPM con el objetivo de racionalizar el consumo de energía. Esto contribuirá significativamente a alcanzar los ODS propuestos. Entre las metas a las que este proyecto contribuiría directamente listamos:

11.3 De aquí a 2030, aumentar la urbanización inclusiva y sostenible y la capacidad para la planificación y la gestión participativas, integradas y sostenibles de los asentamientos humanos en todos los países.

12.2 De aquí a 2030, lograr la gestión sostenible y el uso eficiente de los recursos naturales.

Capítulo 7 : Planificación y presupuesto

Se ha desarrollado el trabajo en el tiempo previsto, con el siguiente presupuesto:

Tabla 6: Presupuesto

RECURSO	UNIDADES	€ / MES	TOTAL
Portátil	24 meses de vida útil	55,00	275,00
Instalaciones, consumibles y servicios	- Alquiler - Internet	922,00	5.532,00
Mano de obra ⁴	1	1.350,00	6.750,00
TOTAL			12.557,00

4 Media jornada

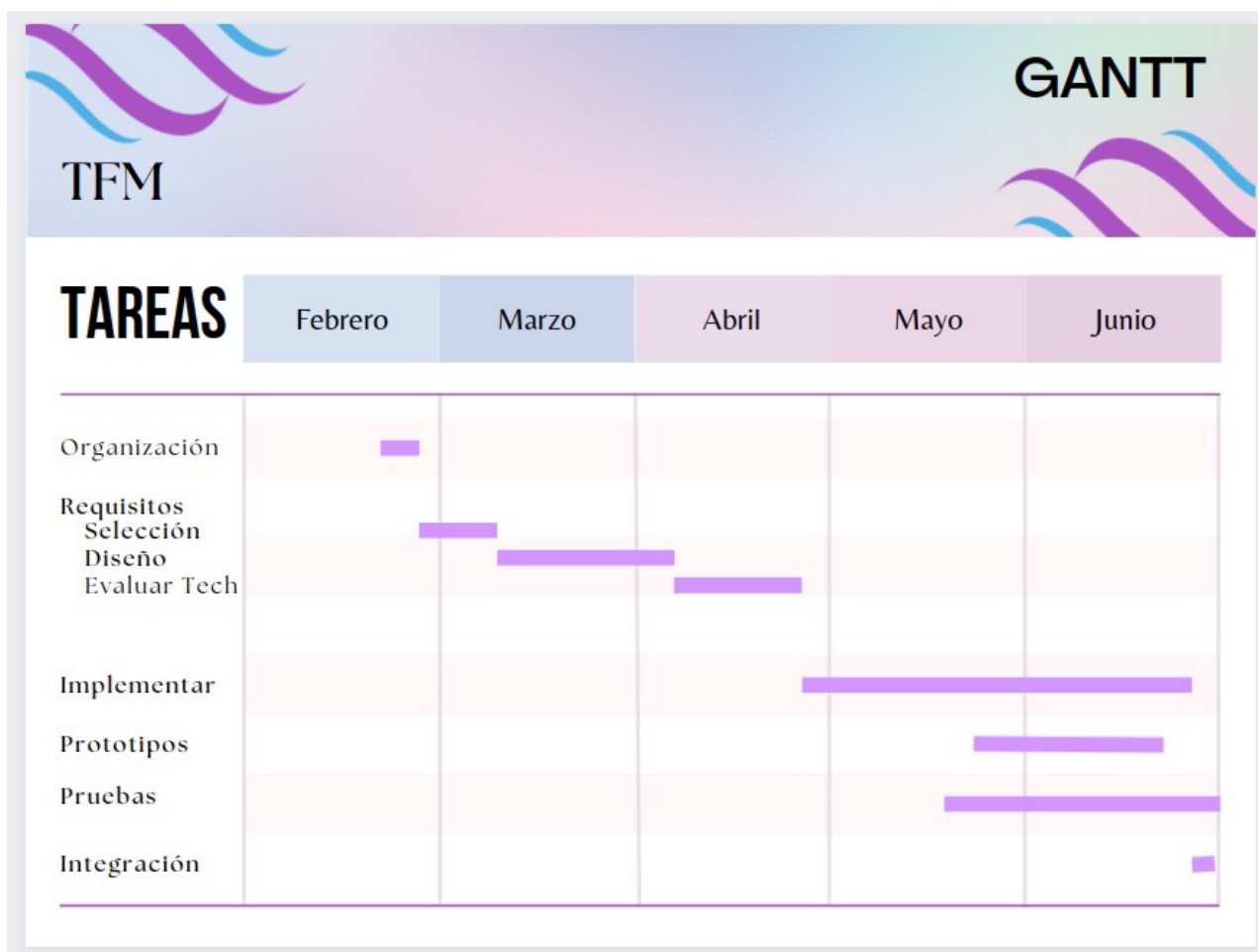


Figura 23: Gráfica Gnatt del desarrollo

Capítulo 8 : Conclusiones

Contar con los requisitos preparados desde el inicio nos ha agilizado mucho todo el desarrollo, y siempre nos han servido como referencia.

Una observación que hicimos fue que la parte más costosa de los cálculos se encontraba en el procesamiento inicial de la información, tras la carga. Por ello prototipamos soluciones basadas en programación reactiva primero y, después, en la distribución de la carga entre varios procesos. Por último añadimos el almacenamiento de datos intermedios de manera que fuera compartido por todos los procesos.

La regresión logística es una herramienta muy útil, ya que los coeficientes obtenidos son una fuente de información que permite analizar las relaciones entre las variables independientes y la clasificación en la que resultan su aplicación.

Para el prototipado e implementaciones finales fue de gran ayuda el lenguaje de programación Python, especialmente apto para estas tareas. La característica que más nos ha ayudado en la realización del trabajo es la riqueza de su ecosistema.

Descubrimos que la distribución de cargas de trabajo mediante Dask se puede llevar a cabo de manera sencilla, usando el propio lenguaje de programación. Incluso su despliegue en un clúster público o privado es sencillo. Pero la configuración personalizada para que el sistema se comporte de la forma deseada no lo es tanto. Si añadimos una capa de abstracción más, como modin, no resolvemos esta parte. Al final optamos por una capa de abstracción que permitía distribuir el trabajo fácilmente entre procesos locales.

El siguiente paso en la evolución del sistema sería completar la integración con una blockchain añadiendo los resultados y datos más significativos. Como ampliación futura se debería añadir una capa de seguridad a los servicios implementados.

Áreas de mejora son las operaciones de la API, añadiendo más opciones de análisis y mejorando la usabilidad de las interfaces de usuario que proporcionan la API y la dashboard.

