



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

MONITOREO DE INDICADORES CLAVE DE DESEMPEÑO DE GESTIÓN
ENERGÉTICA EN UN CAMPUS UNIVERSITARIO APLICANDO
CIENCIA DE DATOS.

CRUZ ZAMBRANO PEDRO RICARDO
INGENIERO DE SISTEMAS

MACHALA
2022



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

MONITOREO DE INDICADORES CLAVE DE DESEMPEÑO DE
GESTIÓN ENERGÉTICA EN UN CAMPUS UNIVERSITARIO
APLICANDO CIENCIA DE DATOS.

CRUZ ZAMBRANO PEDRO RICARDO
INGENIERO DE SISTEMAS

MACHALA
2022



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

TRABAJO TITULACIÓN
PROPUESTAS TECNOLÓGICAS

MONITOREO DE INDICADORES CLAVE DE DESEMPEÑO DE GESTIÓN
ENERGÉTICA EN UN CAMPUS UNIVERSITARIO APLICANDO CIENCIA DE
DATOS.

CRUZ ZAMBRANO PEDRO RICARDO
INGENIERO DE SISTEMAS

HERNANDEZ ROJAS DIXYS LEONARDO

MACHALA, 25 DE FEBRERO DE 2022

MACHALA
2022

Tesis Pedro Cruz

INFORME DE ORIGINALIDAD

0%

INDICE DE SIMILITUD

0%

FUENTES DE INTERNET

0%

PUBLICACIONES

0%

TRABAJOS DEL
ESTUDIANTE

FUENTES PRIMARIAS

Excluir citas

Activo

Excluir coincidencias < 1%

Excluir bibliografía

Activo

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

El que suscribe, CRUZ ZAMBRANO PEDRO RICARDO, en calidad de autor del siguiente trabajo escrito titulado MONITOREO DE INDICADORES CLAVE DE DESEMPEÑO DE GESTIÓN ENERGÉTICA EN UN CAMPUS UNIVERSITARIO APLICANDO CIENCIA DE DATOS., otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

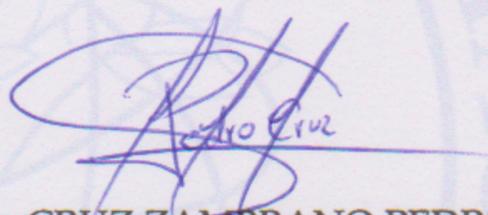
El autor declara que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

El autor como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 25 de febrero de 2022



CRUZ ZAMBRANO PEDRO RICARDO
0705459584

DEDICATORIA

El presente trabajo está dedicado a mi padre y madre por brindarme la oportunidad de cumplir con mis estudios universitarios, por impulsarme y motivarme a cumplir mis sueños y aspiraciones.

Sr. Pedro Ricardo Cruz Zambrano

AGRADECIMIENTOS

Agradezco a mi familia por brindarme su apoyo, guiarme y aconsejarme de manera correcta para poder cumplir con mis metas. A las autoridades y personal que forman parte de la Escuela de Sistemas, por su entrega y devoción por la enseñanza, fueron motivo de inspiración para aspirar a ser un gran profesional. A mis compañeras de proyecto que a lo largo de la carrera por la ayuda brindada en realizar proyectos de alta calidad.

Sr. Pedro Ricardo Cruz Zambrano

RESUMEN

Actualmente, los campus universitarios comienzan a ver utilidad en los datos que generan diariamente, la tendencia apunta a que estas universidades implementen tecnologías que les permitan ser sostenibles y crear valor con dichos datos, este tipo de desarrollo se vincula al concepto de Smart Campus. Estas universidades alrededor del mundo implementan técnicas y medidas que les permitan aumentar la eficiencia energética de sus instalaciones. En un entorno donde todo está conectado como lo es un Campus inteligente, el índice de consumo energético tiende a ser elevado. Debido a ello las Tecnologías de la información y comunicación (TIC), Internet de las cosas (IoT) y herramientas de Inteligencia de negocio (BI) adquieren una gran importancia, debido a que ayudan a monitorear e interpretar los datos de estudio permitiendo así la toma de decisiones pertinentes. Sin embargo, para alcanzar la eficiencia energética primero se debe realizar estudios que permitan entender el entorno, comportamiento y las variables que intervienen en el consumo energético. Para ello se utilizan Indicadores claves de desempeño (KPI) que ayudan a medir y cuantificar el consumo energético de un campus universitario. La gestión de grandes cantidades de datos que genera una infraestructura de un campus universitario referente al consumo energético conlleva una dificultad al generar información para la toma de decisiones, por ello es fundamental el implementar un sistema que permita el monitoreo de indicadores clave de desempeño de gestión energética. La presente propuesta tecnológica describe el desarrollo de un sistema que implemente un panel de control o Dashboard que permita el análisis de datos referentes a gestión energética, generados por medidores ABB M2M instalados en un campus universitario. Para el desarrollo del sistema, se optó por el uso de la metodología de desarrollo ágil SWIRL, la cual consta de las fases: Análisis, planificación, modelado y finalmente la ejecución de la herramienta. El panel de control se diseñó siguiendo la arquitectura API REST que permite la separación del desarrollo del cliente y el servidor, para el desarrollo del Front-End se utilizó el framework de desarrollo web “AngularJS” que maneja el patrón de diseño Modelo Vista Controlador (MVC), también hace uso de librerías como Plotly.js que permite generar gráficas estadísticas dinámicas y jsPDF para la creación de informes. Además, se utilizó Grafana para

la generación de las gráficas en tiempo real de los datos. El servidor encargado de gestionar las peticiones del cliente y proporcionar los datos necesarios, fue implementado con el framework NodeJS y Express.js, haciendo uso de la arquitectura de capas para su desarrollo. Los datos utilizados para la ejecución del sistema fueron adquiridos mediante el protocolo MQTT de medidores ABB M2M que alimentan un servidor local en el campus, dichos datos fueron sometidos a un proceso de extracción, transformación y carga (ETL), para posteriormente cargarlos a una Base de Datos Relacional como lo es PostgreSQL. Para la gestión de calidad se utilizó la normativa ISO/IEC-9126 haciendo uso de diferentes herramientas web gratuitas, dando como resultado que el sistema cumple con todas las métricas de calidad evaluadas por dicha normativa. En conclusión, el presente trabajo logró desarrollar un sistema web que permite un monitoreo de KPI's de consumo energético con una interfaz eficiente y multiplataforma escalable y fácil de administrar.

Palabras clave: KPI, Dashboard, AngularJS, Gestión energética.

ABSTRACT

Currently, university campuses are beginning to see utility in the data they generate daily. The trend is for these universities to implement technologies that allow them to be sustainable and create value with said data. This type of development is linked to the concept of Smart Campus. These universities around the world implement techniques and measures that allow them to increase the energy efficiency of their facilities. In an environment where everything is connected, such as a smart Campus, the rate of energy consumption tends to be high. Due to this, Information and Communication Technologies (TIC), Internet of Things (IoT) and Business Intelligence (BI) tools acquire great importance, because they help to monitor and interpret the study data, thus allowing the making relevant decisions. However, to achieve energy efficiency, studies must first be carried out to understand the environment, behavior and the variables that intervene in energy consumption. For this, Key Performance Indicators (KPI) are used to help measure and quantify the energy consumption of a university campus. The management of large amounts of data generated by an infrastructure of a university campus regarding energy consumption entails a difficulty in generating information for decision making, for this reason it is essential to implement a system that allows the monitoring of key management performance indicators. energetic. This technological proposal describes the development of a system that implements a control panel or Dashboard that allows the analysis of data related to energy management, generated by ABB M2M meters installed on a university campus. For the development of the system, the use of the SWIRL agile development methodology was chosen, which consists of the phases: Analysis, planning, modelling and finally the execution of the tool. The control panel was designed following the API REST architecture that allows the separation of client and server development, for the development of the Front-End the "AngularJS" web development framework was used, which manages the Model View Controller design pattern (MVC), also makes use of libraries such as Plotly.js that allows to generate dynamic statistical graphs and jsPDF for the creation of reports. In addition, Grafana was used to generate real-time graphs of the data. The server in charge of managing client requests and providing the necessary data was

implemented with the NodeJS and Express.js framework, making use of the layered architecture for its development. The data used for the execution of the system was acquired through the MQTT protocol of ABB M2M meters that feed a local server on the campus, these data were subjected to an extraction, transformation and loading (ETL) process, to later upload them to a database. Of Relational Data such as PostgreSQL. For quality management, the ISO/IEC-9126 standard was used, making use of different free web tools, resulting in the system complying with all the quality metrics evaluated by said standard. In conclusion, the present work managed to develop a web system that allows monitoring of energy consumption KPIs with an efficient and scalable and easy-to-manage multiplatform interface.

Keywords: KPI, Dashboard, AngularJS, Energy management.

