

UNIVERSIDAD DE VALLADOLID

MÁSTER UNIVERSITARIO

Ingeniería Informática



TRABAJO FIN DE MÁSTER

**Determinación de los estados de funcionamiento y
predicción del consumo energético del edificio
LUCIA**

Realizado por

Enrique García Miravalles

Tutores

José Belarmino Pulido Junquera

Carlos Javier Alonso González



Universidad de Valladolid

Junio de 2022

Agradecimientos

A mis tutores, los profesores D. José Belarmino Pulido Junquera y D. Carlos Javier Alonso González, por la confianza que han depositado en mí y toda la ayuda que me han brindado a lo largo del desarrollo de este Trabajo Fin de Máster.

A Sergio Lorenzo González por permitirnos acceder al edificio LUCIA y emplear sus datos para la realización del presente proyecto, además de toda la información que nos ha aportado para mejorar nuestra comprensión sobre el funcionamiento del edificio.

A mi familia y pareja por todo el apoyo recibido y que tanto he necesitado para poder terminar este proyecto y especialmente a mis padres por enseñarme que sin esfuerzo y sacrificio no hay recompensa.

Resumen

El auge del concepto de desarrollo sostenible ha puesto el foco en la reducción de consumo energético como método para paliar los gastos de energía innecesarios. Esta nueva tendencia hacia el desarrollo y consumo sostenible ha propiciado la creación de numerosos *Smart Buildings* buscando conseguir importantes ahorros energéticos en uno de los sectores con mayor demanda de energía como es el de la construcción.

A pesar del nacimiento de estos nuevos edificios inteligentes es necesario disponer de herramientas y sistemas que faciliten el estudio y la gestión de los recursos energéticos. En este trabajo se presenta una propuesta para la predicción del consumo energético en *Smart Buildings* utilizando para ello datos del edificio LUCIA, un *Smart Building* de la Universidad de Valladolid.

Nuestra propuesta implica la identificación de los estados de funcionamiento del edificio mediante la aplicación de técnicas de aprendizaje automático no supervisado en una primera instancia. Posteriormente aplicando técnicas de *deep learning* se generan modelos para predecir el consumo energético para cada estado de funcionamiento. Finalmente se comparan los resultados de los modelos generados para cada estado con los resultados obtenidos con un modelo general para todos los estados.

El trabajo producto de este proyecto que hace referencia a la identificación de los estados de funcionamiento y predicción del consumo energético del edificio LUCIA queda integrado en una aplicación web también implementada durante el desarrollo de este Trabajo Fin de Máster. La aplicación web se ha empleado como demostrador de todo el proceso llevado a cabo y se expone también en el presente documento.

Palabras clave

Edificios inteligentes, estados de funcionamiento, aprendizaje automático, aprendizaje profundo, agrupamiento, series temporales, predicción de consumo energético, redes LSTM, modelos.

Abstract

The rise of the concept of sustainable development has put the focus on reducing energy consumption as a method to mitigate energy waste. This new trend towards sustainable development and consumption has led to the creation of numerous Smart Buildings looking to achieve significant energy savings in one of the sectors with the highest energy demand such as construction industry.

Despite the construction of these new Smart Buildings, it is necessary to have tools and systems to make easy the study and management of energy resources. This paper presents a proposal for the prediction of energy consumption in Smart Buildings using data from the LUCIA Smart Building of the University of Valladolid.

Our proposal first identifies the operation modes of the building by applying unsupervised machine learning techniques. Then, predictive models are generated using deep learning techniques to forecast the energy consumption in each operation mode of the building. Finally, the results of the models generated for each state are compared with the results obtained with a general model for all states.

The work related to the identification of the operation modes and the prediction of the energy consumption of the LUCIA Smart Building is integrated in a web application also implemented during the development of this project. The web application has been used as a demonstrator of the whole process and is also presented in this document.

Keywords

Smart buildings, operation modes, machine learning, deep learning, clustering, time series, energy consumption forecasting, LSTM networks, models.

Índice general

1. Introducción	9
1.1 Contexto	10
1.2 Objetivos	10
1.3 Estructura de la memoria	12
2. Metodología y planificación	13
2.1 Metodología de trabajo	13
2.1.1 Adaptación e implantación de la metodología	15
2.2 Planificación	16
2.3 Costes del proyecto	23
3. Entorno tecnológico	25
4. Contexto del proyecto: predicción de carga energética en <i>Smart Buildings</i>	29
4.1 Caso de Estudio: LUCIA un <i>Smart Building</i> en la UVA	29
4.1.1 Sistema de monitorización del edificio	31
4.1.2 Proceso de extracción de datos	33
4.2 Estado del arte en predicción de carga energética en <i>Smart Buildings</i>	34
4.2.1 Métodos de <i>clustering</i>	36
4.2.2 Predicción de series temporales con redes neuronales.....	37
5. Modelo de datos	45
5.1 Exploración del <i>raw data</i>	45
5.1.1 Clasificación de variables y sistemas	48
5.1.2 Datos de consumo energético del edificio	50
5.1.3 Datos para la determinación de los estados de funcionamiento del edificio.....	51
5.2 Limpieza de datos	57
5.3 Transformación de datos y extracción de variables.....	58
5.3.1 Codificación del tiempo	61
5.3.2 Normalización y estandarización de los datos	62
5.4 Selección de características	62
6. <i>Clustering</i>	65
6.1 Métricas de evaluación	65
6.2 Modelos y resultados	66
7. Predicción del consumo energético	73
7.1 Método de predicción de consumo energético: redes LSTM	73
7.2 Conjuntos de entrenamiento y test.....	76
7.3 Entrenamiento y parametrización de los modelos	79
7.4 Métricas de evaluación	82
7.5 Modelos de predicción a corto plazo	83
7.6 Modelos de predicción a medio plazo	84
7.7 Modelos de predicción a largo plazo	86

7.8	Discusión de los resultados	88
8.	Aplicación desarrollada	91
8.1	Definición y alcance	91
8.2	Análisis	92
8.2.1	Requisitos	92
8.2.2	Casos de uso	93
8.3	Diseño	104
8.3.1	Arquitectura <i>cliente-servidor</i>	104
8.3.2	Arquitectura física	105
8.3.3	Arquitectura lógica	106
8.3.4	Diseño de base de datos	107
8.3.5	Diagrama de secuencia de una petición	109
8.4	Implementación: tecnologías y estructura de componentes de aplicación	110
8.5	Despliegue de la aplicación en un entorno de producción	113
9.	Conclusiones y líneas futuras	115
	Referencias bibliográficas	117
	Anexo I - Manual de instalación de la aplicación web	122
	Anexo II - Manual de usuario de la aplicación web	124

Índice de figuras

Figura 1: Arquitectura y diseño del edificio LUCIA	29
Figura 2: Estructura de una red FFN	38
Figura 3: Convolución de 1 dimensión	39
Figura 4: Función de activación ReLU	39
Figura 5: Función max pooling	40
Figura 6: Red neuronal recurrente y su despliegue en el tiempo	40
Figura 7: Estructura uno a uno	41
Figura 8: Estructura uno a muchos.....	41
Figura 9: Estructura muchos a uno.....	42
Figura 10: Estructura muchos a muchos	42
Figura 11: Detalle de la estructura de una celda de red LSMT	43
Figura 12: Serie de datos del analizador de red 39 en el tiempo.....	49
Figura 13: Serie de datos del analizador de red 29 en el tiempo.....	51
Figura 14: Temperatura de retorno del fancoil de la primera planta para la producción de frío	56
Figura 15: Temperatura de retorno del fancoil de la primera planta para la producción de calor	56
Figura 16: Porcentaje de humedad relativa del aire exterior.....	57
Figura 17: Temperatura del aire exterior.....	57
Figura 18: Consumo energético horario del edificio LUCIA + CPD sin procesar	58
Figura 19: Consumo energético horario de las nuevas variables generadas (febrero 2021)	60
Figura 20: Codificación usando one-hot de los días de la semana.....	61
Figura 21: Matriz de correlación con las variables finales.....	64
Figura 22: Dendrograma resultante al aplicar clustering jerárquico sobre los datos del área de producción.....	67
Figura 23: Grupos generados usando k-medias y 3 clústeres para las variables del área de producción.....	68
Figura 24: Etiquetado de las observaciones del área de producción usando k-medias y 3 clústeres.....	69
Figura 25: Dendrograma resultante al aplicar clustering jerárquico sobre las variables de condiciones externas	70
Figura 26: Grupos generados usando k-medias y 2 clústeres para las variables de condiciones externas.....	70
Figura 27: Etiquetado de las observaciones de condiciones externas usando k-medias y 2 clústeres.....	71
Figura 28: Etiquetado de las observaciones del área de producción (k=3) y de condiciones externas (k=2).....	71
Figura 29: Arquitectura del modelo de capas LSTM apiladas.....	74

Figura 30: Fechas elegidas para los datos del modelo de clúster reciente	77
Figura 31: Conjunto de datos de entrenamiento del modelo “general”	77
Figura 32: Conjunto de datos de entrenamiento del modelo “clúster área de producción”	78
Figura 33: Conjunto de datos de entrenamiento del modelo “clúster condiciones externas”	78
Figura 34: División del conjunto de datos usados para el modelo de “clúster reciente” en entrenamiento y test	79
Figura 35: Valor de la función de pérdida para los datos de entrenamiento y validación	82
Figura 36: Predicción a 1 hora usando el mejor modelo "clúster área de producción"	83
Figura 37: Predicción a 1 hora usando el mejor modelo "clúster condiciones externas"	83
Figura 38: Predicción a 1 hora usando el mejor modelo "clúster reciente"	84
Figura 39: Predicción a 1 hora usando el mejor modelo "general"	84
Figura 40: Predicción de las siguientes 48 horas al 28/02/2021 usando el mejor modelo “clúster área de producción”	85
Figura 41: Predicción de las siguientes 48 horas al 05/03/2021 usando el mejor modelo “clúster área de producción”	85
Figura 42: Predicción de las siguientes 48 horas al 28/02/2021 usando el mejor modelo “clúster condiciones externas”	85
Figura 43: Predicción de las siguientes 48 horas al 05/03/2021 usando el mejor modelo “clúster condiciones externas”	85
Figura 44: Predicción de las siguientes 48 horas al 28/02/2021 usando el mejor modelo “clúster reciente”	86
Figura 45: Predicción de las siguientes 48 horas al 05/03/2021 usando el mejor modelo “clúster reciente”	86
Figura 46: Predicción de las siguientes 48 horas al 28/02/2021 usando el mejor modelo “general”	86
Figura 47: Predicción de las siguientes 48 horas al 05/03/2021 usando el mejor modelo “general”	86
Figura 48: Predicción a 168 horas usando el mejor modelo “clúster área de producción”	87
Figura 49: Predicción a 168 horas usando el mejor modelo “clúster condiciones externas”	87
Figura 50: Predicción a 168 horas usando el mejor modelo “clúster reciente”	87
Figura 51: Predicción a 168 horas usando el mejor modelo “general”	87
Figura 52: Diagrama de casos de uso	94
Figura 53: Arquitectura cliente-servidor	105
Figura 54: Arquitectura física	106
Figura 55: Arquitectura lógica	107
Figura 56: Diseño de base de datos	108
Figura 57: Diagrama de secuencia genérico	109
Figura 58: Endpoints del servicio API	111

Índice de tablas

Tabla 1: Sprints del proyecto	17
Tabla 2: Composición del visualizador de planta (sistemas principales).....	32
Tabla 3: Estructura de los datos para cada sensor monitorizado.....	46
Tabla 4: Estadísticas de los analizadores de red 29 y 39	51
Tabla 5: Variables de temperatura del área de producción	54
Tabla 6: Estadísticas de las variables de temperatura del área de producción.....	56
Tabla 7: Estadísticas de las variables de condiciones externas	56
Tabla 8: Estadísticas de las nuevas variables generadas tras el proceso de transformación	60
Tabla 9: Variables eliminadas mediante la selección de características con la matriz de correlación de Pearson	64
Tabla 10: Rendimiento de los modelos de clustering sobre los datos del área de producción	67
Tabla 11: Rendimiento de los modelos de clustering sobre las variables de condiciones externas	69
Tabla 12: Resultados de los modelos para la predicción a 1 hora.....	83
Tabla 13: Resultados de los modelos para la predicción a 48 horas	85
Tabla 14: Resultados de los modelos para la predicción a 168 horas	87
Tabla 15: Caso de uso 1 - Mostrar datos mensuales	95
Tabla 16: Caso de uso 2 - Cargar nuevo mes	95
Tabla 17: Caso de uso 3 - Eliminar mes.....	96
Tabla 18: Caso de uso 4 - Eliminar datos mensuales	97
Tabla 19: Caso de uso 5 - Ver gráfico de datos	97
Tabla 20: Caso de uso 6 - Ver datos totales	97
Tabla 21: Caso de uso 7 - Eliminar variables.....	98
Tabla 22: Caso de uso 8 - Limpiar datos.....	98
Tabla 23: Caso de uso 9 - Agrupar datos	99
Tabla 24: Caso de uso 10 - Ver matriz de correlación	99
Tabla 25: Caso de uso 11 - Generar dataset	100
Tabla 26: Caso de uso 12 - Exportar datos.....	100
Tabla 27: Caso de uso 13 - Mostrar datasets generados	101
Tabla 28: Caso de uso 14 - Visualizar contenido del dataset.....	101
Tabla 29: Caso de uso 15 - Eliminar dataset.....	102
Tabla 30: Caso de uso 16 - Generar modelo	103
Tabla 31: Caso de uso 17 - Mostrar listado de ejecuciones guardadas	103
Tabla 32: Caso de uso 18 - Visualizar detalle de ejecución.....	103
Tabla 33: Caso de uso 19 - Eliminar ejecución.....	104

Capítulo 1

1. Introducción

La sostenibilidad[1] es un concepto cada vez más arraigado en la sociedad actual y se refiere a la explotación de los recursos por debajo de su límite de renovación sin comprometer la posibilidad de explotación de los mismos recursos por parte de generaciones futuras. Los objetivos del desarrollo sostenible buscan mejorar las condiciones de vida de las personas y aminorar las consecuencias causadas por el hombre vinculadas al cambio climático.

Las nuevas políticas que fomentan el desarrollo sostenible y los avances tecnológicos tanto en inmótica (automatización de inmuebles), como domótica (automatización inteligente de la vivienda) han favorecido la construcción de numerosos *Smart Buildings* o edificios inteligentes. Gran parte de estos edificios buscan lograr el distintivo nZEB[2] (nearly zero-energy building), requiriendo una eficiencia energética muy alta con una demanda de energía muy baja (cercana a cero) que es cubierta en gran medida gracias a energías renovables. La denominación nZEB está claramente ligada al concepto de sostenibilidad. Otro factor que ha favorecido la construcción de estos edificios son las iniciativas por parte de los gobiernos de proyectos de *Smart Cities* o ciudades inteligentes para cumplir con las exigencias marcadas en acuerdos como el de París[3], buscando ciudades completamente sostenibles siendo los *Smart Buildings* una pieza fundamental para la consecución de la distinción de *smart city*.

Los edificios inteligentes cuentan con instalaciones y sistemas que permiten una gestión y control integrada y automatizada con el fin de aumentar la eficiencia energética, la seguridad, la usabilidad y la accesibilidad en sus instalaciones, en definitiva, permiten un control completo sobre el edificio con el objetivo de mejorar la habitabilidad y reducir los costes de su uso. Entre los sistemas que gestionan estos edificios inteligentes podemos encontrar sistemas de climatización, iluminación, seguridad, telecomunicaciones, etc., pudiendo automatizarse todos y cada uno de ellos para conseguir los propósitos de edificio inteligente.

Debido al énfasis de estos edificios en la eficiencia energética y sostenibilidad y a la existencia de pérdidas energéticas evitables debido a condiciones de operación ineficientes[4], aparece la necesidad de contar con herramientas y aplicaciones que permitan el estudio y

análisis de su completo funcionamiento. Los avances en los dispositivos IoT y las capacidades de procesamiento de las computadoras han permitido que cada vez más edificios estén equipados con sensores/medidores que generan grandes cantidades de datos todos los días. Este hecho presenta una oportunidad única para poder aplicar herramientas de aprendizaje automático (*machine learning* en inglés) a fin de identificar patrones y generar modelos predictivos. Estos patrones y modelos tienen como finalidad aumentar el rendimiento de los sistemas encargados de gestionar los edificios y su autonomía, además de ofrecer información de valor añadido a responsables acerca de su funcionamiento.

1.1 Contexto

El edificio LUCIA (Lanzadera Universitaria de Centros de Investigación Aplicada) de la Universidad de Valladolid, situado en el campus Miguel Delibes, es un *Smart Building* que cuenta con diferentes dispositivos destinados a aumentar su eficiencia energética, razón por la que es considerado un edificio sostenible bajo la denominación de edificio nZEB.

A pesar de que el edificio LUCIA cuenta con un sistema de gestión y control de las instalaciones y sistemas totalmente parametrizado y eficaz, necesita de programación por parte de usuarios administradores, para determinar los diferentes modos de funcionamiento en base a consignas para las diferentes variables monitorizadas. Aunque los *Smart Building* se diseñan pensando en que factores externos tengan el menor impacto posible sobre ellos, estas consignas tienen una dependencia directa con variables externas como puede ser el clima o los periodos de ocupación del edificio, lo que hace intuir que el edificio tendrá diferentes estados de funcionamiento.

La motivación del presente TFM presentado por el grupo de investigación GSI (Grupo de Sistemas Inteligentes) con el que se ha realizado la estancia de la asignatura “I+D+i en Informática” del Máster en Ingeniería Informática y que posteriormente se ha concluido de forma individual, surge, debido a nuestra suposición acerca de la existencia de estados de funcionamiento y a la existencia de la posibilidad de mejorar la autonomía y prestaciones del edificio basándose en la información de los datos obtenidos de la monitorización del edificio.

1.2 Objetivos

Con la realización de este trabajo se pueden distinguir dos tipos de objetivos dependiendo de su procedencia. Por una parte, objetivos y motivaciones personales de la realización de este proyecto y, por otra parte, determinar los estados de funcionamiento y predecir el consumo

energético del edificio LUCIA, utilizando para ello una aplicación desarrollada durante el propio TFM.

Los objetivos personales, que a la vez se presentan como parte de lo que pretendo conseguir con la realización de este TFM, son demostrar los conocimientos y competencias adquiridos durante el transcurso de la docencia del Máster en Ingeniería Informática de la Universidad de Valladolid, así como demostrar mi capacidad de análisis de soluciones en temas de la misma casuística que el caso práctico a realizar. Por otra parte, es una motivación mejorar y adquirir nuevos conocimientos acerca del uso de técnicas de análisis, exploración, limpieza y transformación de datos, así como de aprendizaje supervisado y semi-supervisado entre otros aspectos.

En cuanto a lo referido al desarrollo del TFM, se pueden identificar los siguientes objetivos con sus respectivos objetivos secundarios:

1. Estudio y comprensión del funcionamiento del edificio LUCIA y los datos que se pueden obtener del sistema de monitorización del edificio.
 - 1.1. Comprensión de los diferentes sistemas y subsistemas implicados en la eficiencia energética, los modos de funcionamiento y activación de estos sistemas y el uso de consignas para el control del estado de este edificio inteligente.
 - 1.2. Comprensión y análisis del sistema de monitorización del edificio, de las variables registradas en este sistema y de los datos que pueden obtenerse.
2. Búsqueda de los conjuntos de datos sobre los que aplicar los modelos de agrupamiento o *clustering* para la estimación de los estados de funcionamiento y los modelos para la predicción de consumo energético.
 - 2.1. Aplicación de técnicas de análisis, exploración, limpieza y transformación de datos.
3. Construcción de modelos de *clustering* y regresión utilizando técnicas de aprendizaje automático y selección de los modelos óptimos tras la evaluación de los resultados obtenidos.
4. Desarrollo de una aplicación web *ad hoc* aportando al usuario un entorno de trabajo que integre las funcionalidades necesarias para poder realizar los trabajos de análisis, exploración, limpieza y transformación de datos del edificio LUCIA, junto con la posibilidad de ejecutar diferentes modelos para su análisis y estudio. Con esta aplicación se pretende construir un demostrador para validar la consecución de los objetivos 2 y 3.

1.3 Estructura de la memoria

La presente memoria de Trabajo Fin de Máster está dividida en nueve capítulos donde se detallan los procesos llevados a cabo para poder completar los objetivos detallados anteriormente.

En el **capítulo 1**, en el cual nos encontramos actualmente, se presenta la temática del proyecto y el carácter motivacional para su desarrollo. El **capítulo 2** se centra en la metodología y planificación seguidas para el desarrollo del proyecto y las adaptaciones realizadas durante el proceso. En el **capítulo 3** se presenta el entorno tecnológico empleado para llevar a cabo los trabajos.

En el **capítulo 4** se presenta el edificio LUCIA, caso de estudio del presente trabajo, junto con las técnicas más desarrolladas en el estudio de predicción y estimación de energía en edificios inteligentes.

En el **capítulo 5** se presenta el modelo de datos disponible junto con las técnicas de preprocesado y transformación de datos para conseguir que se puedan emplear tanto en modelos de *clustering*, los cuales se presentan en el **capítulo 6**, como en modelos para la predicción de energía, los cuales se presentan en el **capítulo 7**.

En el **capítulo 8** se expone la aplicación web desarrollada integrando todas las funcionalidades de los capítulos anteriores, indicando el alcance funcional de la aplicación, la arquitectura y el proceso de implementación.

Por último, en el **capítulo 9** se exponen las conclusiones y los posibles trabajos futuros.

Capítulo 2

2. Metodología y planificación

El desarrollo del proyecto ha conllevado la elección de una metodología concreta y una planificación previa en base a los objetivos existentes y las pautas marcadas por la metodología escogida. En esta sección se explica la metodología elegida y las diferentes configuraciones aplicadas, además del plan de proyecto establecido.

2.1 Metodología de trabajo

Desde un principio el desarrollo de este Trabajo Fin de Máster ha tenido una orientación plenamente centrada en la investigación acerca de la estimación de los estados de funcionamiento y de la predicción de consumo energético del edificio LUCIA. Tras ver cómo los trabajos del proceso de investigación se podían aplicar al desarrollo de una herramienta de software capaz de servir como demostrador de los avances realizados, el proyecto ha pasado a tener un pequeño porcentaje de trabajo asociado al desarrollo de software con un producto final definido.

Para el desarrollo del proyecto se ha optado por el uso de una metodología ágil, siendo *Scrum*[\[5\]](#) la metodología elegida. A pesar de ser *Scrum* una metodología originalmente enfocada en el desarrollo de software cada vez está siendo más utilizada en equipos de investigación[\[6\]](#). Los proyectos de investigación generalmente son predecibles, no así sus resultados, que te pueden llevar a cambiar las hipótesis y los experimentos. Por este motivo la flexibilidad que presenta *Scrum* para adaptarse a la naturaleza de sus procesos y gracias a algunas de las herramientas que definen la propia metodología, ésta se amolda perfectamente a sus necesidades.

Scrum centra sus bases en el trabajo colaborativo y cuenta con una serie de tareas que deben realizarse regularmente y una serie de roles bien definidos con unas habilidades y responsabilidades que en conjunto determinan la propia metodología. Además, *Scrum* es una

metodología ideal para proyectos donde prima la obtención de resultados a corto plazo. Esto se debe a la posibilidad de comprobar el avance existente de forma periódica, además de poder determinar nuevas tareas que en un principio no estarían definidas por la existencia de incertidumbre acerca del alcance del proyecto, o por las posibles vías de consecución de los objetivos inicialmente definidos. De esta forma, *Scrum* permite una mejor adaptación del trabajo a las necesidades del proyecto en cada momento de su ciclo de vida.

En la metodología *Scrum* podemos encontrar una serie de roles clave para el cumplimiento de los requisitos de la metodología:

- *Product owner*: es el propietario del producto y entre las funcionalidades que tiene son las de establecer los objetivos o el trabajo que se requiere realizar siendo el nexo entre las partes interesadas (*stakeholders*) y el personal de desarrollo.
- Personal de desarrollo: suele estar formado por un número de entre 3 y 9 personas y se encarga del desarrollo del producto o tareas asignadas. El objetivo del equipo de desarrollo es obtener el incremento demandado en cada *sprint*, que se definirá justo a continuación, y siguiendo las pautas y procedimientos marcados en *Scrum*. Un *sprint* es un período temporal de entre 1 y 4 semanas durante el cual se genera un incremento en el proyecto o producto final.
- *Scrum master*: este rol es desempeñado por una persona del equipo o personal de desarrollo y se encarga de velar por la consecución de los objetivos del *sprint* y el cumplimiento de los procedimientos marcados por la metodología *Scrum*.

Además de los roles la metodología *Scrum* tiene una serie de actividades o tareas que se realizan de forma periódica en intervalos con duraciones de entre 1 y 4 semanas. Estos intervalos de tiempo se denominan *sprints* y cada uno de ellos finaliza con un incremento en el desarrollo del proyecto. En cada uno de los *sprints* de un proyecto siguiendo la metodología *Scrum*, podemos encontrar las siguientes actividades:

- *Sprint planning*: se trata de una reunión al principio de cada *sprint* donde se fijan las tareas del *sprint* a realizar. Estas tareas se incluyen en el *sprint backlog* y se generan a partir de las historias de usuario (HU) del *product backlog*. En el *product backlog* se pueden encontrar todas las diferentes funcionalidades o requisitos definidos para el proyecto hasta la fecha, ordenados por prioridad de realización ya sea por importancia o por dependencias entre ellos. En el *sprint planning* están presentes tanto el personal de desarrollo como el *product owner*.
- *Daily scrum*: reunión diaria de breve duración donde se exponen las contribuciones de cada miembro en el desarrollo del *sprint*, los trabajos que se van a realizar durante el

propio día y si se han encontrado impedimentos o dificultades de cara a la finalización de alguna tarea o historia de usuario.

- *Sprint review*: se realiza al final de cada sprint una vez se haya concluido el incremento planificado. Estas reuniones se destinan al análisis del trabajo realizado junto con los *stakeholders*. El objetivo es determinar si se debería realizar algún cambio o si el avance cumple con las expectativas.
- *Sprint retrospective*: tiene como finalidad analizar el proceso de trabajo e identificar “qué ha ido bien” y “qué no ha ido bien” en el sprint. En estas reuniones el equipo de *Scrum* puede auto-examinarse para determinar posibles formas de mantener aquello que ha sido productivo durante el desarrollo del sprint y aportar ideas para lograr mejorar en determinadas áreas. Estas reuniones tienen la finalidad de lograr un proceso de mejora continuo, que también es otro enfoque del desarrollo ágil.
- *Refinement*: se trata de una reunión que se ha incluido para aclarar conceptos, tareas o HU que no quedaron completamente especificadas durante el *sprint planning* o sobre las que han surgido dudas en el equipo de desarrollo.

2.1.1 Adaptación e implantación de la metodología

La metodología que aplicaremos no se ajustará totalmente a las pautas y procedimientos marcados por *Scrum* debido a las condiciones de trabajo, las dimensiones del proyecto o el número de componentes presentes en el proceso de desarrollo. Por estos motivos, a continuación, se mostrarán algunas de las adaptaciones que se han realizado para conseguir implantar la metodología siguiendo en la medida de lo posible los estándares de *Scrum*.

El rol del equipo de desarrollo en el presente trabajo va a estar compuesto por un único integrante, el presente autor/alumno del proyecto. El hecho de que el equipo de desarrollo sólo cuente con un integrante supone que el mismo integrante también asume el rol de *Scrum master* buscando el cumplimiento de los procedimientos de la metodología. Esto mismo ha supuesto que las *daily meetings* hayan sido un trabajo de observación y análisis autónomo, determinando qué avances se han logrado, qué problemas se han encontrado y qué tareas se van a realizar cada día trabajo.

Por otra parte, el rol de *product owner* estará cubierto por los tutores del presente TFM, quienes determinarán el alcance del proyecto y producto final, además de los requisitos que deben cubrirse.

El curso de máster realizado ha sido en modalidad online, motivo por el cual todas las reuniones realizadas a lo largo del desarrollo del proyecto se han efectuado mediante herramientas de videoconferencia. A pesar de ello, no ha supuesto ningún contratiempo para el desarrollo del proyecto e implantación de la metodología *Scrum* con las adaptaciones comentadas.

2.2 Planificación

El proyecto final ha sido conformado por un total de 9 *sprints* de una duración comprendida entre 3 y 4 semanas a excepción de algunos casos puntuales en los que de por medio ha coincidido un periodo vacacional. Dentro del ciclo de vida del proyecto también podemos encontrar un periodo de parón entre los meses de agosto y noviembre de 2021.

Las actividades de *sprint review* y *sprint planning* se han realizado en la misma reunión, a excepción de la primera *planning* y última *review* del proyecto. Esta planificación de reuniones se ha realizado con el objetivo de poder aprovechar el tiempo con mayor eficacia y generalmente, siempre que ha sido posible, estas reuniones se han intentado realizar el primer lunes de cada mes.

Para la estimación de esfuerzo se han utilizado puntos de historia o *story points* (SP) como una unidad de medida de esfuerzo relativo, sin necesidad de indicar un valor temporal y como una forma más fácil de indicar el volumen de trabajo de cada tarea. En concreto la asignación de los puntos de historia a cada tarea se ha basado en la técnica del “tamaño relativo”. El uso de esta técnica implica que una tarea que se considere el doble de compleja que otra, recibe el doble de puntos de historia. Esta técnica de estimación de esfuerzo también permite tener en cuenta los posibles riesgos de desarrollo de una tarea a la hora de asignar los puntos de historia. El uso de este método agiliza el proceso de estimación frente al uso de unidades de tiempo al permitir reducir el esfuerzo invertido en cada valoración. Además, aporta mayor flexibilidad ante posibles contratiempos, riesgos, complejidades o incluso la incertidumbre existente a la hora de realizar una tarea.

En la siguiente tabla se muestran los *sprints* realizados junto con una breve descripción, la duración que han tenido y los puntos de historia que se han asignado a cada sprint:

<i>Sprint</i>	Descripción y ámbito temporal de desarrollo	Puntos de historia
1º	El comienzo de este <i>sprint</i> se corresponde con el comienzo del proyecto. La fecha de comienzo de este <i>sprint</i> ha sido el 10 de mayo de 2021, mientras que la duración del <i>sprint</i> ha sido de 4 semanas finalizando el 07 de junio de 2021.	10 puntos de historia

2°	La duración del segundo <i>sprint</i> ha sido de 4 semanas, comenzando el 07 de junio de 2021 y finalizando el 05 de julio del mismo año.	14 puntos de historia
3°	Su duración ha sido de 3 semanas, comprendidas entre el 05 de julio de 2021 y el 26 de julio de 2021 que tuvo lugar la <i>sprint review</i> . Se trata del último <i>sprint</i> previo al periodo de estacionamiento del proyecto.	10 puntos de historia
4°	El 02 de noviembre de 2021 da comienzo el cuarto <i>sprint</i> del proyecto. ¹ El <i>sprint</i> tiene una duración de 4 semanas siendo la <i>sprint review</i> el día 29 del mismo mes.	12 puntos de historia
5°	El quinto <i>sprint</i> , con una duración fuera de lo normal debido a los periodos vacacionales del mes de diciembre, realmente ha tenido tan solo 3 semanas de trabajo. El comienzo del <i>sprint</i> es el 29 de noviembre de 2021 y el fin del <i>sprint</i> el 10 de enero de 2022.	15 puntos de historia
6°	Con una duración de 4 semanas el sexto <i>sprint</i> ha comenzado el día 10 de enero y finalizado el día 07 de febrero de 2022 con la <i>sprint review</i> .	10 puntos de historia
7°	El séptimo <i>sprint</i> del proyecto ha tenido una duración de 4 semanas comprendidas entre los días 07 de febrero y 07 de marzo de 2022.	16 puntos de historia
8°	El penúltimo <i>sprint</i> del proyecto ha tenido una duración de 4 semanas desde el 07 de marzo al 04 de abril de 2022 donde se ha desarrollado la aplicación web como demostrador de los trabajos previamente realizados.	20 puntos de historia
9°	El último <i>sprint</i> con menor carga de trabajo y donde se han completado algunas tareas técnicas que habían salido del <i>feedback</i> obtenido de anteriores <i>sprints</i> y finalizado la memoria, ha tenido una duración de 3 semanas desde el 04 de abril de 2022 al 25 de abril de 2022.	10 puntos de historia

Tabla 1: Sprints del proyecto

A continuación, se van a mostrar las historias de usuario escogidas del *product backlog* para realizar en cada *sprint* junto con las diferentes tareas definidas para cada historia de usuario.

¹ El proyecto ha estado parado durante un periodo de 3 meses coincidiendo con las vacaciones de verano y octubre de 2021.

Las tareas realizadas en un sprint y los puntos de esfuerzo que tienen asignadas conforman el *sprint backlog*.

Las historias de usuario se presentan con la denominación [HU-X], siendo “X” el número de historia de usuario, mientras que las tareas se presentan con el formato [T-X.Y], siendo “X” el identificador de la historia de usuario a partir de la cual se ha definido la tarea e “Y” el número de tarea de la historia en cuestión.

Primer sprint

Es el punto de partida y la primera toma de contacto con el proyecto. Para este primer sprint se ha fijado la consecución de las siguientes historias de usuario:

- [HU-1] Contextualizar el caso de estudio.
- [HU-2] Documentar primer *sprint*.

Las tareas definidas, junto con los puntos de historia asignados, son las siguientes:

- [T-1.1] Estudio de la edificación inteligente (*Smart Buildings* y *Smart Cities*) (2 SP).
- [T-1.2] Estudio del sistema HVAC (estructura y diseño) (1 SP).
- [T-1.3] Estudio del edificio LUCIA I (estructura y diseño) (2 SP).
- [T-1.4] Estudio del edificio LUCIA II (sistemas implicados en maximizar la eficiencia energética) (3 SP).
- [T-2.1] Documentación del caso de estudio y la metodología elegida (2 SP).

Segundo sprint

Con el desarrollo del segundo sprint se han cubierto las siguientes historias de usuario:

- [HU-3] Estado del arte en predicción energética en edificios.
- [HU-4] Método y herramientas necesarias para la obtención de los datos del edificio LUCIA.
- [HU-5] Obtener datos del edificio LUCIA I.
- [HU-6] Documentar segundo *sprint*.

Las tareas definidas, junto con los puntos de historia asignados, son las siguientes:

- [T-3.1] Búsqueda, estudio y análisis de trabajos orientados a la predicción energética en edificios y al uso de modelos destinados a la determinación de modos de funcionamiento de edificios inteligentes (3 SP).
- [T-3.1] Estudio y comprensión del software DESIGO INSIGHT y su funcionalidad (1 SP).
- [T-4.2] Acceso al panel de control del DESIGO y estudio del método de almacenamiento de los datos del edificio (1 SP).
- [T-5.1] Creación de un repositorio para el almacenamiento de los datos del edificio LUCIA (1 SP).
- [T-5.2] Selección del formato de almacenamiento de los datos al obtenerlos del software DESIGO INSIGHT (1 SP).
- [T-5.2] Obtención y almacenamiento de los datos históricos del funcionamiento del edificio LUCIA (5 SP).
- [T-6.1] Documentación del estado del arte y técnicas y herramientas de obtención de datos del edificio LUCIA (2 SP).

Tercer *sprint*

Historias de usuario completadas durante el tercer sprint:

- [HU-7] Obtener datos del edificio LUCIA II.
- [HU-8] Obtener información de alto nivel acerca del *raw data* del edificio LUCIA.

Las tareas definidas, junto con los puntos de historia asignados, son las siguientes:

- [T-7.1] Finalizar el proceso de obtención y almacenamiento de los datos históricos del funcionamiento del edificio LUCIA (5 SP).
- [T-8.1] Carga y estudio preliminar del *raw data* del edificio LUCIA (5 SP).

Cuarto *sprint*

Tras el parón de dos meses que ha sufrido el proyecto, durante el cuarto sprint se han completado las siguientes historias de usuario:

- [HU-9] Agrupar los datos totales en formato de serie temporal.

- [HU-10] Obtener información completa del raw data.
- [HU-11] Documentar cuarto *sprint*.

Las tareas definidas, junto con los puntos de historia asignados, son las siguientes:

- [T-9.1] Unificar los datos mensuales de cada variable en formato de serie temporal (3 SP).
- [T-10.1] Descripción de las variables, registros y clasificación por sistemas (1 SP).
- [T-10.2] Identificación de los tipos de datos (1 SP).
- [T-10.3] Obtención de estadísticas de las diferentes variables (1 SP).
- [T-10.4] Identificación de valores ausentes y *outliers* (2 SP).
- [T-10.5] Visualización de distribuciones y tendencias (2 SP).
- [T-11.1] Documentación del proceso de exploración del *raw data* (2 SP).

Quinto *sprint*

En el quinto sprint se ha generado el modelo de datos final y se ha comenzado con el estudio de técnicas de aprendizaje no supervisado:

- [HU-12] Obtener conjunto de datos final.
- [HU-13] Analizar métodos y métricas para la aplicación de algoritmos de aprendizaje no supervisado.
- [HU-14] Documentar quinto *sprint*.

Las tareas definidas, junto con los puntos de historia asignados, son las siguientes:

- [T-12.1] Limpieza de datos (3 SP).
- [T-12.2] Transformación de datos y extracción de nuevas variables (3 SP).
- [T-12-3] Selección de *features* (2 SP).
- [T-13.1] Estudio de los algoritmos de *clustering* (3 SP).
- [T-13.1] Estudio de métricas para la selección del modelo óptimo (2 SP).
- [T-14.1] Documentación del proceso de generación del conjunto final de datos y las técnicas y métricas de aprendizaje no supervisado (2 SP).

