

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

**IDENTIFICACIÓN DE KPI Y PRUEBAS DE FUNCIONALIDAD,
USABILIDAD Y SEGURIDAD**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

WILLIAN JACOB RIVERA MERA

willian.rivera@epn.edu.ec

DIRECTOR: MARCO POLO SÁNCHEZ AGUAYO

marco.sanchez01@epn.edu.ec

Quito, agosto 2024

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Willian Jacob Rivera Mera, bajo mi supervisión.

Ing. Marco Polo Sánchez Aguayo
DIRECTOR DE PROYECTO

DECLARACIÓN

Yo, Willian Jacob Rivera Mera, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Willian Jacob Rivera Mera

DEDICATORIA

Este proyecto lo dedico a quienes un día creyeron en mí, a aquellos que dieron su tiempo y esfuerzo en brindarme sus consejos y buenos deseos. A todos los que un día miraron en mi dirección, alentándome a seguir adelante, permitiéndome, a pesar de las circunstancias, culminar esta gran etapa de mi vida.

Quiero también dedicar esta obra a todo el pueblo ecuatoriano, ya que gracias a sus esfuerzos, los estudiantes de todos los sectores públicos pueden estudiar y formarse. Esta dedicatoria es para todo el pueblo trabajador y honesto, que aporta su grano de esperanza en las nuevas generaciones.

Dedico este proyecto a mis familiares y seres queridos, porque ellos me han levantado emocionalmente para lograr hacer esto posible.

Muchas gracias y bendiciones a todos.

AGRADECIMIENTOS

Quiero agradecer en primer lugar a mi tío Ernesto Rivera, quien me abrió las puertas de su hogar cuando no tenía dónde ir. Le agradezco porque fue gracias a él, a sus métodos de estudio y a sus consejos que hoy día me encuentro donde estoy. Muchas gracias a mis tíos por siempre extenderme la mano.

Quiero agradecer a mis padres por la formación que me brindaron, la cual me permitió ser quien soy hoy; y a mis hermanas por creer en mí hasta el final. Agradezco a Eloisa, porque ver sus ojos llenos de felicidad al verme llegar me llenaba el corazón de alegría y me hacía sentir amado en mis épocas de soledad. Quiero agradecer a mi pareja sentimental quien estuvo a mi lado en buenas y malas apoyandome.

Quiero agradecer a mi madre, por ser una persona fuerte que me enseñó a levantarme siempre que caía, y a mi hermana, que daba sus mejores palabras de aliento en el momento exacto.

CONTENIDO

| | |
|--|----------|
| Resumen | 1 |
| Abstract | 2 |
| 1 INTRODUCCIÓN | 3 |
| 1.1 Descripción del Proyecto | 4 |
| 1.2 Descripción del Componente Desarrollado | 4 |
| 1.2.1 Definición de Claves Indicadores de Rendimiento | 4 |
| 1.2.2 Programación y creación del Dashboard | 5 |
| 1.2.3 Realización de pruebas de calidad del proyecto | 5 |
| 1.3 Objetivo General | 5 |
| 1.4 Objetivos Específicos | 5 |
| 1.5 Alcance | 6 |
| 1.6 Marco Teórico | 6 |
| 1.6.1 Indicadores Claves de Rendimiento (KPIs) | 6 |
| 1.6.2 Metodología de Desarrollo iterativa e incremental | 7 |
| 2 METODOLOGÍA | 8 |
| 2.1 Revisión de la Literatura Sobre Gestión Energética en el Ecuador | 9 |
| 2.1.1 Establecimiento de KPIs | 10 |
| 2.2 Entrevista con Constructoras | 13 |
| 2.2.1 Creación de encuesta | 13 |
| 2.2.2 Encuesta a los representantes de constructoras | 13 |
| 2.2.3 Análisis de resultados | 14 |
| 2.3 Selección de lenguaje de programación y herramientas de desarrollo | 14 |
| 2.3.1 Lenguaje de programación | 14 |
| 2.3.2 Herramientas de desarrollo | 15 |
| 2.4 Análisis y Diseño | 16 |
| 2.4.1 Historias de Usuario | 16 |
| 2.4.2 MockUp en Figma | 20 |
| 2.5 Desarrollo del prototipo | 22 |
| 2.5.1 Login | 22 |

| | | |
|----------|---|-----------|
| 2.5.2 | Registro | 23 |
| 2.5.3 | DashBoard | 23 |
| 2.5.4 | Módulo Devices | 25 |
| 2.5.5 | Módulo Energy Comsumption | 27 |
| 2.5.6 | Módulo CO2 Emissions | 28 |
| 2.5.7 | Módulo Energy Savings | 29 |
| 2.5.8 | Módulo Cost Savings | 30 |
| 2.6 | Pruebas Del Prototipo | 31 |
| 2.6.1 | Pruebas Estáticas | 31 |
| 2.6.2 | Pruebas Dinámicas | 34 |
| 3 | RESULTADOS Y DISCUSIÓN | 37 |
| 3.1 | Sección Login | 38 |
| 3.2 | Sección Register | 38 |
| 3.3 | Sección Dashboard | 39 |
| 3.3.1 | Módulo Devices | 40 |
| 3.3.2 | Módulo Consumption | 40 |
| 3.3.3 | Módulo Emissions CO ₂ | 41 |
| 3.3.4 | Módulo Energy Savings | 42 |
| 3.3.5 | Módulo Cost Savings | 42 |
| 3.4 | Resultados de Usabilidad | 43 |
| 4 | CONCLUSIONES | 45 |
| 5 | REFERENCIAS BIBLIOGRÁFICAS | 47 |
| 6 | ANEXOS | I |
| 6.1 | Entrevista con Constructoras | I |
| 6.1.1 | Percepciones y Preferencias sobre Edificios Inteligentes en la Industria de la Construcción | I |
| 6.1.2 | Análisis de resultados | I |
| 6.1.3 | Visualización de resultados en PowerBI | I |
| 6.2 | ANÁLISIS Y DISEÑO | II |
| 6.2.1 | Desarrollo del prototipo | II |
| 6.3 | Pruebas | II |
| 6.3.1 | Ejecución en tiempo real con estudiantes | II |
| 6.3.2 | Encuesta SUS para estudiantes | III |
| 6.3.3 | Reporte de pruebas | III |

RESUMEN

En este proyecto se decidió realizar una investigación acerca del consumo energético dentro del Ecuador para establecer KPIs capaces de realizar cálculos en tiempo real para un sistema de gestión energética dentro de una simulación en el edificio de la Facultad de Ingeniería en Sistemas. Esta simulación será capaz de enviar información a la base de datos, donde el componente BackEnd la tomará para realizar cálculos en tiempo real, como el consumo energético, emisiones de CO₂, ahorro económico, etc. Una vez realizados los cálculos, se desarrolló un FrontEnd amigable capaz de mostrar dicha información en tiempo real. Por último, se realizaron pruebas dinámicas y estáticas del proyecto integrado.

ABSTRACT

This document aims to develop a FrontEnd for energy management within a university infrastructure. It is based on the research of Key Performance Indicators (KPIs) for energy management in Ecuador and includes both static and dynamic testing phases of the integrated final project.

Keywords: FrontEnd, Management, Energy, KPIs, Testing, Design, System

1 INTRODUCCIÓN

En esta época nos enfrentamos a nuevos retos dentro de nuestro entorno. Los avances tecnológicos van de la mano con los cambios en nuestras vidas, pero sobre todo en nuestro ecosistema. Después de adentrarnos en una era donde nuestro principal motor es la electricidad, desde cocinas eléctricas hasta medios de transporte eléctricos. Hoy en día se hace más notable la dependencia de dicho recurso.

La dependencia y el consumo de energía tienden a crecer. Solo en Ecuador para el año 2022, el consumo de energía eléctrica por habitante aumentó un 24.7%, alcanzando 1.569 kWh por persona. [1].

Hoy en día se vive una crisis energética debido a que nuestra mayor dependencia energética proviene de recursos hídricos, entonces debido a las sequías, el agua no abastece el consumo necesario para satisfacer la demanda de electricidad en el país.

Más allá de la falta de agua o del crecimiento poblacional, en Ecuador existe un consumo de energía descontrolado, donde no se tiene un seguimiento más allá de un medidor que indique cuánto se consumió. Sin embargo, se pueden definir nuevos sistemas que ayuden al control energético tanto en sectores comerciales como residenciales.

Que tal si todos los sectores tengan la capacidad de gestionar su consumo energético en el transcurso de los días, con el objetivo de optimizar de mayor forma su gasto tanto económico como energético. Imagínese un sistema capaz de gestionar el consumo energético mediante dispositivos inteligentes, que muestren el ahorro energético y económico que se está teniendo.

Hoy en día se vive en un mundo automatizado e inteligente para mejorar la calidad de vida de las personas. Entonces, también se puede gestionar y automatizar el consumo energético de manera inteligente mediante Smart Buildings.

1.1 DESCRIPCIÓN DEL PROYECTO

Este proyecto tiene como objetivo la creación de un prototipo basado en la gestión de edificios inteligentes con el objetivo de optimizar su consumo energético, ahorro económico y emisiones de CO2 dentro del Ecuador.

En este proyecto se desarrolló un modelo simulado para un SmartBuilding que será automatizado para garantizar la comodidad del usuario sobre la gestión energética del edificio, mediante una interfaz que comprenderá indicadores claves de rendimiento (KPIs) para la gestión energética del edificio.

El enfoque principal de este proyecto es lograr un sistema capaz de gestionar energía basándose en investigaciones desde revisión sistemática de la literatura, hasta entrevista con empresas constructoras, analizando sus problemáticas desde los costos energéticos dentro del país, hasta la necesidad de reducir su consumo, con el objetivo de evitar gastos innecesarios y poder tener una repartición energética mas estable.

1.2 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Para el desarrollo de este componente, se tomaron en cuenta 3 procesos fundamentales para el avance del proyecto.

1.2.1 Definición de Claves Indicadores de Rendimiento

En esta fase se realizó una investigación para establecer los Indicadores Clave de Rendimiento (KPIs) basados en la gestión energética, para tener una visión clara del resultado deseado en base al funcionamiento energético tanto de una conexión tradicional como de una conexión inteligente, y así saber cuándo se logró este objetivo.

La implementación de KPIs se convierte en un paso indispensable. Estos permiten cuantificar los objetivos, proporcionando una ruta clara hacia su logro y asegurando que cada paso dado se alinee con los objetivos energéticos del proyecto.

1.2.2 Programación y creación del Dashboard

Cuando se establecen los KPIs, se da paso a establecer objetivos claros para la gestión, ahorro y consumo energético. Se puede elaborar un dashboard que muestre en datos cuantificables el alcance de los objetivos establecidos para la gestión energética. Por lo tanto, este componente se enfocó en identificar los lenguajes de programación que se usarán para el desarrollo del proyecto y, a su vez, en desarrollar la implementación del dashboard.

1.2.3 Realización de pruebas de calidad del proyecto

En este componente se estableció como la fase final de todo el proyecto con el objetivo de garantizar la calidad del sistema. Estas pruebas incluyen pruebas estáticas, dinámicas de seguridad y usabilidad en que se demuestra que el proyecto realiza las funciones indicadas correctamente y también establecen una confianza en el que, en este caso, el sistema mostró información referente los Indicadores Clave de Rendimiento (KPIs) obtenidos de una manera correcta y también que sus funciones cumplan de manera correcta.

1.3 OBJETIVO GENERAL

Desarrollar un prototipo dirigido a la gestión eficiente de edificios inteligentes, que simule el funcionamiento de sus componentes y utilice un tablero de control para recoger datos de dispositivos IoT. Este sistema irá de la mano con la generación de KPIs esenciales para una administración optimizada.