ÍNDICE DE CONTENIDO

DEDICATORIA	1
AGRADECIMIENTOS	2
RESUMEN	3
ABSTRACT	5
INTRODUCCIÓN	13
1. CAPÍTULO I: DIAGNOSTICO DE NECESIDADES Y REQUERIMIENTOS	15
1.1. Ámbito de Aplicación: Descripción de contexto y hechos de interés	15
1.2. Establecimiento de requerimientos	16
1.3. Justificación del requerimiento a satisfacer	16
2. CAPÍTULO II: DESARROLLO DEL PROYECTO	17
2.1. Definición del prototipo tecnológico	17
2.1.1. Arquitectura Back-End	17
2.1.2. Arquitectura Front-End	18
2.1.3. Arquitectura de recolección de Datos	19
2.2. Fundamentación teórica del prototipo	21
2.2.1. IoT (Internet Of Things)	21
2.2.1.1. Protocolo MQTT	21
2.2.2. Smart Campus	22
2.2.2.1. Smart Grid	22
2.2.2.2. Smart Building	22
2.2.2.3. Smart Energy	23
2.2.3. KPI's	23
2.2.4. Business Intelligence	24
2.2.4.1. ETL	24
2.2.4.2. Big Data	24
2.2.4.3. Data Warehouse	25
2.2.5. AngulaJS	26
2.2.6. Node.js	26
2.2.7. Expres.js	26
2.2.8. Grafana	26
2.2.9. Revisión sistemática de la literatura	26
2.3. OBJETIVOS DEL PROTOTIPO	28
2.3.1. Objetivo General	28

2.3.2.	Objetivos específicos	28
2.4.	DISEÑO DEL PROTOTIPO	28
2.4.1.	Fase de análisis	30
2.4.1.1.	Alcance del proyecto	30
2.4.1.2.	Análisis de Factibilidad	30
2.4.1.2.1.	Factibilidad operativa	30
2.4.1.2.2.	Factibilidad técnica	31
2.4.1.2.3.	Factibilidad económica	31
2.4.1.3.	Identificación de interesados o “stakeholder”	32
2.4.1.4.	Requerimientos	33
2.4.1.4.1.	Requerimientos funcionales	33
2.4.1.4.2.	Requerimientos no funcionales	34
2.4.2.	Fase de planificación	34
2.4.2.1.	Historia de usuarios	34
2.4.2.2.	Gestión de cronograma	37
2.4.2.3.	Estructura de desglose de trabajo	39
2.4.3.	Fase de modelado	39
2.4.3.1.	Diseño de base de datos relacional	39
2.4.3.2.	Diseño navegacional	41
2.4.3.3.	Prototipado de interfaces de usuarios	41
2.4.3.4.	Diagramas UML	45
2.4.3.4.1.	Diagramas de casos de uso	45
2.4.3.4.2.	Diagrama de secuencias	48
2.5.	Ejecución y/o ensamble del prototipo	51
2.5.1.	Proceso ETL	51
2.5.1.1.	Extracción	51
2.5.1.2.	Transformación	51
2.5.1.3.	Carga	53
2.5.2.	Datos para simulación	53
2.5.3.	Desarrollo del Back-End	56
2.5.4.	Desarrollo Front-End	58
2.5.5.	Desarrollo de gráficas en Grafana	61
2.5.6.	Interfaces gráficas del sistema	61
3.	Capítulo III. Evaluación del prototipo	62
3.1.	Plan de Evaluación	62
3.2.	Resultados de Evaluación	63

3.2.1. Evaluación de calidad	63
3.2.2. Evaluación con herramientas	64
3.3. Conclusiones	66
3.4. Recomendaciones	67
BIBLIOGRAFÍA	67
Anexos	72
Anexo 1: Análisis de Riesgos	72
Anexo 2: Gestión de la calidad	76
Anexo 3: Acta de constitución	78

INDICE DE TABLAS

Tabla 1: Comparativa entre Data WareHouse	25
Tabla 2: Revisión sistemática de la literatura de KPI's	27
Tabla 3: KPI's del sistema	29
Tabla 4: Factibilidad Operativa.....	30
Tabla 5: Factibilidad Técnica	31
Tabla 6: Factibilidad Económica	31
Tabla 7: Stakeholder.....	33
Tabla 8: Requerimientos Funcionales	34
Tabla 9: Requerimientos no Funcionales	34
Tabla 10: Historia de usuario N1 - Ingreso al Sistema.....	34
Tabla 11: Historia de usuario N2 - Visualización de datos Grafana	35
Tabla 12: Historia de usuario N3 - Visualización de datos Histórico.....	35
Tabla 13: Historia de usuario N4 - Filtro de Gráficos	36
Tabla 14: Historia de Usuario - Datos de Simulación	36
Tabla 15: Historia de usuario N6 - Reportes	37
Tabla 16: Cronograma	37
Tabla 17: Estructura de tablas de la base de datos.....	40
Tabla 18: Data de consumo histórico.....	53
Tabla 19: Cálculo de consumo.....	54
Tabla 20: Porcentaje de Consumo por medidor	54
Tabla 21: Cálculo de porcentaje por línea de carga	55
Tabla 22: Escala de Likert	63
Tabla 23: Evaluación de calidad	63
Tabla 24: Herramientas de SEO.....	64
Tabla 25: Matriz de riesgos	73
Tabla 26: Interesados de la calidad	76
Tabla 27: Roles y Responsabilidades en la calidad.....	77
Tabla 28: Acta de constitución	78

ÍNDICE DE GRÁFICOS

Ilustración 1. Arquitectura General.....	17
Ilustración 2. Arquitectura Back-End.....	18
Ilustración 3. Arquitectura Front-End	19
Ilustración 4. Arquitectura de recolección de datos	20
Ilustración 5. EDT.....	39
Ilustración 6. Diseño relacional de la base de datos	40
Ilustración 7. Diseño navegacional.....	41
Ilustración 8. Prototipo Login	42
Ilustración 9. Prototipo Inicio.....	42
Ilustración 10. Prototipo Dashboard tiempo real	43
Ilustración 11. Prototipo dashboard histórico por mes	43
Ilustración 12. Prototipo dashboard histórico por año.....	44
Ilustración 13. Prototipo visualización de datos de simulación	44
Ilustración 14. Prototipo CRUD de Datos de simulación.....	45
Ilustración 15. Caso de uso Inicio de sección	45
Ilustración 16. Caso de uso Visualizar dashboard en tiempo real	46
Ilustración 17. Caso de uso visualizar dashboard histórico	46
Ilustración 18. Caso de uso Crear reporte.....	47
Ilustración 19. Caso de uso Editar datos de simulación.....	47
Ilustración 20. Diagrama de secuencia - Inicio de sección	48
Ilustración 21. Diagrama de secuencia - Cargar Dashboard Grafana.....	48
Ilustración 22. Diagrama de secuencia - Dashboard histórico	49
Ilustración 23. Diagrama de secuencia - Generar Reporte	49
Ilustración 24. Diagrama de secuencia - CRUD datos simulación	50
Ilustración 25. Carga de datos en Drive	51
Ilustración 26. Importación de librerías	51
Ilustración 27. Eliminación de datos no útiles	51
Ilustración 28. Agregación de datos de identificador	52
Ilustración 29. Reorganización de datos	52
Ilustración 30. Formateo de Fecha.....	52
Ilustración 31. Guardado de datos.....	52
Ilustración 32. Programa de carga de datos a base de datos	53
Ilustración 33. Datos referenciales del muestreo	53
Ilustración 34. Datos correspondientes a la simulación	55
Ilustración 35. Estructura del Back-End	56
Ilustración 36. Configuración del servidor Back-end	56
Ilustración 37. Configuración de conexión a la base de Datos.....	57
Ilustración 38. Codificación de rutas del servidor	57
Ilustración 39. Codificación de controller del servidor	57
Ilustración 40. Estructura del Front-End	58
Ilustración 41. Maquetado del menú.....	58
Ilustración 42. Hoja de Estilos del menú principal	59
Ilustración 43. Maquetado de Dashboard y carga de gráficos de Grafana.....	59
Ilustración 44. Codificación de Gráficos histórica	60
Ilustración 45. Codificación de generación de reporte	60
Ilustración 46. Gráficas generadas de Grafana	61

Ilustración 47. Dashboard en Tiempo Real	61
Ilustración 48. Dashboard Histórico	61
Ilustración 49. Gráficas históricas	62
Ilustración 50. Datos de simulación	62
Ilustración 51. Google Speed	65
Ilustración 52. GTMetrix.....	65
Ilustración 53. Functional Accessibility Evaluator 2.2	66
Ilustración 54. Equipo de calidad	76

INTRODUCCIÓN

En la actualidad, la gestión energética se ha vuelto una prioridad en muchos países, por lo tanto, las instituciones, como lo son las universidades, se preocupan y desarrollan infraestructura que les permita gestionar su consumo energético. Cuando hablamos de gestión energética y campus universitario, surgen conceptos como lo son Smart Energy, Smart Campus, Smart Building, entre otros. De igual manera se trae a colación tecnologías y herramientas que se utilizan para estructurar estos sistemas, como lo son IoT (Internet Of Things), KPI (Indicadores claves de desempeño), Business Intelligence, entre otros.

Actualmente, las universidades optan por implementar Smart Campus en sus instalaciones, las cuales tienden a generar gran demanda de consumo energético y de la misma forma generan un volumen elevado de datos [1]. Según Villegas *et al.* [2], una de las características de un Smart Campus es la creación de ambientes seguros, sostenibles y confortables que contribuyan con la gestión y desarrollo del aprendizaje de manera efectiva. No obstante, contar con infraestructura que permita la recolección de datos de consumo energético no se deriva en una gestión eficiente del mismo. Como lo menciona Torres *et al.* [3], en su artículo, la observación y análisis de los datos permiten desarrollar prototipos y sistemas que ayudan a generar información de valor en cuando al consumo energético, ellos desarrollan un sistema telemétrico con IoT que permite observar en tiempo real el consumo energético en áreas residenciales.

Para el análisis de la gestión energética en un Smart campus se puede hacer uso de Indicadores claves de desempeño (KPI). Los KPI nos dan la posibilidad de realizar comparaciones entre diferentes infraestructuras o áreas dentro de un Smart Campus permitiéndonos realizar cotejos en el consumo de energía [4].

El presente trabajo tiene como objetivo realizar una gestión energética en un campus universitario mediante la utilización de indicadores claves de desempeño, basado en datos generados por una infraestructura IoT, que cuenta con medidores ABB M2M implementados en redes eléctricas de diferentes áreas, haciendo uso de un sistema web basado en la arquitectura API REST,

que integre técnicas y herramientas de inteligencia de negocios para la creación de un dashboard como un sistema de apoyo de decisiones.

