ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

MÁSTER EN BIG DATA: TECNOLOGÍA Y ANALÍTICA AVANZADA

Proceso de generación, ingesta, almacenamiento,visualización y análisis de curvas de consumo para segmentación de clientes

Autor: Miren Ostolaza Larrañaga

Director: Borja Ayerdi Vilches

Madrid

Julio 2021

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Proceso de generación, ingesta, almacenamiento, visualización y análisis de curvas de consumo para segmentación de clientes

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2020/21 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

Fdo.: Miren Ostolaza Larrañaga Fecha: …04…/ …07…/ …2021…

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Borja Ayerdi Vilches Fecha: ..11../ ..07../ 2021

Vº Bº del Coordinador de Proyectos

Fdo.: Carlos Morrás Ruiz-Falcó Fecha: ……/ ……/ ……

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D.Miren Ostolaza Larrañaga DECLARA ser el titular de los derechos de propiedad intelectual de la obra: PROCESO DE GENERACIÓN, INGESTA, ALMACENAMIENTO,VISUALIZACIÓN Y ANÁLISIS DE CURVAS DE CONSUMO PARA SEGMENTACIÓN DE CLIENTES, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar "marcas de agua" o cualquier otro sistema de seguridad o de protección.
b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
e) Asignar por defecto a estos trabajos una licencia Creative Commons.
f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente)*.

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

a) Que la Universidad identifique claramente su nombre como autor de la misma
b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
c) Solicitar la retirada de la obra del repositorio por causa justificada.
d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.

d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusive del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a ……04….. de ……07…….... de ……2021….

ACEPTA

Fdo…Miren Ostolaza Larrañaga……

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

MÁSTER EN BIG DATA: TECNOLOGÍA Y ANALÍTICA AVANZADA

Proceso de generación, ingesta, almacenamiento, visualización y análisis de curvas de consumo para segmentación de clientes

Autor: Miren Ostolaza Larrañaga

Director: Borja Ayerdi Vilches

Madrid

Julio 2021

Agradecimientos

Quisiera agradecer a todas las personas que han formado parte de este año en el máster. A todos los profesores que han contribuido a mi formación, por su vocación y dedicación; y a mi tutor Borja, por sus consejos a lo largo de este tiempo.

No quiero olvidarme de los compañeros que he conocido y con los que he compartido tantas horas. Gracias por haberme acompañado y por la ayuda recibida cuando la he necesitado.

Por último, querría agradecer a mi familia por haberlo hecho posible. Por los ánimos que he recibido de principio a fin, y por la confianza depositada en mí.

PROCESO DE GENERACIÓN, INGESTA, ALMACENAMIENTO, VISUALIZACIÓN Y ANÁLISIS DE CURVAS DE CONSUMO PARA SEGMENTACIÓN DE CLIENTES

Autor: Ostolaza Larrañaga, Miren.
Director: Ayerdi Vilches, Borja.
Entidad Colaboradora: Iberdrola, S. A.

RESUMEN DEL PROYECTO

Este proyecto se ha realizado con la colaboración de Iberdrola y consistirá en desarrollar un pipeline completo de generación, ingesta, almacenamiento, visualización y análisis de curvas de consumo para segmentación de clientes en función de los patrones y hábitos de consumo energético.

Palabras clave: Consumo energéticos, Segmentación, Real Time

1. Introducción

Iberdrola recibe información de consumos energéticos procedente de sus clientes, por lo que busca extraer valor de esa información para poder ofrecerles un mejor servicio en el futuro, en base a sus hábitos de consumo.

Para ello, es necesario contar con una aplicación capaz de recibir esos datos, almacenarlos y transformarlos en información útil que puedan tenerlo en cuenta para decisiones futuras.

2. Definición del Proyecto

Para conseguir una segmentación de clientes en función de sus hábitos de consumo y patrones que se puedan identificar en el conjunto de datos, se ha desarrollado un proyecto que incluye cinco módulos que conjuntamente serán capaces de conseguir el objetivo fijado. El pipeline constará de los siguientes módulos: generación de datos, ingesta, almacenamiento, visualización y análisis de los datos mediante algoritmos de aprendizaje automático.

3. Descripción del modelo/sistema/herramienta

Tal y como se ha mencionado se han requerido cinco bloques que manejen el flujo de datos hasta llegar a los resultados deseados. A continuación, se describirá cada uno de los módulos:

1. Generación de datos sintéticos que simulen consumos reales de clientes de Iberdrola y representen diferentes perfiles de usuarios para posteriormente ser clasificados acorde a las características impuestas en este módulo.
2. Ingesta de los datos generados en tiempo real mediante colas Kafka para ser depositados en un sistema de almacenamiento.
3. Almacenamiento de los datos ingestados en un sistema de almacenamiento NoSQL, MongoDB, para realizar consultas y acceder fácilmente al conjunto de datos.

4. Visualización de la información almacenada para un mayor conocimiento de lo que transmiten los datos mediante la librería Dash en Python, así como representación en tiempo real de los registros que se estén ingestando en el momento desde el generador.
5. Análisis del conjunto de datos almacenado en un periodo de una semana para realizar la segmentación final de clientes mediante el algoritmo de clustering k means.

En la Ilustración 1 se muestra la ruta seguida compuesta por estos cinco módulos desde que se genera el dato hasta que se visualiza y se realiza el agrupamiento utilizando las diferentes herramientas mostradas.

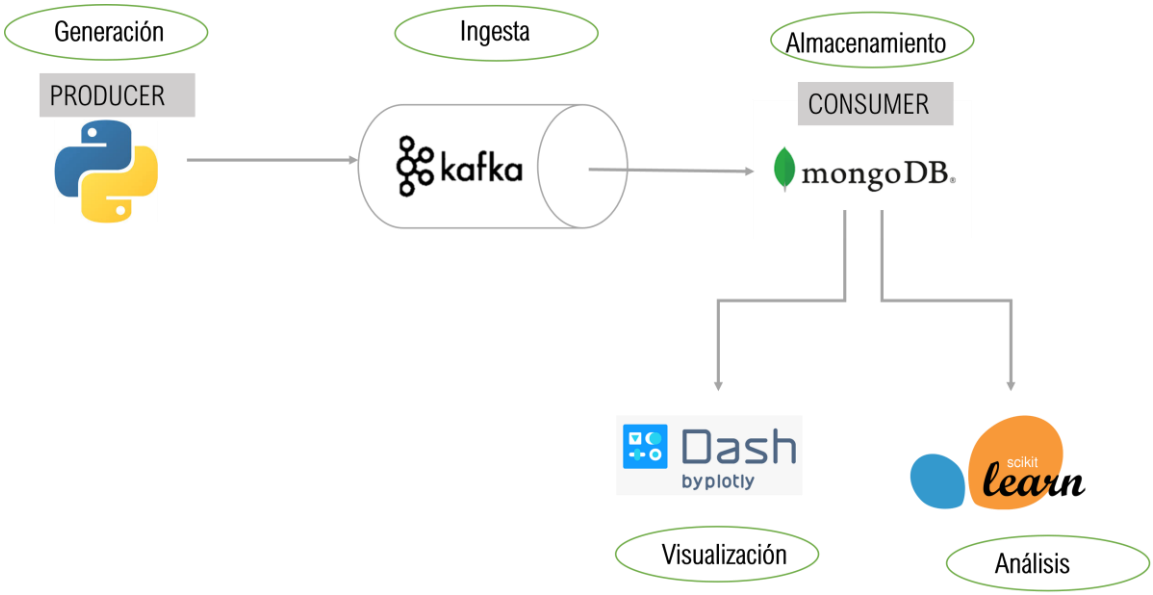

Ilustración 1 – Arquitectura de los cinco módulos mencionados con las herramientas tecnológicas utilizadas en cada uno de ellos.

4. Resultados

El resultado, por tanto, es la agrupación de los registros almacenados en cuatro *clusters* en función de la hora del día y del día de la semana en el que ocurre el consumo. Como se muestra a continuación en la Ilustración 2, el modelo ha distinguido cada usuario con su perfil correspondiente y acorde a lo definido en el primer módulo del proyecto.

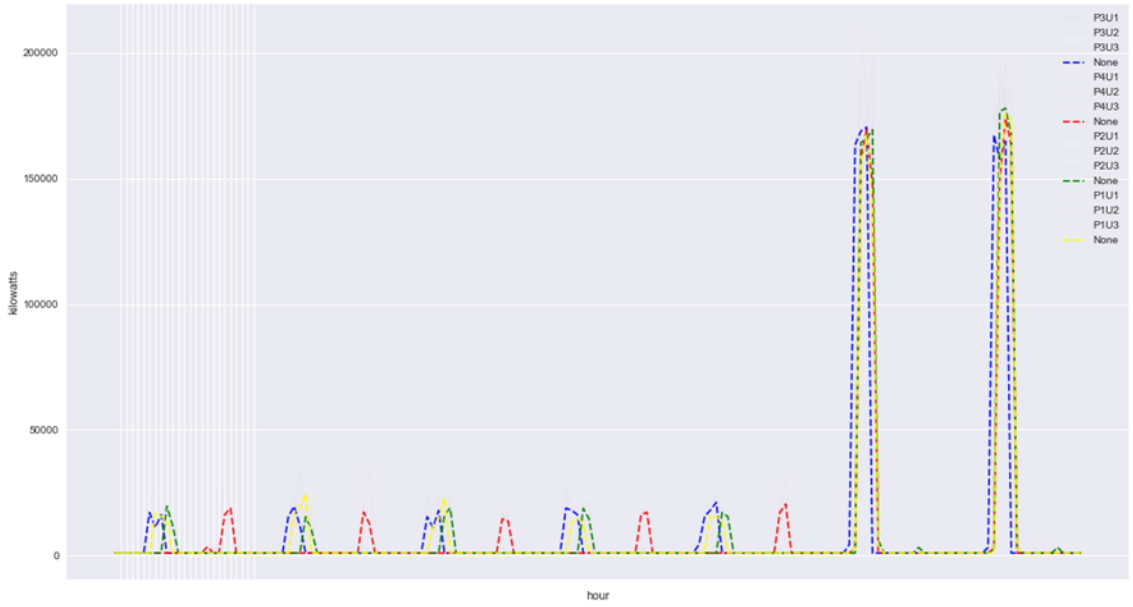

Ilustración 2 – Segmentación de los clientes en cuatro clusters generados por el modelo.

5. Conclusiones

Se ha visto que se ha cumplido con los objetivos marcados al inicio del proyecto, y en un futuro se debería intentar aumentar el número de registros con el que trabaja la aplicación para contextualizarlo en el mundo real.

PROCESO DE GENERACIÓN, INGESTA, ALMACENAMIENTO, VISUALIZACIÓN Y ANÁLISIS DE CURVAS DE CONSUMO PARA SEGMENTACIÓN DE CLIENTES

Autor: Ostolaza Larrañaga, Miren.
Director: Ayerdi Vilches, Borja.
Entidad Colaboradora: Iberdrola, S. A.

ABSTRACT

This project has been carried out in collaboration with Iberdrola and will consist of developing a complete pipeline of generation, ingestion, storage, visualization and analysis of consumption curves for customer segmentation based on energy consumption patterns and habits.

Keywords: Energetic consumptions, customer segmentation, Real Time

1. Introduction

Iberdrola receives information on energy consumption from its customers, so it seeks to extract value from this information to be able to offer them a better service in the future based on their consumption habits.

To do this, it is necessary to have an application capable of receiving this data, storing it, and transforming it into useful information that can be taken into account for future decision-making.

2. Project definition

In order to achieve customer segmentation based on their consumption habits and patterns that can be identified in the data set, this project has been developed including five modules that together will be able to achieve the set objective. The pipeline will consist of the following modules: data generation, ingestion, storage, visualization and analysis of the data using machine learning algorithms.

3. Description of the model/system/tool

As mentioned above, five blocks have been required to handle the data flow until the desired results are reached. Each of the modules will be explained below:

1. Generation of synthetic data that simulates real consumption of Iberdrola customers and representing different user profiles to be classified according to the characteristics imposed in this module.

2. Ingestion of the data generated in real time using Kafka queues to be deposited in a storage system.

3. Storage of the ingested data in a NoSQL storage system, MongoDB, to access easily to the dataset.

4. Visualisation of the stored information for a better understanding of what the data is transmitting through the Dash library in Python, as well as real-time representation of the records that are being ingested in real time from the generator.

5. Analysis of the data set stored over a period of one week to perform the final segmentation of customers using the k means clustering algorithm.

Illustration 1 shows the route followed by these five modules from the time the data is generated until it is visualised, and the clustering is performed using the different tools shown.

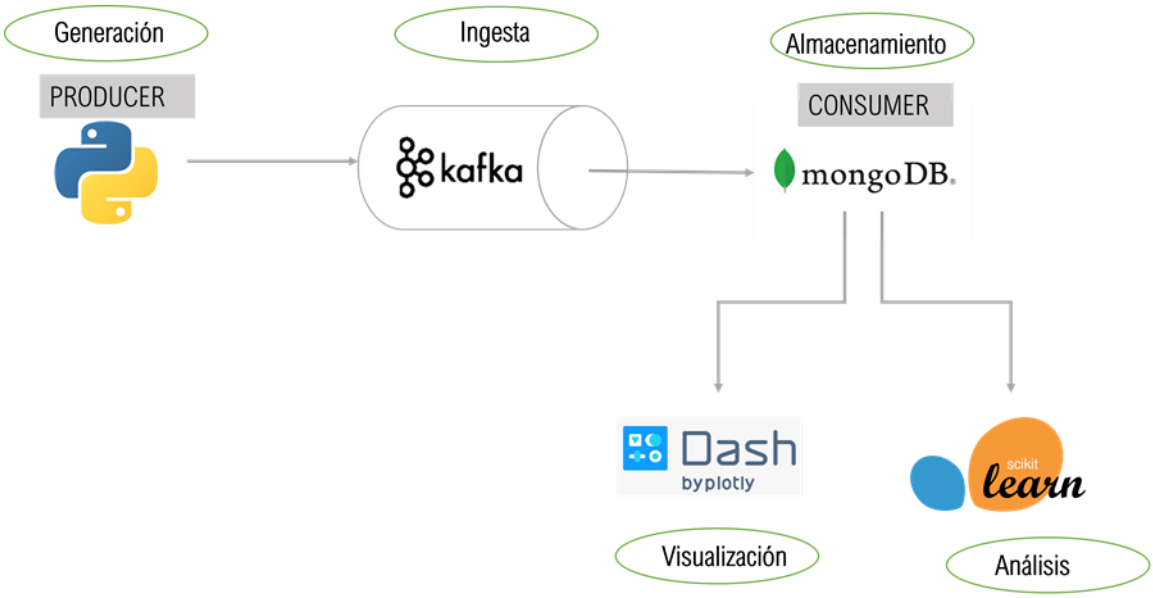

Ilustración 3 - Architecture of the five modules mentioned above with the technological tools used in each of them.

4. Results

The result, therefore, is the grouping of the records stored in four clusters according to the time of day and the day of the week in which consumption occurs. As shown below, the model has distinguished each user with their corresponding profile and according to what was defined in the first module of the project.

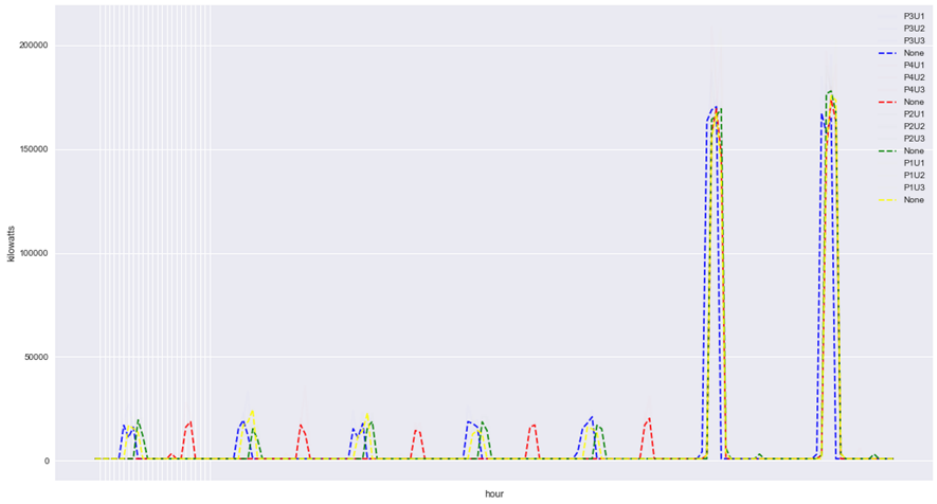

Ilustración 4 – Customer segmentation into four clusters generated by the model.

5. Conclusions

It has been seen that the objectives set at the beginning of the project have been met, and in the future an attempt should be made to increase the number of records with which the application works in order to be a more realistic scenario.

Índice de la memoria

Índice de figuras

Índice de tablas

Capítulo 1. INTRODUCCIÓN

Debido al avance tecnológico y proliferación de sistemas de la información, como sociedad somos capaces de generar y almacenar más datos que nunca. Por esa capacidad de crear datos de forma masiva, se han convertido en un recurso muy valioso en todos los ámbitos.
Se dice que estamos entrando en la era de los grandes datos, y es que la velocidad a la que somos capaces de producirlos es digna de mención. Por comentar algunos ejemplos, en una aplicación tan conocida como Youtube, por segundo se sube una hora de vídeo, lo que equivaldría a diez años de contenido al día. En la famosa multinacional estadounidense Walmart, se dan alrededor de un millón de transacciones por hora suponiendo bases de datos de 2,5 petabytes [1].

Por otra parte, debido al bajo coste que supone su almacenamiento, es habitual que se almacenen los datos hasta formar bases de datos de gran volumen. Por ello, se busca sacar el máximo provecho a este recurso accesible para muchos.

Podría denominarse como datificación a este cambio en nuestra sociedad a causa de la evolución tecnológica. Un cambio que se caracteriza por la capacidad de producir datos que puedan ser leídos por un ordenador [2]. En esta nueva realidad en la que nos encontramos, conceptos como el Big data, la Inteligencia Artificial o el Machine Learning forman parte del contexto social actual. Parte importante de toda esta revolución tecnológica son las compañías que cada vez en mayor medida incorporan las tecnologías y técnicas para hacer frente a esos nuevos retos. Entre estas organizaciones implicadas en la innovación se encuentra Iberdrola, empresa colaboradora para el desarrollo del proyecto que se presenta en esta memoria.

Iberdrola S.A. es un grupo empresarial con sede principal en Bilbao, País Vasco, que se dedica a la producción, distribución y comercialización de la energía. Fundada en 1992 como resultado de la fusión de las compañías llamadas Hidroeléctrica Española e Iberduero, es

considerada una de las empresas líderes entre los grupos energéticos a nivel nacional. Cabe destacar además su presencia internacional. Situado a la cabeza del sector eólico mundial, en la actualidad está presente en 30 países dando servicio a más de 27 millones de clientes.

Convirtiéndose en el quinto líder de la historia del Ibex 35 en el año 2020 por capitalización bursátil, en los últimos años Iberdrola ha dedicado gran parte de su inversión a la innovación. Una de las vertientes principales de dicha innovación es la transformación digital. Conscientes de la importancia creciente del uso de tecnologías digitales, Iberdrola tomó la decisión de aumentar su inversión en I+D+I a 400 millones de euros anuales a partir de 2025 [3], con el fin de reforzar la eficiencia y la productividad en todas sus áreas de negocio.

Consecuencia de la agravante preocupación por el medio ambiente y los efectos adversos del cambio climático, particulares y empresas se ven obligados a cambiar de hábitos con el fin de reducir su impacto en nuestro planeta. El sector energético juega un papel fundamental en dicha transición, con suma relevancia la presencia de energías renovables.
Dentro de esta transición energética la innovación es la herramienta fundamental para conocer con anterioridad las necesidades que este problema pueda suponer en la sociedad. Es por ello por lo que, a día de hoy, la mayoría de las empresas pertenecientes a este sector incluyen el Big data y la Inteligencia Artificial como parte de su estrategia y dedican cada vez más recursos a los adelantos en el área tecnológica que facilitan la toma de decisiones y mejoran la eficiencia de las compañías, razón con la cual empresas del calibre de Iberdrola priorizan cada vez más estas áreas.

