

**Universidad Internacional de La Rioja (UNIR)**

**ESIT**

**Máster Universitario en Industria 4.0**

Implementación de  
un sistema en la  
nube para el ahorro  
de consumo de  
energía eléctrica.

**Trabajo Fin de Máster**

**Presentado por:** Vaccaro Cedillo, Christopher Javier

**Directora:** Díez Platas, María Luisa

## Resumen

Debido a los últimos eventos suscitados por la pandemia a nivel mundial, la economía se ha visto fuertemente afectada en empresas e instituciones que manejaban una modalidad presencial y que además involucran un grupo masivo de personas, como es el caso de universidades. Para ello el presente trabajo se enfoca en uno de los principales egresos, el consumo energético, desarrollando un sistema de monitorización, control y predicción mediante sistemas basados en la nube para generar un ahorro en la factura de consumo. El sistema será accesible desde una página web desarrollada con herramientas de libre distribución, que sea compatible con el hardware especializado para el control de equipos de forma remota, además de un aprendizaje automático que le permita al usuario visualizar un consumo predictivo para que pueda tomar medidas de ahorro energético de forma oportuna. Dependiendo del tipo de infraestructura que posea la universidad la solución puede variar ligeramente, sin embargo, se evalúan diferentes opciones a nivel de software y hardware que sean compatibles con la mayoría de las instituciones universitarias.

**Palabras clave:** Inmótica, sensores, ahorro energético, aprendizaje supervisado, algoritmo predictivo

## Abstract

Due to the latest events caused by the pandemic worldwide, the economy has been strongly affected in companies and institutions that handled a face-to-face modality and that also involve a massive group of people, such as universities. For that, this work focuses on one of the main expenses, energy consumption, developing a monitoring, control and prediction system using cloud-based systems to generate savings in the consumption bill. The system will be accessible from a web page developed with free distribution tools, that is compatible with specialized hardware for remote control of equipment, as well as automatic learning that allows the user to visualize predictive consumption so that they can take measures of energy savings in a timely manner. Depending on the type of infrastructure that the university has, the solution may vary slightly, however, different options are evaluated at the software and hardware level that are compatible with most university institutions.

**Keywords:** *Inmotics, sensors, energy saving, supervised learning, predictive algorithm*

# Índice de contenidos

1. Introducción .....	9
1.1. Motivación .....	9
1.2. Planteamiento del trabajo .....	11
1.3. Estructura de capítulos .....	12
2. Contexto y estado del arte .....	13
2.1. Descripción general del contexto del proyecto .....	13
2.2. Proyectos relacionados con el tema del TFM .....	15
2.2.1. Soluciones inteligentes al consumo energético .....	15
2.2.2. Aplicaciones móviles/web orientadas al ahorro energético .....	18
2.2.3. El uso de machine learning para alcanzar la eficiencia energética .....	19
2.3. Tecnologías relacionadas con el tema del TFM .....	20
2.3.1. Sensores IoT .....	21
2.3.2. Actuadores IoT .....	22
2.3.3. Protocolos de comunicación a través de IP (Internet Protocol) .....	23
2.3.4. Plataformas IoT .....	24
2.3.5. Aprendizaje Automático .....	25
2.4. Conclusiones sobre el estado del arte .....	28
3. Descripción general de la contribución del TFM .....	29
3.1. Objetivos .....	29
3.2. Metodología del trabajo .....	29
3.2.1. Planificación .....	30
3.2.2. Diseño .....	30
3.2.3. Implantación .....	30
3.2.4. Ejecución .....	31
3.3. Descripción general de las partes o componentes de la propuesta .....	31
- Actividad A .....	37
- Actividad B .....	37

- Actividad C .....	37
- Actividad D .....	37
- Actividad E .....	37
- Actividad F .....	37
4. Desarrollo específico de la contribución.....	40
4.1. Planificación .....	40
4.2. Diseño .....	41
4.3. Implantación.....	43
4.3.1. Evaluación y selección del sensor IoT .....	43
4.3.2. Adquisición de datos .....	44
4.3.3. Selección de aprendizaje automático .....	44
4.3.4. Desarrollo de plataforma IoT .....	50
4.3.5. Desarrollo de hardware para control remoto .....	56
4.4. Ejecución.....	57
4.4.1. Análisis del modelo predictivo ARIMA .....	57
4.4.2. Análisis del modelo de clasificación.....	58
4.4.3. Análisis de escenario de consumo excedido.....	59
5. Conclusiones y trabajos futuros.....	59
5.1. “Líneas de trabajo futuras”.....	61
Referencias bibliográficas .....	62
Anexo A Código implementado para adquisición de datos .....	65
Anexo B Datos de entrenamiento Grupo 1 .....	66
Anexo C Datos de entrenamiento Grupo 2.....	67
Anexo D Muestra de datos de entrenamiento para árbol de decisiones .....	68
Anexo E Código implementado para entrenamiento del árbol de decisión .....	70
Anexo F Árbol de decisión sin límites de profundidad.....	71
Anexo G Imágenes de navegación en ThingsBoard .....	72

## Índice de figuras

Figura 1. Demanda horaria/mensual (kWh/mes) .....	13
Figura 2. Vista de tablas y gráficos del sensor SMAPPEE .....	14
Figura 3. Vista principal de la plataforma DINOMI .....	14
Figura 4. Tabla de mediciones de la plataforma DINOMI .....	15
Figura 5. Adopción de la Nube pública para empresas.....	18
Figura 6. Evolución de las revoluciones industriales.....	20
Figura 7. Tipos de Sensores .....	21
Figura 8. Sensores de corriente .....	21
Figura 9. Consumo de energía vs Recursos requeridos .....	24
Figura 10. Fases de la metodología cascada.....	30
Figura 11. Arquitectura completa de la solución.....	34
Figura 12. Comportamiento del usuario frente a la solución.....	41
Figura 13. Arquitectura de infraestructura en plataforma IoT .....	42
Figura 14. Almacenamiento de datos en servidor local.....	44
Figura 15. Consumo energético de ESPOL (2017-2021) .....	45
Figura 16. Datos desestacionalizados, tendencia lineal.....	47
Figura 17. Algoritmo predictivo sin considerar datos de pandemia .....	47
Figura 18. Algoritmo predictivo considerando datos durante pandemia .....	48
Figura 19. Árbol de decisión con máxima profundidad de 7 .....	50
Figura 20. Página inicial de ThingsBoard .....	51
Figura 21. Página inicial de ThingsBoard con datos predictivos.....	52
Figura 22. Ventana a nivel de Laboratorio.....	52
Figura 23. Cadena de regla raíz .....	53
Figura 24. Cadena “ <i>createAsset</i> ” .....	53
Figura 25. Cadena “ <i>Relay</i> ” .....	54
Figura 26. Cadena “ <i>Alarma</i> ” .....	54
Figura 27. Cadena “ <i>validateID</i> ” .....	55

Figura 28. Cadena “ <i>updateParent</i> ” .....	55
Figura 29. Proceso de aprendizaje de comandos infrarrojos .....	56
Figura 30. Shelly instalado en panel eléctrico .....	57
Figura 31. Dispositivo IoT controlando equipos.....	57

## Índice de tablas

Tabla 1. QS Ranking Latinoamérica 2020 .....	9
Tabla 2. Tasa de crecimiento en exportaciones. Pronóstico 2020 (Variación porcentual).....	10
Tabla 3. Enfoques para medir el uso real de energía .....	16
Tabla 4. Tecnologías inteligentes para ahorro energético .....	17
Tabla 5. Ahorro energético en edificios por tecnologías inteligentes .....	17
Tabla 6. Comparación de modelos predictivos.....	19
Tabla 7. Comparación de sistemas de monitorización IoT .....	23
Tabla 8. Modelos de aprendizaje automático .....	26
Tabla 9. Comparación cuantitativa entre modelos predictivos de series de tiempo.....	27
Tabla 10. Costo de Hardware .....	35
Tabla 11. Costo de Software.....	35
Tabla 12. Costo de desarrollo .....	36
Tabla 13. Costo final.....	36
Tabla 14. Tiempo y dependencias de actividades generales .....	38
Tabla 15. Tiempo y dependencias de tareas específicas .....	38
Tabla 16. Requerimientos no funcionales.....	40
Tabla 17. Resumen de requerimientos funcionales .....	41
Tabla 18. Factores estacionales grupo 1 .....	46
Tabla 19. Factores estacionales grupo 2.....	48
Tabla 20. Variable independientes para algoritmo clasificadorio .....	49
Tabla 21. Errores porcentuales entre consumos reales y consumos calculados.....	58

## 1. Introducción

El presente trabajo consiste en interrelacionar tecnología existente para cambiar procesos internos y externos de un campus universitario y de esta manera poder evidenciar un impacto positivo en la facturación mediante un sistema que permita la monitorización, control y predicción de consumo eléctrico a nivel de aulas o laboratorios.

### 1.1. Motivación

La Escuela Superior Politécnica del Litoral (ESPOL)<sup>1</sup> es una institución pública de educación superior, categoría “A” a nivel nacional, considerada entre las mejores universidades de Latinoamérica por el Ranking Quacquarelli Symonds en 2020 (Top 100) como se puede apreciar en la Tabla 1, cuenta con un prestigio histórico debido a su innovación constante y la adaptabilidad a los cambios tecnológicos.

Tabla 1. QS Ranking Latinoamérica 2020

Rank	Universidad	País	Puntaje general
1	Pontificia Universidad Católica de Chile (UC)	Chile	100
2	Universidade de São Paulo	Brasil	99.4
3	Tecnológico de Monterrey	México	98.4
<b>64</b>	<b>Escuela Superior Politécnica del Litoral (ESPOL)</b>	<b>Ecuador</b>	<b>52.2</b>
100	Universidad Tecnológica Nacional (UTN)	Argentina	43.3

Fuente: Ranking Quacquarelli Symonds

En ESPOL laboran aproximadamente mil quinientos empleados, servidores públicos repartidos en las diferentes áreas de investigación docente, formación de líderes, vinculación con la sociedad, investigación, desarrollo e innovación; en cuanto a la población estudiantil, ESPOL abarca aproximadamente a once mil estudiantes de pregrado distribuidos en sus ocho unidades académicas (facultades) que imparten carreras afines a la ciencia e Ingeniería, Educación Comercial, Salud, Servicios, Humanidades, Arte y Diseño.

Según un informe de la Comisión Económica para América Latina y el Caribe (CEPAL, 2020), el 11 de marzo del 2020 la Organización Mundial de la Salud (OMS) declara que la enfermedad por el Coronavirus (COVID 19) es considerada pandemia, esto generó que los gobiernos empiecen a tomar medidas como estado de excepción, restricciones vehiculares, cuarentena, entre otras. Por esta crisis sanitaria se vieron paralizadas muchas de las

---

<sup>1</sup> **Sitio Web:** <https://www.espol.edu.ec/>

actividades cotidianas, generando además una crisis económica en el mundo. De acuerdo con esto, los países exportadores de materia prima son los más afectados. Se puede observar un pronóstico de las tasas de crecimiento en exportaciones para el 2020 en la Tabla 2.

Tabla 2. Tasa de crecimiento en exportaciones. Pronóstico 2020 (Variación porcentual)

Región	Volumen	Precio	Valor
<b>América Latina y el Caribe</b>	<b>-6.0</b>	<b>-8.8</b>	<b>-14.8</b>
Exportadores de petróleo	-4.7	-14.6	-19.2
Exportadores de minerales	-7.4	-9.3	-16.7
Exportadores de productos agroindustriales	-6.2	-4.0	-10.2

**Fuente:** Comisión Económica para América Latina y el Caribe (CEPAL)

Otro de los sectores afectados en el país por esta crisis es el de la educación pública, ESPOL no fue la excepción al sufrir dos recortes presupuestarios en el año 2020, comprometiendo la cantidad de profesores contratados y la calidad de educación. Es por esto que ESPOL se vio en la necesidad de reducir y controlar de mejor forma sus egresos mensuales y anuales.

Uno de los egresos mensuales que fue analizado por el departamento de Sostenibilidad de ESPOL, fue el del consumo de energía eléctrica. No se dispone de la información necesaria para poder identificar los diferentes comportamientos de consumo y las acciones que pueden contribuir a utilizar de forma más eficiente la energía dentro del campus.

Por otro lado, el confinamiento también afectó a trabajadores de empresas públicas del servicio eléctrico a nivel nacional, como el caso de la Corporación Nacional de Electricidad (CNEL)<sup>2</sup> quienes tuvieron inconvenientes al momento de facturar los consumos, ocasionando una serie de quejas en el país por parte de los usuarios llegando a más de 68.000 reclamos, informado por uno de los diarios más importantes a nivel nacional en su página web (C.A. EL UNIVERSO, 2021). Esto demuestra la importancia y necesidad de contar al menos con un sistema de monitoreo en infraestructuras que tienen un consumo considerable, como es el caso de las universidades que cuentan con varios edificios, pudiéndose ahorrar miles de dólares en el consumo eléctrico; si a esto se le suma una visualización de consumo predictivo, el personal de la universidad tendría una mejor noción del desperdicio que genera, tomando mayor conciencia en el ahorro energético. Finalmente, al darle la comodidad de hacerlo remotamente incrementaría el ahorro evitando consumos innecesarios por olvido involuntario pudiéndose apagar equipos sin necesidad de estar físicamente presente en el sitio.

---

<sup>2</sup> **Sitio Web:** <https://www.cnelep.gob.ec/>

## 1.2. Planteamiento del trabajo

Mediante entrevistas con el personal del “Centro de Energías Renovables y Alternativas” (CERA) y el personal de mantenimiento eléctrico de ESPOL, se conoce que existen estudios e investigaciones en los cuales, mediante la monitorización inteligente de varios edificios de ESPOL, se ha identificado 4 categorías generales de consumo fuertemente relacionadas con los usuarios, las dos más relevantes son el uso de aires acondicionados y todo el consumo referente a iluminación.

La estrategia integral involucra la interacción entre la tecnología sumada al comportamiento de los usuarios con el fin de generar un ahorro energético, lo que implica una reducción en la factura mensual a pagar. Esta tecnología incluye componentes de hardware y software distribuidos en cada edificio de la universidad; esto permitirá a los usuarios ver su consumo en tiempo real y cuál es el consumo que se esperaría generar a lo largo del mes. Con esta información los usuarios podrán tener un mejor control del consumo energético, además de saber qué áreas están generando un posible consumo innecesario, ya sea por equipos antiguos o por desperdicios inadvertidos.

La solución que se propone ante esta problemática es implementar un sistema de internet de las cosas, también llamado sistema IoT por sus siglas en inglés *Internet of Things*, el cual está conformado por diferentes componentes. A nivel de monitorización se dispone de una página web de código abierto almacenada en una plataforma en la nube, la cual no genera egresos adicionales a la universidad y a su vez cumple con las funciones necesarias detalladas posteriormente. Esto implica que el sistema debe ser capaz de mostrar el consumo energético en tiempo real, además de predicciones mensuales que permitan al usuario mantener un límite de consumo, para ello el usuario tendrá la facilidad de poder controlar tanto el sistema de luminarias como los aires acondicionados o cualquier dispositivo con tecnología infrarroja, todo esto será de forma remota ayudando a evitar consumos innecesarios fuera de horario laboral.

Para el control de equipos eléctricos de forma remota, el sistema los segmenta en dos categorías: dispositivos infrarrojos y sistema de luces. Esta división surge del comportamiento eléctrico de los mismos, por ejemplo, una luz de techo puede apagarse al quitarle la energía eléctrica por completo y no sufrirá daño alguno, por otro lado, si se realiza lo mismo para un aire acondicionado, proyector o televisor con el tiempo la electrónica podría dañarse, lo ideal es que estos equipos se apaguen mediante comandos infrarrojos a través de un dispositivo emisor genérico que tenga acceso a la nube, conocido también como dispositivo IoT. Estos dispositivos IoT serán los encargados de controlar de forma remota el encendido y apagado de las luces y cualquier equipo infrarrojo, concretamente los aires acondicionados.

## 1.3. Estructura de capítulos

### 1. Introducción

En este capítulo se describe la justificación del trabajo, el por qué es necesario su implementación e investigación, definiendo de manera clara el problema encontrado actualmente y cómo se lo pretende solucionar dando ideas generales del proyecto.

### 2. Contexto y estado del arte

En esta sección se analiza estado actual del problema, enfocado en el lugar donde se realizará la implementación de la solución, también se realiza una investigación teórica de métodos similares o que traten la problemática abordada en el capítulo anterior para que no haya una repetición de métodos, sino que se genere una solución innovadora o complementaria a lo ya existente. Para este trabajo el estado del arte se enfoca en la importancia de una monitorización, control y predicción, cómo estas tres acciones conllevan a un ahorro energético de manera directa e indirecta. También se analiza los diferentes algoritmos utilizados para predecir el consumo energético.

### 3. Descripción general de la contribución del TFM

Aquí se detalla tanto el objetivo general como los específicos, los cuales serán importantes al momento de evaluar los resultados del trabajo, además se detalla la metodología a utilizar; por último, se mostrará el contenido del trabajo, incluyendo todas las tecnologías que se van a implementar, qué resultados se espera obtener, el presupuesto invertido y la planificación.

### 4. Desarrollo específico de la contribución

Corresponde a la descripción más detallada del capítulo anterior, se menciona en más profundidad todos los componentes que se implementarán en la solución, tanto de software como de hardware, es decir los requisitos necesarios para el desarrollo; luego cómo se desarrolla la parte de software, la parte de hardware y cómo se hace la integración entre estos dos componentes y finalmente la evaluación de la solución a través de evidencias de los resultados obtenidos a corto plazo y resultados tentativos a largo plazo.

### 5. Conclusiones y trabajos futuros

En este segmento final se incluye un resumen de los resultados obtenidos en el capítulo anterior y cómo estos pudieron alcanzar los objetivos planteados al inicio del documento, dado que existe un alcance y limitaciones de la solución. En este capítulo también se detalla las posibles mejoras o posibles cambios que se pueden abarcar, analizando escenarios distintos a los planteados en el presente trabajo.

## 2. Contexto y estado del arte

### 2.1. Descripción general del contexto del proyecto

Como se mencionó en el capítulo anterior, ESPOL necesita reducir sus consumos para ahorrar dinero y evitar los recortes de proyectos y de personal. Antes del 2019 no existía un sistema que monitorizara el consumo eléctrico del campus en tiempo real, sólo disponía de las facturas emitidas por la empresa eléctrica cada mes a través de un único medidor tradicional, obteniendo una monitorización muy genérica como se observa en la **Figura 1**.

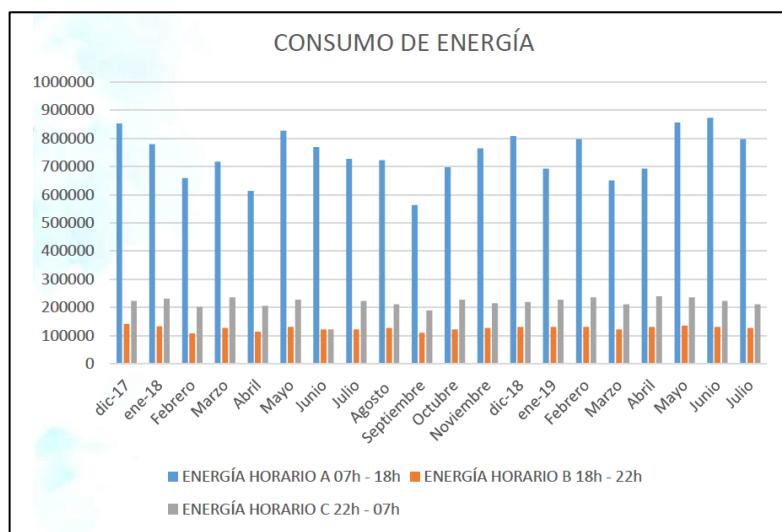


Figura 1. Demanda horaria/mensual (kWh/mes)

**Fuente:** Departamento de Sostenibilidad ESPOL

Las autoridades de la Institución, al notar esta falta de información sobre el consumo energético, decidieron realizar la adquisición de un sensor de energía eléctrica capaz de monitorizar en tiempo real el consumo de un laboratorio y a la vez que lo muestre en una plataforma web. El edificio seleccionado para este proyecto piloto fue el Centro de Energías Renovables y Alternativas (CERA), para el cual se contrató el servicio de la marca SMAPPEE por un año obteniendo buenos resultados, dicho servicio constaba de un medidor capaz de sensar la corriente y voltaje alterno, además de determinar parámetros eléctricos como factor de potencia, energía activa y reactiva, entre otros; estos valores de interés son enviados vía ethernet (o Wifi) a una base de datos de la empresa SMAPPEE, los cuales pueden ser visualizados en una plataforma web por el usuario, quien tiene la opción de seleccionar rango de datos y/o fechas en base a sus propios requerimientos, esta información es observada mediante tablas y/o gráficos lineales (Figura 2) que también pueden ser exportados en archivos CSV, PDF y TXT, además de otras funciones adicionales que no aportan al interés de la Institución.