Referencias

- [1] «Crowdsensing - Wikipedia». <https://en.wikipedia.org/wiki/Crowdsensing> (accedido 4 de julio de 2022).
- [2] «— SDG Indicators». <https://unstats.un.org/sdgs/report/2020/goal-12/> (accedido 5 de julio de 2022).
- [3] N. G. C. Change, «Global Surface Temperature | NASA Global Climate Change», *Climate Change: Vital Signs of the Planet*. <https://climate.nasa.gov/vital-signs/global-temperature> (accedido 6 de julio de 2022).
- [4] H. Jamil, T. Umer, C. Ceken, y F. Al-Turjman, «Decision Based Model for Real-Time IoT Analysis Using Big Data and Machine Learning», *Wirel. Pers. Commun.*, vol. 121, n.º 4, 2021, doi: 10.1007/s11277-021-08857-7.
- [5] S. K. Singh, J. Cha, T. W. Kim, y J. H. Park, «Machine learning based distributed big data analysis framework for next generation web in iot», *Comput. Sci. Inf. Syst.*, vol. 18, n.º 2, 2021, doi: 10.2298/CSIS200330012S.
- [6] J. Moon, S. Kum, y S. Lee, «A Heterogeneous IoT Data Analysis Framework with Collaboration of Edge-Cloud Computing: Focusing on Indoor PM10 and PM2.5 Status Prediction», *Sensors*, vol. 19, n.º 14, p. 3038, jul. 2019, doi: 10.3390/s19143038.
- [7] J. V. Coleman, «Green Smart Campus», Madrid, jul. 2021. [En línea]. Disponible en: <https://oa.upm.es/68208/>
- [8] R. Krishnamurthi, A. Kumar, D. Gopinathan, A. Nayyar, y B. Qureshi, «An overview of iot sensor data processing, fusion, and analysis techniques», *Sens. Switz.*, vol. 20, n.º 21, 2020, doi: 10.3390/s20216076.
- [9] «• Smartphone users 2026 | Statista». <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/> (accedido 2 de julio de 2022).
- [10] «Inteligencia colectiva», *Wikipedia, la enciclopedia libre*. 11 de marzo de 2022. Accedido: 6 de julio de 2022. [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=Inteligencia_colectiva&oldid=142209904
- [11] «Public Cloud vs Private Cloud vs Hybrid Cloud | Microsoft Azure». <https://azure.microsoft.com/en-us/overview/what-are-private-public-hybrid-clouds/> (accedido 6 de julio de 2022).
- [12] «KAA: Data analytics». <https://docs.kaaiot.io/KAA/docs/current/Features/Data-analytics/> (accedido 6 de julio de 2022).
- [13] «Primary energy consumption from renewables», *Our World in Data*. <https://ourworldindata.org/grapher/primary-energy-renewables> (accedido 8 de julio de 2022).
- [14] «Azure Stream Analytics now available on IoT Edge». <http://microsoftonlineguide.blogspot.com/2017/11/azure-stream-analytics-now-available-on.html> (accedido 6 de julio de 2022).
- [15] «What is AWS IoT Analytics? - AWS IoT Analytics». <https://docs.aws.amazon.com/iotanalytics/latest/userguide/welcome.html> (accedido 6 de julio de 2022).

Anexos

Anexo A – Requisitos del sistema desarrollado

Requisitos de Usuario

RU.21. Como usuario SYST/Gestor de la ETSISI quiero ver una gráfica comparativa diaria entre la temperatura y la sensación de temperatura de los estudiantes a la hora que se les ha pasado la encuesta para poder determinar qué temperaturas y humedades son inadecuadas para establecer un buen ambiente.

RU.22. Como usuario SYST/Gestor de la ETSISI quiero ver una gráfica comparativa mensual entre la temperatura y la sensación de temperatura de los estudiantes a la hora que se les ha pasado la encuesta para poder determinar qué temperaturas y humedades son inadecuadas para establecer un buen ambiente.

RU.23. Como usuario SYST/Gestor de la ETSISI quiero ver una gráfica comparativa diaria entre el nivel de CO₂ del aula y la sensación del ambiente de los estudiantes a la hora que se les ha pasado la encuesta para poder determinar qué niveles de CO₂ son inadecuados para establecer un buen ambiente.

RU.24. Como usuario SYST/Gestor de la ETSISI quiero ver una gráfica comparativa mensual entre el nivel de CO₂ del aula y la sensación del ambiente de los estudiantes a la hora que se les ha pasado la encuesta para poder determinar qué niveles de CO₂ son inadecuados para establecer un buen ambiente.

RU.25. Como usuario SYST/Gestor de la ETSISI quiero ver una gráfica comparativa diaria entre la luminosidad del aula y la sensación de iluminación de los estudiantes a la hora que se les ha pasado la encuesta para poder determinar qué niveles de luminosidad son inadecuados para establecer un buen ambiente.

RU.26. Como usuario SYST/Gestor de la ETSISI quiero ver una gráfica comparativa mensual entre la luminosidad del aula y la sensación de iluminación de los estudiantes a la hora que se les ha pasado la encuesta para poder determinar qué niveles de luminosidad son inadecuados para establecer un buen ambiente.

RU.27. Como usuario SYST/Gestor de la ETSISI quiero ver una gráfica comparativa diaria de dos aulas diferentes mostrando la luminosidad de cada aula y la sensación de iluminación de los estudiantes en cada aula a la hora que se les ha pasado la encuesta para poder determinar como influye el tipo de aula.