En la mencionada transformación digital el Big data y la inteligencia artificial juegan un papel fundamental. Dentro de los objetivos estratégicos definidos por Iberdrola se ha querido hacer hincapié en los siguientes presentados a continuación por estar directamente relacionados con el propósito de este proyecto [4]:

- Análisis avanzado de la información recibida de los contadores inteligentes para la toma de decisiones y la previsión de la demanda energética.

- Algoritmos para ofrecer la tarifa más adecuada a cada cliente según sus hábitos y patrones de consumo.
- Planes personalizados y segmentación de los clientes.

Objetivos y metas accesibles mediante técnicas como la aplicación de algoritmos de aprendizaje automático o inteligencia artificial en entornos de Big data. Herramientas que pueden suponer un impacto positivo en la toma de decisiones y en la calidad de sus productos y servicios.
El Big data engloba el manejo y procesamiento de datos masivos que se generan y que permiten, entre otras mejoras, un mayor conocimiento del consumo de sus clientes. Esto implica una mejora en la relación con el cliente, posibilitando una personalización de sus servicios en función de sus consumos y necesidades.

Este proyecto se centrará en reproducir un caso de uso real de esta compañía energética donde se reciben datos masivos de consumos eléctricos, por lo que les es fundamental un manejo y procesamiento eficientes de esas recepciones de datos.

En un escenario real, a una empresa energética como Iberdrola estos datos provenientes de sus clientes son recibidos mediante diferentes sistemas, entre ellos los contadores inteligentes. Gracias a contadores inteligentes, las compañías energéticas y en este caso Iberdrola son capaces de obtener información periódicamente de los consumos de sus clientes. Un contador inteligente es una herramienta de medida que sirve para identificar el consumo que se esté produciendo en el momento. De esta forma, los usuarios pueden tener acceso continuo a su propio gasto energético, sin tener que enviar las lecturas para obtener esa información [5].

El reciente salto en la tecnología de los contadores inteligentes ha abierto nuevas perspectivas en la monitorización del comportamiento de los usuarios en sus residencias y puede utilizarse para varias aplicaciones diferentes [6].

Una de ellas es realizar una segmentación de clientes en función de características que puedan tener en común entre ellas. Las nuevas especificaciones que surgen en el mercado energético hacen necesario un acercamiento a una medición y gestión eficaz del consumo energético del usuario y sus tendencias, no sólo en lo que respecta al cliente de gran consumo sino también al usuario residencial de medio y alto consumo energético cuyo perfil de consumo muestra patrones desequilibrados de picos de consumo de energía, por lo que será necesario diferenciar los distintos tipos de clientes que pueda tener la compañía para mejorar sus servicios [7].

Esta segmentación permite a la organización conocer mejor las necesidades energéticas de los edificios, y más información sobre el uso real del hogar ayudaría a determinar un diseño más adecuado de sus servicios. También se podría reducir el consumo de energía si los programas de política de ahorro de energía se dirigieran a diferentes grupos de usuarios, de esta forma una mayor personalización de las tarifas de consumo repercutiría en un ahorro energético orientado a diferentes grupos de consumidores [8].

1.1 DESCRIPCIÓN DE LAS TECNOLOGÍAS

El proyecto está formado por varios módulos por lo que también se hará esa distinción en este apartado donde se realizará un estudio de las tecnologías para cada uno de los módulos.

1.1.1 GENERACIÓN DE DATOS

La generación de datos es una tarea clave para garantizar la seguridad y la ausencia del dato en casos en los que se quiera preservar el anonimato o cuando no se disponga de un conjunto de datos lo suficientemente grande. Una posible solución para hacer frente a esos inconvenientes es la generación de datos sintéticos.

Los datos auténticos que se reciben de clientes pasan por una fase de creación de datos sintéticos, de manera que mantengan las mismas características que los datos de origen.

Estos datos sintéticos se podrán crear con más o menos dispersión para controlar el parecido a los reales en función del grado de privacidad que se quiera conseguir.

Una de las técnicas más utilizadas para generar datos sintéticos es la utilización del algoritmo de Deep Learning conocido como GAN (Generative Adversarial Networks). Estas redes neuronales consisten en dos procesos inversos: un generador y un discriminador. El generador aprende a crear datos que parecen reales mientras que el discriminante aprende a diferenciar los reales de las falsificaciones creadas por el generador.
En la Figura 1 se puede apreciar la arquitectura compuesta por esas dos fases de los algoritmos GAN.

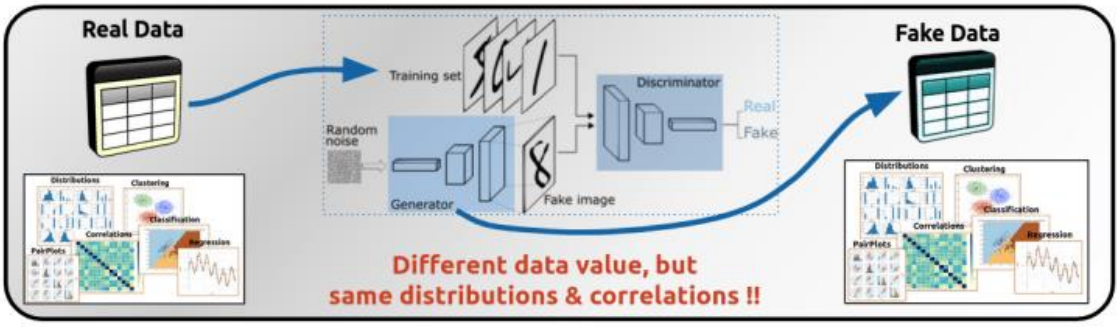

Figura 1. Diagrama de estructura de un algoritmo GAN [9]

Sin embargo, cabe recalcar que para llevar a cabo este proceso es necesario disponer de datos originales iniciales para anonimizarlos con el proceso, sin embargo, en este caso concreto no se han podido facilitar datos reales que puedan servir como base para la generación con GAN por lo que esta tecnología ha sido descartada en el proyecto.

1.1.2 INGESTA DE DATOS

La ingesta de datos es el proceso de trasladar los datos desde una fuente a un destino donde pueden ser almacenados y analizados posteriormente de la forma más eficiente y correcta

posible. Este se ha convertido en un problema importante ya que cada vez son más los datos generados en un periodo corto de tiempo, al igual que la deduplicación, el mantenimiento de las restricciones de integridad y la carga masiva de datos [10].

Con el desarrollo de tecnologías emergentes como el IoT, es inevitable que el volumen de los datos crezca exponencialmente hasta un nivel sin precedentes. Según un informe de IBM Marketing Cloud, "10 Key Marketing Trends For 2017" [11], el 90% de los datos del mundo actual se han creado solo en los dos últimos años, con 2,5 quintillones de bytes de datos al día. Este desmedido volumen, y la variedad dentro de los datos dificulta el almacenamiento y el procesamiento de datos. En estos entornos de Big data el traslado de los datos y su manejo puede repercutir en los procedimientos posteriores que se apliquen a los datos, así como la calidad de los datos y la rapidez con la que viajen desde la fuente al destino. Es por ello por lo que las empresas, y entre ellos Iberdrola, necesitan herramientas y tecnologías específicas para llevar a cabo de manera eficiente la ingesta de datos que pueda recoger y analizar diferentes tipos de datos. Con la generación de datos masivos surge un nuevo concepto, el procesamiento en streaming o en tiempo real. Las aplicaciones y sistemas recientes como el IoT (Internet of Things) buscan proporcionar un modelo exacto del mundo real con el fin de dar cabida a la toma de decisiones en tiempo real [12].

Dependiendo del caso de uso y de las características del contexto, existen diferentes tecnologías que cumplen con la función de ingestar datos en la actualidad. Lo que se busca en este tipo de herramientas es que sean capaces de entregar con rapidez los datos recibidos, que sean fáciles de implantar y con un coste económico asumible.

El tipo más común de ingesta de datos es el procesamiento en *batch*. Se agrupan periódicamente los datos de origen en lotes que luego se enviarán al sistema de destino. Los lotes pueden procesarse en función de cualquier orden lógico. Cuando el caso de uso no requiere la disposición de datos casi en tiempo real, se suele utilizar el procesamiento por lotes, ya que suele ser más fácil y asequible que la ingestión en *streaming*. Las herramientas más conocidas de *Open Source* para procesamiento en *batch* en entonos Big data son Spark y MapReduce, pertenecientes al *framework* Hadoop.

En cuanto al procesamiento en tiempo real, este no implica ningún tipo de agrupación por lotes. Los datos se obtienen, se manipulan y se cargan tan pronto como llegan desde su creación. Este tipo de ingesta es más caro, ya que requiere que los sistemas supervisen constantemente las fuentes y acepten nueva información. Sin embargo, puede ser apropiado para los análisis que trabajan con datos continuamente actualizados. Cabe destacar que algunas plataformas de procesamiento en *streaming* (como Apache Spark Streaming) utilizan en realidad un procesamiento especial por lotes, más conocido como *micro batching*. Es decir, no se procesan de manera individual, sino que se forman lotes, pero de menor tamaño y con mayor frecuencia.

Una de las herramientas más conocidas para procesamiento en tiempo real, y que se utilizará en este proyecto, es Apache Kafka. A continuación, se procederá a exponer esta herramienta en mayor profundidad. Otras herramientas que podrían cumplir con los requerimientos establecidos en el proyecto podrían ser Flink o Kinesis.

Tal y como se ha mencionado previamente, para esta parte del pipeline se ha decidido hacer uso de sistema distribuido Apache Kafka para que se encargue de trasladar los datos generados en la fase anterior a una base de datos donde se almacenarán posteriormente. Apache Kafka es una plataforma de *streaming* de eventos distribuidos de código abierto originalmente desarrollado por la empresa Linkedin a principios de 2011 y su popularidad le ha llevado a ser utilizada por miles de empresas reconocidas como Walmart, Netflix o Microsoft Azure.

Algunas funciones añadidas, como la compactación de registros y las consultas interactivas han hecho de Kafka una solución óptima para procesamientos en *streaming*. Kafka es un sistema de mensajería de publicación-suscripción rápido, escalable y tolerante a fallos.

Esta tecnología se basa en la transmisión de eventos, la práctica de capturar datos en tiempo real desde fuentes de eventos como bases de datos, sensores, dispositivos móviles, servicios en la nube y aplicaciones de software en forma de flujos de eventos. Almacenar los flujos de eventos de forma duradera para su posterior recuperación, y la posibilidad de manipular y procesar los flujos de eventos en tiempo real. Kafka también permite redireccionar los flujos

de eventos a diferentes tecnologías de destino tras su procesamiento. El streaming de eventos garantiza así un flujo y una interpretación continuos de los datos, de modo que la información correcta esté en el lugar y el momento adecuados [13].

Kafka combina tres capacidades clave que le permiten adaptarse a diferentes casos de uso para el *streaming* de eventos de principio a fin [14]:

- Publicar (escribir) y suscribir (leer) flujos de eventos, incluyendo la importación/exportación continua de sus datos desde otros sistemas.
- Para almacenar flujos de eventos de forma duradera y fiable durante todo el tiempo que desee.
- Procesar flujos de eventos a medida que se producen o de forma retrospectiva.

Estas características hacen de Kafka la principal herramienta utilizada por empresas diversas de todo el mundo cuando se trabaja en real time.

En la arquitectura de Kafka se definen varios componentes que será importante comprender antes de implementarlo en el proyecto.

- **Evento**: se denomina registro o mensaje. Cuando se leen o escriben datos en Kafka, se hace en forma de eventos. Conceptualmente, un evento tiene una clave, un valor y una marca de tiempo.

- **Producer**. Los producers son aquellas aplicaciones cliente que publican (escriben) eventos en Kafka, y los **consumers** son aquellos que se suscriben a (leen y procesan) estos eventos. En Kafka, los productores y los consumidores están totalmente desacoplados y es un elemento clave para lograr la alta escalabilidad por la que es conocido Kafka. Los productores nunca tienen que esperar a los consumidores [14].

- **Topic**: Un *topic* podía definirse como una carpeta dentro de un sistema de archivos, y los eventos son los archivos de esa carpeta. Los *topics* son siempre multi-

productores y multi-suscriptores, es decir, un tema puede tener cero, uno o muchos productores que escriben eventos en él, así como cero, uno o muchos consumidores que se suscriben a estos eventos. Los eventos de un *topic* pueden leerse tantas veces como sea necesario, y a diferencia de los sistemas de mensajería tradicionales, los eventos no se eliminan tras su consumo [15].

- **Particiones:** Los *topics* están particionados, lo que significa que un *topic* estará repartido en un número de partes ubicados en diferentes *brokers* de Kafka. Esta distribución de los datos es muy importante para la escalabilidad, ya que permite leer y escribir los datos en muchos *brokers* al mismo tiempo. Cada partición es una secuencia ordenada e inmutable de datos. A cada dato almacenado en una partición se le asigna un identificador secuencial llamado **offset** que identifica a cada dato anexado en la partición [16].

- **Zookeaper**: Para ofrecer a los *brokers* metadatos sobre los procesos que se ejecutan en el sistema y para facilitar la comprobación el estado y la elección del *broker* que será líder [17].

A continuación, en la Figura 2 se presenta la arquitectura de Apache Kafka que se puede encontrar en la documentación oficial y que engloba todas las componentes mencionadas previamente para facilitar la comprensión de los mismos.

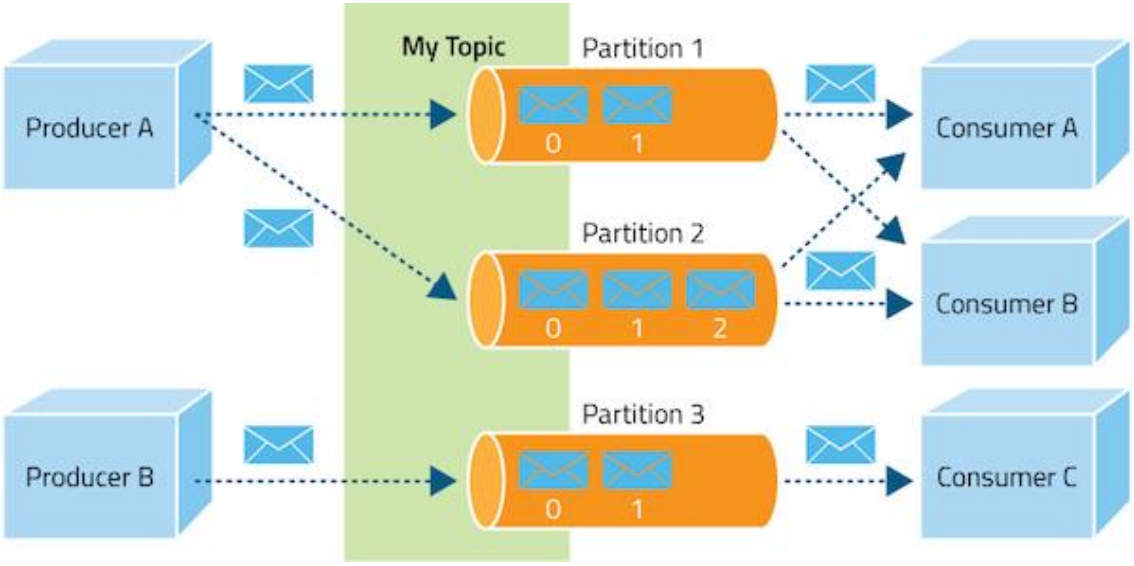

Figura 2. Arquitectura de Apache Kafka [18]

1.1.3 ALMACENAMIENTO

Otro punto fundamental en este proyecto y en todos los proyectos que trabajan con registros que luego deben ser analizados o procesador es conocer dónde y cómo se almacenarán los datos para poder acceder a ellos posteriormente de la manera más eficiente posible.

El almacenamiento dentro del mundo *big data* es el ejercicio que se ocupa de almacenar y gestionar los datos de forma escalable, satisfaciendo las necesidades de las aplicaciones que requieren acceso a los datos. El sistema ideal de almacenamiento de *big data* permitiría almacenar una cantidad de datos que pueda hacer frente a altas tasas de acceso de lectura y escritura, que sea flexible y eficiente en la gestión de diferentes tipos de datos, que admita tanto datos estructurados y no estructurados y, por razones de privacidad, sólo funcionara con datos encriptados. Todas estas características son imposibles de cubrir al mismo tiempo por el momento. Sin embargo, cada vez existen más sistemas de almacenamiento capaces de cumplir al menos en parte muchos de esos requerimientos ideales.

Tanto las empresas como los usuarios individuales requieren cada vez un mayor espacio de almacenamiento de datos para satisfacer las necesidades computacionales de proyectos cada vez más ambiciosos que muchas veces se relacionan con la inteligencia artificial (IA), el

aprendizaje automático o el internet de las cosas (IoT). Además del espacio de almacenamiento excesivo, los sistemas de almacenamiento deben ser capaces de evitar la pérdida de datos frente a fallos y que se pueda seguir accediendo a ellos [19].

Existen diferentes tipos de bases de datos. La mayor diferenciación que se hace entre estos suele hacer es la de bases de datos NoSQL y SQL. SQL por sus siglas en Structured Query Language, es un lenguaje de programación diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales [20]. Por ello, el siguiente concepto a aclarar debe ser el de una base de datos relacional. Se trata de un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Basado en el modelo relacional, es una representación intuitiva de los datos en forma de tabla done cada registro se identifica con una clave única. Las bases de datos SQL más utilizadas son MySQL u Oracle.

Por otra parte, las bases de datos NoSQL (Not Only SQL), cuya aparición en los últimos años ha aumentado debido a la necesidad de recopilar grandes cantidades de datos no estructurados por las organizaciones. NoSQL define un conjunto de tecnologías que suponen un cambio a lo que aportan las bases de datos relacionales clásicas. NoSQL es usado como un término general por todas las bases de datos y almacenes de datos que no siguen los populares y bien establecidos principios RDBMS (Relational Database Management System), y a menudo está relacionado con grandes conjuntos de datos y su manipulación en una escala Web. Este término hace referencia al conjunto de tecnologías en bases de datos, que buscan alternativas al sistema de bases de datos relacional, donde priman la velocidad, el manejo de grandes volúmenes de datos y la posibilidad de tener un sistema distribuido [21].

Las principales ventajas que han incorporado estas bases de datos frente a las bases de datos SQL son las siguientes [22]:

- Requieren menos recursos de computación. Esto a su vez repercutirá en el coste económica de la implantación del sistema.
- Escalabilidad horizontal: la posibilidad de añadir más nodos si es necesario.

- Pueden manejar gran cantidad de datos: Esto es debido a que utiliza una estructura distribuida, en muchos casos mediante tablas Hash.
- No genera cuellos de botella: una ejecución más simple que supone un tiempo de ejecución menor.

Dentro de las bases de datos NoSQL se puede hacer una clasificación en base a la estructura que siguen los datos:

• Basadas en **documentos**, almacenan la información como si fueran documentos. Los formatos más utilizados son JSON o XML. Dentro de este grupo se encuentran MongoDB, CouchDB o Cloudant

• Orientadas a **grafos**, este tipo de bases de datos almacena la información en forma de grafo. Se representan mediante nodos y aristas donde cada nodo representa un objeto y una arista la conexión o relación entre dos objetos. Las herramientas asociadas a los grafos son Neo4J y InfoGrid entre otros.

• Orientadas a **columnas**, almacenan toda la información en columnas de esta forma las lecturas son muy rápidas, pero se sacrifica mucho tiempo para las escrituras. La información de cada columna se almacena por separado en disco de manera contigua, Es muy rápida para aplicaciones analíticas que quieren realizar cálculos agregados por columnas. Otra de las ventajas de este tipo de bases de datos es que la compresión es muy eficiente ya que toda la columna es del mismo tipo. Las bases de datos más conocidas pertenecientes a este subgrupo son Hbase, Kudu y Parquet.

• **Clave-valor,** es la forma más usada y simple donde se almacena una clave asociada a un único valor o colección. Suelen ser rápidas para las operaciones de lectura/escritura y proporcionan comandos simples (put, get, delete) para gestionar la información en disco o memoria.

En este proyecto se ha optado por almacenar los datos en una base de datos orientada a documentos, MongoDB. Desarrollada en 2007 por 10gen, es una de las herramientas más utilizadas actualmente que almacena los datos en formato BSON (binary JSON). En MongoDB los documentos se agrupan en colecciones en base a su estructura inicial y con un tamaño máximo de 16 MB.