La estructura del presente informe se muestra a continuación:

En el Capítulo 1, describe la necesidad de implementar un Sistema para el monitoreo de variables energéticas y sus requerimientos.

El Capítulo 2, describe la estructura del prototipo, la documentación teórica, objetivos, diseño y ejecución del prototipo.

El Capítulo 3, se define el plan de evaluación y los resultados obtenidos del mismo, las conclusiones y recomendaciones acorde a los resultados.

1. CAPÍTULO I: DIAGNOSTICO DE NECESIDADES Y REQUERIMIENTOS

1.1. **Ámbito de Aplicación: Descripción de contexto y hechos de interés**

Al hablar de monitoreo de variables de gestión energética en campus universitarios naturalmente se surge el concepto de Smart campus, y en un contexto mundial, entre los proyectos más destacables debemos mencionar la alianza entre el MIT y Microsoft Research iniciada en 1999 llamada **“iCampus”**, cuyo objetivo es demostrar el potencial revolucionario de la tecnología en el plan de estudios universitarios.[5]. Otro proyecto a destacar es **“Green Campus”**, un proyecto europeo comenzado en 2012 conformado por 4 universidades ubicadas en Helsinki, Lisboa, Luleå, y Milan. El objetivo del proyecto fue evidenciar el rol central que tiene las TIC's en instalaciones universitarias. Logrando evidenciar resultados en el ahorro de energía y la diferencia del ahorro obtenido según la estructura y ubicación de la infraestructura en donde se implementaron los equipos de TIC [6].

Según Rico-Bautista *et al.* [7], a nivel latinoamericano en las universidades el uso de procesos, arquitecturas y tecnologías “Smart” se enfoca principalmente en la implementación de tecnologías como: IA (inteligencia artificial), Cloud computing, IoT y Big data. Menciona que Colombia, Ecuador y Argentina son los países que destacan por tener mayor cantidad de trabajos relacionados con tecnología Smart en la región.

A nivel nacional Recalde *et al.* [8], destaca proyectos como el “Balzay Smart Grid Scientific and Technological Center” de la universidad de Cuenca que implementa dinámica de fluidos, bioenergía, ciudades sostenibles aplicada a laboratorios de geomática e informática avanzada. Otro proyecto destacable es “The Intelligent Network Experimental and Research Smart Grid Laboratory” por parte de la Escuela Politécnica Nacional que abarca temas como: energía solar, generación eólica con emulación de sistemas de transmisión y cargas variables controlada por un sistema de gestión energética.

La presente propuesta tecnológica plantea la implementación de sistema web, que integre un dashboard que permita la visualización del comportamiento de indicadores claves de desempeño de gestión energética en tiempo real y con data histórica, proporcionados por medidores instalados en un campus universitario. También el sistema debe constar con un generador de datos que simulará los datos que alimentaran al dashboard, esto es debido a que las circunstancias mundiales no permiten acceder al servidor del campus universitario y al no haber clases presenciales los datos que captan los medidores no representarían un comportamiento normal de clases habituales, para la generación de datos se realizará uso de datos adquiridos antes de la pandemia, dichos datos serán sometidos a un proceso de ETL (Extraer, Transforma y Cargar).

1.2. Establecimiento de requerimientos

El sistema permitirá realizar conexión con datos que se generan en tiempo real sobre variables de gestión energéticas almacenados en una base de datos relacional. La Base de Datos a la cual se debe realizar conexión es PostgreSQL. En cuanto a la visualización de datos se empleará Grafana como generador de gráficos en tiempo real y Plotly.js para la generación de gráficos históricos de la data almacenada, estos mismos se inyectarán en una página WEB que se implementará mediante Angular.js como el front-End y Node.js como back-End.

En cuando al sistema de simulación de cargas eléctricas, será desarrollado en su totalidad con javascript utilizando la arquitectura API Rest para comunicarse con el front-end. Lo concerniente a la generación de informes se desarrollará en JavaScript y generará archivos PDF's descargables. El sistema tendrá la capacidad de modificar o agregar data correspondiente a la simulación de los datos generados de variables de simulación.

1.3. Justificación del requerimiento a satisfacer

Es imprescindible la creación de un aplicativo web que permita el monitoreo de las principales variables de gestión energética de un campus universitario y su importancia reside en poder detectar y corregir errores en la

implementación o distribución de las cargas eléctricas, esto se traduce en una mayor eficiencia, una reducción de huella de carbono y en un ahorro de costos en el consumo energético, además de la facilidad que ofrece la visualización de los gráficos estadísticos para a toma de decisiones y análisis histórico de patrones estadísticos.

El objetivo de la propuesta tecnológica es el diseño, desarrollo e implementación de un prototipo que permita visualizar en tiempo real del comportamiento de variables de gestión energética a partir de un conjunto datos que permita la simulación de datos que asemejen el comportamiento habitual de un campus universitario.

2. CAPÍTULO II: DESARROLLO DEL PROYECTO

2.1. Definición del prototipo tecnológico

El prototipo está basado en la arquitectura REST que permite separar el desarrollo del back-end con el front-end y logra la comunicación entre ellos mediante solicitudes HTTP lo cual se traduce en comunicaciones más ligeras debido a la flexibilidad de acceder solamente a los recursos necesarios lo cual vuelve a la aplicación mantenible y escalable.

Ilustración 1. Arquitectura General



Fuente: Elaboración propia

2.1.1. Arquitectura Back-End

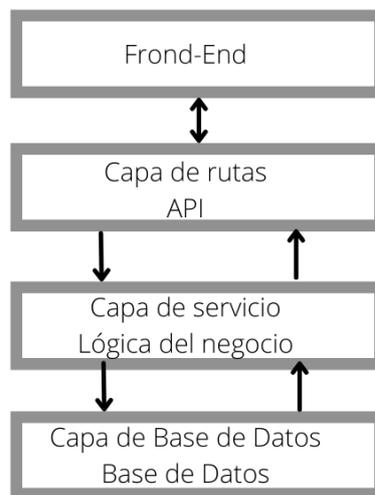
Para el desarrollo de back-end se utilizó la arquitectura de capas la cual consiste de 3 capas que se describen a continuación:

Capa de rutas: Se encarga de manejar la interfaz de programación de aplicaciones (API) y su trabajo consiste en retornar las respuestas del servidor al cliente

Capa de Servicios: Se encarga de manejar la lógica de negocios de la aplicación

Capa de Base de datos: Se encarga de realizar conexión con la base de datos para realizar las operaciones CRUD (Create, Read, Update, Delete), maneja la lógica relacionada a solicitudes y respuestas del servidor

Ilustración 2. Arquitectura Back-End



Fuente: Elaboración propia

2.1.2. Arquitectura Front-End

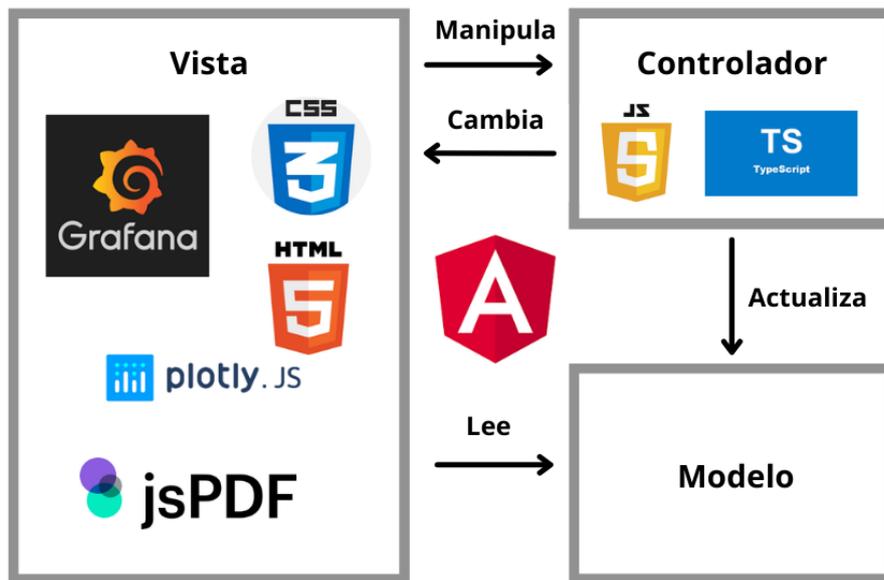
El desarrollo del front-End está basado en el patrón MVC (Modelo-Vista-Controlador) propio de Angular que provee la facilidad de administrar y ejecutar cambios en los módulos sin que estos interfieran entre sí.

Modelo: establece los datos

Vista: Presenta y actualiza los datos utilizando los controladores o eventos

Controlador: Decide como se muestran los datos

Ilustración 3. Arquitectura Front-End



Fuente: Elaboración propia

2.1.3. Arquitectura de recolección de Datos

El Campus universitario está conformado por diferentes facultades de las cuales se escoge la facultad de ingeniería civil una como eje central de estudio para el análisis del caso.

La facultad de Ingeniería de civil esta dividida en edificios, el edificio de la escuela de informática que está dividida en plantas y a su vez en aulas y laboratorios, de la cual, la obtención de los datos se realizó mediante medidores ABB M2M que son analizadores de redes eléctricas Máquina a Máquina (M2M) implementado en cada planta del edificio, el cual permite analizar diferentes parámetros en una red eléctrica.

La facultad almacena todos los datos en InfluxDB que es una base de datos que trabaja con series de temporales que permite manejar altas cargas de lecturas y escritura de datos, lo cual la hace ideal cuando se trabaja con tecnología como IoT, sin embargo, no cuenta con un sistema capaz de permitirle integrar y analizar dichos datos, de manera que sirvan como soporte a decisiones gerenciales y de mandos medios.