1.4 OBJETIVOS ESPECÍFICOS

- Realizar un estudio detallado del consumo energético individual de cada dispositivo electrónico, para analizar profundamente orientado a la optimización de la gestión energética.
- Desarrollar Indicadores Clave de Rendimiento (KPIs) específicos para establecer metas mensurables en el diseño de módulos destinados a la gestión eficiente de edificios inteligentes, indicadores que muestren la cantidad de ahorro de costos, consumo de

energía, consumo de CO2, etc.

- ❑ Desarrollar basado a un lenguaje de programación establecido la creación del Dashboard para el manejo y gestión de un SmartBuilding con el fin de garantizar la automatización inteligente y gestión energética.
- ❑ Realizar pruebas de verificación, estáticas, dinámicas y de usabilidad dentro del sistema con el fin de garantizar la calidad dentro del sistema.

1.5 ALCANCE

Durante el proyecto, se seleccionarán metodologías de investigación para entender los objetivos y establecer metas cuantificables basadas en la gestión energética de edificios tradicionales versus inteligentes. Se definirá el lenguaje de programación adecuado y se diseñará un dashboard utilizando los KPIs obtenidos. Además, se identificarán y evaluarán herramientas para pruebas estáticas y dinámicas, garantizando calidad y seguridad. Se establecerán KPIs desde el inicio del desarrollo, y se realizarán pruebas continuas para asegurar la calidad y usabilidad del proyecto.

1.6 MARCO TEÓRICO

En esta sección se habla sobre los temas importantes utilizados para investigación y desarrollo dentro del proyecto.

1.6.1 Indicadores Claves de Rendimiento (KPIs)

Los KPI (Key Performance Indicator) son indicadores que permiten cuantificar la efectividad y eficiencia dentro del desarrollo de un proceso, esto ayuda a poder tener una idea cuantitativa del logro de un objetivo, con el fin de tener una visión clara del desempeño de estos objetivos para así tener un mejor rumbo a lo largo del desarrollo de un objetivo [2].

Ahora basándose en un contexto energético dentro de edificios. Los KPIs ayudan a establecer objetivos cuantificables para la gestión energética, por ejemplo medir el ahorro de energía para evaluar el rendimiento de las mejoras que se están realizando [3].

Unos ejemplos podrían ser:

- Ahorro de energía
- Recuperación de inversión
- Emisiones de CO2
- Consumo total energético

1.6.2 Metodología de Desarrollo iterativa e incremental

Al hablar de una metodología para el desarrollo de un sistema, en el área evolutiva puede dividirse en dos metodologías que pueden ir complementando entre ellas.

El modelo incremental se basa en crear un prototipo pequeño, el cual irá creciendo conforme las nuevas versiones, agregando nuevas funcionalidades, esta a su vez aplica secuencias lineales donde cada secuencia será un incremento.

El modelo iterativo, en cambio, se basa en la creación de prototipos ejecutables y evaluables, donde cada nueva iteración supone una mejora y un cambio. Al igual que el modelo en cascada, cada iteración será un pequeño ciclo de vida del sistema de software [4].

Al tener una combinación de los dos modelos, el iterativo e incremental, esta metodología se centra en crear pequeñas partes del sistema de software que se pueden ir mejorando de manera iterativa. Esto ayuda a identificar problemas y resolverlos lo antes posible. Este modelo facilita la creación de un sistema que sea fácil de diseñar, implementar y evaluar de manera temprana [5].

2 METODOLOGÍA

En esta sección se detallaron las etapas que se llevaron a cabo durante el desarrollo del prototipo para la gestión de edificios inteligentes (SmartBuilding). El proceso comenzó con la investigación y planificación del proyecto, seguido de la ejecución y desarrollo del componente principal. Para cumplir con los objetivos establecidos, se empleó la metodología de desarrollo iterativo e incremental.

El motivo por el cual se recurrió a esta metodología es por su facilidad de crear pequeños sistemas software para irlos escalando conforme el número de iteraciones, entonces al tener un componente basado estrictamente en desarrollo y pruebas, era necesario un modelo que permitiera desarrollar un pequeño sistema mientras se lo vaya probando y aprendiendo de lecciones adquiridas por iteración[6].

Entonces, a diferencia de modelos tradicionales, su primera etapa define los requerimientos básicos e incluso funcionalidades suplementarias que se van a ir mejorando conforme sus iteraciones [7].

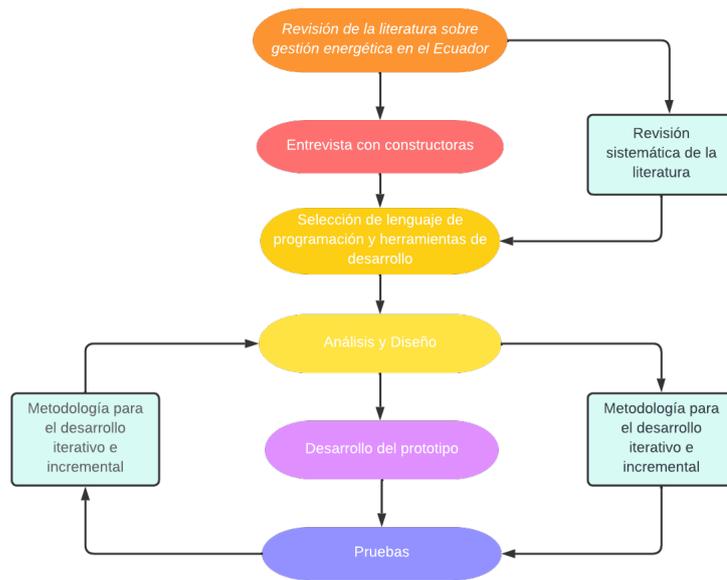


Figura 2.1: Modelo iterativo e incremental

2.1 REVISIÓN DE LA LITERATURA SOBRE GESTIÓN ENERGÉTICA EN ECUADOR

En esta sección se realizó una revisión sistemática de la literatura, con el objetivo de visualizar cuál es el consumo energético dentro del país y cómo calcular los consumos energéticos promedio.

El informe de Balance Energético Nacional del año 2022 reveló que tanto los sectores comerciales como el de transporte usan la electricidad como una de sus principales fuentes de energía. Esto causa una gran dependencia de esta fuente como parte esencial del desarrollo y sostenibilidad a nivel nacional [1].

Entonces, para la realización de este componente, se estableció la pregunta: ¿Cuáles son los indicadores claves de rendimiento que se necesitan? Algo que una persona quiere saber es cuánto está consumiendo, cuánto le está costando su consumo y cuánto podría ahorrar.

Siempre, los indicadores permiten mejorar los sistemas al realizar comparaciones tanto antes como después, con el objetivo de verificar las mejoras que se pueden realizar dentro de una gestión energética [8].

Entonces, en base a las necesidades del país y la necesidad de estrategias para medir el consumo energético, para la interfaz de este proyecto se establecieron los siguientes KPIs.

2.1.1 Establecimiento de KPIs

Para la creación de los KPIs, se debe tener en cuenta tanto en el consumo energético dentro del país como en los costos por su uso en este caso en la ciudad de Quito. En este caso, también se hizo una pequeña comparativa entre los tipos de focos, tanto incandescentes, que son los que normalmente se utilizan en zonas residenciales, como fluorescentes y LEDs, obteniendo los siguientes resultados para un alumbrado necesario de 1600-1800 lúmenes [9].

Basándonos en la revisión de la literatura sobre la gestión energética en el país y la necesidad de herramientas más eficientes para reducir el consumo energético, se plantearon 9 preguntas enfocadas en las necesidades de las empresas constructoras para gestionar el consumo energético y con ello los siguientes KPIs.

| Tipo de Foco | Consumo de Energía (Watts) | Lúmenes Generados |
|--------------------|----------------------------|-------------------|
| Incandescente | 100 | 1600-1800 |
| Fluorescente (CFL) | 23-30 | 1600-1800 |
| LED | 16-20 | 1600-1800 |

Tabla 2.1: Comparativa del Consumo de Energía para Diferentes Tipos de Focos

Basándonos en la revisión de la literatura sobre la gestión energética en el país y la necesidad de herramientas más eficientes para reducir el consumo energético, se plantearon 9 preguntas enfocadas en las necesidades de las empresas constructoras para gestionar el consumo energético y con ello los siguientes KPIs.

2.1.1.1 Consumo energético

Para calcular el consumo energético se debe tener en cuenta la potencia de cada elemento que se va a calcular en este caso Watt, también el tiempo en el que estará prendido el dispositivo junto con los días obteniendo la siguiente fórmula [10].

$$\text{Energía} = (P W) \left(\frac{T \text{ hrs}}{\text{día}} \right) (D \text{ días}) \quad (2.1)$$

Como un ejemplo si se tiene un foco de 100W que estará prendido durante 8 horas al día en un período de 30 días por mes se puede calcular su consumo energético de la siguiente manera.

$$\text{Energía} = (100 W) \left(\frac{8 \text{ hrs}}{\text{día}} \right) (30 \text{ días}) = 24000 Wh = 24 KWh$$

2.1.1.2 Gasto Económico

Este es un concepto de cuanto será el costo por KiloWatt usado. En Ecuador, específicamente en la ciudad de Quito el costo energético es de 9.160 c\$/kWh. Además la formula para calcula el costo energético se mide de la siguiente manera [11].

$$\text{Costo} = (\text{Energía en kWh}) \cdot \left(\frac{\text{Precio en \$}}{\text{kWh}} \right) \quad (2.2)$$

Entonces para el mismo caso si tenemos con el costo de energía que se tiene para el país junto teniendo en cuenta el mismo ejemplo anterior para un foco incandescente de 100W se tendría el siguiente costo.

$$\text{Costo} = (24 \text{ kWh}) \cdot \left(\frac{0.10 \$}{\text{kWh}} \right) = \$2.40$$

Esto nos quiere decir que un foco prendido durante 8 horas al día en un mes consumiría un valor de 2.40 dólares.

2.1.1.3 Ahorro Económico

Se determina un ahorro económico tomando los datos de gasto económico inicial y final con el objetivo de realizar una comparación y ver cuanto es el ahorro económico.

$$\text{Ahorro} = \text{Costo Inicial} - \text{Costo Final} \quad (2.3)$$

Como ejemplo, usando la Tabla 2.1, si en el primer mes se tiene una habitación con un foco incandescente de 100 W que estará prendido 8 horas al día durante 30 días del mes se tendría un consumo de 24 kWh para un costo de \$2.20.

Ahora para el siguiente mes se decide cambiar a un foco LED con la misma cantidad de lámparas donde esta vez tendrá un consumo de 30W encendido ahora por 6 horas durante los 30 días del mes se tendría un consumo de 5.4 kWh con un costo de \$0.50.

Usando la fórmula de ahorro tanto para el mes donde se usó el foco incandescente y el LED se tiene el siguiente resultado.

$$\text{Ahorro} = 2.20 \$ - 0.50 \$ = 1.70 \$$$

Se estaría teniendo un ahorro de \$ 1.70.

2.1.1.4 Ahorro Energético

Se determina un ahorro energético tomando los datos de consumo energético inicial y final con el objetivo de realizar una comparación y ver cuánta energía se está guardando.

$$\text{Ahorro Energético} = \text{Consumo Inicial} - \text{Consumo Final} \quad (2.4)$$

Tomando el ejemplo anterior se tiene el siguiente resultado con un ahorro energético de 18.6 kWh.

$$\text{Ahorro Energético} = 24 \text{ kWh} - 5.4 \text{ kWh} = 18.6 \text{ kWh}$$

2.1.1.5 Emisiones CO2

Para el cálculo de las emisiones de CO2 es necesario entender para qué se está calculando y qué tipo de emisión está tomando protagonismo. En este caso se usan emisiones indirectas de la generación de electricidad y de calor que son las emisiones que derivan del consumo energético [12].