Capaz de almacenar información no estructurada, es caracterizado por la flexibilidad que ofrece sin tener que definir un schema inicialmente (schemaless). MongoDB identifica cada documento con un _id asignado, sobre el cual se crea un índice. Puede soportar un máximo de 64 índices, pero solo uno de ellos podrá usarse para el sharding (particionamiento horizontal). Las propiedades más importantes son la durabilidad de los datos que se garantiza con la replicación, y la alta escalabilidad que proporciona [23].

Su uso es especialmente recomendable en aplicaciones Web como alternativa a las bases de datos relacionales y cuando se requiere análisis entiempo real debido a las bajas latencias en su acceso. Es por ello, que se ha visto apropiado su uso para el propósito del proyecto.

1.1.4 VISUALIZACIÓN

Con tanta información recopilada a través del análisis de datos en la actualidad, la forma de representar y transmitir toda esa información es fundamental para garantizar su interpretabilidad. Esto implica presentarlos de manera que la mente humana los comprenda por medio de la identificación de tendencias, patrones y valores atípicos en gráficos. La visualización de datos es la presentación de datos en un formato gráfico y que facilita la toma de decisiones con un análisis presentado de forma visual [24].

El objetivo principal de la visualización de datos es facilitar la identificación de patrones, tendencias y valores atípicos en grandes conjuntos de datos. La visualización de datos es uno de los pasos del proceso en el procesamiento de datos, que después de que los datos hayan sido almacenados y transformados, se visualizan para poder sacar conclusiones y extraer información de otra forma sería mucho más complejo adquirir [25].

Aunque se suela pensar que los gráficos estadísticos y la visualización de datos son relativamente modernos, lo cierto es que la representación gráfica de la información cuantitativa viene de mucho antes con su comienzo fijado con la elaboración de mapas. La tecnología ha jugado un papel fundamental en su historia hasta llegar al punto en el que nos encontramos en la actualidad [26].

Un momento clave en la historia de esta disciplina fue la inserción de los ordenadores en las compañías a partir de la década de 1950. La necesidad de garantizar la facilidad de uso de los ordenadores por parte de personas no técnicas forzó a poner el foco en las interfaces gráficas y el desarrollo de representaciones graficas utilizando el ordenador.

Posteriormente, con la llegada de Internet, la necesidad de unas interfaces y entornos amigables para los usuarios de estas máquinas, lo que obligó a desarrolladores a una contar con una comprensión de cómo las personas perciben la información en formato gráfico [27].

Por lo mencionado anteriormente parece lógico pensar que se trata de un ejercicio clave en cualquier proceso de análisis de datos, y es que la visualización de datos proporciona una forma rápida y eficaz de comunicar información de manera universal utilizando información visual. Esta práctica tiene varias aplicaciones que ayudan a la actividad de las empresas, y es por ello que en la mayoría de ellas hoy en día se prima el manejo de herramientas de visualización que facilitan y agilizan la toma de decisiones.

La creciente popularidad de los proyectos de *Big data* y análisis de datos ha hecho que la visualización sea más importante que nunca. Las empresas utilizan cada vez más el aprendizaje automático para recopilar cantidades masivas de datos que pueden ser difíciles y lentas de clasificar, comprender y explicar. La visualización es capaz de acelerar ese proceso y presentar la información de manera comprensible. La visualización de grandes conjuntos de datos requiere potentes sistemas informáticos que recojan los datos en bruto, los procesen y los conviertan en representaciones gráficas que los humanos puedan utilizar para extraer rápidamente información [25].

Las representaciones además pueden revelar características que los modelos pasen por alto, es útil para la limpieza de datos, la exploración, la detección de valores atípicos y patrones destacables, la identificación de tendencias y la presentación de los resultados. Es esencial para el análisis exploratorio de datos y ayudar a familiarizarse con la estructura y las características de los datos con los que se está trabajando [28]. Y al igual que los gráficos son útiles para comprobar los resultados de los modelos, también puede ser útil a la inversa, para reforzar la idea que sugieren inicialmente los gráficos [29].

Debido a todas las aplicaciones posibles que la visualización de datos tiene en el escenario actual de diferentes sectores y empresas también existen variedad de herramientas de visualización en el mercado que se ajusten más a sus necesidades dependiendo del caso de uso. A continuación de indican algunas de las más utilizadas:

- Tableau: Herramienta de Business Intelligence que permite la visualización interactiva de datos de fácil manejo para usuarios de todos los niveles.
- Power BI: Herramienta de Business Intelligence de Windows. Es intuitiva, pero con mayor complejidad de uso por lo que está más enfocada a fines empresariales.
- Qlikview: La ventaja principal de esta herramienta es que permite desarrollar conexiones con otras aplicaciones intermedias, pero implicaría conocimientos más avanzados en lenguajes de programación no accesibles a todos los públicos.
- Ploty: Una aplicación simple y de fácil manejo que grandes compañías de tecnología utilizan en su día a día con frecuencia. Su aplicación suele ir ligada a la utilización de su API para realizar proyectos en JavaScript y Python.
- D3.js: Esta herramienta más avanzada está pensada para los desarrolladores con conocimientos técnicos que se ejecuta en JavaScript y usa HTML, CSS y SVG. Actualmente, es la más completa del mercado.

Para la realización de este proyecto, y teniendo en cuenta que uno de los requerimientos definidos es la visualización en tiempo real, la parte de visualización de los datos se desarrollará mediante la librería *open source* Plotly, por su sencillez y su compatibilidad con el lenguaje de programación predominante en el resto de módulos del pipeline, Python.

1.1.5 ANÁLISIS

La última fase del proceso consistirá en la realización de un análisis de los registros de los que se dispone mediante algoritmos de aprendizaje automático. El objetivo será construir un modelo que sea capaz de generar agrupamientos de clientes basados en características y patrones que puedan compartir los diferentes perfiles que se detecten en el conjunto de datos.

Los problemas de este tipo pertenecen a aquellos del tipo no supervisado dentro del aprendizaje automático o Machine Learning. El fin de este tipo de modelos es llegar al resultado sin la participación de un supervisor o profesor que proporcione las respuestas correctas o un grado de error para cada observación. Es frecuente que el número de observaciones sea superior a un problema de aprendizaje supervisado, y que las propiedades sean más complejas. Aportan información sobre las asociaciones entre variables y si pueden o no ser consideradas como funciones de un conjunto más pequeño de variables.

En el aprendizaje supervisado, existen medidas de éxito, que puede utilizarse para juzgar la adecuación en situaciones particulares, pero en el contexto del aprendizaje no supervisado, no existe tal medida; por lo que es difícil determinar la validez de las inferencias que se extraen de los resultados [30].

Conocido también como aprendizaje sin supervisor, el objetivo de estos algoritmos no es entrenarse para poder clasificar o predecir nuevos resultados, sino ser capaz de identificar estructuras entre los datos que se analizan para aprender más sobre ellos. Una de las funciones principales del aprendizaje no supervisado es la agrupación, que consiste en dividir el conjunto de datos en grupos de ejemplos similares [31].

También llamado aprendizaje descriptivo, se caracteriza por buscar patrones que describan los datos de entrada. No se conoce, sin embargo, qué tipo de patrones se buscan ya que los datos no han sido etiquetados previamente a diferencia de en el aprendizaje supervisado, donde se puede comparar la predicción generada por el modelo con los valores reales. Al solo disponer de datos de entrada, el programa debe ser capaz de encontrar la unión existente entre los registros. Este tipo de aprendizaje es muy útil para reducir la dimensionalidad de

los datos reduciendo la pérdida de información. En este caso, sin embargo, se añade un id con la información del grupo al que pertenece cada registro de usuario por lo que se podrá comprobar si el agrupamiento ha resultado exitoso.

Esta rama del aprendizaje automático tiene diversas funcionalidades aplicables en diferentes ámbitos como el reconocimiento de imagen, o la segmentación de clientes en el mundo empresarial. En biología, la clasificación de especies de plantas y animales según sus características, o incluso identificar posibles fraudes de seguros.

En aprendizaje no supervisado podría generalizarse en dos principales tipos de problemas:

- Clustering
- Reducción de la dimensionalidad

En este proyecto se profundizará sobre el primer caso, ya que será el método que se trabajará en el caso de estudio. A continuación, en la Figura 3 se presenta un ejemplo visual del aprendizaje no supervisado:

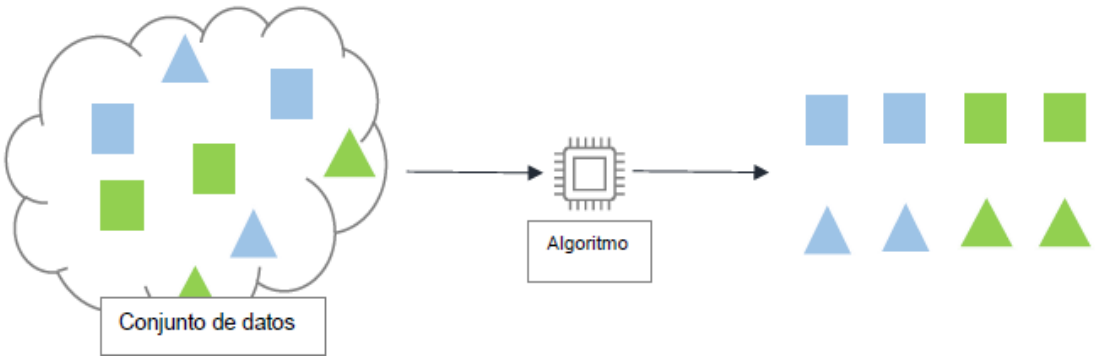

Figura 3. Esquema de algoritmos de aprendizaje no supervisado.

El método más utilizado dentro del aprendizaje no supervisado es el análisis clúster, o análisis de grupo. Lo que se consigue mediante un análisis exploratorio de los datos de entrada es encontrar patrones aparentemente escondidos o agrupar la información para entenderla mejor.

Para obtener esos grupos o “clusters”, el modelo atiende a la similitud entre los diferentes registros definida por diferentes métricas como pueden ser la distancia euclídea o probabilística.

El objetivo principal es la agrupación o la segmentación de una colección de objetos en subconjuntos o "clusters", de manera que los objetos que pertenecen al mismo grupo están más estrechamente relacionados que a los objetos asignados a diferentes grupos. Un objeto puede ser descrito por un conjunto de medidas, o por su relación con otros objetos. Además, es capaz también de organizar los “clusters” en una jerarquía, lo que implica agrupar sucesivamente los propios grupos de manera que, en cada nivel jerárquico, los registros dentro del mismo grupo son más similares entre sí que los de los diferentes grupos. Para llevar a cabo esa segmentación, se tiene en cuenta la similitud (o disimilitud) entre los objetos individuales que se agrupan [30].

Es importante que los grupos generados no se solapen entre sí, cada observación debe pertenecer a un único grupo. Dentro del mismo grupo, las observaciones deben estar relativamente cerca unos de otros, claramente mucho más cerca que los ejemplos de los otros grupos. Los algoritmos suelen necesitar un mecanismo para evaluar la distancia entre un ejemplo y un grupo, como puede ser la distancia euclídea [32].

Un algoritmo muy frecuente en el clustering es el algoritmo de K-medias, por ser fácil de implementar a la vez que eficiente. El algoritmo K-Means tiene como objetivo encontrar y agrupar en clases los puntos de datos que tienen una alta similitud entre ellos. Esta similitud se entiende como lo opuesto a la distancia entre puntos de datos. Cuanto más cerca estén los puntos de datos, más similares serán y mayor probabilidad habrá de que pertenezcan al mismo clúster.

Dado una serie de datos, el objetivo de este algoritmo es dividir el dataset en K clusters, de manera que cada punto es similar al resto de puntos que forman el mismo clúster. En primer lugar, genera k grupos de manera que cada ejemplo pertenezca a un único grupo. Después de esto, se calculan las coordenadas y las distancias de todos los centroides. El centroide más cercano define entonces el “cluster” al que la observación debería pertenecer. Si el ejemplo

ya pertenece a ese grupo inicial, no se realizará ningún cambio; de lo contrario, se transfiere del grupo actual al correcto. Después de la reubicación, los centros de los dos grupos afectados (el que perdió el ejemplo, y el que lo ganó) deben ser recalculados [32].

Este algoritmo divide la base de datos en k grupos, de tal manera que la variación total dentro del grupo es la menor posible. Los resultados obtenidos dependerán de la asignación inicial del número de grupos, por lo que es importante llegar al valor óptimo para la K.

La funcionalidad del algoritmo podría resumirse en dos pasos generales:

1. Para cada centro identifica el subconjunto de puntos de entrenamiento (su grupo) que está más cerca de él que cualquier otro centro.

2. Se calculan las medias de cada atributo para los puntos de cada grupo, y ese vector de medias se convierte en el nuevo centro de ese grupo [30].

En comparación a otros métodos de clustering, este es capaz de procesar en un tiempo más reducido un volumen grande de datos. Además, es relativamente sencillo de implementar. Sin embargo, la principal dificultad de este modelo es la elección idónea del valor k, que condicionará la agrupación posterior.

Dejando a un lado el contexto teórico necesario para la comprensión de los conceptos detrás del objetivo de este proyecto, otro factor importante es la herramienta con la que se ha diseñado el modelo que cumpliera con ese clustering que se quiere conseguir.

Siguiendo la línea del resto de fases del proyecto, el lenguaje de programación utilizado será Python con la ayuda de la librería específica diseñada para algoritmos de aprendizaje supervisado, la librería Scikit-learn. Una librería creada para problemas de aprendizaje automático de software libre que incluye algoritmos de aprendizaje supervisado y no supervisado.

Capítulo 2. DEFINICIÓN DEL TRABAJO

En este capítulo se presentarán los objetivos marcados al inicio del proyecto, así como la metodología y la planificación seguida para cumplir con esos objetivos. También se incluye un estudio económico que ha supuesto el desarrollo del proyecto.

2.1 OBJETIVOS

El principal objetivo del proyecto realizado junto a Iberdrola es el desarrollo completo de un pipeline que incluya la generación, ingesta, almacenamiento, visualización y análisis de curvas de consumo energético en base a diferentes perfiles de clientes.

El modelo por tanto constará de diferentes fases, y cada una de ellas deberá cumplir ciertas especificaciones para la correcta ejecución de cada una de ellas. A continuación, se explayan los objetivos relacionados a cada una de las fases mencionadas que han de cumplirse para el correcto funcionamiento del conjunto del proyecto.

2.1.1 PUNTO 1

Generar datos sintéticos sobre el consumo energético en función de la franja horaria y en base a diferentes perfiles de cliente.

Debido a razones de confidencialidad, no se han facilitado datos internos de la compañía sobre consumo energéticos reales de sus clientes. Por ello, la fase inicial consistirá en la generación de datos aleatorios que simulen los consumos energéticos que reciban en Iberdrola desde los clientes. El objetivo de esta parte es que los datos generados se puedan sustituir por datos reales más adelante y que el modelo dé resultado con la misma efectividad, siendo capaz de diferenciar consumos por perfiles de cliente. Estos perfiles se caracterizan por las franjas horarias en las que se produce el consumo en cuestión, así como el día de la semana en el que ocurre. Este requerimiento condicionará el resto del proyecto, con especial repercusión en la fase analítica ya que su funcionamiento se basará en los perfiles generados

en esta fase inicial, por ello será fundamental la correcta distinción de los diferentes perfiles y las definiciones de sus características de manera eficiente.

Otra característica que se quiere imponer es que sea un modelo en tiempo real, es decir que el periodo y la frecuencia sean reflejo de un escenario posible en la vida real. Para cumplir con este objetivo, se debe tener en cuenta el momento de ejecución y la frecuencia con la que se generan los registros.

2.1.2 PUNTO 2

Ingesta de los datos generados en tiempo real, para poder transportarlos de un punto a otro. En el momento en el que se ejecuta el modelo de generación, la transmisión se realizará en tiempo real para almacenarlos posteriormente en una base de datos accesible que permita el manejo y el análisis de los datos sintéticos obtenidos en la generación.

Será necesaria la conexión del modelo de generación para que todos los registros que se generen lleguen a su destino y con bajas latencias, ya que la ausencia de registros puede perjudicar el análisis posterior que se realice y los resultados que se obtengan de ese análisis. Otra responsabilidad del sistema de ingesta de datos será depositar cada registro en el destino de almacenamiento en el orden de entrada.

2.1.3 PUNTO 3

Visualización de la información que se ha almacenado previamente en tiempo real y de manera agregada para facilitar así el análisis posterior encontrando información relevante que pueda ser utilizada en el análisis. Esto repercutirá de manera positiva en la toma de decisiones ayudando a la comprensión de las curvas de carga mediante representaciones gráficas.

El objetivo principal de esta tarea dentro del proyecto será la representación de manera clara y concisa de la información que expresan los datos, de manera que el usuario pueda interpretar la visualización de forma rápida y con facilidad. Se quiere representar información relacionada a los ciclos de consumo, la actividad y el consumo total en tiempo real, por ende, se reproducirán gráficos tanto en tiempo real como gráficos de información

agregada previamente almacenada. Con esto se quiere conseguir que la información utilizada sea comunicada de forma sencilla para que el usuario pueda sacar conclusiones y evidencias de los datos.

2.1.4 PUNTO 4

Agrupar y clasificar la información almacenada con el fin de segmentar a los clientes, y hacer grupos. Una vez se han podido ingestar y almacenar los datos generados inicialmente, se procederá al estudio del comportamiento de los datos mediante técnicas de Machine Learning con el fin de identificar patrones. Este análisis se llevará a cabo teniendo en cuenta una franja temporal de unos días determinados con la intención de simplificar la tarea.

Se quiere conseguir una segmentación que evidencie las diferencias y singularidades de cada grupo en función de las características que se han recogido. Este agrupamiento de diferentes clases permitirá tomar decisiones en función de a las características que se han extraído de cada una de las clases generadas. Esto permite a la compañía ofrecer servicios personalizados a los clientes que sean clasificados en una clase u otra, ya que se habrá conseguido obtener información sobre los hábitos de consumo de cada perfil.

Una vez se haya completado el proceso entero, debe mantenerse consistente y eficaz frente a futuros escenarios en los que cambien los datos de entrada, y a medida que aumente el número de registros.

Estos serán los principales objetivos que se han tenido en cuenta a la hora de diseñar y desarrollar el proyecto. Al final de esta memoria se analizará si se ha conseguido cumplimentar cada uno de los objetivos marcados.

2.2 METODOLOGÍA

Para llevar a cabo este proyecto, con el fin de conseguir los objetivos marcados y teniendo en cuenta que se compone de varias partes importantes que complementan el proyecto, se ha seguido la metodología agile, primando la funcionalidad del modelo desde un inicio para posteriormente añadir complejidad de forma gradual.

Puede ser interesante poner en contexto la metodología en cuestión para entender mejor su elección a la hora de desarrollar este proyecto. La metodología agile surge en 2001, cuando un grupo de experimentados desarrolladores de software se dio cuenta de que estaban practicando el desarrollo de software de forma diferente a la metodología clásica en cascada. Este grupo, compuesto por referentes en la tecnología como Kent Beck, Martin Fowler, Ron Jeffries, Ken Schwaber y Jeff Sutherland, elaboró el Manifiesto Ágil que documentaba sus creencias compartidas sobre cómo debería funcionar un proceso de desarrollo de software moderno. En este manifiesto se hace referencia a los siguientes principios:

- Individuals and interactions over processes and tools

Las personas y las interacciones por encima de los procesos y las herramientas

- Working software over comprehensive documentation

Software funcional por encima de documentación comprensiva

- Customer collaboration over contract negotiation

Colaboración con el cliente por encima de la negociación del contrato

- Responding to change over following a plan

Responder a los cambios por encima de seguir un plan

Así nació esta metodología tan utilizada en la actualidad y que se ha expandido a diferentes áreas de las organizaciones por ser más efectivo y productivo que el método clásico de cascada [33].

Para llevar a cabo el propósito general del proyecto se distinguen tareas u objetivos menos ambiciosos que se irán cumpliendo de manera secuencial y rápida hasta conseguir completar el proceso con todas sus particularidades.

Todas las tareas realizadas han sido supervisadas por el tutor asignado por la compañía Iberdrola mediante reuniones periódicas semanales donde se informaba de los avances realizados y los problemas que pudieran haber surgido para buscar una solución conjuntamente.