Sexto *sprint*

Una vez realizado el preprocesamiento de los datos y obtenido un conjunto de datos final de calidad, pasamos a la fase de extracción de conocimiento con el objetivo de determinar los posibles estados de funcionamiento del edificio LUCIA:

- [HU-15] Determinar los estados de operación del edificio LUCIA.
- [HU-16] Documentar sexto *sprint*.

Las tareas definidas, junto con los puntos de historia asignados, son las siguientes:

- [T-15.1] Selección y configuración de los modelos de *clustering* (2 SP).
- [T-15.2] Implementación de los modelos de *clustering* (5 SP).
- [T-15.3] Evaluación de los modelos de *clustering* (2 SP).
- [T-16.1] Documentación de los resultados de los métodos de *clustering* aplicados para determinar los estados de funcionamiento del edificio LUCIA (1 SP).

Séptimo *Sprint*

Una vez identificados los estados de operación del edificio LUCIA, el objetivo es implementar los modelos para predecir el consumo energético del edificio:

Se han identificado las siguientes HU:

- [HU-17] Analizar métodos y métricas para la aplicación de métodos de predicción de series temporales.
- [HU-18] Implementar modelos para la predicción del consumo energético del edificio LUCIA aplicando redes LSTM.
- [HU-19] Evaluar los modelos de predicción de consumo energético.
- [HU-20] Documentar séptimo *sprint*.

Las tareas definidas, junto con los puntos de historia asignados, son las siguientes:

- [T-17.1] Estudio de redes LSTM, tipos de predicción y configuraciones (2 SP).
- [T-17.2] Estudio de métricas para la evaluación de modelos de predicción de series temporales (1 SP).

- [T-18.1] Implementación de los modelos para la predicción de consumo energético (8 SP).
- [T-19.1] Evaluación y análisis de los resultados obtenidos con los modelos de predicción de consumo energético (2 SP).
- [T-20.1] Documentación del proceso de construcción de los modelos y de los resultados obtenidos (3 SP).

Octavo Sprint

Con el objetivo de demostrar los trabajos realizados se va a implementar una aplicación que integre las funcionalidades anteriormente desarrolladas al cubrir las diferentes historias de usuario.

Se han identificado las siguientes HU:

- [HU-21] Herramienta para la carga, exploración, limpieza y transformación del *raw data* del edificio LUCIA y generación de un conjunto de datos final.
- [HU-22] Interfaz para la implementación y evaluación de modelos de *clustering* y predicción de consumo energético.

Las tareas definidas, junto con los puntos de historia asignados, son las siguientes:

- [T-21.1] Implementación de una interfaz para la carga y visualización de los datos (descripción de variables, tipos de datos, gráficas de datos y tendencias...) (3 SP).
- [T-21.2] Implementación de una interfaz para aplicar las diferentes técnicas de limpieza y transformación de datos aplicadas en *sprints* anteriores (5 SP).
- [T-22.1] Implementación de una interfaz para la selección y configuración del modelo a implementar (2 SP).
- [T-22.2] Implementación de una interfaz para la visualización de resultados, métricas de los modelos y selección de los parámetros óptimos (5 SP).
- [T-22.3] Implementación de una interfaz para la visualización de los resultados finales de los modelos generados (5 SP).

Noveno Sprint

Último *sprint* del proyecto dedicado a completar la memoria final y realizar algunas tareas del *feedback* obtenido en anteriores *sprints*. Se han identificado las siguientes HU:

- [HU-23] *Feedback* de *sprints* anteriores.
- [HU-24] Finalizar memoria del TFM.

Las tareas definidas, junto con los puntos de historia asignados, son las siguientes:

- [T-23.1] Desarrollo de tareas técnicas del *feedback* de *sprints* anteriores (5 SP).
- [T-24.1] Documentación final de la memoria del TFM (5 SP).

2.3 Costes del proyecto

Podemos diferenciar entre dos tipos de costes: los costes materiales y los costes de personal que son necesarios para el desarrollo del proyecto.

El coste total de los recursos materiales utilizados para el desarrollo del proyecto, contando los meses de conexión a internet estando el proyecto activo y el coste de los equipos empleados, queda desglosado de la siguiente forma:

- Coste del portátil: 1.300 €.
- Coste de internet: 270 € (9 meses x 30 €/mes).
- Coste de máquina virtual: 0 € (Costes asumidos por la Escuela de Ingeniería Informática de Valladolid).

Para poder incurrir los costes de personal es necesario tener un número de horas aproximado, aunque hallamos trabajado con puntos de historia. Estimamos una dedicación de 2 horas diarias los días hábiles de la duración del proyecto, desde mayo de 2021 hasta abril de 2022 incluido (teniendo en cuenta el periodo de parón del proyecto), para obtener el número de horas totales empleadas. Se han contabilizado 179 días hábiles a lo largo de la duración del proyecto, lo cual implica que el número de horas empleado asciende a 358. Teniendo en cuenta que el salario promedio estimado de un analista-programador en España es de 30.000 € al año o 15,38 € por hora en 2022, los costes de personal ascienden a la cifra de 5.506,04 €.

Coste total estimado para la realización del proyecto: 7.076,04 €.

Capítulo 3

3. Entorno tecnológico

Para el desarrollo del TFM, han sido necesarias numerosas tecnologías que nos permiten poder desarrollar o completar cada una de las tareas a realizar y que conforman el proyecto completo. Algunas de estas tecnologías por sí solas no ofrecen toda la funcionalidad requerida para el desarrollo del proyecto, por lo que en algunos casos se ha requerido de la inclusión de funcionalidades complementarias mediante el uso de bibliotecas o extensiones. A continuación, se van a describir cada una de las tecnologías usadas y las bibliotecas de Python que se han requerido para incluir nuevas funcionalidades:

- ❖ **Python:** lenguaje de programación usado para la realización de la parte lógica del proyecto. Python es un lenguaje de programación interpretado que aporta diferentes soluciones y herramientas para un desarrollo rápido de aplicaciones en gran variedad de plataformas. En concreto se ha utilizado la versión 3.9.8[7].
 - **Numpy:** biblioteca de Python que proporciona soporte para el uso de arrays multidimensionales (tensores), además de incluir numerosas funciones matemáticas. Esta biblioteca además mejora el rendimiento del código de Python frente al uso de grandes cantidades de datos en forma de arrays.
 - **Pandas**[8]: es una biblioteca de Python que está escrita como una extensión de la biblioteca Numpy que nos ofrece herramientas de manipulación de datos de alto nivel. Pandas maneja principalmente tres tipos de datos:
 - **Series:** se tratan de listas de una dimensión, que pueden crearse como un diccionario, es decir, con una clave y un valor asociado a dicha clave.
 - **DataFrame:** son tablas de dos dimensiones y al igual que las series pueden ser del tipo clave-valor. Además, en los *dataframe*, una de las particularidades que tienen, es que podemos especificar tanto el nombre de las filas (*index*), como el de las columnas (*columns*).

- *Panels*: son tablas de 3 dimensiones, con características similares al *DataFrame*.

De los tres tipos de datos expuestos anteriormente, el más destacado y el que podríamos denominar estructura clave de Pandas, es el *DataFrame*. Un *DataFrame*, podemos definirlo como una colección ordenada de columnas que pueden tener un identificador. La estructura de un *DataFrame*, podríamos decir que se asemeja a una tabla de una base de datos relacional, en las cuales tenemos un nombre bien definido para cada columna y los datos que representan dicha columna. En un *DataFrame* al igual que en las tablas de las bases de datos, cada columna puede albergar diferentes tipos de datos.

Además de ofrecernos la posibilidad de crear *DataFrames*, contiene numerosas opciones para el trabajo con datos como puede ser la creación de series, comparación de elementos, filtrado de estos o incluso la lectura de datos en diferentes formatos como puede ser Excel, CSV, JSON...

- **Matplotlib**: biblioteca de Python que permite hacer representaciones gráficas de conjuntos de datos. En concreto permite realizar visualizaciones en 2D, con la posibilidad de incluir *widgets* para el control de la visualización, consiguiendo representaciones dinámicas e interactivas.
- **Scikit-learn**[\[9\]](#): biblioteca de Python expresamente creada para aprendizaje automático. Ofrece soporte tanto para el uso de diversas herramientas creadas para realizar análisis estadísticos básicos, correlaciones, PCA (Principal Component Analysis), etc., como para aplicar algoritmos de regresión, clasificación o *clustering* sobre un conjunto de datos cualesquiera.

En concreto se ha utilizado esta biblioteca para la implementación de los modelos de *clustering* y determinación de los estados de funcionamiento del edificio LUCIA.

- **Keras**[\[10\]](#): biblioteca de código abierto para *deep learning* escrita en Python, que es capaz de ejecutarse sobre el *framework* TensorFlow, como es el caso del actual proyecto, además de otros como Toolkit o Microsoft Cognitive.

En concreto Keras actúa como una API de TensorFlow abstrayendo sus funcionalidades. De esta forma Keras permite crear modelos de redes neuronales simplificando el proceso frente a la utilización de forma directa de los *frameworks* nombrados anteriormente. Gracias a Keras se han podido implementar las redes neuronales recurrentes para la predicción de consumo energético.

- ❖ **MySQL:** sistema gestor de bases de datos relacional, donde los elementos que forman parte de la relación, como cada subconjunto del producto cartesiano de n conjuntos, se pueden representar como tuplas. Si abstraemos la idea de relación a una tabla con el gestor de datos MySQL, tenemos que cada fila de la tabla se representa como una tupla, mientras que cada columna de una tabla son los campos de ésta. En el caso del desarrollo del TFM actual, el servidor MySQL usado se ha instalado sobre una máquina virtual de la Escuela de Ingeniería Informática de la Universidad de Valladolid.
- ❖ **Jupyter notebook:** entorno computacional basado en web y de código abierto para la ejecución de código a través del navegador web, aceptando diferentes lenguajes de programación, entre los que se encuentra Python, utilizado en el presente proyecto. Jupyter Notebook permite crear y visualizar archivos de código y los resultados de sus ejecuciones en tiempo de ejecución, motivo por el cual sus usos principalmente se extienden al trabajo con datos, ejecución de operaciones matemáticas y procesos de limpieza transformación o modelado sobre los propios datos.

Jupyter notebook se ha utilizado principalmente a la hora de implementar los diferentes modelos para la predicción energética en el edificio LUCIA.
- ❖ **FastAPI**[\[11\]](#): *framework* web de código abierto de alto rendimiento para el desarrollo de APIs con Python. Destaca por una estructura simple y permite el desarrollo completo de un servicio API REST (*Representational State Transfer*) como el que usa la aplicación web que se ha implementado. FastAPI ha sido el *framework* elegido por el rendimiento que presenta respecto de otros *frameworks* más pesados, además de estar basado en Python, necesario para poder implementar todas las funcionalidades que requieren de algoritmos de aprendizaje automático.
- ❖ **Angular**[\[12\]](#): se trata de un *framework* de código abierto para aplicaciones web basado en TypeScript, aunque también se puede trabajar con JavaScript puro. Angular utiliza el patrón modelo vista controlador (MVC) separando la parte lógica de la interfaz en la aplicación. Además, es un *framework* modular y basado en componentes lo cual aporta gran escalabilidad y facilidad para el control y mantenimiento de los diferentes módulos. Aunque el principal lenguaje de programación que usa Angular es TypeScript, en los componentes también encontramos los archivos de lenguaje de marcas (HTML) para el desarrollo de las interfaces y las hojas de estilos (CSS) para el diseño. Para el desarrollo de la aplicación web se ha utilizado la versión 10.2.0.
- ❖ **Node.js:** entorno en tiempo de ejecución para la capa de servidor basado en JavaScript, que utiliza el motor V8 de Google (Chrome V8). Node.js es imprescindible para el desarrollo de la parte *frontend* con Angular, puesto que permite generar un servidor

local para el despliegue de la aplicación localmente, facilitando de esta forma el proceso de desarrollo.

El gestor de paquetes de Node.js, npm, ha sido necesario para la instalación de dependencias y paquetes requeridos para poder finalizar la aplicación usando Angular.

- ❖ **GitLab:** se trata de un servicio web para control de versiones y *DevOps* basado en Git. GitLab es principalmente de código abierto, sólo algunas funcionalidades requieren de licencia.

En el presente proyecto GitLab se ha utilizado como repositorio de código para la gestión de las diferentes versiones y cambios, permitiendo trabajar con diferentes ramas o *branches* para desarrollar las diferentes funcionalidades de la aplicación. Además de servir como repositorio de código, se han utilizado las herramientas que pone GitLab a disposición del usuario para realizar integración, entrega y despliegue continuo[13] (CI/CD *continuous integration/continuous delivery/continuous deployment*), que han permitido automatizar el despliegue de la aplicación desarrollada, aprovechando también GitLab Pages[14] que permite alojar webs estáticas en una instancia de GitLab y bajo un subdominio propio de la marca.

- ❖ **Máquina virtual:** contenedor de software aislado que puede albergar un sistema operativo y aplicaciones, siendo de esta forma capaz de emular el comportamiento de un computador real. La máquina virtual usada, ha desempeñado la funcionalidad de servidor para el servicio API REST para la aplicación desarrollada. Además, como se ha indicado antes también se encuentra instalado el servidor de bases de datos utilizado para la persistencia de datos.

Capítulo 4

4. Contexto del proyecto: predicción de carga energética en *Smart Buildings*

En este capítulo se va a presentar el *Smart Building* LUCIA, caso de estudio del presente Trabajo Fin de Máster. También se revisarán las técnicas que mejores resultados han dado en materia de predicción energética y estimación de modos de funcionamiento en edificios inteligentes.

4.1 Caso de Estudio: LUCIA un *Smart Building* en la UVA

El edificio LUCIA emplazado en el campus Miguel Delibes, es un *Smart Building* que ha conseguido la denominación de edificio nZEB mediante el uso, en gran medida, de energías renovables o que generan muy pocas emisiones para desempeñar la actividad normal del edificio. Además, dispone de una estructura perfectamente diseñada con el objetivo de maximizar la eficiencia energética de las instalaciones.

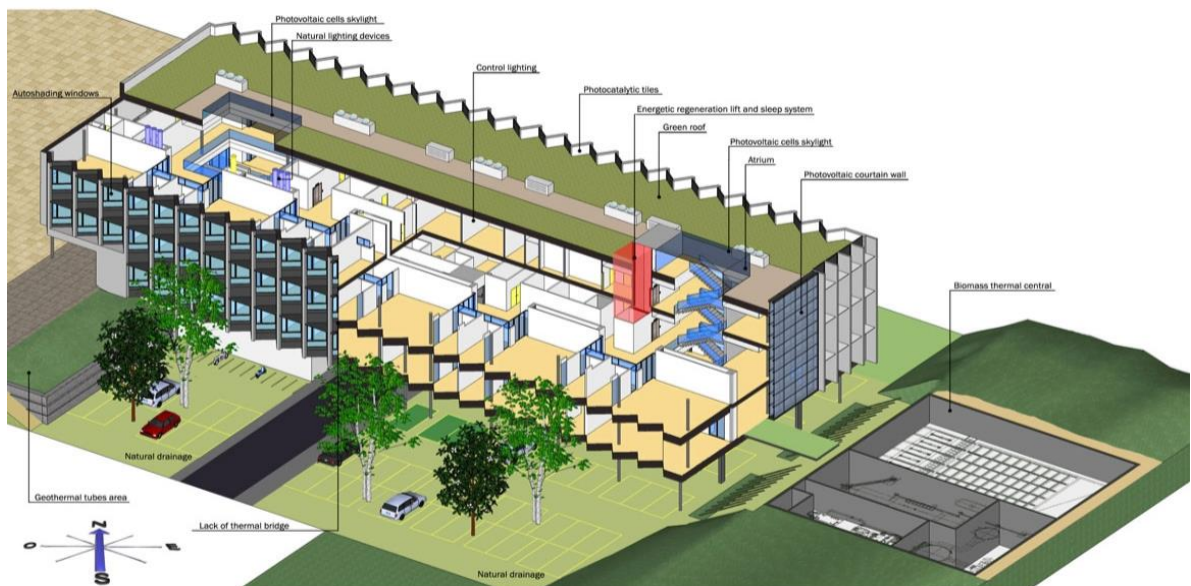


Figura 1: Arquitectura y diseño del edificio LUCIA [15]

El edificio, como podemos ver en la figura 1, tiene un diseño atípico en algunos aspectos, como puede ser las fachadas en “zig-zag”, el aparcamiento abierto en la parte baja, el uso de múltiples paneles fotovoltaicos... Siempre en busca de mejorar su eficiencia energética.

- Las fachadas con forma dentada o en “zig-zag”, tienen una orientación hacia el sur y este, propiciando ganancias térmicas con el mayor aprovechamiento de la luz en invierno y un efecto auto sombreado en verano, reduciendo la necesidad de sistemas de climatización y producción de frío durante el periodo estival.
- El aparcamiento abierto permite optar por un modelo de ventilación natural, además de poder aprovechar la iluminación solar. Además, esta zona cuenta con una cubierta verde buscando crear un microclima con temperaturas más suaves en las inmediaciones del edificio LUCIA.
- Otro aspecto que permite una reducción de consumo energético es el uso de pozos de luz y lucernarios. El uso de estos mecanismos mejora el uso de la iluminación natural reduciendo casi en la mitad la energía necesaria para la iluminación del edificio.
- Por último, hay que destacar los coeficientes de aislamiento térmico de las fachadas y superficies del edificio, que presentan valores muy buenos, limitando pérdidas por transmisión y la consiguiente reducción de demanda.

Entre los sistemas implantados en el edificio ajenos a su diseño podemos encontrar sistemas como EAHX (*Earth-to-Air Heat Exchanger*), UTA (unidad de tratamiento de aire), climatización, producción de calor y frío o paneles fotovoltaicos entre otros.

Un sistema EAHX o pozo canadiense utiliza las condiciones de temperatura existentes en la tierra o terreno para suavizar las temperaturas del aire del exterior previamente a su entrada en los sistemas de ventilación del edificio. En el caso del edificio LUCIA el sistema EAHX está formado por 52 conductos para la circulación del aire exterior permitiendo un caudal de 15000 metros cúbicos a la hora.

El aire exterior que entra por el sistema EAHX pasa directamente a la unidad de tratamiento del aire (UTA). La UTA es el principal sistema de climatización y se encarga del 100% de la ventilación del edificio. Entre los subsistemas que podemos encontrar está el de *free cooling* que se encarga de utilizar el aire exterior para refrigerar las instalaciones del edificio. El sistema de *free cooling* es de gran utilidad en un clima como el de Valladolid donde el rango entre temperaturas máximas y mínimas diarias es muy amplio. Otro subsistema que podemos encontrar en la UTA es el recuperador de calor que hace pasar tanto el aire de salida como el de entrada por un intercambiador, con el objetivo de suavizar la temperatura del aire exterior que entra en el edificio.

El sistema de climatización del edificio LUCIA debido a los diferentes subsistemas con los que cuenta tiene 4 modos de funcionamiento dependiendo de si se usa el sistema EAHX o no y de si se usa el sistema de *free cooling* o no.

El edificio LUCIA, también cuenta con un sistema HVAC (*Heating, Ventilating and Air Conditioning*) aire-agua para el suministro de calefacción y refrigeración. Este sistema basa su uso en 4 ventiloconvectores (*fancoils* en inglés) que son equipos que constan de un ventilador, un intercambiador de calor y un filtro pudiendo funcionar calentando o refrescando el ambiente. Los ventiloconvectores del edificio LUCIA permiten enfriamiento y calefacción simultáneo al contar con 4 tubos, dos de retorno y dos de impulsión.

Además de los sistemas destinados a la climatización y ventilación, podemos encontrar un muro fotovoltaico de doble piel en la fachada sureste y con los lucernarios en las zonas de escaleras del edificio. Los sistemas fotovoltaicos instalados permiten el paso de luz natural reduciendo las necesidades de iluminación artificial. Éste es un sistema también controlado para aumentar la eficiencia energética y basa su uso en la ocupación del edificio y el nivel de intensidad lumínica de las diferentes salas.

El edificio LUCIA debido al uso de los sistemas de eficiencia energética y al diseño adaptado reduce la demanda de energía anual de 232 kWh/m² a 65,9 kWh/m², lo que le permite ser denominado edificio nZEB.

4.1.1 Sistema de monitorización del edificio

Los diferentes sistemas implantados en el edificio para mejorar su eficiencia energética están controlados y monitorizados desde el software DESIGO INSIGHT^{[16][17]}, desarrollado por SIEMENS y que permite la gestión centralizada de los sistemas de producción de calor, frío, ventilación, iluminación... Este software se trata de un sistema SCADA (Supervisión, Control y Adquisición de Datos), típicamente usado para la gestión de instalaciones complejas y edificios industriales, y que destaca por la facilidad para ver en tiempo real el estado de los sistemas monitorizados y modificar las consignas de funcionamiento de los diferentes sistemas.

El uso de este sistema de monitorización requiere de presencialidad en el edificio LUCIA, debido a que el software está instalado sobre un servidor de la red interna del edificio, siendo este el único método de acceso. Debido a esta necesidad los trabajos tanto de estudio y análisis del software DESIGO INSIGHT, como de la extracción de datos, se han estado realizando desde el propio edificio.

En el software DESIGO podemos distinguir dos interfaces de trabajo por su importancia: por una parte, tenemos un visualizador de planta para la gestión de los sistemas mediante el control de consignas de funcionamiento y la visualización de valores en tiempo real y, por otra parte, un visualizador de tendencias que permite representar los valores de los diferentes sensores existentes previa elección de los mismos y bajo algunas limitaciones.

El visualizador de planta cuenta con un menú de selección de sistemas y subsistemas mostrando su disposición física, el estado actual de los valores de los diferentes sensores implantados y el valor de las diferentes consignas establecidas, además de permitir su modificación para gestionar el funcionamiento de cada sistema. Desde el visualizador de planta se pueden distinguir los siguientes sistemas y subsistemas:

Sistema	Subsistema	Descripción
Climatización	UTA	Unidad de tratamiento de aire
	Caudales	Volumen de los caudales de agua necesarios por el edificio para el desempeño de la actividad
	Señales	Indicadores de aviso de situaciones excepcionales en el sistema
Planta	<i>Fancoils</i> P1	Consignas y valores asignados a cada espacio de la primera planta
	<i>Fancoils</i> P2	Consignas y valores asignados a cada espacio de la segunda planta
	<i>Fancoils</i> PB	Consignas y valores asignados a cada espacio de la planta baja
Producción	<i>Heating</i>	Área de producción de frío
	<i>Cooling</i>	Área de producción de calor
Electricidad	Analizadores de red	Medidores de las propiedades eléctricas de la red del edificio completo
	Iluminación P1	Sensores asociados a los elementos de iluminación de la primera planta
	Iluminación P2	Sensores asociados a los elementos de iluminación de la segunda planta
	Iluminación PB	Sensores asociados a los elementos de iluminación de la planta baja.

Tabla 2: Composición del visualizador de planta (sistemas principales)

La otra interfaz, que se corresponde con el visualizador de tendencias, permite mostrar hasta 10 variables registradas por el sistema de forma simultánea, en un formato de gráfica de líneas y la “exportación” de los datos graficados. En concreto se tienen 3 tipos de datos dependiendo del estado temporal en el que son o han sido registrados:

- Datos en tiempo real del sistema: se pueden visualizar los datos actuales de todos los sensores existentes en el sistema, así como valores de consignas asignadas y señales entre otros.
- Datos registrados del último mes: conjunto de datos perteneciente a los últimos 30 días desde el día actual. Este conjunto de datos no contempla todos los sensores existentes del edificio LUCIA, ni los valores de consignas u otras tendencias. La reducción del número de registros se debe a problemas de espacio físico a la hora de almacenar los datos, de forma que, sólo quedan registrados aquellos datos que los administradores han considerado relevantes.
- Datos históricos: se tratan de los mismos datos que los registrados el último mes, en cuanto a sensores se refiere. La cantidad de datos a cargar en el visualizador de datos históricos se corresponde con un mes natural del año, es decir, desde el primer día del mes hasta el último, pudiendo elegir el mes a visualizar mediante la carga de un archivo con formato “.art”, propio de DESIGO, con los datos referentes al mes elegido.

Además de las herramientas definidas, el software DESIGO instalado en el edificio LUCIA cuenta con un conjunto de funcionalidades adicionales bastante amplio, aunque las aquí mostradas son las funcionalidades principales y las que usaremos en este trabajo.

4.1.2 Proceso de extracción de datos

El proceso de extracción de datos se ha descubierto como un punto crítico en el desarrollo del presente TFM, criticidad que viene dada por el hermetismo del sistema DESIGO INSIGHT y la imposibilidad de acceder a los datos históricos almacenados de una forma sencilla y poder extraer todo el conjunto almacenado en una sola instancia.

Se tiene constancia de que hay datos registrados desde febrero del año 2019, al existir archivos “.art” donde quedan registrados los datos históricos mensuales datados desde ese mes. Estos archivos “.art” no se han logrado abrir mediante otros programas, ni existe una herramienta propia de DESIGO, más allá del visualizador de tendencias, para presentar todos los datos de forma simultánea y rápida.

El problema del visualizador de tendencias es que únicamente permite visualizar un máximo de 10 variables de forma simultánea sobre las que extraer datos, lo cual implica ir alternando las diferentes variables existentes sobre el total de registradas, siendo un proceso lento y costoso. Además, la única forma de obtener los datos es mediante una acción “*copy-paste*” en un archivo externo (Excel, CSV, txt...), aumentando los tiempos del proceso de extracción.

Gracias a la ayuda del técnico del software DESIGO, se ha conseguido implantar una aplicación para realizar copias instantáneas o “*snapshots*” de los datos para un rango de fechas dadas, facilitando de esta forma la recogida de los datos actuales, pero sin resolverse el problema con los datos del histórico.

Tras realizar el estudio del Edificio LUCIA y el sistema de monitorización DESIGO, por un lado, tenemos una gran cantidad de señales y registros en el tiempo que nos permiten estudiar el edificio, y por otro lado los operadores del edificio han determinado unos modos de funcionamiento que queremos estudiar, además de predecir el consumo energético. Por este motivo en el estado del arte vamos a hablar de aprendizaje no supervisado y de predicción de series temporales para *Smart Buildings*.

4.2 Estado del arte en predicción de carga energética en *Smart Buildings*

Para la realización del presente trabajo se han considerado diversos estudios realizados donde se da una visión general sobre la importancia de gestionar y automatizar los procesos de predicción de consumos de energía, centrándose en el modelado de los sistemas HVAC[4] al ser los principales consumidores de energía de los edificios.

Naug y Biswas[18] desarrollan una nueva técnica basada en datos para el modelado del consumo energético en grandes edificios. En esta propuesta utilizan métodos de aprendizaje no supervisado para la caracterización de las condiciones ambientales en las que operan los edificios y posteriormente desarrollan modelos para la predicción del consumo energético basándose en los datos de cada conjunto de condiciones ambientales identificado. La aplicación de esta nueva técnica para el modelado del consumo energético con datos reales de un edificio del campus de la Universidad de Vanderbilt ha proporcionado importantes ahorros energéticos.

Esta idea de que las condiciones externas pueden afectar al modo de operación de un *Smart Building* ha sido explorada en otros trabajos. Arias-Requejo et al.[4] proponen un enfoque que implica el uso de algoritmos de aprendizaje no supervisado para la estimación de los modos de operación del *Smart Building* Alice Perry situado en el campus de la Universidad Nacional de

Irlanda. Una vez obtenidos los diferentes modos de operación desarrollan modelos de predicción energética específicos para cada clúster o grupo, probando que los resultados que obtienen con estos modelos ofrecen un mejor rendimiento frente al uso de un modelo general. Benedetto Grillone et al.[19] generan un modelo capaz de proporcionar estimaciones dinámicas de los ahorros energéticos conseguidos gracias a las técnicas de eficiencia energética empleadas en tres edificios comerciales de Barcelona. Para generar este modelo se basan en los patrones de consumo energético de los edificios, los cuales son obtenidos mediante la aplicación de técnicas de *clustering*.

Por otro lado, en materia de predicción energética en edificios, la mayoría de las técnicas basadas en datos para este fin quedan englobadas en el campo de la predicción de series temporales. Una serie temporal se define como la sucesión de observaciones de una variable a lo largo del tiempo ordenadas cronológicamente y con una periodicidad constante. Los métodos de predicción de series temporales se enfocan en el uso de datos históricos para la construcción de modelos predictivos. Podemos encontrar los siguientes tipos de modelo de predicción de series temporales:

- Modelos estadísticos[20]: buscan relaciones entre las diferentes variables de una muestra poblacional para abstraer patrones o comportamientos que se podrían aplicar a toda la población. Estos modelos están basados en suposiciones sobre la generalización de los datos observados con la muestra y su abstracción a poblaciones similares pero de un tamaño mucho mayor.
- Modelos de aprendizaje automático[20]: estos modelos no requieren suposiciones previas sobre las relaciones subyacentes entre las variables de entrada al modelo. Los algoritmos de aprendizaje automático procesan la cantidad ingesta de datos y descubren patrones sin la necesidad de conocer previamente ciertas reglas o relaciones. Como resultado el rendimiento de estos modelos suele ser mejor que el de los modelos estadísticos. Por contra es complicado interpretarlos, motivo por el que se les denomina modelos de caja negra.
- Modelos híbridos: representan una mezcla de los dos tipos de modelo nombrados anteriormente.

En el presente proyecto nos vamos a centrar en los modelos de aprendizaje automático y en especial en las técnicas de aprendizaje profundo (*deep learning* en inglés)[21]. El aprendizaje profundo es una familia del aprendizaje automático que busca emular el proceso de aprendizaje de los seres humanos basándose en el uso de modelos neuronales. El calificativo de profundo viene dado por la cantidad de conexiones y capas que no tenían los modelos neuronales iniciales y que se ha solucionado gracias a los avances en el procesamiento.

Las técnicas de aprendizaje profundo están cobrando cada vez mayor protagonismo para tareas de predicción de series temporales gracias al rendimiento que obtienen. La predicción de energía no está siendo una excepción y en gran cantidad de publicaciones de investigación se están usando técnicas de aprendizaje profundo para este fin. Hrnjica y Mehr[22] realizan una aproximación para la predicción de demanda energética de una solución en la nube de Smart Cities. Para ello se apoyan en el uso de redes neuronales recurrentes (RNN), en concreto de redes LSTM (*Long Short-Term Memory*)[23]. Arias-Requejo et al.[24] llevaron a cabo el desarrollo de modelos predictivos para estimar la demanda energética del *Smart Building Alice Perry* haciendo uso de redes LSTM. Sobre el mismo conjunto de datos del edificio Alice Perry, Alejandro Barón[25] aplica diversos métodos de predicción a corto plazo en series temporales con el objetivo de analizar su rendimiento. Los resultados obtenidos constatan cómo el uso de redes LSTM ofrece mejor rendimiento que otros métodos como las redes neuronales pre-alimentadas (FFN) o las redes neuronales convolucionales (CNNs).

Algunos estudios como los de Bianchi et al.[26] y Alberto Gasparin et al.[27] realizan una revisión de RNNs para la predicción de carga energética a corto plazo. En sus estudios comparan el rendimiento de redes LSTM, unidades recurrentes cerradas (GRU) y redes neuronales recurrentes de Elman (ERNN). En ambos estudios determinan que las redes LSTM obtienen mejores precisiones, pero que por el complejo proceso de implementación que tienen, la mejor opción para el pronóstico de carga son las redes ERNN.

Mocanu et al.[28] investigan el uso de diferentes técnicas de aprendizaje profundo para la predicción energética en series temporales con el objetivo de comprobar qué algoritmos ofrecen mejores resultados. Los resultados de su trabajo concluyen que el uso de modelos basados en Máquinas de Boltzmann Restringidas (RBMs) superan en rendimiento a otros métodos como son las redes neuronales artificiales (ANNs), las RNNs o las máquinas de vector soporte (SVM).

4.2.1 Métodos de *clustering*

En este apartado se van a mostrar algunos de los métodos de agrupamiento o *clustering* empleados en los estudios mencionados anteriormente para la determinación de los modos de operación en *Smart Buildings*.

- *Clustering* jerárquico (*hierarchical clustering*)[29]: es un método de agrupamiento que busca construir una jerarquía de conglomerados basada en la distancia entre las observaciones. El objetivo es que los datos que pertenezcan a un mismo clúster o grupo sean similares entre sí. Podemos distinguir dos tipos de *clustering* jerárquico:
 - La aproximación **aglomerativa** parte de que cada observación pertenece a su propio clúster. Los clústeres similares se agrupan de forma sucesiva formando

nuevos clústeres hasta obtener 1 o k clústeres. Por este motivo a este tipo de *clustering* jerárquico también se le denomina *bottom-up* (abajo-arriba), porque las iteraciones para agrupar los clústeres parten de la unidad básica.

- La aproximación **divisiva** parte de que todas las observaciones pertenecen a un mismo clúster que se divide recursivamente en dos clústeres similares hasta que solo quede un clúster para cada observación. Esta aproximación se denomina *top-down* ya que actúa de forma opuesta al *clustering* jerárquico aglomerativo.
- K-medias (*k-means*)[\[30\]](#): agrupa las diferentes observaciones de un conjunto en k grupos basándose en las cualidades o características de cada observación. De esta forma cada observación pertenece al grupo cuyo valor medio es el más cercano al de la propia observación. El algoritmo de k-medias busca minimizar la suma de las distancias entre las observaciones y el centroide (punto equidistante de los elementos de un clúster) del grupo al que pertenecen mediante la ejecución de los siguientes pasos:
 - Inicialización: se establecen k centroides sobre el conjunto de observaciones.
 - Asignación: cada observación se asigna a su centroide más cercano
 - Actualización de centroides: la posición de los centroides de cada clúster se actualiza al punto equidistante de todas las observaciones pertenecientes a su grupo.

Los dos últimos pasos se ejecutan de forma iterativa hasta que converge el algoritmo.

4.2.2 Predicción de series temporales con redes neuronales

Se van a presentar algunas de las técnicas de aprendizaje profundo para la predicción de series temporales usadas en los estudios analizados anteriormente. En concreto nos vamos a centrar en las redes neuronales explicando los principales tipos usados:

- Redes neuronales pre-alimentadas (*feed forward network* FFNs): constan de un mínimo de dos capas, una con N neuronas de entrada y otra con M neuronas de salida. En caso de tener más capas entre la capa de entrada y la capa de salida, estas se denominan capas ocultas. Cada una de las entradas propaga la información hacia la siguiente capa (*feedforward*). Las neuronas de la capa de entrada se conectan con las neuronas de la siguiente capa y así de forma sucesiva hasta la capa de salida. La información siempre viaja hacia adelante. En la figura 2 se puede ver la estructura de una red FFN.

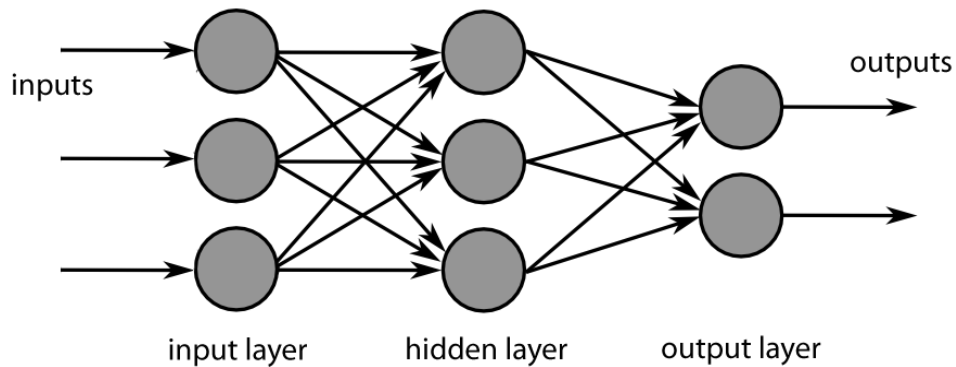


Figura 2: Estructura de una red FFN

- Redes neuronales convolucionales (CNNs): este tipo de redes destaca por tratar de imitar el córtex visual y han destacado por tener buen rendimiento en tareas de visión artificial. Entre las diversas capas que forman una red CNN existe una jerarquía, de forma que las primeras capas se encargan de identificar los elementos más básicos que se van combinando para formar patrones o elementos más complejos en las capas más profundas. Las redes convolucionales siempre tienen la misma estructura con los siguientes componentes:
 - Una capa de convolución donde se extraen las características (*features*) de los datos de entrada pudiendo ser estos datos un tensor[31] de múltiples dimensiones, aunque en el caso de la predicción de series temporales van a ser unidimensionales. En la figura 3 se puede ver un ejemplo de una convolución de 1 dimensión.

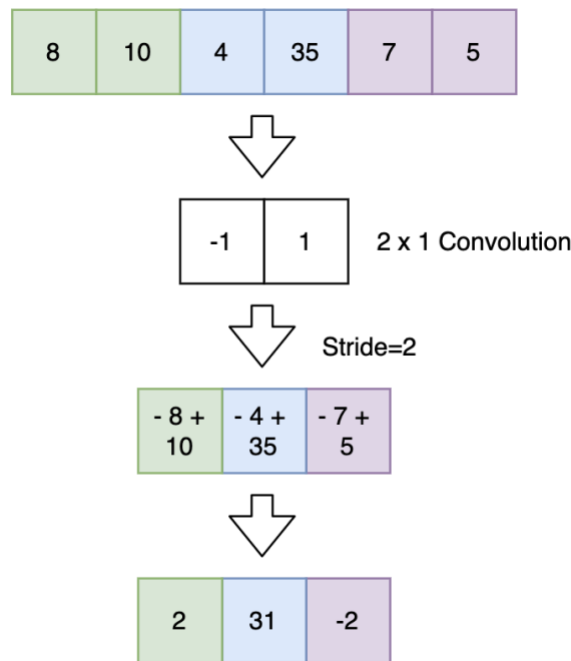


Figura 3: Convolución de 1 dimensión

- Función de activación que suele ser *ReLU* (*rectified linear unit*) para la eliminación de valores negativos empleando la función: $f(x) = \max(0, x)$

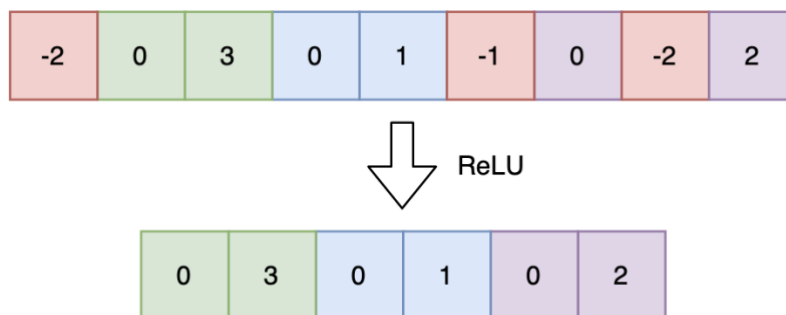


Figura 4: Función de activación ReLU

- Una capa de *Pooling* que se encarga de reducir la dimensionalidad de la matriz de datos mediante la aplicación de un filtro sobre la salida de la capa anterior y la selección de un único valor como salida. Hay diversos métodos de *Pooling*, pero generalmente el que mejor funciona es *Max Pooling* que selecciona el valor máximo como salida.