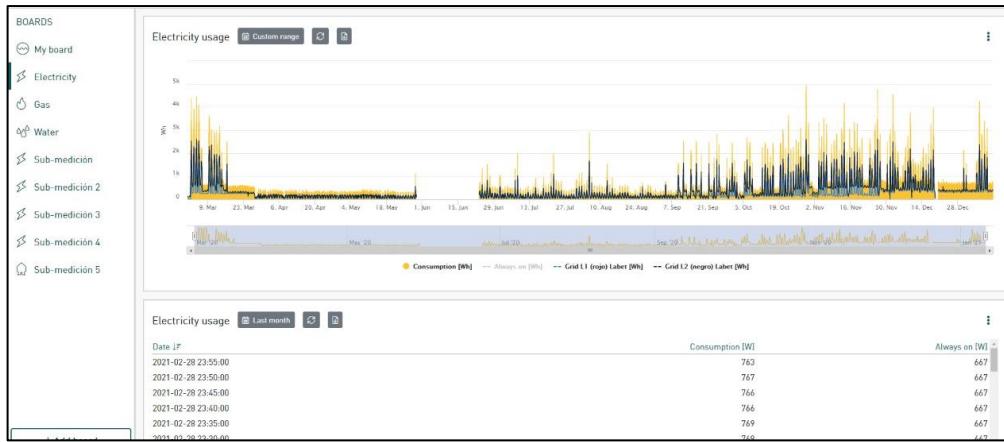


Figura 2. Vista de tablas y gráficos del sensor SMAPPEE

**Fuente:** Elaboración propia

Luego del año de pruebas se obtuvieron buenos resultados; sin embargo, instalar un SMAPPEE en cada uno de los 85 edificios que dispone ESPOL, hacían que la solución no sea económicamente viable, por lo que se decidió realizar una compra de sensores inteligentes similares al SMAPPEE, pero de menor precio y que cumpla con las características principales (sensado y monitorización en tiempo real). Se contrató el servicio de la empresa DINOMI como se observa en la **Figura 3**, pero al ser un modelo más económico, la plataforma de monitorización tiene más limitaciones, como por ejemplo, las gráficas lineales sólo muestran las últimas 24h de sensado, si se desea un rango diferente se lo debe realizar desde otro apartado, aunque no será una visualización gráfica como se muestra en la **Figura 4** sino por tablas; en contraposición como ventaja, estos sensores pueden enviar datos a través del protocolo MQTT (*Message Queue Telemetry Transport*) hacia algún servidor propio del cliente, de esta manera se puede obtener la información de los sensores y desarrollar un sistema IoT que se adapte a las necesidades específicas de la Institución.

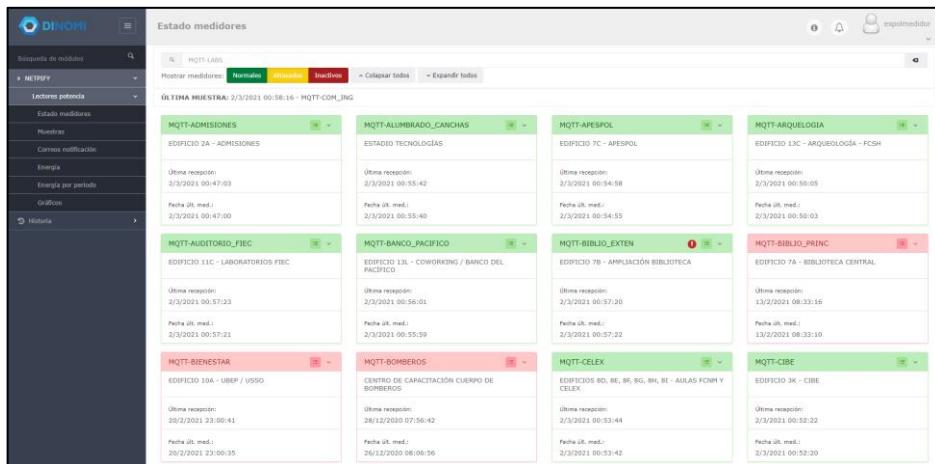


Figura 3. Vista principal de la plataforma DINOMI

**Fuente:** Elaboración propia

Figura 4. Tabla de mediciones de la plataforma DINOMI

Fuente: Elaboración propia

## 2.2. Proyectos relacionados con el tema del TFM

Luego del análisis previo de lo que actualmente se tiene implementado en ESPOL, es necesario revisar proyectos similares para generar posibles mejoras, con el fin de conseguir un óptimo ahorro energético y que a su vez sea de bajo costo.

### 2.2.1. Soluciones inteligentes al consumo energético

El problema principal en el que se enfoca resolver este trabajo, corresponde al uso innecesario de energía eléctrica por desconocimiento por parte de los usuarios, en la mayoría de casos la percepción de las personas hacia el consumo energético tiende a ser errónea, esto se menciona en el artículo (Lesic, de Bruin, Davis, Krishnamurti, & Azevedo, 2018), el cual indica que las personas tienden a sobreestimar el uso de electricidad para aparatos de bajo consumo y subestimar el de aparatos de alto consumo, esto ocurre porque no se tiene un medio para entender de mejor manera el comportamiento de los equipos de uso diario; los mismos autores basados en investigaciones científicas y bajo una apreciación subjetiva, calificaron diferentes enfoques para la medición del uso real de energía mostrado en la **Tabla 3**; se observa cuatro tipos de medición, las dos primeras estimaciones no son exactas porque se basan en casos generales y en una autoapreciación de consumo por parte de los usuarios, aunque la complejidad es baja al igual que el costo de medición, obtener datos de ambas maneras no permitirían realizar acciones específicas acertadas para ahorrar energía. La tercera forma mencionada sería ideal para casos domésticos, donde el consumo se limita a una sola residencia familiar; sin embargo, el enfoque de este trabajo es dirigido hacia edificios de una universidad, por lo cual usar sensores domésticos en un campus universitario daría una baja precisión en la medición, además de que no soportarían corrientes elevadas. Finalmente, con la última forma se tendría una alta precisión en los datos, aunque con información algo compleja y costosa, pero mediante diferentes técnicas de filtrado de información e implementación de tecnologías libres se podría manejar estas desventajas.

Tabla 3. Enfoques para medir el uso real de energía

	Accesibilidad a los datos	Costo de medición	Precisión de los datos	Complejidad de los datos
Estimaciones generales de la literatura existente.	Muy alto	Muy bajo	Bajo	Bajo
Estimaciones basadas en el uso de energía por autoapreciación.	Medio	Bajo	Bajo	Bajo
Estimaciones basadas en lecturas de medidores a nivel doméstico	Medio	Medio	Medio	Alto
Medidas de consumo de energía en tiempo real a partir de medidores inteligentes	Muy bajo	Muy alto	Muy alto	Muy alto

Fuente: Traducción de (Lesic, de Bruin, Davis, Krishnamurti, & Azevedo, 2018)

La clave principal para conseguir un ahorro energético relevante es un sistema que monitorice en tiempo real el consumo de energía, y que la recolección de datos sea mostrada de manera entendible a las personas que generan directamente este consumo. Tal como se menciona en el artículo (Feltham, 2014); un punto crítico es determinar los usuarios que tendrían acceso a este sistema de monitorización, y al mismo tiempo determinar cuánta información sería útil mostrar.

El inconveniente con la mayoría de medidores inteligentes que incorporan un sistema de monitorización, es que muestran de manera detallada muchos parámetros eléctricos que sí pueden ser interpretados por un técnico, pero no necesariamente por un consumidor general, lo cual no generaría un gran impacto y por ende el consumidor general perdería el interés, otro punto a considerar es el nivel de granularidad, es decir cuan específica será la medición, si a nivel de edificio, de habitación, de equipos finales, etc.

En (Karjalainen, 2011) se demuestra que una monitorización muy específica a nivel individual tiende a alterar negativamente el comportamiento del usuario, porque lo considera algo invasivo para su privacidad. En lugar de ello, se recomienda medir el uso de energía a nivel de categorías, como consumo de luces, consumo de aires acondicionados, entre otras. De esta manera se pretende identificar los equipos o áreas que consumen la mayor parte de la energía eléctrica y con esta información poder implementar una o varias soluciones acorde con la infraestructura, que van desde reemplazar equipos antiguos o dañados, hasta cambiar el comportamiento del consumidor final que está directamente relacionado. De esta forma, se consigue realizar un ahorro energético más específico y centralizado sin necesidad de invadir la privacidad del usuario.

Aun cuando sólo la monitorización genera un ahorro energético, también se puede adicionar un control inteligente para incrementar este ahorro. Con la nueva revolución industrial 4.0, el tema de edificios inteligentes ha ganado más interés por parte de las personas por ser algo innovador y de vanguardia. Pero más interés genera el poder ahorrar energía eléctrica, está demostrado que implementar tecnologías IoT genera un ahorro energético significativo como se muestra en la [¡Error! No se encuentra el origen de la referencia..](#)

Tabla 4. Tecnologías inteligentes para ahorro energético

Sistema	Tecnología	Ahorro energético
Calefacción, ventilación y aire acondicionado	Variador de frecuencia	15% – 50%
Calefacción, ventilación y aire acondicionado	Termostato inteligente	5% – 10%
Tomacorriente	Tomacorriente inteligente	50% – 60%
Tomacorriente	Regleta inteligente	25% – 50%
Luces	Control de luces avanzado	45%
Luces	Sistema de administración web de luces	20% – 30%

Fuente: (King & Perry, 2017)

Dependiendo del tipo de infraestructura ciertas tecnologías serán más adecuadas respecto a otras. En el caso específico de universidades, será necesario analizar el tipo de edificio de cada área, puede ser edificio orientado a la educación (aulas), edificio administrativo (oficinas) o edificios de laboratorios.

En la Tabla 5, se presenta un resumen de las mejores tecnologías aplicadas y cuánto es el ahorro energético aproximado [Tabla 5.](#)

Tabla 5. Ahorro energético en edificios por tecnologías inteligentes

Tipo de edificio	Superficie del edificio (m <sup>2</sup> )	Tecnología inteligente	Energía promedio consumida (kWh/año)	Porcentaje de ahorro
Educación	100000	Sistema web para control de luces	190000	11%
Oficinas	50000	Sistema de control remoto para aires acondicionados	850000	23%
Laboratorios	70000	Sensores de presencia Controladores en tiempo real	980000	40%

Fuente: (King & Perry, 2017)

## 2.2.2. Aplicaciones móviles/web orientadas al ahorro energético

Lo antes mencionado demuestra la estrecha relación que debe existir entre un sistema de monitorización y un sistema de control inteligente cuando se está buscando tener un ahorro eficiente de energía eléctrica; esta relación puede ser establecida mediante una aplicación móvil o web para la monitorización y control.

Existen proyectos de aplicaciones móviles (Forcén, 2020) que se centran en reducir la huella ecológica, el cual es un tema muy relacionado con el ahorro energético. Estos trabajos de diseño muestran claramente el interés de las personas por cuidar el medio ambiente, lo que indica que existe una alta probabilidad de aceptación para este tipo de aplicaciones que aparte de ahorrar dinero, también cuidan del medio ambiente. La aceptación de este tipo de aplicaciones no solo ocurre a nivel de usuario final, sino también a nivel industrial en la monitorización de centrales eléctricas (Caicedo, 2020) que genera grandes ingresos haciéndolo un proyecto económicamente viable.

Por otro lado, hay que considerar la ciberseguridad. Se puede decir que las tecnologías IoT son herramientas de doble filo, a pesar de que ofrecen muchas ventajas en los procesos industriales y domésticos, también generan nuevas vulnerabilidades que podrían comprometer la integridad de información privada, especialmente en el caso de aplicaciones web. Un estudio realizado en Ecuador (Desiderio, 2020) muestra una de las tantas opciones que se pueden ejecutar para mejorar la seguridad a nivel de plataformas web, que incluye utilizar herramientas de escaneo frente ataques de tipo “SQL injection” eficientes en sistemas conectados con alguna base de datos dentro de servidores locales. Otra solución viable es usar tecnologías en la nube a nivel de “infraestructura como servicio” o IaaS por sus siglas en inglés *Infrastructure as a Service*, como por ejemplo Microsoft Azure, Amazon Web Services (AWS), Google Cloud, entre otras, que se encargan de la ciberseguridad de forma transparente al cliente. Cada una de estas tecnologías en la nube tienen pequeñas diferencias que irán acorde a las necesidades específicas de cada empresa. En la **Figura 5** se muestra el uso de estas tecnologías en el 2020.

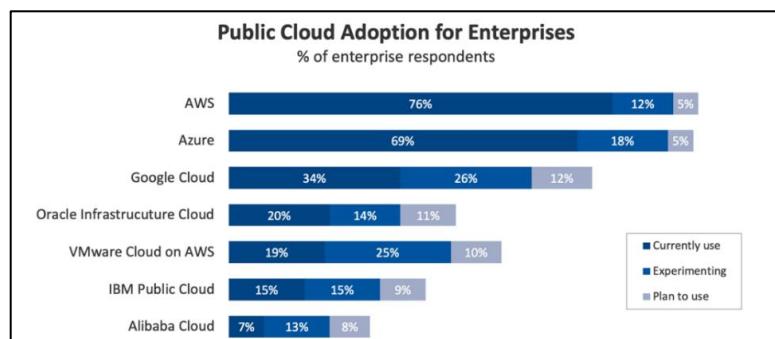


Figura 5. Adopción de la Nube pública para empresas

Fuente: Flexera 2020 State of the Cloud Report

La inclinación de las empresas por usar infraestructura en la nube se debe principalmente al ahorro de costos y la capacidad de escalar en infraestructura fácilmente. Incluso hay trabajos (Jiang, Perng, Li, & Chang, 2012) que muestran procesos para obtener una óptima eficiencia en la administración de este recurso en la nube, lo que se debe a la variación de las necesidades de la empresa dando la posibilidad de crecer rápidamente y que los costos sean acordes con lo implementado.

### 2.2.3. El uso de machine learning para alcanzar la eficiencia energética

Con el rápido crecimiento poblacional, uno de los problemas que se deben resolver a nivel mundial es la demanda de energía eléctrica en los próximos años. Un estudio realizado en China (Huang, Tang, & Chen, 2018) analiza las ventajas y posibles algoritmos para realizar esta predicción de consumo y poder tomar medidas a tiempo; un caso similar ocurre en Corea del Sur, donde se ha hecho un estudio centrándose en la predicción mediante redes neuronales (Woo & Roper, 2009) considerando parámetros como el producto interno bruto (PIB), población, importaciones y exportaciones. Estos estudios demuestran claramente los beneficios a nivel de países sobre predicciones de consumo energético. Para casos más pequeños y específicos como el de una industria también existen artículos de investigación (Castorena, Silva, Martínez, Campos, & Martínez, 2020) en las que concluyen que, para evitar pérdidas económicas, no solo basta la monitorización y control sino también una predicción de estos consumos.

Existen varios métodos predictivos. En la **Tabla 6** se muestran las ventajas y desventajas de dos modelos predictivos generales. Dependiendo de la situación será conveniente escoger uno de ellos, dentro de estos dos modelos generales existen métodos más específicos descritos en (Deb, Zhang, Yang, Eang, & Wei, 2017). Por ejemplo, en el caso de predicciones basadas en datos anteriores se cuenta con: redes neuronales, máquinas de vectores de soporte, series de tiempo difusas, entre otras.

Tabla 6. Comparación de modelos predictivos

Técnica	Ventajas	Desventajas
Predicciones basadas en datos pasados	<ul style="list-style-type: none"><li>• Rápido procesamiento con datos en tiempo real</li><li>• Adecuado para modelos no lineales</li><li>• Más precisos que modelos deterministas</li></ul>	<ul style="list-style-type: none"><li>• Requiere datos pasados</li><li>• Difícil obtener un modelo general</li></ul>
Predicciones determinísticas	<ul style="list-style-type: none"><li>• No necesita de entrenamiento</li><li>• Fácil de obtener un modelo general</li></ul>	<ul style="list-style-type: none"><li>• Difícil para modelar escenarios reales</li><li>• No dispone de parámetros de la infraestructura</li><li>• No tan preciso</li></ul>

**Fuente:** (Deb, Zhang, Yang, Eang, & Wei, 2017)

## 2.3. Tecnologías relacionadas con el tema del TFM

Basado en los proyectos anteriores se pueden apreciar ciertas tecnologías en común que están muy relacionadas con la Revolución Industrial 4.0.

En la actualidad, con la cuarta revolución industrial y el crecimiento de las tecnologías de información muchos procesos han quedado obsoletos mientras otros han tenido mejoras significativas, lo que ha dado paso a innumerables soluciones más eficientes y eficaces, tal es el caso de distintos sectores como automotriz, agrícola, medicina, etc.

Aunque en América Latina recién se está normalizando el uso de soluciones relacionadas con la Industria 4.0, en otros continentes como Europa ya empieza a mostrar grandes diferencias en la contribución directa al PIB (13%) y un incremento del 10% en empleos. (Román, 2006)

En la **Figura 6** se muestra la evolución a lo largo del tiempo de las diferentes revoluciones industriales en la historia, actualmente la Industria 4.0. Esta revolución se caracteriza por la interconectividad de las cosas en tiempo real, ha traído nuevos conceptos y hecho tendencias otros, como es el caso de sistemas ciberfísicos, Internet de las Cosas, *Big Data*, Aprendizaje automático y Computación en la nube.

Al comparar la Industria 4.0 con la 3.0 se puede observar algunos beneficios nuevos; al estar todos los sistemas interconectados en tiempo real es fácil observar un incremento de producción en los procesos, lo cual influye en reducción de costes y sobre todo en la calidad de un producto. Esta calidad también será mejorada por algoritmos de aprendizaje automático al poder predecir productos defectuosos o tiempos de vida útil. No solo las máquinas están conectadas entre ellas en tiempo real, también lo están con software de monitorización incrementando los beneficios aún más.

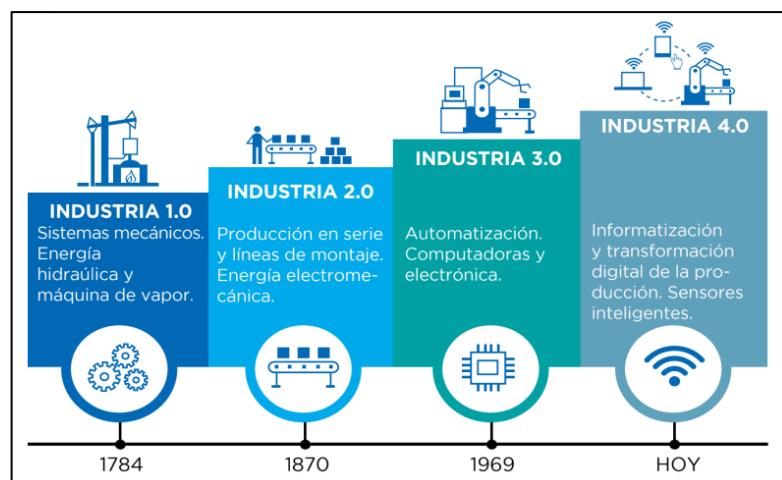


Figura 6. Evolución de las revoluciones industriales

**Fuente:** (Basco, Beliz, Coatz, & Garnero, 2018)

### 2.3.1. Sensores IoT

Para poder implementar un sistema de ahorro energético, un habilitador esencial son los sensores IoT, no es suficiente con tener medidores de consumo energético, sino que además éstos deben ser capaces de tener conectividad con una red privada y/o pública, con el fin de poder visualizar los datos en alguna plataforma en tiempo real.

Existen diversos tipos de sensores como se puede apreciar en la **Figura 7**. Cada uno cumple una función diferente, pero por sí solos no pueden considerarse sensores IoT; para que esto ocurra deben estar conectados a un microcontrolador o microprocesador y que éste a su vez tenga un sistema de comunicación que le permita conectarse con otros nodos formando una red de sensores.