2.3 PLANIFICACIÓN

Las tareas que se han llevado a cabo se describen en este apartado, y posteriormente se representarán en un diagrama de Gantt a modo de expresar el periodo necesario para ejecutar cada tarea.

1. Documentación:

   Lo primero será conocer mejor el contexto, herramientas que sean de utilidad y valores aproximados de consumo reales como base. Una vez se tenga una idea más clara de lo que se quiere conseguir, se buscarán formas de generar datos a modo de sustitución de valores de consumos reales. Este primer estudio de la información y herramientas disponibles en el mercado para poder satisfacer las necesidades del problema es fundamental para poder elegir la opción más adecuada para el caso de uso concreto.

   Una parte fundamental de esta primera fase será la medición de consumos utilizando un medidor de consumo para que los datos utilizados en el proyecto sean verosímiles.

2. Generación de datos sintéticos:

   Este paso condicionará el resto de proceso, por ello será fundamental generar datos que sigan una distribución coherente y reflejen la realidad. Se tendrán en cuenta variables como franjas horarias, el tipo de electrodoméstico, y los perfiles de clientes con sus características correspondientes de manera que el *clustering* posterior tenga sentido.

3. Escribir en kafka:

   Se quiere generar un bus de Kafka que pueda transferir los datos generados. Para ello, y una vez ha sido posible familiarizarse con la herramienta tras su instalación, se generan los *topics* en donde el *producer* escribirá los mensajes.

4. Leer de kafka y escribir en MongoDB:

   Una vez se han escrito los mensajes en Kafka, se deben llevar hasta la base de datos, donde se almacenarán. Por ello es imprescindible relacionar esta base de datos con los *topics* mencionados en el paso anterior donde consumirán todos los mensajes que vayan llegando y se seguirá la organización de un *topic* por colección.

5. Visualización:

   Tras llevar los datos a su destino, esos datos serán representado con la herramienta de generación de visualizaciones Plotly en Python. Será esencial generar gráficos interpretables y que aporten información relevante sobre los datos que se han ingestado, así como mostrar al usuario la llegada de datos en tiempo real.

6. Análisis:

   Se realizará un análisis exploratorio de los datos y las variables que se han almacenado en un periodo de tiempo definido para facilitar el clustering posterior. Se trabajará únicamente con los datos almacenados anteriormente en la base de datos en un periodo de una semana, sin tener en cuenta los registro que puedan llegar en tiempo real.

7. Clustering:

   En esta fase final del proceso se realizará un clustering que permita conocer mejor las características de cada grupo de los datos, que representarán a los diferentes perfiles que podrían identificarse en la compañía. Se verá si la clasificación realizada por el modelo es la adecuada y se analizarán las diferencias percibidas en las que se ha basado el modelo para hacer esa distinción de grupos.

8. Mejoras y añadidos:

   Cuando se haya comprobado el correcto funcionamiento del proceso completo y a unión de las diferentes herramientas que se encargan de cumplimentar los objetivos de cada fase por separado, se podrá añadir complejidad al proceso y mejoras futuras

que permitan hacer del pipeline algo más completo. Entre ellas, añadir más registros, hacer el código más legible o detalles estéticos.

9. Redacción:

   La redacción de la memoria a entregar donde se documentará el desarrollo y la explicación del proyecto realizado.

Estas tareas se han llevado a cabo de forma secuencial y en un periodo de tiempo que se representa en el siguiente diagrama de Gantt en la Figura 4 .

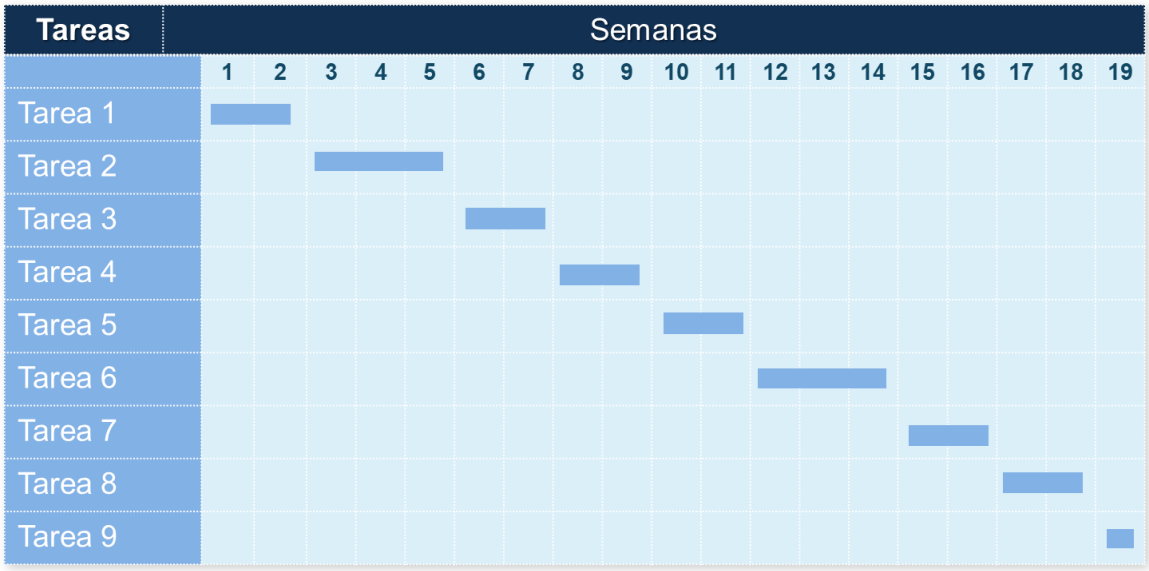

Figura 4. Diagrama de Gantt de las tareas a realizar para llevar a cabo el proyecto.

2.4 ESTIMACIÓN ECONÓMICA

En este apartado se realizarán los cálculos del coste que supone este proyecto. El coste estimado se dividirá en tres categorías: el coste de recursos humano, el coste informático y el coste del equipo utilizado. Se deberían tener en cuenta los gastos asociados al inmovilizado material y al material fungible; No obstante, se ha considerado que en este proyecto no son aplicables y no se tendrán en cuenta para el cálculo de los costes.

2.4.1 COSTE DE RECURSOS HUMANOS

El coste de la mano de obra será el correspondiente a los costes de la persona a cargo de la realización del proyecto. Siendo yo la persona contrada por Iberdrola, y con la ayuda de uno de sus empleados, he sido la única responsable en el desarrollo del proyecto. Por tanto, el único coste de recursos humanos asociado a este proyecto ha sido el sueldo de los cinco meses de contratación para la ejecución del proyecto.

2.4.2 COSTE DE SOFTWARE

Para la ejecución de los diferentes módulos que componen el proyecto se ha hecho uso de varias herramientas tecnológicas. Sin embargo, la instalación y la aplicación de estos medios son de uso gratuito y libre, por lo que el coste informático es nulo.

2.4.3 COSTE DE EQUIPAMIENTO

Finalmente, se ha visto oportuno la adquisición de un medidor de consumo en la fase de generación de datos, para conocer los consumos reales que implican los electrodomésticos seleccionados. En consecuencia, el coste de equipamiento asciende al coste de este medidor de consumo con un precio de 15.99€.

Capítulo 3. SISTEMA/MODELO DESARROLLADO

En este capítulo se describirá el diseño de la aplicación desarrollado conjuntamente y de cada componente por separado, así como los pasos a seguir para la implementación del sistema.

3.1 DISEÑO

El proyecto constará de cinco fases principales que se han desarrollado de manera secuencial para cumplir con los diferentes objetivos mencionados en el apartado anterior. Las fases que se describirán detenidamente a continuación son: generación de datos, ingesta, almacenamiento, visualización y análisis. Cada una de estas fases es complementaria y se han desarrollado en el orden en el que se presentarán a continuación tal y como se puede comprobar en la Figura 5.

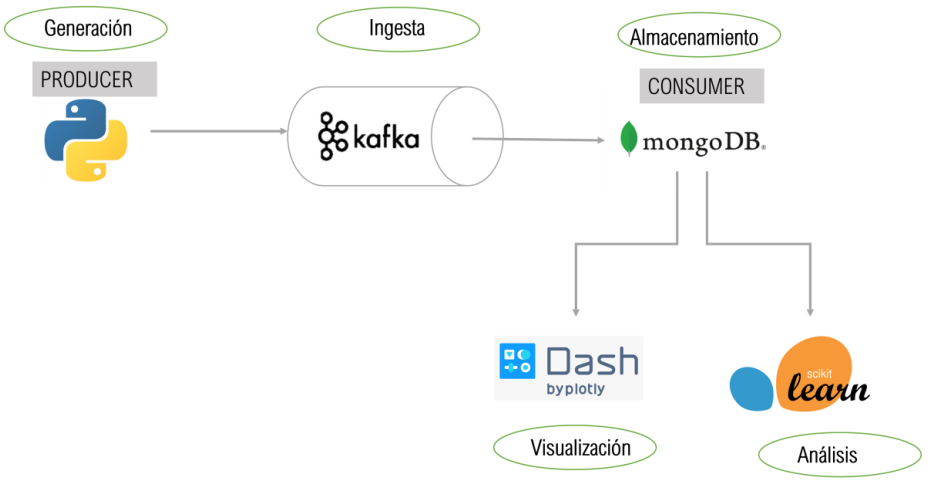

Figura 5. Diseño del proyecto desarrollado.

Además de la ejecución por separado de cada módulo del sistema, la unión entre los módulos será de suma importancia para que los datos lleguen a cada uno de los distintos módulos con la mayor brevedad posible y sin que ningún registro se quede por el camino.

3.1.1 GENERACIÓN DE DATOS

Esta primera parte será la encargada de generar datos que puedan ser sustituidos por datos reales más adelante y que el modelo dé resultado con la misma efectividad.

La generación de datos sintéticos será la primera pauta del pipeline que condicionará el resto del proceso. Debido a razones de confidencialidad, no se han podido facilitar datos reales de consumo energético referentes a los clientes de la compañía. Es por ello, que para la realización del proyecto se ha tenido que hacer uso de datos sintéticos generados como input para el resto de las fases definidas en el pipeline.

Debido a la naturaleza de este proyecto en colaboración con una compañía energética, los datos generados deben simular los consumos energéticos que reciban desde los clientes con una frecuencia de un segundo. Para completar esta tarea con eficacia se han seguido diferentes pasos.

Primero se realizó un estudio de consumos probables acorde a diferentes electrodomésticos, los ciclos que cada uno pueda cumplir y los hábitos de usuarios que se dan en España, que servirán para tomar decisiones futuras.

Lo siguiente fue la definición de perfiles de clientes con características propias relacionadas con franjas horarias. Estos perfiles serán de gran utilidad en el análisis posterior donde se segmentarán los clientes en base a diferentes singularidades. Se han diseñado un total de cuatro perfiles que se presentan a continuación junto con sus características. Para hacer las suposiciones pertinentes se ha hecho una consulta a personas que cumplen o se asemejen a los perfiles descritos. Se quiere recalcar que los perfiles son generalización de diferentes tipos de personas basándose en la información obtenida de personas concretas. Por otra parte, se han querido generar perfiles distintos entre sí para que los resultados que se obtengan en

el análisis sean más relevantes. A continuación, se extiende la descripción de cada uno de los perfiles.

3.1.1.1 PERFIL 1: Estudiantes

Este primer perfil representará un piso donde conviven dos estudiantes con rutinas similares estableciéndose las siguientes condiciones teniendo en cuenta los diferentes electrodomésticos, el día de la semana y la franja horaria en la que sucede el consumo en cuestión.

Dicho esto, se muestran a continuación en la Tabla 1 los criterios tenidos en cuenta en la construcción del modelo de generación para el primer perfil:

Electrodoméstico	Dia laboral	Fin de semana	Nº de veces /semana
Cafetera	*7:00-10:00*	*9:00-12:00*	*7*
Lavadora	*19:00-21:00*	*10:00-13:00*	*2*
Frigorífico	*cte*	*X*	*X*
Tostador	*7:00-10:00*	*9:00-12:00*	*7*
Lavavajillas	*X*	*X*	*X*
Secador de pelo	*7:00- 9:00 o 20:00-21:00*	*9:00-10:00 o 16:00-18:00*	*6*

Plancha	*X*	*10:00-13:00*	*2*

Tabla 1. Características de consumo del perfil 1.

Tal y como se indica en la tabla anterior, los consumos se producen dependiendo de si se trata de día laborable o fin de semana las franjas horarias en las cuales es más frecuente hacer uso de estos electrodomésticos. Cabe destacar la ausencia de un lavavajillas en esta vivienda y que el frigorífico es un aparato que se mantiene encendido en todo momento por lo que no tiene sentido incluir la actividad de encendido o apagado.

3.1.1.2 PERFIL 2: Jubilado

El segundo tipo de cliente que se genere será la vivienda de una persona retirada que vive sola. El estilo de vida de este cliente hipotético será diferente al piso de estudiantes que se ha presentado, es por ello, lógico pensar que también lo serán los hábitos de consumo. Las suposiciones que se han hecho en este caso para medir la actividad de este perfil se recogen en la siguiente Tabla 2.

Electrodoméstico	Dia laboral	Fin de semana	Nº de veces /semana
Cafetera	9:00-11:00		7
Lavadora	12:00-14:00 o 19:30		1
Frigorífico	cte	X	X
Tostador	9:00-11:00		7
Lavavajillas	14:00-15:00 o 20:00-21.00		3
Secador de pelo	x	x	x

Plancha	10:00-13:00	2

Tabla 2. Características de consumo del perfil 2.

Se puede observar que en este caso no se hacen distinciones por día de la semana ya que se trata de una persona sin horario fijo por estudios o trabajo. En este caso sí que se posee un lavavajillas, pero se prescinde del secador de pelo.

3.1.1.3 PERFIL 3: Familia con hijos

En tercer lugar, en este perfil se refleja la vivienda de una familia compuesta por dos adultos y sus dos hijos. De nuevo, los hábitos vuelven a ser diferentes para este tipo de vivienda y se muestran en la Tabla 3:

Electrodoméstico	Dia laboral	Fin de semana	Nº de veces /semana
Cafetera	6:00-9:00	8:00-11:00	7
Lavadora	19:00-21:00	10:00-13:00	4
Frigorífico	cte	X	X
Tostador	6:00-9:00	8:00-11:00	7
Lavavajillas	20:30-22:30	20:30-22:30	7
Secador de pelo	7:00- 9:00 o 20:00-21:00	9:00-10:00 o 16:00-18:00	6
Plancha	19:00-21:00	10:00-13:00	3

Tabla 3. Características de consumo del perfil 3.

Al aumentar el número de habitantes de la vivienda, también será mayor la actividad energética y, por tanto, el consumo total. Además, se puede comprobar que en este caso todos los electrodomésticos están en uso y se utilizan con frecuencia.

3.1.1.4 PERFIL 4: Trabaja de noche

Por último, también se tendrán en cuenta aquellos clientes que tengan un horario de trabajo diferente, con una jornada laboral nocturna y un estilo de vida condicionado por ese horario. Las características que tendrán los registros pertenecientes a este perfil son los siguientes que se muestran en la Tabla 4:

Electrodoméstico	Dia laboral	Fin de semana	Nº de veces /semana
Cafetera	19:00-21:00	9:00-12:00	7
Lavadora	16:00-18:00	10:00-13:00	1
Frigorífico	cte	X	X
Tostador	19:00-21:00	9:00-12:00	7
Lavavajillas	14:00-15:00 20:00-21.00	14:00-15:00 20:00-21.00	3
Secador de pelo	X	X	X
Plancha	16:00-18:00	10:00-13:00	1

Tabla 4. Características de consumo del perfil 4.

La diferencia principal con el resto de clientes es la franja horaria de uso de los electrodomésticos por la razón que se ha comentado. Se repite la ausencia de un secador de pelo.

Cuando ya se han definido los cuatros perfiles con sus características correspondientes, lo siguiente ha sido una realizar una medición de los consumos reales de los electrodomésticos participantes en el proyecto mediante un medidor de consumo. Se han tenido en cuenta los valores reales que se han registrado en una residencia familiar para cada electrodoméstico, así como la duración aproximada de cada ciclo de uso como base para el módulo de generación.

Se han seleccionado y medido un total de siete electrodomésticos que suelen ser comunes en cualquier vivienda familiar y que suponen los consumos más elevados en la factura energética.

Una vez se tiene recogida la información que se utilizará sobre los consumos de los electrodomésticos, se procede a la generación de los datos que servirán como entrada al resto del pipeline. La generación, que se ejecutará con el lenguaje de programación Python, estará compuesta por tres vértices. En diferentes tablas se recogerá información que se necesitará para las fases futuras como la visualización y el análisis del conjunto de datos.

Uno de esos tres *datasets* se encargará de almacenar los ciclos de cada electrodoméstico teniendo en cuenta los valores obtenidos con el medidor de consumo con frecuencia de diez segundos por consumo. El segundo *dataset* refleja la actividad de cada perfil en las franjas horarias seleccionadas. Por ello, se producirán cuatros generadores, uno por cada perfil descrito previamente y que cumpla con las características definidas. Por último, será fundamental calcular el consumo total teniendo en cuenta los ciclos de cada electrodoméstico y la actividad en base a los cuatro perfiles mencionados. Estos datos se irán almacenando en una base de datos, y por otra parte también se generarán datos en tiempo real siguiendo la misma estructura, pero con una frecuencia de un segundo y que se almacenarán en una base de datos distinta para posteriormente acceder a ella en el módulo de visualización.

A modo de simplificar el problema y reducir el tiempo de ingesta de la fase posterior, los generadores se han reproducido para un periodo de tiempo de una semana laborable, con el comienzo definido en el preciso momento que se ejecute el modelo. La hora y el día de la

semana en el que se genere cada registro será lo que condicione la actividad relacionada a cada perfil de consumidor.

Los datos generados en esta fase inicial serán trasladados tal y como se indica en la siguiente etapa del proyecto a una base de datos. A pesar de que no se trate de datos de clientes reales facilitados por la compañía, la generación de datos será fundamental en el pipeline para que en futuras consideraciones se puedan reemplazar dichos datos por registros que provengan de clientes de Iberdrola y el resto del proceso no se vea afectado y dé resultado con la misma efectividad.

3.1.2 INGESTA

Para que el modelo sea capaz de mover los datos generados en la fase previa será necesario la utilización de una plataforma distribuida de transmisión de datos con la característica de que esa transmisión sea en tiempo real. Por ello, tal y como se ha mencionado en el capítulo anterior, la participación de apache Kafka es fundamental para el correcto funcionamiento del flujo de datos.

El destino de los datos generados será un sistema de almacenamiento no relacional, MongoDB. Por tanto, en este caso, el *producer* o el productor de eventos ha sigo el generador desarrollado en el lenguaje de programación Python, y el consumidor de dichos eventos será por consiguiente MongoDB.

Otro de las componentes necesarias en cualquier despliegue de Kafka, tal y como se ha explicado en el capítulo dos de esta memoria, serán los *topics* en los que se escribirá y de los que se leerá toda la información que se quiera mover.

En este caso, los eventos se han repartido en diferentes *topics* en donde posteriormente el *consumer* se subscribe. Se han generado tres *topics* en total, uno por dataframe generado, tal y como se ha indicado en el apartado anterior. De esta manera, se consigue mantener los registros que aportan información diferente en diferentes colecciones en la base de datos de MongoDB. La conexión intermediaria de Kafka y MongoDb se ha desarrollado con un script

de Python que permite conectar al mismo tiempo Apache Kafka y MongoDB. El script que cumple con esta función se puede encontrar en el anexo A.

En esta fase del proyecto, se ha conseguido por tanto que los datos generados sean trasladados de manera continua y en tiempo real a un sistema de almacenamiento a medida que se van generando, y con posibilidad de añadir indistintamente un número mayor de registros de forma rápida y eficiente. La instalación de esta herramienta se incluye en el siguiente apartado de este capítulo.

3.1.3 ALMACENAMIENTO

Como se podido ver, es sistema de almacenaje elegido para la realización de este proyecto ha sido MongoDB, donde poder conservar los registros que se hayan generado. Ya se ha repetido en varias ocasiones a lo largo de esta memoria, los datos que se producen en el primer módulo son transportados mediante colas Kafka y se terminan depositando en una base de datos MongoDB. Tal y como se ha indicado, se han generado tres *topics* diferentes que hacen referencia a información proveniente de diferentes *dataframes* acorde a la información que aportan. Por ello, también se generarán tres colecciones en la base de datos de Mongo. Cada una de las colecciones ingestarán los datos recibidos de un *topic* diferente, siendo así el resultado una base de datos de tres colecciones con registros históricos que se irán recibiendo en tiempo real gracias a subscribirse a los *topics* de Kafka.