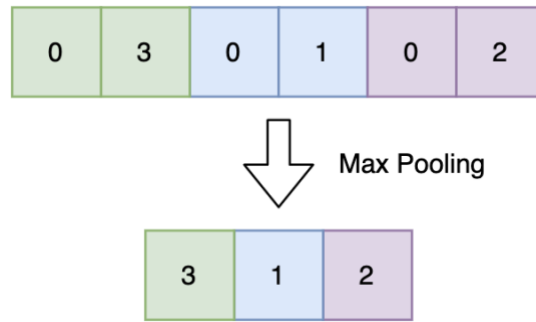


Figura 5: Función max pooling

- Una capa *Fully Connected* para terminar la red convolucional, que simplemente se trata de una red neuronal multicapa que se usa para transformar la matriz de características.
- Redes neuronales recurrentes (RNN): este tipo de redes al contrario que las redes neuronales convencionales en las que la información siempre viaja hacia delante entre capas tienen ciclos entre neuronas, lo que les permite mantener estados internos con información. Las RNN alimentan las entradas a la red con información del anterior instante temporal formando bucles sobre las propias neuronas. De esta forma, si el valor de salida de una neurona se vuelve a llevar a la entrada de la neurona, el siguiente valor de salida podría verse influenciado por la información previamente almacenada.

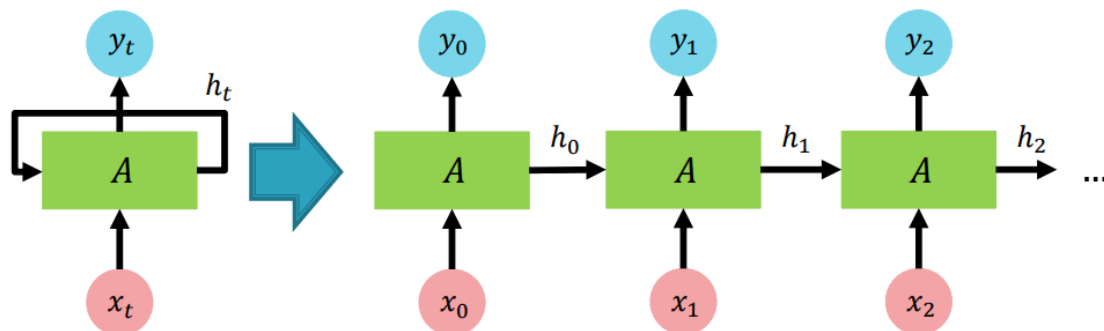


Figura 6: Red neuronal recurrente y su despliegue en el tiempo [23]

Las RNN por sus características son idóneas para el desarrollo de modelos de predicción de series temporales ya que permiten aprovechar la dependencia temporal de los datos de trabajo, de forma que ante el procesamiento de un valor en un instante concreto la información de los valores en instantes previos pueda resultar relevante.

Dependiendo de la estructura de ejecución de los modelos de redes neuronales recurrentes podemos distinguir 4 tipos:

- Uno a uno (*One to one*): este tipo consta de una única entrada y una salida y actúa como una red neuronal tradicional.

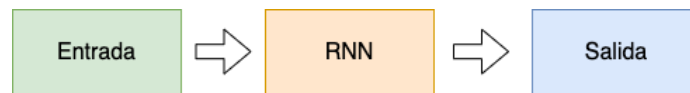


Figura 7: Estructura uno a uno

- Uno a muchos (*One to many*): este tipo de modelo produce múltiples salidas a partir de una única entrada.

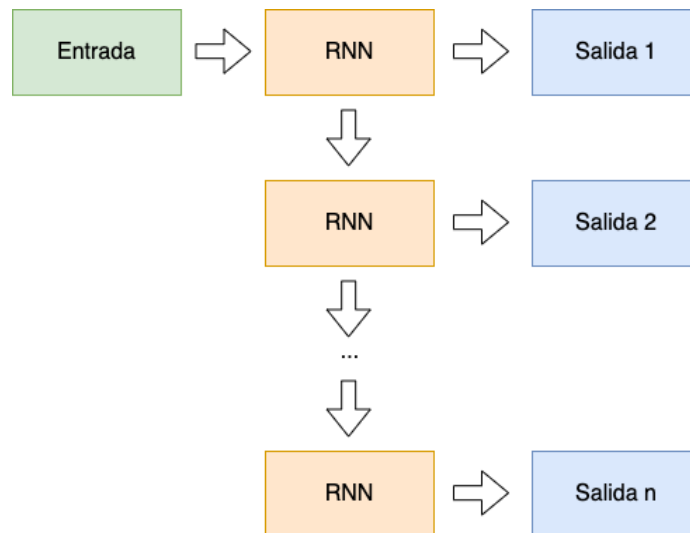


Figura 8: Estructura uno a muchos

- Muchos a uno (*Many to one*): se produce una única salida a partir de múltiples entradas. Este tipo de modelo se adecua a los problemas de predicción de secuencias que pretenden predecir el valor siguiente a una serie de observaciones dadas.

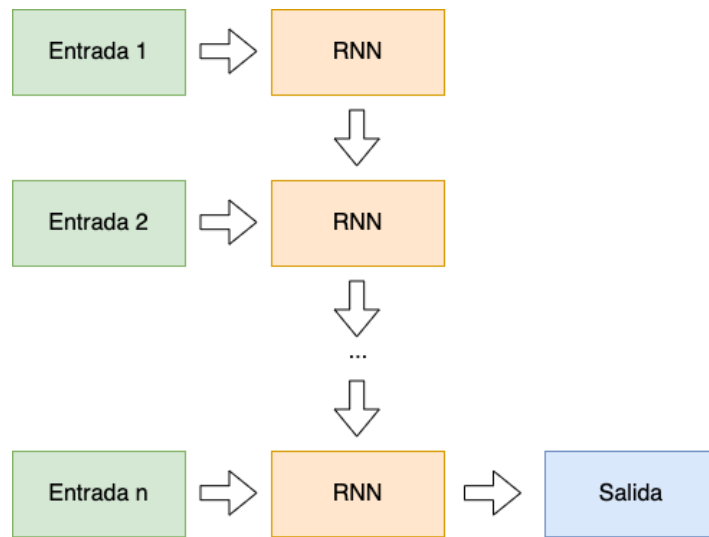


Figura 9: Estructura muchos a uno

- Muchos a muchos (*Many to many*): este tipo de modelo genera una secuencia de salida a partir de una secuencia de entrada. A este tipo de modelo también se le denomina *seq2seq* (*sequence-to-sequence*).

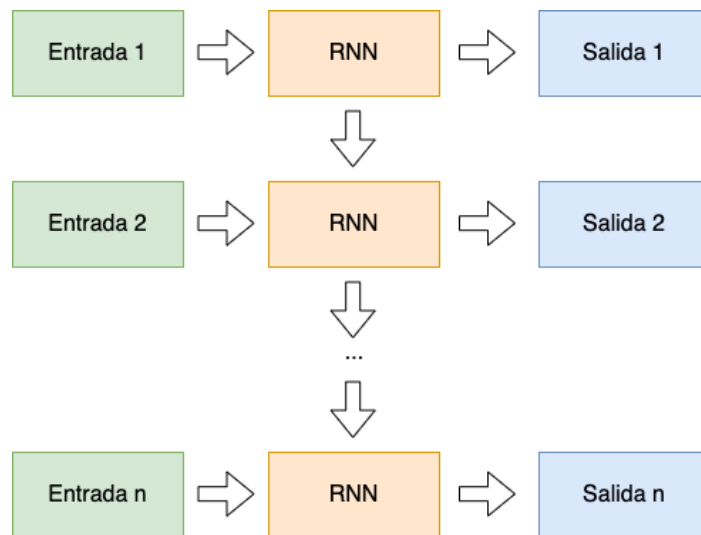


Figura 10: Estructura muchos a muchos

En los estudios mencionados anteriormente ha predominado el uso de un tipo específico de redes neuronales recurrentes como son las redes LSTM. Este tipo de RNN surgieron como una solución al problema del gradiente que se desvanece (*vanish gradient*) de las redes neuronales recurrentes estándar. Las redes LSTM pueden mantener su estado interno por varios intervalos de tiempo lo que permite resolver el problema de gradiente que se desvanece. La diferencia entre las redes LSTM y las las

redes RNN estándar la encontramos en la composición de las celdas que siguen una estructura como la representada en la figura 11. La celda de las redes LSTM está compuesta por 4 capas de red neuronal y tiene una entrada y salida adicional C y un estado oculto H . Esta nueva entrada y salida se conoce como celda de estado y permite la persistencia de la información más relevante a lo largo del proceso de entrenamiento. Esta celda de estado interactúa con la red a través de tres puertas (*gates* en inglés) que permiten modificar su contenido en caso de que la nueva información sea más relevante:

- Puerta de olvido (*forget date*): controla la información del instante previo que pasa a formar parte del presente estado mediante el uso de una función sigmoide que devuelve un 0 o un 1 dependiendo de si la información debe mantenerse.
- Puerta de entrada (*input gate*): es responsable de actualizar el valor de la celda de estado. Utilizando el producto de los valores de la celda de actualización y de la puerta candidata (*candidate gate*) y sumando el resultado del producto de los valores de la puerta de olvido y del valor previo de la celda de estado se actualiza el valor de la celda de estado.
- Puerta de salida (*output gate*): se encarga de actualizar el valor del estado oculto mediante el producto de aplicar la tangente hiperbólica a la celda de estado y el valor de la puerta de salida.

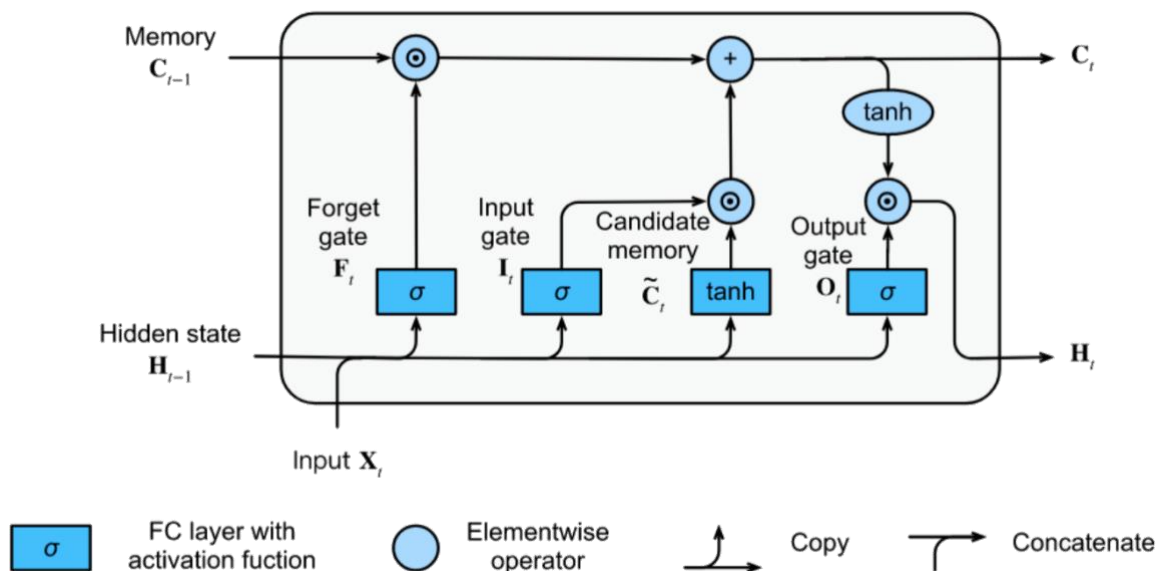


Figura 11: Detalle de la estructura de una celda de red LSMT [23]

Los trabajos realizados por Arias-Requejo et al.[4][24] van a ser utilizados como punto de partida del presente proyecto y como base para comparar los resultados que se irán obteniendo

con el desarrollo del TFM. Además, para la determinación de los estados de funcionamiento del edificio vamos a usar los algoritmos de aprendizaje no supervisado k-medias y *clustering* jerárquico, mientras que para la predicción energética nos vamos a centrar en el uso de redes LSTM. La selección de las redes LSTM se apoya en el rendimiento que han obtenido los modelos desarrollados con este tipo de redes neuronales en los estudios analizados anteriormente sobre predicción energética.

Capítulo 5

5. Modelo de datos

En los siguientes apartados se va a mostrar la exploración realizada sobre los *raw data* y los procesos de limpieza, transformación y de selección de características o *features* (en inglés) implementados. Estos procesos han permitido generar los *datasets* de trabajo empleados a la hora de construir tanto los modelos de *clustering* para determinar los estados de funcionamiento del edificio, como los modelos para la predicción del consumo energético del edificio LUCIA.

5.1 Exploración del *raw data*

Raw data o datos sin procesar son aquellos que tenemos en el momento de su recolección. Estos datos raramente están en un formato adecuado para su uso, además del desconocimiento que tenemos acerca de las diferentes variables, registros o tipos de datos que podemos encontrarlos. La exploración de datos tiene como objetivo mejorar el conocimiento sobre un conjunto de datos para facilitar su uso en tareas posteriores. Por este motivo es vital extraer y comprender toda la información posible del *raw data*.

Los *raw data* del edificio LUCIA se ha obtenido de los diferentes sensores monitorizados por el software DESIGO y siguiendo el método de recolección definido en el apartado [4.1.2](#). En total se han obtenido datos de 29 meses desde el mes de enero de 2019 al mes de mayo de 2021, ambos incluidos. En concreto son datos de un total de 870 días, entre el 01/01/2019 y el 21/05/2021.

Tras realizar un primer análisis del *raw data* y ver cómo se estaban almacenando las diferentes variables monitorizadas, se ha encontrado un patrón para todas ellas siguiendo una misma estructura de datos como la que se representa en tabla 3, donde se muestra la estructura de datos asociadas al sensor que mide la humedad del aire en el exterior, medida como humedad relativa en tanto por ciento.

ditech://LUCIA:CondExt' HuOa.PrVal;CondExt'Trn d2		
LUCIA:Condiciones exteriores'Humedad de aire exterior.Valor actualValor actual;Tendencia 01		
Fecha/hora	%RH	Tag de tipo
1/1/2021 0:04:47	83.1	Bien
1/1/2021 0:19:47	84.8	Bien
1/1/2021 0:34:47	86.5	Bien
...
31/1/2021 23:56:13	81.6	Bien

Tabla 3: Estructura de los datos para cada sensor monitorizado

En la estructura representada anteriormente encontramos un primer campo en la primera fila con la dirección del sensor y nombre asignado en el software DESIGO. En el primer campo de la segunda fila se muestra una descripción del sensor incluyendo el sistema y subsistema a los que pertenece, el nombre o el tipo de registro entre otros datos. Los campos de las dos primeras filas son informativos e independientes de los registros temporales que se muestran en el visualizador de tendencias del software DESIGO para cada sensor. En la tercera fila podemos encontrar la cabecera de las tres columnas con los datos que se están registrando del sensor y que si se corresponden con los que se representan en el visualizador de tendencias:

- La primera columna registra los datos de fecha y hora como nos indica el primer campo de la cabecera.
- En la segunda columna podemos encontrar los valores registrados por el sensor, siendo la cabecera de la columna el tipo de unidad de medida de estos valores.
- La tercera columna nos permite conocer el estado y validez de cada registro del visualizador de tendencias. La cabecera de esta columna se denomina “Tag de tipo” y hace referencia a las diferentes etiquetas o *tags* asociados a un registro.

Durante el proceso de recogida de datos se ha podido observar cómo el número de variables monitorizadas en los diferentes meses puede variar, llegando a encontrar grandes diferencias. Los meses de mayo de 2019 con 200 variables y el mes de octubre de 2020 con 340 variables

son los meses con menor y mayor número de monitorizaciones respectivamente. Esta gran diferencia, que también ocurre con otros meses, aunque en menor medida, generalmente se debe a las variables del “área de unidades terminales” que registran los datos de temperatura en los diferentes habitáculos del edificio y no siempre quedan registradas en los diferentes meses. Además de las desigualdades ocasionadas por las variables del “área de unidades terminales” también se han encontrado diferencias causadas por variables de otros sistemas del edificio. Estas discrepancias entre las variables monitorizadas para cada mes provocan que el número de variables comunes a todos los meses sea 155, bastante más bajo que el número de variables de mayo de 2019.

Otro detalle que se ha podido observar durante el primer estudio de los datos es la desigualdad en el número de registros para cada variable y la fluctuación en la frecuencia de registro incluso para una misma variable.

Finalizados el proceso de extracción de datos y un primer estudio de los datos totales recogidos, se ha obtenido la siguiente información global del *raw data*:

- Datos de un total de 29 meses
- Un total de 870 días entre la fecha del primer y último registro. (01/01/2019 - 21/05/2021)
- Un total de 392 variables monitorizadas entre los diferentes meses.
- Una media de 246 variables monitorizadas por mes. Con 340 variables monitorizadas para el mes con mayor número y 200 para el mes con menor número de monitorizaciones.
- 155 variables comunes a los 29 meses.
- 12515668 registros totales.
- Variable presente en todos los meses con más registros: *CondExt`TOa* (temperatura del aire exterior) con 115042 registros.
- Variable presente en todos los meses con menos registros: *AInt`PI`AnlzRed12`AI65* (analizador de red 12 de la primera planta) con 3479 registros.
- Unidades de medida de las diferentes variables: kVAr (kilovoltamperio reactivo), kW (kilovatio), m³/h (metros cúbicos por hora), kWh (kilovatio hora), MWh (megavatio hora), Pa (pascales), g/kg (gramos por kilogramo), kJ/kg (kilojulios por kilogramo), ppm (partes por millón), bar, % (porcentaje), %RH (porcentaje de humedad relativa), L

(litros), Valor (dependiendo de la variables o sensor puede ser un tipo de unidad distinta), °C (grados Celsius).

- Tipos de sistema con registros: SeVar (señales varias), CondExt (condiciones exteriores), AProd (área de producción), ACl (área de climatización), AInt (área de integraciones) y AUT (área de unidades terminales).

A continuación, pasamos a explicar qué son estos sistemas en base a los cuales se agrupa u organizan las medidas del sistema.

5.1.1 Clasificación de variables y sistemas

En el visualizador de tendencias del software DESIGO las variables están clasificadas por el tipo de sistema al que pertenecen según la funcionalidad del sensor/medidor del que provienen los datos. El sistema al que pertenece una variable lo podemos identificar revisando la descripción de la variable. Siguiendo el criterio de clasificación de variables del visualizador de tendencias de DESIGO, tenemos los siguientes grupos:

- Las variables que en el software DESIGO pertenecen al **sistema de señales varias** (SeVar) generalmente registran datos de caudal de agua en litros, aunque podemos ver alguna variable que registra datos de temperatura en localizaciones como el sótano, doble piel del edificio o el ático.
- Uno de los principales sistemas que podemos encontrar en el edificio LUCIA es el **sistema o área de producción** (AProd). Las variables que DESIGO engloba dentro del área de producción generalmente registran datos de temperatura (en algún caso aislado datos de presión) del agua que se encuentra en el circuito de los subsistemas involucrados tanto en la producción de calor como en la de frío. Las mediciones se realizan en diferentes dispositivos como pueden ser los *fancoils* de las diferentes plantas, la enfriadora, el climatizador...

El sistema de producción es el principal componente del sistema HVAC del edificio y se presenta como un sistema clave sobre el que centrarse para determinar los modos de operación del edificio. Esto se debe a la dependencia existente entre los diferentes subsistemas implicados en la producción de calor y frío y las condiciones meteorológicas existentes y el estado de ocupación del edificio. Debemos comprobar si se pueden identificar modos usando estas variables de forma análoga a los estudios expuestos en el apartado [4.2](#) donde se usan las variables externas (meteorológicas) para ese mismo fin.

- Existen dos variables que están clasificadas como **condiciones exteriores** (CondExt). Estas variables se corresponden con los sensores de una pequeña estación meteorológica

instalada en el edificio que registran los datos de la temperatura y la humedad relativa del aire exterior. Como se puede ver en el apartado [4.2](#) la mayoría de los estudios realizados para la determinación de los modos de operación del edificio se basan en las condiciones meteorológicas. El hecho de tener estos datos de las condiciones exteriores nos va a permitir emplearlos con algunos modelos de *clustering* para comparar los resultados obtenidos con los de otros modelos que utilicen datos del área de producción y determinar si a partir de estos últimos también se pueden determinar los estados de funcionamiento del edificio LUCIA.

- Otro conjunto de variables o sensores que requieren de especial atención son las pertenecientes al **área de integraciones (AInt)**. En concreto a esta área pertenecen los analizadores de red del edificio. Un analizador de red es un instrumento destinado a la monitorización de todos los fenómenos eléctricos de una instalación. En el caso del edificio LUCIA los analizadores de red registran datos de energía activa, de potencia total activa y de energía reactiva. Los datos de energía activa que están registrando los analizadores de red realmente son datos de energía acumulada con el paso del tiempo. Por este motivo con el avance del tiempo los mismos analizadores de red presentan cifras de kilovatios por hora y megavatios por hora superiores.

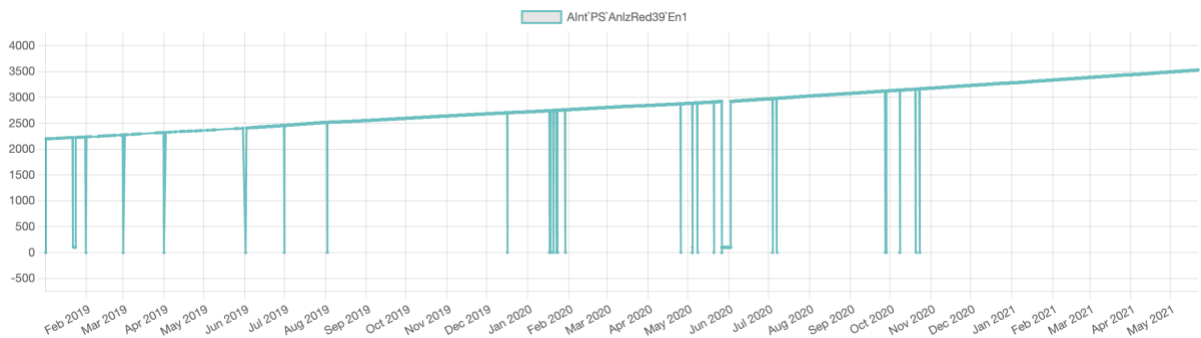


Figura 12: Serie de datos del analizador de red 39 en el tiempo

En la figura 12 podemos ver los datos de consumo energético en MWh registrados por el analizador de red 39 de la planta sótano, que en concreto registra el consumo general de la red del edificio. Como podemos ver el consumo tiene una tendencia creciente y a pesar de que se pueden vislumbrar algunos valores atípicos o *outliers*, se tratan de valores acumulados. La obtención de las mediciones de consumo energético entre registros requiere del procesado de todas las variables cuya unidad de medida sea o bien kWh o MWh.

Los datos de los analizadores de red van a ser de gran utilidad a la hora de implementar los modelos de predicción de consumo energético. Los datos que están registrando estos analizadores de red van a ser bastantes constantes y sería difícil determinar los estados de funcionamiento a partir de ellos. Esto se debe a que uno de

los objetivos de un edificio nZEB es mejorar la eficiencia energética independientemente de las condiciones exteriores existentes. Esto implica buscar un consumo de energía muy bajo, y tener unos consumos similares y lo más reducidos posible durante todo el año.

- Las variables del **área de unidades terminales** (AUT) registran los datos de la temperatura existente en los diferentes habitáculos del edificio.
- Las variables pertenecientes al **área de climatización** registran datos con unidades de medida variadas. Entre estas variables podemos encontrar registros de temperatura del aire, de humedad relativa del aire, de energía... Cabe destacar que todas las variables pertenecientes a esta área forman parte de la unidad de tratamiento del aire (UTA).

5.1.2 Datos de consumo energético del edificio

En la planta sótano del edificio LUCIA se encuentra alojado un CPD de la Universidad de Valladolid cuyo consumo es independiente del funcionamiento del edificio. Por lo tanto, los datos de consumo energético referentes al CPD no se tendrán en cuenta a la hora de obtener los datos de consumo energético del edificio LUCIA.

Tras analizar las diferentes variables del *raw data* y en concreto las que se corresponden con los analizadores de red del edificio, se ha podido observar cómo los datos de consumo energético que hacen referencia únicamente al edificio van a depender de dos variables: por una parte tenemos el analizador de red 39 de la planta sótano que registra los valores de consumo energético del edificio completo incluyendo los consumos del CPD en MWh, y por otra parte el analizador de red 29 de la planta sótano que registra únicamente el consumo del CPD en kWh. El consumo energético que consideraremos únicamente del edificio LUCIA se calculará como la diferencia entre los datos de los dos analizadores de red, aunque requerirá de un procesado previo de los datos de ambas variables que se mostrará en los siguientes apartados.

$$\text{Consumo Edificio LUCIA}(t) = \text{AnlzRed39}(t) * 1000 - \text{AnlzRed29}(t)$$

En la tabla 4 se muestran las estadísticas de las dos variables pertenecientes a los analizadores de red 29 y 39 (el primer y último valor de las variables se registraron el 01/01/2019 y 21/05/2021 respectivamente). En las figuras 12 (mostrada anteriormente) y 13 se pueden visualizar las gráficas con los consumos energéticos a lo largo del tiempo de ambas variables.

Variable	N.º de Registros	Mínimo	Media	Máximo	Desviación estándar	Valores nulos
AnlzRed39 (MWh)	66916	-4	2941.5	3529.95	102550.85	0
AnlzRed29 (kWh)	66944	-4	195319.68	372232.2	429.43	0

Tabla 4: Estadísticas de los analizadores de red 29 y 39

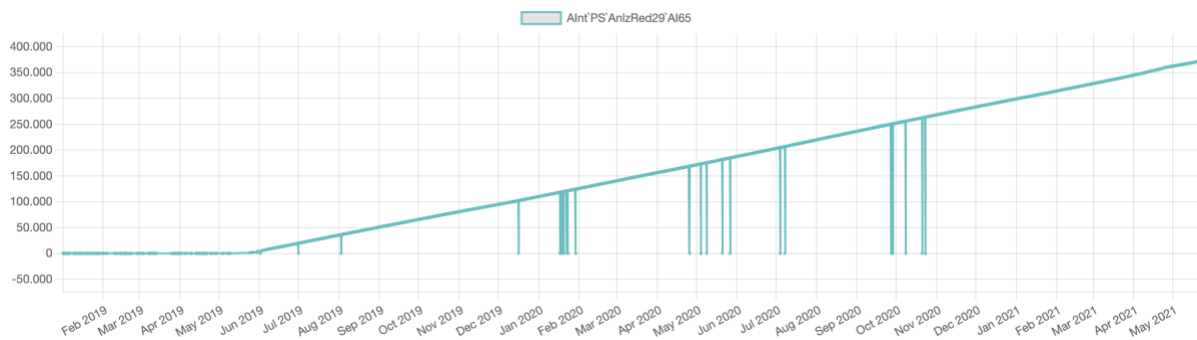


Figura 13: Serie de datos del analizador de red 29 en el tiempo

5.1.3 Datos para la determinación de los estados de funcionamiento del edificio

Debido a la gran cantidad de variables disponibles y a la baja influencia que tienen algunas sobre el sistema HVAC del edificio, el cual queremos modelar por ser el mayor consumidor de energía del edificio LUCIA, se han seleccionado las variables del área de producción para determinar los estados de funcionamiento del edificio.

El área de producción representa la parte más importante del sistema HVAC del edificio LUCIA. Por este motivo los datos de las variables clasificadas dentro de esta área se presentan como información clave para el modelado de sus estados de funcionamiento.

En concreto del conjunto de variables comunes a todos los meses englobadas dentro del área de producción (39 variables), nos vamos a centrar en las que se encargan de registrar los datos de temperatura del agua que se encuentra en los circuitos de los subsistemas tanto de producción de frío como de producción de calor (37 variables). Todos los datos de temperatura están medidos en grados Celsius y para todas las variables el primer registro se produjo el 01/01/2019, mientras que el último está datado el 21/05/2021. En la tabla 5 se puede ver el listado de variables junto con la descripción de la variable en el software DESIGO y en la tabla 6 se muestran las estadísticas de las variables. En las figuras 14 y 15 se pueden ver las gráficas con

los datos de temperatura del agua de retorno de los *fancoils* de la primera planta para la producción de frío y calor respectivamente.

Etiqueta	Variable	Sistema	Equipo
AProd`C`CtoP`Abs`TImpRef	Temperatura Impulsión de refrigeración	Refrigeración y enfriamiento - Circuito primario	Equipo de absorción
AProd`C`CtoP`Abs`TInTwr	Temperatura de entrada torre	Refrigeración y enfriamiento - Circuito primario	Equipo de absorción
AProd`C`CtoP`Abs`TOutTwr	Temperatura de salida torre	Refrigeración y enfriamiento - Circuito primario	Equipo de absorción
AProd`C`CtoP`Abs`TRetRef	Temperatura de retorno de refrigeración	Refrigeración y enfriamiento - Circuito primario	Equipo de absorción
AProd`C`CtoP`Abs`TRetTwr	Temperatura de retorno de torre	Refrigeración y enfriamiento - Circuito primario	Equipo de absorción
AProd`C`CtoP`Ch`TIn	Temperatura de entrada	Refrigeración y enfriamiento - Circuito primario	Enfriadora
AProd`C`CtoP`Ch`TOut	Temperatura de salida	Refrigeración y enfriamiento - Circuito primario	Enfriadora
AProd`C`CtoP`CtoAbsH`TRet	Temperatura de retorno	Refrigeración y enfriamiento - Circuito primario	Circuito de absorción
AProd`C`CtoP`CtoAbsH`TSu	Temperatura de impulsión	Refrigeración y enfriamiento - Circuito primario	Circuito de absorción
AProd`C`CtosS`CL`TRet	Temperatura de retorno	Refrigeración y enfriamiento - Circuitos secundarios	Climatizador
AProd`C`CtosS`CL`TSu	Temperatura de impulsión	Refrigeración y enfriamiento - Circuitos secundarios	Climatizador
AProd`C`CtosS`FncP1`TRet	Temperatura de retorno	Refrigeración y enfriamiento - Circuitos secundarios	<i>Fancoils</i> primera planta
AProd`C`CtosS`FncP1`TSu	Temperatura de impulsión	Refrigeración y enfriamiento - Circuitos secundarios	<i>Fancoils</i> primera planta

AProd`C`CtosS`FncP2`TRet	Temperatura de retorno	Refrigeración y enfriamiento - Circuitos secundarios	<i>Fancoils</i> segunda planta
AProd`C`CtosS`FncP2`TSu	Temperatura de impulsión	Refrigeración y enfriamiento - Circuitos secundarios	<i>Fancoils</i> segunda planta
AProd`C`CtosS`FncPB`TRet	Temperatura de retorno	Refrigeración y enfriamiento - Circuitos secundarios	<i>Fancoils</i> planta baja
AProd`C`CtosS`FncPB`TSu	Temperatura de impulsión	Refrigeración y enfriamiento	<i>Fancoils</i> planta baja
AProd`C`TColRet	Temperatura del colector de retorno	Refrigeración y enfriamiento	Colector
AProd`C`TColSu	Temperatura de colector de impulsión	Refrigeración y enfriamiento	Colector
AProd`C`TDep	Temperatura del depósito de inercia	Refrigeración y enfriamiento	Depósito de inercia
AProd`H`CtosS`ACS`TAguFri	Temperatura del agua fría	Calefacción - Circuitos secundarios	Agua caliente sanitaria
AProd`H`CtosS`ACS`TDep	Temperatura del depósito ACS	Calefacción - Circuitos secundarios	Agua caliente sanitaria
AProd`H`CtosS`ACS`TRet	Temperatura de retorno	Calefacción - Circuitos secundarios	Agua caliente sanitaria
AProd`H`CtosS`ACS`TRetSec	Temperatura de retorno secundario	Calefacción - Circuitos secundarios	Agua caliente sanitaria
AProd`H`CtosS`ACS`TSu	Temperatura de impulsión	Calefacción - Circuitos secundarios	Agua caliente sanitaria
AProd`H`CtosS`ACS`TSuSec	Temperatura de impulsión secundaria	Calefacción - Circuitos secundarios	Agua caliente sanitaria
AProd`H`CtosS`CI`TRet	Temperatura de retorno	Calefacción - Circuitos secundarios	Climatizador
AProd`H`CtosS`CI`TSu	Temperatura de impulsión	Calefacción - Circuitos secundarios	Climatizador

AProd`H`CtosS`FncP1`TRet	Temperatura de retorno	Calefacción - Circuitos secundarios	Fancoils primera planta
AProd`H`CtosS`FncP1`TSu	Temperatura de impulsión	Calefacción - Circuitos secundarios	Fancoils primera planta
AProd`H`CtosS`FncP2`TRet	Temperatura de retorno	Calefacción - Circuitos secundarios	Fancoils segunda planta
AProd`H`CtosS`FncP2`TSu	Temperatura de impulsión	Calefacción - Circuitos secundarios	Fancoils segunda planta
AProd`H`CtosS`FncPB`TRet	Temperatura de retorno	Calefacción - Circuitos secundarios	Fancoils planta baja
AProd`H`CtosS`FncPB`TSu	Temperatura de impulsión	Calefacción - Circuitos secundarios	Fancoils planta baja
AProd`H`CtosS`TColRet	Temperatura del colector de retorno	Calefacción - Circuitos secundarios	Colector
AProd`H`CtosS`TColSu	Temperatura del colector de impulsión	Calefacción - Circuitos secundarios	Colector
AProd`H`TColBot	Temperatura del colector de botellón	Calefacción	Colector

Tabla 5: Variables de temperatura del área de producción

Variable	N.º de registros	Mínimo	Media	Máximo	Desviación estándar	Valores nulos
AProd`C`CtoP`Abs`TImpRef (°C)	86041	3	15.83	28.1	4	0
AProd`C`CtoP`Abs`TInTwr (°C)	87257	-3	18.77	36.9	4.85	0
AProd`C`CtoP`Abs`TOutTwr (°C)	87256	-3	16.49	36.7	5.27	0
AProd`C`CtoP`Abs`TRetRef (°C)	86040	0	16.22	29	3.76	0
AProd`C`CtoP`Abs`TRetTwr (°C)	86039	0	17.99	37.8	4.17	0
AProd`C`CtoP`Ch`TIn (°C)	86560	3	16.38	26.4	3.57	0

AProd`C`CtoP`Ch`TOut (°C)	86560	0	16.13	28.7	4.03	0
AProd`C`CtoP`CtoAbsH`TRet (°C)	86042	3	39.9	88.6	14.73	0
AProd`C`CtoP`CtoAbsH`TSu (°C)	86040	3	35.92	88.6	16.71	0
AProd`C`CtosS`CL`TRet (°C)	86042	0	17.54	29.2	3.22	0
AProd`C`CtosS`CL`TSu (°C)	86042	3	17.75	30	3.22	0
AProd`C`CtosS`FncP1`TRet (°C)	86042	3	16.79	28.4	2.98	0
AProd`C`CtosS`FncP1`TSu (°C)	86043	0	16.47	28.2	3.49	0
AProd`C`CtosS`FncP2`TRet (°C)	86043	3	16.67	28.7	3.04	0
AProd`C`CtosS`FncP2`TSu (°C)	86043	3	16.02	28.2	3.62	0
AProd`C`CtosS`FncPB`TRet (°C)	86043	3	17.01	28.2	3.04	0
AProd`C`CtosS`FncPB`TSu (°C)	86043	3	16.4	27.2	3.42	0
AProd`C`TColRet (°C)	86041	3	15.81	28.6	3.12	0
AProd`C`TColSu (°C)	86041	0	15.1	25.5	3.58	0
AProd`C`TDep (°C)	86558	3	15.77	24.9	4.19	0
AProd`H`CtosS`ACS`TAguFri (°C)	86043	3	18.75	30.2	4.08	0
AProd`H`CtosS`ACS`TDep (°C)	86043	3	54.45	71.8	6.7	0
AProd`H`CtosS`ACS`TRet (°C)	86042	3	40.58	79.6	18.52	0
AProd`H`CtosS`ACS`TRetSec (°C)	86043	3	37.78	59.3	10.22	0
AProd`H`CtosS`ACS`TSu (°C)	86042	3	44.25	90.1	24.08	0
AProd`H`CtosS`ACS`TSuSec (°C)	86041	0	43.43	63.9	9.5	0
AProd`H`CtosS`Cl`TRet (°C)	86043	3	41.39	90.6	18.53	0
AProd`H`CtosS`Cl`TSu (°C)	86043	3	42.49	91.3	23.08	0
AProd`H`CtosS`FncP1`TRet (°C)	86043	3	37.31	67.4	13.04	0
AProd`H`CtosS`FncP1`TSu (°C)	86043	3	33.56	83.6	17.95	0
AProd`H`CtosS`FncP2`TRet (°C)	86042	3	36.6	72.1	12.74	0
AProd`H`CtosS`FncP2`TSu (°C)	86042	3	33.33	82.7	17.35	0
AProd`H`CtosS`FncPB`TRet (°C)	86042	3	38.71	70.7	14.43	0
AProd`H`CtosS`FncPB`TSu (°C)	86042	3	38.74	83.7	20.13	0
AProd`H`CtosS`TColRet (°C)	86042	3	47.56	90	19.31	0

AProd`H`CtosS`TColSu (°C)	86042	3	57.03	92.7	24.34	0
AProd`H`TColBot (°C)	80224	1.9	65.65	93.4	21.42	0

Tabla 6: Estadísticas de las variables de temperatura del área de producción

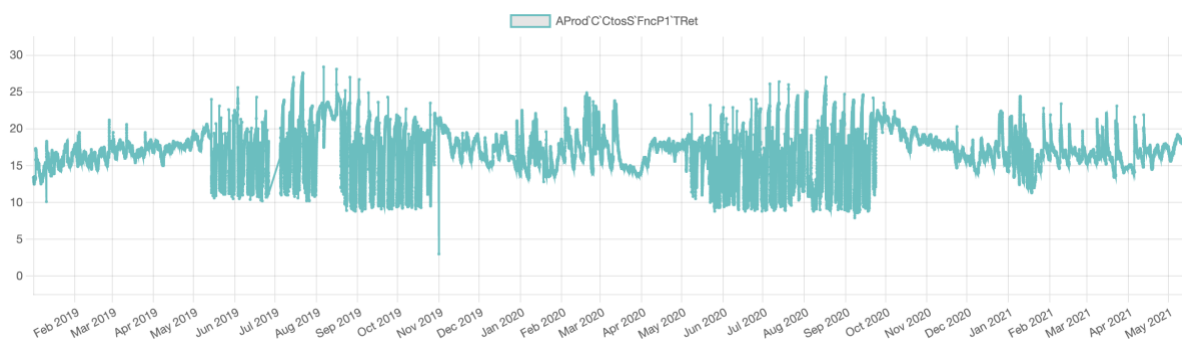


Figura 14: Temperatura de retorno del fancoil de la primera planta para la producción de frío

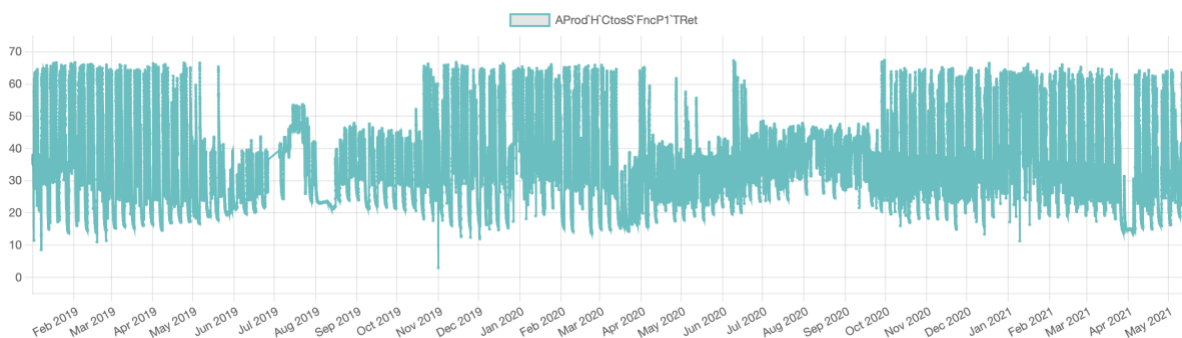


Figura 15: Temperatura de retorno del fancoil de la primera planta para la producción de calor

Por otro lado, además de las variables del área de producción vamos a emplear las variables de condiciones externas para generar un modelo de *clustering* sobre el que comparar los resultados del modelo con los datos del área de producción con el fin validar el uso de estos últimos para la estimación de los estados de funcionamiento.