Entonces la fórmula para calcular la emisión de CO2 dentro del consumo energético es la

siguiente.

$$\text{Emisiones de CO}_2 = \text{Consumo (kWh)} \times \text{Factor de Emisión (kg CO}_2\text{/kWh)} \quad (2.5)$$

Ahora en Ecuador se sabe que el factor de emisión es $0.5015 \text{ KgCO}_2\text{/KWh}$. Basandose en el ejemplo anterior para un foco incandescente de 100W con un consumo de 24Kwh por 30 días se tiene el siguiente resultado [13].

$$\text{Emisiones de CO}_2 = 24, \text{kWh} \times 0.5015 \frac{\text{kg CO}_2}{\text{kWh}} = 12.036 \text{ kg CO}_2$$

2.2 ENTREVISTA CON CONSTRUCTORAS

Para este caso fue necesario entender la problemática sobre el consumo energético desde la perspectiva de las empresas constructoras. por lo que se procedió a realizar los siguientes pasos.

2.2.1 Creación de encuesta

En base al establecimiento de KPIs para la gestión energética dentro del país y a la necesidad de tener mejores herramientas que permitan una mayor eficiencia para generar una reducción mínima del consumo energético, se logró plantear 9 preguntas que se enfocan en la necesidad de las empresas constructoras de contar con un sistema capaz de gestionar el consumo energético con ayuda de la implementación de edificios inteligentes, como se muestra en el anexo 6.1.1.

2.2.2 Encuesta a los representantes de constructoras

Dentro del hotel Swissôtel Quito se invitó a las empresas constructoras a un desayuno, donde se dio un espacio para poder entregar las encuestas y que las pudieran llenar de manera manual.

Una vez llenadas las encuestas, los datos fueron añadidos en la herramienta de PowerBI para realizar un análisis gráfico sobre los resultados, llegando a las conclusiones respecti-

vas encontradas en el anexo 6.1.2.

2.2.3 Análisis de resultados

Una vez llenada la encuestas los datos fueron agregados en la herramienta de PowerBI para realizar una análisis gráfico sobre los resultados, donde se llegaron a las conclusiones respectivas encontradas en el anexo 6.1.3 dando paso a la siguiente fase para la creación de una sistema de gestión energética que se enfoque los siguientes puntos.

- Visualización de un Dashboard en tiempo real.
- Capacidad de Gestión de dispositivos.
- Visualización del consumo energético en tiempo real.
- Visualización de las emisiones de Carbono.
- Visualización del ahorro energético respecto al mes anterior.
- Visualización del ahorro económico respecto al mes anterior.

2.3 SELECCIÓN DE LENGUAJE DE PROGRAMACIÓN Y HERRAMIENTAS DE DESARROLLO

2.3.1 Lenguaje de programación

Dentro de un análisis sobre el uso del lenguaje de programación que se usará, se debe entender que todo el proyecto, como aplicación web, se dividirá en dos partes importantes: el BackEnd y el FrontEnd.

Para la sección del BackEnd, lo que se desea es obtener datos y analizarlos para dar los resultados conforme a los KPIs. Existen varios lenguajes de programación capaces de gestionar datos, pero uno de los más populares por su simplicidad, alta demanda, enorme comunidad y facilidad de uso es Python.

Python, al ser un lenguaje de programación, cumple con todos los requerimientos como alternativa a MATLAB o R, teniendo algunas ventajas sobre ellos, como la facilidad de lectura

de código, visualización de resultados de manera eficiente y ejecución de comandos. Pero, sobre todo, su capacidad de soportar programación orientada a objetos y su utilidad para el desarrollo web[14].

Para la sección del FrontEnd como es lo habitual se decidió usar el lenguaje de programación de JavaScript ya que este tiene una gran capacidad de realizar páginas web de manera dinámica permitiendo una comunicación y ejecución directa desde el navegador. Este lenguaje es ampliamente completo, ya que en este permite hacer programación FullStack. En el caso de FrontEnd este tiene una gran apego tanto con HTML y CSS permitiendo desarrollar un diseño con ayuda de Frameworks como React, Angular o Vue dando así mayor comodidad a la hora del desarrollo FrontEnd [15].

2.3.2 Herramientas de desarrollo

2.3.2.1 Figma

Figma es una herramienta de diseño colaborativo para crear prototipos y mockups completamente online. En este caso para el desarrollo de FrontEnd ayuda a crear un diseño MockUp para saber como de ser la funcionalidad del desarrollo, una de las ventajas de figma es la capacidad colaborativa, quiere decir que entre varios usuarios pueden colaborar en la creación del diseño para una página web por lo que es muy buena opción en la fase de diseño [16].

2.3.2.2 Bootstrap

Como se habló anteriormente al decidir usar un lenguaje de programación web como JavaScript es necesario facilitar y mejorar la eficiencia en desarrollo del mismo, por lo que JavaScript ofrece Frameworks para lograr una mejor comodidad a la hora de desarrollar.

Bootstrap es un framework que tiene gran parte ventajas sobre React como Angular que también son muy populares. Estas ventajas permiten a Bootstrap ser una mejor opción a la hora de programar [17].

- ❑ **Facilidad de aprendizaje:** Bootstrap tiene una curva de aprendizaje baja por lo que es mucho más rápido de aprender.

- ❑ **Compatible con los navegadores:** Bootstrap tiene una compatibilidad amplia con la mayoría de los navegadores como Chrome, Safari, Mozilla Firefox, etc.

- ❑ **Comunidad y soporte:** Bootstrap cuenta con una gran comunidad y soporte permitiendo siempre encontrar documentación y videos sobre como usar cada uno de sus componentes.

- ❑ **Flexibilidad y modularidad:** Vue.js usa un sistema de componentes, estos al ser llamados desde uno principal, creando un proyecto escalable y adaptable a cambios.

Pero una de las mas importantes es que permite una excelente integración con Flask, que este fue desarrollado en el componente del Backend, entonces tener con python una buena manipulación de datos, tener integración con bases de datos junto con herramientas de procesamiento de datos y a la misma vez una interfaz llamativa y escalable permite una buena combinación para manejar Bootstrap y Flask en el desarrollo web para Java Script y Python [18].

2.4 ANÁLISIS Y DISEÑO

En esta fase de análisis y diseño se deben establecer las funcionalidades que debe tener el sistema para lograr tener una idea clara y poder empezar con su desarrollo. En esta fase se realizaron las historias de usuario basándose en las necesidades del sistema [19].

2.4.1 Historias de Usuario

Las historias de usuario son herramientas para agilizar las descripciones de las funcionalidades de los sistemas. Esto ayuda a disminuir una comunicación formal, evitando una gran cantidad de documentación. En ellas se establece lo que el cliente quiere transmitir para su programa y cómo se validaría. Para el desarrollo FrontEnd se tienen las siguientes historias de usuario [20].

| Historia de Usuario | |
|---------------------------------|---|
| Número: | 1 |
| Usuario: | Cliente |
| Nombre historia: | Navegación por DashBoard |
| Prioridad en negocio: | Alta |
| Riesgo en desarrollo: | Medio |
| Iteración asignada: | 1 |
| Programador responsable: | Willian Rivera |
| Descripción: | El usuario debe poder navegar en el dashboard de manera libre, teniendo la oportunidad de visualizar todos los KPIs establecidos. |
| Validación: | El cliente al dar clic en cualquier módulo del dashboard podrá ingresar en una ventana aparte todos los componentes del módulo establecido. |

Tabla 2.2: Historia de Usuario (Navegación por DashBoard)

| Historia de Usuario | |
|---------------------------------|--|
| Número: | 2 |
| Usuario: | Cliente |
| Nombre historia: | Visualizar y gestionar dispositivos |
| Prioridad en negocio: | Alta |
| Riesgo en desarrollo: | Alto |
| Iteración asignada: | 1 |
| Programador responsable: | Willian Rivera |
| Descripción: | El usuario debe poder observar todos los dispositivos conectados y a su vez poder gestionarlos como el poder apagarlos y prenderlos. |
| Validación: | Poder observar los dispositivos conectados desde el BackEnd y su estado, con eso poder decidir entre el el estado del dispositivo |

Tabla 2.3: Historia de Usuario (Visualizar y gestionar dispositivos)

| Historia de Usuario | |
|---------------------------------|--|
| Número: | 3 |
| Usuario: | Cliente |
| Nombre historia: | Visualizar el consumo energético |
| Prioridad en negocio: | Alta |
| Riesgo en desarrollo: | Alto |
| Iteración asignada: | 1 |
| Programador responsable: | Willian Rivera |
| Descripción: | El usuario debe poder ver en un gráfico su consumo energético mediante la página al entrar en su respectivo módulo |
| Validación: | Al dar clic en " <i>Consumption</i> " el usuario deberá poder ver una gráfica con los datos tomados del backend para su consumo energético |

Tabla 2.4: Historia de Usuario (Visualizar el consumo energético)

| Historia de Usuario | |
|---------------------------------|--|
| Número: | 4 |
| Usuario: | Cliente |
| Nombre historia: | Visualizar el ahorro energético |
| Prioridad en negocio: | Alta |
| Riesgo en desarrollo: | Alto |
| Iteración asignada: | 1 |
| Programador responsable: | Willian Rivera |
| Descripción: | El usuario debe poder ver en un gráfico su ahorro energético mediante la página al entrar en su respectivo módulo |
| Validación: | Al dar clic en " <i>Energy Savings</i> " el usuario deberá poder ver una gráfica con los datos tomados del backend para su ahorro energético |

Tabla 2.5: Historia de Usuario (Visualizar el ahorro energético)

| Historia de Usuario | |
|---------------------------------|--|
| Número: | 5 |
| Usuario: | Cliente |
| Nombre historia: | Visualizar el ahorro de CO2 |
| Prioridad en negocio: | Alta |
| Riesgo en desarrollo: | Alto |
| Iteración asignada: | 1 |
| Programador responsable: | Willian Rivera |
| Descripción: | El usuario debe poder ver en un gráfico el consumo de CO2 emitido por uso de energía |
| Validación: | Al dar clic en “ <i>Emissions CO2</i> ” el usuario deberá poder ver una gráfica con los datos tomados del backend para la emisión de CO2 producida así como su ahorro respecto a dicho mes |

Tabla 2.6: Historia de Usuario (Visualizar el ahorro de CO2)

| Historia de Usuario | |
|---------------------------------|---|
| Número: | 6 |
| Usuario: | Cliente |
| Nombre historia: | Visualizar ahorro económico |
| Prioridad en negocio: | Alta |
| Riesgo en desarrollo: | Alto |
| Iteración asignada: | 1 |
| Programador responsable: | Willian Rivera |
| Descripción: | El usuario debe poder ver en un gráfico el ahorro económico por uso de energía |
| Validación: | Al dar clic en “ <i>Cost Savings</i> ” el usuario deberá poder ver una gráfica con los datos tomados del backend para su ahorro económico |

Tabla 2.7: Historia de Usuario (Visualizar ahorro económico)

| Historia de Usuario | |
|---------------------------------|--|
| Número: | 7 |
| Usuario: | Cliente |
| Nombre historia: | Ingreso de credenciales |
| Prioridad en negocio: | Alta |
| Riesgo en desarrollo: | Alto |
| Iteración asignada: | 1 |
| Programador responsable: | Willian Rivera |
| Descripción: | El cliente podrá observar una ventana de login donde colocará sus credenciales y podrá acceder a su dashboard |
| Validación: | Al ingresar a la página, el cliente podrá colocar sus credenciales en caso de ser verdadera, ingresará a la página en caso de no se enviará un mensaje de error. |

Tabla 2.8: Historia de Usuario (Ingreso de credenciales)

| Historia de Usuario | |
|---------------------------------|---|
| Número: | 8 |
| Usuario: | Cliente |
| Nombre historia: | Registro de usuario |
| Prioridad en negocio: | Alta |
| Riesgo en desarrollo: | Alto |
| Iteración asignada: | 1 |
| Programador responsable: | Willian Rivera |
| Descripción: | El cliente podrá registrarse en caso de no tener credenciales |
| Validación: | Al dar clic en registro el cliente ingresará su correo y contraseña, tales que se guardarán en la base de datos para el momento de ingresar sus credenciales. |

Tabla 2.9: Historia de Usuario (Registro de usuario)

2.4.2 MockUp en Figma

La parte del Frontend en el desarrollo del componente se enfoca en lo que la interfaz de usuario, es decir lo que el usuario va a observar al momento de navegar por el sistema,

entonces siempre es necesario la creación de un diseño para tener una idea de como se va a ver el sistema, un mockup es el bosquejo de la interfaz de usuario que permite hacer una representación de como lucirá al momento del desarrollo [21].