Por otra parte, existirá una segunda base de datos con la misma estructura que se generará en tiempo real y servirá como entrada a una parte fundamental de las visualizaciones posteriores. Esta base de datos irá creciendo a medida que se ejecute el modelo en tiempo real, y servirá como intermediario antes de ser visualizados y al mismo tiempo como lugar donde se almacenen para poder ser consultados en caso de que se detecte algún error en la recepción.

Para que esta estructura de lectura en tiempo real se lleve a cabo es esencial definir como *consumer* el cliente de Mongo en Python, *pymongo*. Los parámetros requeridos para el funcionamiento correcto de ese bus de Kafka al igual que su instalación en el entono de trabajo se explicarán en el siguiente apartado con mayor precisión.

A medida que se reciben los registros procedentes del generador se almacenarán con el fin de poder acceder y realizar consultas a posterior, que serán necesarios en las siguientes etapas del proyecto.

3.1.4 VISUALIZACIÓN

La visualización de datos es fundamental en un pipeline de procesamiento del dato, para facilitar el análisis y la comprensión de los datos que se tienen almacenado. Para este caso de uso concreto donde uno de los requerimientos es que la ingesta sea en tiempo real, será interesante que la visualización se capaz de seguir esa ingesta y mostrar así los datos que se vayan recibiendo y los atributos que se quieran representar.

Sin embargo, también se querrá reflejar información relevante en relación a los históricos almacenados en la base de datos para poder analizar posteriormente y conocer en mayor profundidad lo que los datos puedan aportar.

Para llevar a cabo esta fase de visualización y siendo conscientes de que debe cumplir con las exigencias establecidas previamente, se ha decidido hacer uso de la librería Dash de Plotly disponible para Python. Se trata de una herramienta diseñada para construir aplicaciones analíticas web que se muestran en el navegador web [34].

A continuación, en la Figura 6 se muestra el *dashboard* creado mediante Dash compuesto por un total de cuatro representaciones gráficas basadas en los datos recogidos en la base de datos de Mongo y en los que se puedan recoger en el momento en el que se despliega el navegador web y se estén ingestando los datos en ese momento preciso.

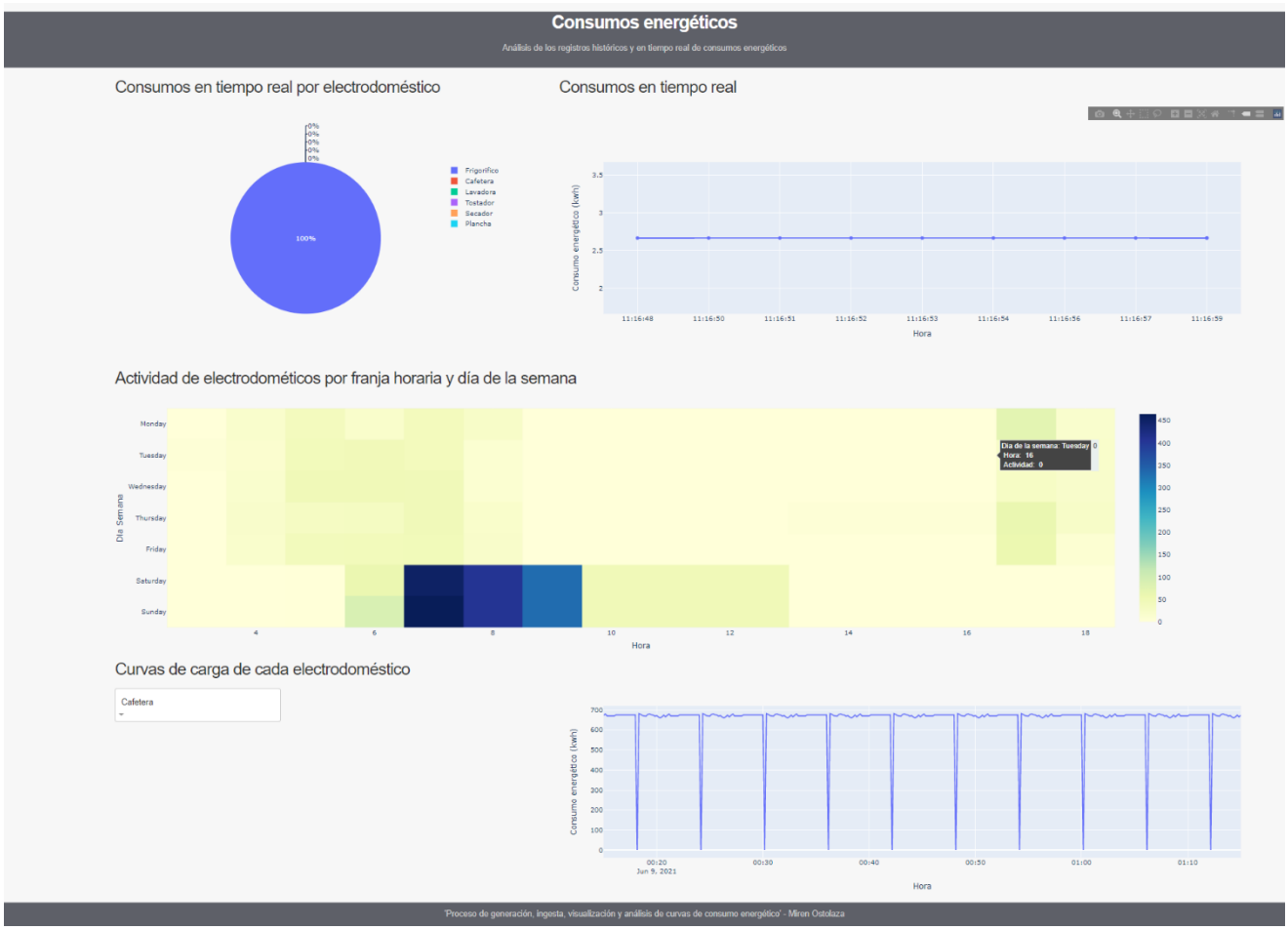

Figura 6. Dashboard generado en el módulo de visualización.

Por lo tanto, podría dividirse el *dashboard* entre las visualizaciones basadas en procesamiento en tiempo real (gráfico 1 y gráfico 2) y aquellos que recogen todos los datos históricos almacenados de los que se dispongan para el periodo de tiempo analizado. A continuación, se procederá a ofrecer una breve explicación de cada una de las visualizaciones y las variables que se requieren para su construcción.

3.1.4.1 Gráfico 1: Consumos en tiempo real por electrodoméstico

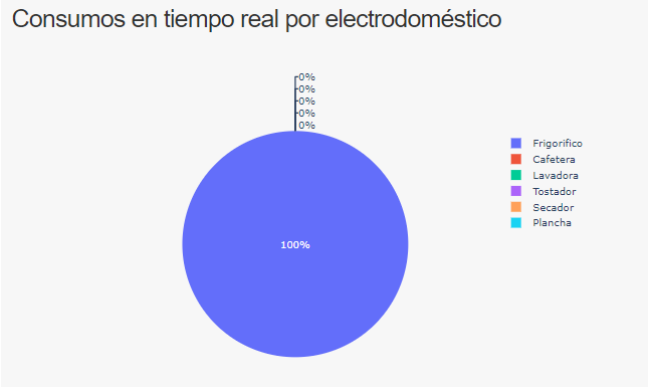

Figura 7. Gráfico 1 del dashboard.

Tal y como se ha mencionado, la fuente de datos será en tiempo real, es decir, los inputs de esta visualización serán aquellos registros que se lean desde Kafka y se almacenen en Mongo en el momento en el que se esté ejecutando el modelo. Se tendrán en cuenta los últimos 10 registros recibidos y la visualización se actualizará cada segundo.

En este pie-chart mostrado en la Figura 7 se representa el consumo total registrado para el segundo en concreto y dividido en cada uno de los electrodomésticos activos en ese instante. Se conocerá así el porcentaje al que pertenece cada electrodoméstico en el instante recibido el consumo.

3.1.4.2 Gráfico 2: Consumos en tiempo real

La Figura 8 será el segundo gráfico que se base en los datos recibidos en tiempo real, y que será actualizado en función de los nuevos datos de entrada. En este caso, el gráfico de líneas proporciona el valor del consumo total en cada instante registrado. De nuevo, se muestran los diez últimos datos recibidos.

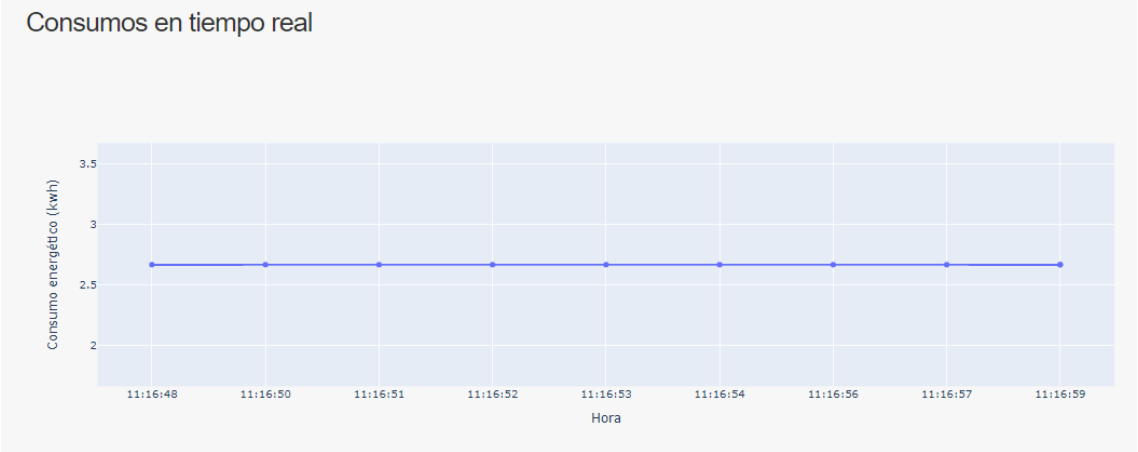

Figura 8. Gráfico 2 del dashboard.

3.1.4.3 GRÁFICO 3: Actividad de la cafetera

Este *heatmap* de la Figura 9 visualiza datos históricos almacenado en la base de datos previamente. Representará la actividad del electrodoméstico en cuestión por día de la semana y hora del día. Permitirá saber más sobre los hábitos de consumo de los clientes y cuáles son las horas punta de mayor consumo. Esta actividad representa el número de aparatos encendidos en esas franjas.

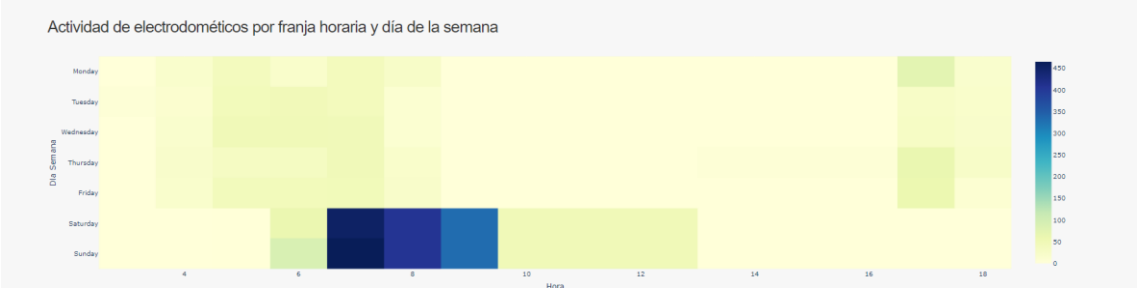

Figura 9. Gráfico 3 del dashboard.

Al tratarse de un *heatmap*, el rango de actividad se refleja mediante una tercera variable que será el color. El gradiente incluido a la derecha informa de las franjas de valores que recoge.

3.1.4.4 Gráfico 4: Ciclos de consumo

Por último, en la Figura 10 se muestran las curvas de carga de los electrodomésticos que se han tenido en cuenta para este proyecto. La visualización estará conectada al seleccionador que se encuentra en el lado izquierdo del gráfico, se puede seleccionar el electrodoméstico que se quiere mostrar. Permitirá conocer los consumos que implican cada uno de los electrodomésticos, así como la duración de los ciclos y los patrones que puedan seguir durante el ciclo. La franja horaria que se ha tenido en cuenta en la construcción de estos gráficos ha sido de una hora.

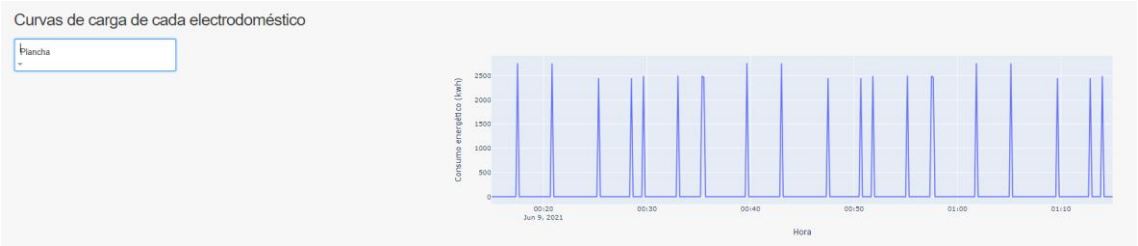

Figura 10. Gráfico 4 del dashboard.

Además de las visualizaciones, los elementos estéticos como el tamaño, posición, títulos y demás han sido editados para que el aspecto del *dashboard* completo fuera más atractivo e interactivo.

Todas estas visualizaciones aportan varias conclusiones importantes sobre los datos que y que se detallarán con mayor precisión en el capítulo siguiente.

3.1.5 ANÁLISIS

Por último, con el fin de realizar la segmentación de clientes en función de diferentes perfiles identificados, se ha optado por la librería Scikit Learn de Python. Esta librería que ofrece gran variedad de algoritmos de clasificación, regresión o clustering ha sido suficiente para

cumplir, aunque en este caso será suficiente con el algoritmo K mean perteneciente al grupo de clustering.

En esta fase y con la ayuda de otra librería muy conocida de manejo de dataframes, Pandas, se ha realizado primero un análisis exploratorio inicial para tener una idea más clara de lo que recogen los registros almacenados en la base de datos de históricos. Como parte de ese estudio inicial, se ha querido conocer las horas en las que se producen más consumos, así como el día de la semana con mayor actividad. Para ello se ha calculado la media de cada hora y de cada día sin hacer más distinciones en los datos. Los resultados de estos cálculos se muestran en el capítulo 4 de esta memoria.

Para facilitar los modelos siguientes, se ha generado una tabla dinámica teniendo en cuenta únicamente las siguientes variables: ID, Dia de la semana, Hora del día y el consumo total. Esta tabla será de ayuda para el modelo de *machine learning* posterior. A continuación, se muestra parte de la tabla generada para facilitar la comprensión de lo que se está hablando en la Figura 11:

	Total										...	6		
weekday	0										...	6		
hour	0	1	2	3	4	5	6	7	8	9	...	14	15	16
ID														
P1U1	957.453	960.12	960.12	960.12	960.12	960.12	960.120	19794.642	21238.575	11116.242	...	960.12	960.12	960.120
P1U2	957.453	960.12	960.12	960.12	960.12	960.12	960.120	17054.042	15521.375	17947.442	...	960.12	960.12	960.120
P1U3	957.453	960.12	960.12	960.12	960.12	960.12	960.120	12783.442	12780.775	13679.042	...	960.12	960.12	960.120
P2U1	957.453	960.12	960.12	960.12	960.12	960.12	960.120	960.120	960.120	17906.842	...	960.12	960.12	960.120
P2U2	957.453	960.12	960.12	960.12	960.12	960.12	960.120	960.120	960.120	19614.642	...	960.12	960.12	960.120
P2U3	957.453	960.12	960.12	960.12	960.12	960.12	960.120	960.120	4335.120	19614.642	...	960.12	960.12	960.120
P3U1	957.453	960.12	960.12	960.12	960.12	960.12	15346.242	8555.642	18896.375	957.453	...	960.12	960.12	960.120
P3U2	957.453	960.12	960.12	960.12	960.12	960.12	17054.042	11116.242	5990.175	957.453	...	960.12	960.12	960.120

Figura 11. Tabla dinámica generada con los datos.

Esta tabla presentada será la que se ha utilizado como entrada del modelo de aprendizaje automático donde se quieren crear agrupaciones en función de las variables introducidas. El parámetro más importante de un clustering y que necesita el algoritmo de k means, será el valor de k, o el número de grupos que se quieren o se deben formar en base a la información

que se tiene. Uno de los métodos más utilizados que calcula el valor óptimo de esa k para el problema en cuestión es el método de Silhouette. Esta técnica asignara una puntuación a diferentes valores de k para conocer el mejor. Representando gráficamente los valores y las puntuaciones de Silhouette para nuestro problema se obtiene lo siguiente:

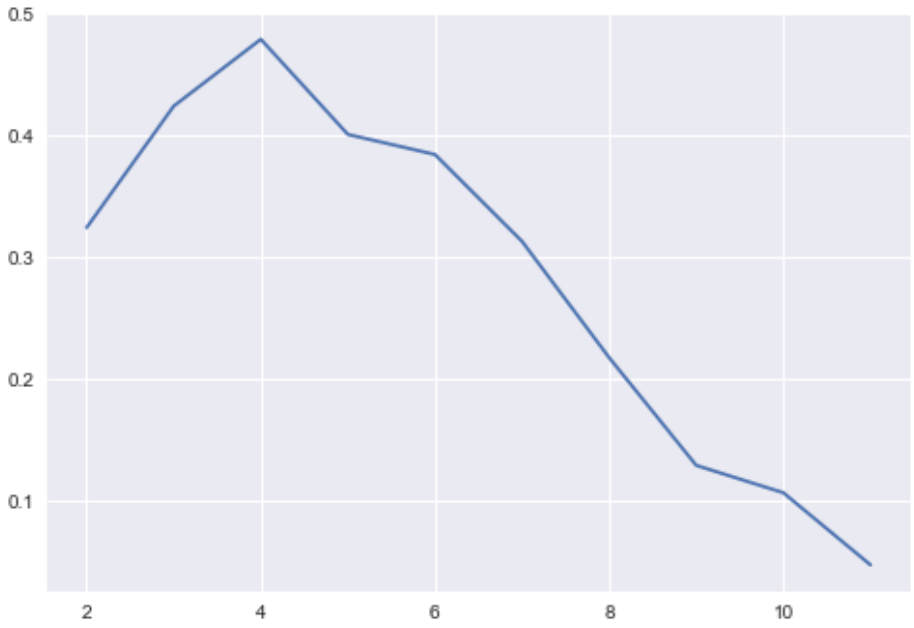

Figura 12. Representación del valor óptimo de k.

En la Figura 12 se puede observar claramente que el valor optimo que se debería tener en cuenta en la clusterización de estos datos es de k=4. Por ello, en el algoritmo de *k means* se introduce manualmente este valor de k para que cada registro del conjunto de datos sea asignado a uno de los cuatro grupos que se generan.