En la tabla 7 se muestran las estadísticas de las dos variables pertenecientes al grupo de condiciones externas (el primer y último valor de las variables se registraron el 01/01/2019 y 21/05/2021 respectivamente), mientras que en las figuras 16 y 17 se pueden ver las gráficas con los valores de ambas variables.

Variable	N.º de Registros	Mínimo	Media	Máximo	Desviación estándar	Valores nulos
CondExt`HuOa (%RH)	87265	-3	64.42	100	26.12	0
CondExt`TOa (°C)	115042	-5.5	14.07	46.7	8.43	0

Tabla 7: Estadísticas de las variables de condiciones externas

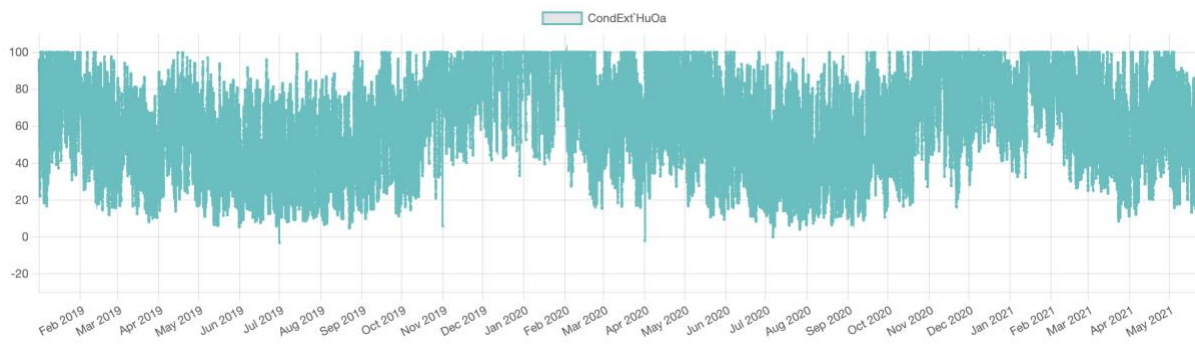


Figura 16: Porcentaje de humedad relativa del aire exterior

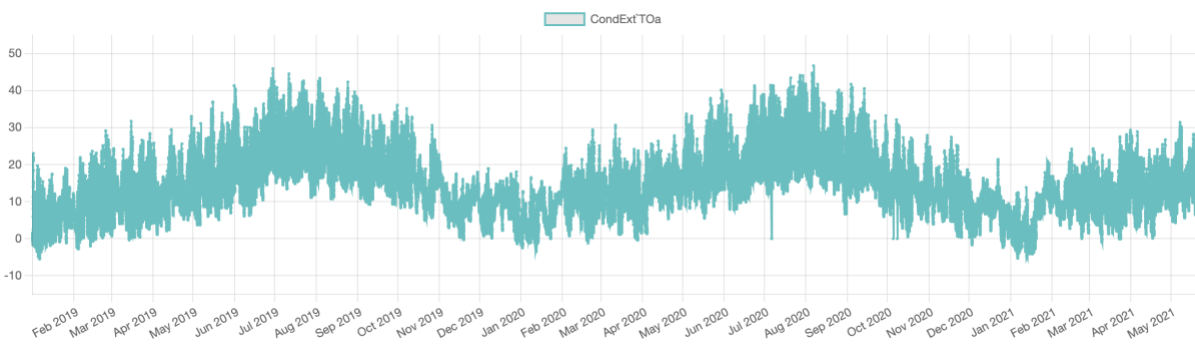


Figura 17: Temperatura del aire exterior

5.2 Limpieza de datos

Durante el proceso de exploración de datos apoyándonos en el uso de gráficas para representar los datos en el tiempo y en el uso de histogramas y diagramas de caja (*boxplots*) identificamos algunos *outliers* tanto en los datos de consumo energético como en los datos del área de producción.

La eliminación de *outliers* se ha realizado empleando diferentes métodos para los datos del área de producción y de consumo energético. Por una parte, los *outliers* identificados en las variables del área de producción y las variables de condiciones externas tras aplicar el método *Z-Score*[\[32\]](#), han sido eliminados. El método *Z-Score* nos indica el número de desviaciones típicas que hay entre el valor de un registro y la media de los registros del conjunto. En concreto en nuestro caso hemos escogido como límite un *Z-Score* igual a 3, de forma que cualquier valor que diste del valor medio en la suma de 3 desviaciones típicas se considera *outlier*. Por otro lado, la eliminación de los *outliers* de los analizadores de red 29 y 39 se ha realizado apoyándose tanto en los histogramas como en las gráficas de datos de las variables, y aplicando un rango de valores que dejase fuera registros de consumo energético considerados anómalos como puede ser el caso de algunos valores negativos representados en las gráficas.

En el proceso de limpieza de datos también se han filtrado los registros de las variables por el campo “Tag de tipo” dejando únicamente aquellos registros que tuviesen asociado el tag “Bien”, que nos indica que la medición se ha producido y registrado correctamente.

A pesar de que no se ha identificado ningún valor nulo en la exploración de las variables tanto del área de producción como de las variables de consumo energético, se ha procedido a aplicar un método para la eliminación de valores nulos para verificar la inexistencia de éstos.

5.3 Transformación de datos y extracción de variables

Como se ha indicado anteriormente los datos de consumo energético de los analizadores de red 29 y 39 son datos acumulados siendo necesario obtener el valor de consumo energético entre registros. Para la obtención de los datos de consumo energético entre registros hacemos uso de la estructura *DataFrame* de *Pandas* donde cada variable representa una columna del *DataFrame*. Empleamos la función *diff*[\[33\]](#) para hallar la diferencia entre los valores de cada fila y la fila anterior obteniendo el consumo energético que hay entre fechas de registro. Con los nuevos valores hallados generamos dos nuevas columnas en el *DataFrame*: una que representa los datos de consumo energético del edificio incluyendo el consumo del CPD en kWh (*totalEnergy*) y otra con los datos de consumo energético del CPD también en kWh (*cpdEnergy*).

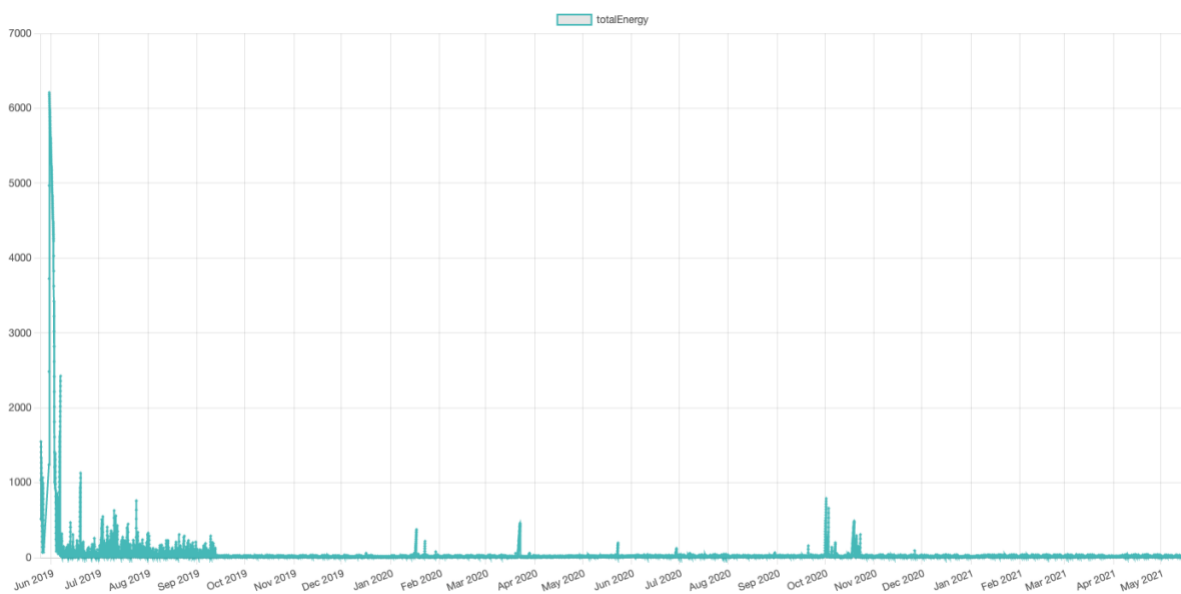


Figura 18: Consumo energético horario del edificio LUCIA + CPD sin procesar

Como se puede observar en la figura 18 donde se muestran los datos de consumo energético la nueva variable *totalEnergy* una vez aplicado el proceso de transformación, se han generado

múltiples outliers, especialmente en las fechas iniciales hasta octubre de 2019. Al revisar los datos de la nueva variable *cpdEnergy* generada, presenta el mismo patrón con datos inconsistentes hasta casi octubre de 2019. Tras revisar los *raw data* de nuevo, se ha podido observar como el número de registros en los meses anteriores a octubre de 2019 de los analizadores de red 29 y 39 son mucho más bajos que en otros meses siendo el motivo por el que se están generando datos inconsistentes al emplear el método de transformación. Para la generación de las nuevas variables de consumo energético finalmente se usan los datos disponibles desde octubre de 2019.

En los datos de las nuevas columnas tras aplicar el método *Z-Score* identificamos algunos *outliers*, aunque no los eliminamos. En este caso aplicamos una máscara sobre el conjunto de datos de forma que aquellos valores que no se encuentran dentro del rango impuesto al aplicar el método *Z-Score* se reemplazan por interpolación. Este último paso se ha realizado para evitar un conflicto a la hora de agrupar los datos con el uso de la estructura *DataFrame*.

En el caso del analizador de red 39, y previamente al proceso detallado anteriormente, convertimos los valores que se encontraban en MWh a kWh. En el siguiente código se puede ver el proceso de generación de las nuevas variables:

```
# Analizador de red 39 (MWh)
if(key == 'AInt`PS`AnlzRed39`En1'):
    # Conversión de datos a kWh *= 1000
    df['totalEnergy'] = df[key] * 1000
    # Diferencia entre filas
    df['totalEnergy'] = df['totalEnergy'].diff()
    # Limpieza e imputación
    valid = np.abs(stats.zscore(df['totalEnergy'], nan_policy='omit'))
    < 3
    df['totalEnergy'] = df['totalEnergy'].where(valid).interpolate()
    df = df[df['totalEnergy'] > 0]
# Analizador de red 29
if(key == 'AInt`PS`AnlzRed29`AI65'):
    # Diferencia entre filas
    df['cpdEnergy'] = df[key].diff()
    # Limpieza e imputación
    valid = np.abs(stats.zscore(df['cpdEnergy'], nan_policy='omit')) <
    3
    df['cpdEnergy'] = df['cpdEnergy'].where(valid).interpolate()
    df = df[df['cpdEnergy'] > 0]
```

Una vez obtenidas las variables con los consumos del edificio y del CPD se han agrupado los datos de forma horaria utilizando el valor medio de los registros pertenecientes a cada hora.

Al agruparse los valores por fechas se han generado algunos valores ausentes que han sido eliminados del conjunto de datos.

A partir de los datos agrupados se ha generado una nueva variable que representa el consumo energético del edificio LUCIA sin el CPD:

```
# Consumo energético del edificio LUCIA
df['totalBuildingEnergy']= df['totalEnergy'] - df['cpdEnergy']
```

En la tabla 8 se pueden ver las estadísticas de las nuevas variables generadas, mientras que en la figura 19 se pueden ver graficados los datos de consumo energético de las nuevas variables generadas durante el mes de febrero de 2021.

Variable (kWh)	F. Inicial	F. Final	N.º de registros	Mínimo	Media	Máximo	Desviación estándar
cpdEnergy	1/10/2019	21/5/2021	14015	4.22	5.58	21.7	0.81
totalBuildingEnergy	1/10/2019	21/5/2021	14015	0.83	11.6	29.24	5.68
totalEnergy	1/10/2019	21/5/2021	14015	10	17.25	60	6.09

Tabla 8: Estadísticas de las nuevas variables generadas tras el proceso de transformación

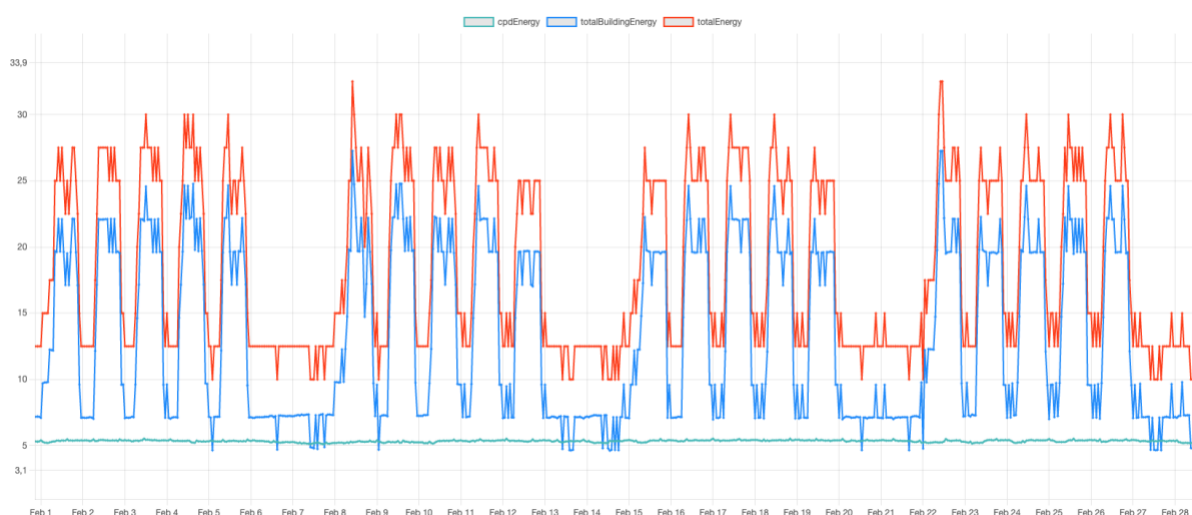


Figura 19: Consumo energético horario de las nuevas variables generadas (febrero 2021)

Los datos de las variables del área de producción y de condiciones externas han sido agrupados semanalmente utilizando el valor medio de los registros pertenecientes a cada semana.

5.3.1 Codificación del tiempo

Con el objetivo de tener en consideración el día de la semana y la hora del día en la que una observación ha tenido lugar, nuestra aproximación implica utilizar datos categóricos del campo fecha como insumos del modelo para la predicción de consumo energético.

Para poder utilizar estos nuevos datos como entrada en el modelo es necesario transformar el campo fecha de los datos de consumo energético. El método elegido para generar los nuevos datos ha sido *one-hot encoder*, que permite crear una nueva característica binaria para cada categoría existente asignando un 1 a la característica de la muestra que se corresponde con su categoría original.

En la figura 20 se puede visualizar el resultado de aplicar el método *one-hot* para codificar los días de la semana.

Día de la semana	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Dominago
Lunes	1	0	0	0	0	0	0
Martes	0	1	0	0	0	0	0
Miércoles	0	0	1	0	0	0	0
Jueves	0	0	0	1	0	0	0
Viernes	0	0	0	0	1	0	0
Sábado	0	0	0	0	0	1	0
Domingo	0	0	0	0	0	0	1

Figura 20: Codificación usando *one-hot* de los días de la semana

La codificación con las horas se realiza de la misma forma, pero con 24 categorías posibles, una por cada hora. Debido al aumento de dimensionalidad que supone el uso de esta técnica con la que se generan 24 nuevas variables para la codificación de horas y 7 nuevas variables para la codificación de días y a la cantidad no demasiado abundante de datos de la que disponemos (datos desde octubre 2019 a mayo 2021), no se ha procedido a la categorización de los meses del año que supondría añadir otras 12 variables más al conjunto de datos de entrada al modelo.

5.3.2 Normalización y estandarización de los datos

El objetivo de la normalización es transformar los datos de las diferentes variables a una escala común sin distorsionar las diferencias entre los rangos de valores. Con la normalización evitamos que variables con rangos de valores diferentes puedan influenciar en el resultado al aplicar distintas técnicas de aprendizaje automático.

Los datos de las variables del área de producción y de condiciones externas han sido normalizados a una escala de valores $[0, 1]$ mediante la aplicación de la función *MinMaxScaler*[\[34\]](#) de la biblioteca *Scikit-learn*.

Con la estandarización se ajusta la distribución de los datos a una distribución normal estándar mediante la aplicación de dos pasos:

- Se descentraliza la media para que tenga un valor de 0.
- Se escala la varianza para que tome como valor 1.

Los datos de consumo energético han sido transformados a una distribución normal utilizando el proceso de estandarización para que tuviesen una media de 0 y una varianza de 1.

5.4 Selección de características

El proceso de selección de características o *features* nos permite reducir la dimensionalidad de un conjunto de variables que se van a usar como entrada de un modelo de aprendizaje automático. Reducir la dimensionalidad del conjunto de variables va a permitir reducir el coste computacional del proceso de modelado, inclusive, en algunos casos va a suponer un aumento en el rendimiento del modelo final.

Los métodos de selección de características evalúan la relación existente entre las diferentes variables de un conjunto. Con estos métodos podemos ver qué variables son similares entre sí y nos ofrecen información que podríamos considerar redundante o irrelevante de cara a reducir el número de variables de un conjunto.

Para la selección de características del conjunto de variables pertenecientes al área de producción se ha usado el coeficiente de correlación de *Pearson*[\[35\]](#) como métrica para identificar la relación existente entre variables. El coeficiente de correlación de *Pearson* mide la dependencia lineal entre dos variables cuantitativas continuas. Este coeficiente puede tomar un valor dentro del rango $[-1, 1]$ y queda definido por la siguiente función:

$$r = \frac{\sigma_{xy}}{\sigma_x \cdot \sigma_y}$$

Si el valor de r es 0 nos indica que no existe correlación entre dos variables, si el valor de r es negativo existe una correlación negativa y de lo contrario una correlación positiva. Los valores de los extremos -1 y 1 representan una correlación negativa y positiva perfecta respectivamente.

En concreto para la selección de características de las variables del área de producción se ha generado una matriz de correlación donde quedan representados los coeficientes de correlación de Pearson entre cada una de las variables del conjunto, facilitando de esta forma la visualización de todas las relaciones. Una vez interpretada la matriz de correlación generada se ha decidido eliminar aquellas variables con un valor absoluto de correlación igual o superior a 0.9. Con la aplicación de este filtro se han eliminado un total de 23 variables de las 37 del área de producción, representando este número a un 62% de las variables y reduciéndose la dimensionalidad del conjunto considerablemente. En la tabla 9 se puede ver el listado de variables eliminadas junto con las variables del conjunto final que mayor relación tenían y el coeficiente de correlación existente entre ambas.

Variable eliminada	Variable con mayor relación	Coefficiente de correlación
Aprod'C'CtoP'Abs'TOutTwr	Aprod'C'CtoP'Abs'TInTwr	0.97
Aprod'C'CtoP'Abs'TRetRef	Aprod'C'CtoP'Abs'TImpRef	0.95
Aprod'C'CtoP'Abs'TRetTwr	Aprod'C'CtoP'Abs'TInTwr	0.92
Aprod'C'CtoP'Ch'TIn	Aprod'C'CtoP'Abs'TInTwr	0.93
Aprod'C'CtoP'Ch'TOut	Aprod'C'CtoP'Abs'TInTwr	0.99
Aprod'C'CtoP'CtoAbsh'TSu	Aprod'C'CtoP'CtoAbsh'TRet	0.92
Aprod'C'CtosS'CL'TSu	Aprod'C'CtosS'CL'TRet	0.95
Aprod'C'CtosS'FncP1'TRet	Aprod'C'CtosS'CL'TRet	0.95
Aprod'C'CtosS'FncP1'TSu	Aprod'C'CtosS'CL'TRet	0.90
Aprod'C'CtosS'FncP2'TSu	Aprod'C'CtosS'FncP2'TRet	0.96
Aprod'C'CtosS'FncPB'TRet	Aprod'C'CtosS'FncP2'TRet	0.91
Aprod'C'CtosS'FncPB'TSu	Aprod'C'CtosS'FncP2'TRet	0.90
Aprod'C'TColRet	Aprod'C'CtoP'Abs'TInTwr	0.93

Aprod`C`TColSu	Aprod`C`CtoP`Abs`TInTwr	0.95
Aprod`C`TDep	Aprod`C`CtoP`Abs`TInTwr	0.93
Aprod`H`CtosS`ACS`TAguFri	Aprod`C`CtoP`Abs`TInTwr	0.91
Aprod`H`CtosS`ACS`TRetSec	Aprod`H`CtosS`ACS`TRet	0.93
Aprod`H`CtosS`ACS`TSuSec	Aprod`H`CtosS`ACS`TRet	0.90
Aprod`H`CtosS`FncP2`TRet	Aprod`H`CtosS`FncP1`TRet	0.93
Aprod`H`CtosS`FncPB`TRet	Aprod`H`CtosS`FncP1`TRet	0.93
Aprod`H`CtosS`FncPB`TSu	Aprod`H`CtosS`FncP1`TSu	0.91
Aprod`H`CtosS`ACS`TSu	Aprod`H`CtosS`ACS`TRet	0.92
Aprod`H`CtosS`TColSu	Aprod`H`CtosS`TColret	0.91

Tabla 9: Variables eliminadas mediante la selección de características con la matriz de correlación de Pearson

El conjunto de datos queda con las variables que se muestran en la matriz de correlación que se puede ver en la figura 21, donde quedan representadas junto con los coeficientes de correlación de *Pearson*.

	AProd`C`CtoP`Abs`	AProd`C`CtoP`Abs`	AProd`C`CtoP`CtoA	AProd`C`CtosS`CL`	AProd`C`CtosS`FncI	AProd`H`CtosS`ACS	AProd`H`CtosS`ACS	AProd`H`CtosS`CI`T	AProd`H`CtosS`CI`T	AProd`H`CtosS`FncI	AProd`H`CtosS`FncI	AProd`H`CtosS`FncI	AProd`H`CtosS`TCo	AProd`H`TColBot
AProd`C`CtoP`Abs`TImpRef	1.00	-0.64	-0.54	0.75	0.75	-0.09	-0.20	0.25	0.44	0.17	0.48	0.32	-0.19	-0.15
AProd`C`CtoP`Abs`TInTwr	-0.64	1.00	0.58	-0.33	-0.20	-0.02	0.14	-0.45	-0.74	-0.44	-0.71	-0.61	-0.00	-0.07
AProd`C`CtoP`CtoAbsH`TRet	-0.54	0.58	1.00	-0.45	-0.38	0.35	0.60	0.27	-0.12	0.38	-0.08	-0.03	0.74	0.62
AProd`C`CtosS`CL`TRet	0.75	-0.33	-0.45	1.00	0.89	-0.08	-0.07	0.15	0.27	-0.01	0.22	0.12	-0.22	-0.28
AProd`C`CtosS`FncP2`TRet	0.75	-0.20	-0.38	0.89	1.00	-0.12	-0.09	0.06	0.16	-0.03	0.17	0.02	-0.22	-0.27
AProd`H`CtosS`ACS`TDep	-0.09	-0.02	0.35	-0.08	-0.12	1.00	0.76	0.50	0.39	0.38	0.22	0.10	0.44	0.51
AProd`H`CtosS`ACS`TRet	-0.20	0.14	0.60	-0.07	-0.09	0.76	1.00	0.60	0.38	0.54	0.22	0.12	0.71	0.59
AProd`H`CtosS`CI`TRet	0.25	-0.45	0.27	0.15	0.06	0.50	0.60	1.00	0.86	0.79	0.68	0.62	0.70	0.57
AProd`H`CtosS`CI`TSu	0.44	-0.74	-0.12	0.27	0.16	0.39	0.38	0.86	1.00	0.70	0.79	0.66	0.40	0.37
AProd`H`CtosS`FncP1`TRet	0.17	-0.44	0.38	-0.01	-0.03	0.38	0.54	0.79	0.70	1.00	0.84	0.74	0.81	0.76
AProd`H`CtosS`FncP1`TSu	0.48	-0.71	-0.08	0.22	0.17	0.22	0.22	0.68	0.79	0.84	1.00	0.82	0.40	0.41
AProd`H`CtosS`FncP2`TSu	0.32	-0.61	-0.03	0.12	0.02	0.10	0.12	0.62	0.66	0.74	0.82	1.00	0.43	0.38
AProd`H`CtosS`TColRet	-0.19	-0.00	0.74	-0.22	-0.22	0.44	0.71	0.70	0.40	0.81	0.40	0.43	1.00	0.84
AProd`H`TColBot	-0.15	-0.07	0.62	-0.28	-0.27	0.51	0.59	0.57	0.37	0.76	0.41	0.38	0.84	1.00

Figura 21: Matriz de correlación con las variables finales

Capítulo 6

6. *Clustering*

Suponemos que los resultados de los modelos de *clustering* usando los datos del área de producción van a ser similares a los que se pueden obtener usando datos meteorológicos, pero con la posibilidad de determinar nuevos estados asociados al nivel de ocupación del edificio. En el presente apartado se van a comparar los resultados de aplicar técnicas de *clustering* sobre los conjuntos de datos del área de producción y condiciones meteorológicas.

Las técnicas de aprendizaje no supervisado elegidas para obtener los clústeres que determinen los estados de funcionamiento del edificio han sido k-medias y *clustering* jerárquico aglomerativo. En total para cada conjunto de datos (área de producción y datos meteorológicos) se han realizado 12 experimentos diferentes (2 algoritmos x 6 tamaños de clúster).

6.1 Métricas de evaluación

Para la evaluación de los modelos se han empleado dos métricas comunes a los dos tipos de algoritmo utilizados. De esta forma, podemos comparar los resultados de ambos algoritmos para la selección del modelo óptimo:

- Índice Davies-Bouldin[36][37]: esta métrica evalúa la similitud promedio de cada clúster con su clúster más similar. El índice Davies-Bouldin queda definido por la siguiente fórmula:

$$DB(k) = \frac{1}{k} \sum_{i=1}^k \max_{j \leq k, j \neq i} D_{ij}$$

En la fórmula D_{ij} representa el ratio de similitud más alto con el clúster i . Para obtener este valor se emplea la siguiente fórmula, donde \bar{d}_i y \bar{d}_j representan la distancia

promedio entre las observaciones de un clúster y d_{ij} representa la distancia entre los clústeres i y j :

$$D_{ij} = \frac{(\bar{d}_i + \bar{d}_j)}{d_{ij}}$$

Al emplear el índice Davies-Bouldin los valores más bajos indican una mejor agrupación con clústeres más definidos, por este motivo buscamos minimizar el valor de esta métrica.

- Método de Silhouette[38][39]: este método permite identificar el número de clústeres óptimo empleando el coeficiente de Silhouette. El coeficiente de Silhouette queda definido como la diferencia entre la distancia media a las observaciones del clúster más cercano y la distancia media existente entre las observaciones pertenecientes al propio clúster dividida entre el máximo de los dos valores anteriores:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

Cuando usamos el método de Silhouette buscamos el número de clústeres o k para el cual se maximiza el valor del coeficiente de Silhouette, siendo 1 el valor más alto que puede alcanzar.

Además de las métricas comunes mencionadas, también se han usado otras técnicas de apoyo con el fin de seleccionar el número óptimo de clústeres, aunque estas últimas de forma individualizada para cada algoritmo. Para el algoritmo k-medias se ha empleado el método de Elbow[39] que calcula la distorsión promedio en los clústeres. Esta distorsión se calcula como la distancia media que hay entre cada observación perteneciente a un clúster y su centroide. Para el algoritmo de *clustering* jerárquico se ha empleado un dendograma que permite ver la distancia entre cada par de observaciones o subgrupos fusionados de manera secuencial. El dendograma de esta forma nos permite ver cómo se han ido fusionando las diferentes clases que se van formando al emplear el algoritmo de *clustering* jerárquico aglomerativo.

6.2 Modelos y resultados

Por una parte, se van a mostrar los resultados de las diferentes ejecuciones realizadas para cada conjunto de datos y posteriormente se van a comparar las agrupaciones generadas con los mejores modelos generados. La finalidad de la comparación es validar el uso de los datos del área de producción para la estimación de los estados de funcionamiento del edificio.

Los resultados obtenidos para el conjunto de datos del área de producción se pueden observar en la tabla 10 (el índice Davies-Bouldin se muestra con las siglas DB).

Algoritmo	Métrica	Tamaño de clúster					
		k = 2	k = 3	k = 4	k = 5	k = 6	k = 7
k-medias	Silhouette	0.38	0.41	0.28	0.25	0.24	0.25
	DB	1.19	1	1.19	1.24	1.26	1.25
Clustering jerárquico	Silhouette	0.34	0.39	0.34	0.35	0.23	0.23
	DB	1.27	1.14	1.2	1.06	1.3	1.24

Tabla 10: Rendimiento de los modelos de clustering sobre los datos del área de producción

Como se puede observar los mejores resultados se obtienen con un valor de $k=3$ para ambos algoritmos de *clustering*. Este hecho se puede visualizar además de en la tabla con las métricas en el dendrograma de la figura 22, utilizado para ayudar en la selección del número óptimo de clústeres para el algoritmo de *clustering* jerárquico.

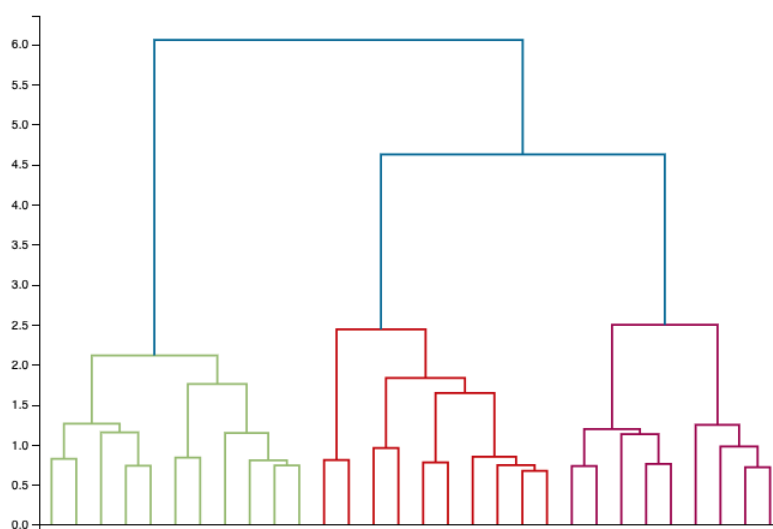


Figura 22: Dendrograma resultante al aplicar clustering jerárquico sobre los datos del área de producción

En la figura 23 podemos ver los conglomerados generados usando las variables del área de producción que componen el conjunto de datos final obtenido en el apartado 5.4 tras realizar la selección de características y reducir su dimensionalidad.

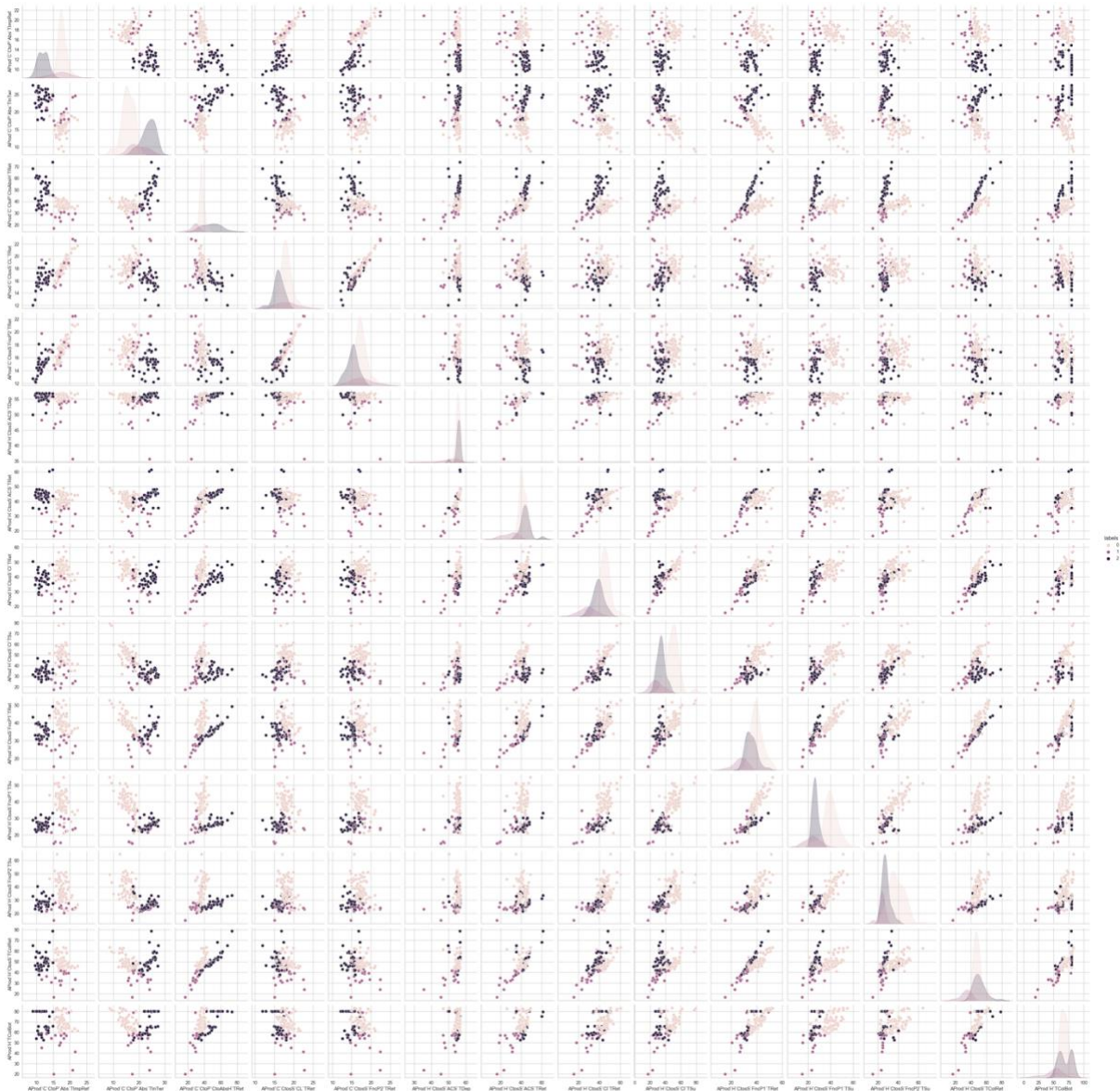


Figura 23: Grupos generados usando k-medias y 3 clústeres para las variables del área de producción

A pesar de que con el uso de ambos algoritmos el número de clústeres con mejores resultados es el mismo, el algoritmo k-medias obtienen mejores resultados tanto para el índice Davies-Bouldin como para el coeficiente de Silhouette, motivo por el cual el modelo con mejor rendimiento es el de k-medias con 3 clústeres. En la figura 24 se puede ver cómo queda el etiquetado de las observaciones a lo largo del tiempo. En esta figura vemos dos asignaciones que predominan, las del grupo que pertenecen a la etiqueta “0”, que se corresponden con los meses más fríos donde predomina el uso de la producción de calor y la etiqueta “1” que se corresponde con meses de más calor y donde predomina la producción de frío. Además, tenemos un tercer clúster con la etiqueta “2” que podría corresponderse con periodos con condiciones climatológicas muy favorables para la habitabilidad en el edificio donde los

sistemas de producción casi no entrasen en funcionamiento, o bien con periodos de baja ocupación como se puede observar en algún momento del verano o incluso en las semanas de marzo de 2020 donde la situación de emergencia sanitaria por COVID-19 provocó una alta desocupación del edificio.