Los datos son recolectados a través del protocolo MODBUS que consiste en la transmisión de información entre distintos equipos electrónicos conectados a un mismo bus de datos. Luego se comunica los datos con el servidor

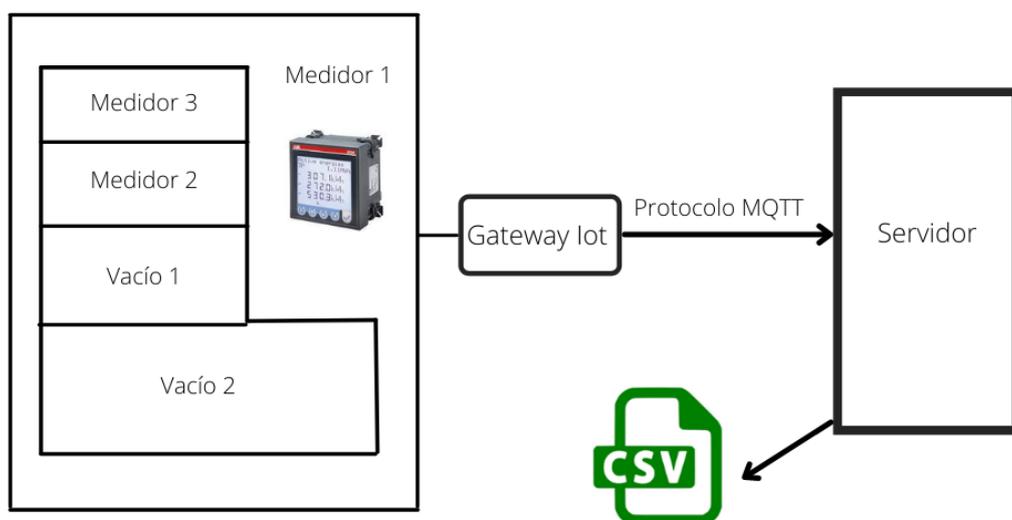
usando el protocolo MQTT que utiliza el modelo de publicación y publicación, modelo ideal para el uso de mensajería de IoT, como lo son los medidores utilizados en el presente trabajo.

Al no poder acceder al medidor y extraer los datos en tiempo real, se realizó un backup en formato CSV con los datos necesarios para llevar a cabo el desarrollo de la presente propuesta tecnológica.

Los datos a utilizar constan de 313437 registros correspondientes a 3 medidores, que conciernen a 13 días de 2 meses del año 2020, que dan un total de 82.5 MB en total con mediciones tomadas cada segundo en un periodo de 7am a 15pm de lunes a viernes. Lo que corresponde a grandes volúmenes de datos si la medición se vuelve contantes en un intervalo de un año.

La estructuración de los medidores consta de un medidor general y dos medidores que recopilan datos de dos de las cuatro plantas que tiene el edificio, por lo tanto, se empleará de ciencia de datos para extraer la información correspondiente a los pisos sin presencia de medidores.

Ilustración 4. Arquitectura de recolección de datos



Fuente: Elaboración propia

2.2. Fundamentación teórica del prototipo

2.2.1. IoT (Internet Of Things)

Según Novillo et al. [9], el Internet de las Cosas es una de las tecnologías más importantes de los últimos años gracias a los beneficios que promete su implementación en entornos domésticos, ya que permite que sensores recopilen datos, proporcionando así tecnología de gran importancia para una variedad de sistemas inteligentes. La aplicación de tecnología IoT en las industrias se refleja en múltiples beneficios como mayor productividad, rentabilidad sostenible, optimización de recursos, optimización de procesos y recopilación de datos [10]–[16].

Dicho esto debemos repasar también los inconvenientes que acarrea las soluciones de IoT, en su artículo Yaïci et al. [17], menciona que en muchos casos es complicado aplicar gran cantidad de dispositivos IoT debido a la intercomunicación que deben establecerse entre estos dispositivos, y menciona que no siempre es factible una solución cableada para la interconexión, de igual manera menciona que los desafíos y limitaciones varían en función al problema a resolver y resalta un problema fundamental, que es el tiempo de inactividad de los sensores o dispositivos que conllevan errores o datos sin recoger de vital importancia, por lo que visibiliza la importancia del monitoreo y ubicación de los dispositivos para la supervisión o remplazo de los equipos IoT [18], [19].

2.2.1.1. Protocolo MQTT

MQTT son las siglas de Message Queue Telemetry Transport (Transporte de telemetría de cola de mensajes) según Guaman et al [20], es el protocolo perfecto para dispositivos que cuentan con memoria y batería limitada, escalable en entornos de redes no confiables. Se basa en el paradigma Publicación-Suscripción, funciona sobre el protocolo TCP/IP, cuenta con mensajes de longitud fija de 8 bits, los mensajes son enviados por los publicadores al broker, broker se denomina al nodo central que utiliza la arquitectura Publicación-Suscripción, y son recibidos por los suscriptores [21]–[25].

2.2.2. Smart Campus

De acuerdo con Vasileva et al. [26], al referirnos a Smart Campus hablamos de un ambiente donde todos sus involucrados cuentan con una experiencia educativa en cualquier lugar y cualquier momento. En su artículo Villegas, Palacios y Luján [1], se alude que un Smart campus debe permitir una gestión óptima de recursos, proveer un ambiente ameno para el aprendizaje, de la misma manera ofrece a la población universitaria una mejor convivencia.

Dicho esto, sabemos que un Smart campus al tener diferentes tipos de tecnología implementada genera de igual forma diferentes tipos de datos, estos datos nos pueden proveer de forma global una perspectiva general de las acciones e interacciones realizadas dentro del mismo, por lo general son datos sueltos o datos en bruto que pueden ser organizados y transformados para la realización de análisis específicos conforme a los datos de interés [27]–[29].

2.2.2.1. Smart Grid

Según Collatta y Pau [30], smart grid o redes inteligentes son la evolución del sistema convencional de distribución eléctrica, y surgen debido a la exigente demanda de energía, la implementación de fuentes de energía renovables, y los sistemas basados en las TIC. De acuerdo con Mbungu et al [31], una red inteligente hace referencia a una red capaz de gestionar la energía eléctrica, acondicionar la energía limpia de manera inteligente y permitir participar al consumidor de manera eficiente. En su artículo H. Al Haj Hassan et al [32], se define a Smart grid como una evolución de la red eléctrica, asentada en la unificación de comunicaciones bidireccionales hacia todas las partes de la red. En particular una forma económica de mejorar la eficiencia y aumentar su capacidad a través de la integración de nuevos servicios. Por último, Ullah, Faheem y kim [33], se define a Smart grid de modo sencillo, como la integración de tecnologías de comunicación con la infraestructura de energía eléctrica.

2.2.2.2. Smart Building

En su artículo Divina, Vela y Torres [34], definen como Smart Building a edificios que hacen uso de sensores, actuadores y microchips, que permitan compilar datos y administrarlos, de esta forma procesar los datos del

consumo energético permitiendo interpretar dichos datos para mejorar su eficiencia, detección y diagnóstico de fallas. Por otra parte, Benaventes [35], define a Smart Building como el uso inteligente de sensores y tecnologías de comunicación que permitan controlar y optimizar el uso del recurso energético con el fin de ofrecer comodidad para sus ocupantes.

2.2.2.3. Smart Energy

En su artículo Sittón et al [36], define a Smart energy como un espacio donde se implementa tecnología IoT que contribuya al empleo eficiente de la energía, de este modo obtener sistemas o aplicaciones capaces de medir, gestionar y controlar el consumo de energía. Cuando nos referimos a Smart energy hablamos de un modo eficiente de producir, distribuir y consumir energía eléctrica, tomando en consideración esto, el cliente o consumidor pasa a ser el centro del sistema, smart grid es un término que surge para agrupar todo lo relacionado con Smart energy.

En definitiva, Smart energy provee grandes beneficios en su implementación, pero requiere de un gran esfuerzo al igual que una acción firme, continua y decidida. En su artículo Dincer y Acar [37]. Mencionan que para considerar inteligente a un sistema energético este debe cumplir con varios requisitos y expectativas simultáneas. También mencionan los siguientes puntos como las expectativas claves de los sistemas de Smart energy.

- Enérgicamente sano.
- Energéticamente seguro.
- Ambientalmente benigno.
- Económicamente factible.
- Comercialmente viable.
- Socialmente aceptable.
- Integrable.
- Confiable.

2.2.3. KPI's

Según Li *et al.* [38], los KPI o indicadores claves de desempeño tienen la finalidad de representar una clave multinivel de información, permitiendo medir el avance realizado y ofrecer una comprensión del desempeño y áreas

que necesiten mejoras. En el artículo de Amrina e Imasuri. [39], nos expone que al realizar una ponderación de KPI se logra determinar que unos tienen mayor relevancia respecto a otros, esto nos ayuda a mejorar el rendimiento en general al tomar decisiones oportunas.

2.2.4. Business Intelligence

BI (Inteligencia de negocio) cumple un papel decisivo para casi todo tipo de negocio. Por lo general las empresas lo utilizan para analizar datos y respaldar decisiones empresariales, pues provee un impacto positivo en la decisión de los negocios mediante sistemas de apoyo de decisiones [40], [41].

2.2.4.1. ETL

ETL son las siglas de Extract-Transform-Load (Extracción-Transformación-Carga). En breves palabras ETL consiste en mover datos de múltiples fuentes, darles formato para luego integrarlos en un solo lugar listo para ser procesados. “Los procesos ETL generalmente juegan un papel importante en la integración de datos y se utilizan siempre que los datos de diferentes sistemas de origen deban transferirse a otros sistemas de almacenamiento de datos y adaptarse a los nuevos requisitos” [42].