Si de nuevo se representa visualmente lo que devuelve el modelo de *Machine Learning* se consigue lo siguiente:

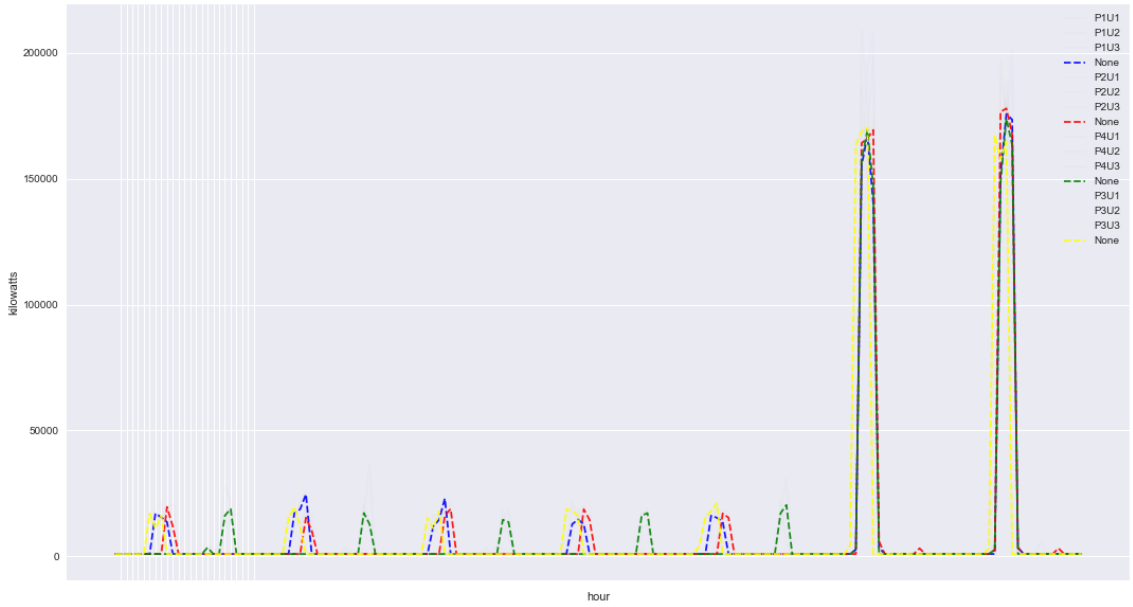

Figura 13. Representación de los clusters generados.

Se puede comprobar que el modelo ha agrupado los registros en cuatro clases diferentes en función de los patrones identificados en el *dataset* y además se puede afirmar que la clasificación que ha realizado en modelo es la correcta observando el ID de los datos pertenecientes a cada *cluster*. Los resultados obtenidos en esta fase se analizarán y comentarán en mayor profundidad en el siguiente capítulo de esta memoria.

3.2 IMPLEMENTACIÓN

En este apartado se proporcionará una breve descripción de cómo han sido implementados las diferentes tecnologías y herramientas necesitadas para las partes que componen la aplicación siguiendo la arquitectura que se ha definido en el apartado anterior.

Los diferentes componentes han sido desarrollados mediante el lenguaje de programación Python al igual que la cohesión entre ellos. Debido a las diferentes características y naturalezas de cada una de las partes, también se han necesitado diversas librerías por lo que se ha visto apropiado la instalación de un entorno virtual que englobara todas esas librerías teniendo en cuenta las versiones y no entren en conflicto con librerías previas que puedan

estar ya instaladas. Estas librerías estarán recogidas en el archivo requirements.txt, y se instalarán una vez se haya creado y activado el entorno virtual.

3.2.1 APACHE KAFKA

Una de las herramientas principales que permiten el correcto funcionamiento y sirve como unión entre una componente y otra del modelo es la ingesta mediante colas Kafka que se encarga de transportar los registros desde su origen a su destino. Para ello, se ha requerido la implementación de la herramienta Kafka. Para la instalación de esta plataforma en Windows será necesario disponer de JAVA 8 SDK previamente. En este caso, la versión elegida ha sido kafka_2.12-2.8.0 que se puede encontrar en la página oficial (https://kafka.apache.org). Tras instalarlo se debe configurar de modo que los directorios donde se almacenen los logs se sitúen donde el usuario desee. Tras esto, y siempre que se quiera empezar con la ingesta de registros, será necesario inicializar el servidor Zookeaper desde la línea de comandos tal y como se indica a continuación:

```
.\bin\windows\zookeeper-server-start.bat .\config\zookeeper.properties
```

Posteriormente desde otra consola de comandos se ejecuta el servidos de Kafka con el comando que se muestra a continuación:

```
.\bin\windows\kafka-server-start.bat .\config\server.properties
```

Con esto ya se tendría la plataforma en funcionamiento. Otro elemento fundamental que ya se ha mencionado en el capítulo anterior es el *topic* donde escribirá el *producer* y se suscribirá el *consumer*. Para generar un *topic* se utiliza el siguiendo comando:

```
.\bin\windows\kafka-topics.bat --create --zookeeper localhost:2181 --replication-
factor 1 --partitions 1 --topic sampleTopic
```

3.1.2 MONGO

Tras cumplimentar los pasos comentados, será posible empezar a enviar los registros que se generen en la fase de generación. Siguiendo la línea de ejecución presentada, esos datos

serán recibidos para almacenarse en un sistema de almacenamiento NoSQL elegido. Por tanto, el siguiente requerimiento será la puesta en marcha de Mongo DB accediendo a la página oficial.

Bastaría con seguir los pasos explicados en la página web para poder utilizar mongo desde la consola de comandos. Sin embargo, a modo de simplificación y conseguir una interfaz más manejable e interpretable se ha decidido instalar también Mongo DB Compass, que se trata de una GUI (Interfaz gráfica de usuario) como alternativa a la Shell de Mongo que permite explorar, analizar e interactuar con el contenido almacenado en una base de datos MongoDB sin necesidad de utilizar *queries*.

3.1.3 DRIVERS

Cuando se han instalado correctamente las herramientas mencionadas se debe asegurar la unión entre las mismas de forma que los registros generados lleguen a su destino sin percances y en la mayor brevedad posible. Este objetivo se conseguirá mediante las llamadas *producer* y *consumer* que se configuran en distintos scripts de Python. Cada una de ellas debe tener definidos unos parámetros específicos para que el sistema sea ejecutable. Se han necesitado diferentes drivers para la comunicación entre los sistemas.

- Kafka-Python: Cliente de Python para el sistema de procesamiento de flujos distribuidos Apache Kafka. kafka-python está diseñado para funcionar de forma muy parecida al cliente oficial de Java. Se ha hecho uso de la API KafkaProducer, que permite publicar los registros en el *topic* que se indique. Del mismo modo, la API KafkaConsumer es un consumidor de mensajes de alto nivel, indicando el *topic* o los *topics* a los que subscribirse y el servidor al que se debe conectar que por defecto será localhost:9092 [35].

- Pymongo: esta API servirá como conexión con la base de datos Mongo desde Python. Genera un cliente mediante la configuración de un servidor que será 'localhost:27017' y permite acceder a las diversas colecciones de forma rápida y directa [36].

3.2.4 DASH

Para la fase de visualización y con el fin de cumplir con los objetivos definidos se ha pensado en Dash que es una biblioteca de Python de código abierto pensada para crear aplicaciones basadas en aplicaciones web.

Se trata de una biblioteca de interfaz de usuario para crear aplicaciones web analíticas de manera sencilla para el usuario y que ayuda a los científicos de datos a construir aplicaciones web analíticas sin requerir conocimientos avanzados de desarrollo web.

El código de la aplicación Dash es declarativo y reactivo, por ello la creación de aplicaciones complejas es más simple que con otras herramientas y no requiere de un conocimiento previo demasiado avanzado para su uso. Los elementos estéticos que lo componen son personalizables: El tamaño, el posicionamiento, los colores, las fuentes… mediante CSS en función de la estética que se quiera conseguir y al ser desplegado en el navegador web, no es necesario escribir ningún Javascript o HTML [37].

Esta librería de código abierto está compuesta por tres tecnologías con diferentes propósitos [38]:

- Flask proporciona la funcionalidad del servidor web.
- React.js renderiza la interfaz de usuario de la página web.
- Plotly.js genera los gráficos utilizados en su aplicación.

Dicho esto, y para desarrollar la aplicación de visualización con esta herramienta, se ha generado un nuevo directorio donde se almacena el código y un entorno virtual limpio de Python 3. Los comandos que permitirán hacer esto serán los siguientes:

```
$ mkdir nombre_dir && cd nombre_dir
$ python3 -m venv venv
$ source venv/bin/activate
```

Una vez activado el entorno virtual generado se podrán instalar las librerías. Las que se muestran a continuación serán esenciales para conseguir el resultado deseado en la aplicación web:

- dash para el despliegue de la aplicación.
- dash_core_components para crear componentes interactivos como gráficos, desplegables etc.
- dash_html_components para acceder a las etiquetas HTML.
- pandas para facilitar el manejo de los datos.

Con estas librerías se ha podido realizar el diseño del *dashboard* que se ha mostrado en el apartado anterior. Para poder ejecutar la aplicación se debe incluir el siguiente comando en el código de la aplicación:

```
if __name__ == "__main__":
49     app.run_server(debug=True)
```

Una vez se tiene la aplicación definitiva y tras ejecutarlo, para poder ver el dashboard diseñado se deberá acceder a *http://localhost:8050* en el navegador.

3.2.5 SCIKIT-LEARN

Para la parte de análisis, se ha utilizado la librería skicit-learn, una librería creada para aprendizaje automático de software libre que incluye algoritmos de aprendizaje supervisado y no supervisado.

Para la instalación y set up en el entorno local, ha sido suficiente con instalar la librería mencionada. Por otra parte, para la parte analítica del proyecto se decidió hacer uso del entono Jupyter de Anaconda. Jupyter es un lenguaje agnóstico y soporta entornos de ejecución en diferentes lenguajes, entre ellos Python.

De esta forma, la obtención de resultados por cada celda ha facilitado la comprensión y la corrección de erratas además de devolver instantáneamente las representaciones gráficas que se han visto recomendables para este módulo analítico.

Capítulo 4. ANÁLISIS DE RESULTADOS

En este capítulo se comentarán en detalle los resultados obtenidos tras la ejecución e implementación del modelo. Como se ha repetido en varias ocasiones, el proyecto está compuesto por cinco principales núcleos construidos cada uno para su propio fin. A continuación, se hará hincapié en los resultados obtenidos en las fases de almacenaje, visualización y análisis.

4.1 ALMACENAJE

Una vez los registros han podido ser trasladados a su destino donde serán almacenados para su posterior estudio, se pueden distinguir las tres colecciones pertenecientes a la base de datos donde se han almacenado los registros referentes a un periodo de una semana. La primera colección recoge los ciclos de consumo de cada electrodoméstico cuando está en uso. Por tanto, habrá una variable por electrodoméstico además de la variable "Date" que indicará el momento preciso del día en segundos almacenado como *timestamp*. Esta variable será la que se multiplique con su correspondiente de la segunda colección para conocer el consumo total en ese instante. Esta tabla también registra un "_id" asignado por Mongo y el *timestamp* del momento en el que es recibido desde Kafka.

A continuación, en la Figura 14 se muestra la estructura de esta primera colección extraída desde la aplicación GUI Mongo DB Compass donde se puede apreciar que está compuesta de 8639 documentos.

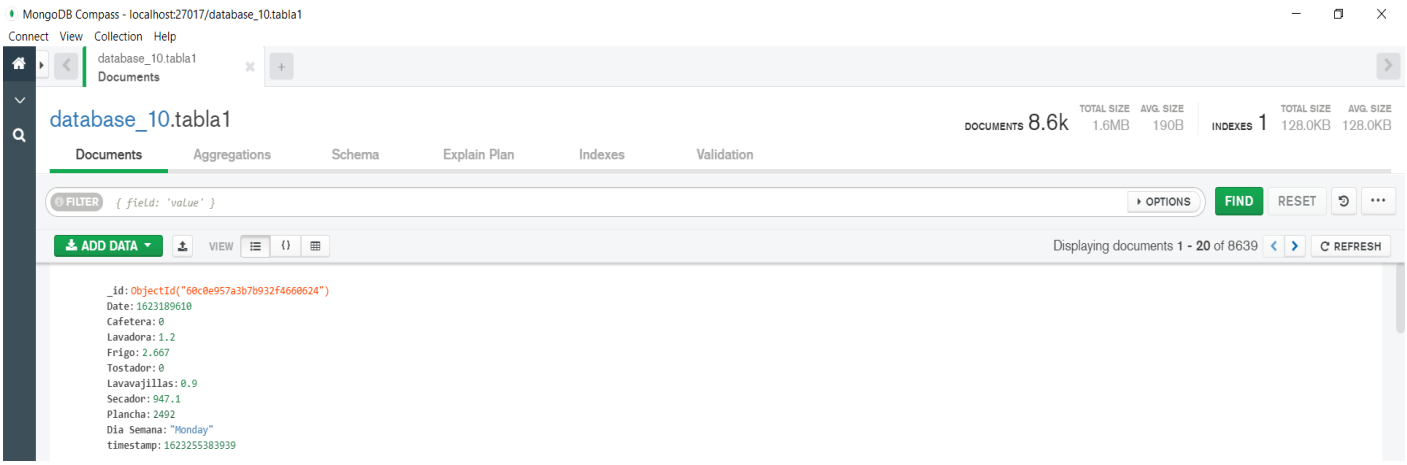

Figura 14. Primer registro almacenado en la colección 1 de la base de datos en Mongo.

Por simplificación del modelo y reducir el tiempo de ejecución la frecuencia entre registros es de diez segundos, y será suficiente con cubrir los pertinentes a un día ya que los ciclos son repetitivos.

La siguiente colección que forma parte de la base de datos es la encargada de definir la actividad de cada usuario acorde al perfil al que pertenecen. Esto se ha conseguido mediante unas condiciones en base a la hora y el día de la semana en el que sucede el consumo. El resultado, por tanto, es una colección que indica en cada registro qué electrodoméstico estará encendido y permanecerá encendido lo que dure ese ciclo de uso. En la Figura 15 se muestra el primer documento de esta segunda colección para comprender mejor lo comentado.

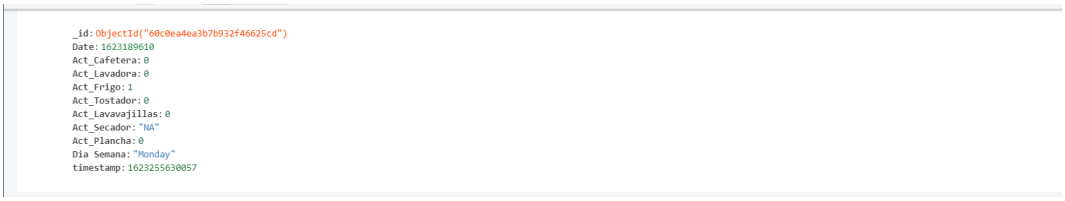

Figura 15. Primer registro almacenado en la colección 2 de la base de datos en Mongo.

Por las razones explicadas, las variables "Día Semana" y "Date", que recoge la fecha y la hora exacta del consumo, serán los que condicionen la actividad en las variables de cada electrodoméstico, siendo estas variables booleanas que reflejarán un 1 cuando se dé el caso de estar encendidas. En el documento mostrado en la Figura 15 se puede observar que el único aparato encendido es el frigorífica, el cual siempre permanece en activo. Por otra parte, el valor correspondiente al secador de pelo devuelve un "NA", porque tal y como se menciona en el capítulo 3, algunos de los perfiles descritos no poseen este aparato en sus viviendas.

Por último, esta base de datos también facilita información sobre los consumos totales que se han registrado en las viviendas de los usuarios cada diez segundos durante una semana. Esto será resultado de la multiplicación de los ciclos de consumo reflejado en la primera colección y la actividad reflejada en la segunda colección. Se muestra a continuación en la Figura 16 la apariencia de este último componente de la base de datos.


```
_id: ObjectId("60d0a92fa3b7b90e38c6c93a")
Date: 1624226410
Cafetera: 0
Lavadora: 0
Frigorifico: 2.667
Tostador: 0
Lavavajillas: 0
Secador: "NA"
Plancha: 0
Total: 2.667
Dia Semana: "Monday"
ID: "P4U1"
timestamp: 1624287504217
```


Figura 16. Primer registro almacenado en la colección 3 de la base de datos en Mongo.

En él se puede ver que corresponde a lo visto en la Figura 14 y en la Figura 15, y que efectivamente el consumo total sería únicamente el respectivo al frigorífico. Además de las variables vistas anteriormente, se añade una nueva variable que cobrará importancia en el módulo analítico de este proyecto. Se trata de la variable "ID", donde se refleja el perfil al que pertenecen y el número de usuario dentro de ese perfil. En este caso, se puede apreciar que se trata del registro de consumo del usuario 1 que forma parte del perfil 1. Cabe recordar

que en total se ha trabajado con cuatro perfiles distintos y tres usuarios pertenecerán a cada uno de los cuatro perfiles.

Por otra parte, cuando el modelo se está ejecutando también llegarán esos registros a una base de datos de Mongo en tiempo real. En este caso, los registros ocurrirán con un periodo de un segundo y serán reproducidos en la parte de visualización. Esta segunda base de datos mantendrá la misma estructura de las tres colecciones mencionadas previamente.

4.2 VISUALIZACIÓN

El siguiente módulo del que merece la pena comentar lo logrado es el de visualización, donde gracias a las representaciones construidas se ha podido deducir información de los datos del conjunto.

Tal y como se menciona en el capítulo anterior, las primeras dos visualizaciones mostrarán lo recibido en esos momentos en los que el modelo completo se está ejecutando para ser visualizados en tiempo real, por lo que la información que puedan aportar será cambiante dependiendo del momento en el que se ejecute. Por consiguiente, este capítulo se va a centrar en aquellas visualizaciones que reflejen información correspondiente a los datos ya almacenados anteriormente en la base de datos que se acaba de mostrar.

Se ha querido conocer qué franjas horarias son las más activas, así como el día en la semana en el que se producen más consumos. Para conocer esta información se ha hecho uso de la segunda colección donde se refleja la actividad de los electrodomésticos, y tras realizar los cálculos apropiados como sumar la actividad total por registros para tener en cuenta todos los electrodomésticos encendidos en ese momento, y agruparlo por día y hora se ha obtenido el siguiente mapa de calor que se muestra en la Figura 17.

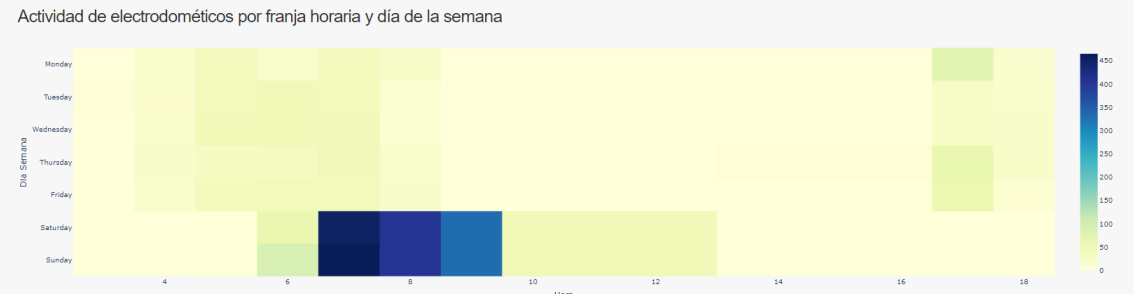

Figura 17. Gráfico que recoge la actividad de los electrodomésticos por hora y día de la semana.

En él se puede ver claramente, que la mayor actividad, o expresado de otra forma, el mayor número de electrodomésticos encendidos ha sido el domingo en la franja horaria de 7:00-8:00. Parece que esa alta actividad de prolonga hasta las 10:00 de ese mismo día y con un comportamiento similar el sábado.

Otra información que ofrece el *dashboard* construido se pueden conocer los patrones que siguen los electrodomésticos cuando están en uso. Estas curvas de carga serán mostradas en función del electrodoméstico elegido con el seleccionador. En cada uno de ellos se pueden observar diferentes rasgos.

El primero mostrado en la lista es la cafetera, cuya curva de carga se presenta a continuación con la Figura 18.

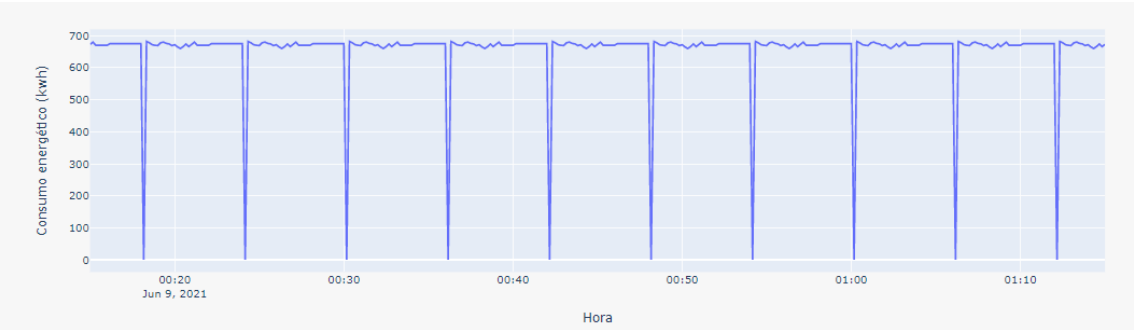

Figura 18. Curva de carga de la cafetera.