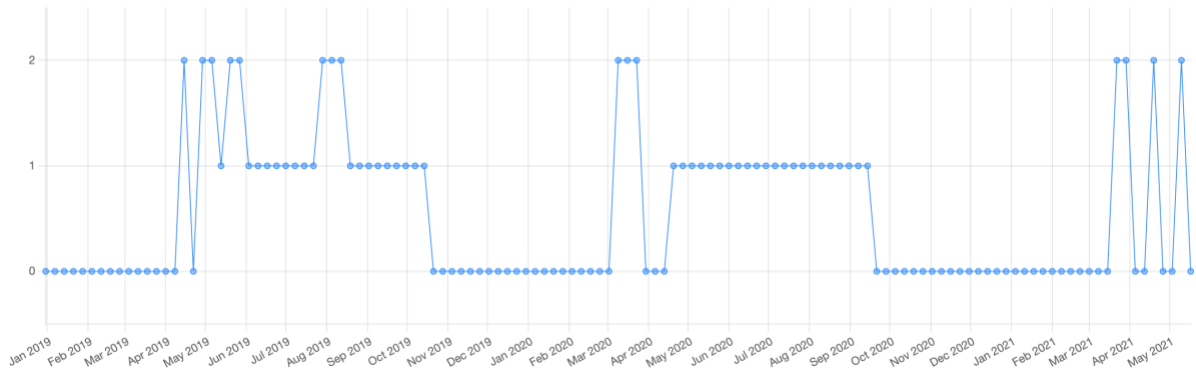


Figura 24: Etiquetado de las observaciones del área de producción usando *k*-medias y 3 clústeres

Los resultados obtenidos utilizando los datos de las dos variables de condiciones climatológicas se pueden observar en la tabla 11 (el índice Davies-Bouldin se muestra con las siglas DB).

Algoritmo	Métrica	Tamaño de clúster					
		k = 2	k = 3	k = 4	k = 5	k = 6	k = 7
k-medias	Silhouette	0.52	0.47	0.42	0.4	0.41	0.41
	DB	0.67	0.73	0.88	0.96	0.9	0.81
Clustering jerárquico	Silhouette	0.46	0.44	0.4	0.37	0.37	0.4
	DB	0.7	0.71	0.91	0.98	0.92	0.82

Tabla 11: Rendimiento de los modelos de clustering sobre las variables de condiciones externas

Los mejores resultados en este caso se obtienen con dos clústeres para ambos algoritmos. En la figura 25 se muestra el dendrograma usado para ayudar en la selección del número óptimo de clústeres para el algoritmo de *clustering* jerárquico. Al igual que con los modelos del área de producción los mejores valores de rendimiento se han obtenido con el algoritmo k-medias.

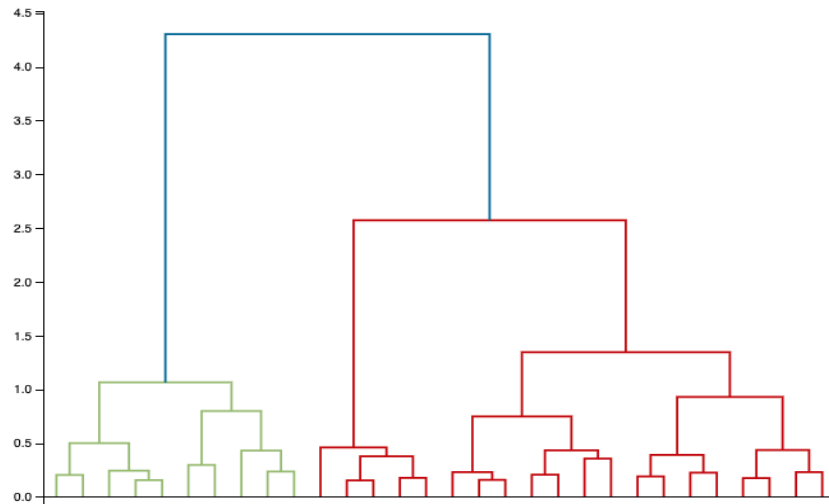


Figura 25: Dendrograma resultante al aplicar clustering jerárquico sobre las variables de condiciones externas

En la figura 26 podemos ver los conglomerados generados usando las variables de humedad relativa del aire y temperatura del aire que tenemos clasificadas como condiciones externas por el software DESIGO.

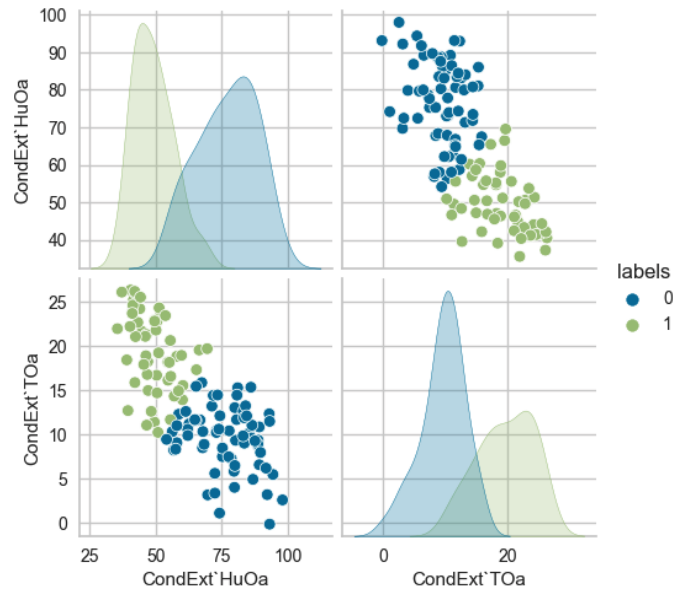


Figura 26: Grupos generados usando k-medias y 2 clústeres para las variables de condiciones externas

La figura 27 muestra el etiquetado de las observaciones en el tiempo, donde podemos ver como el grupo con la etiqueta “0” se corresponde con los periodos más fríos del año, mientras que la etiqueta “1” se corresponde con los periodos más cálidos. Esta clasificación a pesar de que sólo quedan identificados 2 clústeres representa en cierta medida el clima extremo que se

puede encontrar en la meseta castellana, con veranos muy cálidos, inviernos muy fríos y sin apenas influencia de las estaciones de primavera y otoño.

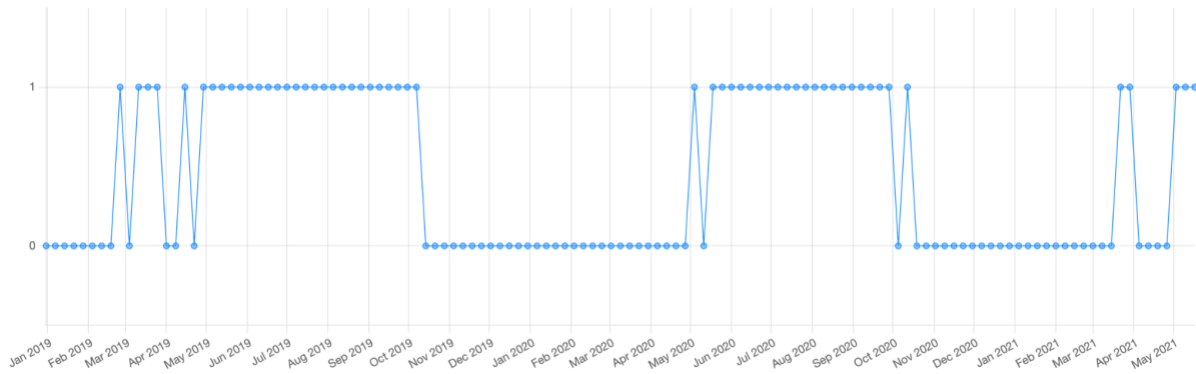


Figura 27: Etiquetado de las observaciones de condiciones externas usando *k-medias* y 2 clústeres

Una vez seleccionados los modelos con mejores métricas de rendimiento tanto para los datos del área de producción como para los datos de condiciones meteorológicas, podemos comparar los resultados del etiquetado. Como se puede observar la figura 28 el etiquetado de ambos conjuntos presenta bastantes similitudes, especialmente en los periodos del año que podríamos considerar más fríos y cálidos representados por los clústeres con las etiquetas 0 y 1 respectivamente.

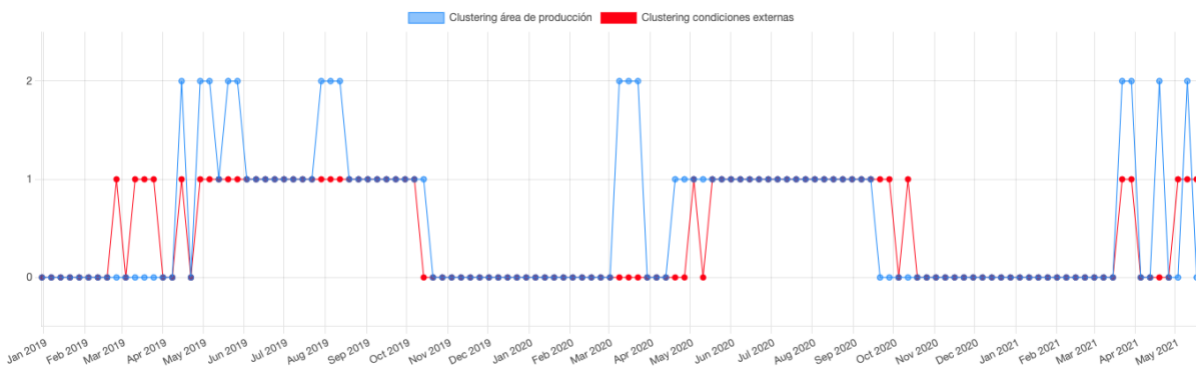


Figura 28: Etiquetado de las observaciones del área de producción ($k=3$) y de condiciones externas ($k=2$)

Actualmente el edificio LUCIA cuenta con 3 estados de funcionamiento operativos, activándose uno de ellos únicamente los fines de semana (no podemos identificarlo al aplicar *clustering* semanal), mientras que los otros dos modos se corresponden con los estados de funcionamiento identificados al aplicar *clustering* sobre las variables de condiciones externas. Las pocas semanas discordantes con los dos clústeres con mayor semejanza identificados para las variables del área de producción, hacen que los 3 clústeres identificados a partir de estas últimas variables puedan ser de interés para los administradores de los sistemas del edificio, ya

que representan un patrón claramente distinto, y no se pueden asociar exclusivamente a periodos vacacionales (como Semana Santa o el mes de agosto).

Nuevos datos futuros, ya sean de las variables del área de producción o de las variables de condiciones externas, se pueden clasificar en el clúster más cercano mediante el cálculo de su distancia al centroide de cada clúster. Una vez clasificados, para la predicción del consumo energético se utilizaría el modelo generado específicamente para ese clúster. En el siguiente capítulo se van a construir diferentes modelos para la predicción de consumo energético que nos permitirán identificar si los modelos de predicción para los estados de funcionamiento identificados para las variables de condiciones externas son mejores que los modelos para los estados de funcionamiento identificados para las variables del área de producción y viceversa.

Capítulo 7

7. Predicción del consumo energético

La segunda fase de este Trabajo Fin de Máster consiste en desarrollar modelos para la predicción del consumo energético del edificio LUCIA. Se van a desarrollar diversos tipos de modelo dependiendo del intervalo temporal a predecir (predicciones a corto, medio y largo plazo) y además para cada tipo de modelo se van a realizar diferentes propuestas. Se van a considerar los dos mejores modelos de clúster obtenidos, tanto usando las variables de condiciones externas, como con las variables del área de producción, para generar dos modelos predictivos para los estados de funcionamiento del edificio identificados a partir de cada conjunto de datos. Además, se van a construir otros dos modelos: un modelo con datos de consumo energético recientes y pertenecientes todos ellos a un mismo clúster y un modelo general con todos los datos de consumo del edificio LUCIA y que se usará como control para saber si a partir de modelos generados exclusivamente para uno de los estados de funcionamiento o con datos recientes podemos obtener estimaciones más precisas.

7.1 Método de predicción de consumo energético: redes LSTM

Las redes LSTM se han convertido en la arquitectura más popular para la predicción de series temporales y además debido al rendimiento que ofrecen han sido las elegidas para la construcción de los modelos de predicción de consumo energético. Para la implementación de los modelos de redes LSTM se ha empleado la biblioteca de Python Keras que proporciona una API de alto nivel para el uso de TensorFlow facilitando la implementación de redes neuronales artificiales.

Todos los modelos construidos en el presente proyecto tienen una misma arquitectura de capas LSTM apiladas como la que se muestra en la figura 29.

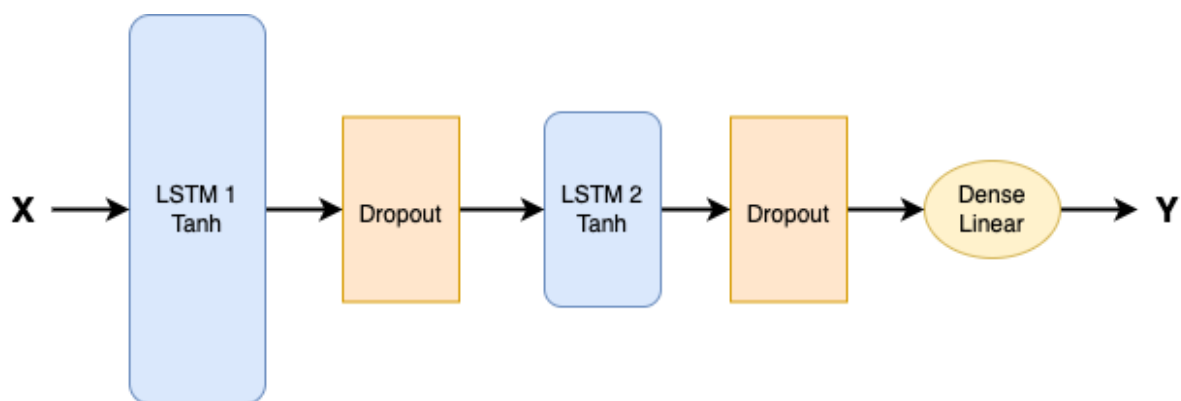


Figura 29: Arquitectura del modelo de capas LSTM apiladas

El modelo consta de las siguientes capas:

- Dos capas de red LSTM que utilizan como función de activación la tangente hiperbólica.
- Dos capas de *Dropout* o de abandono como técnica para reducir el sobreajuste en el modelo. Esta técnica implica la desactivación de un porcentaje de las neuronas de la siguiente capa de entrada a la capa *Dropout* en función del valor de un factor de olvido asignado. El factor de olvido puede tomar valores dentro del rango $[0, 1]$ representando un porcentaje. Con el empleo de estas capas se consigue que la red no tenga dependencia de ciertas neuronas evitando el sobreajuste.
- Una capa *Dense* (densa) o también denominada profundamente conectada ya que las neuronas de esta capa están conectadas con cada neurona de la capa anterior. Estas capas permiten modificar los formatos y dimensiones de los datos de entrada en función del número de unidades (definen el tamaño del vector de salida) que conforman la propia capa.

Las redes LSTM en el presente trabajo se han empleado con dos propósitos dependiendo del espacio temporal a predecir:

- **Predicción de un solo paso en el tiempo:** el objetivo de estos modelos es predecir el siguiente paso en el tiempo a partir de una secuencia de valores que en nuestro caso está compuesta por los datos de consumo energético y la codificación de las fechas en instantes previos. De esta forma queremos predecir el valor de y_t a partir de:

$$x = [[y_{t-1}, y_{t-2}, \dots, y_{t-k}], [d_t, d_{t-1}, \dots, d_{t-k+1}], [h_t, h_{t-1}, \dots, h_{t-k+1}]]$$

Donde x es el vector de entrada al modelo, $[y_{t-1}, y_{t-2}, \dots, y_{t-k}]$ representa los valores de consumo energético en los instantes previos, $[d_t, d_{t-1}, \dots, d_{t-k+1}]$ representa el valor categórico de los días y $[h_t, h_{t-1}, \dots, h_{t-k+1}]$ representa el valor categórico de las horas.

Para todos los modelos construidos se han utilizado las últimas 24 observaciones para predecir el siguiente paso en el tiempo, es decir, la siguiente hora en nuestro caso. El vector de entrada al modelo ha tenido una forma de 24 x 32, donde 24 son los pasos previos en el tiempo usados, y 32 el número de variables (31 variables pertenecientes al tiempo en días y horas codificado mediante *one-hot*, más el valor de consumo energético en el instante previo). La salida va a ser un único valor con la predicción realizada, por lo que la capa *dense* de la arquitectura utilizada va a tener una única unidad.

- **Predicción de múltiples pasos en el tiempo:** el objetivo de estos modelos es predecir múltiples pasos en el tiempo (k). Para ello el modelo toma una secuencia de datos como entrada y produce una nueva secuencia de salida con los valores de las predicciones. De esta forma queremos predecir:

$$y = [y_t, y_{t+1}, \dots, y_{t+k}]$$

Utilizando para ello:

$$x = [[y_{t-1}, y_{t-2}, \dots, y_{t-k}], [d_t, d_{t-1}, \dots, d_{t-k+1}], [h_t, h_{t-1}, \dots, h_{t-k+1}]]$$

Donde y es el vector de salida con las predicciones y x es el vector de entrada al modelo. En el vector de entrada $[y_{t-1}, y_{t-2}, \dots, y_{t-k}]$ representa los valores de consumo energético en los instantes previos, $[d_t, d_{t-1}, \dots, d_{t-k+1}]$ representa el valor categórico de los días y $[h_t, h_{t-1}, \dots, h_{t-k+1}]$ representa el valor categórico de las horas.

Las redes LSTM al igual que otras redes neuronales son capaces de generar un vector de salida dado un vector de entrada mediante el mapeo de los valores[40]. Por este motivo la arquitectura presentada en el punto 7.1 se puede emplear tanto para el problema de predicción de un solo paso en el tiempo, como para el presente problema de predicción de múltiples pasos en el tiempo usando una secuencia de entrada, también denominado *Seq2Seq*. En algunos de los trabajos analizados en el apartado 4.2 se utilizan estructuras más complejas y especializadas para la predicción de secuencias como la arquitectura *encoder-decoder*[40], aunque en esta ocasión haremos uso de la capacidad de las redes LSTM para devolver un vector como salida representando una secuencia de predicciones realizadas y ver el rendimiento que ofrecen ante el problema planteado.

Se han realizado dos tipos de predicción de múltiples pasos en el tiempo:

- Predicción a 48 horas: se han utilizado las últimas 48 observaciones para predecir los siguientes 48 pasos en el tiempo, es decir, 48 horas o dos días a futuro. El vector de entrada ha tenido una forma de 48 x 32, donde 48 son los

pasos previos en el tiempo usados y 32 el número de variables (31 variables pertenecientes al tiempo en días y horas codificado mediante *one-hot*, más el valor de consumo energético en el instante previo). El vector de salida está compuesto por 48 valores con la predicción de cada uno de los siguientes 48 pasos en el tiempo, lo cual implica que el número de unidades de la última capa *dense* de la arquitectura de modelo empleada va a ser 48.

- Predicción a 168 horas: se han utilizado las últimas 168 observaciones para predecir los siguientes 168 pasos en el tiempo, es decir, se predicen los valores de una semana completa o 7 días a futuro. El vector de entrada ha tenido una forma de 168×32 , donde 168 son los pasos previos en el tiempo usados y 32 el número de variables (mismas que para la predicción a 48 horas). El vector de salida está compuesto por 168 valores con las predicciones de los diferentes pasos en el tiempo a futuro. Esto implica que el número de unidades de la última capa *dense* de la arquitectura de modelo empleada va a ser 168.

7.2 Conjuntos de entrenamiento y test

Después de haber identificado los estados de funcionamiento del edificio LUCIA mediante la aplicación de técnicas de *clustering*, hemos podido seleccionar los conjuntos de datos para la construcción de dos modelos “clúster” para los estados de funcionamiento identificados empleando datos del área de producción y condiciones externas y del modelo “clúster” con datos recientes. El modelo “general” utilizará todos los datos de consumo energético disponibles, ya que lo usaremos como control para contrastar nuestra hipótesis.

Para construir los modelos sobre clústeres identificados empleando los datos del área de producción y los datos de condiciones externas, se han seleccionado dos conjuntos representativos con las observaciones de consumo energético pertenecientes a las fechas de un clúster identificado para cada conjunto de datos. De esta forma el modelo que denominaremos “clúster área de producción” utilizará las observaciones con la etiqueta 0 de la figura 30 y que pertenezcan a los datos del área de producción, mientras que el modelo “clúster condiciones externas” va a emplear las observaciones con la etiqueta 0 de la figura 30 pero que pertenezcan a los datos de condiciones externas. Finalmente, para el modelo “clúster” con datos recientes y que denominaremos “clúster reciente”, se van a emplear las observaciones comprendidas entre el 18/1/2021 y el 15/03/2021. En la figura 30 podemos ver las fechas elegidas para el modelo “clúster reciente” y como todas pertenecen a los clústeres con la etiqueta “0” tanto para los datos del área de producción como de condiciones externas.



Figura 30: Fechas elegidas para los datos del modelo de clúster reciente

Para poder comparar los resultados de todos los modelos los datos pertenecientes al conjunto de test son los mismos en todos los casos. A continuación, se muestran los conjuntos de entrenamiento que utilizará cada modelo y el conjunto de test común empleado para comparar sus rendimientos:

- Modelo “general”:
 - Entrenamiento: 01/10-2019 - 01/03/2021 (el conjunto de datos se puede ver en la figura 31).

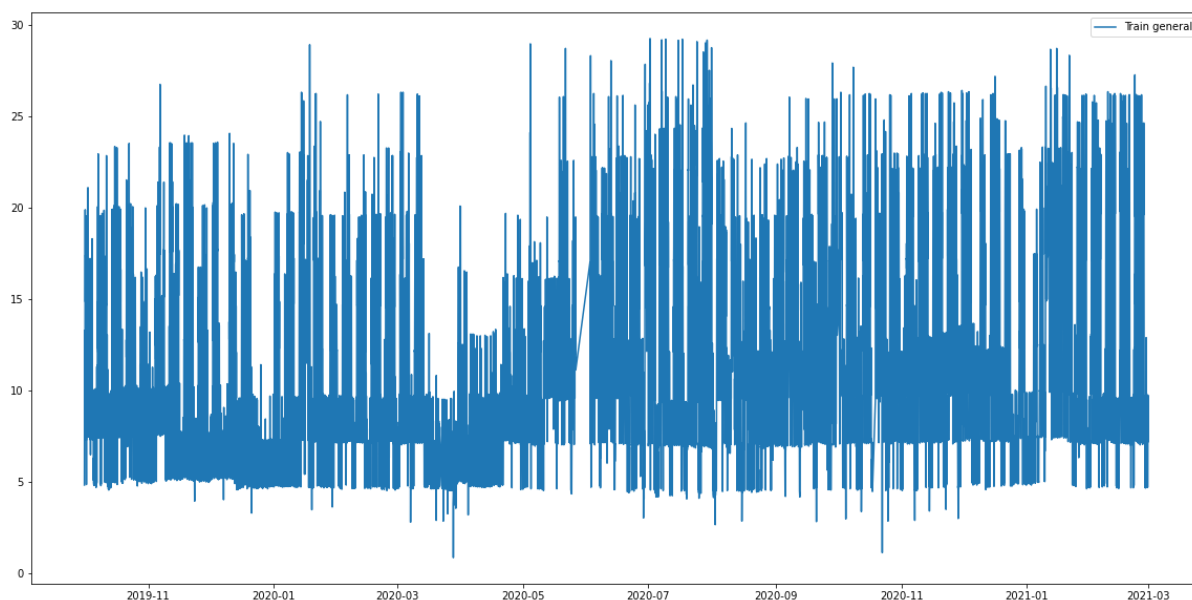


Figura 31: Conjunto de datos de entrenamiento del modelo “general”

- Modelo “clúster área de producción”:
 - Entrenamiento: 21/10/2019 - 02/03/2020, 30/03/2020 - 13/04/2020 y 21/09/2020 - 01/03/2021 (el conjunto de datos se puede ver en la figura 32).

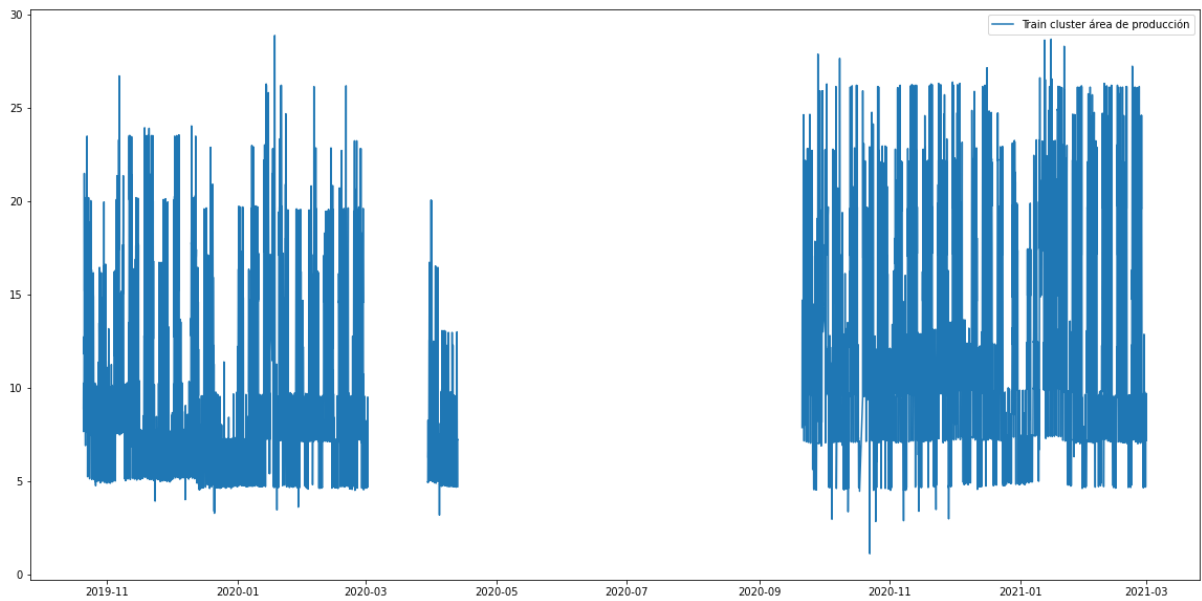


Figura 32: Conjunto de datos de entrenamiento del modelo “clúster área de producción”

- Modelo “clúster condiciones externas”:
 - Entrenamiento: 14/10/2019 - 27/04/2020, 11/05/2020 - 18/05/2020, 05/10/2020 - 12/10/2020 y 19/10/2020 – 01/03/2021 (el conjunto de datos se puede ver en la figura 33).

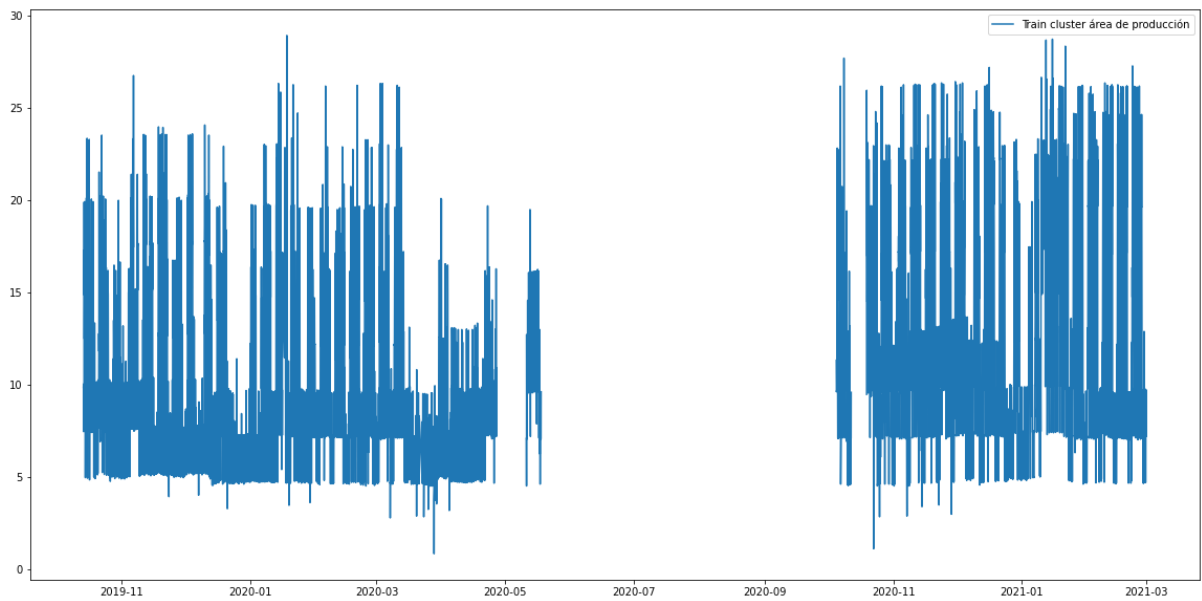


Figura 33: Conjunto de datos de entrenamiento del modelo “clúster condiciones externas”

- Modelo “clúster reciente”:
 - Entrenamiento: 18/01-2021 - 01/03/2021 (el conjunto de datos se puede ver en azul en la figura 34).
 - Todos los modelos generados se han comparado para el mismo conjunto de test: 01/03/2021 - 15/03/2021 (el conjunto de datos se puede ver en naranja en la figura 34).

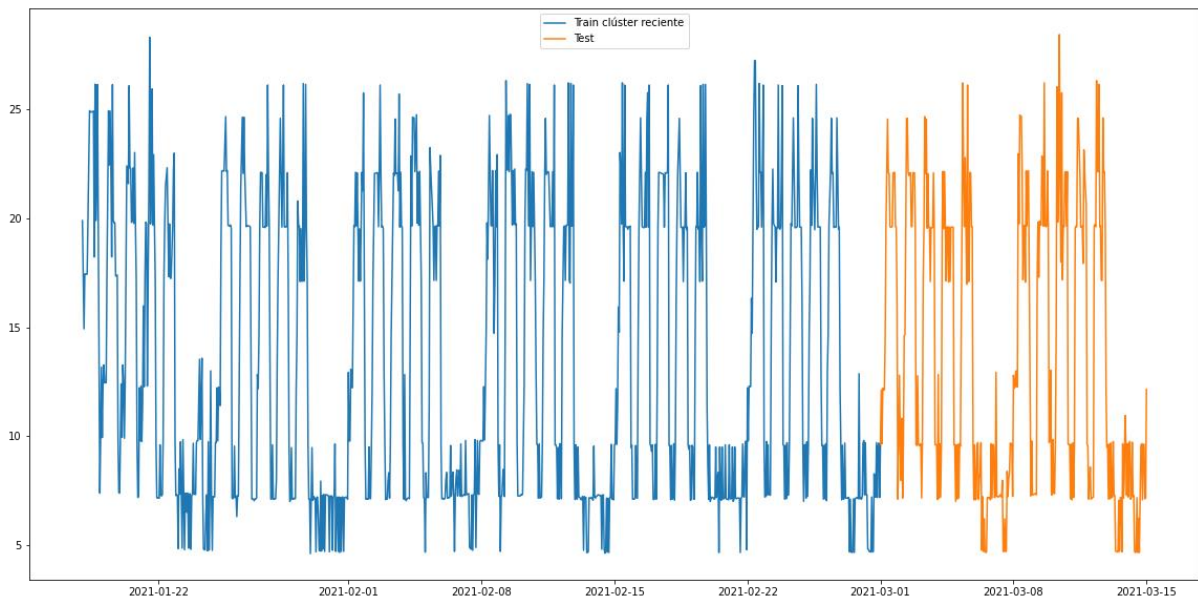


Figura 34: División del conjunto de datos usados para el modelo de “clúster reciente” en entrenamiento y test

7.3 Entrenamiento y parametrización de los modelos

Una vez determinados la arquitectura del modelo y los conjuntos de entrenamiento y test, empieza el proceso de entrenamiento. En el proceso de entrenamiento hay una serie de parámetros, hiperparámetros, funciones y consideraciones que se deben tener en cuenta a la hora de configurar los modelos:

- Función de pérdida (*loss function*): cuantifica el error existente entre las predicciones realizadas y los valores esperados en base a una función objetivo elegida. De esta forma la función de pérdida nos indica el rendimiento del modelo ante el problema planteado. Para los modelos generados la función de pérdida elegida ha sido MSE (*mean squared error*).

- Optimizador (*optimizer*): algoritmo usado para minimizar el valor de la función de pérdida y maximizar la eficiencia del modelo. Los optimizadores se encargan de actualizar los parámetros o pesos (*weights*) para minimizar el error existente entre los valores reales y las predicciones realizadas por el modelo (valor de la función de pérdida) en función de una tasa de aprendizaje indicada. Para los modelos generados hemos usado el algoritmo ADAM (*Adaptive Moment Estimation*) por su eficiencia y rápida convergencia
- Función de activación de las capas LSTM: define la salida de un nodo para una entrada o conjunto de entradas. Los modelos generados han usado la tangente hiperbólica (*tanh*) como función de activación.
- Validación del modelo: para validar el modelo hemos usado el 20% de los datos de entrenamiento. Gracias al uso de este porcentaje de los datos de entrenamiento para la validación de los modelos se ha podido comprobar su estado de ajuste mediante la comparación de los valores de la función de pérdida para los datos de entrenamiento y validación.
- Número de neuronas: representan el número de unidades que van a componer las capas LSTM de la arquitectura de dos capas LSTM apiladas utilizada para los diferentes modelos generados.
- Tamaño de lote (*batch size*): hace referencia al número de observaciones que el modelo procesa antes de cada actualización de los pesos y parámetros para minimizar el valor de la función de pérdida.
- Épocas (*epochs*): son ciclos de propagación hacia atrás y hacia adelante con la finalidad de reducir el valor de la función de pérdida. En cada ciclo se determinan los mejores pesos de cara a minimizar el error del modelo, así pues, se trata de un proceso iterativo que necesita de múltiples épocas. El número de épocas se corresponde con el número de veces que el modelo procesa el conjunto de entrenamiento completo.
- Valor del factor de olvido (*dropout*): representa el porcentaje de valores de salida de las capas de *dropout* respecto de la cantidad de valores de entrada.

Para determinar qué parametrización se adapta mejor a cada problema obteniendo un mejor rendimiento, se ha generado un conjunto de ejecuciones con la combinación de distintos valores para los hiperparámetros tamaño de lote, número de neuronas de las capas LSTM y valor del factor del olvido de la capa *Dropout* mediante la aplicación del siguiente método:

```

#Conjunto de ejecuciones
def models_set_up(self):
    setUps = []
    dropouts = [0.1, 0.2, 0.4, 0.5, 0.6]
    neurons = [[16, 8], [32, 16], [64, 32], [128, 64]]
    batchSize = [4, 8, 16, 32, 64, 128]
    epochs = [20, 50]
    for b in batchSize:
        for d in dropouts:
            for n in neurons:
                for e in epochs:
                    #Copia de los parámetros generales del modelo
                    modelCopy = self.params.copy()
                    modelCopy['dropout'] = d
                    modelCopy['numNeurons'] = n
                    modelCopy['batch_size'] = b
                    modelCopy['epochs'] = e
                    setUps.append(modelCopy)

    return setUps

```

En total se generan 240 ejecuciones con diferentes combinaciones de hiperparámetros para cada tipo de predicción y tipo de modelo (“clúster” o “general”). Además, para la selección del número óptimo de épocas evitando que los modelos no queden correctamente ajustados, se ha empleado una gráfica con los valores de la función de pérdida sobre el conjunto de entrenamiento (*loss*) y los datos empleados para validar los modelos (*val_loss*)[\[41\]](#). A partir de esta gráfica se ha buscado el número óptimo de épocas para los mejores modelos seleccionados, buscando el punto en el que los valores *loss* y *val_loss* estuviesen más próximos.