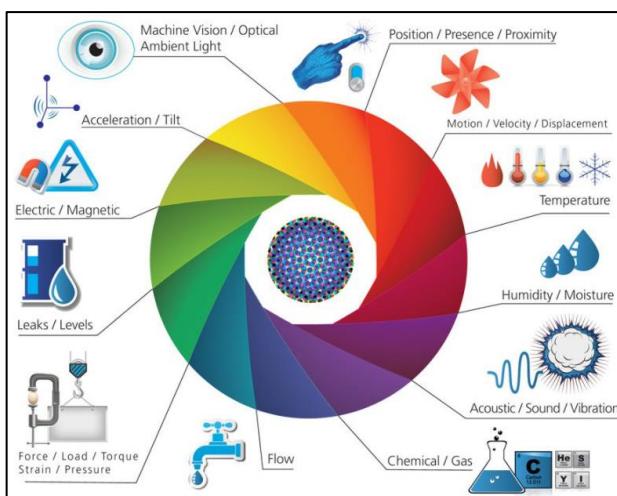


Figura 7. Tipos de Sensores

Fuente: (Postscapes, 2015)

Para hacer mediciones de energía eléctrica será necesario sensar tanto la corriente como el voltaje. Para el caso de la corriente existen dos tipos de sensores: invasivos y no invasivos. Los sensores invasivos son aquellos que deben conectarse en serie dentro del circuito a medir, normalmente hacen uso de resistencias pequeñas de alta precisión menores a  $1\Omega$  para calcular la caída de voltaje y de esta manera obtener el valor de la corriente. Los no invasivos en cambio aprovechan el campo magnético generado por la corriente y reducen su valor mediante pequeños transformadores. Los modelos más comerciales invasivo y no invasivo se muestran en la **Figura 8**.

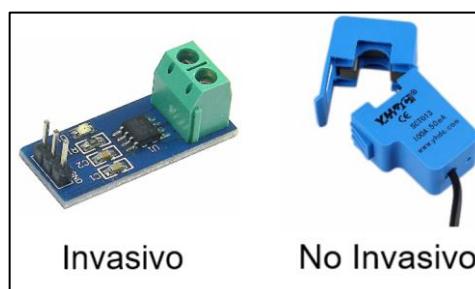


Figura 8. Sensores de corriente

Fuente: (Elaboración propia)

### 2.3.2. Actuadores IoT

En cuanto a los actuadores IoT también existen una extensa variedad, al igual que los sensores IoT, éstos deben tener conexión a una red ya sea a través de microcontroladores o mediante un módulo wifi incorporado. Los actuadores, al contrario de los sensores, reciben una señal eléctrica y lo transforman en una acción o activación de un proceso. En el ámbito IoT se usa comúnmente para apagar y encender dispositivos. Entre los actuadores más utilizados para este propósito se cuenta con el relé mecánico y contactores eléctricos, que permiten o impiden el paso de una corriente eléctrica hacia un dispositivo de potencia como es el caso de motores, luces, etc.

Hoy en día se dispone también de actuadores con conexiones a Internet, como los enchufes inteligentes, regletas inteligentes y hasta interruptores inteligentes. El comportamiento es algo diferente ya que no reciben una señal eléctrica como accionador, sino que reciben una señal digital a través del protocolo de Internet (*Internet Protocol - IP*), por lo que deben estar conectados a una red inalámbrica Wifi para poder controlarlos remotamente. Son muy útiles para controlar dispositivos comunes que no tienen una interfaz inalámbrica o cableada para conectarse a la red, brindándole más opciones como control por voz o celular, encendido y apagado por horarios, entre otras funciones. Sin embargo, como ya se mencionó anteriormente se deben conectar dispositivos que no dispongan de alguna circuitería electrónica que pueda averiarse por una supresión de energía abrupta y repetitiva.

Finalmente existen soluciones IoT parciales y completas que incluyen tanto sensores como actuadores IoT. Los más comunes son los relacionados con el consumo energético. En el mercado actual se puede encontrar una diversa variedad de estas soluciones considerados medidores inteligentes, como por ejemplo las marcas SMAPPEE<sup>3</sup>, SHELLY<sup>4</sup>, SONOFF<sup>5</sup>. Todas estas marcas enfocan su mercado al control y sensado inteligente de la energía eléctrica, es decir a la domótica e inmótica. Estas marcas a su vez tienen diversos productos que se acoplan a las necesidades de sus clientes y van desde sencillos medidores para una residencia de una planta, hasta medidores trifásicos para edificios e infraestructuras más exigentes a nivel eléctrico.

La marca SMAPPEE ofrece un producto denominado Smappee PRO, que ofrece un monitoreo en tiempo real de sistemas monofásicos o trifásicos, con la capacidad de acoplar y extender varios sensores de corriente lo que lo convierte en un medidor escalable. También posee una conexión ethernet para el envío de datos a la nube, en donde los usuarios pueden monitorizar sus consumos.

Al igual que SMAPPEE, la marca SHELLY tiene un producto equivalente llamado Shelly 3EM que realiza las mismas funciones, pero limitado a tres sensores de corriente. Dependiendo de la necesidad, este producto puede trabajar en modo trifásico, bifásico o monofásico de 3 canales. Una diferencia muy clara es la posibilidad de no utilizar la nube de SHELLY, sino enviar los datos a algún servidor o plataforma propia.

---

<sup>3</sup> **Sitio Web:** <https://www.smappee.com/es/>

<sup>4</sup> **Sitio Web:** <https://shelly.cloud/>

<sup>5</sup> **Sitio Web:** <https://sonoff.tech/>

Por otro lado, la marca SONOFF está más orientada a dispositivos IoT para el control domótico, aunque sí dispone de un medidor inteligente llamado SONOFF POWR3, pero es más limitado, pudiendo solo monitorizar y controlar un sistema monofásico.

Una comparativa final con más detalle se muestra en la **Tabla 7**.

Tabla 7. Comparación de sistemas de monitorización IoT

Sistema IoT	Corriente Max./canal	Soporta tres fases	Plataforma IoT	Actuadores	Tipo de sensores	Precio
Smappee Pro	200 A	Sí	Propia	Sí	No invasivo	\$640
Shelly 3EM	120 A	Sí	Propia o externa	Sí	No invasivo	\$118.80
SONOFF POWR3	25 A	No	Externa	Sí	Invasivo	\$34.99

**Fuente:** (Elaboración propia)

### 2.3.3. Protocolos de comunicación a través de IP (Internet Protocol)

El protocolo MQTT (*Message Queuing Telemetry Transport*) es considerado un protocolo para IoT, que fue introducido en el año de 1999. Su arquitectura es de cliente/bróker y se maneja a través de publicaciones y suscripciones (Naik, 2017). Trabaja sobre el protocolo IP, contiene dos bytes de encabezado y un tamaño de mensaje máximo de 256MB.

Esta comunicación se puede hacer más confiable utilizando seguridad TLS/SSL (*Transport Layer Security / Secure Sockets Layer*) y una calidad de servicio de tres niveles: “*Como máximo entregar una vez*”, “*Al menos entregar una vez*” y “*Entregar exactamente una vez*” (Hillar, 2017). Es ideal para dispositivos de baja potencia dado que no consume demasiada energía eléctrica.

Otro protocolo usado en IoT es el protocolo HTTP (*Hypertext Transfer Protocol*), el cual es muy utilizado en la navegación de páginas web y su estructura es cliente/servidor. En el inicio de la comunicación, el cliente siempre realizará una petición a un servidor y éste le dará una respuesta para dicha solicitud. Las peticiones por lo general son realizadas a través de funciones *GET* y *POST*, las cuales permiten obtener y enviar información al servidor respectivamente. Sin embargo, este protocolo consume bastante energía por lo que es recomendable utilizarlo en casos donde el sistema embebido tenga una fuente de energía estable.

Un fuerte competidor para MQTT es el protocolo CoAP (*Constrained Application Protocol*), que tiene menor consumo a pesar de enviar más información en las cabeceras de los mensajes. Sin embargo, no es un protocolo 100% estandarizado por lo que no hay mucho

soporte en la comunidad de Internet, su forma de trabajo es similar al protocolo *HTTP*, pero más simplificado y también utiliza los métodos *GET* y *POST* para la recepción y envío de información respectivamente.

En la **Figura 9** se muestra una comparación entre cuatro protocolos de comunicación orientados a IoT basado en su nivel de consumo.

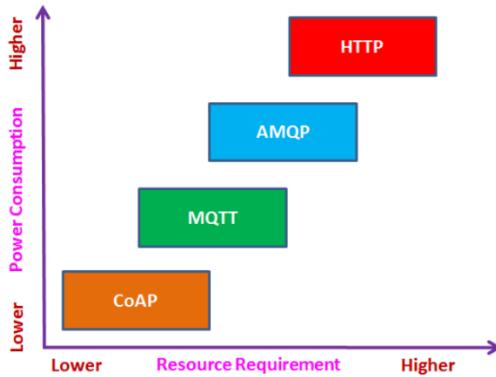


Figura 9. Consumo de energía vs Recursos requeridos

Fuente: (Naik, 2017)

El estudio realizado por (Bartnitsky, 2018) hace una comparativa a nivel energético y de eficiencia entre los dos protocolos más usados hoy en día: MQTT y HTTP. En este estudio se analizan diferentes escenarios al enviar mensajes a través de los dos protocolos mencionados. El primer escenario es la recepción de mensajes, evaluando el tiempo de retardo de cada uno de ellos. El resultado final es que MQTT es 25 veces más rápido que HTTP.

El siguiente escenario es comparar la cantidad de paquetes necesarios para establecer y cerrar una conexión. El resultado fue un 10% menos a través de MQTT, tanto en conexión insegura como en conexión segura (HTTPS y MQTT/SSL). Por último, se evalúa el consumo energético a través de una Raspberry implementando estos protocolos y una vez más se obtiene que el protocolo MQTT es más eficiente llegando a un 22% menos de consumo energético.

#### 2.3.4. Plataformas IoT

Las plataformas IoT permiten normalmente el monitoreo de parámetros de interés, en algunos casos permiten interactuar remotamente con actuadores. Existen diversas plataformas IoT en línea que ofrecen una versión académica o de prueba gratuita y otra versión de pago más profesional que viene con mayores funcionalidades; entre las más conocidas se cuenta con: ThingsBoard, Ubidots, ThingSpeak, KAA, entre otras.

ThingsBoard se diferencia del resto de plataformas porque es de código abierto y cumple las funciones principales como adquisición de datos, administración de equipos, procesamiento y visualización de información. Es compatible con todos los protocolos IoT mencionados en la

sección anterior, y soporta dos tipos de implementaciones: locales y en la nube (ThingsBoard, 2021).

A diferencia de otras plataformas, ThingsBoard ofrece una versión gratuita (*Community Edition*) sin límites de dispositivos o usuarios. Esta versión debe ser instalada en servidores locales o en servidores en la nube como Azure o AWS.

La versión profesional (*Professional Edition*) que sí tiene un costo mensual, difiere de la versión gratuita en el soporte a las tecnologías de baja potencia y largo alcance o Redes LPWAN por sus siglas en inglés *Low Power Wide Area Network*, es más completo en cuanto a parámetros de configuración, además permite la exportación de datos en CSV/XLS.

Otra plataforma IoT conocida es Ubidots, está desarrollada para trabajar sobre los protocolos MQTT o HTTP y dispone de varios planes de pago que dependerán de la cantidad de dispositivos conectados, la cantidad de datos a enviar, funciones para preprocesamiento de datos, generación de alarmas a través de correos o mensajes de celular, entre otras funciones. También dispone de una versión gratuita denominada STEM, pero es con fines académicos, por lo que será una plataforma muy limitada y difícil de administrar. Una desventaja relevante es que no se puede integrar la plataforma con servidores propios y es requisito utilizar los servidores de Ubidots.

ThingSpeak también es considerada una plataforma IoT, perteneciente a MathWorks, a pesar de que permite mostrar valores y gráficos en su plataforma, está más orientada al procesamiento de datos en conjunto con MATLAB, lo que encarece un poco la solución dado que MATLAB es un software de pago. Sin embargo, sigue cumpliendo con las funciones básicas de una plataforma IoT que sería el almacenamiento, visualización y obtención de datos a través de los protocolos MQTT y HTTP.

Kaa es considerada una plataforma IoT como servicio o PaaS por sus siglas en inglés *Platform-as-a-Service*. Al igual que las demás plataformas, también tiene una versión gratuita pero más limitada aún, porque solo permite la conexión de 5 dispositivos dentro de la nube de Kaa, se puede utilizar servidores propios para alojar la plataforma Kaa, incrementando la cantidad de dispositivos a 25, pero este plan ya tiene un costo mensual de \$99.

### **2.3.5. Aprendizaje Automático**

El Aprendizaje Automático (*Machine Learning - ML*) es la capacidad que tienen las máquinas o sistemas de aprender patrones basado en el comportamiento de datos. Los modelos de ML están muy relacionados con probabilidades y estadísticas.

Existen dos grandes categorías para el aprendizaje automático, el aprendizaje supervisado y no supervisado. El aprendizaje supervisado es aquel que requiere de un grupo de datos para

entrenar un algoritmo y determinar un valor o categoría con nuevos datos de entrada. Por otro lado, el aprendizaje no supervisado no tiene preestablecido los resultados posibles, es el mismo algoritmo el encargado de determinar los valores similares o categorías, por lo que suelen entregar datos menos exactos y un tanto subjetivos.

En ambos casos se tienen algunos modelos que pueden entregar datos cuantitativos o datos cualitativos. En la **Tabla 8** se puede observar una lista de los modelos más comunes de aprendizaje automático clasificado según su resultado final.

Tabla 8. Modelos de aprendizaje automático

	Valores cuantitativos	Valores cualitativos
Aprendizaje supervisado	<ul style="list-style-type: none"><li>• Regresión Lineal</li><li>• Regresión Polinómica</li><li>• Regresión por vectores de soporte (SVR)</li></ul>	<ul style="list-style-type: none"><li>• Árbol de decisión</li><li>• Máquina de vectores de soporte (SVM)</li><li>• K – vecinos más cercanos</li></ul>
Aprendizaje no supervisado	<ul style="list-style-type: none"><li>• Algoritmos determinísticos</li></ul>	<ul style="list-style-type: none"><li>• k – medias</li><li>• Agrupamiento jerárquico</li></ul>

**Fuente:** (Elaboración propia)

En (Hetal Bhavsar, 2012) se menciona que algunos de los métodos indicados anteriormente tienen características únicas que son útiles en ciertos escenarios en particular. Por ejemplo, el modelo de árbol de decisión permite clasificar elementos basado en ciertas decisiones, que fueron establecidas previamente con un conjunto de datos de entrenamiento, y siempre se buscará el árbol de decisión óptimo y eficiente. Los vectores de soporte utilizados tanto para regresión (*Support Vector Regression - SVR*) como clasificación (*Support Vector Machine - SVM*) se basan en un dato de entrenamiento para encontrar vectores que segmenten o clasifiquen el resto de los datos en el plano. Algo similar ocurre con el método de “k-vecinos más cercanos” en que se trata de agrupar todo un conjunto de datos que estén relativamente cercanos para poder categorizarlos.

Además de los algoritmos mencionados, existe otro tipo de algoritmos que son más usados en predicciones de consumo energético, las series de tiempo basado en datos anteriores. Esta categoría es ideal para los consumos que pueden ser graficados a lo largo del tiempo. En la *¡Error! No se encuentra el origen de la referencia.* se muestra una comparativa entre diferentes modelos junto a su error absoluto medio, que son relativamente pequeños por lo que es conveniente utilizar cualquiera de ellos. En (Deb, Zhang, Yang, Eang, & Wei, 2017) se hace un estudio de 9 técnicas predictivas para consumo energético utilizando series temporales, algunas técnicas son de aprendizaje automático y otras enfocadas al aprendizaje

profundo o *Deep Learning*, como es el caso de las redes neuronales. A continuación, se detallan algunas de estas técnicas:

#### 2.3.5.1. Red neuronal artificial (*Artificial Neural Network - ANN*)

Es uno de los modelos más relacionados con el aprendizaje profundo. Está basado en el funcionamiento neuronal de las personas, cada nodo o neurona contiene información exclusiva y al tener toda una red neuronal entrenada en base a un conjunto de valores podrá ser capaz de predecir y/o categorizar valores.

La efectividad de este modelo radica en el entrenamiento de datos, la cantidad de neuronas en las capas de entrada, salida y las capas intermedias u ocultas. Utilizar esta técnica de aprendizaje profundo permite que el modelo aprenda progresivamente sin la necesidad de una intervención humana.

#### 2.3.5.2. Modelo autorregresivo integrado de media móvil (*Autoregressive Integrated Moving Average - ARIMA*)

Este modelo transforma la serie de tiempo en estacionaria a través de diferenciación, utiliza la regresión lineal para encontrar patrones y predecir valores futuros. Esto implica que las predicciones no dependen de variables independientes, sino de sus propios valores pasados.

Es un modelo completo conformado por tres componentes: la componente autorregresiva (*Autoregressive-AR*), la componente integrada (*Integrated-I*) y la componente de media móvil (*Moving Average-MA*). Estos tres componentes corresponden a los parámetros de la ecuación general ARIMA ( $p, d, q$ ), donde  $p, d$  y  $q$  son valores positivos.

#### 2.3.5.3. Series de tiempo difusas (Fuzzy)

Este modelo se basa en valores cualitativos, primero se realiza una asignación numérica dentro de un intervalo específico y luego se realiza el análisis numérico correspondiente. La longitud de cada intervalo difuso está estrechamente relacionada con la precisión del pronóstico.

Tabla 9. Comparación cuantitativa entre modelos predictivos de series de tiempo.

Modelo	Tipo de dato	Tamaño de muestreo	Error absoluto medio	Tiempo consumido
ANN	Carga eléctrica por hora	1 año	1.69% – 1.81%	35 s
ARIMA	Demanda pico mensual	6 años	1.05% – 2.59%	N/A
Fuzzy	Carga eléctrica diaria	6 meses	1.23% – 1.63%	N/A

Fuente: (Deb, Zhang, Yang, Eang, & Wei, 2017)

## 2.4. Conclusiones sobre el estado del arte

En todos los trabajos mencionados hasta el momento, se muestra una combinación entre algunas tecnologías y algoritmos. El trabajo que se expone busca unir todas y cada una de las tecnologías mencionadas sin que su costo de implementación se eleve considerablemente.

Para ello se considerará el uso de plataformas IoT de gran capacidad en su versión gratuita y sensores económicos pero fiables. Lo principal es conseguir un ahorro económico a partir de un ahorro energético basado en monitorización, control y predicción. Si se desea tener un ahorro energético más relevante, será necesario implantar más sensores, pero el costo del trabajo se incrementaría demasiado y ya no representaría un ahorro económico viable.

A nivel de plataforma se realizará la monitorización con ThingsBoard, dado que permite utilizar gratuitamente servidores propios. Esto da la opción de implementarlo en Microsoft Azure junto con dos algoritmos predictivos: árboles de decisión y el modelo de medias móviles que es el algoritmo ARIMA con los parámetros  $p=0$  y  $d=0$ . El árbol de decisión se utiliza para categorizar los consumos y generar alertas oportunas, mientras que el modelo ARIMA hará las predicciones de consumo basadas en los datos anteriores. Finalmente, el hardware seleccionado es Shelly 3EM que proporciona las herramientas y funciones necesarias a un precio accesible y es compatible con uno de los protocolos más eficientes que es MQTT.

### 3. Descripción general de la contribución del TFM

#### 3.1. Objetivos

##### Objetivo general

El objetivo principal de este proyecto es desarrollar un sistema de monitorización, control y predicción de consumo energético de un edificio para el ahorro en la factura mensual mediante sistemas basados en la nube.

##### Objetivos específicos

- Compilar datos de energía consumida mediante instalación de sensores en el centro de carga del edificio con el fin de conocer el actual consumo de energía eléctrica y optimizar el uso de equipos eléctricos.
- Utilizar un algoritmo de aprendizaje automático apropiado para estimar el consumo energético de los siguientes meses y reducir costos.
- Determinar consumos eléctricos innecesarios a través de un algoritmo de clasificación para generar alertas al usuario final.
- Visualizar datos y predicciones obtenidos en tiempo real mediante una plataforma web pública para toma de decisiones preventivas a nivel energético.
- Crear un sistema de control de luminarias y aires acondicionados con un módulo inalámbrico integral operado desde una plataforma web pública con la finalidad de moderar el consumo innecesario.