Entonces para el desarrollo de este bosquejo se utilizó la herramienta de Figma por su facilidad de uso.

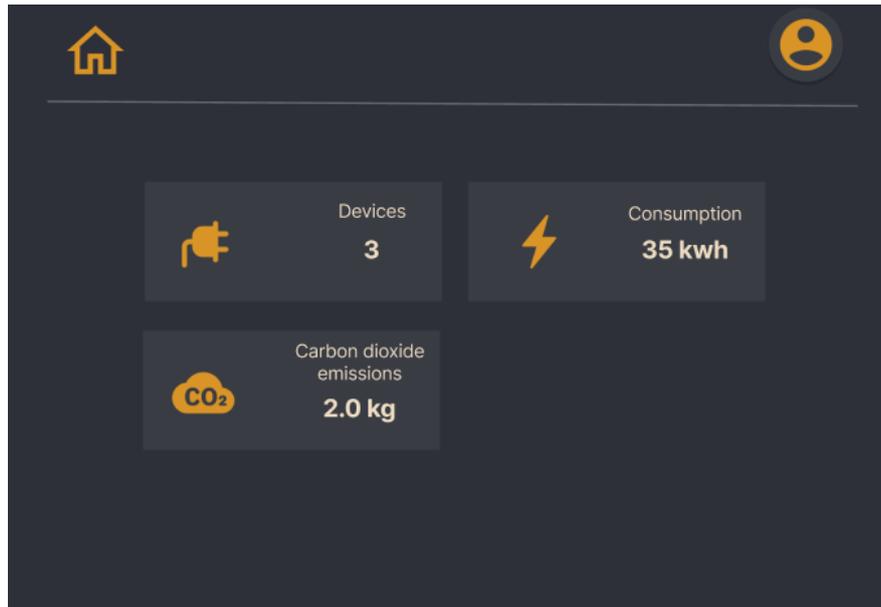


Figura 2.2: Bosquejo para sección DashBoard

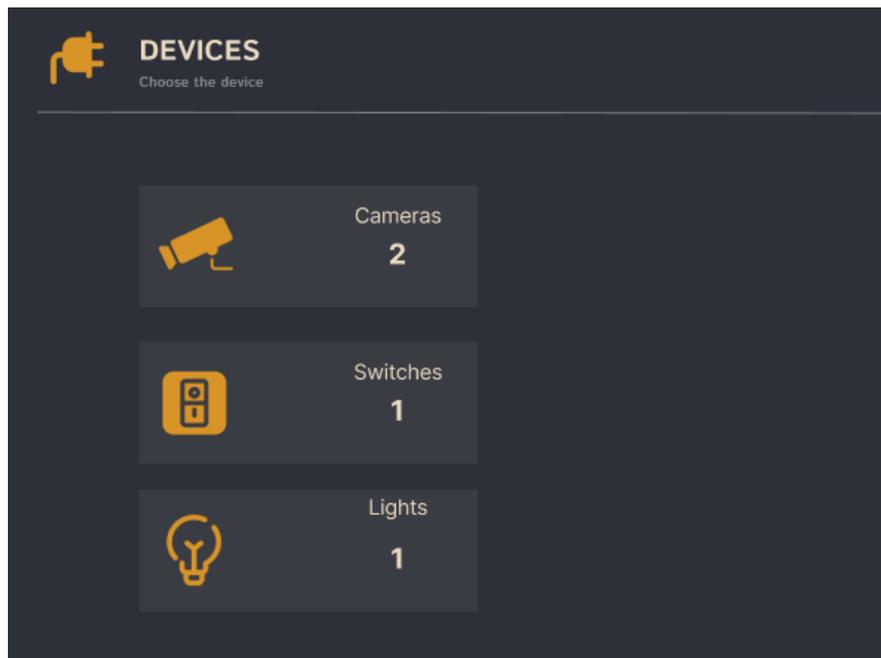


Figura 2.3: Bosquejo para el módulo "Devices"

2.5 DESARROLLO DEL PROTOTIPO

En esta parte se realizó el prototipo. Lo más importante es la configuración y el llamado de datos desde el backend creado en Flask, por lo que se empezó a programar por partes, enfocándose en la parte HTML. Cabe resaltar que en esta etapa se consideran las primeras iteraciones del prototipo. En la sección de resultados se presentará la iteración final.

2.5.1 Login

Como se visualizó en la historia de usuario 6, el cliente debe poder obtener sus credenciales, por lo que se propuso el siguiente código, capaz de llamar a los datos del backend que, a su vez, se guardan en su base de datos.

login.html

```
1 <div class="wrapper fadeInDown">
-   <div id="formContent">
-       <form action="{{ url_for('login') }}" method="POST">
-           <input type="email" id="login" class="fadeIn second" name="email"
-               placeholder="email" autocomplete="email">
5           <input type="password" id="password" class="fadeIn third" name="
-               password" placeholder="password" autocomplete="current-password">
-           <input type="submit" class="fadeIn fourth" value="Log In">
-       </form>
-       <form action="{{ url_for('register') }}" method="GET">
-           <button type="submit">Register</button>
10      </form>
-      {% with messages = get_flashed_messages() %}
-          {% if messages %}
-              <ul class=flashes>
-                  {% for message in messages %}
-                      <li>{{ message }}</li>
15                      {% endfor %}
-                  </ul>
-              {% endif %}
-          {% endwith %}
```

2.5.2 Registro

Como se estableció en la historia de usuario 7 el cliente deberá poder crear un usuario, donde se guardará en la base de datos directamente desde el backend haciendo la llamada del mismo como se muestra en el siguiente bloque de código.

registration.html

```

1 <div id="formContent">
-   <h1 class="fadeIn">Register </h1>
-   <form action="/register" method="post" id="registration-form">
-       <input type="email" id="email" name="email" class="fadeIn second"
-           placeholder="Email" autocomplete="email" required>
5       <input type="password" id="password" name="password" class="fadeIn
-           third" placeholder="Password" autocomplete="new-password"
-           required>
-       <input type="password" id="confirm_password" name="confirm_password"
-           class="fadeIn fourth" placeholder="Confirm Password" autocomplete
-           ="new-password" required>
-       <div class="error">{{ error }} </div>
-       <button type="submit" class="fadeIn">Register </button>
-   </form>
10 </div>

```

2.5.3 Dashboard

En la sección del Dashboard, tal como se pidió en la historia de usuario 1, debe haber una interfaz donde se pueda ingresar a cada módulo. Entonces, se decidió hacer una lista de módulos que hagan referencia a los diferentes templates, para poder navegar sobre ellos, como se muestra en el siguiente bloque de código.

Lo importante en la programación del FrontEnd fue la capacidad de realizar cálculos, por

ejemplo, de la cantidad de dispositivos, consumo total, emisiones de carbono, ahorro energético y ahorro económico, todo con cálculos realizados desde el backend, como se muestra en el siguiente bloque de código.

dashboard.html

```
1 <div class="col-md-4">
-   <a href="/devices" class="info-panel">
-       
-       <div class="info-text">
5           <h3>{{ total_devices }}</h3>
-           <p>3</p>
-       </div>
-   </a>
- </div>
10 <div class="col-md-4">
-   <a href="/consumption" class="info-panel">
-       
-       <div class="info-text">
-           <h3>Consumo</h3>
15           <p>{{ total_energy }} kWh</p>
-       </div>
-   </a>
- </div>
- <div class="col-md-4">
20   <a href="/emissions" class="info-panel">
-       
-       <div class="info-text">
-           <h3>Emisiones de Carbono</h3>
-           <p>{{ total_emissions }} KgCo2</p>
25       </div>
-   </a>
- </div>
```

2.5.4 Módulo Devices

En este módulo se nos permitirá visualizar todos los dispositivos directamente desde el backend, el cual toma los dispositivos desde la base de datos y permite ver su estado en tiempo real. Por otro lado, el Frontend permite la visualización y cambio del estado. Cabe añadir que, dentro de la base de datos, se puede visualizar el tipo de dispositivo. Entonces, en base al tipo de dispositivo, el sistema coloca un ícono respectivo, con el fin de mejorar visualmente los dispositivos, permitiendo una comodidad y mejor entendimiento para el usuario, como se muestra en el siguiente bloque de código.

devices.html

```
1 <div id="device-list" class="list-unstyled w-100">
-   {% for device in devices_db %}
-     <div class="device-card">
-       <div class="device-info">
5         
-         <span class="device-name">{{ device.customName }}</
-           span>
-       </div>
-       <label class="switch">
-         <input type="checkbox" class="device-toggle" data-
10         device-id="{{ device.id }}"
-           {% if device.status[0].value %}>checked{% endif
-             %}>
-         <span class="slider"></span>
-       </label>
-     </div>
-   {% endfor %}
15 </div>
```

Como pueden ver este template llama a un archivo main.js que es el que se encargará de la lógica para actualizar el estado de un dispositivo directamente desde la base de datos, permitiendo la interacción tanto de la base de datos y el backend.

main.js

```
1 $(document).ready(function () {
-   var socket = io();
-
-   socket.on('updated_devices', function (data) {
5     var deviceList = $("#device-list");
-     deviceList.empty();
-     data.devices_db.forEach(function (device) {
-       var checked = device.status[0].value ? 'checked' : '';
-       var listItem = $("<div class='device-card'><div class='device-info'>"
+ device.customName + " (" + device.id + ")</div><label class='
switch'><input type='checkbox' class='device-toggle' data-device-
id='" + device.id + "' " + checked + "><span class='slider'></
span></label></div>");
10      deviceList.append(listItem);
-    });
-  });
-
-  function debounce(func, wait) {
15    let timeout;
-    return function () {
-      const context = this, args = arguments;
-      clearTimeout(timeout);
-      timeout = setTimeout(() => func.apply(context, args), wait);
20    };
-  }
-
-  function sendUpdateRequest(deviceId, isActive) {
-    $.post("/update_device_status", {device_id: deviceId, new_status:
isActive}, function (data) {
25      if (!data.success) {
-        alert('Error updating device status. Please try again.');
```

socket.emit('request_updated_devices');

```
    }
-    $('<div class='device-toggle'>').prop('disabled', false);
30  }).fail(function () {
-    alert('Error sending update request. Please try again.');
```

socket.emit('request_updated_devices');

```
    $('<div class='device-toggle'>').prop('disabled', false);
```

```

-     });
35  }
-
-     const debouncedSendUpdateRequest = debounce(sendUpdateRequest, 300);
-
-     $(document).on('change', '.device-toggle', function () {
40     var deviceId = $(this).data('device-id');
-     var isActive = $(this).is(':checked');
-     $(this).prop('disabled', true);
-     debouncedSendUpdateRequest(deviceId, isActive);
-     });
45
-     socket.emit('request_updated_devices');
- });

```

2.5.5 Módulo Energy Consumption

Este módulo permite visualizar el consumo energético por mes. Toma las horas en las que los dispositivos estuvieron encendidos durante el mes actual y, utilizando chart.js para llamar a los cálculos realizados, muestra en un gráfico de barras una tabla de consumo por día, donde el eje horizontal representa los días y el eje vertical el consumo en kWh. Todos estos datos se llaman desde el backend, como se muestra en el siguiente bloque de código.

consumption.html

```

1  const devices = {{ devices_info|tojson }};
-     const labels = devices.length > 0 ? devices[0].time.map((_, index) =>
-         'Day ${index + 1}') : [];
-
-     const totalConsumptionPerDay = labels.map((_, dayIndex) => {
5     return devices.reduce((total, device) => {
-         return total + device.energy_kwh[dayIndex];
-     }, 0);
-     });
-

```

```

10  const ctxConsumption = document.getElementById('consumptionChart').
    getContext('2d');
-   const consumptionChart = new Chart(ctxConsumption, {
-     type: 'bar',
-     data: {
-       labels: labels,
15     datasets: [{
-       label: 'Consumo Total (kWh)',
-       data: totalConsumptionPerDay,
-       backgroundColor: 'rgba(255, 193, 7, 0.2)',
-       borderColor: 'rgba(255, 193, 7, 1)',
20       borderWidth: 1
-     }]
  }

```

2.5.6 Módulo CO2 Emissions

En este módulo permite, visualizar las emsiones de carbono por día como usando el factor de emisión dentro del Ecuador.