En la figura 35 podemos ver un ejemplo de una de las ejecuciones realizadas durante el proceso de entrenamiento. En concreto el ejemplo representado se corresponde con un buen ajuste de modelo, aunque en el caso de aumentarse el número de épocas y como se puede ver con la última ejecución, el valor de *val_loss* podría tender a aumentar superando el valor de *loss* y provocando el sobreajuste del modelo. En el caso en el que los valores de *loss* y *val_loss* se encontrasen muy distantes, estaríamos hablando de un modelo poco ajustado.

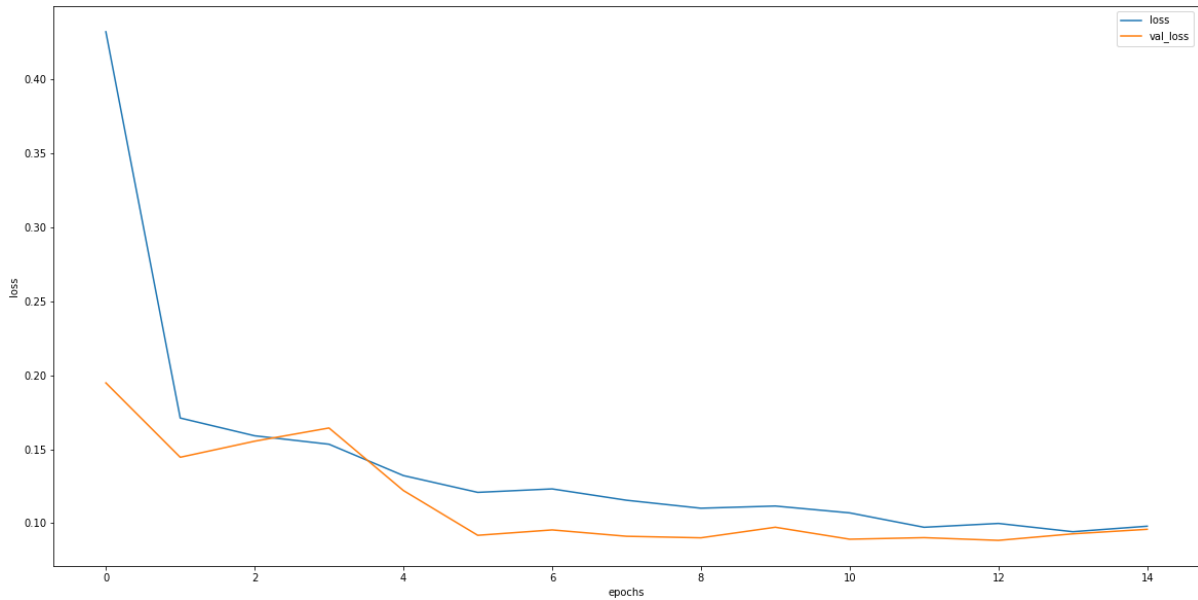


Figura 35: Valor de la función de pérdida para los datos de entrenamiento y validación

7.4 Métricas de evaluación

A continuación, se van a mostrar las dos métricas usadas para la evaluación de los modelos generados:

- RMSE (raíz del error cuadrático medio): se trata de la métrica más popular usada en problemas de regresión y predicción de series temporales. El RMSE mide el error promedio existente entre dos conjuntos de datos. En el presente caso se va a comparar el conjunto de datos correspondiente a las predicciones realizadas con los modelos y los datos reales existentes para las fechas de las predicciones obtenidas.

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (predicted_i - actual_i)^2}{N}}$$

- CV(RMSE) (coeficiente de variación de la raíz del error cuadrático medio): mide la variación media del RMSE respecto del valor medio de los valores reales y se utilizará para calcular la precisión de los modelos predictivos como se sugiere en los estándares internacionales ASHRAE[42]. El valor del CV(RMSE) al multiplicarlo por 100 representa el porcentaje de desviación del RMSE.

$$CV(RMSE) = \frac{1}{actual} \sqrt{\frac{\sum_{i=1}^N (predicted_i - actual_i)^2}{N}}$$

En la predicción y simulación de consumo energético en edificios, el RMSE y el CV(RMSE) son las métricas más usadas a la hora de calibrar los modelos para medir el rendimiento del edificio.

7.5 Modelos de predicción a corto plazo

Los mejores resultados para la predicción de un solo paso en el tiempo, 1 hora en nuestro caso, se han obtenido con el modelo “clúster reciente” de predicción y empleando una configuración con 64 y 32 neuronas para las capas de LSTM, 4 de tamaño de lote, 0.1 para las capas de *dropout* y 25 épocas. El modelo “general” ha obtenido mejores resultados que los modelos “clúster condiciones externas” y “clúster área de producción”. En la tabla 12 se pueden ver los mejores resultados obtenidos junto con las configuraciones de hiperparámetros empleadas para cada tipo de modelo.

Modelo	Épocas	Tamaño de lote	N.º neuronas	Dropout	RMSE	CV(RMSE)	Tiempo de ejecución
Clúster AP	30	64	[64, 32]	0.4	2.5756	0.1861	60.9041 seg
Clúster CE	22	64	[128, 64]	0.2	2.5432	0.1837	74.2977 seg
Reciente	25	4	[64, 32]	0.1	2.0605	0.1489	41.4440 seg
General	20	32	[64, 32]	0.4	2.3287	0.1682	98.5398 seg

Tabla 12: Resultados de los modelos para la predicción a 1 hora

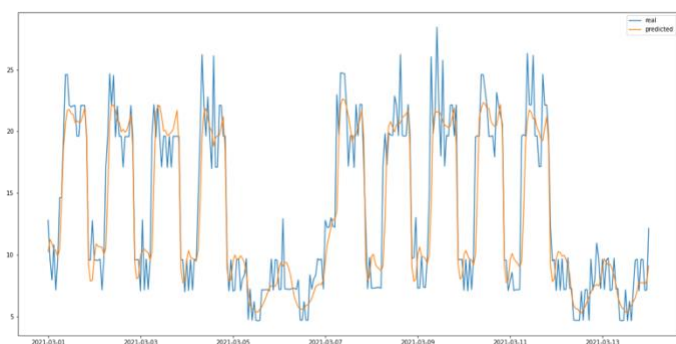


Figura 36: Predicción a 1 hora usando el mejor modelo "clúster área de producción"

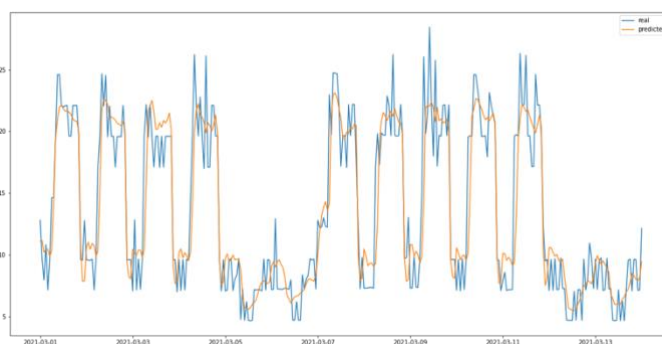


Figura 37: Predicción a 1 hora usando el mejor modelo "clúster condiciones externas"

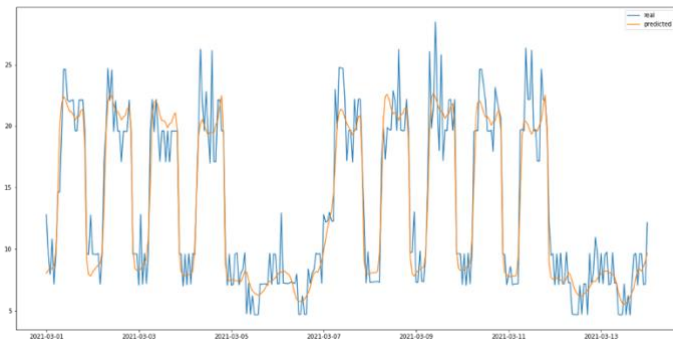


Figura 38: Predicción a 1 hora usando el mejor modelo "clúster reciente"

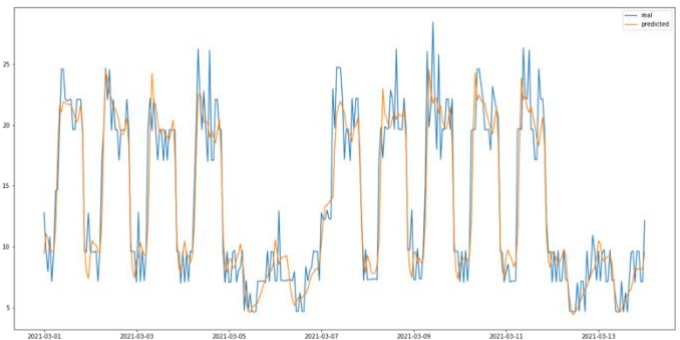


Figura 39: Predicción a 1 hora usando el mejor modelo "general"

Como se puede ver en las figuras 36, 37, 38 y 39 los modelos capturan el comportamiento general del consumo energético del edificio LUCIA, pero no son capaces de capturar los picos de carga energética. El consumo energético del edificio tiene una tendencia bastante definida con un consumo energético mayor entre semana y en concreto durante el periodo de ocupación del edificio, mientras que el consumo energético baja los fines de semana y por las noches. A pesar de tener un patrón definido, los pequeños picos y bajadas de carga que podemos ver en las gráficas no siguen un patrón definido y los modelos no son capaces de capturarlos.

7.6 Modelos de predicción a medio plazo

Los mejores resultados para la predicción a 48 horas se han obtenido con el modelo "clúster reciente" del mismo modo que ocurría con la predicción a corto plazo y empleando una configuración con 64 y 32 neuronas para las capas de LSTM, 4 de tamaño de lote, 0.2 para las capas de *dropout* y 15 épocas. En esta ocasión el modelo "clúster condiciones externas" ha obtenido unos resultados bastante cercanos a los del modelo "clúster reciente" y bastantes mejores al modelo "general" y modelo "clúster área de producción", que al igual que con la predicción a corto plazo ha obtenido los peores resultados. En la tabla 13 se pueden ver los mejores resultados obtenidos para cada tipo de modelo junto con las configuraciones de hiperparámetros empleadas.

Modelo	Épocas	Tamaño de lote	N.º neuronas	Dropout	RMSE	CV(RMSE)	Tiempo de ejecución
Clúster AP	50	8	[32, 16]	0.1	3.3551	0.2424	907.8494 seg
Clúster CE	17	4	[64, 32]	0.1	2.9325	0.2119	549.8763 seg
Reciente	15	4	[64, 32]	0.2	2.8675	0.2072	60.0385 seg
General	30	4	[32, 16]	0.2	3.2772	0.2368	1016.8095 seg

Tabla 13: Resultados de los modelos para la predicción a 48 horas

En las figuras 40 y 41 podemos ver la predicción de las siguientes 48 horas a los días 28/02/2021 y 05/03/2021 con el mejor modelo “clúster área de producción”.

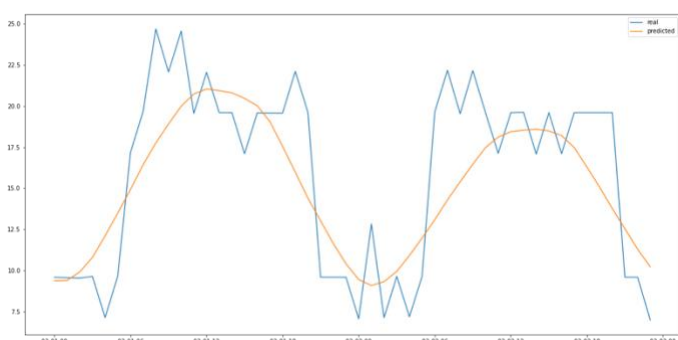


Figura 40: Predicción de las siguientes 48 horas al 28/02/2021 usando el mejor modelo “clúster área de producción”

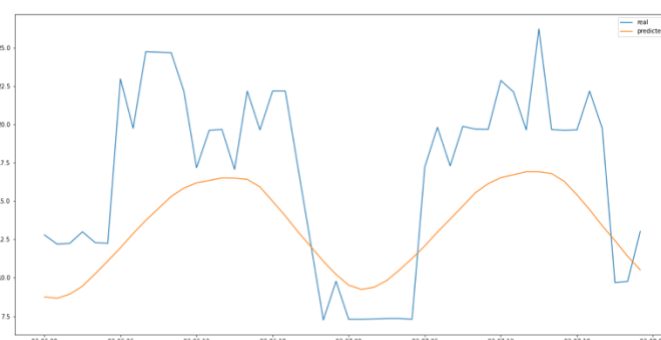


Figura 41: Predicción de las siguientes 48 horas al 05/03/2021 usando el mejor modelo “clúster área de producción”

En las figuras 42 y 43 podemos ver la predicción de las siguientes 48 horas para los mismos dos días que con el modelo “clúster área de producción” pero empleando para ello el modelo “clúster condiciones externas”.

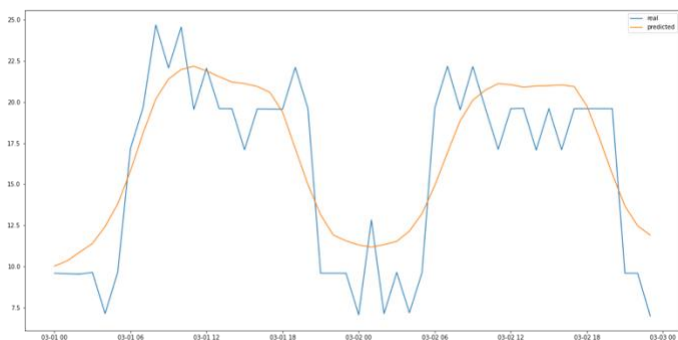


Figura 42: Predicción de las siguientes 48 horas al 28/02/2021 usando el mejor modelo “clúster condiciones externas”

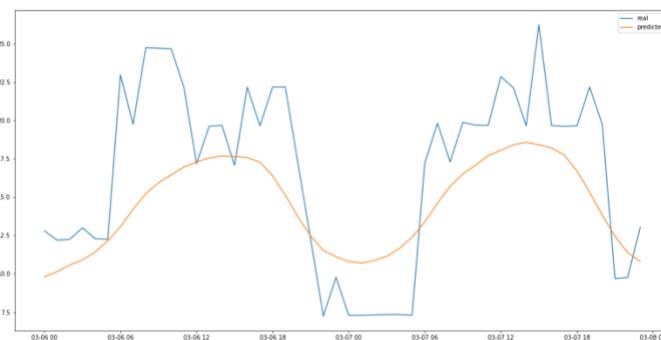


Figura 43: Predicción de las siguientes 48 horas al 05/03/2021 usando el mejor modelo “clúster condiciones externas”

La predicción de las siguientes 48 horas a los días 28/02/2021 y 05/03/2021 con el mejor modelo “clúster reciente” se puede visualizar en las figuras 44 y 45.

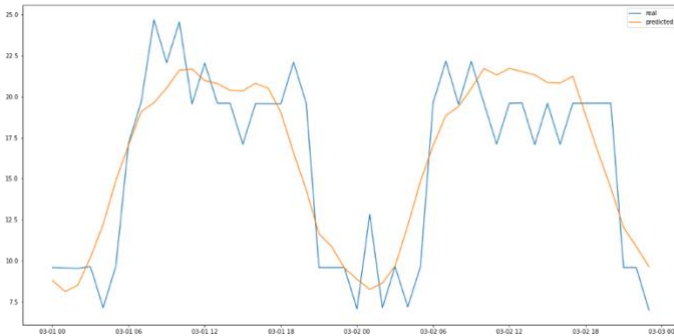


Figura 44: Predicción de las siguientes 48 horas al 28/02/2021 usando el mejor modelo “clúster reciente”



Figura 45: Predicción de las siguientes 48 horas al 05/03/2021 usando el mejor modelo “clúster reciente”

Las figuras 46 y 47 muestran la predicción de las siguientes 48 horas a los días 28/02/2021 y 05/03/2021 con el mejor modelo “general”.

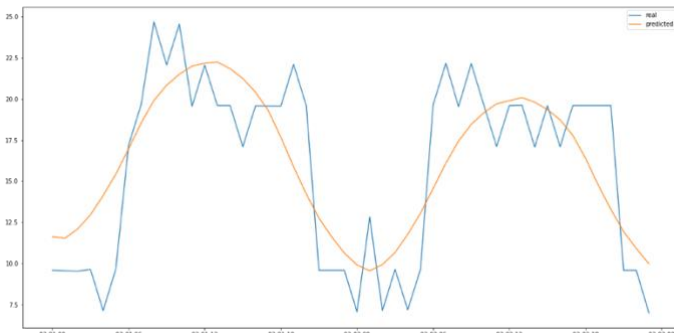


Figura 46: Predicción de las siguientes 48 horas al 28/02/2021 usando el mejor modelo “general”

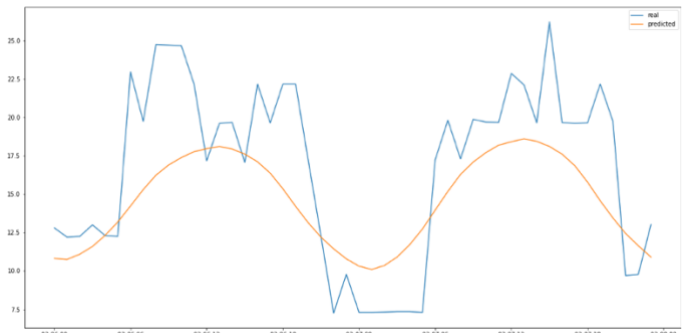


Figura 47: Predicción de las siguientes 48 horas al 05/03/2021 usando el mejor modelo “general”

7.7 Modelos de predicción a largo plazo

Siguiendo la misma tendencia que con la predicción a corto y medio plazo, los mejores resultados para la predicción a 168 horas se han obtenido también con el modelo “clúster reciente” de predicción. En esta ocasión, además, la diferencia entre el rendimiento del resto de modelos respecto del modelo “clúster reciente” es bastante mayor que con la predicción a corto y medio plazo. El modelo “clúster condiciones externas” ha obtenido mejor rendimiento que los modelos “general” y “área de producción”, siendo este último el que peores métricas de rendimiento ha obtenido de igual forma que ha pasado con la predicción a corto y medio plazo.

En la tabla 14 se pueden ver los mejores resultados obtenidos junto con las configuraciones de hiperparámetros empleadas para cada tipo de modelo.

Modelo	Épocas	Tamaño de lote	N.º neuronas	Dropout	RMSE	CV(RMSE)	Tiempo de ejecución
Clúster AP	30	8	[128, 64]	0.1	4.2772	0.3087	1103.3871 seg
Clúster CE	22	4	[64, 32]	0.1	3.8978	0.2816	2818.3017 seg
Reciente	20	4	[128, 64]	0.1	2.9584	0.2137	218.3965 seg
General	18	4	[128, 64]	0.1	4.0221	0.2906	3877.2945 seg

Tabla 14: Resultados de los modelos para la predicción a 168 horas

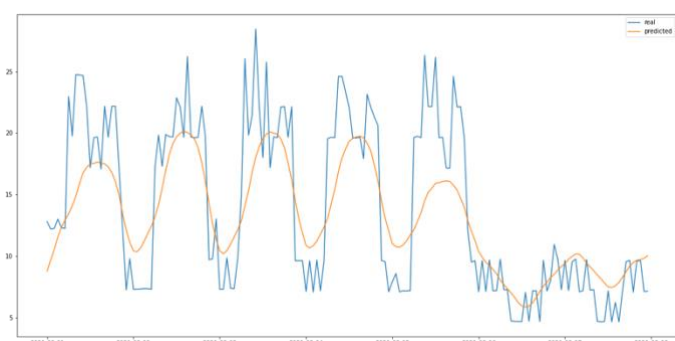


Figura 48: Predicción a 168 horas usando el mejor modelo “clúster área de producción”

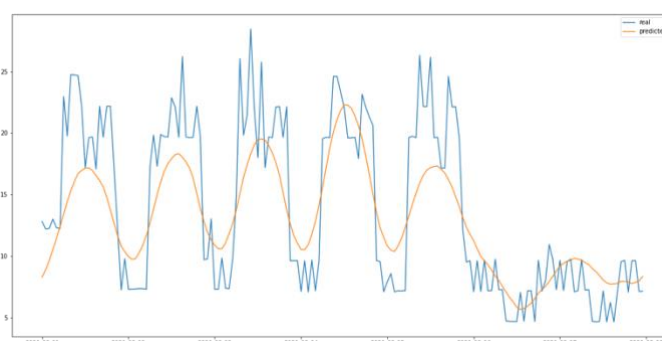


Figura 49: Predicción a 168 horas usando el mejor modelo “clúster condiciones externas”

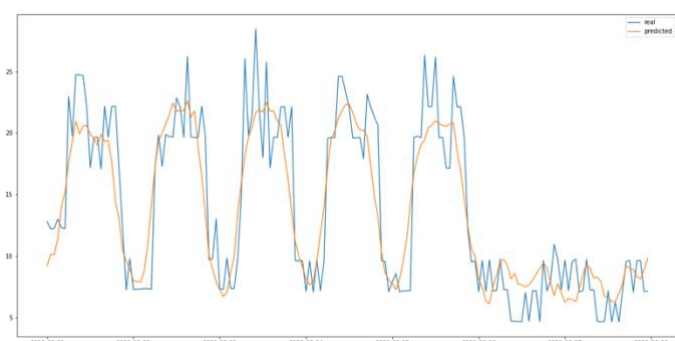


Figura 50: Predicción a 168 horas usando el mejor modelo “clúster reciente”

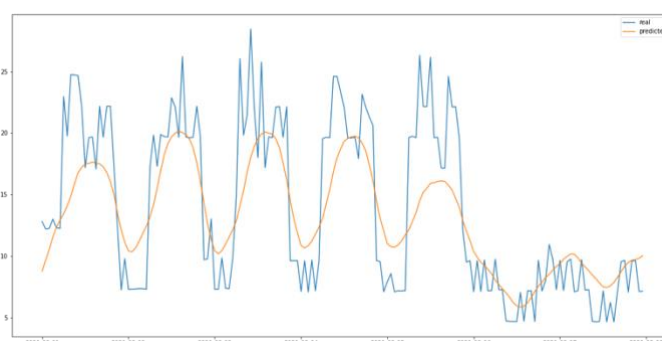


Figura 51: Predicción a 168 horas usando el mejor modelo “general”

Al igual que pasaba con los modelos para la predicción de un solo paso en el tiempo (predicción a 1 hora), los modelos de predicción de múltiples pasos en el tiempo (predicción a 48 y 168 horas) son capaces de capturar la tendencia del consumo energético del edificio, pero

su rendimiento se ve igualmente influenciado por los pequeños picos y bajadas de carga existentes y que no son capaces de capturar.

7.8 Discusión de los resultados

La propuesta presentada en el presente documento no se puede comparar con otros modelos puesto que son los primeros generados para el edificio LUCIA. No obstante, los resultados obtenidos en los puntos anteriores muestran cómo las aproximaciones vistas en los estudios analizados en punto [4.2](#), que abogaban por la determinación de los estados de funcionamiento del edificio y la construcción de modelos predictivos de consumo energético para cada estado de funcionamiento, son aplicables en otros edificios inteligentes como es el caso del edificio LUCIA.

En los resultados hemos podido comprobar como con el uso de los clústeres identificados sobre los datos de condiciones externas, los modelos predictivos obtienen un mejor rendimiento frente al uso de clústeres identificados con los datos del área de producción. Exceptuando el problema de predicción a corto plazo, el modelo “clúster condiciones externas” también obtiene mejores resultados que el modelo “general”, algo que no ocurre con el modelo “clúster área de producción”.

En los resultados el modelo “clúster reciente”, que hay que recordar que sería válido para ambos tipos de clúster, ha obtenido el mejor rendimiento para los diferentes tipos de predicción. No obstante, el modelo “clúster condiciones externas” al emplear más datos históricos pueden obtener mejor rendimiento en situaciones anómalas frente al modelo “clúster reciente” que en un principio va a ser más sensible a anomalías locales.

La primera conclusión que podemos sacar es que la mayor cantidad de datos del modelo “general” le permite hacer una buena estimación a corto plazo, pero tiene problemas para hacer predicciones a medio y largo plazo. La segunda conclusión es que el modelo “clúster condiciones externas” realiza mejores predicciones que el modelo “clúster área de producción” y también que el modelo “general”. En consecuencia, si queremos usar estos modelos para realizar estimaciones de funcionamiento o envejecimiento será mejor usar los modelos de “clúster” y “clúster reciente” relacionados con las condiciones externas. La visión del modelo “general” es menos efectiva en esos casos. Otra conclusión es que los consumos del área de producción, si bien pueden identificar zonas de funcionamiento nuevas donde los operadores del sistema pueden fijar nuevas consignas, no se corresponden con datos homogéneos que generen buenos modelos predictivos.

También podemos mencionar que a pesar de haber empleado una arquitectura de capas LSTM apiladas que resulta bastante simple, son comparables a los resultados de arquitecturas más complejas para otros conjuntos de datos. El uso de una arquitectura más compleja para la predicción de múltiples pasos en el tiempo, como puede ser la arquitectura *encoder-decoder* creada específicamente para este propósito, podría generar unas predicciones más precisas, aunque se presenta como un trabajo a futuro.

Finalmente, respecto a la hipótesis inicial del trabajo, relativo a que se pueden identificar modos de funcionamiento distinto en base a variables externas, este trabajo también lo confirma para el caso del edificio LUCIA. Aunque se identifican claramente 3 patrones de funcionamiento en base a los consumos del área de producción, los modelos que obtienen mejores resultados se basan en los dos patrones que genera el *clustering* usando las variables de condiciones externas. Esto reafirma la buena selección existente en el control del edificio LUCIA. Quedaría por comprobar si un análisis semanal, separando los días laborables de los fines de semana, y poniendo también a un lado las semanas de vacaciones, que tienen otro patrón de uso distinto, podrían mejorar estas previsiones. También se deja como posible trabajo futuro.

Capítulo 8

8. Aplicación desarrollada

En el presente capítulo se detalla la aplicación web desarrollada como demostrador de todos los trabajos realizados en los capítulos previos.

8.1 Definición y alcance

La aplicación web desarrollada durante el TFM integra los trabajos realizados y presentados en los capítulos previos con el objetivo de demostrar el proceso realizado de forma gráfica y facilitando su comprensión. Aunque la aplicación se trata de un prototipo susceptible de mejoras al ser la primera y única versión implementada de un producto que en un primer instante no había sido planificado, se posiciona como un futurible sistema para seguir con el estudio y análisis del rendimiento energético del edificio LUCIA.

En una primera instancia se pensó en la aplicación como una herramienta de apoyo para realizar la exploración de los datos mensuales del edificio LUCIA con el objetivo de visualizar gráficos de datos o algunos datos estadísticos de las variables de forma interactiva, una funcionalidad de gran ayuda debido al gran número de variables existentes. Posteriormente, la aplicación ha evolucionado con la incorporación de diferentes interfaces gráficas para la aplicación de las funcionalidades de limpieza, transformación de datos y selección de variables implementadas en el capítulo 5. Por último, se han incluido interfaces para el desarrollo de modelos empleando tanto las técnicas de *clustering* mostradas en el capítulo 6, como las técnicas de predicción de consumo energético usadas en el capítulo 7. Además, la aplicación cuenta con una funcionalidad propia que puede ser de gran ayuda como es la posibilidad de tener accesibles diferentes conjuntos de datos finales. La composición de estos conjuntos finales va a depender de las técnicas de limpieza y transformación de datos aplicadas, las variables utilizadas para generarlos o la finalidad con la que se han creado. Estos conjuntos de datos finales están accesibles tanto para su descarga en formato CSV, como para su uso en la propia aplicación.

La aplicación cubre la mayoría de las posibles tareas que pueden ser requeridas para los trabajos de exploración, limpieza y transformación de datos, pero en lo que respecta a la generación de modelos la funcionalidad se limita a los presentados en los capítulos 6 y 7. Por este motivo se ha incluido la posibilidad de exportar los datos de los conjuntos finales generados para que puedan emplearse con otras herramientas y para otros fines.

8.2 Análisis

En el presente apartado se van a mostrar las necesidades identificadas para la aplicación web, incluyendo los requisitos que debe cumplir el sistema y los diferentes casos de uso que deben poder completarse haciendo uso de la propia aplicación.

8.2.1 Requisitos

Un requisito es una característica demandada del sistema que puede expresar una capacidad de acción en el mismo o puede referirse a una restricción del servicio que ofrece o del propio proceso de desarrollo. A continuación se muestra el listado de requisitos definidos para la aplicación web:

- **Requisito 1:** El sistema debe permitir cargar y almacenar los raw data del edificio LUCIA a través de archivos Excel.
- **Requisito 2:** El sistema debe permitir visualizar un resumen descriptivo de los raw data cargados y sus variables.
- **Requisito 3:** El sistema debe permitir visualizar el listado de variables totales existentes en los raw data junto con su descripción, además de permitir su exportación.
- **Requisito 4:** El sistema debe permitir mostrar los datos totales agrupados en forma de serie temporal por variables.
- **Requisito 5:** El sistema debe permitir el estudio de las diferentes variables mediante diferentes tipos de gráficas o herramientas.
- **Requisito 6:** El sistema debe permitir exportar los datos totales una vez se han agrupado en forma de serie temporal.

- **Requisito 7:** El sistema debe permitir aplicar los diferentes métodos de limpieza y transformación de datos visualizados en el capítulo 5.
- **Requisito 8:** El sistema debe permitir aplicar el método de selección de características basado en el coeficiente de relación de Pearson visualizado en capítulo 5.
- **Requisito 9:** El sistema debe almacenar los conjuntos de datos finales generados tras la aplicación de diferentes métodos de limpieza, transformación y selección de características para su posterior uso en tareas de extracción de conocimiento.
- **Requisito 10:** El sistema debe permitir visualizar el listado de conjuntos finales generados.
- **Requisito 11:** El sistema debe permitir gestionar los conjuntos de datos finales generados (visualizar su composición, exportarlos o eliminarlos).
- **Requisito 12:** El sistema debe permitir probar modelos de *clustering* con diferentes configuraciones y sobre los diferentes conjuntos de datos finales generados.
- **Requisito 13:** El sistema debe permitir probar modelos de predicción con diferentes configuraciones y sobre los diferentes conjuntos de datos finales generados.
- **Requisito 14:** El sistema debe almacenar las ejecuciones de los mejores modelos con sus resultados y configuraciones.
- **Requisito 15:** El sistema debe permitir visualizar el listado de ejecuciones de modelos guardadas.
- **Requisito 16:** El sistema debe permitir visualizar el detalle de las ejecuciones de modelos guardadas, incluyendo las configuraciones empleadas y los resultados obtenidos

8.2.2 Casos de uso

Los casos de uso describen la interacción de los diferentes actores o usuarios en cada uno de los posibles escenarios que se pueden dar al interactuar con un sistema o aplicación. A continuación, se van a presentar todos los casos de uso cubiertos con la aplicación web desarrollada.



Figura 52: Diagrama de casos de uso

CU-1	Mostrar datos mensuales	
Actor	Usuario	
Descripción	El sistema debe ser capaz de mostrar los datos de los diferentes meses permitiendo seleccionar el mes a visualizar en cada momento	
Precondición		
Secuencia normal	Paso	Acción
	1	El usuario inicia la aplicación o accede a la opción de menú de meses cargados

	2	El sistema muestra la lista de meses con el último mes cargado seleccionado y sus datos visibles en una tabla con todas las variables que lo componen
	3	El usuario selecciona la visualización de otro mes
	4	El sistema muestra los datos del nuevo mes seleccionado
Excepciones	Paso	Acción
	1	Si no hay meses cargados el sistema muestra el caso de uso Cargar nuevo mes (CU-2)

Tabla 15: Caso de uso 1 - Mostrar datos mensuales

CU-2	Cargar nuevo mes	
Actor	Usuario	
Descripción	El sistema debe mostrar un campo para incluir nuevos datos, procesarlos una vez el usuario ha cargado un archivo, mostrar una previsualización de los datos y guardarlos en caso de que el usuario lo indique	
Precondición		
Secuencia normal	Paso	Acción
	1	El usuario indica que quiere cargar un nuevo mes desde la vista de meses cargados
	2	El sistema muestra un botón para cargar archivos Excel
	3	El usuario adjunta un archivo con los datos de un nuevo mes
	4	El sistema muestra una previsualización de los datos del archivo cargado y la posibilidad de guardar el mes
	5	El usuario indica que los datos del nuevo mes cargado se deben guardar
	6	El sistema muestra la vista de meses cargados y se realiza el caso de uso CU-1
Excepciones	Paso	Acción
	4	El usuario incluye un archivo con una extensión no permitida y el sistema muestra un error
	5	El usuario accede a otra opción de menú y el caso de uso queda sin efecto

Tabla 16: Caso de uso 2 - Cargar nuevo mes

CU-3	Eliminar mes	
Actor	Usuario	
Descripción	El sistema debe eliminar los datos del mes seleccionado por el usuario	
Precondición		
Secuencia normal	Paso	Acción
	1	El usuario indica que quiere eliminar los datos del mes seleccionado
	2	El sistema muestra un <i>pop-up</i> de confirmación
	3	El usuario indica que quiere eliminar el mes
	4	El sistema elimina los datos y recarga los datos mensuales volviendo al caso de uso CU-1
Excepciones	Paso	Acción
	3	El usuario indica que no quiere eliminar el mes y el caso de uso queda sin efecto
	4	Si el mes eliminado era el único mes guardado el sistema muestra la vista de nuevo mes volviendo al caso de uso CU-2

Tabla 17: Caso de uso 3 - Eliminar mes

CU-4	Eliminar datos mensuales	
Actor	Usuario	
Descripción	El sistema debe eliminar todos los datos cargados y mostrar la vista para cargar nuevos datos	
Precondición		
Secuencia normal	Paso	Acción
	1	El usuario indica que quiere eliminar todos los datos mensuales
	2	El sistema muestra un pop-up de confirmación
	3	El usuario indica que quiere eliminarlos
	4	El sistema elimina los datos y muestra la vista para cargar un nuevo mes (CU-2)
Excepciones	Paso	Acción

	3	El usuario indica que no quiere eliminarlos y el caso de uso queda sin efecto
--	----------	---

Tabla 18: Caso de uso 4 - Eliminar datos mensuales

CU-5	Ver gráfico de datos	
Actor	Usuario	
Descripción	El sistema debe mostrar el tipo de gráfico elegido por el usuario con los datos de la variable o variables seleccionada	
Precondición	Hay al menos una variable seleccionada	
Secuencia normal	Paso	Acción
	1	El usuario indica que quiere visualizar el gráfico de datos de las variables seleccionadas
	2	El sistema muestra un pop-up con los datos de las variables

Tabla 19: Caso de uso 5 - Ver gráfico de datos

CU-6	Ver datos totales	
Actor	Usuario	
Descripción	El sistema debe mostrar los datos agrupados de las variables comunes a todos los meses	
Precondición	Hay al menos un mes cargado	
Secuencia normal	Paso	Acción
	1	El usuario accede a la opción de menú de datos totales
	2	El sistema muestra una tabla con las variables comunes y los datos de las variables

Tabla 20: Caso de uso 6 - Ver datos totales

CU-7	Eliminar variables	
Actor	Usuario	
Descripción	El sistema debe eliminar las variables seleccionadas por el usuario del conjunto de datos visualizados en la vista de datos totales	
Precondición	Hay al menos una variable seleccionada	
Secuencia	Paso	Acción

normal	1	El usuario indica que quiere eliminar las variables seleccionadas
	2	El sistema elimina las variables y recarga la vista y tabla de datos totales

Tabla 21: Caso de uso 7 - Eliminar variables

CU-8	Limpiar datos	
Actor	Usuario	
Descripción	El sistema debe completar las acciones destinadas a limpiar, filtrar o preprocesar los datos cargados en la vista de datos totales	
Precondición	Hay datos cargados y el usuario se encuentra en la opción de menú de datos totales.	
Secuencia normal	Paso	Acción
	1	El usuario indica las acciones que el sistema debe realizar al seleccionarlas en un menú desplegable con el listado de las posibilidades existentes
	2	El sistema realiza las acciones indicadas por el usuario y recarga la vista y tabla de datos totales con los nuevos datos obtenidos
Excepciones	Paso	Acción
	2	Si algunas de las opciones indicadas por el usuario son incompatibles o se produce algún error al tratar los datos, el sistema muestra un pop-up de error y vuelve a cargar los datos totales iniciales volviendo al caso de uso CU-6

Tabla 22: Caso de uso 8 - Limpiar datos

CU-9	Agrupar datos	
Actor	Usuario	
Descripción	El sistema debe agrupar los datos de las variables disponibles en la vista de datos totales siguiendo las indicaciones del usuario.	
Precondición	Hay datos cargados y el usuario se encuentra en la opción de menú de datos totales.	
Secuencia	Paso	Acción

normal	1	El usuario indica que desea agrupar los datos en el selector de tipo de agrupación
	2	El sistema muestra un pop-up indicando al usuario que debe seleccionar el tipo de tratamiento de valores ausentes y si la agrupación debe realizarse por el valor medio o por la suma para cada tipo de dato
	3	El usuario indica el método de tratamiento de valores ausentes y el método de agrupación
	4	El sistema agrupa los datos y recarga la vista y la tabla con los nuevos datos agrupados
Excepciones	Paso	Acción
	2	El usuario cancela el proceso de agrupación y el caso de uso queda sin efecto

Tabla 23: Caso de uso 9 - Agrupar datos

CU-10	Ver matriz de correlación	
Actor	Usuario	
Descripción	El sistema debe mostrar una matriz con los coeficientes de correlación de Pearson entre las diferentes variables que conforman el conjunto de datos	
Precondición	Un tipo de agrupación de datos se ha aplicado sobre los datos totales	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la visualización de la matriz de correlación de Pearson
	2	El sistema muestra la matriz de correlación de Pearson con los diferentes coeficientes y clasificándolos por colores dependiendo del valor de correlación existente entre las variables