#### 3.2. Metodología del trabajo

Existe una diversa variedad de metodologías para el desarrollo de un proyecto. Una de ellas es la metodología cascada o secuencial. Es una metodología tradicional que se basa en organizar el proyecto en fases secuenciales, en la que cada una de ellas depende de su fase anterior. Debido a que el presente trabajo involucra componentes de hardware y software, es necesario finalizar cada una de las fases antes de continuar con la siguiente, haciendo que este método sea ideal para el desarrollo del proyecto. Al tener segmentado el proyecto en fases o etapas, es más fácil tener una organización del proyecto, que permite trabajar completamente en una fase hasta corregir posibles errores y que no afecten a la siguiente. Al finalizar cada etapa se verifica que existan todos los requisitos necesarios para empezar la siguiente etapa.

La parte negativa de utilizar esta metodología es que el proyecto puede complicarse en caso de existir cambios durante el proceso que afecten el flujo de trabajo. Otra desventaja es el

hecho de que el usuario final no se involucra en el proceso del proyecto, sino hasta el final de esta, por lo tanto, se definen los objetivos y entregables de forma clara desde el inicio.

La **Figura 10** muestra las fases de forma secuencial.



**Figura 10.** Fases de la metodología cascada

Todo el proceso se realiza basado en las fases mencionadas anteriormente, las cuales son:

### **3.2.1. Planificación**

En esta fase se obtienen los requerimientos de la universidad con base en la problemática presentada, quiénes serán los usuarios finales y dónde se ejecutará la solución inicialmente. Se identifica el estado de soluciones (en caso de existir) por parte de la universidad frente a la problemática, luego se indica el alcance y limitaciones del proyecto. También se establecen tiempos aproximados para el desarrollo de la solución y un ahorro económico estimado para establecer la viabilidad del proyecto a mayor escala.

### **3.2.2. Diseño**

Se evalúa las soluciones ofrecidas actualmente por el mercado nacional e internacional para ofrecer una solución innovadora y de bajo costo, de esta forma se seleccionan las que mejor se adapten al proyecto y requerimientos de la universidad.

En esta fase se hace más evidente los componentes de hardware y software, aquí se define la estructura de cada componente y cómo se hará la integración entre ellas, incluyendo protocolos de transmisión, algoritmos de aprendizaje automático, etc.

### **3.2.3. Implementación**

Una vez definida la arquitectura general del proyecto, se realiza la selección de cada componente y se procede a realizar las configuraciones respectivas a nivel de hardware y software. A nivel de hardware, el envío de datos se hará hacia un computador, que cumplirá el rol de un servidor local encargado de levantar los servicios necesarios como la página web y el receptor de mensajes para los sensores IoT.

Todo el proceso de software se lo realiza con datos simulados y en un entorno local, es decir que la plataforma IoT solo podrá ser accedida desde una red interna.

### 3.2.4. Ejecución

Esta es la etapa final, para iniciar la ejecución el requisito previo es que en la fase anterior se hayan hecho todas las pruebas a nivel local. En esta etapa ya se usará un servidor en la nube que tenga almacenada la plataforma IoT y sea de acceso público. Los sensores de corriente se instalarán en un laboratorio y enviarán datos directamente a la nube.

## 3.3. Descripción general de las partes o componentes de la propuesta

La metodología seleccionada se enfoca en dos componentes, el primero es la implementación de hardware, el cual incluye la lectura de sensores de corriente para obtener el consumo desde un panel eléctrico centralizado, también se controla el circuito de luces y dispositivos controlados por señales infrarrojas como los aires acondicionados o proyectores.

El segundo componente es el desarrollo del software, que incluye un algoritmo de aprendizaje automático para detectar consumos no deseados generando alertas al usuario, además de un algoritmo predictivo, que muestra si el comportamiento del usuario está contribuyendo a un ahorro energético. Todo esto es accedido desde una plataforma IoT que además permite monitorizar el consumo eléctrico en tiempo real y controlar dispositivos remotamente.

Finalmente es necesario unir ambas partes, esta integración se hará mediante la red wifi de la universidad empleando algún protocolo IoT en los sensores de corriente para enviar los datos a la nube y a través de este mismo protocolo se podrá controlar el circuito eléctrico de las luces y el control de los aires acondicionados y proyectores.

### Alcance y limitaciones

El alcance del proyecto define la implantación de un sistema de actuadores y sensores IoT en un laboratorio de la universidad, que enviarán datos históricos de consumo eléctrico hacia un servidor para ser mostrados en una plataforma web, además de una proyección del consumo en el resto del mes para ahorrar costos, esta plataforma y el almacenamiento de datos se encontrarán dentro de servicios en la nube, concretamente en Microsoft Azure con una licencia académica.

El control de manera remota se hará a las dos cargas más significativas de la universidad: aires acondicionados e iluminación. A diferencia de la iluminación, el sistema de aires acondicionados no puede ser controlado directamente desde el panel eléctrico, dado que apagarlo de esta manera puede generar fallas a corto plazo en los equipos, en lugar de ello se instalará un módulo infrarrojo para el sistema de aires acondicionados del laboratorio.

Para la ejecución final sólo se enviarán datos reales de un laboratorio, para observar el funcionamiento del algoritmo predictivo y de la plataforma IoT a mayor escala, se generarán datos simulados de otros edificios.

### **Listado de participantes**

Como la metodología utilizada para el presente trabajo es Cascada o secuencial, los participantes sólo estarán involucrados al inicio y al final del proyecto, es decir en la planificación y aceptación de la ejecución. Los participantes son las autoridades principales de la universidad y usuarios finales, entre los cuales tenemos: rector, jefe del departamento de contabilidad, decanos y administradores de edificios.

### **Tecnologías implicadas**

Como se ha venido indicando anteriormente, el diseño del proyecto consta de una parte de hardware, que incluye control de dispositivos electrónicos y sensado de corriente eléctrica. Estos dos elementos están estrechamente relacionados con sensores y actuadores IoT, es decir, que poseen una conexión a Internet a través del protocolo IP, y a su vez pueden utilizar el protocolo HTTP o MQTT. En base a lo analizado en el capítulo anterior, el protocolo MQTT es el más eficiente para este proyecto.

La parte de software incluye tecnologías como plataformas IoT y aprendizaje automático. Uno de los requerimientos de la universidad es utilizar plataformas IoT de código abierto que puedan ser implementadas en servidores propios, por lo tanto, la plataforma más acorde a este requerimiento es ThingsBoard, que estará alojada en la nube de Microsoft Azure. Normalmente el uso de Azure tiene un costo de uso, pero la ESPOL ya dispone de licencias, es decir que utilizar Microsoft Azure no incurrirá gastos adicionales para la universidad, en esta misma máquina de Azure se establecerá el algoritmo predictivo de consumo energético a través del modelo de medias móviles utilizado para predicciones a través del tiempo que no disponen de variables independientes, pero sí valores en tiempos pasados. Así mismo, se hará uso de un middleware programado en Python para entrenar el modelo de árbol de decisión y luego generar alertas en caso de que categorice un consumo elevado.

### **Arquitectura, componentes e integración de tecnologías.**

En la **Figura 11** se muestra el diagrama completo del proyecto. La parte de hardware, ubicada en la esquina superior izquierda, se compone por dos sensores IoT de corriente ubicados en el panel eléctrico de un laboratorio de clases, estos sensores enviarán la señal eléctrica al dispositivo Shelly 3EM, que tiene una comunicación bidireccional con la máquina en Microsoft Azure. La comunicación es bidireccional porque el dispositivo Shelly tiene salidas que conectadas a un contactor pueden controlar el encendido y apagado de las luces directamente desde el panel eléctrico. Para el caso de los aires acondicionados se usa otro dispositivo IoT

con tecnología infrarroja que será el encargado de encenderlos y apagarlos, en este caso la comunicación es solo en un sentido. Tanto el dispositivo infrarrojo como el Shelly, se conectan hacia un bróker público MQTT instalado en la máquina de Microsoft Azure.

En la parte inferior se encuentra la arquitectura del software correspondiente a la máquina en Microsoft Azure, como se puede observar, está dividida en tres secciones: bróker MQTT, Thingsboard y un middleware.

El bróker MQTT es el encargado directo de comunicar el componente de hardware con el componente de software, está implementado en Mosquitto con el lenguaje de JavaScript. También es el encargado de procesar los datos entrantes y enviarlo a las otras dos secciones, envía los datos procesados a ThingsBoard para ver el consumo en tiempo real, de igual forma envía los datos al middleware para que se ejecuté los algoritmos de aprendizaje automático y posterior a ello sean visualizados en ThingsBoard.

Se implementa un middleware programado en Python para poder definir el algoritmo predictivo y categorizar consumos en base al árbol de decisión. Esta comunicación es en un solo sentido, recibe los datos procesados del Bróker en Mosquitto, luego se ejecutan los algoritmos y posteriormente se envían los resultados a ThingsBoard para ser presentados al usuario final.

La sección de ThingsBoard está conformada por un servidor web y otro bróker MQTT, diferente al detallado en los párrafos anteriores. La comunicación entre estos dos segmentos no es directa, es a través de una base de datos SQL (PostgreSQL) ya implementada en la misma plataforma IoT. A pesar de que ya existía un bróker MQTT de ThingsBoard, se tuvo que implementar otro en Mosquitto, esto debido a que los datos que envían los dispositivos IoT no son compatibles con la estructura de datos que debe recibir ThingsBoard. La comunicación entre los dos brókers es bidireccional porque uno envía los datos sensados y el otro envía los comandos de control remoto de parte de los usuarios finales conectados a través de ThingsBoard por el protocolo HTTPS.

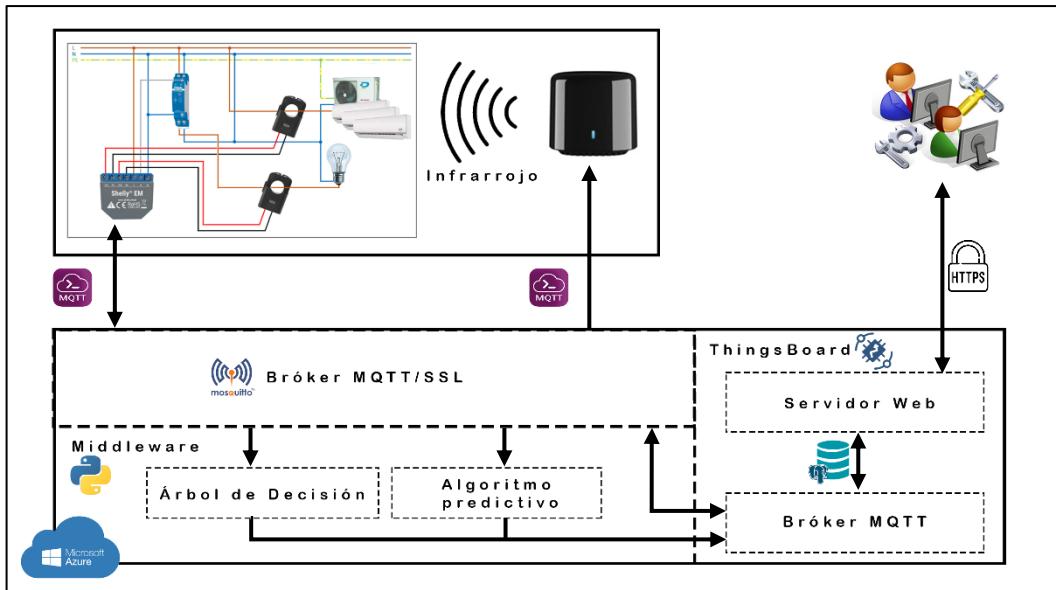


Figura 11. Arquitectura completa de la solución

Fuente: (Elaboración propia)

## Resultados esperados

Al utilizar la metodología en cascada, cada una de las fases debe tener un resultado final antes de continuar con la siguiente fase, estos resultados se detallan a continuación:

- **Planificación:** Al finalizar esta fase, se espera tener una organización bien definida del proyecto, incluyendo las actividades a realizar con sus respectivos tiempos de trabajo.
- **Diseño:** En esta fase se espera tener seleccionado todos los componentes y tecnologías que se van a utilizar, materiales que se deben adquirir con presupuestos y ganancias esperadas.
- **Implantación:** El resultado de esta fase deben ser pruebas exitosas de cada componente de forma aislada, validando los algoritmos y comunicaciones, pero todo a nivel local o con datos simulados.
- **Ejecución:** En la etapa final se espera tener listo un pequeño prototipo funcionando al 100% con datos reales de un laboratorio de clases en el que se pueda controlar las luces y el aire acondicionado.

Una vez listo los resultados técnicos, se espera obtener un resultado administrativo relacionado con la economía de la universidad. Se debe evidenciar que la factura mensual de energía eléctrica sea menor que la factura del mes anterior sumado a un porcentaje del dinero invertido en la implementación. Sin embargo, como la implementación final será en un solo laboratorio y no en toda la universidad, se hará una proyección económica basada en la diferencia de consumo energético medido en kilovatios-hora (kWh).

### Presupuesto y retorno esperado de la inversión

Uno de los requisitos principales que estableció la universidad es que el proyecto no debe generar egresos significativos a corto plazo, mientras que a largo plazo se debe percibir una disminución considerable del consumo energético, por lo menos una disminución del 10%.

Es por esto que todos los componentes de hardware deben ser de código abierto o que no generen gastos adicionales a la universidad, los precios mostrados en la Tabla 10 están basados en precios nacionales e internacionales de los mismos fabricantes.

Tabla 10. Costo de Hardware

Equipo	Cantidad	Precio
Shelly 3EM	1	\$118.80
Control infrarrojo genérico	1	\$14.99
Contactor 25A	1	\$15.00
<b>Total</b>		\$148.79 (126,89 €)

**Fuente:** (Elaboración propia)

Para los costos de infraestructura en la nube, se tiene la Tabla 11; sin embargo, estos costos **no serán considerados** debido a que la plataforma ThingsBoard *Community Edition* (gratuita) cumple con los requisitos necesarios para la implementación, sin necesidad de contratar la versión profesional. Además, la máquina en Microsoft Azure ya es costeada por una de las licencias que posee la universidad, por lo tanto, no genera egresos adicionales.

Tabla 11. Costo de Software

Servicio	Cantidad	Precio mensual	Precio anual
Standar_B2s – 2vcpu 4 GiB de memoria (Azure)	1	\$36.43	\$437.16
Disco duro SSD estándar 30GiB	1	\$4.81	\$57.72
Dirección IP pública dinámica		\$0.00	\$0.00
Datos de salida		\$0.00	\$0.00
ThingsBoard Profesional (Prototipo)	1	\$149.00	\$1 788.00
<b>Total</b>		\$190.24 (162,10 €)	\$2 282.88 (1 945.20 €)

**Fuente:** (Elaboración propia)

En cuanto los costos de desarrollo, dependerá de la universidad si desea contratar personal nuevo o utilizar a algún trabajador calificado para el desarrollo e implementación. En la Tabla 12 se muestra un aproximado de costos por desarrollo.

Tabla 12. Costo de desarrollo

Personal	Cantidad	Precio mensual	Precio semestral
Desarrollador	1	\$901.00	\$5 406.00
Instalador eléctrico	1	\$901.00	\$5 406.00
<b>Total</b>	\$1 802.00 (1536,77 €)		\$10 812.00 (9220,61 €)

**Fuente:** (Elaboración propia)

En la Tabla 13 se puede observar el presupuesto condensado para la implementación del sistema IoT en un laboratorio de clases.

Tabla 13. Costo final

Concepto	Precio mensual
Equipos	\$148.79
Servicios	\$0.00
Desarrollo	\$1 802.00
<b>Total</b>	\$1 950.79 (1663,66 €)

**Fuente:** (Elaboración propia)

Asumiendo que el personal se lo contrate por 6 meses, la inversión para un laboratorio sería de  $1(\$148.79) + 6(\$1 802) = \$10 960.79$ , si se logra el ahorro mínimo del 10% equivalente a \$40 aproximadamente por edificio se tendría un retorno de inversión en **22 años**; sin embargo, si se implementa la solución en los 85 edificios de ESPOL y el personal es contratado por dos años, la inversión total sería de  $85(\$148.79) + 24(\$1 802) = \$55 895.15$ , con un ahorro del 10% en toda la ESPOL equivalente a \$3 400, el retorno de inversión se reduce a **1 año y 4 meses**. Este tiempo de retorno, es un aproximado, debido a que depende bastante de la motivación del personal de ESPOL por el ahorro energético.

### Planificación general

La planificación general corresponde a la primera etapa de la metodología cascada, se enfoca principalmente en determinar los tiempos necesarios y actividades para llevar a cabo el proyecto en su totalidad, luego de haberse definido los requerimientos por parte de la universidad.

Las actividades más relevantes y críticas ocurren en las fases dos y tres, fase de diseño e implantación respectivamente.

Para la segunda fase (Diseño) se establece una actividad principal:

**- Actividad A**

Elaboración de la arquitectura general que incluye el análisis de tecnologías implicadas, investigación de soluciones similares, cotización de equipos y elaboración de presupuesto y retorno de la inversión.

Para la tercera fase (implantación), se definen cinco actividades:

**- Actividad B**

Se evalúa distintos sensores de corriente que permitan una medición más fiable y que su protocolo de comunicación se ajuste al proyecto, una vez encontrado el modelo adecuado se instalará dos sensores de corriente (50A y 120A) y uno de voltaje en el centro de carga (tablero eléctrico) para empezar con las mediciones.

**- Actividad C**

Para la adquisición de datos se conectarán estos sensores a un controlador, el cual procesa la información y obtiene los parámetros de interés, como la corriente, el voltaje y la potencia consumida, esta información será visualizada directamente desde el servidor web que genera el controlador por defecto, esta web por defecto es solo para la fase de implantación.

**- Actividad D**

Se selecciona el algoritmo de aprendizaje automático más apropiado para predecir consumos futuros, basado en el comportamiento de los datos sensados, para ello se requerirá de una base de datos de al menos un mes para dicho modelo; sin embargo, el sistema será capaz de ir aprendiendo al hacer una retroalimentación entre los valores sensados y los valores predichos.

**- Actividad E**

En esta etapa se establece la comunicación entre los sensores y la plataforma IoT a nivel local para visualizar los datos obtenidos en tiempo real, así como también mostrar las predicciones del algoritmo y otro apartado para realizar el control remoto de ciertos dispositivos eléctricos que trabajen con tecnología infrarroja.

**- Actividad F**

Desarrollo del hardware para control de dispositivos infrarrojos como los aires acondicionados, esto se realiza mediante un controlador genérico con conexión a internet para poder comunicarse con la plataforma IoT, también se controlará el flujo de corriente de un circuito dentro del centro de carga a través del controlador seleccionado en la actividad B.

En la **Tabla 14**, se establece el tiempo para cada actividad y la relación directa entre ellas. Sólo las actividades iniciales de cada fase son independientes del resto, para el caso de la actividad B se define un tiempo de 6 meses para poder tener información suficiente y poder entrenar de mejor forma el algoritmo predictivo.

Tabla 14. Tiempo y dependencias de actividades generales

Actividad	Tiempo total estimado	Dependencia
A Diseño de la arquitectura general	1 semana	-
B Selección de Hardware/Software	3 semanas	-
C Adquisición de datos	6 meses	B
D Selección de algoritmo ML	1 mes	C
E Desarrollo de plataforma IoT	2 meses	-
F Desarrollo de actuadores	2 meses	B,E

**Fuente:** (Elaboración Propia)

Por último a cada actividad general, se le asigna un grupo tareas específicas que deben ser realizadas dentro del tiempo determinado. Al igual que en el caso de las seis actividades mencionadas anteriormente, en la **Tabla 15** se muestra una estimación en días de lo que deben durar cada tarea y si dependen o no de la culminación de una tarea previa.