Cada dispositivo emite carbono, por lo que el sistema toma el consumo energético diario y lo multiplica por el factor de emisión, mostrando en un análisis lineal las emisiones de carbono durante el mes, igualmente usando chart.js, como se muestra en el siguiente bloque de código.

emissions.html

```

1  const totalCo2EmissionsPerDay = {{ total_co2_emissions_per_day|tojson }};
-   const labels = totalCo2EmissionsPerDay.map((_, index) => 'Day ${index
-     + 1}');
-
-   const ctxEmissions = document.getElementById('emissionsChart').
    getContext('2d');
5  const emissionsChart = new Chart(ctxEmissions, {
-     type: 'line',
-     data: {
-       labels: labels,
-       datasets: [{

```

```

10         label: 'Total CO2 Emissions (kg CO2)',
-         data: totalCo2EmissionsPerDay,
-         backgroundColor: 'rgba(255, 193, 7, 0.2)',
-         borderColor: 'rgba(255, 193, 7, 1)',
-         borderWidth: 2,
15         fill: true
-     }}
- },

```

2.5.7 Módulo Energy Savings

Este módulo permite visualizar el ahorro energético, tomando en cuenta el consumo entre dos meses. Se toman las horas consumidas por día durante dos meses y, en base al cálculo del consumo energético, se realiza una suma total de los datos obtenidos. Después, como se mostró previamente, se realiza una resta entre el consumo inicial y el consumo final. Si el valor es positivo, significa que hubo un ahorro significativo; de lo contrario, hubo un aumento en el consumo.

Para este módulo, igualmente se usó chart.js, donde, con una gráfica de pasteles, se toman los datos desde el backend para mostrarlos en la interfaz, como se muestra en el siguiente bloque de código.

energy_savings.html

```

1  const energyData = {{ total_energy_by_month|tojson }};
-     const labels = Object.keys(energyData);
-     const data = Object.values(energyData);
-
5     const ctxEnergySavings = document.getElementById('energySavingsChart')
-         .getContext('2d');
-     const energySavingsChart = new Chart(ctxEnergySavings, {
-         type: 'pie',
-         data: {
-             labels: labels,
10         datasets: [{
-             label: 'Energy Consumption (kWh)',

```

```

-         data: data,
-         backgroundColor: [
-             'rgba(255, 193, 7, 0.2)', // Light Yellow
15             'rgba(128, 0, 128, 0.2)' // Light Purple
-         ],
-         borderColor: [
-             'rgba(255, 193, 7, 1)', // Yellow
-             'rgba(128, 0, 128, 1)' // Purple
20         ],
-         borderWidth: 1
-     }
-     }}

```

2.5.8 Módulo Cost Savings

Este módulo permite visualizar el ahorro económico entre los dos últimos meses. Como se observó en la fórmula de gasto económico, en base al consumo energético por día se calcularon los gastos en Ecuador, para después realizar una suma total y al igual que el ahorro energético obtener la diferencia.

Para este módulo de igual manera se usó chart.js para obtener una gráfica de pastel donde compara los gastos económicos de los dos últimos meses como se muestra en el siguiente bloque de código.

economic_savings.html

```

1  const costData = {{ total_cost_by_month | tojson }};
-     const labels = Object.keys(costData);
-     const data = Object.values(costData);
-
5     const ctxEconomicSavings = document.getElementById('
-         economicSavingsChart').getContext('2d');
-     const economicSavingsChart = new Chart(ctxEconomicSavings, {
-         type: 'pie',
-         data: {
-             labels: labels,
-             datasets: [{
10

```

```

-         label: 'Economic Cost ($)',
-         data: data,
-         backgroundColor: [
-             'rgba(255, 193, 7, 0.2)', // Light Yellow
15         'rgba(128, 0, 128, 0.2)' // Light Purple
-         ],
-         borderColor: [
-             'rgba(255, 193, 7, 1)', // Yellow
-             'rgba(128, 0, 128, 1)' // Purple
20         ],
-         borderWidth: 1
-     }}

```

Para una mejor visualización del código se puede acceder en el anexo 6.2.1.

2.6 PRUEBAS DEL PROTOTIPO

2.6.1 Pruebas Estáticas

Para este caso como se había mencionado se usó la herramienta Pylint que nos permitió ver errores dentro del código, tanto errores como de variables que no son correctamente utilizadas, errores de indentación.

Para este caso se usó la herramienta para los dos archivos Python, los cuales fueron `app.py` y `app_model.py`.

`app.py`

| Archivo y Línea | Mensaje de Error |
|-----------------|---|
| app.py:1:0 | Missing module docstring (missing-module-docstring) |
| app.py:8:0 | Line too long (102/100) (line-too-long) |
| app.py:11:0 | Line too long (263/100) (line-too-long) |
| app.py:30:39 | Consider using <code>with</code> for resource-allocating operations (consider-using-with) |

Tabla 2.10: Errores detectados en el archivo `app.py`

| Archivo y Línea | Mensaje de Error |
|-----------------|--|
| app.py:30:39 | Using open without explicitly specifying an encoding (unspecified-encoding) |
| app.py:36:0 | Missing class docstring (missing-class-docstring) |
| app.py:43:4 | Attribute <code>id</code> defined outside <code>__init__</code> (attribute-defined-outside-init) |
| app.py:48:0 | Missing function or method docstring (missing-function-docstring) |
| app.py:61:15 | Catching too general exception <code>Exception</code> (broad-exception-caught) |
| app.py:124:0 | Missing function or method docstring (missing-function-docstring) |
| app.py:126:27 | Redefining name <code>energy_savings</code> from outer scope (line 124) (redefined-outer-name) |
| app.py:136:25 | Redefining name <code>economic_savings</code> from outer scope (line 134) (redefined-outer-name) |
| app.py:189:8 | Unnecessary <code>else</code> after <code>return</code> , remove the <code>else</code> and de-indent the code inside it (no-else-return) |
| app.py:198:8 | Use lazy <code>%</code> formatting in logging functions (logging-fstring-interpolation) |
| app.py:202:0 | Missing function or method docstring (missing-function-docstring) |
| app.py:202:21 | Unused argument <code>docs</code> (unused-argument) |
| app.py:202:36 | Unused argument <code>read_time</code> (unused-argument) |

Tabla 2.10: Errores detectados en el archivo `app.py`

Para dar una idea de los errores capturados por ejemplo en la línea 8 se indica que es una línea de código demasiado larga y eso no es una buena práctica de programación por lo que se realizó los cambios para las futuras iteraciones.

`app_model.py`

| Archivo y Línea | Mensaje de Error |
|---------------------|---|
| app_model.py:1:0 | Missing module docstring (missing-module-docstring) |
| app_model.py:101:0 | Line too long (101/100) (line-too-long) |
| app_model.py:135:0 | Line too long (118/100) (line-too-long) |
| app_model.py:145:0 | Line too long (102/100) (line-too-long) |
| app_model.py:168:0 | Line too long (101/100) (line-too-long) |
| app_model.py:178:0 | Line too long (101/100) (line-too-long) |
| app_model.py:271:0 | Line too long (102/100) (line-too-long) |
| app_model.py:287:0 | Trailing whitespace (trailing-whitespace) |
| app_model.py:295:0 | Line too long (109/100) (line-too-long) |
| app_model.py:309:0 | Line too long (121/100) (line-too-long) |
| app_model.py:466:0 | Final newline missing (missing-final-newline) |
| app_model.py:30:0 | Missing class docstring (missing-class-docstring) |
| app_model.py:90:8 | Attribute <code>id</code> defined outside <code>__init__</code> (attribute-defined-outside-init) |
| app_model.py:33:0 | Missing function or method docstring (missing-function-docstring) |
| app_model.py:50:11 | Catching too general exception <code>Exception</code> (broad-exception-caught) |
| app_model.py:60:4 | Redefining name <code>auth</code> from outer scope (line 6) (redefined-outer-name) |
| app_model.py:74:0 | Missing function or method docstring (missing-function-docstring) |
| app_model.py:142:0 | Missing function or method docstring (missing-function-docstring) |
| app_model.py:181:12 | Use lazy % formatting in logging functions (logging-fstring-interpolation) |
| app_model.py:188:8 | Unnecessary <code>else</code> after <code>return</code> , remove the <code>else</code> and de-indent the code inside it (no-else-return) |
| app_model.py:191:12 | Use lazy % formatting in logging functions (logging-fstring-interpolation) |
| app_model.py:333:12 | Redefining built-in <code>type</code> (redefined-builtin) |
| app_model.py:391:0 | Dangerous default value <code>[]</code> as argument (dangerous-default-value) |

Tabla 2.11: Errores detectados en el archivo `app_model.py`

2.6.2 Pruebas Dinámicas

Dentro de las pruebas dinámicas se decidió optar por pruebas de caja negra. Entre estas pruebas, una de las más importantes son las pruebas de usabilidad.