Tabla 24: Caso de uso 10 - Ver matriz de correlación

CU-11	Generar dataset	
Actor	Usuario	
Descripción	El sistema debe guardar el conjunto de datos resultante de aplicar las diferentes opciones de limpieza y agrupación de datos aplicadas por el usuario	

Precondición	Un tipo de agrupación de datos se ha aplicado sobre los datos totales	
Secuencia normal	Paso	Acción
	1	El usuario indica que quiere generar un <i>dataset</i> con el conjunto de datos existente en la vista de datos totales
	2	El sistema muestra un pop-up para indicar un nombre y descripción para el <i>dataset</i> a generar
	3	El usuario indica el nombre y la descripción del <i>dataset</i> e indica al sistema que debe guardar el <i>dataset</i>
	4	El sistema genera el nuevo <i>dataset</i> y muestra al usuario que el guardado se ha completado con éxito
Excepciones	Paso	Acción
	2	El usuario cancela el guardado del <i>dataset</i> y el caso de uso queda sin efecto

Tabla 25: Caso de uso 11 - Generar dataset

CU-12	Exportar datos	
Actor	Usuario	
Descripción	El sistema debe exportar los datos seleccionado a un archivo CSV	
Precondición		
Secuencia normal	Paso	Acción
	1	El usuario indica selecciona la opción de exportar datos
	2	El sistema genera un archivo en formato CSV con los datos disponibles o el dataste seleccionado

Tabla 26: Caso de uso 12 - Exportar datos

CU-13	Mostrar <i>dataset</i> generados	
Actor	Usuario	
Descripción	El sistema debe mostrar el listado de <i>datasets</i> guardados	
Precondición	Hay al menos un <i>dataset</i> guardado	
Secuencia	Paso	Acción

normal	1	El usuario selecciona la opción de menú de <i>datasets</i> generados
	2	El sistema muestra en una tabla el listado de <i>datasets</i> guardados por el usuario y una serie de acciones para realizar sobre los <i>datasets</i>

Tabla 27: Caso de uso 13 - Mostrar *datasets* generados

CU-14	Visualizar contenido del <i>dataset</i>	
Actor	Usuario	
Descripción	El sistema debe mostrar los datos que conforman el <i>dataset</i> seleccionado por el usuario	
Precondición	Hay al menos un <i>dataset</i> guardado	
Secuencia normal	Paso	Acción
	1	El usuario indica que quiere visualizar el contenido de uno de los <i>datasets</i> mostrados en el listado de <i>datasets</i> generados
	2	El sistema muestra un <i>pop-up</i> con las variables que lo conforman y datos estadísticos e información

Tabla 28: Caso de uso 14 - Visualizar contenido del *dataset*

CU-15	Eliminar <i>dataset</i>	
Actor	Usuario	
Descripción	El sistema debe eliminar el <i>dataset</i> seleccionado por el usuario	
Precondición	Hay al menos un <i>dataset</i> generado	
Secuencia normal	Paso	Acción
	1	El usuario indica que quiere eliminar uno de los <i>datasets</i> del listado
	2	El sistema muestra un <i>pop-up</i> (diálogo) de confirmación
	3	El usuario indica que quiere eliminar el mes
	4	El sistema elimina el <i>dataset</i> y recarga el listado de <i>datasets</i> volviendo al caso de uso CU-13
Excepciones	Paso	Acción

	3	El usuario indica que no quiere eliminar el dataset y el caso de uso queda sin efecto
	4	Si el <i>dataset</i> eliminado era el único <i>dataset</i> el sistema muestra la vista de meses cargados volviendo al caso de uso CU-1

Tabla 29: Caso de uso 15 - Eliminar dataset

CU-16	Generar modelo	
Actor	Usuario	
Descripción	El sistema debe permitir al usuario seleccionar el tipo de tarea, algoritmo, <i>dataset</i> y parámetros asociados al algoritmo	
Precondición	Hay al menos un <i>dataset</i> generado	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la opción de menú de modelado
	2	El sistema muestra una lista de seleccionables
	3	El usuario indica los campos obligatorios (tarea, algoritmo y <i>dataset</i>)
	4	El sistema muestra el listado de parámetros configurables para el algoritmo seleccionado
	5	El usuario indica los parámetros y selecciona probar el modelo
	6	El sistema muestra los resultados de las pruebas
	7	El usuario indica que desea guardar un modelo
	8	El sistema muestra un <i>pop-up</i> para seleccionar los parámetros con los que el modelo ofrece mejores resultados
	9	El usuario indica los parámetros del modelo final
10	El sistema guarda el modelo y los resultados de la ejecución final y muestra la vista del listado de ejecuciones guardadas (CU-17)	
Excepciones	Paso	Acción
	5.1	El usuario cancela el proceso y el caso de uso queda sin efecto
	5.1	El usuario indica un nuevo algoritmo
	5.2	El sistema muestra los parámetros configurables del nuevo algoritmo

	7	El usuario indica que desea configurar una nueva prueba y se vuelve al paso 2
	9	El usuario cancela el guardado del modelo final y el caso de uso queda sin efecto

Tabla 30: Caso de uso 16 - Generar modelo

CU-17	Mostrar listado de ejecuciones guardadas	
Actor	Usuario	
Descripción	El sistema debe mostrar el listado de ejecuciones de modelos guardadas	
Precondición	Hay al menos una ejecución de modelo guardada	
Secuencia normal	Paso	Acción
	1	El usuario selecciona la opción de menú de listado de ejecuciones guardadas
	2	El sistema muestra en una tabla el listado de ejecuciones de modelos guardadas por el usuario y una serie de acciones para realizar

Tabla 31: Caso de uso 17 - Mostrar listado de ejecuciones guardadas

CU-18	Visualizar detalle de ejecución	
Actor	Usuario	
Descripción	El sistema debe mostrar el detalle de la ejecución de modelo seleccionada	
Precondición	Hay al menos una ejecución de modelo guardada	
Secuencia normal	Paso	Acción
	1	El usuario indica que quiere visualizar el detalle de una de las ejecuciones mostradas en el listado de ejecuciones de modelos guardadas
	2	El sistema muestra una nueva vista con diferentes pestañas con los resultados de la ejecución y parámetros del modelo

Tabla 32: Caso de uso 18 - Visualizar detalle de ejecución

CU-19	Eliminar ejecución	
Actor	Usuario	
Descripción	El sistema debe eliminar la ejecución seleccionado por el usuario	
Precondición	Hay al menos una ejecución de modelo guardada	
Secuencia normal	Paso	Acción
	1	El usuario indica que quiere eliminar una de las ejecuciones de modelo del listado
	2	El sistema muestra un <i>pop-up</i> de confirmación
	3	El usuario indica que quiere eliminar la ejecución de modelo
4	El sistema elimina la ejecución y recarga el listado de ejecuciones volviendo al caso de uso CU-17	
Excepciones	Paso	Acción
	3	El usuario indica que no quiere eliminar la ejecución y el caso de uso queda sin efecto

Tabla 33: Caso de uso 19 - Eliminar ejecución

8.3 Diseño

En el presente apartado se va a mostrar la arquitectura de la solución elegida para implementar la aplicación web, así como el diseño de la base de datos y la secuencia de llamadas y conexiones que acontecen para dar servicio a cualquier petición realizada por el usuario desde su navegador web al interactuar con la aplicación.

8.3.1 Arquitectura *cliente-servidor*

En el ámbito informático se define como arquitectura al modelo conceptual que define la estructura, comportamiento, interacción entre componentes y vistas que tiene un sistema. En el presente caso la aplicación desarrollada sigue una arquitectura *cliente-servidor*, basada en la interacción de dos elementos clave:

- Cliente: inicia los procesos de servidor, es decir, es el demandante de servicio y requiere de conexión a un servicio externo que actúe como proveedor del servicio solicitado.

- Servidor: se encarga de proveer los recursos o servicios solicitados en función de las peticiones recibidas. El servidor además de interactuar con el cliente también puede interactuar con la capa de persistencia de datos o base de datos si las peticiones recibidas lo requieren.

La arquitectura *cliente-servidor* es el estándar de interacción con internet, donde el navegador web actúa como cliente e interactúa con diferentes servidores web que le proporcionan respuesta a las peticiones solicitadas. Puesto que la aplicación desarrollada se trata de una aplicación web, sigue esta misma arquitectura. Podemos distinguir distintos tipos de arquitectura cliente-servidor dependiendo del número de niveles que existan, donde cada nivel representa un dispositivo físico distinto. En nuestro caso hay 3 niveles diferenciados: un nivel de cliente y dos niveles de la parte de servidor.

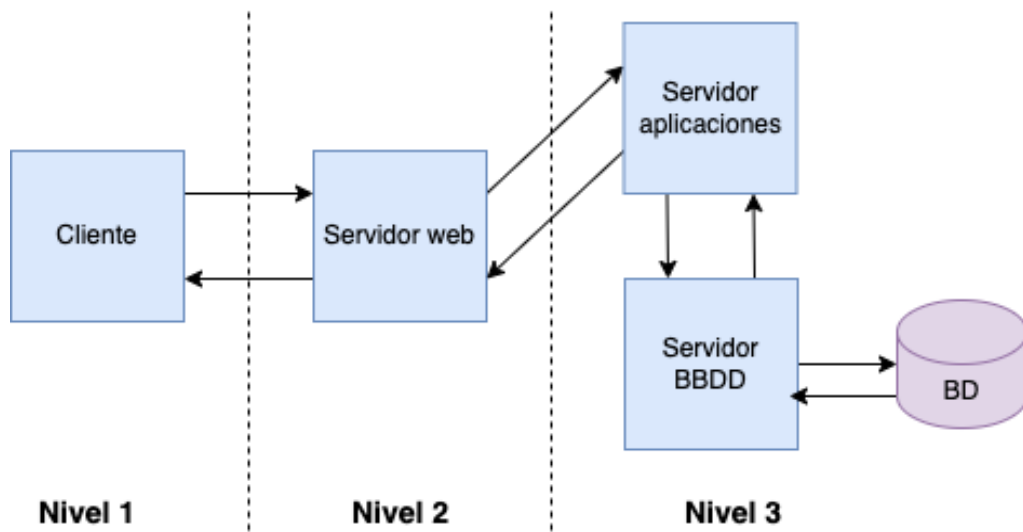


Figura 53: Arquitectura cliente-servidor

8.3.2 Arquitectura física

La arquitectura física de un sistema determina los componentes de hardware que lo conforman y sus interacciones. En la presente aplicación web los componentes que la componen, a excepción de la base de datos que se comunica usando el protocolo TCP, interactúan utilizando el protocolo HTTP. En la figura 54 se puede visualizar el diagrama de despliegue con la arquitectura física del proyecto desarrollado.

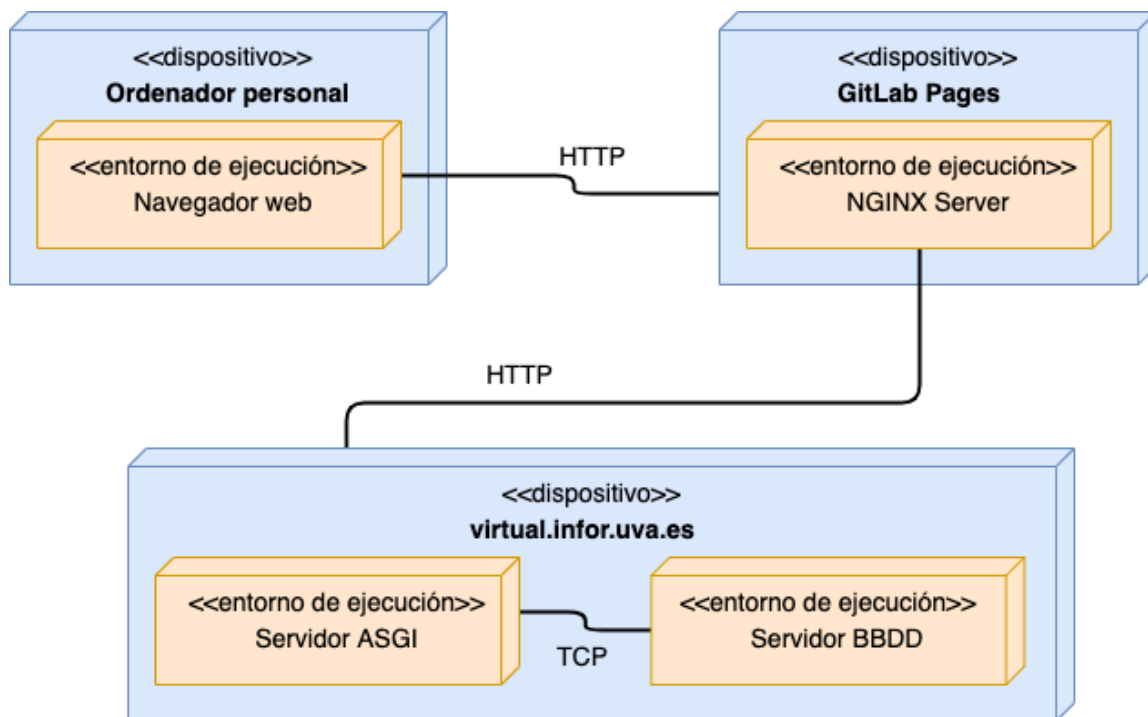


Figura 54: Arquitectura física

8.3.3 Arquitectura lógica

La arquitectura lógica es una representación de la forma en que interactúa el código fuente del software, permitiendo identificar patrones, sistemas y componentes en el proceso de interacción.

En el proyecto desarrollado a nivel lógico tenemos la capa de cliente, una capa intermedia o de lógica de negocio donde se encuentran las funciones principales de la aplicación web, y una capa de datos formada por los servicios que proporcionan los datos persistentes con los que trabaja la capa intermedia o de negocio. En la figura 55 podemos ver una representación de la arquitectura lógica de la aplicación web.

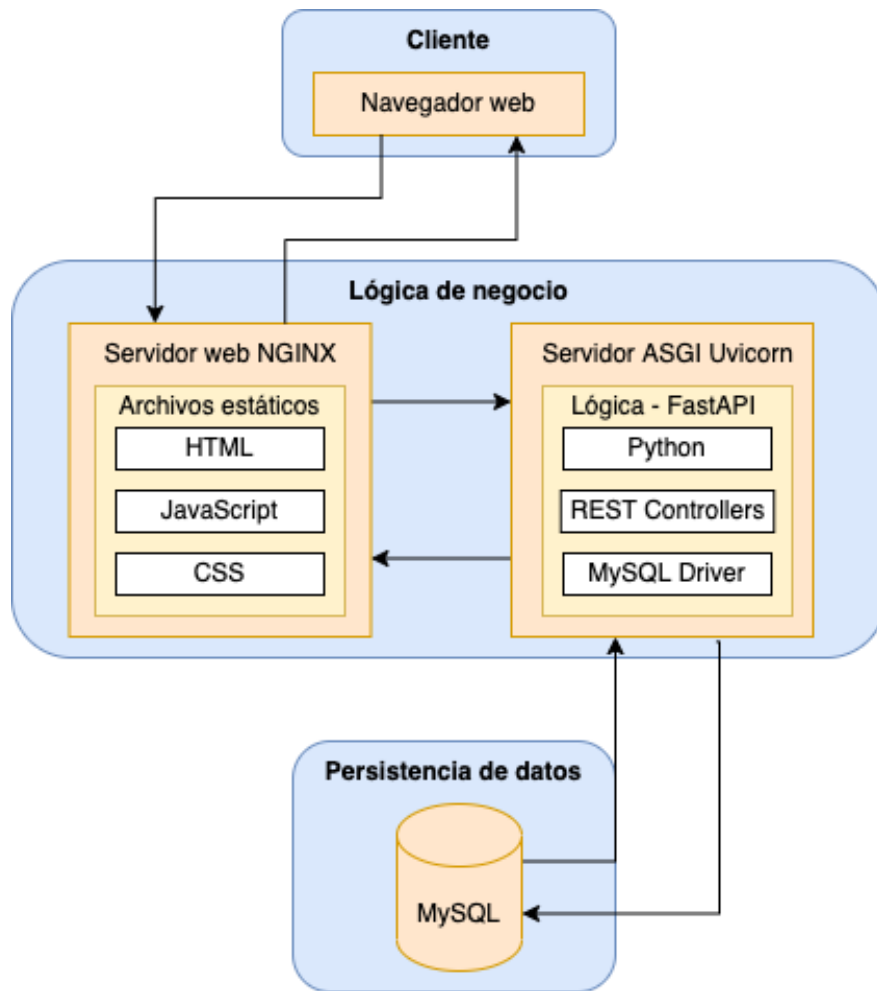


Figura 55: Arquitectura lógica

8.3.4 Diseño de base de datos

El diseño de la base de datos de la aplicación es muy simple y a la vez inusual por la forma de almacenamiento de los datos del edificio LUCIA y que se detallará a continuación. La base de datos está conformada por 4 tablas que quedan representadas en la figura 56.

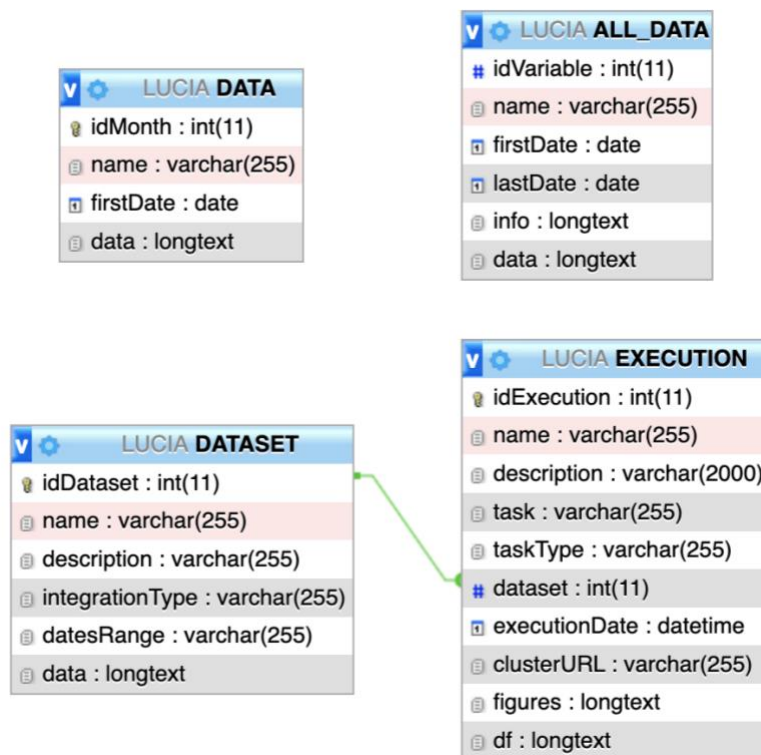


Figura 56: Diseño de base de datos

Debido a la gran cantidad de datos del edificio LUCIA con la que se trabaja (más de 12.000.000 de registros) a la hora de implementar la aplicación se ha primado la búsqueda de un mayor rendimiento. Por este motivo se ha hecho uso de la capacidad de MySQL para alojar objetos JSON con la finalidad de leer estos objetos directamente y facilitar su conversión a la estructura *DataFrame* de Pandas de forma casi inmediata para su uso.

Los campos del tipo “*longtext*” visualizados en la figura 56 están almacenando objetos JSON con los diferentes datos del edificio LUCIA. A continuación, se detalla el uso de cada una de las tablas mostradas en la figura 56:

- **DATA:** en esta tabla se almacenan los datos mensuales que se corresponden con los raw data del edificio. Quedan guardados: un identificador único para cada mes, el nombre del mes, la fecha del primer registro para ordenarlos y los datos una vez leídos de un archivo Excel y procesados para almacenarlos en un formato JSON que facilite su lectura.
- **ALL_DATA:** almacena cada una de las variables comunes a todos los meses junto con sus datos totales (una vez agrupados los meses), las fechas de primer y último registro e información estadística sobre la variable.

- **DATASET:** almacena los conjuntos de datos finales generados. Quedan guardados un identificador único para cada *dataset*, el nombre y descripción asignados para identificarlo en la aplicación web, el tipo de agrupación de datos aplicado (horas, días o semanas), el rango de fechas de los datos que conforman el *dataset* y los propios datos.
- **EXECUTION:** almacena las ejecuciones que son pruebas de modelo final sobre un *dataset* concreto y referenciado en la tabla mediante una clave foránea a la tabla **DATASETS**. También quedan guardados un identificador único para cada ejecución, el tipo de tarea realizada (*clustering* o predicción), el tipo de algoritmo empleado, la fecha de la ejecución, las URL de figuras generadas con la ejecución (guardadas en la API) y los resultados de la ejecución.

Con el uso de objetos para el almacenamiento de los datos se obtienen importantes beneficios de rendimiento a excepción del momento de carga de datos para un nuevo mes y consiguiente agrupación de los datos mensuales tras la carga.

8.3.5 Diagrama de secuencia de una petición

El diagrama de la figura 57, que se trata de un ejemplo de diagrama de secuencia, muestra las conexiones y llamadas necesarias para poder dar servicio a cualquier petición generada por el usuario desde su navegador web en relación con la aplicación desarrollada. En concreto, el diagrama se corresponde con el ejemplo de la secuencia genérica que seguiría cualquiera de los casos de uso mostrados en el apartado 8.2.2, aunque cada uno de estos lo haría con sus funciones y componentes (controladores, servicios...) asociados.

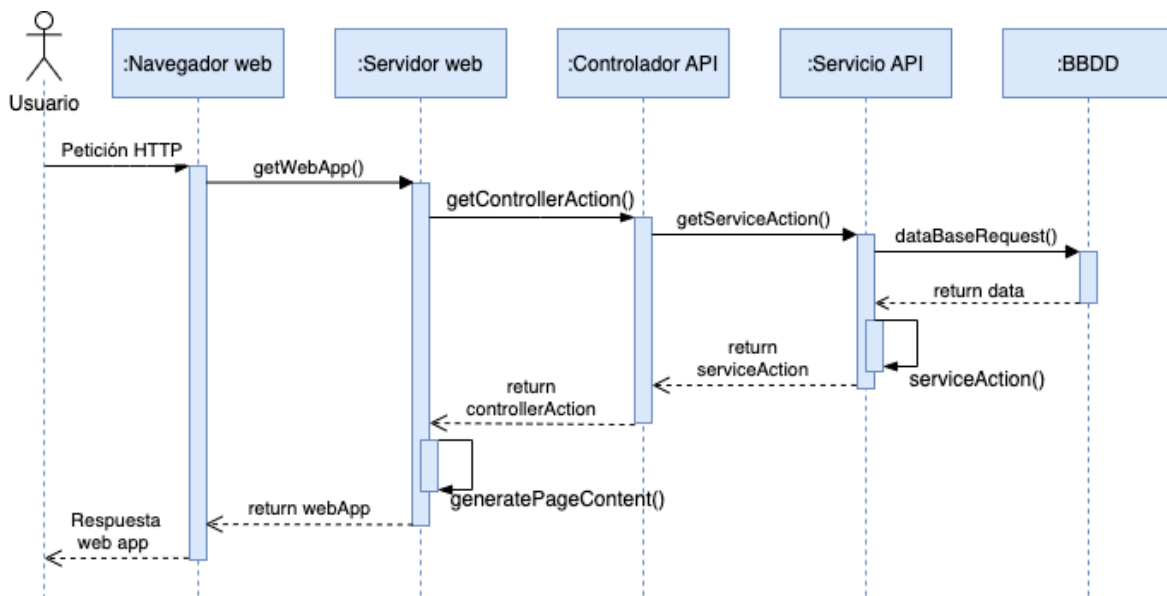


Figura 57: Diagrama de secuencia genérico

8.4 Implementación: tecnologías y estructura de componentes de aplicación

El servicio API REST donde se centra la parte lógica de la aplicación se ha desarrollado con FastApi el *framework* de Python específicamente pensado para el desarrollo de servicios ligeros y de gran rendimiento. La API desarrollada incluye todas las funcionalidades implementadas en los capítulos 5, 6 y 7 también realizadas con Python, lo cual ha facilitado su adaptación a servicios web REST.

La estructura del servicio API se ha dividido en los siguientes componentes:

- Archivo principal FastApi: incluye la configuración necesaria para desplegar el servicio conectando todos los controladores implementados y configuraciones existentes.
- Controladores: se encargan de gestionar las solicitudes HTTP que llegan por los diferentes *endpoints* (puntos finales de comunicación) configurados. Cada controlador desarrollado se encarga de gestionar las peticiones referentes a su campo de trabajo (procesado de datos, tareas de *machine learning*...) y de llamar a los servicios necesarios para cubrir las peticiones recibidas.
- Servicios: incluyen la lógica funcional necesaria para dar servicio a las peticiones recibidas por los diferentes *endpoints* configurados. La ejecución de las funciones incluidas en los servicios es orquestada por los controladores, existiendo al menos un servicio para cada controlador desarrollado.
- Modelos: archivos que representan objetos y que cuentan con las estructuras de datos y funciones necesarias para modelar un concepto real. En el presente caso se han desarrollado modelos para cada algoritmo de clustering y para las redes LSTM usadas.
- Directorio *public*: directorio público/accesible al exterior. En este directorio se incluyen todas las figuras o archivos a los que debe poder accederse externamente para mostrarlos en la aplicación web.
- Archivos de configuración: podemos encontrar tres archivos de configuración distintos: uno para la base de datos con las funciones genéricas para poder realizar consultas u otras operaciones que requieran de modificación de los datos, otro archivo con la configuración para el despliegue del entorno en un servidor usando certificados, hilos... y por último un archivo para variables globales de la aplicación.
- Archivo de variables de entorno: en este archivo se guarda la cadena de conexión a la base de datos empleada para la aplicación. En el caso de añadir cualquier nueva configuración que afecte al entorno y que requiera de una cadena de conexión o

variables individuales y con valores únicos para el entorno en cuestión, esta información debería incluirse en este archivo.

En la figura 58 se muestra el listado con los diferentes *endpoints* configurados en el servicio API. La figura se ha obtenido gracias a una de las funcionalidades de FastAPI que permite listar todos los *endpoints* existentes en el *framework* junto con el tipo de método bajo el que actúa cada uno de ellos.

FastAPI 0.1.0 OAS3
[/openapi.json](#)

Endpoints

POST	/ml/testModel	Testmodel
POST	/ml/generateModel	Generatemodel
POST	/ml/getExecutions	Getexecutions
POST	/ml/deleteExecution	Deletedataset
POST	/ml/getExecution	Getexecution
POST	/datasets/saveDataset	Savedataset
POST	/datasets/getDataset	Getdataset
POST	/datasets/getDatasetInfo	Getdatasetinfo
POST	/datasets/getDatasets	Getdatasets
POST	/datasets/deleteDataset	Deletedataset
POST	/datasets/deleteAllDatasets	Deletealldatasets
POST	/data/processFile	Upload File
POST	/data/saveDataMonth	Savedatamonth
POST	/data/getData	Getdata
POST	/data/resetData	Resetdata
POST	/data/dropData	Dropdata
POST	/data/setAllData	Setalldata
POST	/data/getVarsData	Getvarsdata
POST	/data/getAllDataVars	Getalldatavars
POST	/data/dropVars	Dropvars
POST	/data/getAllData	Getalldata
POST	/data/checkAllData	Checkalldata
POST	/data/getSortestRangeForValues	Getsortestrangeforvalues
POST	/data/filterDataByValue	Filterdatabyvalue
POST	/data/filterData	Filterdata
POST	/data/integrateData	Integratedata
POST	/data/getCorrelationMatrix	Getcorrelationmatrix
POST	/data/generateBoxPlot	Getcorrelationmatrix
POST	/data/getExportData	Getexportdata

Figura 58: Endpoints del servicio API

La parte de cliente web se ha desarrollado con el *framework* Angular siguiendo el patrón MVC (modelo, vista, controlador). Al ser Angular un *framework* modular hemos podido trabajar mediante componentes. Cada uno de estos componentes generalmente representa una vista de la aplicación, aunque podemos encontrar algunas excepciones de componentes auxiliares con una funcionalidad específica y que sirven de apoyo al resto de componentes generales.

A continuación, se van a presentar los componentes desarrollados para representar las principales interfaces de la aplicación junto con su funcionalidad asociada:

- Componente de meses: muestra los datos de los diferentes meses cargados, pudiendo seleccionar el mes a visualizar, generar gráficas de datos para las variables que lo componen y visualizar datos estadísticos de las propias variables.
- Componente nuevo mes: permite la carga de un archivo Excel con los datos de un nuevo mes. Una vez cargado y procesado el archivo se muestra una previsualización de su contenido en una vista similar a la del componente de meses. La vista de previsualización incluye una opción para guardar definitivamente los datos del mes cargado.
- Componente de variables: muestra una tabla con el nombre y descripción de todas las variables cargadas con los diferentes meses. El componente permite filtrar el listado de variables de la tabla y exportar el contenido de la tabla a formato CSV.
- Componente de datos totales: es el componente con más funcionalidad de la aplicación. Muestra el listado de variables comunes a todos los meses una vez sus datos han sido unificados en forma de serie temporal. Este componente además incluye la funcionalidad necesaria para realizar la limpieza y transformación de datos, pudiendo agrupar los datos por horas, días o semanas y posteriormente ver una matriz de correlación como método para la selección de variables finales. El componente de datos totales también cuenta con la funcionalidad para guardar los *datasets* o conjuntos de datos finales.
- Componente de *datasets*: muestra una tabla con el listado de *datasets* finales generados. Los campos de la tabla son: nombre y descripción del *dataset*, tipo de agrupación de datos (horas, días o semanas) en caso de haberse realizado y el rango de fechas de los datos que conforman el *dataset*. Además, sobre cada *dataset* se pueden realizar una serie de acciones como son la exportación de su contenido en formato CSV, la visualización del contenido con mayor detalle para ver las variables que lo componen y algunos datos estadísticos y la opción de eliminar el *dataset*.

- Componente de modelado o test: incluye la funcionalidad desarrollada en los capítulos 6 y 7 para generar modelos tanto de *clustering*, como de predicción de consumo energético. El componente permite la selección del tipo de trabajo a realizar (*clustering* o predicción), el tipo de algoritmo dependiendo del trabajo a realizar, el *dataset* a emplear y la configuración de diferentes parámetros para el algoritmo seleccionado. Una vez se ejecuta el modelo se pueden visualizar los resultados de la ejecución y las métricas usadas en los capítulos 6 y 7 para evaluar el rendimiento de los modelos. Estas ejecuciones se pueden volver a lanzar ajustando los parámetros empleados, o en el caso de considerar que la configuración es óptima se pueden guardar los resultados finales obtenidos y la parametrización utilizada.
- Componente de ejecuciones realizadas: muestra una tabla con el listado de ejecuciones guardadas desde el componente de modelado. Cada ejecución se puede ver en detalle mostrando los resultados y parametrización del modelo asociado.

8.5 Despliegue de la aplicación en un entorno de producción

Para el despliegue de la aplicación web en un entorno de producción se requiere de un servidor web para alojar el contenido estático, un servidor de aplicaciones Python y un servidor de bases de datos, que en nuestro caso va a ser MySQL. A continuación, se muestran los recursos utilizados para alojar y desplegar la aplicación web:

- GitLab Pages: servicio que ofrece la compañía GitLab para el despliegue de páginas estáticas. El uso del servicio requiere tener un repositorio con el proyecto que se desea alojar y un archivo con la denominación “.gitlab-ci.yml” donde se indican las tareas o pasos que la herramienta de CI/CD debe ejecutar para lograr desplegar el contenido estático. En el caso del presente proyecto el repositorio contiene la parte *frontend*, es decir, los desarrollos realizados con Angular. En el archivo con extensión “yml” se han indicado los scripts para instalar las dependencias de Node.js, compilar el proyecto Angular para generar los archivos estáticos (HTML, JavaScript y CSS) y crear y mover los archivos estáticos a un directorio denominado “*public*” que es servido por el servidor NGINX de GitLab.
- Máquina virtual: en ella se han instalado todas las dependencias necesarias del servicio API desarrollado, el servidor ASGI (*Asynchronous Server Gateway Interface*) que proporciona una interfaz entre servidores web y aplicaciones desarrolladas en Python y el servidor de bases de datos MySQL.

La aplicación web finalmente queda alojada sobre GitLab con un dominio predefinido por la propia compañía al no disponer de un dominio propio. Una de las ventajas de tener la

aplicación web alojada bajo un dominio de GitLab es la seguridad, pudiendo acceder bajo HTTPS al tener un certificado SSL/TLS asociado al propio dominio.

El hecho de tener la aplicación web alojada utilizando la herramienta de CI/CD facilita el despliegue de cualquier cambio de código y funcional que se realice gracias al pipeline de despliegue automatizado que se ha creado con este servicio y que se ejecuta con cada nuevo cambio que se incluye en el repositorio del proyecto.

Capítulo 9

9. Conclusiones y líneas futuras

Con el presente proyecto, desde el punto de vista del trabajo realizado, se han cubierto los objetivos inicialmente planteados.

El estudio del edificio LUCIA, los diferentes sistemas implicados en la eficiencia energética, los sensores monitorizados, etc., y el análisis de diferentes artículos referentes a la predicción de consumo energético en *Smart Buildings*, ha permitido contextualizar el caso de estudio del presente proyecto.

A pesar de que el *dataset* del edificio LUCIA ha resultado ser complejo, se ha conseguido identificar los sistemas y variables clave, además de preprocesar y transformar los datos para dejarlos listos para la fase de extracción de conocimiento. Para ello ha sido necesario realizar un estudio exhaustivo de las variables que lo componen, identificar y comprender los tipos de datos que se estaban registrando, identificar y tratar las irregularidades y *outliers* existentes y generar nuevas variables necesarias para poder predecir el consumo energético del edificio LUCIA. Esta última tarea ha resultado desafiante al no disponer de los datos de consumo energético del edificio completo en una única variable y de forma accesible. Para obtener los datos de consumo energético ha sido necesario identificar los analizadores de red necesarios para poder generar una variable que represente el consumo del edificio y posteriormente procesar los datos de consumo para transformarlos a registros entre fechas, ya que los analizadores estaban registrando consumos acumulados.

Mediante la aplicación de *clustering* se han podido identificar estados de funcionamiento del edificio LUCIA. El uso de *clustering* sobre los datos meteorológicos ha confirmado la presencia de los estados de funcionamiento elegidos por los gestores del edificio LUCIA, mientras que el uso de los datos del área de producción nos ha permitido descubrir otros patrones que podrían requerir de otro control por parte de los gestores del edificio. Los dos mejores modelos de clúster obtenidos, tanto usando las variables de condiciones externas, como con las variables del área de producción, se han usado para generar modelos predictivos para los estados de

funcionamiento identificados a partir de cada conjunto de datos. La presencia de clústeres de operación asociados a las condiciones meteorológicas hace más factible usar modelos de predicción a una semana asumiendo que podemos usar las predicciones meteorológicas para saber en qué modo nos podemos encontrar la semana siguiente.

Los resultados de predicción de consumo energético empleando la arquitectura de capas LSTM apiladas han demostrado como el rendimiento de un modelo para cada modo de operación del edificio identificado mediante el clúster, usando tanto datos recientes como datos históricos de ese clúster, proporcionan predicciones más precisas que usando un modelo global que no considere modos. El modelo de clúster usando datos recientes ha obtenido mejor rendimiento que el resto de los modelos probados. Además, se ha comprobado cómo el rendimiento de modelos individuales para cada estado de funcionamiento identificado empleando datos de condiciones externas son mejores que los de un modelo general y que los de un modelo generado para un clúster identificado con datos del área de producción.

Por último, se ha conseguido implementar una aplicación web para demostrar los trabajos realizados a lo largo del desarrollo del presente proyecto. A pesar de ser un prototipo susceptible de mejoras permite realizar y mostrar los resultados de modelos de *clustering* y predicción además de estudiar, preprocesar el *dataset* del edificio LUCIA y generar nuevos datasets que pueden ser empleados en otros trabajos (como TFGs o TFMs que no tendrán que procesar todos los datos en bruto).

De cara a un trabajo futuro, por una parte, la mejora del rendimiento de los modelos predictivos se podría conseguir mediante la aplicación de arquitecturas más complejas o mediante la distinción de los datos de consumo perteneciente al tercer estado de funcionamiento operativo del edificio LUCIA como se ha mencionado en el apartado [7.8](#). Por otra parte, la aplicación web desarrollada está abierta a la inclusión de mejoras y nuevas funcionalidades siendo una posible vía de trabajos futuros a realizar.