Tabla 15. Tiempo y dependencias de tareas específicas

Actividad	Tarea	Estimación (días)	Dependencia
1 A	Investigación de soluciones actuales y cotizaciones	2	-
2 A	Elaboración de presupuestos y arquitectura general	3	-
3 B	Investigar sensores/actuadores comercialmente locales y hacer pruebas de fiabilidad	3	-
4 B	Investigar sensores/actuadores comercialmente internacionales y estudiar casos de usos correspondientes	2	-
5 B	Investigar soluciones IoT a nivel local e internacional	5	-

6	B	Pruebas de fiabilidad del sistema IoT seleccionado	5	5
7	C	Instalación del sistema de monitorización IoT	1	6
8	C	Selección de una base de datos temporal	2	-
9	C	Almacenamiento de datos	180	7, 8
10	D	Investigación de algoritmos enfocados en predicciones de consumo	15	-
11	D	Selección de algoritmo más eficiente	10	9
12	E	Desarrollo en servidores locales MQTT	10	-
13	E	Desarrollo en servidores locales para base de datos	15	-
14	E	Desarrollo en servidores locales para plataforma IoT	15	-
15	E	Integración en servidores locales	15	12, 13, 14
16	E	Integración en servidores en la nube	10	15
17	F	Pruebas aisladas de actuadores	15	3, 4
18	F	Integración con plataforma IoT	15	16,17
19	F	Optimización de hardware	15	17

**Fuente:** (Elaboración Propia)

## 4. Desarrollo específico de la contribución

En este capítulo se detallan los pasos a nivel de software y hardware para la implementación del sistema IoT capaz de monitorear, controlar y predecir el consumo energético a través de aplicaciones en la nube para una infraestructura de un campus universitario. Debido a la extensa variedad de campus existentes en el mundo, es un poco complicado ofrecer una solución IoT que se adapte a todos los casos posibles, para efectos prácticos se analizará la situación actual de la Escuela Superior Politécnica del Litoral (ESPOL) ubicada en la ciudad de Guayaquil-Ecuador.

### 4.1. Planificación

El primer paso es obtener los requerimientos funcionales y no funcionales por parte de la universidad. Se definen como requerimientos no funcionales aquellos que establecen el comportamiento y capacidad del sistema que se va a desarrollar. En la **Tabla 16** se muestra los requerimientos no funcionales más relevantes para el proyecto.

Tabla 16. Requerimientos no funcionales

Requerimiento no funcional		Descripción
1	Rendimiento	Ser eficiente al procesar las peticiones de los usuarios
2	Disponibilidad	Ser accesible a cualquier hora y desde cualquier lugar
3	Robustez	Ser capaz de tolerar situaciones críticas de consumo
4	Escalabilidad	Poder expandir su capacidad a nivel de software o hardware
5	Compatibilidad	No depender de una sola tecnología específica

**Fuente:** (Elaboración Propia)

Los requerimientos funcionales son todos aquellos que la persona o institución cliente ofrece para solucionar su problemática. En este caso corresponde a la ESPOL, que especificó que la solución al problema de consumo energético debía ser mediante la implementación de un sistema de monitorización, control y predicción. El sistema debe mostrar como mínimo el consumo de un laboratorio, pero al mismo tiempo poder visualizar el consumo de edificios, facultades y toda la universidad en su totalidad. Además, se debe controlar el sistema de luces y aires acondicionados que son los factores con mayor consumo energético y por último ver una predicción de consumo para el siguiente mes a nivel de toda la ESPOL.

En la **Tabla 17** se muestra los requerimientos funcionales junto a la tecnología relacionada.

Tabla 17. Resumen de requerimientos funcionales

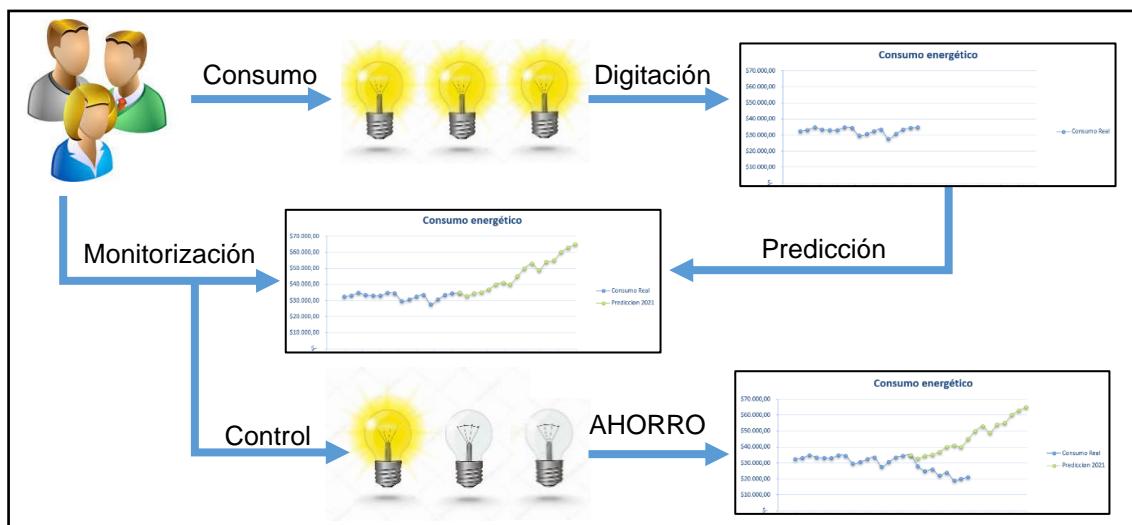
Requerimiento	Tecnologías
1 Consumo mínimo de un laboratorio	Componente de Hardware (sensores IoT)
2 Agregación de consumos por edificios, facultades y totalidad	Componente de Software (plataforma IoT)
3 Control de cargas eléctricas más significativas	Componente de hardware y software (plataforma IoT con conexión a actuadores IoT)
4 Predicción de consumo del siguiente mes	Componente de Software (algoritmo predictivo)

**Fuente:** (Elaboración Propia)

Una vez especificado todos los requerimientos, se procede con la elaboración de actividades generales con su respectivo tiempo de elaboración y determinar las dependencias entre ellas. Esta planificación se detalla en el capítulo anterior.

## 4.2. Diseño

En esta fase se determina la arquitectura del proyecto con base en los requerimientos especificados anteriormente. Para ello, primero se elabora un esquema general del comportamiento del usuario mostrado en la **Figura 12**. Se observa que inicialmente el usuario tendrá su consumo normal, esta información será digitalizada en una plataforma, para luego generar datos predictivos y que el usuario pueda monitorizarlos, haciendo que finalmente consuma menos y se refleje en un ahorro energético.



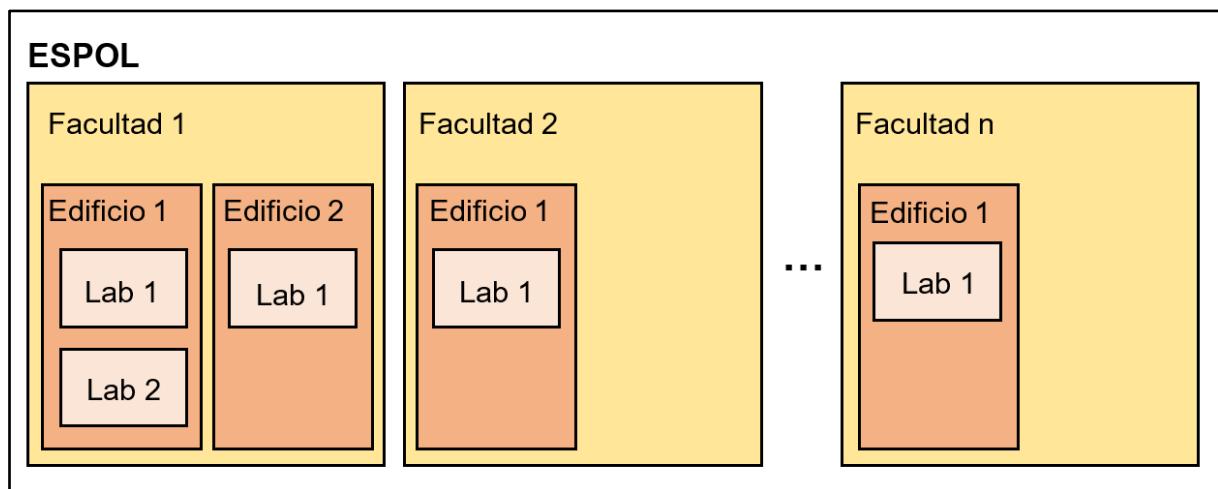
**Figura 12.** Comportamiento del usuario frente a la solución

**Fuente:** (Elaboración propia)

Luego de definirse el diagrama de interacción, se analiza los componentes necesarios que debe tener la solución, la arquitectura general se muestra en la **Figura 11**. Cada componente es necesario para acoplarse al diagrama de interacción.

Para poder digitar los datos de consumo eléctrico es necesario un componente de hardware que incluya sensores de corriente con capacidad máxima de 100 A, estos sensores deben tener la capacidad de conectarse vía ethernet o wifi hacia Internet para procesar los datos y convertirlos en información útil. Para la parte de predicción se hace uso de un componente de software, en este caso puede ser en algún servidor en la nube o un servidor local de la universidad, con base en los requerimientos no funcionales de rendimiento y disponibilidad, se escoge un servidor en la nube. Este servidor contendrá los algoritmos predictivos, la capacidad de mostrar la información en una plataforma IoT y controlar componentes electrónicos de forma remota a través de un dispositivo IoT con acceso a Internet.

Existen dos tipos de entidades en la plataforma IoT: infraestructura y dispositivos. Las entidades de tipo infraestructura representan los laboratorios, edificios y facultades de la universidad, cada una de estas entidades están distribuidas con base en la arquitectura mostrada en la **Figura 13**. Cada infraestructura tiene cuatro atributos: el primer atributo es el “*id*” de tipo numérico, es un valor único dentro de cada infraestructura contenedora. El segundo atributo es “*imageUrl*” que corresponde al enlace web donde se almacena el diseño de la entidad, ya sea un mapa o un plano. El tercer atributo es “*lastValues*” que almacena un arreglo de las últimas mediciones de cada infraestructura contenida. Finalmente están los atributos de coordenadas que van a especificar la ubicación de esa entidad dentro del diseño de la entidad contenedora.



**Figura 13.** Arquitectura de infraestructura en plataforma IoT

**Fuente:** (Elaboración propia)

Las entidades de tipo dispositivos representan a los sensores y actuadores IoT que van a monitorizar y controlar el consumo energético. Estas entidades siempre estarán contenidas dentro de una infraestructura de tipo laboratorio. Además, tienen asignadas dos atributos, las coordenadas de ubicación dentro de la entidad laboratorio y otro atributo “*on\_off*” para controlar los actuadores IoT.

## 4.3. Implantación

### 4.3.1. Evaluación y selección del sensor IoT

Se dispone de 3 opciones SMAPPEE, SHELLY 3EM y SONOFF POWR3 descritas en capítulos anteriores.

Como se indicó anteriormente, SMAPPEE es una solución completa y modular que abarca las necesidades de la universidad, posee actuadores y sensores de distintas capacidades, así mismo dispone de una plataforma web que permite la monitorización remota de todos los equipos conectados, genera históricos de consumo y permite exportarlos en diferentes formatos, pero esta solución no solo abarca las necesidades de la universidad sino que además ofrece otras funcionalidades como control y medición de nivel de gas, consumo de agua, paneles solares. Todas estas funcionalidades extras generan un costo adicional innecesario, además de que el sistema es cerrado, es decir que el usuario no puede acoplar tecnologías ajenas a la empresa fácilmente y no se cumpliría el requerimiento de compatibilidad.

En el caso de SHELLY 3EM dispone de más limitantes que SMAPPEE, solo puede sensar tres líneas de corriente y tres niveles de voltaje, en cuanto a sus actuadores sólo dispone de una salida que debe acoplarse un contactor eléctrico. La plataforma web es algo sencilla, está limitada a configuraciones del dispositivo, mediciones en tiempo real y mediciones acumuladas, como ventaja tiene que su costo es hasta 5 veces menor que el SMAPPEE, además de ser de código abierto, dándole la oportunidad al usuario de editar su programación para mejorar y adaptar funcionalidades, esto es muy útil si se desea integrar una plataforma web propia con los requerimientos específicos de la universidad, por ejemplo la integración de algún algoritmo predictivo.

Finalmente, SONOFF POWR3 es un dispositivo que en cuanto a funcionalidades es similar a SHELLY 3EM, con la diferencia de que es más orientado a domótica por lo que la corriente máxima que soporta es de 25 amperios, limitando bastante la aplicación a edificios completos de facultades dentro de la universidad.

Con base en este análisis, se escoge el sensor Shelly 3EM por su precio y funcionalidades suficientes para la ejecución del proyecto.

#### 4.3.2. Adquisición de datos

Una vez seleccionado el controlador que se va a usar, se procede con la adquisición de datos para su almacenamiento temporal, puede ser directamente a un bróker MQTT local o la plataforma web por defecto que genera el controlador.

Inicialmente se decide utilizar la plataforma web por defecto del sensor Shelly, pero esto generaba un paso adicional para la digitalización de los datos, entonces se cambió la configuración para enviar los datos a un bróker local y que los vaya almacenando internamente. La estructura de los datos puede ser manipulable, el código implementado se encuentra en el Anexo A y el resultado en la **Figura 14**.

```
C:\Users\PC Principal\Desktop\TESIS\database>node ./src/database
connected true
Conectado a ThingsBoard
Enviando a ThingsBoard: {"total0":12954.5,"total1":0,"power0":61.82,"power1":0}
  { relay: false }
  { relay: true }
  { relay: false }
  { relay: true }
Enviando a ThingsBoard: {"total0":12954.5,"total1":0,"power0":61.96,"power1":0}
Enviando a ThingsBoard: {"total0":12954.5,"total1":0,"power0":62.31,"power1":0}
Enviando a ThingsBoard: {"total0":12954.5,"total1":0,"power0":61.98,"power1":0}
Enviando a ThingsBoard: {"total0":12954.5,"total1":0,"power0":62.21,"power1":0}
```

**Figura 14.** Almacenamiento de datos en servidor local

**Fuente:** (Elaboración propia)

#### 4.3.3. Selección de aprendizaje automático

Para esta actividad se definen dos modelos para entrenar y utilizar. El primer modelo será el encargado de predecir los consumos eléctricos a lo largo del mes a nivel general de toda la ESPOL. El segundo modelo será el encargado de definir si un consumo es innecesario o no a nivel de laboratorio o aula, en caso de ser un consumo innecesario se mandará una alerta a través de una plataforma IoT. Por lo tanto, se necesita un algoritmo predictivo y otro algoritmo clasificadorio.

Para el algoritmo predictivo, se tiene en consideración que no existe información necesaria para utilizar ciertos modelos, por ejemplo, el modelo de Regresión por vectores de soporte (SVR) necesita de algunas variables independientes que afecten directamente el uso de la energía eléctrica, pero sólo se tiene el consumo histórico a lo largo del tiempo. En el caso de regresión lineal tampoco se adapta al proyecto directamente, porque el consumo eléctrico no tiene un comportamiento lineal. Sin embargo, si se emplea el modelo ARIMA, se puede transformar el consumo eléctrico en una señal estacionaria y aplicar regresión lineal.

Para la elaboración de este algoritmo se usaron los datos de las planillas de consumo de toda la ESPOL desde enero del año 2017 hasta agosto del 2021. En la **Figura 15** se muestra la tendencia del consumo a lo largo del tiempo.



**Figura 15.** Consumo energético de ESPOL (2017-2021)

**Fuente:** (Elaboración propia)

Como se observa en la imagen anterior, hasta el año 2019 el consumo tiene una tendencia periódica anual, pero luego de los eventos ocurridos por la pandemia se aprecia una disminución considerable y una nueva tendencia de consumo, debido a esto se realizaron dos entrenamientos para este modelo. El modelo se aplica de igual forma en ambos casos, pero los datos de entrenamiento son segmentados en dos grupos, el primero es desde enero/2017 hasta diciembre/2019 y el segundo sólo abarca el año 2020. No se considera el año 2021 porque el modelo necesita periodos exactamente iguales, en este caso periodos de 12 meses.

Al analizar la gráfica del primer grupo de entrenamiento, se observa una tendencia periódica: de enero a abril el consumo disminuye debido al periodo vacacional, luego se incrementa en el mes de mayo por el inicio de clases, después empieza a disminuir progresivamente hasta el mes de septiembre que son vacaciones parciales y se empieza a incrementar hasta el mes de diciembre por las fiestas. Los datos de consumo para el entrenamiento del primer grupo se encuentran en el **Anexo B**.

Para poder hacer la predicción es necesario aislar estos datos estacionarios mediante el método de medias móviles equivalente al método de ARIMA con parámetros  $p=0$  y  $d=0$ . El primer paso es obtener el factor estacional para cada mes, esto se hace obteniendo el promedio de consumo de cada mes a lo largo de los años, y dividiéndolo para el promedio de consumo total. El factor estacional para el mes de enero durante los años de 2017 a 2019 expresado matemáticamente es:

$$F_{enero} = \frac{\frac{1}{n} \sum_{i=0}^n \text{Consumos de enero}}{\frac{1}{m} \sum_{i=0}^m \text{Consumos}}; \text{ donde } n = 3 \text{ y } m = 36$$

La **Tabla 18** muestra los factores estacionales de cada mes luego de haberse aplicado la fórmula anterior. Estos datos representan el peso proporcional de cada mes respecto del consumo total a lo largo de 3 años.

Tabla 18. Factores estacionales grupo 1

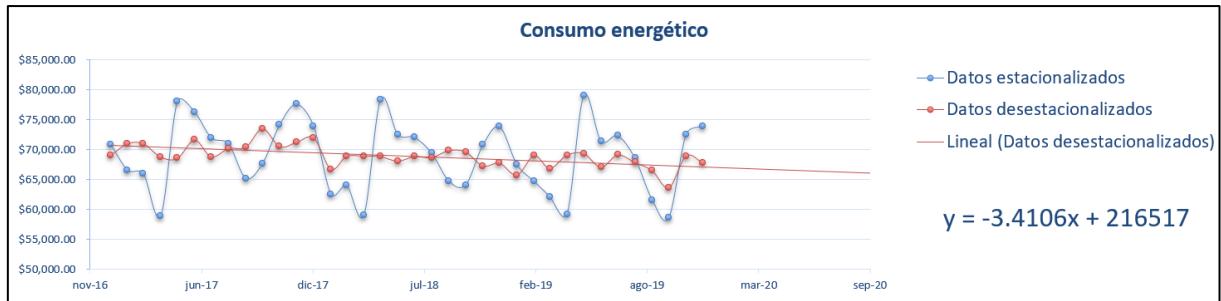
Mes	Factor Estacional
Enero	1.0270
Febrero	0.9370
Marzo	0.9291
Abril	0.8565
Mayo	1.1389
Junio	1.0655
Julio	1.0465
Agosto	1.0106
Septiembre	0.9263
Octubre	0.9206
Noviembre	1.0521
Diciembre	1.0900

**Fuente:** (Elaboración Propia)

Con estos factores estacionales se puede desestacionalizar la serie de tiempo y ver el comportamiento lineal para hacer una predicción mediante regresión lineal. El proceso para desestacionalizarlos es dividir cada valor real para el factor de estacionalidad correspondiente. En la **Figura 16** se muestra la gráfica con los datos estacionalizados y desestacionalizados. Se observa una pendiente negativa que indica que ESPOL cada año ha ido consumiendo menos energía progresivamente; sin embargo, la pendiente es poco significativa comparada con los consumos elevados, haciendo un cálculo para predecir el tiempo en que ESPOL ahorraría 10% del consumo del año 2017, se obtiene un resultado de 5 años y 7 meses aproximadamente.

$$x_2 - x_1 = \frac{0.10y_1}{3.4106};$$

Si  $y_1 = 70\ 000$ , entonces  $x_1 - x_2 = 2052$  días o 5.62 años

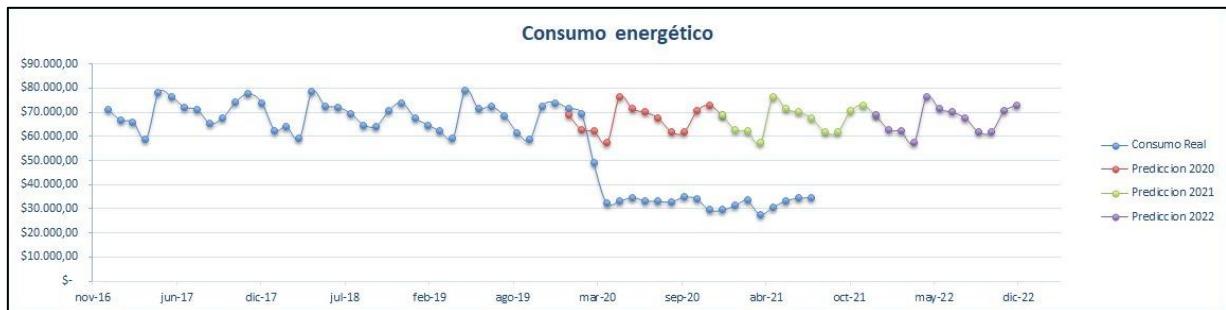


**Figura 16.** Datos desestacionalizados, tendencia lineal

**Fuente:** (Elaboración propia)

Luego de obtener la ecuación lineal, se procede a calcular los datos desestacionalizados de todos los meses de los años 2020, 2021 y 2022 mediante proyecciones, finalmente cada una de estas proyecciones se las multiplica por su factor estacional correspondiente y se obtienen los valores futuros.