Se puede observar cómo el ciclo de uso tiene una duración aproximada de cinco minutos ya que se repite el mismo patrón durante el periodo visualizado. En ese patrón, se identifica una estabilidad alrededor de 650 y 700 kwh hasta que desaparece y vuelve al empezar de nuevo el ciclo.

En cuanto al frigorífico, ya se había mencionado que se mantiene encendido en todo momento, sin embargo, su consumo permanente no supera los 2.67 kwh (tal y como se puede ver en la Figura 19) por lo que no supone un gasto importante en la factura energética.

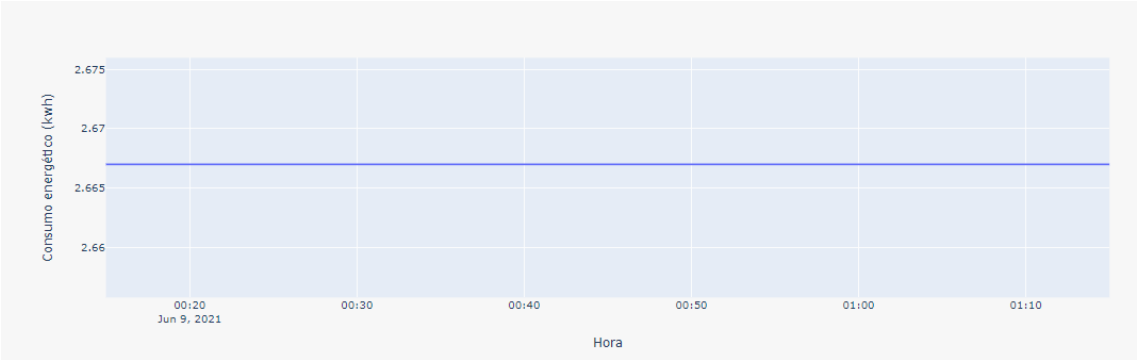

Figura 19. Curva de carga del frigorífico.

El siguiente aparato visualizado será la lavadora. Su uso ha sido medido en un programa corto de ropa delicada y a 40º. La representación de su curva de carga es la siguiente de la Figura 20:

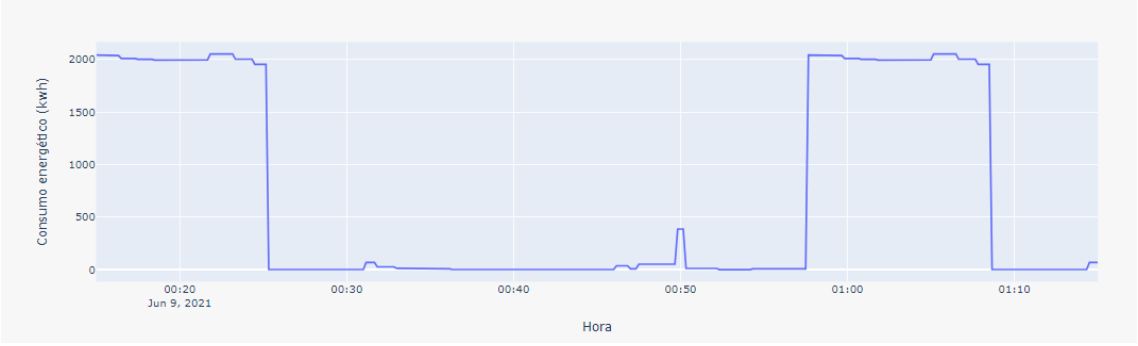

Figura 20. Curva de carga de la lavadora.

En este caso el ciclo es más prolongado y sufre cambios más pronunciados. Comienza con un consumo elevado de alrededor de 2000 kwh durante aproximadamente diez minutos para luego desaparecer casi por completo durante media hora y volver al valor inicial durante otros diez minutos.

Continuando con el estudio de los electrodomésticos con un ciclo de vida largo se encuentra el lavavajillas que puede verse en la Figura 21. El máximo valor que se alcanza vuelve a estar sobre los 2000kwh y en este caso se identifican más variabilidad en el patrón.

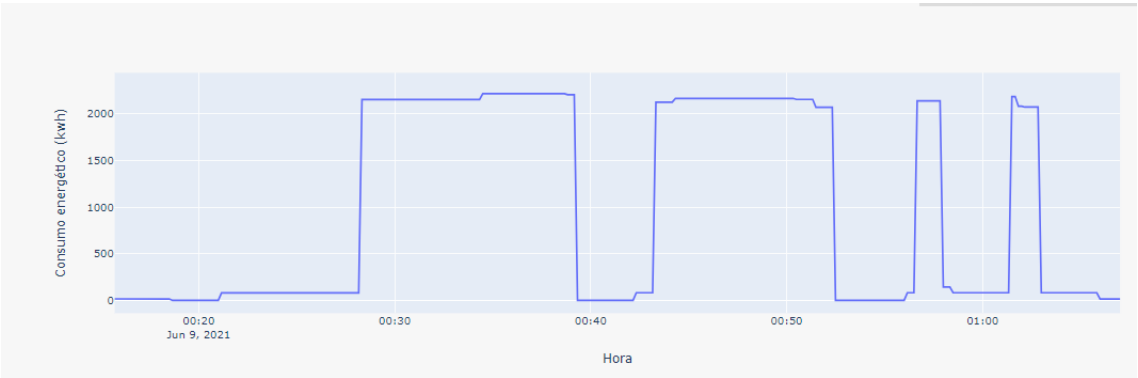

Figura 21. Curva de carga del lavavajillas.

El próximo aparato que se presenta en la Figura 22 será la plancha de ropa. En la figura adjunta a continuación se pueden ver claros picos de consumo, que desde el consumo nulo alcanzan valores aproximados a 2500 kwh. Sin embargo, no se mantienen en el tiempo esos valores elevados y se debe tener en cuenta que en este caso el ciclo de uso del electrodoméstico lo fija el usuario.

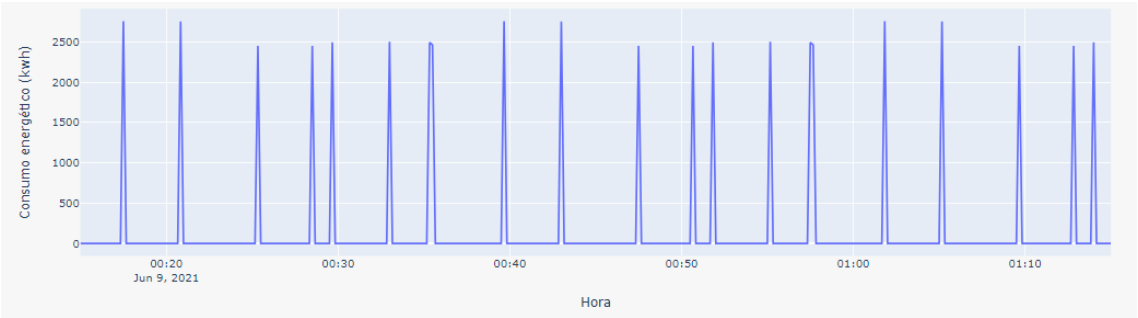

Figura 22. Curva de carga de la plancha.

En lo que corresponde al secador de pelo obtenemos lo siguiente en la Figura 23:

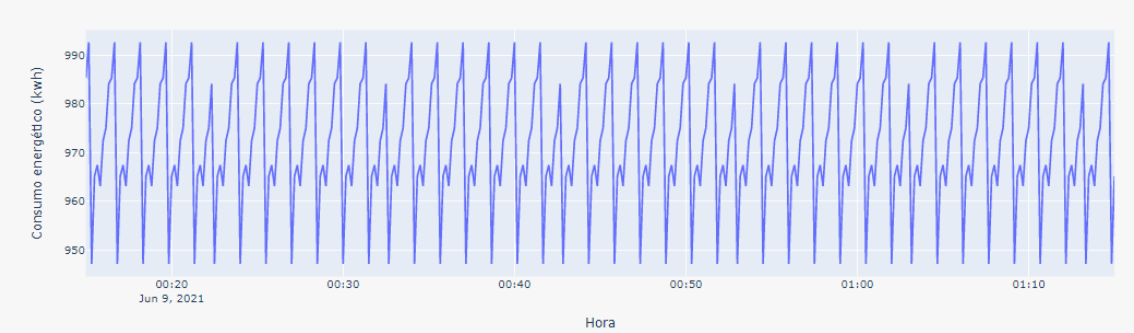

Figura 23. Curva de carga del secador.

Y acotando más el periodo del eje X cono en la Figura 24 se puede apreciar mejor la tendencia:

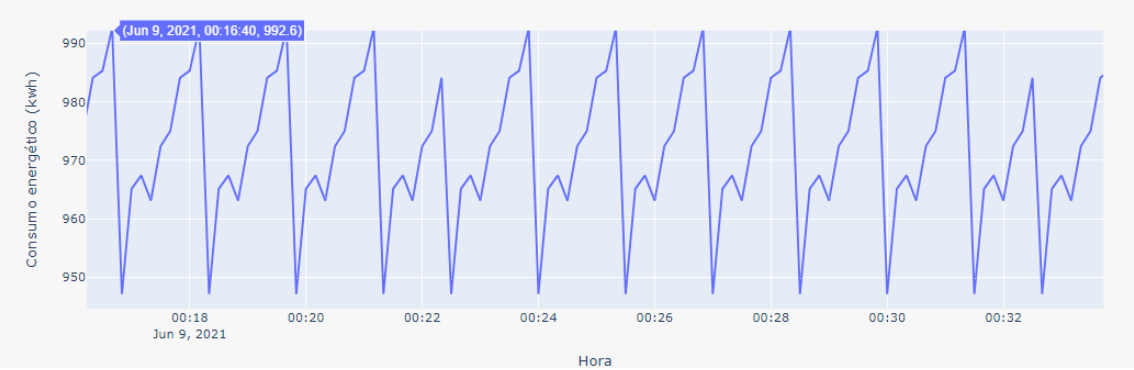

Figura 24. Curva de carga del secador con eje más acotado.

Se ve claramente que cada ciclo dura aproximadamente dos minutos y los valores van oscilando entre 950 y 990 kwh. En este caso también durará tanto como el usuario del aparato lo decida.

Por último, el tostador. Se puede apreciar en la Figura 25 que el ciclo dura aproximadamente tres minutos en los que sufre una caída de consumos desde su valor más estable de unos 850 kwh. La duración la fijará el propio tostador en función del grado de tostado que se le indique.

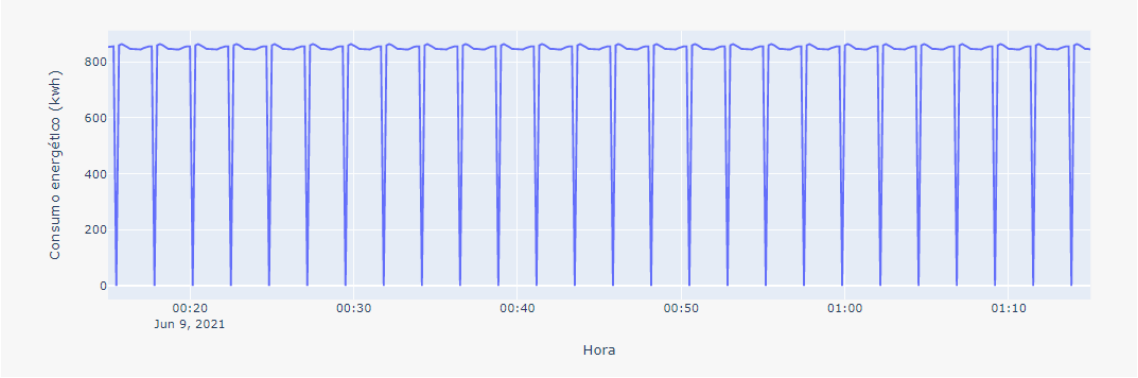

Figura 25. Curva de carga del tostador.

Como conclusión, se podría decir que los electrodomésticos más costosos en cuanto a energía serán el lavavajillas y la lavadora, sin embargo, no se mantienen constantes durante todo su ciclo. En cuanto a los electrodomésticos de menor envergadura, el secador de pelo sería el que más consume y la plancha el que menos.

4.3 ANÁLISIS

Esta última parte del proyecto se ha procedido a realizar un análisis para conocer en mayor profundidad las características asignadas a cada uno de los perfiles definidos en la primera fase. Se ha generado un modelo de *clustering* cuyo propósito era identificar a qué perfil podía pertenecer cada registro de usuario en función de los patrones que fuera capaz de detectar. Para cumplir con esta meta ha sido suficiente con utilizar la colección referente a los consumos totales de la base de datos.

Previo al modelo de Machine Learning implementado, se ha realizado un estudio exploratorio de lo que se encontraba en el conjunto de datos original y se ha podido contrarrestar con lo que se ha visto en la parte de visualización.

En este primer gráfico se puede ver la distribución media de los consumos totales por horas. Cabe recalcar que no representa la misma información que se ha podido ver en la Figura 26 porque en este caso se tiene en cuenta el consumo final y no el número de electrodomésticos encendidos.

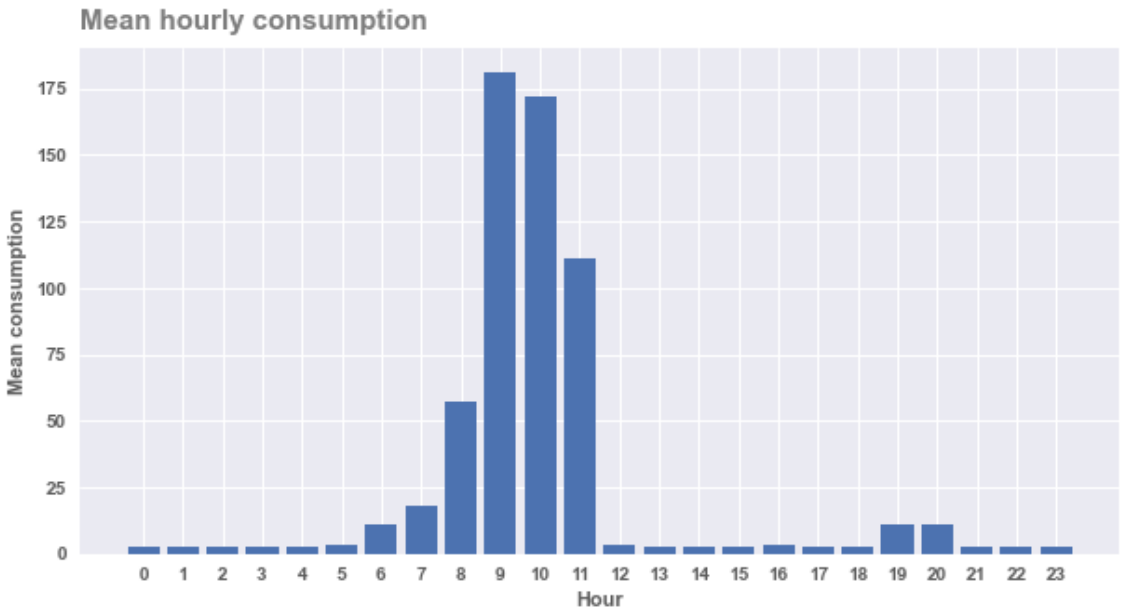

Figura 26. Media de consumo total por horas.

De él podemos concluir que la franja horaria con mayores consumos se produce durante la mañana. Esta afirmación podría deberse porque algunos de los electrodomésticos que se han tenido en cuenta como el tostador, la cafetera o el secador de pelo se suelen utilizar con más frecuencia a primeras horas de la mañana por que la mayoría de la gente desayuna y se ducha antes de empezar su jornada laboral o académica. También se aprecia un aumento entre las 19:00 y 20:00, coincidiendo con el fin de esa jornada en días de labor.

Si por el contrario queremos conocer el día de la semana con los valores más elevados, lo podemos saber con el siguiente gráfico de líneas de la Figura 27.

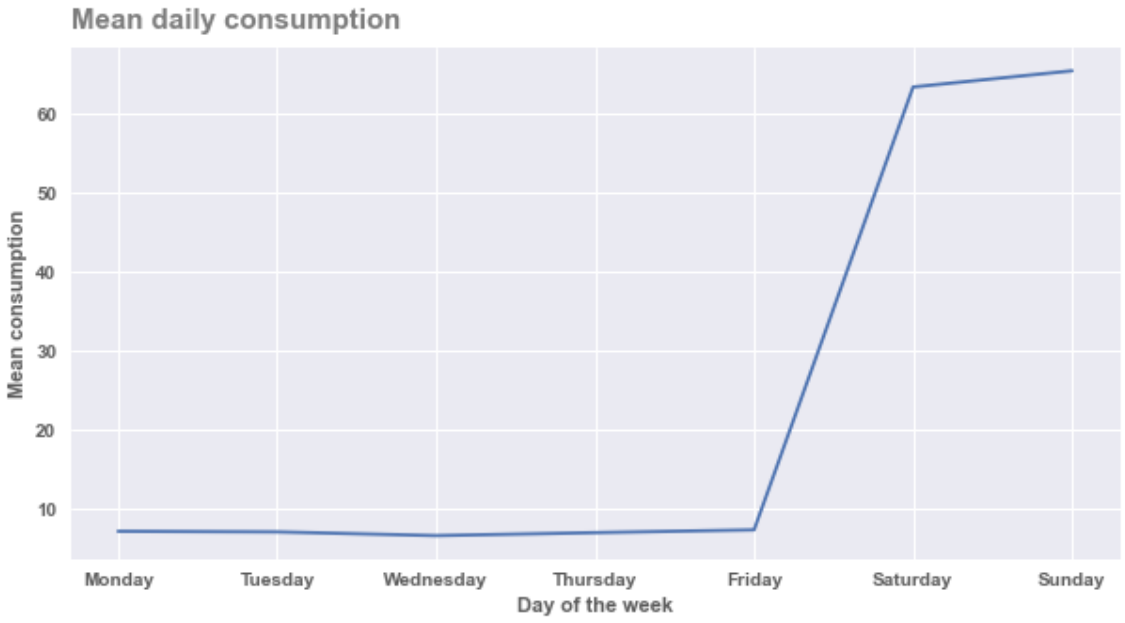

Figura 27. Media de consumo total por día de la semana.

Se ve una clara diferencia entre los días laborables y los fines de semana en los que los usuarios generan consumos. Es probable que esta diferencia tan significativa se deba a la subida de precios reciente que se ha producido en el estado. Los fines de semana serán los días con menor aumento de coste acorde a esta nueva campaña implantada, por lo que muchos usuarios podrían optar por realizar labores domésticas como poner la lavadora, o planchar la ropa esos días.

Una vez construido el modelo con el algoritmo *k means*, cada registro ha sido asignado a un *cluster* de los cuatro generados. Si representamos la información utilizada para ese *clustering* obtenemos lo siguiente:

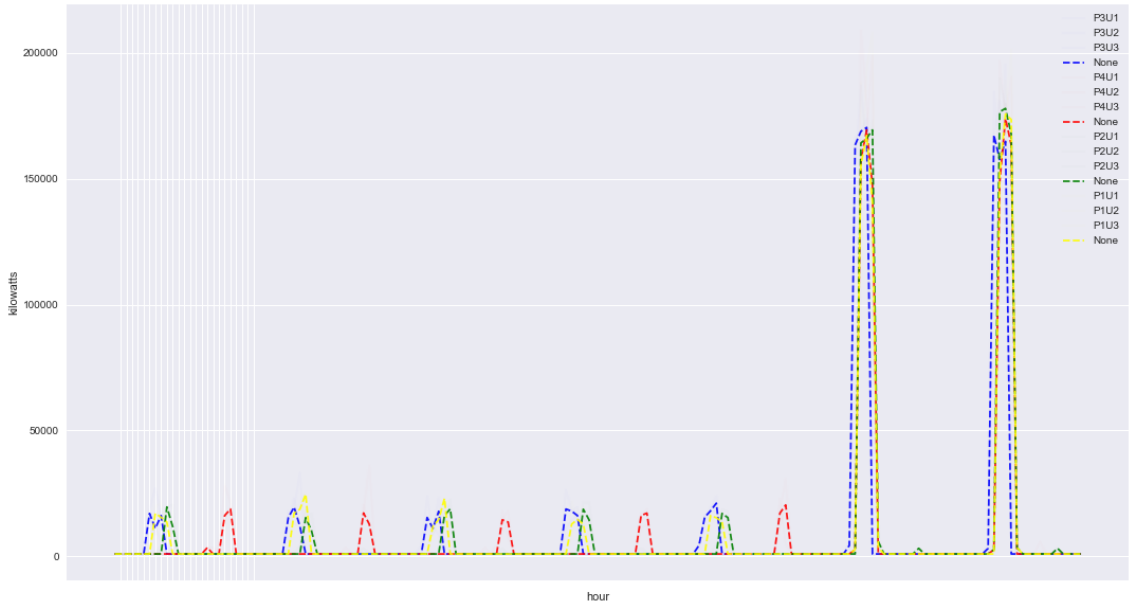

Figura 28. Representación de los clusters generados.