2.2.4.2. Big Data

Xindong Wu et al [43], en su artículo nos explica que nos referimos a Big Data cuando se trabaja con fuentes de datos de gran volumen, estas fuentes pueden ser autónomas, heterogéneas y descentralizadas. Liao et al [44], resalta 5 características de Big Data, las cuales son:

- Variedad: Datos de múltiples fuentes, tipos, formatos y estructuras.
- Volumen: Gran cantidad de datos que se procesan, almacenan y calculan es enorme.
- Velocidad: Procesamiento y acceso rápido a los datos.
- Valor: La gran cantidad de datos proveerá datos válidos e inválidos, la idea es agrupar los datos para crear información y a su vez conocimiento.

- Variabilidad: Se refiere a la diferencia a los tipos y fuente de donde pueden provenir los datos o a su vez la velocidad en que se cargan las diferentes bases de datos.

2.2.4.3. Data Warehouse

- En su artículo Wu et al. [45], definen a un Data Warehouse como “Como un sistema orientado al tema, integrado, variable en el tiempo y de recopilación no volátil de datos que da apoyo al proceso de toma de decisiones de la dirección”. También menciona que las herramientas para su desarrollo y su arquitectura son determinados por la demanda del sistema.

En su artículo [46], realiza una comparación entre Data Warehouse tradicionales y Modernos, entre los puntos resaltables:

Tabla 1: Comparativa entre Data Warehouse

	Data Warehouse tradicionales	Data Warehouse modernos
Datos	Fuentes de datos son bases de datos transaccionales y operacionales	Fuentes de datos y pueden manejar diferentes tipos de datos (sensores, blog, videos y audio).
Arquitectura	Arquitectura orientada a procesos ETL	La arquitectura varía basándose en el problema, arquitectura propietaria u orientada a producto
Tecnología	Herramientas maduras y probadas en gran cantidad de aplicaciones, puede ser software libre o con licencia.	Mayormente, usa tecnología en crecimiento como Hadoop, software libre y procesamiento de conjunto junto con datos de big data.

Fuente: Elaboración propia

2.2.5. AngulaJS

AngulasJS es un framework desarrollado por Google de código abierto diseñado para la creación de SPA (Single Page Application) o aplicaciones de una sola página que permite un desarrollo modular gracias a su modelo MVC que asegura un entorno de desarrollo sostenible, rápido y escalable [47].

2.2.6. Node.js

“Node.JS es el framework basado en JavaScript creado en la parte superior del motor Google Chrome JavaScript V8, con el objetivo de permitir la ejecución de código js fuera del navegador, lo que proporciona el uso de JavaScript para el desarrollo tanto de front-end y back-end” [48].

2.2.7. Expres.js

Express.js es una infraestructura de desarrollo Back-End de aplicaciones Web en Node.js que facilita herramientas de ayuda para creación de aplicaciones Web y API en menor tiempo y proporciona un conjunto de funcionalidades como enrutamiento, gestión de secciones entre otras [49], [50].

2.2.8. Grafana

“Grafana es una conocida aplicación web de código abierto multiplataforma que permita la visualización de datos de forma interactiva y analítica. Provee cuadros, gráficos y alertas para web, es un componente muy popular en pilas de monitoreo y a menudo se utiliza junto con bases de datos de series temporales y plataformas de monitoreo” [51].

2.2.9. Revisión sistemática de la literatura

Se realizó una revisión sistemática de Indicadores claves de desempeño orientados a Smart Campus y Smart building extrayendo los KPI's relevantes y tengan correlación con la investigación realizada en el presente artículo.

Tabla 2: Revisión sistemática de la literatura de KPI's

Tema	KPI's	Unidad	Referencia
A Novel Energy Management Approach for Smart Home Using Bluetooth Low Energy	Hora de encendido (T inicio)		[30]
	Consumo	KWh	
	Valor umbral de poder (Phtr)		
	Valor umbral de retraso (Dthr)		
Implementación de indicadores energéticos en centros educativos. Caso de estudio: Edificio Alejandro Suárez Copete- Universidad Distrital Francisco José de Caldas	Índice de consumo energético por área total	kWh/mes – m ²	[52]
	Índice de potencia instalada por área total	W/m ²	
	Índice de consumo energético por persona	kWh/mes – persona	
	Índice de potencia instalada por persona	W/persona	
	Índice de emisiones de CO ₂ por área de edificio	kg CO ₂ /mes – m ²	
	Índice de emisiones de CO ₂ por persona	Kg CO ₂ /mes – personas	
An energy monitoring and management system based on key performace indicators	Asignación y desarrollo del consumo (KPIloadcons)		[53]
	Consumo total (KPItotcons)		
	Asignación de costos y desarrollo (KPIloadcost)		
	Costo total (KPItotcost)		
	Consumo máximo (Maximun _{consumption_load})		
	Consumo total máximo (Maximun_consumption_distribution)		
	Energía usada (KPIEnergyUse)		
	Curva de duración de carga (KPIReference _{load_Consumption})		
	Distribución horaria (KPILoad_curve)		
A Methodology for Determination and	Pérdidas de energía	kWh / año	[54]
	Variaciones de voltaje	%	

Definition of Key Performance Indicators for Smart Grids Development in Island Energy Systems	Desequilibrio del sistema de voltaje trifásico	%	
	Control de frecuencia	%	
Smart building features and key performance indicators: A review	Demanda de energía y Consumo	(kWh/(m ² .mes o año))	[55]
	Factor de cobertura de carga	%	
	Respuesta de demanda	kWh	
	Reducción de pérdidas de la red técnica	%	
Establishment of key Performance Indicators for Green Building Operations Monitoring – An application to china Case Study	Consumo de energía del sistema de iluminación por área del edificio	kWh/m ²	[56]
	Consumo de energía del sistema de aire acondicionado por área del edificio	kWh/m ²	

Fuente: Elaboración propia

2.3. OBJETIVOS DEL PROTOTIPO

2.3.1. Objetivo General

- Implementar un sistema web que permita el monitoreo de indicadores clave de desempeño de gestión energética.

2.3.2. Objetivos específicos

- Obtener estadísticas del consumo de las diferentes variables en periodos de tiempo determinados.
- Presentar gráficos estadísticos en una interfaz web gráfica amigable.
- Desarrollar un sistema de generación de datos que simule el consumo energético habitual de un campus universitario.
- Generar reportes con datos estadísticos de las diferentes variables de gestión energética.

2.4. DISEÑO DEL PROTOTIPO

Para realizar el diseño del dashboard o cuadro de mando se utilizan herramientas de inteligencia de negocio que representan de forma visual los

KPI's o métricas que se utilizan, por lo tanto, es de vital importancia tener los KPI's bien definidos. A continuación, describiremos los KPI's a emplear en el presente trabajo, sus unidades de medidas y la manera en que se obtienen.

Tabla 3: KPI's del sistema

KPI's	Unidad de medida	Forma de obtención
Consumo	KWh	Medidor ABB M2M
Áreas de consumo	m ²	Planos y mediciones de laboratorios y sitios de interés
Índice de consumo energético por área total	kWh/mes – m ²	KPI's Consumo, Área de consumo y Tiempo
Cantidad de ocupantes en área de interés	Personas (p)	Sistema de inteligencia artificial que detecta rostros.
Demanda de energía y Consumo	(kWh/(m ² . mes o año))	KPI's Consumo, Área de consumo y Tiempo
Voltaje	Voltaje consumido por Fase (V)	Medidor ABB M2M
Intensidad	Intensidad consumida por Fase (I)	Medidor ABB M2M
Energía	Consumo de energía reactiva (ae)	Medidor ABB M2M
Potencia	Potencia consumida por Fase (P)	Medidor ABB M2M

Fuente: Elaboración propia

Una vez definido los KPI's se usará Grafana como herramienta de visualización de datos. Grafana permite generas script para incrustar los gráficos en páginas web externas. En front-end se desarrollará con el framework de javascript AngularJs para ejecutar las funcionalidades del sistema, HTML para el maquetado y css para los estilos y presentación de la web.

Los datos que usará el sistema serán provistos por una API desarrollada en Node.js como back-end y como fuente de datos se empleará PostgreSQL.

El presente proyecto se realizará siguiendo la metodología de desarrollo ágil “SWIRL”, por consiguiente, se detalla el ciclo de vida del proyecto y se desglosa cada punto desarrollado.

2.4.1. Fase de análisis

2.4.1.1. Alcance del proyecto

El alcance del proyecto a grandes rasgos puede ser definido según lo siguiente:

- Obtener estadísticas del consumo de las diferentes variables eléctricas a nivel del campus por planta de facultad en periodos de tiempo.
- Presentar los gráficos estadísticos en una interfaz web gráfica amigable.
- Crear un sistema de simulación que asemeje los valores reales medidos por un medidor ABB M2M.
- Realizar un informe que provea los máximos, mínimos y promedios de los valores medidos en un periodo de tiempo determinado en formato pdf.

2.4.1.2. Análisis de Factibilidad

2.4.1.2.1. Factibilidad operativa

La factibilidad operativa es esencial especificarla en el proyecto debido a que nos permite constatar si se dispone de todos los recursos esenciales para desplegar el sistema web, a continuación, se detallan los recursos necesarios.

Tabla 4: Factibilidad Operativa

Factibilidad operativa		
Número	Actividad	Priorización
1	Servidor para desarrollo Front-End	Alta
2	Servidor para pruebas Back-End	Alta

Fuente: Elaboración propia

2.4.1.2.2. Factibilidad técnica

La factibilidad técnica nos permite especificar y reconocer los recursos necesarios para el desarrollo y funcionamiento del programa, por ello es importante en la siguiente tabla se detallan los recursos de hardware y software utilizados.