En la **Figura 17** se muestra el resultado predictivo de un entrenamiento basado en datos desde enero de 2017 hasta diciembre de 2019. Se puede observar cuál hubiera sido el comportamiento si no hubiera ocurrido la pandemia mundial.



**Figura 17.** Algoritmo predictivo sin considerar datos de pandemia

**Fuente:** (Elaboración propia)

Para el segundo grupo de entrenamiento se realiza el mismo proceso anterior; sin embargo, en el año 2020 se observa que durante los meses de enero, febrero y marzo los consumos son valores transitorios que no siguen una tendencia ni para el primer grupo ni el segundo, por lo que se obvian dichos valores y son reemplazados por valores similares a los meses correspondientes del 2021. Los datos completos se observan en el **Anexo C**.

Para este segundo grupo, los factores estacionales se muestran en la **Tabla 19**. La predicción final se observa en la **Figura 18**.

Tabla 19. Factores estacionales grupo 2

Mes	Factor Estacional
<b>Enero</b>	0.9253
<b>Febrero</b>	0.9830
<b>Marzo</b>	1.0303
<b>Abril</b>	0.9234
<b>Mayo</b>	0.9814
<b>Junio</b>	1.0473
<b>Julio</b>	1.0425
<b>Agosto</b>	1.0438
<b>Septiembre</b>	1.0123
<b>Octubre</b>	1.0707
<b>Noviembre</b>	1.0539
<b>Diciembre</b>	0.9093

Fuente: (Elaboración Propia)

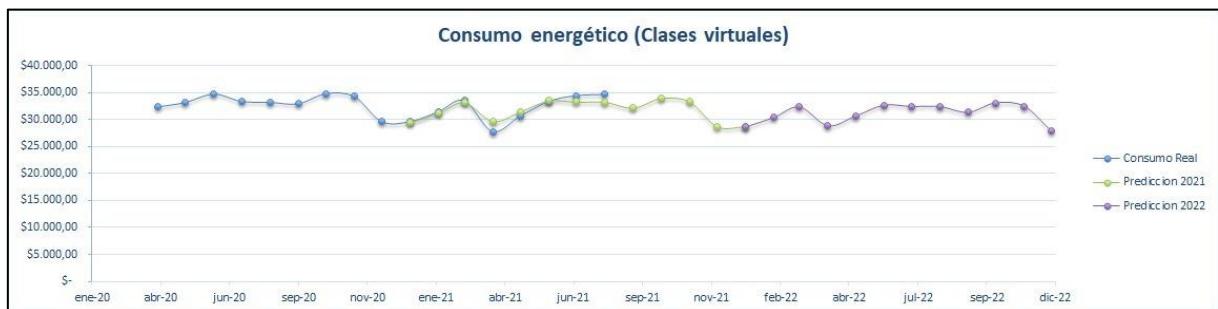


Figura 18. Algoritmo predictivo considerando datos durante pandemia

Fuente: (Elaboración propia)

El siguiente modelo que se va a entrenar es de tipo clasificadorio. En este caso sí se dispone de más variables independientes para determinar si un consumo es desperdicio o no a nivel de laboratorios o aulas. Las variables independientes iniciales sólo son fecha, hora y consumo, pero basados en la fecha y hora se puede obtener otras variables de interés detalladas en la Tabla 20.

Con base en estas variables, el modelo debe clasificar dos estados: Consumo normal o consumo innecesario, es decir que es un algoritmo de clasificación binaria. En un escenario ideal se espera que el consumo de un laboratorio sea menor a 1kW cada hora en los fines de semana, feriados y fuera de horario laboral. Sin embargo, dependiendo de la hora del día existen casos en que los consumos son menores a 1kw porque no hay clases en ese periodo. Por lo tanto, el modelo también debe clasificar si hay un consumo innecesario en estas horas.

Tabla 20. Variable independientes para algoritmo clasificatorio

Variable	Tipo de dato	Descripción
<b>Hora del día</b>	entero	Valor que indica la hora actual (0 - 23)
<b>Consumo</b>	decimal	Cantidad de vatios consumidos
<b>Fin de semana</b>	binario	1 si es sábado o domingo
<b>Feriado/vacaciones</b>	binario	1 si la fecha coincide con feriado nacional
<b>Horario Laboral</b>	binario	1 si se está en horario laboral (7:00 – 19:00)

**Fuente:** (Elaboración Propia)

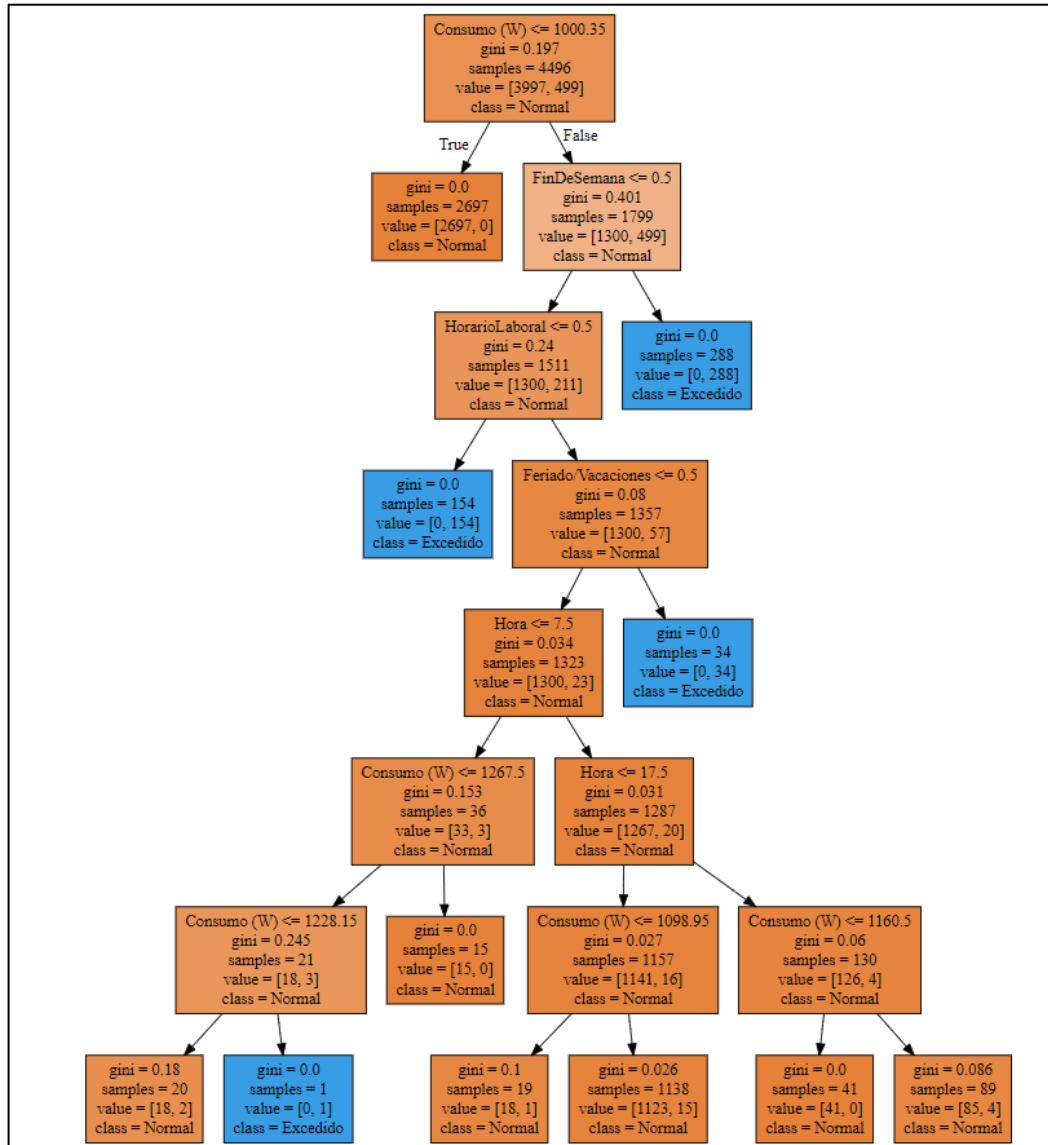
Como se mencionó anteriormente, existen diversos modelos para clasificación, algunos tienen mejor rendimiento cuando los parámetros de entrada son datos continuos, otros cuando son varios datos discretos, pero los parámetros de entrada a utilizar en este proyecto son en su mayoría de tipo binario, y el comportamiento dependerá de que ciertos eventos hayan ocurrido o no, es decir que el modelo que mejor se adapta a la naturaleza de estos datos es el modelo de árbol de decisión.

Para el entrenamiento de este modelo, fue necesario utilizar una gran cantidad de datos, estos datos se los obtuvo del sensor SMAPPEE que tiene almacenado datos desde el 2020. Se hizo una preparación de datos, con la fecha del consumo se generaron las variables de feriado, fin de semana y horario laboral, un ejemplo de la muestra está en el **Anexo D**. Luego se definieron las dos categorías de salida (normal y excedido).

Además de la preparación de los datos, se empleó la librería Scikit-learn, que incluye funciones para aprendizaje automático en Python. Se escogió esta librería por ser una de las más utilizadas hoy en día, aunque se pudo seleccionar cualquier otra librería como Theano, Keras, TensorFlow, entre otros. No se realiza un estudio sobre cada librería porque la finalidad es emplear un algoritmo predictivo independientemente de la librería, interesa más la eficiencia y compatibilidad del modelo, que la velocidad de entrenamiento por alguna librería.

El código en Python utilizado para el entrenamiento se muestra en el

Anexo E. Se usaron 5621 datos, de los cuales se seleccionó el 80% para entrenamiento y el 20% para evaluar la precisión del modelo. Inicialmente se obtuvo una precisión de 0.99, pero el modelo no era óptimo debido a que generaba demasiadas ramas como se observa en el **Anexo F**. Debido a eso, se le establece al modelo un límite de profundidad de 7, generando un árbol más pequeño y con una precisión de 0.975 mostrado en la **Figura 19**, indicando que es un algoritmo con bastante probabilidad de acierto en la clasificación.



**Figura 19.** Árbol de decisión con máxima profundidad de 7

**Fuente:** (Elaboración propia)

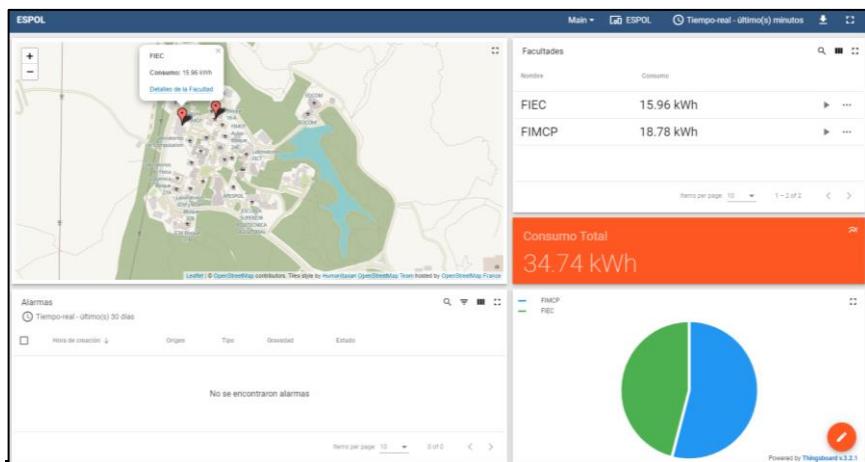
#### 4.3.4. Desarrollo de plataforma IoT

Para esta actividad se desarrolla la plataforma en un servidor local para realizar pruebas de desarrollo antes de ejecutar todo el sistema completo.

De todas las plataformas IoT disponibles, se seleccionó ThingsBoard porque su versión gratuita no genera limitantes respecto de la creación de usuarios, dispositivos o envío de datos. Pero sobre todo se escogió esta plataforma porque puede ser instalada en servidores propios, sin depender directamente de los servidores de ellos.

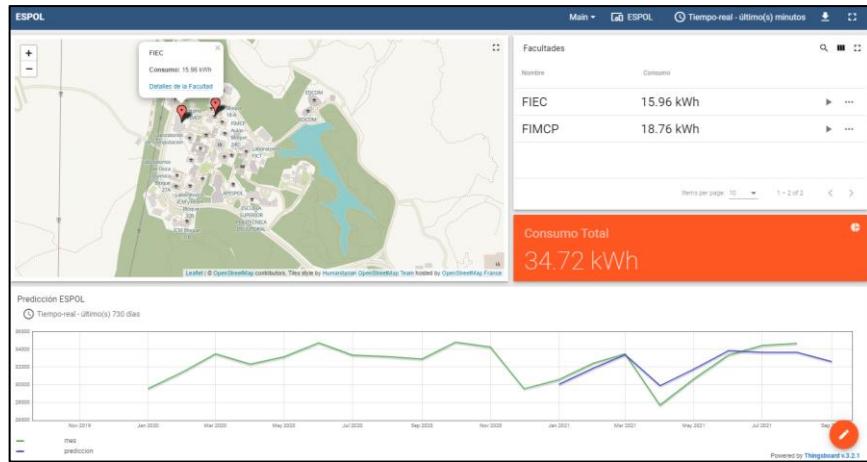
Los primeros pasos son descargar e instalar ThingsBoard y PostgreSQL en un servidor local. ThingsBoard será el encargado de administrar el servidor web, mientras que PostgreSQL almacenará toda la información necesaria.

En la **Figura 20** se muestra el diseño de la plataforma con su vista principal, contiene un mapa general de toda la ESPOL donde se localicen todas las facultades, para tema de pruebas solo se muestran dos facultades, la Facultad de Ingeniería en Electricidad y Computación (FIEC) y la Facultad de Ingeniería en Mecánica y Ciencias de la Producción (FIMCP). Luego se debe mostrar la lista de facultades con su respectivo consumo actual, también el consumo general de toda la ESPOL, que sería el equivalente a la suma de los consumos de todas las facultades. Este elemento de consumo total cuenta con un botón para mostrar los valores predictivos del siguiente mes como se observa en la **Figura 21**. En la parte inferior de la ventana principal se muestra una comparativa en diagrama pastel de los consumos de todas las facultades, esto sirve para prestar mayor atención a la facultad que tenga más consumo referente a las demás. Finalmente, se establece una sección de alarmas, dado que estamos en la vista principal, en esta sección aparecerán todas las alertas de consumo “Excedido” a nivel de ESPOL generado por el algoritmo clasificatorio, al dar doble clic sobre una alarma, inmediatamente la vista cambia a nivel de la facultad generadora de dicha alarma, si se sigue dando doble clic, se mostrará la ventana a nivel de edificio y luego a nivel de laboratorio. Estas ventanas se encuentran en el **Anexo G**.



**Figura 20.** Página inicial de ThingsBoard

**Fuente:** (Elaboración propia)



**Figura 21.** Página inicial de ThingsBoard con datos predictivos

**Fuente:** (Elaboración propia)

En la vista de Laboratorio se presenta la ubicación de los posibles actuadores de esa zona.

En la **Figura 22** se observa la ubicación de 3 actuadores, el Shelly capaz de controlar el sistema de luces y dos dispositivos infrarrojos: un proyector y el aire acondicionado. También se observa la potencia acumulada en KWh de ese laboratorio, como la instalación eléctrica es bifásica, se muestran solo dos gráficas en tiempo real en vatios. En el segmento derecho se diseñan dos botones para activar y desactivar con actuadores. Finalmente, se muestra la lista de alarmas generadas por ese laboratorio en específico, cada alarma tiene dos botones, el primero es para indicar que se ha detectado la alarma y se está tomando las medidas pertinentes, el segundo es para limpiar la alarma, es decir que ya se realizó el respectivo ahorro energético apagando alguno de los actuadores. Mientras no se quite la alarma, la plataforma no llenará de más alarmas generadas por el algoritmo clasificadorio, es decir que serán ignoradas hasta que la primera alarma sea atendida y se corrija.

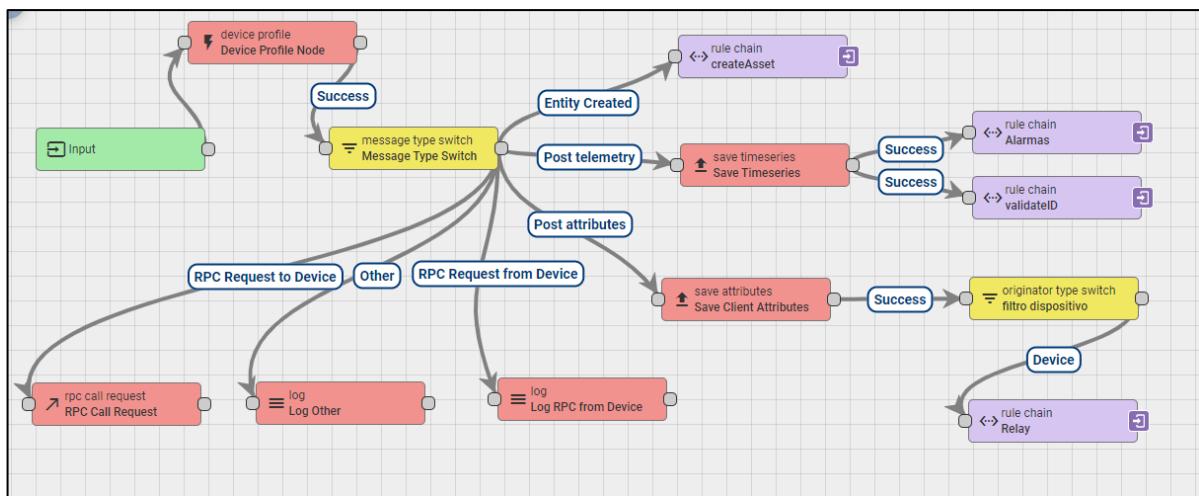


**Figura 22.** Ventana a nivel de Laboratorio

**Fuente:** (Elaboración propia)

Luego de terminar el diseño de la plataforma IoT, lo siguiente fue realizar la parte lógica de la plataforma, en ThingsBoard se lo conoce como cadenas de reglas. Se definieron seis cadenas importantes.

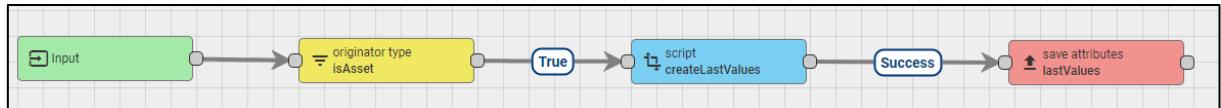
La primera cadena, mostrada en la **Figura 23**, es la cadena principal que procesa los datos de entrada y salida. Cuando se recibe un mensaje de cualquier sensor se actualiza el gráfico en tiempo real correspondiente y luego se reenvía el mensaje a las cadenas de “Alarma” y “ValidateID”. En cambio, el mensaje se redirige a la cadena “Relay” si se actualiza algún atributo de un actuador dentro de la plataforma. Por último, se activa la cadena “createAsset” cuando se crea una nueva entidad de tipo infraestructura en ThingsBoard, como laboratorios, edificios o facultades.



**Figura 23.** Cadena de regla raíz

**Fuente:** (Elaboración propia)

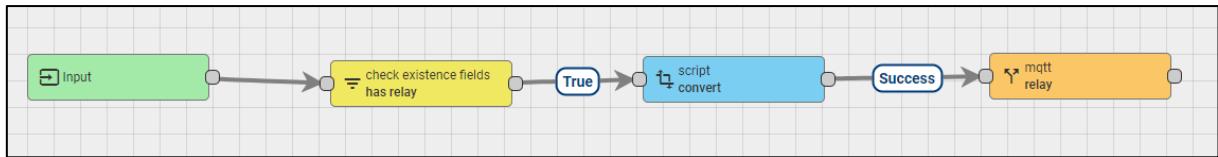
La cadena “createAsset” mostrada en la **Figura 24**, luego de validar que el mensaje corresponde a una entidad de tipo infraestructura, activa un script que la configura (facultad, edificio o laboratorio) con las variables necesarias, las cuales son: “lastValues” e “imageUrl”.