2.6.2.1 Pruebas SUS para usabilidad del prototipo

En esta fase de pruebas se decidió acudir a las instalaciones de la Facultad de Ingeniería en Sistemas con un grupo de voluntarios para que probaran el prototipo funcional, integrando el BackEnd, FrontEnd y la simulación. Una vez realizadas las pruebas, se decidió hacer un cuestionario SUS sobre la usabilidad del prototipo, donde se encuestó a 10 personas, como se muestra en la tabla 2.12.

| Persona | Nombre |
|------------|-----------------------------|
| Persona 1 | Roberth Gancino |
| Persona 2 | José Armando Sarango Cuenca |
| Persona 3 | Danny Alexander |
| Persona 4 | Jose Luis |
| Persona 5 | Elliot Cazar |
| Persona 6 | Cristian Díaz |
| Persona 7 | Brayan Ortiz |
| Persona 8 | Anthony Goyes |
| Persona 9 | Ozzy James Loachamín M. |
| Persona 10 | Fabián Simbaña |
| Persona 11 | Byron Carpio |

Tabla 2.12: Lista de Encuestados

Estas 10 personas estuvieron atentos al funcionamiento del sistema como se muestra en el anexo 6.3.1 además de haber llenado el formulario que se encuentra en el anexo 6.3.2 obteniendo la siguiente información.

| Pregunta | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | P11 |
|---|----|----|----|----|----|----|----|----|----|-----|-----|
| Creo que me gustaría acceder a esta plataforma de Gestión energética frecuentemente | 4 | 4 | 4 | 4 | 3 | 3 | 3 | 4 | 4 | 3 | 4 |
| Encontré que la navegación y análisis energético se encuentra de manera sencilla | 4 | 4 | 3 | 4 | 3 | 4 | 4 | 4 | 4 | 3 | 3 |
| Encontré que el uso y visualización de los módulos son muy fáciles de usar | 4 | 4 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 3 | 3 |
| Creo que puedo usar la interfaz sin necesidad de algún tutorial | 4 | 4 | 4 | 3 | 3 | 3 | 4 | 4 | 3 | 4 | 3 |
| Los módulos y secciones web me parecen bien organizadas | 4 | 3 | 4 | 3 | 3 | 3 | 4 | 4 | 4 | 3 | 3 |
| Me parece que el sistema hace un buen trabajo en presentar la información de manera clara y concisa | 4 | 4 | 4 | 4 | 3 | 4 | 3 | 4 | 4 | 3 | 4 |
| Imagino que las personas podrán aprender a manipular el sistema de manera rápida | 4 | 4 | 3 | 4 | 2 | 3 | 3 | 3 | 4 | 4 | 4 |
| Me sentí muy seguro en navegar por el sistema | 4 | 4 | 3 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 |
| No siento que tenga que pasar tiempo aprendiendo a usar el sistema o programa | 4 | 4 | 3 | 4 | 3 | 3 | 4 | 4 | 4 | 3 | 3 |
| Me sentí muy confiado usando el sistema | 4 | 4 | 4 | 3 | 3 | 4 | 4 | 4 | 4 | 3 | 3 |

Tabla 2.13: Resultados de Encuestados

En base a estos datos se realizaron los respectivos cálculos.

| Persona | Resultados | Resultados /10 |
|---------|------------|----------------|
| P1 | 40 | 100 |

Tabla 2.14: Puntuación SUS

| Persona | Resultados | Resultados /10 |
|---------|------------|----------------|
| P2 | 39 | 97.5 |
| P3 | 35 | 87.5 |
| P4 | 35 | 87.5 |
| P5 | 29 | 72.5 |
| P6 | 33 | 82.5 |
| P7 | 36 | 90 |
| P8 | 39 | 97.5 |
| P9 | 39 | 97.5 |
| P10 | 33 | 82.5 |
| P11 | 34 | 85 |

Tabla 2.14: Puntuación SUS

Para las siguientes pruebas se las observará en el reporte realizado dentro del anexo 6.3.3. En base a estas pruebas se logró establecer nuevas iteraciones con el objetivo de obtener mejoras en la calidad de desarrollo.

3 RESULTADOS Y DISCUSIÓN

Una vez realizado el Dashboard en base a todas las iteraciones, las cuales fueron resumidas, y en base a los cambios realizados, se planteó el siguiente caso:

Durante los meses de junio y julio hubo 3 dispositivos inteligentes con la capacidad de brindar información tanto de sus modelos físicos como de su potencia y tiempo encendido. Todos estos datos se reflejan en la siguiente tabla.

| Id Dispositivo | Type | Watts |
|------------------------|--------|-------|
| 03200309dc4f2219bc50 | pc | 600 |
| eb4f5e6ca883ad7479mawe | light | 9 |
| vdevo170000581142241 | switch | 100 |

Tabla 3.1: Tabla de Dispositivos y Consumo de Energía

Cada dispositivo estuvo encendido durante un periodo de tiempo diario tomando 2 meses como se muestra en el siguiente Json.

```
1 {
-   "Device ID": "03200309dc4f2219bc50",
-   "Watts": 600,
-   "Time Array": {
5     "Junio": [8, 10, 11, 11, 12, 12, 12, 7, 15, 13, 9, 24, 12, 12, 12, 12, 12,
        12, 8, 12, 12, 9, 12, 12, 12, 12, 12, 13, 8, 13],
-     "Julio": [8, 1, 8, 8, 8, 2, 8, 7, 6, 9, 9, 9, 8, 4, 9, 4, 9, 9, 4, 9, 5, 9,
        4, 8, 8, 4, 8, 8, 9, 3, 8]
-   }
- }
- {
10  "Device ID": "eb4f5e6ca883ad7479mawe",
-   "Watts": 9,
-   "Time Array": {
-     "Junio": [15, 13, 15, 24, 15, 15, 15, 15, 16, 11, 13, 13, 13, 14, 8, 9, 8, 9,
```

```

13, 13, 13, 15, 15, 15, 15, 15, 8, 8, 8, 9, 9],
- "Julio": [8, 1, 8, 7, 7, 7, 7, 7, 8, 8, 9, 8, 7, 7, 7, 8, 8, 7, 8, 7, 8, 7,
15 }
- }
- {
- "Device ID": "vdevo170000581142241",
- "Watts": 100,
20 "Time Array": {
- "Junio": [8, 8, 15, 15, 15, 20, 24, 15, 10, 8, 15, 15, 15, 11, 11, 11, 11,
- 13, 13, 13, 9, 9, 9, 9, 9, 9, 9, 9, 8],
- "Julio": [5, 8, 6, 9, 4, 5, 10, 10, 7, 10, 8, 10, 7, 10, 10, 8, 10, 8, 9, 8,
- 10, 8, 7, 9, 10, 8, 7, 8, 8, 7, 5]
- }
- }

```

3.1 SECCIÓN LOGIN

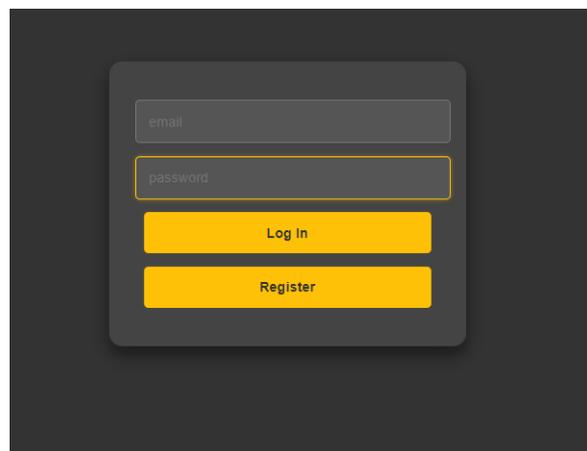


Figura 3.1: Ventana de login

3.2 SECCIÓN REGISTER

En esta pestaña, el usuario podrá registrarse colocando como usuario su correo electrónico y su contraseña, teniendo que confirmarla, como se muestra en la siguiente figura.

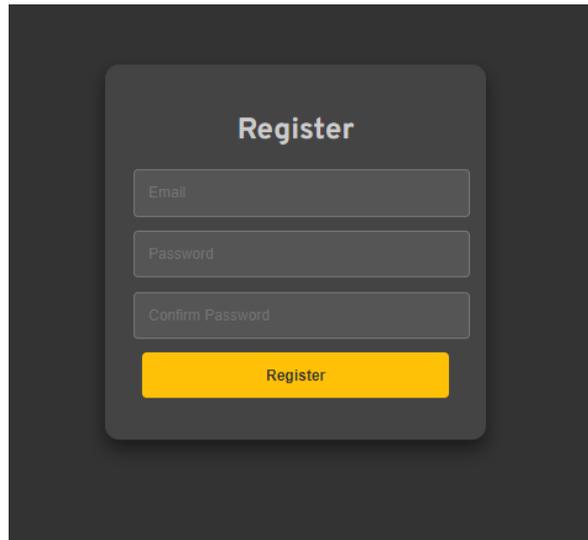


Figura 3.2: Pestaña de register

3.3 SECCIÓN DASHBOARD

En esta sección, el usuario podrá visualizar la ventana de dashboard con los datos tomados en tiempo real. Como se planteó previamente, el sistema toma los datos de potencia y tiempos para realizar los cálculos establecidos y poder mostrarlos en tiempo real, como se muestra en la siguiente figura.

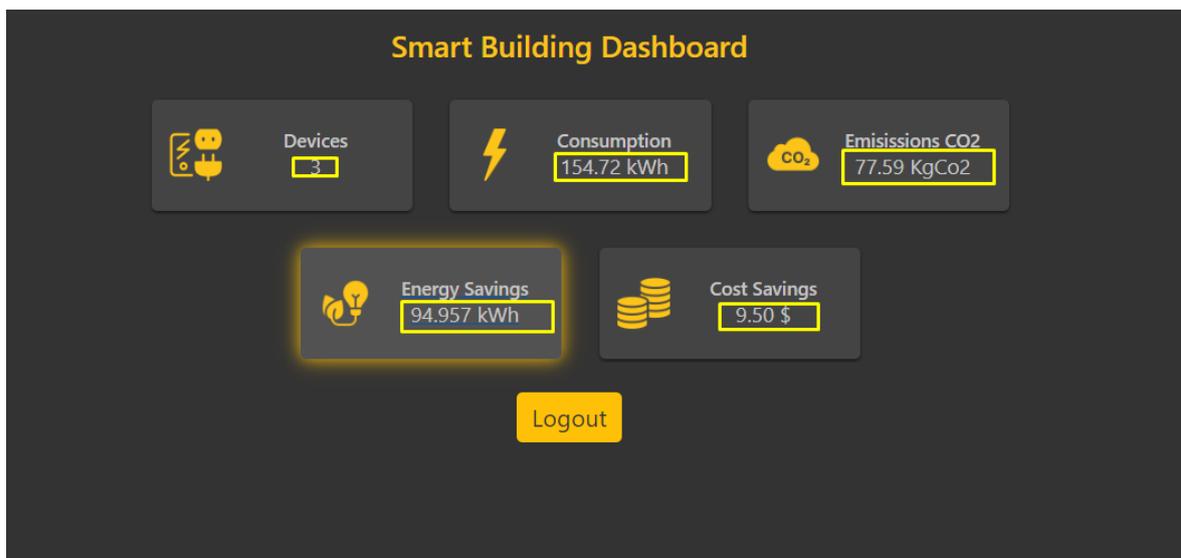


Figura 3.3: Ventana DashBoard

Dentro de esta sección se divide en 5 módulos que muestran diferentes datos referentes a los dispositivos conectados, los cuales se verán a continuación.

3.3.1 Módulo Devices

El módulo *Devices* permitió visualizar los dispositivos conectados conjuntamente con sus estados, permitiendo la interacción directa desde la simulación. La variable *"Type"* permite verificar el tipo de dispositivo para después mostrar el ícono correcto, mejorando la usabilidad del sistema. Además, se puede cambiar el estado de los dispositivos en tiempo real, como se muestra en la siguiente figura.

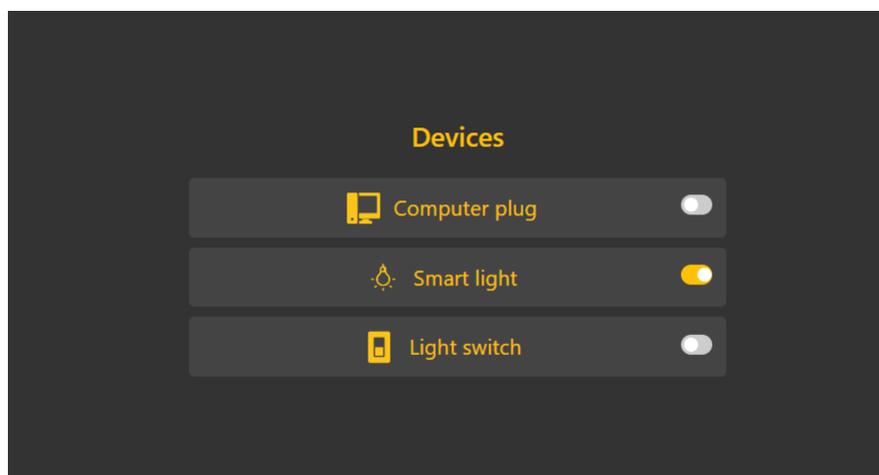


Figura 3.4: Módulo Devices

3.3.2 Módulo Consumption

El módulo *Consumption* permitió visualizar el consumo diario de todos los dispositivos, mostrándolos en un análisis de barra para obtener una mejor gestión, tomando la información requerida desde la base de datos y realizando los cálculos observados en la ecuación 2.1, como se muestra en la siguiente figura.