Tabla 5: Factibilidad Técnica

Factibilidad Técnica			
Recurso de Hardware		Recurso de software	
1	Laptop AMD A8-7410 APU with AMD Radeon R5 Graphics	1	Visual Studio Code
		1	Base de datos PostgreSQL
		1	Node.js
		1	Sistema de control de versiones Git y Github
		1	Dash plotly
		1	Grafana
		1	Angular

Fuente: Elaboración propia

2.4.1.2.3. Factibilidad económica

Para el proyecto de propuesta tecnológica desarrollado como trabajo de titulación no se consideran costos como los de programación, análisis, pruebas o configuración debido a que estos corren por cuenta propia, por ellos se detallan solamente hardware y software, al igual que servicios utilizados en el desarrollo del proyecto.

Tabla 6: Factibilidad Económica

Factibilidad Económica					
Costos de Inversión					
Tipo / Concepto	Parámetro / Descripción	Cantidad	Unid.Med.	P.Unitario (\$)	Total (\$)
Software					0
Visual Studio Code	Editor de código fuente	1	U	0	0

Dash Plotly	Dash es un framework de javascript que está pensado para construir gráficas en aplicaciones web	1	U	0	0
Git	Sistema de control de versiones para código	1	U	0	0
PgAdmin 4	Administrador de Base de datos	1	U	0	0
Grafana	Visualizador de datos	1	U	0	0
Angular	Framework para aplicaciones web	1	U	0	0
NodeJS	Entorno de ejecución para la capa del servidor	1	U	0	0
Hardware					300
Laptop AMD A8-7410 APU with AMD Radeon R5 Graphics	Equipo para el desarrollo y pruebas	1	U	300	300
Servicios					0
Servidor local	Servidor para realizar pruebas	2	Mes	0	0
GitHub	Sistema de control de versiones para código en la nube	1	Mes	0	0
				Total	300

Fuente: Elaboración propia

2.4.1.3. Identificación de interesados o “stakeholder”

Los interesados del proyecto representan a todo individuo que esté involucrado con el desarrollo del proyecto.

Tabla 7: Stakeholder

N°	Nombre	Cargo	Origen	Rol en el Proyecto	Puntos de vista	Responsabilidades
1	Pedro Cruz	Desarrollador	Interno	Programador	Duración del proceso y uso eficiente de recursos	-Diseñar, Planificar, Implementar y probar el sistema
2	Ing. Bertha Mazon	Consultor	Interno	Consultor	Eficiencia de la organización en la adaptación a cambios de condiciones	-Obtener y compartir conocimiento -Disponibilidad de información
3	Ing. Dixys Hernandez	Director de proyecto	Interno	Project Manager	Eficiencia de la organización en la adaptación a cambios de condiciones	-Establecer fechas de entregables y proponer estrategias de desarrollo del proyecto -Obtener y compartir conocimiento -Disponibilidad de información
4	Ing. Johnny Novillo	Consultor	Interno	Consultor	Eficiencia de la organización en la adaptación a cambios de condiciones	-Obtener y compartir conocimiento -Disponibilidad de información

Fuente: Elaboración propia

2.4.1.4. Requerimientos

2.4.1.4.1. Requerimientos funcionales

Los requerimientos funcionales establecen las condiciones y capacidades que se deben implementar dentro del proyecto para cumplir con las expectativas de todos los involucrados.

Tabla 8: Requerimientos Funcionales

Requerimientos Funcionales (RF)	
Código	Descripción
RF_01	Autenticación de usuarios
RF_02	Embeber gráficos estadísticos de Grafana
RF_03	Crear gráficos estadísticos a partir de data histórica
RF_04	Filtrar los gráficos estadísticos por medidor y por tiempo (día, mes y año)
RF_05	Operaciones CRUD de los datos utilizados para la simulación cargas
RF_06	Creación de reporte en PDF

Fuente: Elaboración propia

2.4.1.4.2. Requerimientos no funcionales

Los requisitos no funcionales no intervienen con el funcionamiento del sistema, se enfocan en factores como usabilidad, escalabilidad, seguridad entre otros puntos de importantes.

Tabla 9: Requerimientos no Funcionales

Requerimientos no Funcionales (RNF)	
Código	Descripción
RNF_01	Base de datos estructurada
RNF_02	Interfaz gráfica amigable
RNF_03	Facilidad de uso y entendimiento
RNF_04	Desarrollo modular y fácil de mantener o actualizar

Fuente: Elaboración propia

2.4.2. Fase de planificación

2.4.2.1. Historia de usuarios

Las historias de usuario se basan en los requerimientos presentados en puntos anteriores y permiten dar una explicación general e informal del funcionamiento del sistema.

Tabla 10: Historia de usuario N1 - Ingreso al Sistema

Historia de Usuario	
Número: 1	Usuario: Usuario
Nombre de historia: Solicitar información del usuario para Login	

Prioridad: Alta	Riesgo de desarrollo: Alta
Requerimiento funcional: RF-01	
Programador Responsable: Cruz Zambrano Pedro Ricardo	
Descripción: Los usuarios deberán ingresar su usuario y contraseña para poder acceder al sistema	
Observaciones: Mostrar mensaje de ingreso satisfactorio o de error según el caso	

Fuente: Elaboración propia

El sistema debe permitir la autenticación de usuarios para acceder a las funcionalidades por cuestiones de seguridad.

Tabla 11: Historia de usuario N2 - Visualización de datos Grafana

Historia de Usuario	
Número: 2	Usuario: Usuario
Nombre de historia: Visualizar gráficos estadísticos de Grafana	
Prioridad: Alta	Riesgo de desarrollo: Alta
Requerimiento funcional: RF-02	
Programador Responsable: Cruz Zambrano Pedro Ricardo	
Descripción: El sistema debe permitir presentar gráficos estadísticos en tiempo real generados por Grafana	
Observaciones: Se debe preparar el HTML para embeber los gráficos externos.	

Fuente: Elaboración propia

El sistema debe soportar embeber en HTML los gráficos generados en tiempo real por el software libre "Grafana" del mismo modo debe estructurar y dar estilos para una correcta presentación de los mismos.

Tabla 12: Historia de usuario N3 - Visualización de datos Histórico

Historia de Usuario	
Número: 3	Usuario: Usuario
Nombre de historia: Presentar gráficos estadísticos con data histórica	
Prioridad: Alta	Riesgo de desarrollo: Alta
Requerimiento funcional: RF-03	

Programador Responsable: Cruz Zambrano Pedro Ricardo
Descripción: El sistema debe permitir consultas a la base de datos para obtener datos y graficar el comportamiento de las variables de gestión energética en el tiempo
Observaciones: Se debe utilizar un generador de gráficos dinámico.

Fuente: Elaboración propia

El sistema generará gráficos estadísticos dinámicos que permita al usuario moverse dentro de la gráfica, ajustar la escala y filtrar por variable en caso de tener más de una variable por gráfica.

Tabla 13: Historia de usuario N4 - Filtro de Gráficos

Historia de Usuario	
Número: 4	Usuario: Usuario
Nombre de historia: Presentar gráficas estadísticas por fecha y medidor.	
Prioridad: Alta	Riesgo de desarrollo: Alta
Requerimiento funcional: RF-04	
Programador Responsable: Cruz Zambrano Pedro Ricardo	
Descripción: El sistema debe permitir presentar gráficos estadísticos históricos por lapsos de tiempo definidos.	
Observaciones: Se implementará escalas de tiempo definidas	

Fuente: Elaboración propia

El sistema debe permitir el presentar gráficos con data de día, mes y año y permitir al usuario navegar por el tiempo en la gráfica, al igual que poder seleccionar el medidor del cual se desea ver los datos.

Tabla 14: Historia de Usuario - Datos de Simulación

Historia de Usuario	
Número: 5	Usuario: Usuario
Nombre de historia: CRUD datos de simulación	
Prioridad: Alta	Riesgo de desarrollo: Alta
Requerimiento funcional: RF-05	
Programador Responsable: Cruz Zambrano Pedro Ricardo	

Descripción: El sistema debe permitir realizar las operaciones CRUD de los datos empleados para la simulación del servidor.
Observaciones: Se implementará una tabla para visualizar los datos a modificar o eliminar.

Fuente: Elaboración propia

El sistema debe implementar un módulo que permita visualizar los datos utilizados para la simulación de cargas, de igual forma, debe permitir actualizar, agregar o eliminar dichos datos.

Tabla 15: Historia de usuario N6 - Reportes

Historia de Usuario	
Número: 6	Usuario: Usuario
Nombre de historia: Crear Reportes	
Prioridad: Alta	Riesgo de desarrollo: Alta
Requerimiento funcional: RF-06	
Programador Responsable: Cruz Zambrano Pedro Ricardo	
Descripción: El sistema debe permitir realizar generar reportes en pdf descargables	
Observaciones: Se implementará por medidor y por fecha.	

Fuente: Elaboración propia

El sistema debe crear reportes en PDF descargables según el usuario necesite, estos serán de datos históricos y se filtrará por medidor y por fecha, el reporte constará de los datos máximos, mínimos y promedio de las diferentes variables de gestión energética.