**Figura 24.** Cadena “createAsset”

**Fuente:** (Elaboración propia)

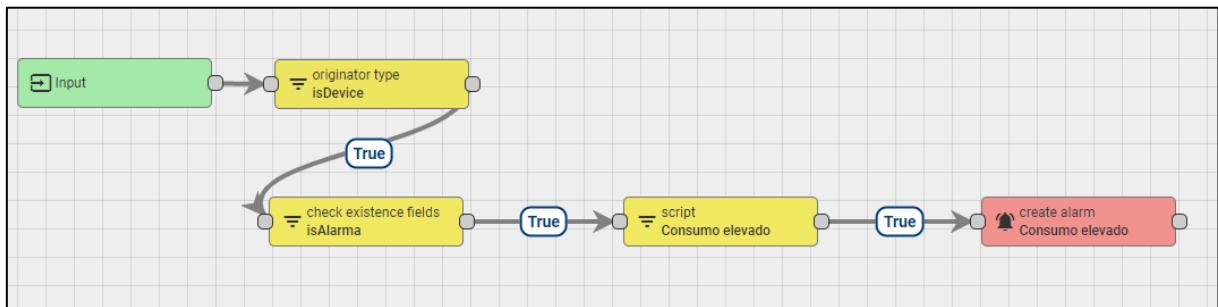
La cadena “Relay” detecta el cambio del atributo “on\_off”, lo procesa y manda un mensaje equivalente por MQTT hacia el actuador IoT para realizar la acción solicitada. El esquema se observa en la **Figura 25**.



**Figura 25.** Cadena "Relay"

**Fuente:** (Elaboración propia)

La **Figura 26** muestra la configuración de la cadena "Alarma", la cual detecta los mensajes recibidos por el algoritmo de árbol de decisión y los muestra en todos los paneles que contengan ese dispositivo, es decir, que se mostrará en la vista laboratorio, edificio y facultad.

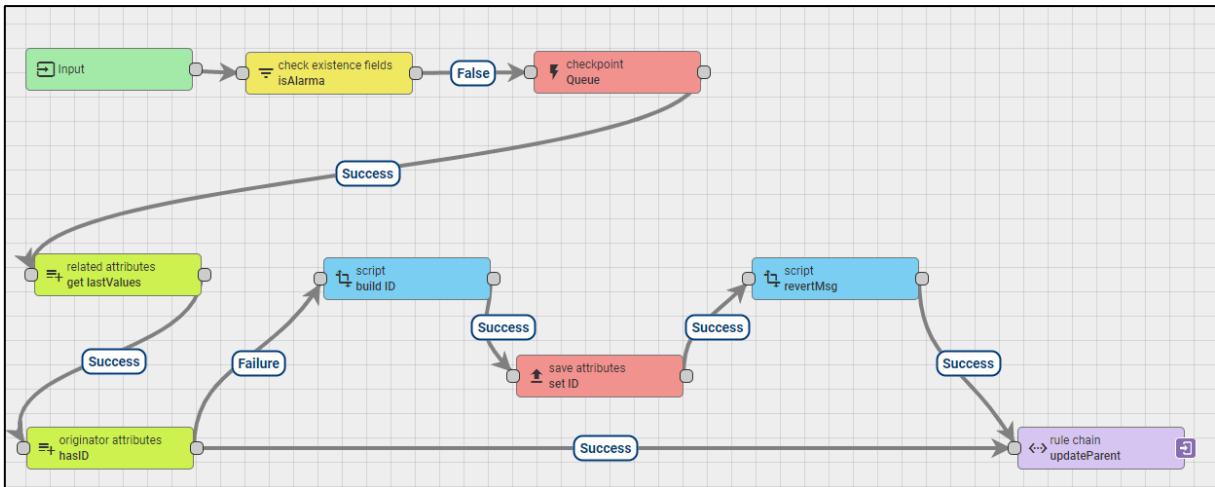


**Figura 26.** Cadena "Alarma"

**Fuente:** (Elaboración propia)

La quinta cadena ("validateID") corresponde a una validación de id, como se especificó en la sección de diseño anterior, cada entidad de tipo infraestructura le corresponde un id único dentro de su entidad contenedora correspondiente.

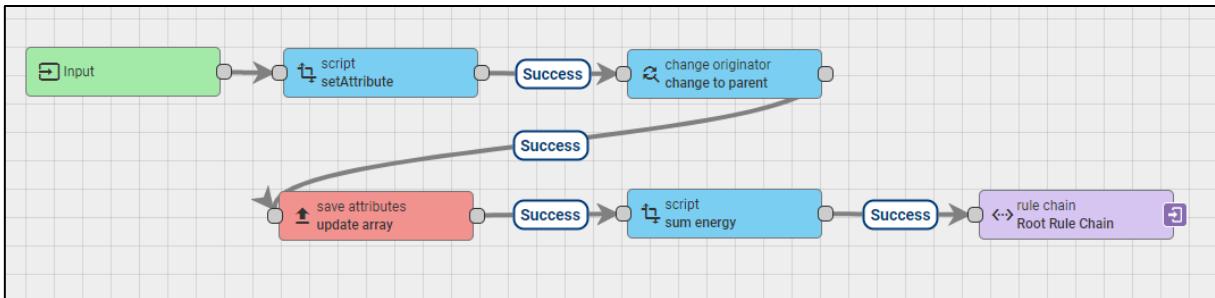
En la **Figura 27** se observa el flujo del mensaje, cuando llega un mensaje de consumo de algún sensor, inmediatamente se actualiza el valor de la entidad de tipo dispositivo; sin embargo, este mensaje debe ser recursivo hacia las entidades de tipo infraestructura que contienen dicho dispositivo, incluyendo el laboratorio, el edificio y facultad a la que pertenece. Para evitar que los mensajes recursivos afecten el funcionamiento normal de las cadenas de reglas, es necesario utilizar el bloque "Queue" que será el encargado de ir apilando todos los mensajes recursivos en el orden en que van llegando. En el momento en el que un mensaje sale de la cola ("Queue") se verifica si la entidad de tipo infraestructura posee el atributo ID, si ya lo posee entonces redirige el mensaje a la cadena "updateParent", caso contrario verifica el último ID creado anteriormente y le asigna el siguiente ID a dicha entidad y posteriormente redirige el mensaje a la siguiente cadena.



**Figura 27. Cadena “validateID”**

**Fuente:** (Elaboración propia)

En la última cadena denominada “*updateParent*” mostrada en la **Figura 28**, se observa que el mensaje inicial cambia de nivel a la entidad contenedora y actualiza el arreglo de los consumos almacenados de sus entidades contenidas, y luego realiza la suma de todos los valores de ese arreglo para actualizarse su valor total de consumo, esto lo hace reenviando el mensaje hacia la cadena raíz formando mensajes recursivos hasta que llega a la entidad máxima que corresponde a la entidad ESPOL.



**Figura 28. Cadena “*updateParent*”**

**Fuente:** (Elaboración propia)

Para finalizar la parte del desarrollo de la plataforma, se establece el medio de comunicación entre los sensores y el software. El medio seleccionado es el protocolo MQTT, se definen dos bróker MQTT: el primero envía mensajes a través del puerto 1883, este bróker viene instalado por defecto en el paquete de ThingsBoard para poder enviar y recibir datos, el gran inconveniente es que ThingsBoard espera recibir mensajes de tipo JSON en los paquetes MQTT, pero los sensores de corriente envían mensajes MQTT pero en texto plano, sin formar un objeto de tipo JSON, debido a esto fue necesario la implementación de un segundo bróker MQTT, para que no haya conflictos entre los dos bróker, el segundo proceso mensajes a

través del puerto 8883. Los sensores envían sus mensajes a este bróker, los mensajes son procesados en formato JSON y reenviados al bróker de ThingsBoard.

#### 4.3.5. Desarrollo de hardware para control remoto

Esta es la actividad final de la fase de implantación, se selecciona un emisor genérico infrarrojo que tenga conexión a Internet, concretamente el modelo Broadlink RM4C mini. Este dispositivo antes de poder enviar comandos debe aprender qué comandos enviará para el encendido y apagado de equipos infrarrojos. En la **Figura 29** se observa el proceso completo de aprendizaje mediante librerías en Python. Al igual que ocurrió con la conexión de los sensores, este actuador también debe comunicarse vía MQTT en el puerto 8883 para recibir las señales de envío procesadas desde la plataforma de ThingsBoard. Todos los comandos que aprende el dispositivo IoT no los guarda en su memoria interna, sino que los envía a una tabla en la base de datos de PostgreSQL que tiene por defecto ThingsBoard, de igual forma al momento de querer enviar un comando infrarrojo, primero debe hacer una consulta a la base de datos.

```
A043B054EA60
Aprendiendo comando de encendido de aire...
Codigo aprendido:
b'\x00X\x00\x00\x01%\x93\x12\x12\x11\x13\x11\x13\x117\x10\x14\x11\x13\x11\x13\x
10\x14\x11\x13\x117\x11\x13\x11\x13\x117\x11\x13\x11\x13\x118\x117\x11\x
12\x11\x13\x11\x13\x11\x13\x11\x13\x11\x13\x10\x14\x12\x12\x11\x13\x11\x13\x117\x
11\x13\x11\x13\x12\x12\x11\x13\x11\x13\x10\x108\x11\x13\x11\x13\x10\x14\x10\x1
4\x11\x00\r\x05'
Aprendiendo comando de apagado de aire...
Codigo aprendido:
b'\x00X\x00\x00\x01%\x93\x12\x12\x12\x11\x13\x117\x11\x13\x11\x13\x12\x12\x
11\x13\x117\x10\x14\x12\x12\x11\x13\x117\x11\x13\x11\x13\x117\x117\x11\x
13\x11\x13\x11\x13\x12\x12\x11\x13\x11\x13\x11\x13\x11\x13\x11\x13\x11\x13\x117\x
10\x14\x11\x13\x11\x13\x12\x12\x11\x13\x117\x126\x12\x12\x11\x13\x11\x13\x11\x1
3\x11\x00\r\x05'
Encendiendo Aire acondicionado...
Aire acondicionado encendido
Apagando Aire acondicionado...
Aire acondicionado apagado
```

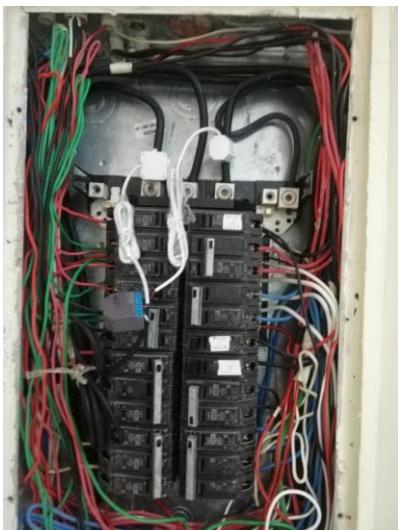
**Figura 29.** Proceso de aprendizaje de comandos infrarrojos

**Fuente:** (Elaboración propia)

#### 4.4. Ejecución

Para la fase final se migra todo el sistema a una máquina virtual en Microsoft Azure, el funcionamiento se mantiene igual que en la fase anterior, con la diferencia de que las direcciones IP deben cambiar a la dirección de la máquina virtual, o activar un nombre de dominio (*Domain Name Server - DNS*). Otro paso importante es activar el proceso de HTTPS para la visualización de la plataforma, para esto es necesario obtener certificados seguros que ThingsBoard mismo ofrece.

A nivel de Hardware se realiza la instalación del sensor Shelly EM con los sensores de corriente conectado a las dos fases principales del laboratorio como se observa en la **Figura 30**. Desde ese mismo panel, se encarga de activar o desactivar el contactor para controlar el sistema de luces. Por último, se busca el lugar ideal en el que el dispositivo IoT pueda enviar todos los comandos infrarrojos a todos los equipos, en este caso tanto al aire acondicionado como al proyector ubicado en la parte superior como se observa en la **Figura 31**.



**Figura 30.** Shelly instalado en panel eléctrico

**Fuente:** (Elaboración propia)



**Figura 31.** Dispositivo IoT controlando equipos

**Fuente:** (Elaboración propia)

##### 4.4.1. Análisis del modelo predictivo ARIMA

Se tiene dos análisis, el primero corresponde a la efectividad del algoritmo correspondiente al periodo de 2017 a 2019. En este periodo no se puede realizar un cálculo de error entre los datos predichos y los datos reales por el tema de la pandemia; sin embargo, se analiza el nivel de ahorro energético que se pudo haber logrado en caso de que no hubiera suspensión de actividades en la universidad.

Según el cálculo realizado en la sección 4.3.3, ESPOL progresivamente ha ido disminuyendo su consumo a lo largo del tiempo, pero de una manera lenta. Sin ningún sistema IoT, lograría

un ahorro del 10% del año 2017 luego de 5 años y 7 meses aproximadamente, en cambio analizando el retorno de inversión con la instalación del sistema IoT en los 85 edificios de ESPOL, ese mismo ahorro se logra al año y cuatro meses, es decir que la implementación del sistema logra un ahorro económico en un 76% menos al tiempo que se tomaría sin implementar un sistema IoT.

Para el segundo análisis se considera el segundo periodo de entrenamiento correspondiente al año 2020. Este modelo entrenado predice los consumos del año 2021 hasta el mes de agosto. En la **Tabla 21** se muestran los errores porcentuales de cada predicción realizada, dando un error promedio de 2.67%. Este promedio es un error aceptable, considerando que sólo se tiene un periodo de datos correspondiente al año 2020, al finalizar el año actual el modelo se vuelve a entrenar con los datos reales de 2020 y 2021 para una mejor predicción del año 2022 y así sucesivamente.

Tabla 21. Errores porcentuales entre consumos reales y consumos calculados

Fecha	Consumo real	Consumo calculado	Error porcentual
Enero 2021	\$30.594,09	\$30.065,77	1,757%
Febrero 2021	\$32.470,55	\$31.908,13	1,763%
Marzo 2021	\$33.507,75	\$33.411,72	0,287%
Abril 2021	\$27.718,45	\$29.916,53	7,347%
Mayo 2021	\$30.663,53	\$31.763,76	3,464%
Junio 2021	\$33.372,00	\$33.862,14	1,447%
Julio 2021	\$34.449,67	\$33.673,71	2,304%
Agosto 2021	\$34.689,90	\$33.680,70	2,996%

Fuente: (Elaboración Propia)

#### 4.4.2. Análisis del modelo de clasificación

A diferencia del algoritmo predictivo, este modelo se modelo de clasificación se entrena a partir de datos de potencia eléctrica a nivel de un laboratorio. No se ha podido realizar comprobaciones reales debido a que el atributo “feriado/vacaciones” está activado para este mes, por lo que cualquier consumo superior a 1kW en esta época del año genera un resultado de Consumo excedido. Pero haciendo el análisis con el 20% de datos anteriores se obtiene un porcentaje de acierto del 97.5% como se indicó al final de la sección 4.3.3.

#### 4.4.3. Análisis de escenario de consumo excedido

Luego de instalarse los equipos en el panel eléctrico y el dispositivo IoT cerca de los equipos infrarrojos. Se tiene el escenario ideal para realizar pruebas de efectividad a nivel de hardware y software.

Al forzar consumos excedidos en el laboratorio, el tiempo de retardo entre que se clasifica como consumo excedido y la activación de la alarma en la plataforma IoT, es menor a 1 segundo. Este análisis es exclusivo para la implementación de un solo laboratorio, en la siguiente sección se mencionan los posibles trabajos futuros para mejorar el desempeño y evaluar escenarios donde el envío de datos sea mayor escala.

## 5. Conclusiones y trabajos futuros

Como fue mencionado en la introducción, el problema principal que ataca este proyecto es el de facturas elevadas en la planilla eléctrica en instituciones de educación superior al no tener una noción cercana de lo que se está consumiendo.

Gracias al desarrollo del sistema IoT para monitorización, control y predicción de consumo eléctrico realizado en este TFM dicho problema queda resuelto parcialmente, tras la implementación de un sistema IoT en un laboratorio de toda la universidad para poder evaluar los requerimientos funcionales y sobre todo los no funcionales.

Las principales contribuciones obtenidas tras la realización de este proyecto son las siguientes:

1. Desarrollar una plataforma IoT que se accesible desde cualquier sitio con acceso a internet al personal administrativo y docente de la ESPOL, que permita no solo monitorizar el consumo energético sino también poder controlar los equipos de mayor consumo de forma remota.
2. Ejecutar algoritmo de aprendizaje automático, que es capaz de predecir consumos futuros a un índice de efectividad del 97% en base a todos los consumos generados anteriormente en toda la ESPOL.
3. Ejecutar algoritmo de clasificación, que genera alarmas a nivel de laboratorio cuando determina que el consumo actual está siendo desperdiciado.

Teniendo en cuenta estos resultados, ha quedado demostrado en el desarrollo del TFM que han sido alcanzados los objetivos específicos planteados ya que:

Objetivo específico 1. Compilar datos de energía consumida mediante instalación de sensores en el centro de carga del edificio con el fin de conocer el actual consumo de energía eléctrica y optimizar el uso de equipos eléctricos.

Este objetivo se considera alcanzado ya que, tal y como queda reflejado en el apartado 4.3.2 de la memoria y según los resultados obtenidos, se logró seleccionar el mejor sensor acorde a las necesidades de la universidad para su posterior envío de información.

Objetivo específico 2. Utilizar un algoritmo de aprendizaje automático apropiado para estimar el consumo energético de los siguientes meses y reducir costos.

Este objetivo se alcanza en la sección 4.3.3, donde se evidencia valores predichos muy cercanos a los reales obtenidos en los primeros meses de 2021

Objetivo específico 3. Determinar consumos eléctricos innecesarios a través de un algoritmo de clasificación para generar alertas al usuario final.

Este objetivo se cumple en la misma sección del objetivo 2, se observa una precisión del 97% de datos clasificados como consumos innecesarios o consumos normales.

Objetivo específico 4. Visualizar datos y predicciones obtenidos en tiempo real mediante una plataforma web pública para toma de decisiones preventivas a nivel energético.

Este objetivo se considera alcanzado ya que, tal y como queda reflejado en el apartado 4.3.4 de la memoria y según los resultados obtenidos, el personal de ESPOL puede tener acceso a la plataforma pública alojada en los servidores de Microsoft Azure en todo momento para visualizar los consumos energéticos.

Objetivo específico 5. Crear un sistema de control de luminarias y aires acondicionados con un módulo inalámbrico integral operado desde una plataforma web pública con la finalidad de moderar el consumo innecesario.

Este objetivo queda desarrollado en la sección 4.3.5 y demostrado en la sección 4.4 cuando desde la plataforma web pública, se logra activar y desactivar las luces y los dispositivos infrarrojos.

Por consiguiente, se puede concluir que el objetivo principal, desarrollar un sistema de monitorización, control y predicción de consumo energético de un edificio para el ahorro en la factura mensual mediante sistemas basados en la nube, ha sido alcanzado tras la realización de este TFM.

## 5.1. “LÍNEAS DE TRABAJO FUTURAS”

Este trabajo se desarrolló con el fin de tener una opción para disminuir el consumo energético, pero se efectuó una etapa inicial en la que se ejecuta el desarrollo de un solo laboratorio, está implementado para ser un producto escalable; sin embargo, para llevar el producto a mayor escala, a 85 edificios, es necesario primero evaluar la cantidad de datos que se van a enviar y en cuanto tiempo, para poder redimensionar el proyecto. Esto implica mayor capacidad de procesamiento de Microsoft Azure, mayor espacio en el disco duro, implementar redundancia en los datos en caso de que exista pérdida de información.

Otro posible trabajo futuro, sería abarcar más controles a equipos electrónicos, dispositivos que no trabajen con señales infrarrojas, sino por ejemplo señales de radio frecuencia, también se puede considerar desarrollar programas que permiten el encendido y apagado de computadoras a través de la red local, para mejorar aún más el consumo energético de laboratorios.

Finalmente, para evitar costos a nivel de hardware, se puede realizar un estudio sobre agregación y desagregación de carga eléctrica, es decir que a partir de un solo medidor se elaboren técnicas que permitan diferenciar todas las cargas conectadas al panel.