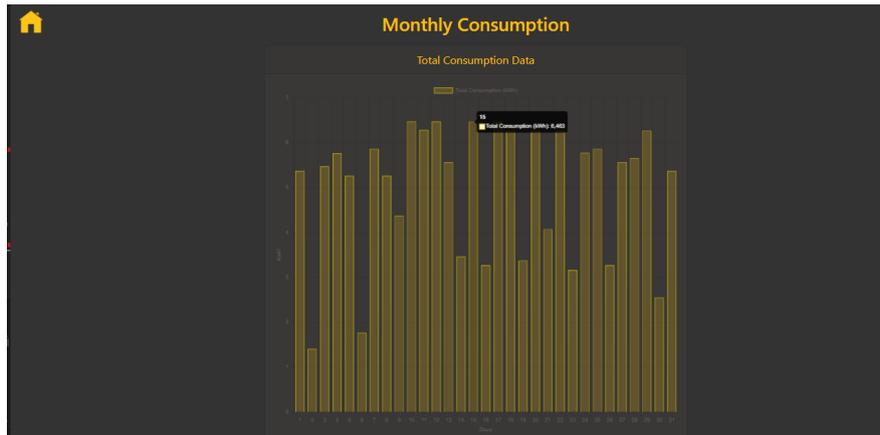


Figura 3.5: Módulo Consumption

3.3.3 Módulo Emissions CO₂

El módulo *Emissions CO₂* permitió visualizar las emisiones de carbono diarias en un análisis lineal en base a la ecuación establecida, como se muestra en la siguiente figura.

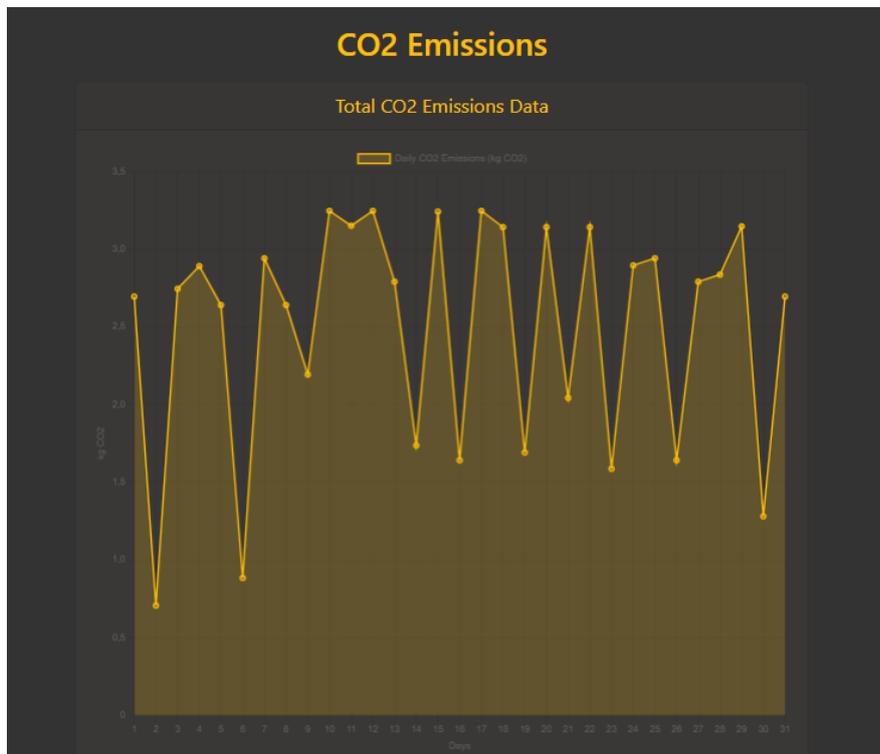


Figura 3.6: Módulo Emissions

El análisis de los resultados muestra que el sistema es capaz de gestionar de manera eficiente el consumo energético de edificios inteligentes, proporcionando datos precisos y

actualizados sobre el consumo y ahorro energético, así como sobre las emisiones de CO₂.

3.3.4 Módulo Energy Savings

El módulo *Energy Savings* permitió visualizar en un lapso de 2 meses una comparación de los dos consumos, entendiendo el caso en el que para el mes de junio no existía una concientización, por lo que el gasto fue más elevado. Sin embargo, con la ayuda del sistema y una mejor gestión energética, se pudo observar un ahorro energético en un análisis de pastel, como se muestra en la siguiente figura.

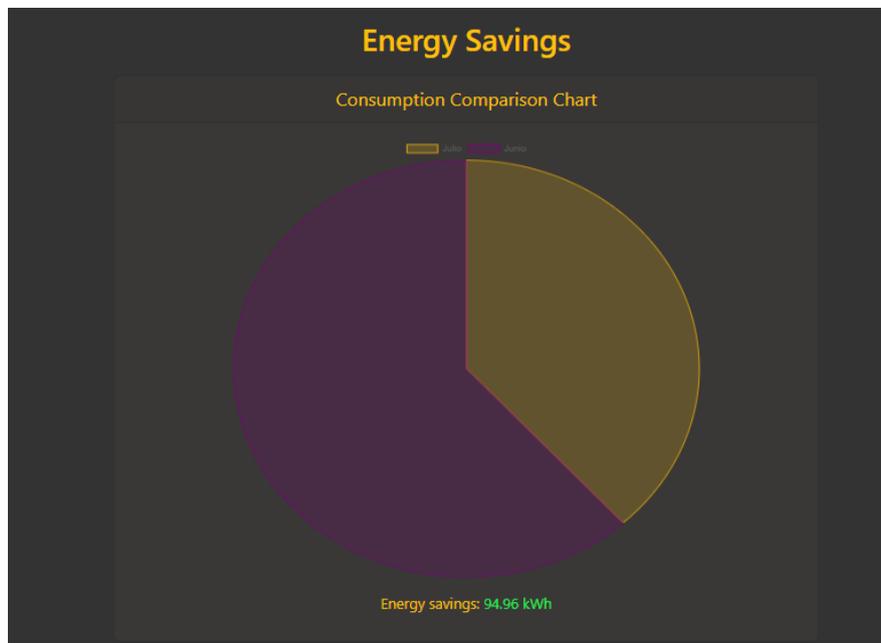


Figura 3.7: Módulo Energy Savings

3.3.5 Módulo Cost Savings

El módulo *Cost Savings* permitió visualizar en un lapso de 2 meses una comparación del costo económico dentro del país por consumo mensual, entendiendo el caso en el que para el mes de junio no existía una concientización, por lo que el gasto fue más elevado. Sin embargo, con la ayuda del sistema y una mejor gestión energética, se pudo observar un ahorro económico en un análisis de pastel, como se muestra en la siguiente figura.



Figura 3.8: Módulo Cost Savings

3.4 RESULTADOS DE USABILIDAD

Teniendo en cuenta el System Usability Scale Scoring como se muestra en la figura 3.9

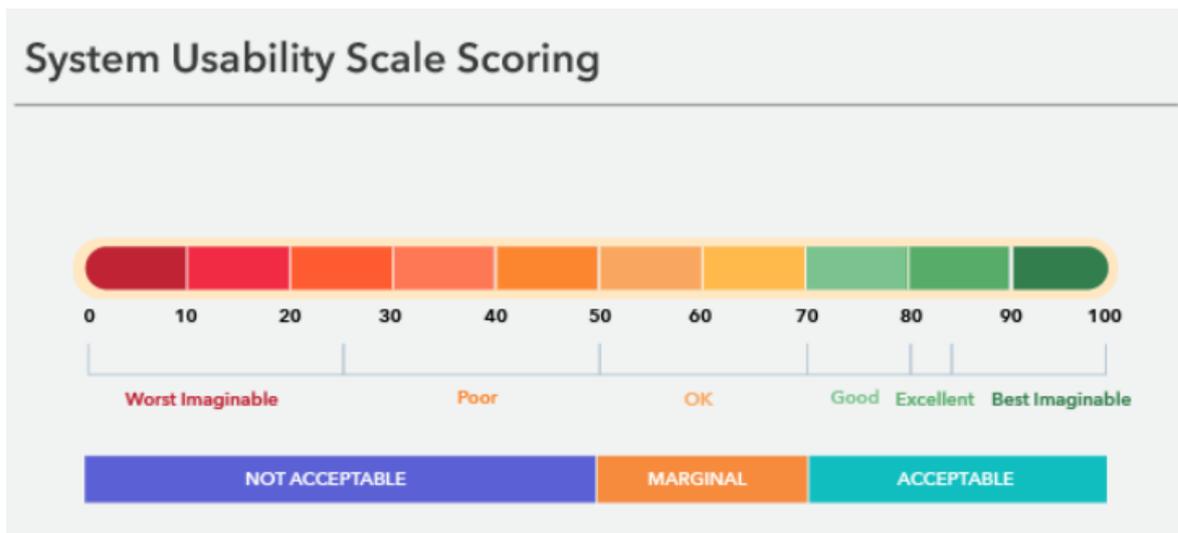


Figura 3.9: Escala de puntuación SUS

La encuesta tuvo un promedio de 89.09 entrando en un estado aceptable para todos los encuestados dando por terminado la parte de resultados logrando así redactar algunas

conclusiones acerca del desarrollo de este proyecto.

Las pruebas funcionales verificaron que todas las funcionalidades del sistema operaran correctamente. Todos los módulos del sistema pasaron las pruebas funcionales sin problemas.

- El proceso de inicio de sesión y registro de usuarios funcionó correctamente.
- La visualización y gestión de dispositivos en el dashboard se realizó de manera precisa.
- La visualización y gestión de dispositivos en el dashboard se realizó de manera precisa.

4 CONCLUSIONES

- ❑ **Conclusión 1:** Dentro del marco de análisis sobre el consumo energético de cada dispositivo mencionado en el proyecto, se puede llegar a la conclusión de que un dispositivo LED inteligente es mucho más eficiente que uno tradicional, puesto que su consumo energético es mucho menor. Si cada edificio contara con un sistema de dispositivos inteligentes capaces de ser monitoreados mediante una interfaz, se tendría una reducción considerable de consumo a nivel nacional, evitando crisis energéticas que, con el aumento de sequías, son mucho más frecuentes.
- ❑ **Conclusión 2:** En base a las investigaciones realizadas sobre el consumo energético a nivel nacional, se lograron establecer KPIs que permiten realizar una gestión energética de manera eficaz y eficiente. Se creó una interfaz que puede mostrar el consumo energético, el ahorro en emisiones de carbono, el ahorro energético y el ahorro económico en un lapso de tiempo determinado, logrando tener una visión clara sobre cuánto puede consumir y contaminar un dispositivo o una persona dentro de una habitación.
- ❑ **Conclusión 3:** Con ayuda de las herramientas establecidas, se logró crear un dashboard que permitió la visualización en tiempo real de todos los KPIs establecidos dentro de un marco mensual, incluso teniendo la capacidad de controlar los estados de los dispositivos conectados al sistema. Esto puede permitir, por ejemplo, que el área administrativa de un edificio tenga un mejor control en caso de que un foco o cualquier otro dispositivo haya quedado encendido, pudiendo actuar de una manera mucho más rápida. Ahora, hablemos de miles de edificios teniendo estos mejores controles.
- ❑ **Conclusión 4:** Para realizar este tipo de dashboards, fue mucho más cómodo usar Bootstrap como herramienta para el FrontEnd, ya que previamente se había optado por usar Vue.js, pero el problema radicó en su complejidad para acoplarlo con Flask. A diferencia de Bootstrap, que interactúa directamente con los templates HTML, te-

niendo un mejor acople con Flask y logrando buenos resultados al momento de crear la interfaz.