RU.28. Como usuario SYST/Gestor de la ETSISI quiero ver una gráfica comparativa diaria de dos aulas diferentes mostrando el nivel de CO₂ de cada aula y la sensación del ambiente de los estudiantes en cada aula a la hora que se les ha pasado la encuesta para poder determinar como influye el tipo de aula.

RU.29. Como usuario SYST/Gestor de la ETSISI quiero ver una gráfica comparativa diaria de dos aulas diferentes mostrando la temperatura y humedad de cada aula y la sensación de temperatura de los estudiantes en cada aula a la hora que se les ha pasado la encuesta para poder determinar como influye el tipo de aula.

Requisitos funcionales

RF.3. El sistema deberá ser capaz de calcular la media de la temperatura medida en el mismo instante de tiempo por todos los distintos NSA instalados en un aula.

RF.4. El sistema deberá ser capaz de calcular la media del nivel de iluminación medido en el mismo instante de tiempo por todos los distintos NSA instalados en un aula.

RF.5. El sistema deberá ser capaz de calcular la media de la humedad relativa medida en el mismo instante de tiempo por todos los distintos NSA instalados en un aula.

RF.6. El sistema deberá ser capaz de calcular la media del nivel de CO₂ medido en el mismo instante de tiempo por todos los distintos NSA instalados en un aula.

Anexo B – Variables de configuración

MODIN_ENGINE=dask

FIREBASE_FEEDBACK_KEYPATH=./keys/tfm2022-72c94-firebase-adminsdk-ucaeg-60cfbff520.json

MONGODB_SENSOR_USERNAME=sensorUser

MONGODB_SENSOR_PASSWORD=password

MONGODB_CACHE_USERNAME=sensorUser

MONGODB_CACHE_PASSWORD=password

SCHEDULER_URL=127.0.0.1:8786

MONGO_HOST=127.0.0.1

MONGO_DATABASE=sensor

MONGO_PORT=27017

MONGO_COLLECTION=readings

MONGO_AUTH_MECHANISM=&authSource=admin&authMechanism=SCRAM-SHA-1

FIREBASE_COLLECTION=feedback

FEEDBACK_ANALYSIS_CATEGORY=Ambiente

FIREBASE_FEEDBACK_KEYPATH/keys/tfm2022-72c94-firebase-adminsdk-ucaeg-60cfbff520.json

USE_FILE_INSTEAD_OF_FIRESTORE=./mod_feedback.csv

Anexo C – Fichero de configuración

datasources:

sensors:

host: "127.0.0.1"

database: "sensor"

port: 27017

collection: "readings"

auth_mechanism: "&authSource=admin&authMechanism=SCRAM-SHA-1"

feedbacks:

collection: "feedback"

category: "Ambiente"

cluster:

engine: "dask"

scheduler_type: "distributed" #"processes" #"distributed"

scheduler_url: "127.0.0.1:8786"

partitions: 2

workers: 4

cache:

enable: False

host: "127.0.0.1"

port: 27017

database: "cache_db"

collection: "cache_collection"

data: # mappings

feedback:

category: "Ambiente" # vote category containing votes related to the environment

sense:

maps a measure with a reason from the votes

temperature:

pos: ['Temperatura Perfecta']

neg: ['Demasiado calor', 'Demasiado frío']

humidity:

pos: ['Temperatura Perfecta']

neg: ['Demasiado calor', 'Demasiado frío']

luminosity:

pos: ['Iluminación correcta']

neg: ['Poca luz']

co2:

pos: ['Ambiente agradable']

neg: ['Ambiente denso']

noise:

pos: ['Sin ruido']

neg: ['Demasiado ruido']

movement:

pos: []

neg: []

sensors:

maps a measure with sensor types

temperature: ['room_temp', 'add_temp', 'surf_temp']

humidity: ['humidity']

luminosity: ['luminosity']

movement: ['movement']

co2: ['co2']

noise: ['noise']

api:

host: '127.0.0.1'

port: 9080