## Referencias bibliográficas

- Bartnitsky, J. (23 de Enero de 2018). *HTTP vs MQTT performance tests*. Obtenido de <https://flespi.com/blog/http-vs-mqtt-performance-tests>
- Basco, A., Beliz, G., Coatz, D., & Garnero, P. (2018). *Industria 4.0: Fabricando el Futuro*. Buenos Aires.
- C.A. EL UNIVERSO. (26 de Mayo de 2021). *De enero a mayo CNEL ha recibido 68.153 reclamos por facturación en el país* | Comunidad | Guayaquil | El Universo. Obtenido de <https://www.eluniverso.com/noticias/seguridad/de-enero-a-mayo-cnel-ha-recibido-68153-reclamos-por-facturacion-en-todo-el-pais-nota/>
- Caicedo, O. (2020). *App para monitorizar centrales de generación eléctrica*. Quito. Obtenido de <https://reunir.unir.net/bitstream/handle/123456789/10720/Caicedo%20Ruales%2c%20Oscar.pdf>
- Castorena, J., Silva, A., Martínez, R., Campos, G., & Martínez, J. (2020). Implementación de las técnicas de predicción en la generación de energía eléctrica en el sector industrial. *Revista de Investigación en Tecnologías de la Información: RITI*, 8(15), 16-26. doi:10.36825/RITI.08.15.003
- CEPAL. (2020). *Informe sobre impacto económico en América Latina y el Caribe de la enfermedad por coronavirus (COVID-19)*. Santiago. Obtenido de [https://repositorio.cepal.org/bitstream/handle/11362/45602/1/S2000313\\_es.pdf](https://repositorio.cepal.org/bitstream/handle/11362/45602/1/S2000313_es.pdf)
- Deb, C., Zhang, F., Yang, J., Eang, S., & Wei, K. (2017). A review on time series forecasting techniques for building energy consumption. *Renewable and Sustainable Energy Reviews*, 74, 902-924. doi:10.1016/j.rser.2017.02.085
- Desiderio, K. (2020). *Metodología para la evaluación efectiva de pruebas de penetración para la aplicación en el desarrollo de aplicaciones web*. Guayaquil. Obtenido de <https://reunir.unir.net/bitstream/handle/123456789/10879/DESIDERIO%20SANCHEZ%2c%20KATTYA%20ROXANA.pdf>
- Eagle Electric. (2019). *¡Ya podés controlar tus electrodomésticos y luminarias mediante un app!* Obtenido de <https://eagle.cr/ya-podes-controlar-tus-electrodomesticos-y-luminarias-mediante-un-app/>
- E-Marmolejo, R. (12 de Noviembre de 2017). *Microcontrolador - qué es y para qué sirve*. Obtenido de <https://hetpro-store.com/TUTORIALES/microcontrolador/>
- FDominec. (4 de Enero de 2007). *Electronic\_component\_relays*.

- Feltham, P. (2014). Real-time Monitoring is Key to Reducing Energy Consumption. *Electricity Today*. Obtenido de <https://www.electricity-today.com/marketplace/real-time-monitoring-is-key-to-reducing-energy-consumption>
- Forcén, P. (2020). *Diseño y prototipado de una aplicación para reducir la huella ecológica*. Valencia. Obtenido de <https://reunir.unir.net/bitstream/handle/123456789>
- García-Martín, E., Faviola, C., Riley, G., & Grahn, H. (2019). Estimation of energy consumption in machine learning. *Journal of Parallel and Distributed Computing*, 134, 75-88. doi:10.1016/j.jpdc.2019.07.007
- Hetal Bhavsar, A. (Septiembre de 2012). A Comparative Study of Training Algorithms for Supervised Machine Learning. *International Journal of Soft Computing and Engineering (IJSCe)*, 2(4). Obtenido de <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.492.6088>
- Hillar, G. (2017). *MQTT Essentials - A Lightweight IoT Protocol*. Birmingham: Packt Publishing.
- Huang, J., Tang, Y., & Chen, S. (2018). Energy Demand Forecasting: Combining Cointegration Analysis and Artificial Intelligence Algorithm. *Mathematical Problems in Engineering*, 2018. doi:10.1155/2018/5194810
- Jiang, Y., Perng, C.-s., Li, T., & Chang, R. (2012). *Intelligent Cloud Capacity Management*. doi:10.1109/NOMS.2012.6211941
- Karjalainen, S. (2011). Consumer preferences for feedback on household electricity consumption. *Energy and Buildings*, 43, 458-467. doi:10.1016/j.enbuild.2010.10.010
- King, J., & Perry, C. (2017). *Smart Buildings: Using Smart Technology to Save Energy in Existing Buildings*. Washington D.C. Obtenido de <https://eecoordinator.info/wp-content/uploads/2017/03/Smart-Building-Report.pdf>
- Lesic, V., de Bruin, W., Davis, M., Krishnamurti, T., & Azevedo, I. (Marzo de 2018). Consumers' perceptions of energy use and energy savings: A literature review. *Environmental Research Letters covers all of environmental science*, 13(3). doi:10.1088/1748-9326/aaab92
- Naik, N. (2017). Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. *2017 IEEE International Systems Engineering Symposium (ISSE)*, 1-7. doi:10.1109/SysEng.2017.8088251
- Postscapes. (2015). *What Is The "Internet of Things"?* Obtenido de Postscapes Smart Devices & Systems: <https://www.postscapes.com/what-exactly-is-the-internet-of-things-infographic/>

- Román, J. L. (2006). *Industria 4.0: la transformación digital de la industria*. Bilbao. Obtenido de <http://coddii.org/wp-content/uploads/2016/11/Informe-CODDII-Industria-4.pdf>
- ThingsBoard. (9 de Marzo de 2021). *ThingsBoard - Open-source IoT Platform*. Obtenido de <https://thingsboard.io/>
- ThingsBoard. (2021). *ThingsBoard Professional Edition | ThingsBoard*. Obtenido de <https://thingsboard.io/products/thingsboard-pe/>
- Woo, Z., & Roper, W. (2009). Energy demand estimation of South Korea using artificial neural network. *Energy Policy*, 4049-4054. doi:10.1016/j.enpol.2009.04.049

## Anexo A Código implementado para adquisición de datos

```
client.on('message', function (topic, message) {
  const categoria = topic.split('/')[4];
  const sensor = parseInt(topic.split('/')[3]);
  const shelly_id = parseInt(topic.split('/')[1].split('-')[1], 16);
  if (topic == "shellies/shellyem-B9E151/attributes") {
    let msj = new String(message)
    msj = msj.substring(1, msj.length - 1); //Elimina las comillas dobles inciales y finales
    client.publish("shellies/shellyem-B9E151/relay/0/command", msj);
  }
  else if(topic == "ML/"){
    let msj = new String(message)
    console.log("Enviando a ThingsBoard: " + msj);
    clientTB.publish("shellies/shellyem-B9E151/emeter", msj.toString());
  }
  else {
    flag++;
    switch (categoria) {
      case 'power':
        if (sensor == 0) datas.power0 = parseFloat(message);
        else if (sensor == 1) datas.power1 = parseFloat(message);
        break;
      case 'total':
        if (sensor == 0) datas.total0 = parseFloat(message);
        else if (sensor == 1) datas.total1 = parseFloat(message);
        break;
      default:
        console.log("Mensaje: " + message + " , 'topico: " + topic + "'");
        flag--;
        break;
    }
    if (flag == 4) {
      console.log("Enviando a ThingsBoard: " + JSON.stringify(datas));
      clientTB.publish("shellies/shellyem-B9E151/emeter", JSON.stringify(datas));
      flag = 0;
    }
  }
});
```

## Anexo B Datos de entrenamiento Grupo 1

Fecha	Consumo mes	Datos Desestacionalizados
ene-17	\$ 70,938.47	\$ 69,074.58
feb-17	\$ 66,555.20	\$ 71,032.66
mar-17	\$ 65,998.27	\$ 71,032.62
abr-17	\$ 58,919.14	\$ 68,789.84
may-17	\$ 78,100.79	\$ 68,577.54
jun-17	\$ 76,330.80	\$ 71,640.47
jul-17	\$ 72,034.10	\$ 68,832.46
ago-17	\$ 70,975.08	\$ 70,232.13
sep-17	\$ 65,235.37	\$ 70,427.27
oct-17	\$ 67,701.20	\$ 73,536.56
nov-17	\$ 74,220.38	\$ 70,548.03
dic-17	\$ 77,653.63	\$ 71,241.69
ene-18	\$ 73,938.47	\$ 71,995.76
feb-18	\$ 62,555.20	\$ 66,763.56
mar-18	\$ 64,027.00	\$ 68,910.98
abr-18	\$ 59,055.00	\$ 68,948.46
may-18	\$ 78,434.00	\$ 68,870.12
jun-18	\$ 72,584.00	\$ 68,123.90
jul-18	\$ 72,077.00	\$ 68,873.46
ago-18	\$ 69,441.00	\$ 68,714.11
sep-18	\$ 64,741.00	\$ 69,893.56
oct-18	\$ 64,115.00	\$ 69,641.26
nov-18	\$ 70,814.58	\$ 67,310.75
dic-18	\$ 73,862.66	\$ 67,763.75
ene-19	\$ 67,550.79	\$ 65,775.91
feb-19	\$ 64,697.55	\$ 69,050.04
mar-19	\$ 62,161.02	\$ 66,902.66
abr-19	\$ 59,191.60	\$ 69,107.95
may-19	\$ 79,035.85	\$ 69,398.59
jun-19	\$ 71,473.77	\$ 67,081.89
jul-19	\$ 72,356.29	\$ 69,140.33
ago-19	\$ 68,618.29	\$ 67,900.01
sep-19	\$ 61,621.17	\$ 66,525.43
oct-19	\$ 58,616.14	\$ 63,668.43
nov-19	\$ 72,578.58	\$ 68,987.47
dic-19	\$ 73,946.66	\$ 67,840.81

## Anexo C Datos de entrenamiento Grupo 2

Fecha	Consumo mes	Datos Desestacionalizados
ene-20	\$29,594.09	\$31,983.50
feb-20	\$31,470.55	\$32,015.21
mar-20	\$33,507.75	\$32,523.87
abr-20	\$32,347.50	\$35,030.35
may-20	\$33,174.00	\$33,802.90
jun-20	\$34,752.00	\$33,182.71
jul-20	\$33,362.00	\$32,002.20
ago-20	\$33,205.00	\$31,812.55
sep-20	\$32,924.00	\$32,523.87
oct-20	\$34,824.33	\$32,523.87
nov-20	\$34,277.07	\$32,523.87
dic-20	\$29,573.07	\$32,523.87

## Anexo D Muestra de datos de entrenamiento para árbol de decisiones

Hora	Consumo (W)	FinDeSemana	Feriado/Vacaciones	HorarioLaboral	EXCESO
0	459,7	0	0	0	0
1	457,9	0	0	0	0
2	456,1	0	0	0	0
3	458,5	0	0	0	0
4	458,6	0	0	0	0
5	438	0	0	0	0
6	457,2	0	0	0	0
7	913,6	0	0	1	0
8	1484,8	0	0	1	0
9	2262,3	0	0	1	0
10	1884,3	0	0	1	0
11	2052,8	0	0	1	0
12	3568	0	0	1	0
13	2251,8	0	0	1	0
14	3527,4	0	0	1	0
15	3063,2	0	0	1	0
16	2838,7	0	0	1	0
17	2286,4	0	0	1	0
18	1472,8	0	0	1	0
19	1203	0	0	1	0
20	1119,5	0	0	0	1
21	644,1	0	0	0	0
22	506,9	0	0	0	0
23	508,4	0	0	0	0
0	504,5	0	0	0	0
1	712,8	0	0	0	0
2	491	0	0	0	0
3	490,9	0	0	0	0
4	490,2	0	0	0	0
5	488,4	0	0	0	0
6	615,6	0	0	0	0
7	893,5	0	0	1	0
8	2562	0	0	1	0
9	3000,2	0	0	1	1
10	2655,3	0	0	1	0
11	2471,9	0	0	1	0
12	1965,2	0	0	1	0
13	2747,4	0	0	1	0

14	2927	0	0	1	0
15	2756,2	0	0	1	0
16	1745	0	0	1	0
17	1168,4	0	0	1	0
18	843,6	0	0	1	0
19	951,5	0	0	1	0
20	571,4	0	0	0	0
21	566,8	0	0	0	0
22	557,2	0	0	0	0
23	553	0	0	0	0
0	536,6	0	0	0	0
1	545,5	0	0	0	0
2	712,4	0	0	0	0
3	748,8	0	0	0	0
4	760,4	0	0	0	0
5	752,7	0	0	0	0
6	583	0	0	0	0
7	842,7	0	0	1	0
8	1787,5	0	0	1	0
9	2112,3	0	0	1	0
10	1920,2	0	0	1	0
11	1893,9	0	0	1	0
12	1772,5	0	0	1	0
13	2881,7	0	0	1	0
14	3454,7	0	0	1	0
15	2929,2	0	0	1	0
16	2511,3	0	0	1	0
17	1567,9	0	0	1	0
18	1527,8	0	0	1	0
19	1018,5	0	0	1	0
20	708,6	0	0	0	0
21	669,3	0	0	0	0
22	542,4	0	0	0	0
23	525,7	0	0	0	0

## Anexo E Código implementado para entrenamiento del árbol de decisión

```
#!/usr/bin/env python3

import numpy as np
import csv
import time
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import precision_score
from sklearn.tree import export_graphviz
from joblib import dump, load

def load_my_dataset():
    with open('dataset2.csv', encoding='utf-8-sig') as csv_file:
        data_file = csv.reader(csv_file)
        temp = next(data_file)
        n_samples = int(temp[0].strip())
        n_features = int(temp[1].strip())
        data = np.empty((n_samples, n_features))
        target = np.empty((n_samples,), dtype=np.int32)
        for i, sample in enumerate(data_file):
            data[i] = np.asarray(sample[:-1], dtype=np.float64)
            target[i] = np.asarray(sample[-1], dtype=np.int32)

    return data, target

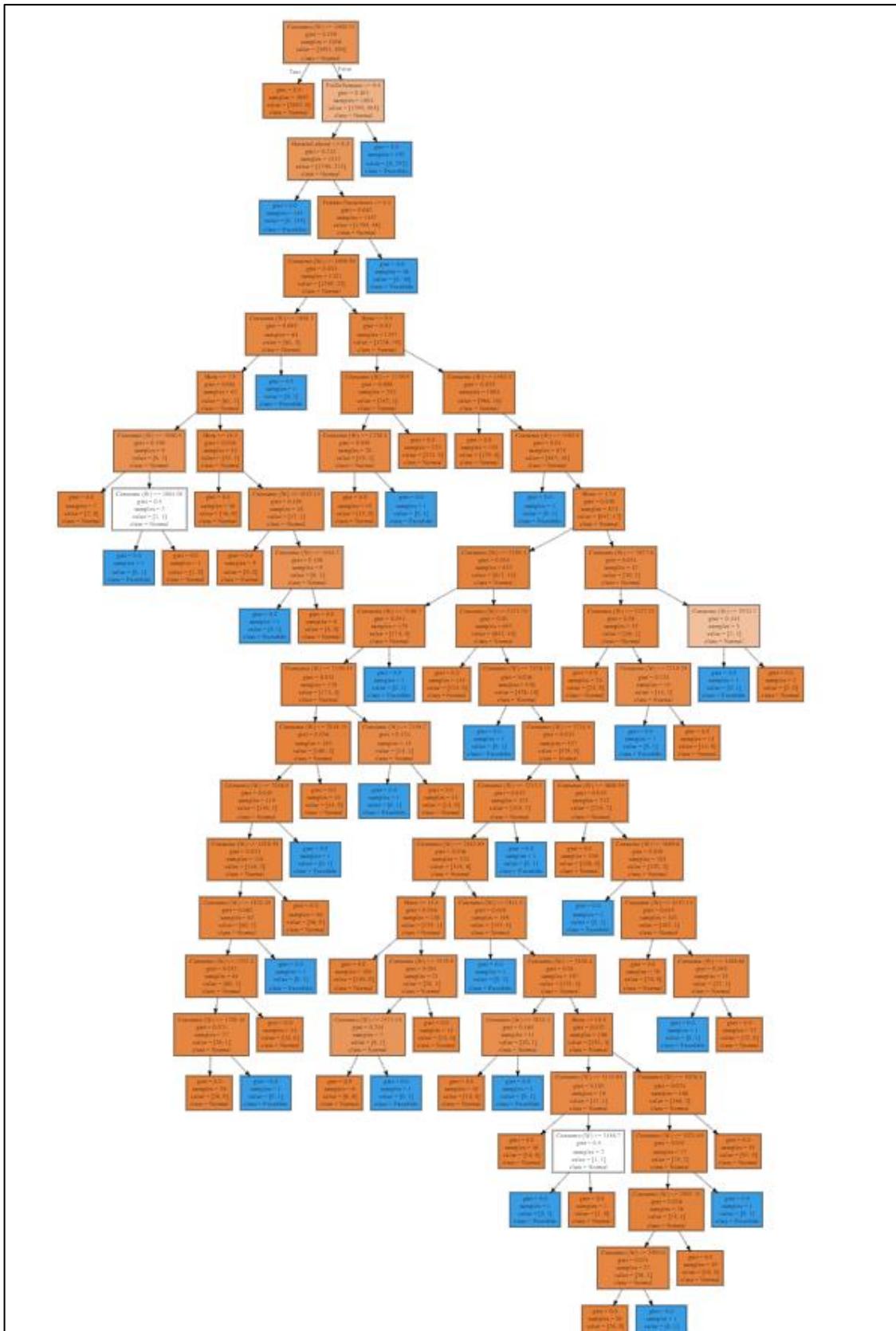
def save_tree(max_precision):
    precision=0
    X,y = load_my_dataset()
    fn=['Hora','Consumo (W)','FinDeSemana','Feriado/Vacaciones','HorarioLaboral']
    cn=['Normal', 'Excedido']

    while precision<max_precision:
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
        algoritmo = DecisionTreeClassifier()
        algoritmo.fit(X_train, y_train)
        y_pred = algoritmo.predict(X_test)
        matriz = confusion_matrix(y_test, y_pred)
        precision = precision_score(y_test, y_pred)
        print(precision)

        dump(algoritmo, 'filename.joblib')
        export_graphviz(algoritmo,out_file="tree.dot", feature_names = fn, class_names=cn, filled = True);
        print('Matriz de Confusión:')
        print(matriz)
        print('Precision del modelo:')
        print(precision)
        print(X_test)


```

## Anexo F Árbol de decisión sin límites de profundidad



## Anexo G Imágenes de navegación en ThingsBoard

The screenshot shows the ThingsBoard interface for the FIEC building. On the left, there is a map of the building complex with several buildings labeled 11A, 11B, 11C, 11D, and 11E. A red marker is placed on building 11C. Below the map is a table for 'Edificios' showing consumption data for 'Edificio 11-C' at 15.96 kWh. To the right is a large orange box displaying 'Consumo Total 15.96 kWh' and a blue circular gauge. The bottom right corner of the interface is labeled 'Powered by Thingsboard v3.2.1'.

The screenshot shows the ThingsBoard interface for the 'Edificio 11-C' under 'Laboratorio de Sistemas Telemáticos'. On the left, there is a map of the building complex with a red marker on building 11C. Below the map is a table for 'Laboratorios' showing consumption data for 'Laboratorio de Sistemas Telemáticos' at 15.96 kWh. To the right is a large orange box displaying 'Consumo Total 15.96 kWh' and a blue circular gauge. The bottom right corner of the interface is labeled 'Powered by Thingsboard v3.2.1'.

The screenshot shows the ThingsBoard interface for the 'Edificio 11-C' under 'Laboratorio de Sistemas Telemáticos'. On the left, there is a detailed floor plan of the laboratory with various equipment labeled: 'Aire Acondicionado (LST)', 'Proyector (LST)', and 'ShellyEM-B9E151'. A red marker is placed on the floor plan. Below the floor plan is a 'New Timeseries - Plot' showing consumption data for 'ShellyEM-B9E151' over the last 5 minutes. The plot shows two lines: 'Fase 1' (blue) and 'Fase 2' (green), with a 'promedio' (average) line. To the right is a table for 'Sensores' showing consumption data for 'ShellyEM-B9E151' at 15.75 kWh and 0.25 kWh. Below the plot is a table for 'Actuadores' showing consumption data for 'Aire Acondicionado (LST)' and 'Proyector (LST)'. The bottom right corner of the interface is labeled 'Powered by Thingsboard v3.2.1'.