- ❑ **Conclusión 5** : Las pruebas permitieron realizar mejoras en cada iteración gracias al modelo iterativo e incremental, entendiendo tanto problemas en el código como en la experiencia de usuario. Esto permitió mejorar mediante nuevas historias de usuario que integraron estas mejoras. No fue necesario incluir todas estas iteraciones en el documento con el fin de evitar demasiado contenido teórico, sino más bien mostrar el resultado final.

- ❑ **Conclusión 6**: Gracias a las pruebas de usabilidad realizadas, se pudo demostrar que el prototipo entró en un marco aceptable con más del 80 % de aprobación, logrando así tener un producto funcional y terminado, siempre esperando implementar nuevas mejoras.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] J. F. Díaz, *BALANCE ENERGÉTICO NACIONAL | 2022*. dirección: https://www.celec.gob.ec/wp-content/uploads/2023/08/Balance-Energetico-Nacional-BEN-2022_.pdf#page=21&zoom=100,0,0 (visitado 03-06-2024).
- [2] J. Martins, *Qué es un KPI, para qué sirve y cómo utilizarlo en tu proyecto*. dirección: <https://asana.com/es/resources/key-performance-indicator-kpi> (visitado 03-06-2024).
- [3] M. Y. (Ho, J. Lai, H. Hou y D. Zhang, «Key Performance Indicators for Evaluation of Commercial Building Retrofits: Shortlisting via an Industry Survey,» *Energies*, vol. 14, n.º 21, pág. 7327, 2021. DOI: 10.3390/en14217327. dirección: <https://www.mdpi.com/1996-1073/14/21/7327>.
- [4] F. J. G. Peñalvo y A. G. Holgado. «Modelos de proceso.» Último acceso: 2024-06-11. (2022), dirección: https://repositorio.grial.eu/bitstream/grial/2514/1/IS_1%20Tema%203%20-%20Modelos%20de%20Proceso.pdf.
- [5] O. F. N. Isyaku Maigari Ibrahim e I. R. Saidu, «Iterative and Incremental Development Analysis Study of Vocational Career Information Systems,» *IJSEA*, vol. 11, n.º 5, pág. 12, 2020. DOI: 10.5121/ijsea.2020.11502. dirección: <https://airconline.com/ijsea/V11N5/11520ijsea02.pdf>.
- [6] D. A. O. Negrín. «El modelo incremental para el desarrollo de un sistema informático de gestión de información.» Último acceso: 2024-06-11. (), dirección: <https://informatica-juridica.com/wp-content/uploads/2021/05/ArticuloBagarotti.pdf>.
- [7] E. Solano-Fernández y D. Porrás-Alfaro, «El modelo iterativo e incremental para el desarrollo de la aplicación de realidad aumentada Amón_{RA},» *Tecnología en marcha*, págs. 165-177, 2020. DOI: 10.18845/tm.v33i8.5518. dirección: <https://airconline.com/ijsea/V11N5/11520ijsea02.pdf>.
- [8] F. Arias y A. Freire, «Study of Energy Performance Indicators (EnPI) of a Basic Hospital in Ecuador,» *Revista Energía*, n.º 20, págs. 33-42, 2023. DOI: 10.37116/

revistaenergia.v20.n1.2023.580. dirección: <https://doi.org/10.37116/revistaenergia.v20.n1.2023.580>.

- [9] Eartheasy. «LED Light Bulbs: Comparison Charts.» (2024), dirección: <https://learn.eartheasy.com/guides/led-light-bulbs-comparison-charts/> (visitado 01-07-2024).
- [10] M. E. M. García, C. C. Hinojosa y R. A. Zamora, «Consumo de energía eléctrica,» Sin información adicional. dirección: <http://hdl.handle.net/11285/631169>.
- [11] E. S. Claudia Moya VerÚnica Marcillo, «ANÁLISIS Y DETERMINACIÓN DEL COSTO DEL SERVICIO PÚBLICO DE ENERGÍA ELÉCTRICA (SPEE),» Dirección de Regulación Económica y Tarifas del Sector Eléctrico, Quito, Ecuador, inf. téc., 2023, Disponible en: https://www.controlrecursosyenergia.gob.ec/wp-content/uploads/downloads/2023/07/INF-DRETSE-2023-041_Costos-SPEE-2024.pdf.
- [12] G. de Catalunya, «Guía práctica para el cálculo de emisiones de gases de efecto invernadero (GEI),» *Barcelona: Oficina Catalana del Canvi climàtic*, 2011.
- [13] M. de Energía y Minas Gobierno del Ecuador, «Factor de Emisión de CO2 del Sistema Nacional Interconectado de Ecuador,» 2022, Informe 2022. dirección: <https://www.recursosyenergia.gob.ec/wp-content/uploads/2023/08/wp-1692720103183.pdf>.
- [14] A. Nagpal y G. Gabrani, «Python for Data Analytics, Scientific and Technical Applications,» en *2019 Amity International Conference on Artificial Intelligence (AICAI)*, IEEE, 2019, págs. 141-145. DOI: 10.1109/AICAI.2019.8701341.
- [15] B. O'Grady, *What is JavaScript and Why Should I Learn It?* <https://codeinstitute.net/global/blog/what-is-javascript-and-why-should-i-learn-it/>, Accessed: 2024-07-03, 2024.
- [16] B. Cabeza, *The Power of Figma as a Design Tool*, <https://www.toptal.com/designers/ui/figma-design-tool>, Accessed: 2024-07-03, 2024.
- [17] R. M. C. Valarezo, «Aplicación web para la gestión de pagos de servicios proporcionados por las unidades de producción de la Universidad Técnica de Ambato utilizando procesamiento de imágenes para la validación de datos,» Tutor: Ing. Santiago David Jara Moya, Mg., Trabajo de titulación modalidad Proyecto de investigación, Universidad Técnica de Ambato, Facultad de Ingeniería en Sistemas, Electrónica e Industrial, Carrera de Software, Ambato, Ecuador, feb. de 2024.
- [18] M. Grinberg, *The New And Improved Flask Mega-Tutorial*, <https://github.com/miguelgrinberg/microblog>, Accessed: 2024-07-24, 2020.

- [19] M. D. L. Iza y J. R. O. Bedoya, *Desarrollo de un sistema de gestión integrado utilizando software libre con el modelo iterativo incremental para llevar el control de los procesos en la empresa "Software y Hardware"*, Proyecto de investigación presentado previo a la obtención del Título de Ingenieros en Informática y Sistemas Computacionales, Latacunga, Ecuador, 2016. dirección: tu-url-aqui.
- [20] A. Menzinsky, G. López, J. Palacio, M. Á. Sobrino, R. Álvarez y V. Rivas, *Historias de Usuario*, <https://www.safecreative.org/work/2009135322450>, Versión 3.01, Agosto de 2022.
- [21] CEDESOFTE, *Entregable A4.3 v2: Análisis de herramientas para el diseño de mockups*, Proyecto: Plataforma tecnológica de georecomendación para productos y servicios en modalidad subasta inversa basado en técnicas de cómputo sensible al contexto, análisis de sentimientos, minería de opiniones, Universidad Tecnológica del Centro de Veracruz – Innovasolver, 2018.

6 ANEXOS

6.1 ENTREVISTA CON CONSTRUCTORAS

6.1.1 Percepciones y Preferencias sobre Edificios Inteligentes en la Industria de la Construcción

https://epnecuador-my.sharepoint.com/:b:/g/personal/willian_rivera_epn_edu_ec/EWVtctpEGz5GkEMGJc...e=WFqO8a

6.1.2 Análisis de resultados

https://epnecuador-my.sharepoint.com/:b:/g/personal/willian_rivera_epn_edu_ec/EfFBr-rnG0hAlb4q6hWY...e=pfNvCe

6.1.3 Visualización de resultados en PowerBI

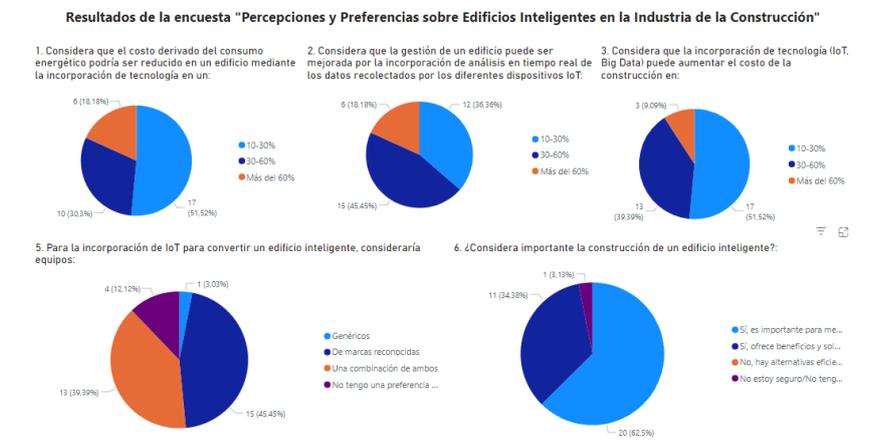


Figura 6.1: Resultados de encuestas única opción

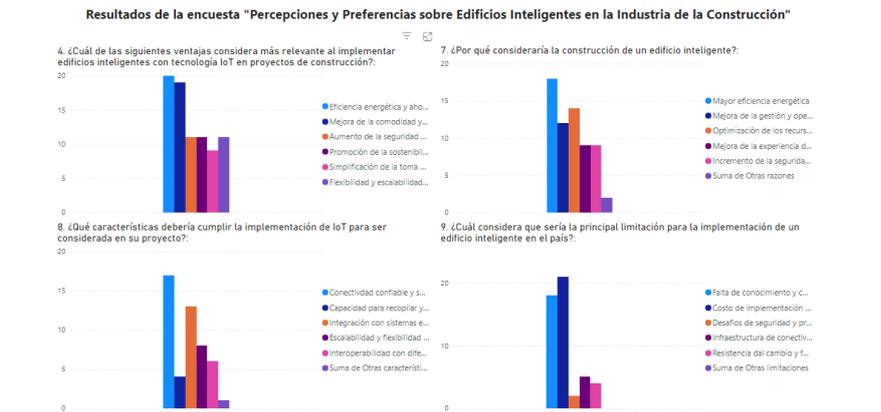


Figura 6.2: Resultados de encuestas opción múltiple

6.2 ANÁLISIS Y DISEÑO

6.2.1 Desarrollo del prototipo

<https://github.com/andreag020/iotDashboard>

6.3 PRUEBAS

6.3.1 Ejecución en tiempo real con estudiantes



Figura 6.3: Exposición del sistema a los estudiantes



Figura 6.4: Pruebas con los estudiantes del sistema

6.3.2 Encuesta SUS para estudiantes

<https://forms.office.com/r/J3vBZ22R7S?origin=lprLink>

6.3.3 Reporte de pruebas

https://epnecuador-my.sharepoint.com/:w:/g/personal/willian_rivera_epn_edu_ec/EbQ69JlyEX9Mt3Fngav0e=Rg6eTF