Referencias bibliográficas

- [1] Organización de las Naciones Unidas - Impacto académico. *Sostenibilidad*. Último acceso: 16/03/2022. URL: <https://www.un.org/es/impacto-académico/sostenibilidad>
- [2] Buildings Performance Institute Europe. *nZEB definitions across Europe*. Último acceso: 18/03/2022. URL: <https://www.bpie.eu/publication/nzeb-definitions-across-europe-2015/>
- [3] Convención Marco de las Naciones Unidas sobre el Cambio Climático. *El acuerdo de París*. Último acceso: 18/03/2022. URL: <https://unfccc.int/es/process-and-meetings/the-paris-agreement>
- [4] Desirée Arias-Requejo, Carlos J. Alonso-González, Belarmino Pulido, Marcus M. Keane (2020). *Energy demand prediction in Smart Buildings using advanced machine learning techniques*. eSIM Conference Proceedings (Canada). eSim 2020, Vancouver, Canada. URL: http://www.ibpsa.org/proceedings/eSimPapers/2021/Contribution_1169_final_a.pdf
- [5] Scrum.org (2018). *What is Scrum?*. Último acceso: 20/05/2022. URL: <https://www.scrum.org/resources/what-is-scrum>
- [6] David B. Ramos, Ilmara M. M. Ramos, Wesley de S. Viana, Gabriel R. e Silva, Elaine H. T. de Oliveira (2016). *On the use of Scrum for the management of research oriented projects*. URL: <http://www.tise.cl/volumen12/TISE2016/589-594.pdf>
- [7] The Python Software Foundation. *Python 3.9.8 documentation*. Último acceso: 25/05/2022. URL: <https://docs.python.org/3.9/>
- [8] Pandas 1.4.1 (2020). *pandas - Python Data Analysis Library - Documentation*. Último acceso: 25/05/2022. URL: https://pandas.pydata.org/docs/user_guide/index.html#user-guide
- [9] Scikit-learn developers. *Scikit-learn guide*. Último acceso: 09/05/2022. URL: https://scikit-learn.org/stable/user_guide.html
- [10] Keras. *Developer guides*. Último acceso: 20/05/2022. URL: <https://keras.io/guides/>
- [11] FastAPI. *FastAPI guide*. Último acceso: 28/04/2022. URL: <https://fastapi.tiangolo.com/>

- [12] Angular. *Documentación de Angular*. Último acceso: 07/04/2022. URL: <https://docs.angular.lat/docs>
- [13] GitLab. *GitLab CI/CD documentation*. Último acceso: 07/04/2022. URL: <https://docs.gitlab.com/ee/ci/>
- [14] GitLab. *GitLab Pages documentation*. Último acceso: 07/04/2022. URL: <https://docs.gitlab.com/ee/user/project/pages/>
- [15] Edificio LUCIA - Universidad de Valladolid (UVa). *El edificio*. Último acceso: 16/03/2022. URL: <http://edificio-lucia.blogspot.com/p/el-edificio.html>
- [16] Siemens. *Desigo™ Insight Operating the management station. V6.0 SP3 User's guide, Volume 1*. Último acceso: 25/03/2022. URL: <https://www.downloads.siemens.com/download-center/Download.aspx?pos=download&fct=getasset&id1=A6V1036222>
- [17] Siemens. *Desigo™ Insight Operating the management station. V6.0 SP2 User's guide, Volume 2*. Último acceso: 25/03/2022. URL: <https://www.downloads.siemens.com/download-center/Download.aspx?pos=download&fct=getasset&id1=A6V1036224>
- [18] Naug, A., & Biswas, G. (2018). *Data Driven Methods for Energy Reduction in Large Buildings*. 2018 IEEE International Conference on Smart Computing (SMARTCOMP), 131-138. URL: <https://ieeexplore.ieee.org/document/8421341>
- [19] Benedetto Grillone, Gerard Mor, Stoyan Danov, Jordi Cipriano, Andreas Sumper (2021). *A data-driven methodology for enhanced measurement and verification of energy efficiency savings in commercial buildings*. ISSN 0306-2619. URL: <https://doi.org/10.1016/j.apenergy.2021.117502>
- [20] Bzdok, D., Altman, N., Krzywinski, M. (2018). *Statistics versus machine learning*. Nat Methods 15, 233–234. URL: <https://doi.org/10.1038/nmeth.4642>
- [21] Aníbal Bregón Bregón, Quiliano Isaac Moro Sancho. *Redes neuronales convolucionales*. (2021). Máster en Ingeniería Informática. Material de la asignatura: *Deep learning* y sus aplicaciones. Escuela de Ingeniería Informática de Valladolid. Universidad de Valladolid.

- [22] Hrnjica B., Mehr A.D. (2020). *Energy Demand Forecasting Using Deep Learning*. URL: https://www.researchgate.net/Energy_Forecasting_Using_Deep_Learning
- [23] Aníbal Bregón Bregón, Quiliano Isaac Moro Sancho. *Redes Neuronales Recurrentes - LSTM*. (2021). Máster en Ingeniería Informática. Material de la asignatura: *Deep learning y sus aplicaciones*. Escuela de Ingeniería Informática de Valladolid. Universidad de Valladolid.
- [24] Desiree Arias-Requejo, Carlos J. Alonso-Gonzalez, Belarmino Pulido, Marcus M. Keane (2020). *Advanced machine learning techniques for predictive maintenance of HVAC subsystems based on energy consumption prediction*. Proceedings of BSO-V2020. 5th Building Simulation and Optimization Virtual Conference. 21 y 22 de septiembre de 2020. Loughborough, UK. URL: http://www.ibpsa.org/proceedings/BSO2020/BSOV2020_Requejo.pdf
- [25] Alejandro Barón García (2020). *Energy Load Forecast in Smart Buildings with Deep Learning Techniques*. Trabajo de Fin de Grado en Ingeniería Informática. Universidad de Valladolid. Escuela de Ingeniería Informática de Valladolid. URL: <https://uvadoc.uva.es/handle/10324/44114>
- [26] Bianchi, FM, Maiorino, E., Kampffmeyer, MC, Rizzi, A., Jenssen, R. (2017). *An overview and comparative analysis of Recurrent Neuronal Networks from Short Term Load Forecasting*. ArXiv:1705.04378 [Cs]. URL: <https://arxiv.org/abs/1705.04378>
- [27] Alberto Gasparin, Slobodan Lukovic, and Cesare Alippi (2019). *Deep Learning for Time Series Forecasting: The Electric Load Case*. 2019. URL: <https://arxiv.org/abs/1907.09207>
- [28] Elena Mocanu, Phuong H. Nguyen, Madeleine Gibescu, Wil L. Kling. (2016). *Deep learning for estimating building energy consumption, Sustainable Energy, Grids and Networks*. ISSN 2352-4677. URL: <https://doi.org/10.1016/j.segan.2016.02.005>
- [29] Murtagh, F., Contreras, P. (2011). *Methods of Hierarchical Clustering*. URL: <https://arxiv.org/abs/1105.0121>
- [30] Ahmed, M., Seraj, R., Islam, S.M.S (2020). *The k-means Algorithm: A Comprehensive Survey and Performance Evaluation*. URL: <https://doi.org/10.3390/electronics9081295>

- [31] Montalvo, D. (2020). *Tensores y datos tensoriales*. Trabajo de Fin de Grado en Matemáticas. Universidad de Valladolid. Facultad de ciencias. URL: <https://uvadoc.uva.es/handle/10324/43948>
- [32] Cousineau, D., Chartier, S. (2010). *Outliers detection and treatment: a review*. International Journal of Psychological Research. URL: <https://www.redalyc.org/pdf/2990/299023509004.pdf>
- [33] Pandas. `pandas.DataFrame.diff`. Último acceso: 29/03/2022. URL: <https://pandas.pydata.org/pandas.DataFrame.diff.html>
- [34] Scikit-learn. `preprocessing.MinMaxScaler`. Último acceso: 04/04/2022. URL: <https://scikit-learn.org/MinMaxScaler.html>
- [35] Walpole, R. E., Myers, R. H., Myers, S. L. (2007). Probabilidad y estadística para ingeniería y ciencias (8a. ed.). México Distrito Federal: Pearson Educación. 2012. ISBN: 978-607-32-1417-9
- [36] Scikit-learn. *Davies-Bouldin-Score*. Último acceso: 08/05/2022. URL: https://scikit-learn.org/davies_bouldin_score.html
- [37] Davies, D. L., Bouldin, D. W. (1979). *A Cluster Separation Measure*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2), 224–227. URL: <https://doi.org/10.1109/TPAMI.1979.4766909>
- [38] Scikit-learn. *Selecting the number of clusters with silhouette analysis*. Último acceso: 09/04/2022. URL: https://scikit-learn.org/stable/cluster/plot_kmeans_silhouette_analysis.html
- [39] Georgion Drakos (2020). *Silhouette Analysis vs Elbow Method vs Davies-Bouldin Index: Selecting the optimal number of clusters for KMeans clustering*. URL: <https://gdcoder.com/silhouette-analysis-vs-elbow-method-vs-davies-bouldin-index-selecting-the-optimal-number-of-clusters-for-kmeans-clustering/>
- [40] Jason Brownlee (2018). *Multi-Step LSTM Time Series Forecasting Models for Power Usage*. URL: <https://machinelearningmastery.com/how-to-develop-lstm-models-for-multi-step-time-series-forecasting-of-household-power-consumption/>

- [41] Jason Brownlee (2017). *How to Diagnose Overfitting and Underfitting of LSTM Models*.
URL: <https://machinelearningmastery.com/diagnose-overfitting-underfitting-lstm-models/>
- [42] J. Lovvorn, N.C., Watson, T.E., Arnold, C.G., Baxter, V.D., Borges, D.S., Cabot, P.W., Clements, W.S., Coward, C.W., Crowder, H.L., Dougherty, B.P., Evans, R.A., Hallstrom, A.D., Hermans, R.D., Hogan, J.F., Kohloss, F.H., Landman, W.J., Lewis, R.H., Montgomery, R.D., Novosel, D., Stanke, D.A., Tavares, M., Taylor, S.T., Wright, J.R., Burgett, L.W., Holness, G.V.R., Ramspeck, C.B., 2002. ASHRAE STANDARDS COMMITTEE 2001-2002, 170.

Anexo I - Manual de instalación de la aplicación web

El uso de la aplicación desarrollada requiere de la instalación del sistema gestor de bases de datos MySQL, Python 3, Node.js y Angular Cli. Una vez instalados lo primero que debemos hacer es clonar el repositorio con el código fuente de los desarrollos y algunos de los archivos con datos mensuales del edificio LUCIA para realizar pruebas. Ejecutando el siguiente comando clonamos el repositorio:

```
git clone https://gitlab.inf.uva.es/enrgarc/tfm-edificio-lucia.git
```

En el repositorio podemos encontrar 3 directorios:

- visualizador: se trata de la aplicación desarrollada con Angular.
- api: es el servicio API REST desarrollado con FastAPI para el trabajo con los datos y conexión con la base de datos MySQL, que como se ha indicado antes deberemos tener instalada en el sistema para poder trabajar.
- Datos mensuales: en esta carpeta podemos encontrar datos de algunos meses para poder probar la aplicación.

Una vez clonado el repositorio comenzaremos con la instalación del servicio API REST siguiendo los siguientes pasos:

1. Accedemos al directorio del repositorio local denominado “api”.
2. Ejecutamos el siguiente comando con el cual instalamos todas las dependencias necesarias para el proyecto:

```
pip install -r requirements.txt
```

3. Modificamos el nombre del archivo “.env_default” a “.env” e incluimos los nuevos datos de conexión a nuestra base de datos MySQL.
4. Desde el directorio “api” ejecutando el siguiente comando lanzaremos el servicio localmente haciéndolo accesible en el puerto 8000:

```
uvicorn main:app --reload
```

Una vez instalado el servicio API REST queda por instalar la aplicación desarrollada con Angular. Para lograrlo seguiremos los siguientes pasos:

1. Accedemos al directorio “visualizador” del repositorio y ejecutamos el siguiente comando para instalar las dependencias necesarias de Angular:

npm install.

2. Editamos el archivo “./src/enviroments/enviroment.ts” indicando la dirección en la que está accesible el servicio API REST que hemos lanzado anteriormente (por defecto viene indicada la ruta local al puerto 8000).
3. Ejecutamos el siguiente comando con el cual nos abrirá una instancia de la aplicación web en el navegador:

ng serve -o

Una vez ejecutados los pasos anteriores tendremos la aplicación web desplegada localmente y accesible a través de nuestro navegador web en la ruta local y puerto 4200:

http://localhost:4200/

Anexo II - Manual de usuario de la aplicación web

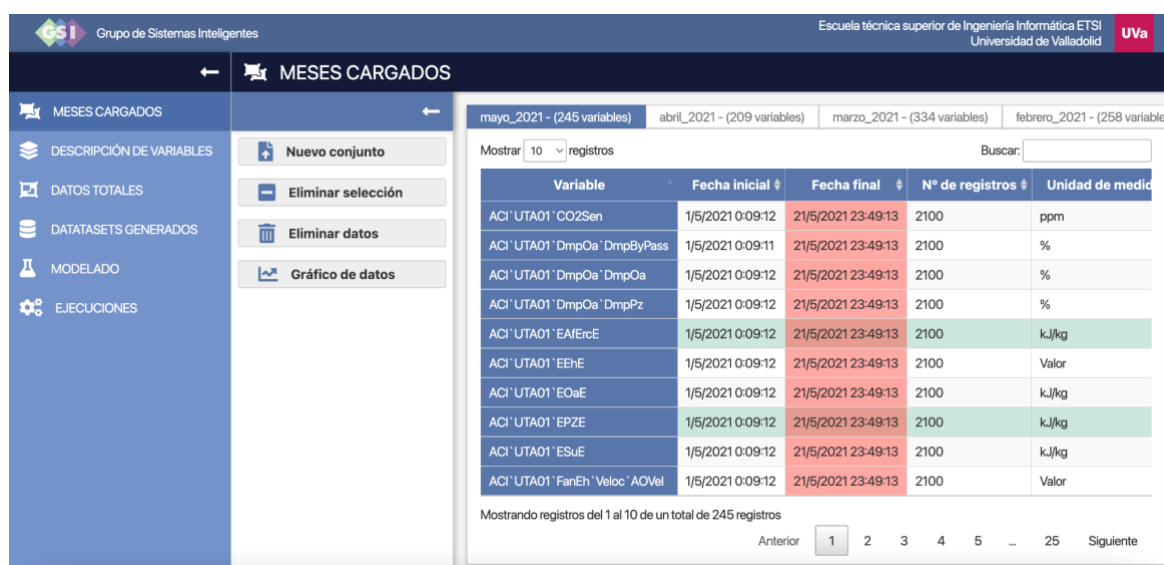
Como se ha mencionado en el apartado [8.5](#), la aplicación web se ha desplegado en un entorno en producción encontrándose accesible al exterior a través de la siguiente dirección web:

Solicitar a gsi@infor.uva.es esta dirección

Algunas de las funcionalidades originales de la aplicación web desarrollada no se encuentran disponibles en la aplicación a la que podemos acceder a través del anterior enlace. En concreto en la versión desplegada no se encuentra disponible la funcionalidad de exportar en formato CSV debido a que los datos del edificio LUCIA no son de dominio público para su uso. Además de la funcionalidad de exportar, no está disponible la posibilidad de generar modelos predictivos por la incompatibilidad entre las dependencias de Python relacionadas con Tensorflow y la máquina virtual proporcionada por la Universidad de Valladolid donde se encuentra alojado el servicio API.

1. Navegación y pantalla de meses cargados

Al iniciar la aplicación podemos ver la vista de meses cargados y un *sidebar* (barra lateral) desplegable con las diferentes opciones de navegación disponibles. Además, cada pantalla seleccionada cuenta con una “miga de pan” o hilo de Ariadna para saber en qué componente de la navegación nos encontramos en cada momento:



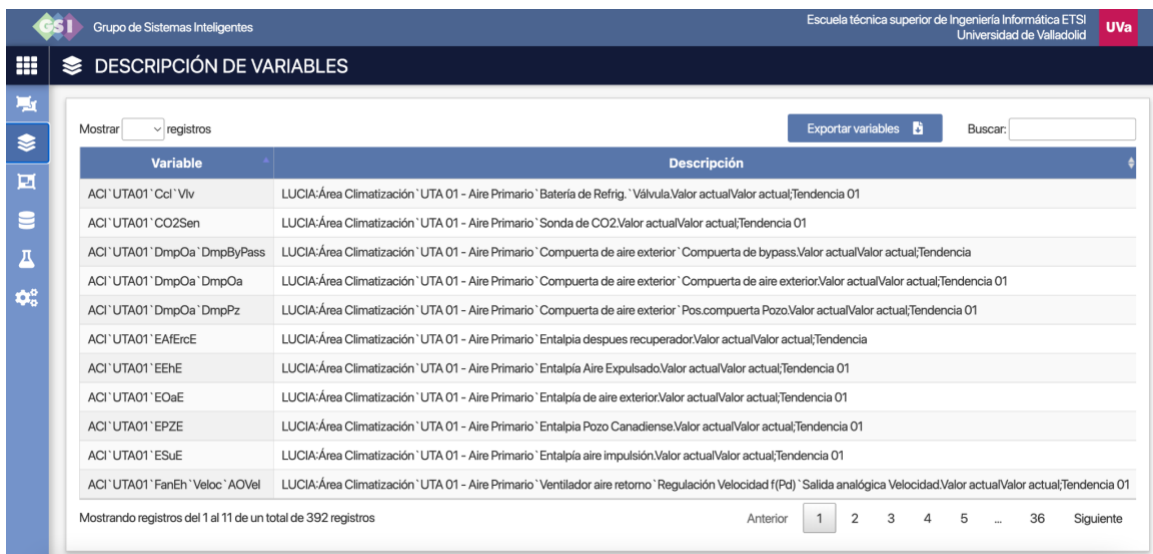
The screenshot displays the 'MESES CARGADOS' (Loaded Months) interface. The top navigation bar includes the logo for 'Grupo de Sistemas Inteligentes' and 'Escuela técnica superior de Ingeniería Informática ETSI Universidad de Valladolid UVA'. The main content area shows a list of loaded months: 'mayo_2021 - (245 variables)', 'abril_2021 - (209 variables)', 'marzo_2021 - (334 variables)', and 'febrero_2021 - (258 variables)'. Below this, there is a table with columns for 'Variable', 'Fecha inicial', 'Fecha final', 'Nº de registros', and 'Unidad de medic'. The table lists 10 variables, all with a start date of 1/5/2021 0:09:12 and an end date of 21/5/2021 23:49:13, and 2100 records each. The units of measurement are ppm, %, kJ/kg, and Valor. A sidebar on the left contains navigation options: 'MESES CARGADOS', 'DESCRIPCIÓN DE VARIABLES', 'DATOS TOTALES', 'DATASETS GENERADOS', 'MODELADO', and 'EJECUCIONES'. The interface also includes a search bar and a pagination control at the bottom showing 'Mostrando registros del 1 al 10 de un total de 245 registros'.

Variable	Fecha inicial	Fecha final	Nº de registros	Unidad de medic
ACI`UTA01`CO2Sen	1/5/2021 0:09:12	21/5/2021 23:49:13	2100	ppm
ACI`UTA01`DmpOa`DmpByPass	1/5/2021 0:09:11	21/5/2021 23:49:13	2100	%
ACI`UTA01`DmpOa`DmpOa	1/5/2021 0:09:12	21/5/2021 23:49:13	2100	%
ACI`UTA01`DmpOa`DmpPz	1/5/2021 0:09:12	21/5/2021 23:49:13	2100	%
ACI`UTA01`EAFercE	1/5/2021 0:09:12	21/5/2021 23:49:13	2100	kJ/kg
ACI`UTA01`EEHE	1/5/2021 0:09:12	21/5/2021 23:49:13	2100	Valor
ACI`UTA01`EOaE	1/5/2021 0:09:12	21/5/2021 23:49:13	2100	kJ/kg
ACI`UTA01`EPZE	1/5/2021 0:09:12	21/5/2021 23:49:13	2100	kJ/kg
ACI`UTA01`ESuE	1/5/2021 0:09:12	21/5/2021 23:49:13	2100	kJ/kg
ACI`UTA01`FanEh`Veloc`AOVel	1/5/2021 0:09:12	21/5/2021 23:49:13	2100	Valor

Desde la vista de meses cargados se puede cargar un nuevo mes o conjunto de datos, eliminar el mes seleccionado, elegir un nuevo mes a visualizar a través de la barra superior de la tabla donde se encuentran todos los meses disponibles, eliminar todos los datos existentes o mostrar gráficos de datos para las variables seleccionadas (en verde en la tabla de datos). La tabla donde se muestra información y algunos datos estadísticos de las variables es ordenable, cuenta con un filtro que actúa sobre todas las columnas de la tabla y la paginación es configurable.

2. Descripción de variables

Para poder ver el nombre y descripción de cada una de las variables existentes se ha creado una interfaz que cuenta con una tabla para visualizar el listado de variables, filtrarlas o exportarlas a CSV:



The screenshot shows a web interface titled 'DESCRIPCIÓN DE VARIABLES' from the 'Grupo de Sistemas Inteligentes' at 'Escuela técnica superior de Ingeniería Informática ETSI Universidad de Valladolid'. The interface includes a search bar, an 'Exportar variables' button, and a table with the following data:

Variable	Descripción
ACI`UTA01`Ccl`Vlv	LUCIA:Área Climatización`UTA 01 - Aire Primario`Batería de Refrig.`Válvula.Valor actual Valor actual;Tendencia 01
ACI`UTA01`CO2Sen	LUCIA:Área Climatización`UTA 01 - Aire Primario`Sonda de CO2.Valor actual Valor actual;Tendencia 01
ACI`UTA01`DmpOa`DmpByPass	LUCIA:Área Climatización`UTA 01 - Aire Primario`Compuerta de aire exterior`Compuerta de bypass.Valor actual Valor actual;Tendencia
ACI`UTA01`DmpOa`DmpOa	LUCIA:Área Climatización`UTA 01 - Aire Primario`Compuerta de aire exterior`Compuerta de aire exterior.Valor actual Valor actual;Tendencia 01
ACI`UTA01`DmpOa`DmpPz	LUCIA:Área Climatización`UTA 01 - Aire Primario`Compuerta de aire exterior`Pos.compuerta Pozo.Valor actual Valor actual;Tendencia 01
ACI`UTA01`EAIErcE	LUCIA:Área Climatización`UTA 01 - Aire Primario`Entalpia despues recuperador.Valor actual Valor actual;Tendencia
ACI`UTA01`EEHE	LUCIA:Área Climatización`UTA 01 - Aire Primario`Entalpia Aire Expulsado.Valor actual Valor actual;Tendencia 01
ACI`UTA01`EOaE	LUCIA:Área Climatización`UTA 01 - Aire Primario`Entalpia de aire exterior.Valor actual Valor actual;Tendencia 01
ACI`UTA01`EPZE	LUCIA:Área Climatización`UTA 01 - Aire Primario`Entalpia Pozo Canadiense.Valor actual Valor actual;Tendencia 01
ACI`UTA01`ESuE	LUCIA:Área Climatización`UTA 01 - Aire Primario`Entalpia aire impulsión.Valor actual Valor actual;Tendencia 01
ACI`UTA01`FanEh`Veloc`AOVel	LUCIA:Área Climatización`UTA 01 - Aire Primario`Ventilador aire retorno`Regulación Velocidad f(Pd)`Salida analógica Velocidad.Valor actual Valor actual;Tendencia 01

At the bottom of the table, it indicates 'Mostrando registros del 1 al 11 de un total de 392 registros' and a pagination control showing 'Anterior', '1', '2', '3', '4', '5', '...', '36', and 'Siguiente'.

3. Datos totales

La funcionalidad asociada al capítulo 5 de la memoria la podemos encontrar en la pantalla de datos totales:

Grupo de Sistemas Inteligentes Escuela técnica superior de Ingeniería Informática ETSI Universidad de Valladolid UVa

DATOS TOTALES

Datos totales - (155 variables)

Mostrar registros Buscar:

Variable	Fecha inicial	Fecha final	Nº de registros	Unidad de medida	Mínimo	Medida
ACI' UTA01' CO2Sen	1/1/2019 1:00:00	21/5/2021 2:00:00	86041	ppm	3	451.47
ACI' UTA01' DmpOa' DmpOa	1/1/2019 1:00:00	21/5/2021 2:00:00	86875	%	0	25.72
ACI' UTA01' DmpOa' DmpPz	1/1/2019 1:00:00	21/5/2021 2:00:00	86875	%	0	22.09
ACI' UTA01' EAFercE	1/1/2019 1:00:00	21/5/2021 2:00:00	86041	kJ/kg	0	33.92
ACI' UTA01' EEhE	1/1/2019 1:00:00	21/5/2021 2:00:00	86042	Valor	3	35.37
ACI' UTA01' EOaE	1/1/2019 1:00:00	21/5/2021 2:00:00	86042	kJ/kg	-0.2	29.63
ACI' UTA01' EPZE	1/1/2019 1:00:00	21/5/2021 2:00:00	86042	kJ/kg	3	31.53
ACI' UTA01' ESuE	1/1/2019 1:00:00	21/5/2021 2:00:00	86042	kJ/kg	3	34.11
ACI' UTA01' FanSu' Veloc' AOVel	1/1/2019 1:00:00	21/5/2021 2:00:00	86911	Valor	0	33.48
ACI' UTA01' HuAEh	1/1/2019 1:00:00	21/5/2021 2:00:00	86040	g/kg	0	6.66
ACI' UTA01' HuAFerc	1/1/2019 1:00:00	21/5/2021 2:00:00	86041	%RH	0	46.92

Mostrando registros del 1 al 11 de un total de 155 registros

Anterior 1 2 3 4 5 ... 15 Siguiente

AGrupación

Tipo de agrupación

- Horas
- Días
- Semanas

FILTROS

Sistema

Señales varil... Condiciona...

Unidad de medida

°C Valor %RH bar

Rango de fecha inicial

01/01/2019 - 21/05/2021

Rango de fecha final

01/01/2019 - 21/05/2021

Desde esta pantalla se pueden aplicar filtros, eliminar *outliers*, visualizar gráficos de datos, de distribución... Además, se pueden realizar tareas de transformación de datos como puede ser la agrupación de los datos de las variables por horas, días o semanas. Una vez se han agrupados los datos se puede visualizar la matriz de correlación de Pearson para la selección de variables finales en caso de ser necesario reducir la dimensionalidad del conjunto:

Grupo de Sistemas Inteligentes Escuela técnica superior de Ingeniería Informática ETSI Universidad de Valladolid UVa

DATOS TOTALES

Datos totales - (39 variables) Matriz de correlación

	AProd' C' CtoP' Abs' TImpRef	AProd' C' CtoP' Abs' TInTwr	AProd' C' CtoP' Abs' TOutTwr	AProd' C' CtoP' Abs' TRetRef	AProd' C' CtoP' Abs' TRetTwr	AProd' C' CtoP' Ch' Tin	AProd' C' CtoP' Ch' TOut	AProd' C' CtoP' CtoAbsH' TRet	AProd' C' CtoP' CtoAbsH' TSu	AProd' C' CtosS' CL' TRet	AProd' C' CtosS' CL' TSu	AProd' C' CtosS' FncP1' TRet	AProd' C' CtosS' FncP1' TSu	AProd' C' CtosS' FncP2' TRet	AProd' C' CtosS' FncP2' TSu
AProd' C' CtoP' Abs' TImpRef	1.00	-0.64	-0.63	0.95	-0.60	0.93	0.99	-0.53	-0.62	0.75	0.65	0.64	0.38	0.75	0.31
AProd' C' CtoP' Abs' TInTwr	-0.64	1.00	0.97	-0.57	0.92	-0.50	-0.61	0.58	0.72	-0.31	-0.21	-0.11	0.13	-0.19	-0.17
AProd' C' CtoP' Abs' TOutTwr	-0.63	0.97	1.00	-0.57	0.95	-0.49	-0.60	0.55	0.72	-0.27	-0.16	-0.05	0.23	-0.13	-0.11
AProd' C' CtoP' Abs' TRetRef	0.95	-0.57	-0.57	1.00	-0.55	0.97	0.94	-0.57	-0.63	0.74	0.65	0.65	0.40	0.76	0.78
AProd' C' CtoP' Abs' TRetTwr	-0.60	0.92	0.95	-0.55	1.00	-0.47	-0.58	0.67	0.83	-0.22	-0.11	-0.03	0.24	-0.10	-0.10
AProd' C' CtoP' Ch' Tin	0.93	-0.50	-0.49	0.97	-0.47	1.00	0.93	-0.49	-0.55	0.72	0.63	0.62	0.38	0.75	0.77
AProd' C' CtoP' Ch' TOut	0.99	-0.61	-0.60	0.94	-0.58	0.93	1.00	-0.53	-0.62	0.74	0.64	0.64	0.38	0.75	0.75
AProd' C' CtoP' CtoAbsH' TRet	-0.53	0.58	0.55	-0.57	0.67	-0.49	-0.53	1.00	0.92	-0.40	-0.37	-0.34	-0.21	-0.33	-0.39
AProd' C' CtoP' CtoAbsH' TSu	-0.62	0.72	0.72	-0.63	0.83	-0.55	-0.62	0.92	1.00	-0.42	-0.36	-0.32	-0.12	-0.34	-0.39
AProd' C' CtosS' CL' TRet	0.75	-0.31	-0.27	0.74	-0.22	0.72	0.74	-0.40	-0.42	1.00	0.96	0.95	0.79	0.90	0.87
AProd' C' CtosS' CL' TSu	0.65	-0.21	-0.16	0.65	-0.11	0.63	0.64	-0.37	-0.36	0.96	1.00	0.93	0.82	0.84	0.83
AProd' C' CtosS' FncP1' TRet	0.64	-0.11	-0.05	0.65	-0.03	0.62	0.64	-0.34	-0.32	0.95	0.93	1.00	0.90	0.89	0.87
AProd' C' CtosS' FncP1' TSu	0.38	0.13	0.23	0.40	0.24	0.38	0.38	-0.21	-0.12	0.79	0.82	0.90	1.00	0.72	0.75
AProd' C' CtosS' FncP2' TRet	0.75	-0.19	-0.13	0.76	-0.10	0.75	0.75	-0.33	-0.34	0.90	0.84	0.89	0.72	1.00	0.96
AProd' C' CtosS' FncP2' TSu	0.66	-0.10	-0.08	0.66	-0.09	0.65	0.65	-0.33	-0.34	0.89	0.84	0.89	0.72	0.96	1.00

Las diferentes variables mostradas tanto en la tabla como en la matriz de correlación son seleccionables y se pueden eliminar de forma individual. Una vez finalizados los trabajos de limpieza, transformación de datos y selección de variables el conjunto de datos resultante se puede guardar.

4. Datasets generados

Los conjuntos de datos finales guardados a través de la interfaz de datos totales se pueden visualizar a través de la tabla incluida en la siguiente pantalla:

Nombre	Descripción	Tipo de agrupación	Rango de fechas	
o 2021 - Días	Aprod 19-21, Días, IQR, interpolación fechas, selección de features (+0,9)	Días	3/1/2019 1:00:00 - 21/5/2021 2:00:00	
o 2021 - Días (eliminación ausentes)	Aprod 19-21, Días, IQR, eliminación ausentes, FEATURES (+0.9)	Días	3/1/2019 1:00:00 - 21/5/2021 2:00:00	
S KNN	Aprod 19-20, HORAS, IQR, IMPUTACION KNN, FEATURES (+0.9)	Horas	3/1/2019 1:00:00 - 21/5/2021 2:00:00	
S Eliminación ausentes	Aprod 19-21, Horas, IQR, eliminación ausentes, FEATURES (+0.9)	Horas	3/1/2019 1:00:00 - 21/5/2021 2:00:00	
NAS	Aprod 19-21, SEMANAS, Interpolación, FEATURES (+0,9)	Semanas	3/1/2019 1:00:00 - 21/5/2021 2:00:00	
21 Días	Potencia 19-21, Días, IQR, KNN, features (+0,9)	Días	3/1/2019 1:00:00 - 21/5/2021 2:00:00	
21 Días - Eliminación ausentes	Potencia 19-21, Días, IQR, eliminación ausentes, features (+0,9)	Días	3/1/2019 1:00:00 - 21/5/2021 2:00:00	
as	Condiciones externas 2019 - 2021 (mayo), Z-Score, Interpolación	Días	3/1/2019 1:00:00 - 21/5/2021 2:00:00	
as - week	Condiciones externas 2019 - 2021 (mayo), semanas, Z-Score, Interpolación	Semanas	3/1/2019 1:00:00 - 21/5/2021 2:00:00	
	Load hours	Horas	3/1/2019 1:00:00 - 21/5/2021 2:00:00	

Cada *dataset* de la tabla se puede exportar en formato CSV, se puede seleccionar para visualizar su contenido en detalle o se puede eliminar.

5. Modelado

Para aplicar las técnicas de *clustering* y predicción de consumo energético mostradas en los capítulos 6 y 7 se ha generado una pantalla específica desde la que se pueden configurar ejecuciones de modelos:

Modelado

Tarea: Clustering

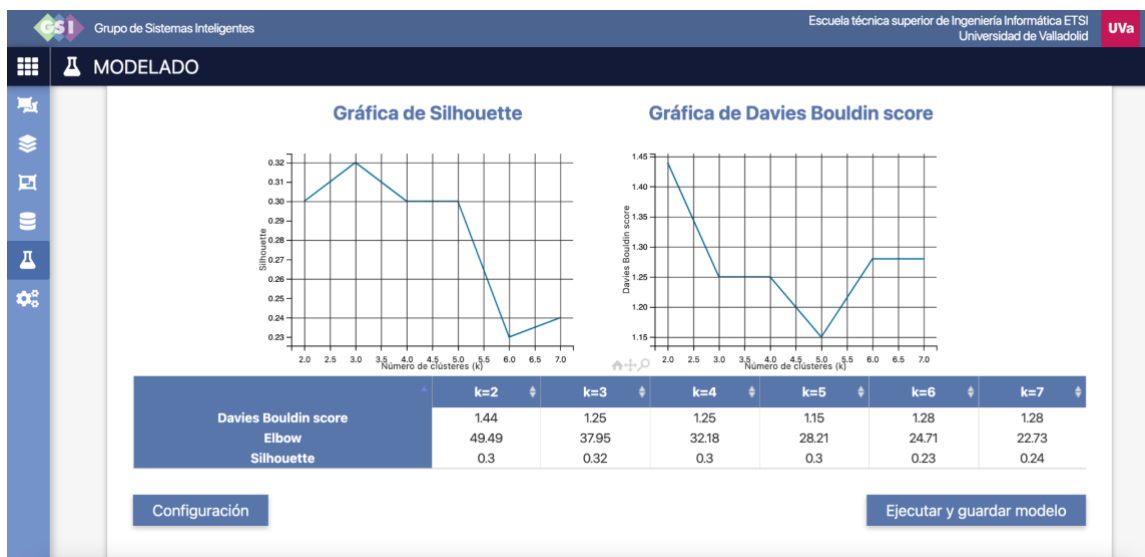
Dataset: Aprod 19-21 SEMANAS

Algoritmo: KMeans

Número de clústers (rango): Rango [2, 8]

Ejecutar test

Una vez ejecutado el test podemos ver los resultados y métricas obtenidos además de guardar el modelo con mejores resultados:



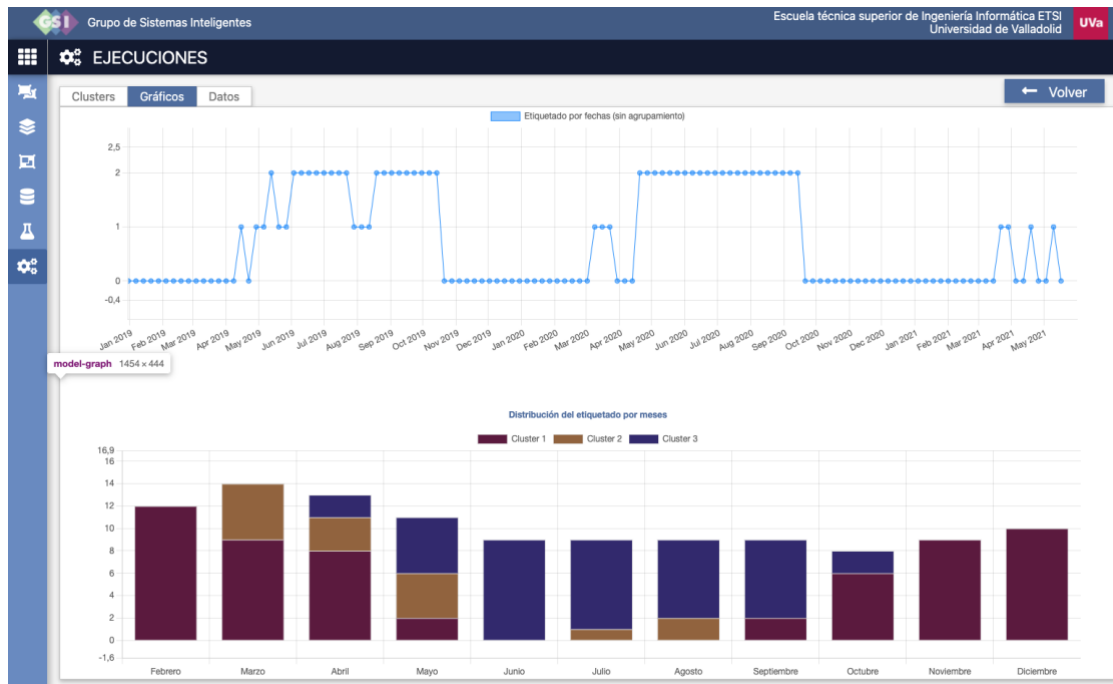
6. Ejecuciones

Los resultados de las ejecuciones guardadas a través del componente de modelado se pueden ver en la pantalla de ejecuciones:

The screenshot shows the 'EJECUCIONES' (Executions) interface. It displays a table with the following columns: idDataset, Nombre, and N° de ejecuciones. The table contains 17 rows of execution records, including dataset names and execution counts. The last three rows (14, 15, 16, 17) include icons for viewing details and deleting the execution.

idDataset	Nombre	N° de ejecuciones
1	APROD 2019-Mayo 2021 - Días	3
2	APROD 2019-Mayo 2021 - Días (eliminación ausentes)	3
3	Aprod 19-21 HORAS KNN	3
4	Aprod 19-21 HORAS Eliminación ausentes	3
5	Aprod 19-21 SEMANAS	4
14	clustering kmeans	8/12/2021 16:30:34
15	clustering dbscan	8/12/2021 16:32:42
16	clustering hierarchial	8/12/2021 16:33:29
17	clustering hierarchial	8/12/2021 16:34:20
6	Potencia - 2019-2021 Días	4
7	Potencia - 2019-2021 Días - Eliminación ausentes	3

Las diferentes ejecuciones guardadas se muestran en una tabla con la opción o bien de eliminarlas, o bien de verlas en detalle (resultados, modelo y configuraciones empleadas):



En las diferentes pestañas (van a depender de si la ejecución es de un modelo de *clustering* o de predicción de series temporales) que podemos encontrar para cada ejecución se incluyen diferentes gráficos para ayudar a la comprensión y visualización de los resultados obtenidos con los modelos.