2.4.2.2. Gestión de cronograma

Tabla 16: Cronograma

EDT	Nombre de tarea	Duración	Comienzo	Fin
1	Monitoreo de Indicadores clave de desempeño de gestión energética en un	77 días	jue 4/11/21	vie 18/2/22

1.1	Análisis	14 días	jue 4/11/21	mar 23/11/21
1.1.1	Definición del sistema	4 días	jue 4/11/21	mar 9/11/21
1.1.2	Estudio de Factibilidad	3 días	mié 10/11/21	vie 12/11/21
1.1.3	Identificación de Riesgos	2 días	lun 15/11/21	mar 16/11/21
1.1.4	Requerimientos Funcionales y no funcionales	2 días	mié 17/11/21	jue 18/11/21
1.1.5	Redacción del capítulo I	3 días	vie 19/11/21	mar 23/11/21
1.2	Planificación	12 días	mié 24/11/21	jue 9/12/21
1.2.1	Gestión del cronograma	3 días	mié 24/11/21	vie 26/11/21
1.2.2	Historias de usuario	4 días	lun 29/11/21	jue 2/12/21
1.2.3	EDT	1 día	vie 3/12/21	vie 3/12/21
1.2.4	Redacción capítulo II	4 días	lun 6/12/21	jue 9/12/21
1.3	Modelado	8 días	vie 10/12/21	mar 21/12/21
1.3.1	Diseño de la base de datos relacional	2 días	vie 10/12/21	lun 13/12/21
1.3.2	Diseño navegacional	3 días	mar 14/12/21	jue 16/12/21
3.3	Diagrama de secuencia	3 días	vie 17/12/21	mar 21/12/21
1.4	Implementación	30 días	mié 22/12/21	mar 1/2/22
1.4.1	Diseño de interfaces de usuario	1 día	mié 22/12/21	mié 22/12/21
1.4.2	Codificación de funciones y componentes	29 días	jue 23/12/21	mar 1/2/22
1.4.2.1	Procesamiento de Datos	4 días	jue 23/12/21	mar 28/12/21

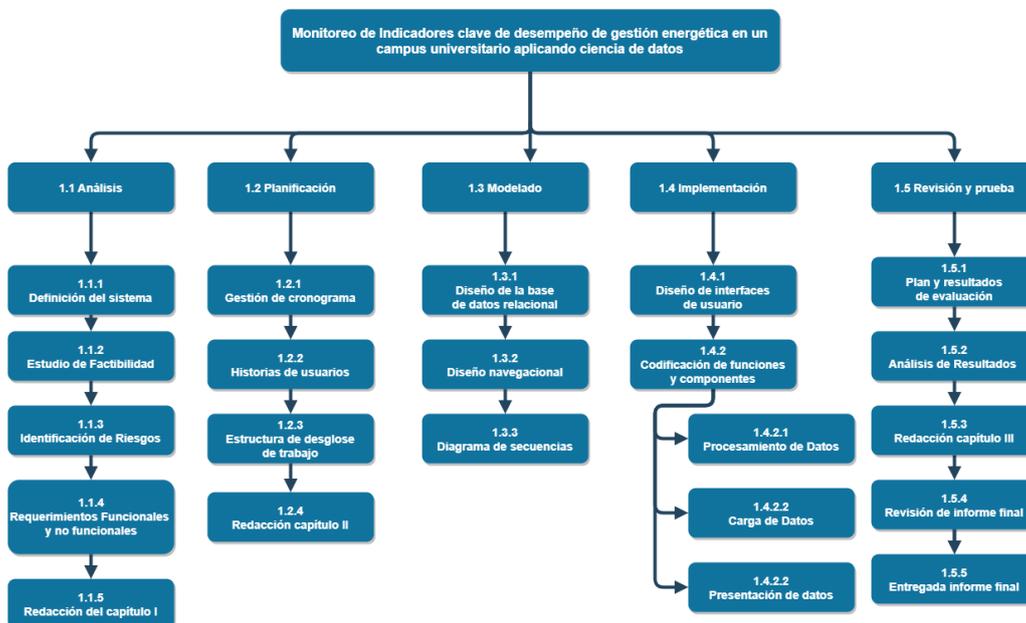
1.4.2.2	Carga de Datos	2 días	mié 29/12/21	jue 30/12/21
1.4.2.3	Presentación de datos	23 días	vie 31/12/21	mar 1/2/22
1.5	Revisión y pruebas	13 días	mié 2/2/22	vie 18/2/22
1.5.1	Plan y resultados de evaluación	1 día	mié 2/2/22	mié 2/2/22
1.5.2	Análisis de Resultados	2 días	jue 3/2/22	vie 4/2/22
1.5.3	Redacción capítulo III	6 días	lun 7/2/22	lun 14/2/22
1.5.4	Revisión de informe final	3 días	mar 15/2/22	jue 17/2/22
1.5.5	Entrega informe final	1 día	vie 18/2/22	vie 18/2/22

Fuente: Elaboración propia

2.4.2.3. Estructura de desglose de trabajo

El EDT o estructura de desglose de trabajo nos provee una visión clara y amplia de la estructura de nuestro proyecto.

Ilustración 5. EDT

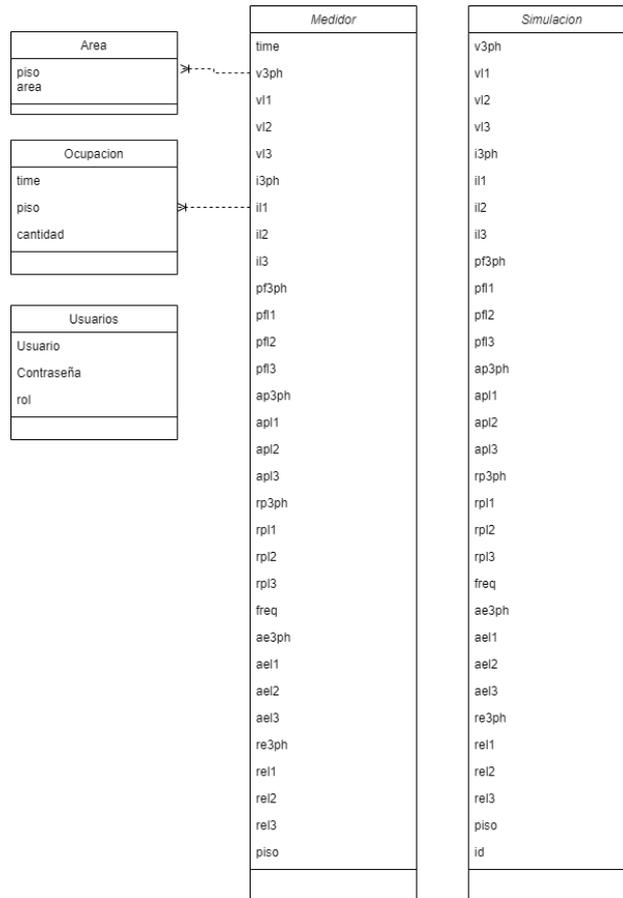


Fuente: Elaboración propia

2.4.3. Fase de modelado

2.4.3.1. Diseño de base de datos relacional

Ilustración 6. Diseño relacional de la base de datos



Fuente: Elaboración propia

El gestor de base de datos utilizado para el presente proyecto es PostgreSQL, se utilizan 4 tablas que se describirán a continuación.

Tabla 17: Estructura de tablas de la base de datos

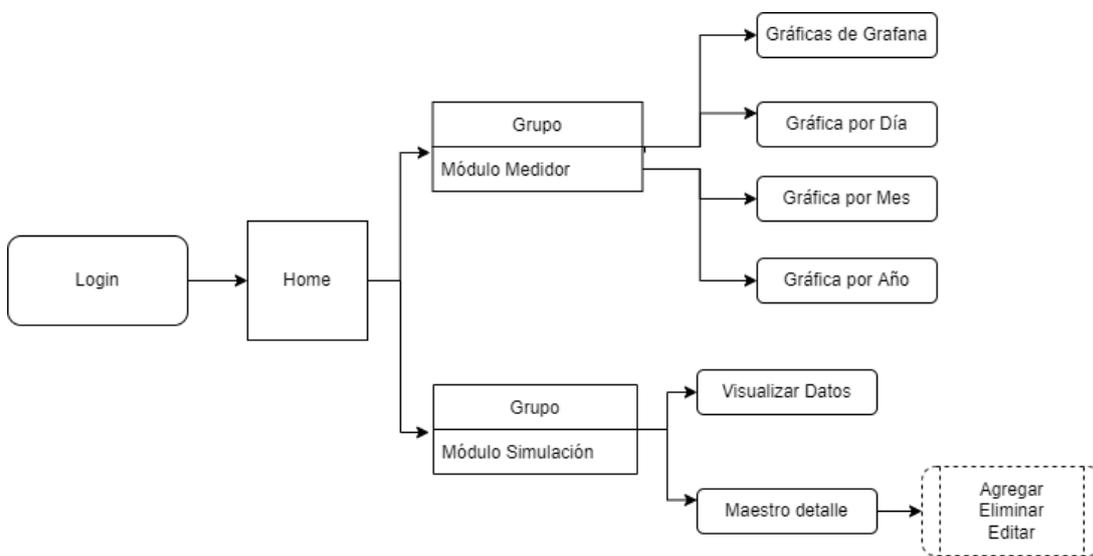
Tablas de la Base de Datos	
Tabla	Descripción
Tabla Area	Esta tabla almacena el valor en metros cuadrados de la superficie útil que abarca un medidor
Tabla Ocupacion	Esta tabla registra la cantidad de personas en un área en tiempo real, se almacena el número y el tiempo en el que se realiza la medición

Tabla Medidor	La tabla almacena todos los datos de utilidad que son proporcionados por medidores ABB M2M
Tabla Simulacion	Esta tabla un conjunto de datos utilizados para realizar la simulación de los medidores ABB M2M

Fuente: Elaboración propia

2.4.3.2. *Diseño navegacional*

Ilustración 7. *Diseño navegacional*



Fuente: Elaboración propia

En la Ilustración 6, se observa el flujo de interacción del usuario tendrá a través del contenido.

2.4.3.3. *Prototipado de interfaces de usuarios*

El prototipado de las interfaces permite conseguir una visión general de flujo gráfico que se presentará al usuario, para el diseño de nuestras interfaces se utilizó la página web “diagrams.net” que nos permite tener un flujo de trabajo eficiente.