De esta Figura 28 se pueden deducir varias afirmaciones. Por un lado, parece que el modelo ha sido capaz de identificar cada uno de los usuarios con su perfil correspondiente tal y como se indica en el lado derecho superior del gráfico.

Cada *cluster*, que se ha representado en distinto color, sigue patrones diferentes entre ellos, aunque se repitan las diferencias entre los días festivos y los laborables.

Para poder percibir mejor las peculiaridades de cada perfil a lo largo de las horas de un día y asumiendo que los días de labor deben seguir un patrón de consumo parecido entre ellos, se ha filtrado por un día de la semana (jueves) y se han representado por separado los cuatro grupos generados con el mismo color seleccionado en la visualización anterior tal y como se muestran a continuación en la Figura 29:

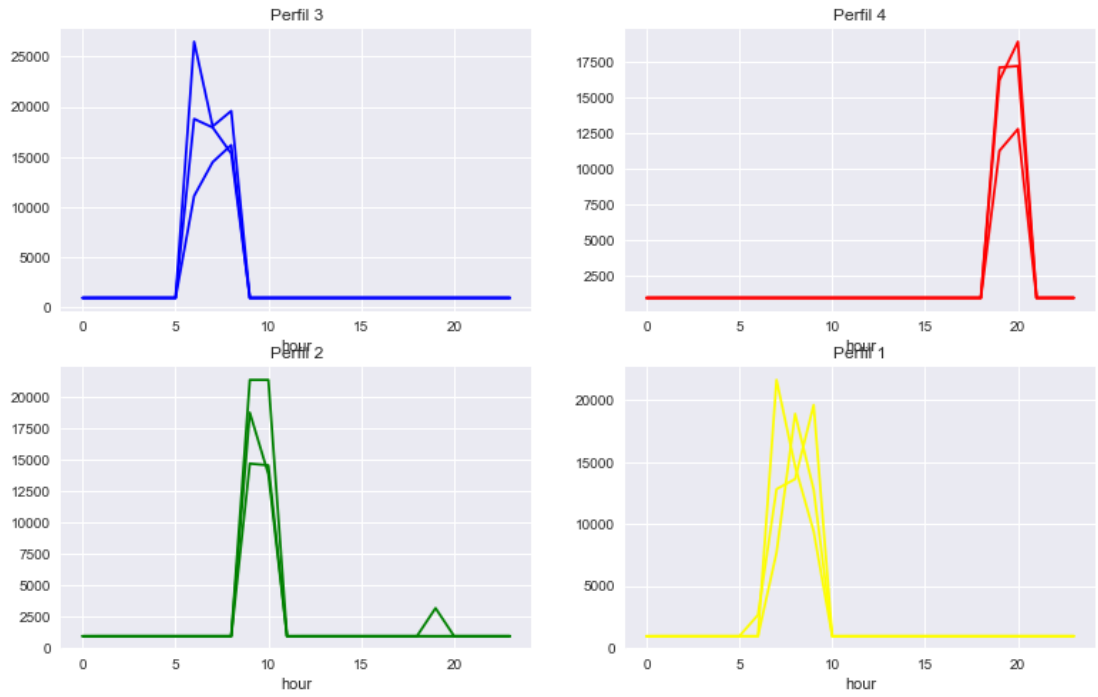

Figura 29. Consumos por horas el jueves por cluster generado.

Esta vez se pueden ver de forma más clara las distinciones que haya podido detectar el modelo a la hora de clasificar cada usuario con su perfil correspondiente.

En el primer gráfico, que pertenece al *cluster* 0 catalogado como perfil 3, parece que los consumos se dan sobre todo entre las 5:00 y las 9:00. Recordando que este perfil representaba a una familia con cuatro ocupantes, parece acertado que sus principales consumos se reproduzcan antes de que empiece su jornada laboral.

En cuanto al segundo grupo, que representa al cuarto perfil descrito anteriormente en esta memoria, los consumos se centran entre las 17:00 y 21:00. Este perfil refleja los hábitos de una persona con horario nocturno, por lo que parece oportuno que sus principales consumos se reproduzcan a esta hora por la misma razón que en el caso anterior, con la diferencia de que en este caso la jornada laboral empezará alrededor de las 22:00.

Continuando con el tercer gráfico en color verde se representan los usuarios pertenecientes al perfil 2, es decir, a aquellas personas retiradas. Si analizamos los horarios de este jueves, se perciben dos picos de consumo. Uno durante la mañana desde 8:00 a 11:00 y otro por la tarde, aunque menos significativa, entre las 18:00 y 20:00. Al tratarse de una persona sin horario fijo por trabajo o estudios se comprenden los consumos a esas horas, y que en otros perfiles teniendo en cuenta que se trata de un día laborable no podrían justificarse tan fácilmente.

Por último, el perfil 1 que refleja el consumo de un piso de estudiantes, podría ser adecuado que la mayoría se los consumos se centren entre las 6:00 y 10:00 de la mañana, antes de que comiencen sus clases.

Con esto se ha podido ver que el modelo ha intuido que el número óptimo de grupos para realizar el *clustering* a este conjunto de datos era de cuatro (k=4), y ha sido capaz de relacionar cada usuario con su respectivo perfil. Por ello, se puede afirmar que el modelo ha sido lo suficientemente bueno para resolver el problema presentado con la base de datos proporcionada con las condiciones anteriormente descritas y asumidas.

Capítulo 5. CONCLUSIONES Y TRABAJOS FUTUROS

5.1 CONCLUSIONES

Una vez desarrollado el modelo completo y unificado, se ha podido llegar a ciertas conclusiones en base a lo obtenido en cada una de las fases. Se debe tener en cuenta que la falta de datos reales ha obligado a generar datos que fueran capaces de ser tratados posteriormente. Por tanto, la falta de registros con los que poder hacer una comparación o en los que basarse en esta primera fase ha complicado la tarea. En un futuro, sería beneficioso disponer de datos reales y generar nuevos datos sintéticos en función de los reales de manera que se mantenga la privacidad de los clientes de la compañía y el resultado se asemeje más a la realidad.

En esta primera parte del pipeline se ha buscado que los datos siguieran unos criterios especificados por perfil y que los registros fueran capaces de ser generados en tiempo real e introducidos en la base de datos. Esto permite que tanto la ingesta como la visualización posterior siga el ritmo de generación por lo que se trate de un modelo en tiempo real. Cuanto mayores latencias existan entre los tiempos de generación de un registro a otro, la visualización mostrará dichos retardos y no representará un escenario real con registros continuos.

Una vez se ha conseguido visualizar los datos registrados tanto en tiempo real como agregados de históricos se han podido hacer las primeras presunciones sobre lo que expresan los datos. Los gráficos referentes a los datos que se estén ingestando en tiempo real nos muestran información sobre el consumo total que se está registrando en esos segundos concretos y el porcentaje que corresponde a cada electrodoméstico de ese consumo total. En cuanto a los datos que se han almacenado en el periodo de tiempo referente a una semana se han podido obtener afirmaciones más generales sobre los hábitos de consumo guardados de los distintos perfiles.

Por último, en la fase final, se han agrupado los registros recibidos de los usuarios en función de los patrones que se han detectado en la base de datos proporcionada como input al modelo. Este agrupamiento permite entender mejor los hábitos de consumo se los usuarios que se han tenido en cuenta y facilitar a la compañía esta información para poder tomas decisiones futuras en base a esa información.

Dicho esto, no cabe duda de que se ha conseguido cumplir con las metas definidas al iniciar este proyecto. El resultado ha sido un proceso completo y en funcionamiento de lo que sería un pipeline de recepción de datos hasta llegar a su análisis final.

5.2 TRABAJOS FUTUROS

Uno de los siguientes pasos en caso de que se dispusiera de más tiempo para el desarrollo del proyecto, sería añadir complejidad y escalabilidad. Hasta el momento se ha trabajado con cuatro perfiles diferentes y tres usuarios por cada perfil en un periodo de una semana. Esta decisión de acotar el número de registros fue tomada por no querer repercutir demasiado en el tiempo de ejecución de los generadores y primar el funcionamiento total del sistema completo. Sin embargo, un escenario real supondría un número mayor de registros por lo que sería interesante continuar haciendo crecer la base de datos para comprobar que el sistema no sufre daños en su ejecución.

Por otra parte, es preciso señalar que todo el despliegue del modelo se ha realizado on premise, es decir, la instalación del programa se ha realizado de manera local. Una modificación futura que podría resultar beneficioso sería su migración a la nube. Para tomar esta decisión y ver si realmente supondría una mejora es necesario analizar las ventajas y desventajas de cada una de las alternativas teniendo en cuenta las características del proyecto en cuestión.

Cada vez es más sonada la instalación de infraestructuras en la nube o en *cloud*, como alternativa cuando se dispone de un volumen muy elevado de datos. La elección depende de

factores como la escalabilidad, el coste, los recursos, el control y la seguridad de la infraestructura.

Con un almacenamiento de datos local una organización debe comprar, desplegar y mantener todo el hardware y el software. Un almacenamiento de datos en la nube, en cambio, no tiene hardware físico. La empresa paga por el espacio de almacenamiento y la potencia de cálculo que necesita en un momento dado. La escalabilidad se consigue añadiendo más recursos en la nube, y no hay necesidad de emplear a personas para desplegar o mantener el sistema porque esas tareas son manejadas por el proveedor [39].

A continuación, se explayan algunas de las ventajas e inconvenientes que esta migración podría suponer y que facilitarán la decisión que se tome en un futuro [40].

Las ventajas principales de un almacén de datos en la nube son la escalabilidad, el coste, la seguridad, la disponibilidad y el tiempo de comercialización.

- Escalabilidad

  La capacidad de almacenaje y computacional no es un problema en estos casos, por lo que los datos pueden fluir sin problemas en los momentos de mayor actividad. Las soluciones suelen ser personalizadas de manera que el usuario solo tiene que pagar por el espacio que se esté utilizando en ese momento.

- Coste

  Con un almacén de datos en la nube, no hay que comprar ni configurar servidores físicos. Las empresas sólo pagan por el almacenamiento y el tiempo de CPU que necesitan. Además, el proveedor de la nube se encarga del mantenimiento, la administración y las actualizaciones.

- Disponibilidad

La disponibilidad y la fiabilidad es otro aspecto a tener en cuenta para tomar la decisión final. La capacidad de replicar los datos en diferentes regiones asegura que sus datos estén altamente disponibles, incluso en caso de fallo.

En cuanto a los aspectos negativos que esta transición podría implicar se incluyen los siguientes [41] :

- Cortes de conexión

Las empresas deben ser conscientes de que esta tecnología, es más propensa a sufrir cortes y otros problemas técnicos por estar conectado a Internet.

- Seguridad

El principal problema de la nube está representado por la seguridad. Las compañías deben ser conscientes de que ponen a disposición de un tercero toda su información sensible. Esto podría suponer un gran riesgo a la empresa, por ellos es de suma importancia la elección del proveedor. Sin embargo, esto dependerá en gran parte del tamaño de la empresa, puesto que si se trata de una pequeña compañía puede incluso beneficiarse de un proveedor con más recursos que él para obtener mayor nivel de seguridad.

- Propenso a ataques

Puede hacer que las empresas sean más vulnerables frente a ataques externos. Es por ello que se ofrece la posibilidad de ocultar los datos sensibles.

- Pérdida de soporte

El servicio de atención al cliente y el soporte recibido por los proveedores es motivo de queja de ciertos usuarios tal y como lo expreso el New York Times: “Si necesita ayuda o si no se siente cómodo tratando de encontrar consejos en foros abiertos, la nube probablemente no sea lo ideal" [39].

Se ha visto que esta migración a la nube puede super o no una mejora dependiendo del caso de uso. En caso de que se decida dar el paso, será fundamental escoger el mejor proveedor de estos servicios que puedan encontrarse actualmente en el mercado. A continuación, se listarán algunas de ellas:

- AWS (Amazon Web Service)

Actualmente podría decirse que es el líder del mercado con una tasa de generación de ingresos al final de 2019 de más de 14 mil millones $, se ha convertido en gran parte de los ingresos de la empresa Amazon [42] .Con un ritmo acelerado de innovación además y una amplío red de servicios es la "opción segura" en este mercado tras ganarse la confianza de sus usuarios.

- Azure (de Microsoft)

Microsoft Azure ocupa el segundo lugar en cuota de mercado [43] .Azure ya es una plataforma muy capaz y amplia, y Microsoft continúa acelerando la velocidad de su nueva función. Microsoft ahora está lanzando sus propias capacidades innovadoras con Azure. Microsoft está centrado en utilizar su capacidad para agrupar Azure con otros productos y servicios de Microsoft.

Estos son los dos principales proveedores de servicio en la nube más utilizados por los usuarios, por lo que se recomendaría que en el caso de que se decidiera migrar en proyecto desarrollado on-premise a la nube, se contemplara utilizar alguna de las dos alternativas expuestas previamente.

Capítulo 6. BIBLIOGRAFÍA

[1] M. H. Amir Gandomi, «Beyond the hype: Big data concepts, methods, and analytics,» 2014.

[2] S. S. Laura Igual, Introduction to Data Science, Springer, 2017.

[3] Iberdrola, «Somos la primera energética privada en Europa y la segunda del mundo por inversión en I+D+i,» [En línea]. Available: https://www.iberdrola.com/innovacion/nuestros-negocios.

[4] Iberdrola, «A la vanguardia en transformación digital,» [En línea]. Available: https://www.iberdrola.com/innovacion/transformacion-digital?utm_source=internal&utm_medium=referral&utm_campaign=contenidoglobal-oct20.

[5] i-de, «CONTADORES INTELIGENTES,» [En línea]. Available: https://www.i-de.es/distribucion-electrica/contadores-inteligentes.

[6] J. L. b. ,. A. A. A. c. ,. M. H. a. Tamás Csoknyai a, «Analysis of energy consumption profiles in residential buildings and impact assessment of a serious game on occupants' behavior,» 2019.

[7] A. Q. ,. J.-L. D. ,. I. D. Ignacio Benítez, «Dynamic clustering segmentation applied to load profiles of energy,» 2013.

[8] O. G. Santin, «Behavioural Patterns and User Profiles related to energy consumption for heating,» 2015.

[9] J. Cambronero, «DATOS SINTÉTICOS CON GANS (I): ¿POR QUÉ DATOS SINTÉTICOS?,» 2020. [En línea]. Available: https://www.datio.com/ai/datos-sinteticos-con-gans-i-por-que-datos-sinteticos/.

[10] C. A. Z. T. D. John Meehan, «Data Ingestion for the Connected World».

[11] Watson Marketing, IBM, «10 Key Marketing Trends fo 2017 and Ideas for Exceeding Customer Expectations».

[12] S. L. C. Y. L. W. L. P. Cun Ji, «IBDP: An Industrial Big Data Ingestion and Analysis Platform and Case Studies».

[13] Apache Kakfa, «APACHE KAFKA,» [En línea]. Available: https://kafka.apache.org/.

[14] R. M. Muñoz, «¿Hay vida más allá de Oracle? Apache Kafka,» 2020. [En línea]. Available: https://dbaoracle4hire.blogspot.com/2020/09/hay-vida-mas-alla-de-oracle-apache-kafka.html.

[15] APACHE KAFKA, [En línea]. Available: https://kafka.apache.org/documentation/#introduction.

[16] M. Cortegana, «Introducción a Apache Kafka,» 2019. [En línea]. Available: https://medium.com/@mcortegana93/introducci%C3%B3n-a-apache-kafka-b67cd7b92ebd.

[17] R. Gour, «Kafka For Beginners,» 2018. [En línea]. Available: https://medium.com/@rinu.gour123/kafka-for-beginners-74ec101bc82d.

[18] Cloudera, «Apache Kafka Overview,» [En línea]. Available: https://docs.cloudera.com/documentation/kafka/1-2-x/topics/kafka.html.

[19] IBM, «What is data storage?,» [En línea]. Available: https://www.ibm.com/topics/data-storage.

[20] N. B. Francisco Morteo, Un enfoque práctico de SQL, 2004.

[21] A. Castro Romero, J. S. González Sanabria y M. Callejas Cuervo, «Utilidad y funcionamiento de las bases de datos NoSQL,» 2012.

[22] Telefonica, «Bases de datos NoSQL».

[23] J. B. Veronika Abramova, «NoSQL Databases: MongoDB vs Cassandra».

[24] sas, «Data Visualization: What it is and why it matters,» [En línea]. Available: https://www.sas.com/en_us/insights/big-data/data-visualization.html.

[25] K. Brush, «data visualization,» [En línea].

[26] F. M, A Brief History of Data Visualization, Springer Handbooks, 2008.

[27] S. E. A. A. G. Tarek Azzam, «Data Visualization and Evaluation,» 2013.

[28] A. Unwin, «Why is Data Visualization,» 2020.

[29] D. Hand, «What is the purpose of statistical modelling?,» 2019.

[30] R. T. F. T. Hastie, The Elements of Statistical Learning,, Springer, 2017.

[31] I. G. a. Y. B. a. A. Courville, Deep Learning, MIT Press, 2016.

[32] M. Kubat, An Introduction to Machine, Springer,, 2017.

[33] I. Sacolick, «What is agile methodology? Modern software development explained,» 2020. [En línea]. Available: https://www.infoworld.com/article/3237508/what-is-agile-methodology-modern-software-development-explained.html.

[34] Plotly, «DASH,» [En línea]. Available: https://plotly.com/dash/.

[35] kafka-python, «KafkaConsumer,» [En línea]. Available: https://kafka-python.readthedocs.io/en/master/apidoc/KafkaConsumer.html.

[36] Mongo Db documentacion, «Pymongo,» [En línea]. Available: https://docs.mongodb.com/drivers/pymongo/.

[37] plotly, «Introducing Dash,» 2017. [En línea]. Available: https://medium.com/plotly/introducing-dash-5ecf7191b503.

[38] D. Castillo, «Develop Data Visualization Interfaces in Python With Dash,» [En línea]. Available: https://realpython.com/python-dash/.

[39] A. Dhiman, «Analysis of On-premise to Cloud Computing Migration».

[40] Stich, «On-premises vs. cloud data warehouses: a comparison,» [En línea]. Available: https://www.stitchdata.com/resources/compare-on-premises-and-cloud-data-warehouse/.

[41] A. Dhiman, «Analysis of On-premise to Cloud Computing Migration,» 2015.

[42] S. Fadilpašić, «AWS now makes up over half of all Amazon revenue,» 2019. [En línea]. Available: https://www.itproportal.com/news/aws-now-makes-up-over-half-of-all-amazon-revenue/.

[43] E. Jones, «Cuota de mercado de la nube – una mirada al ecosistema de la nube en 2021,» 2021. [En línea]. Available: https://kinsta.com/es/blog/cuota-de-mercado-de-la-nube/.

ANEXO A

```
from kafka import KafkaConsumer
from pymongo import MongoClient
from json import loads


consumer = KafkaConsumer(
    'topic1','topic2','topic3',
    bootstrap_servers=['localhost:9092'],
    auto_offset_reset='latest',
    enable_auto_commit=True,
    group_id='my-group',
    value_deserializer=lambda x: loads(x.decode('utf-8')))

client = MongoClient('localhost:27017')
db=client.prueba


doc = {}

for msg in consumer:

    topic = msg.topic

    message=msg.value
    timestamp= msg.timestamp
    message['timestamp'] = timestamp


    if topic == "topic1":
        collection = db.tabla1
        collection.insert(message)
        print('{} added to {}'.format(message, collection))

    elif topic == "topic2":
        collection = db.tabla2
        collection.insert(message)
        print('{} added to {}'.format(message, collection))

    else :
        collection = db.tabla3
        collection.insert(message)

        print('{} added to {}'.format(message, collection))
```