Ilustración 8. Prototipo Login

Ilustración de un prototipo de formulario de inicio de sesión. El formulario tiene un título "Sign In" y dos campos de entrada: "User Name:" con el valor "johndoe" y "Password:" con caracteres ocultos por asteriscos. Un botón azul "SIGN IN" está ubicado en la parte inferior del formulario.

Fuente: Elaboración propia

Este formulario muestra la interfaz gráfica del login, su utilidad reside en autenticar a los usuarios para que accedan solo las personas autorizadas.

Ilustración 9. Prototipo Inicio

Ilustración de un prototipo de interfaz de inicio de una aplicación SPA. La interfaz muestra un navegador con la URL "https://www.draw.io". Hay una barra de pestañas con "Tab 1" (seleccionada), "Tab 2" y "Tab 3". El título de la pestaña es "Monitoreo de KPI de gestión energética". Hay un botón de simulación "ON". El contenido principal muestra "Bienvenida" y "Descripción del prototipo".

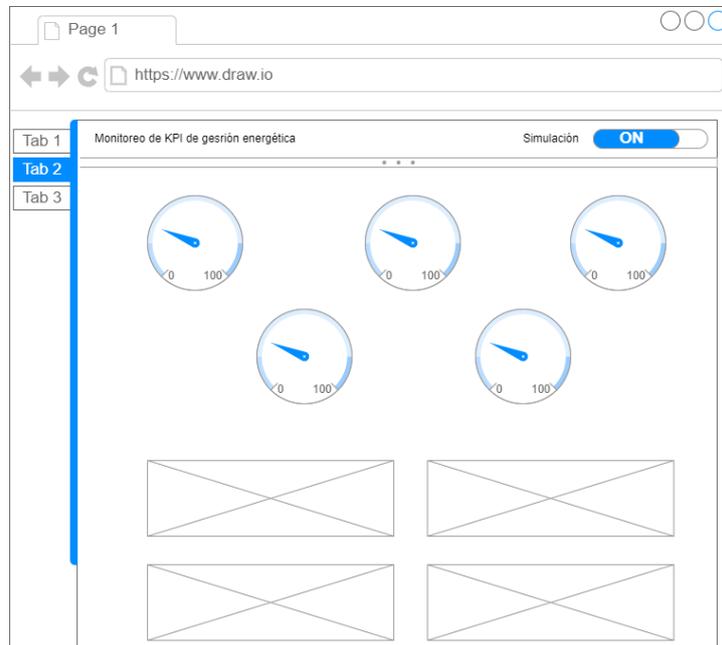
Fuente: Elaboración propia

Este formulario muestra una bienvenida y explica un poco de que trata el sistema, da pautas generales sobre su funcionamiento.

En los siguientes formularios al ser un SPA (Single Page Application) se realiza una reutilización de componentes, por lo tanto, su interfaz se asemeja

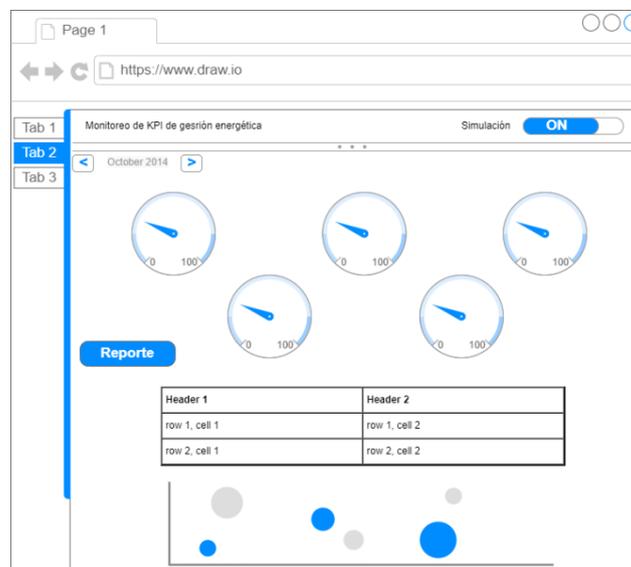
con pequeños cambios en el tipo de gráficos que muestran y el filtrado por fecha que varía entre Tiempo real, día, mes y año.

Ilustración 10. Prototipo Dashboard tiempo real



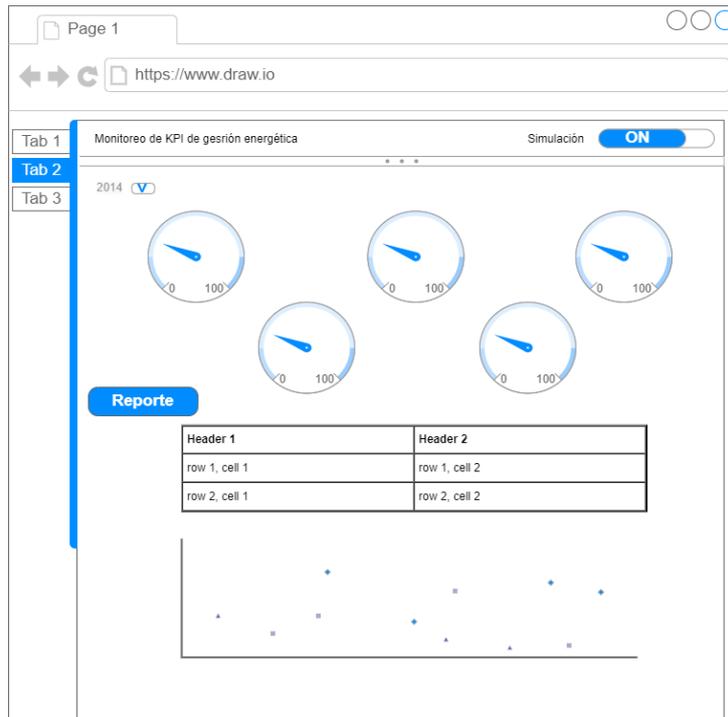
Fuente: Elaboración propia

Ilustración 11. Prototipo dashboard histórico por mes



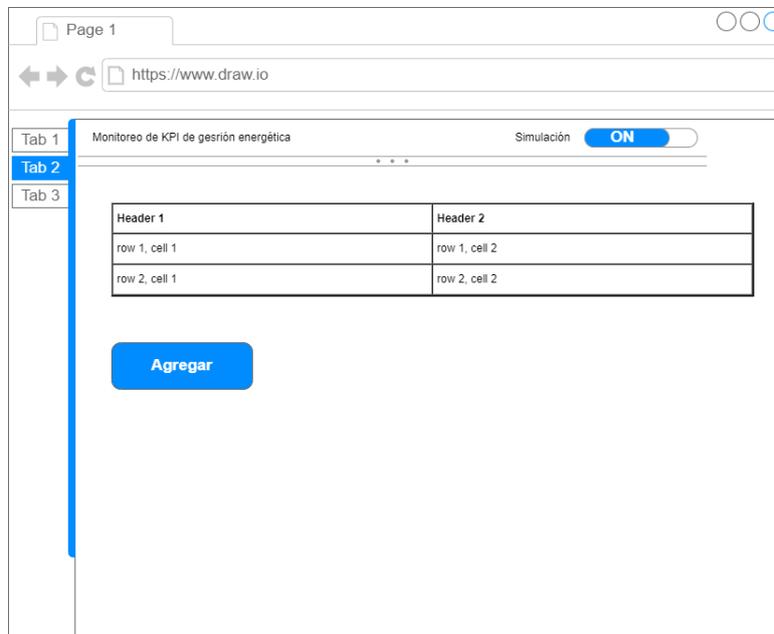
Fuente: Elaboración propia

Ilustración 12. Prototipo dashboard histórico por año



Fuente: Elaboración propia

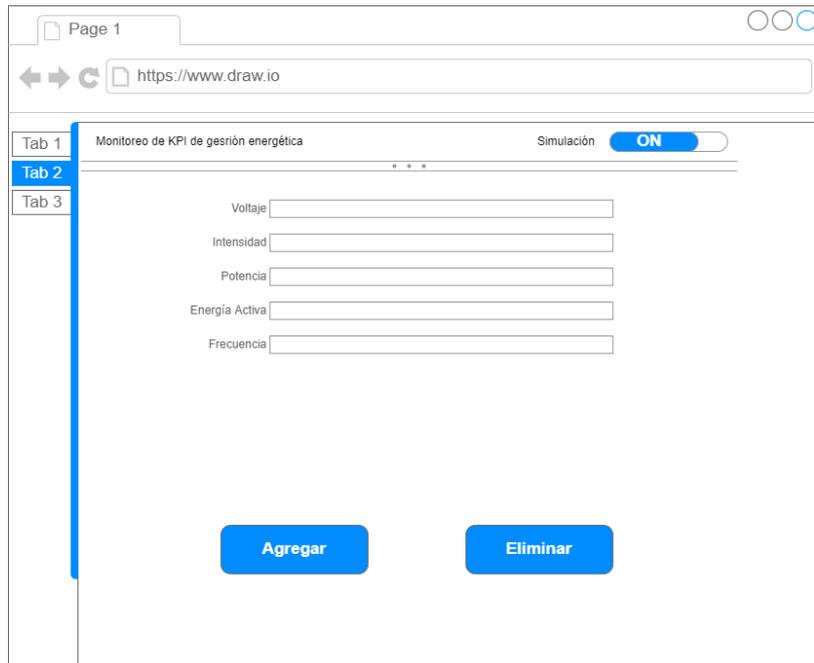
Ilustración 13. Prototipo visualización de datos de simulación



Fuente: Elaboración propia

Esta interfaz muestra los datos que se utilizan para la simulación del sistema, y proporciona la opción de editar o agregar datos de simulación.

Ilustración 14. Prototipo CRUD de Datos de simulación



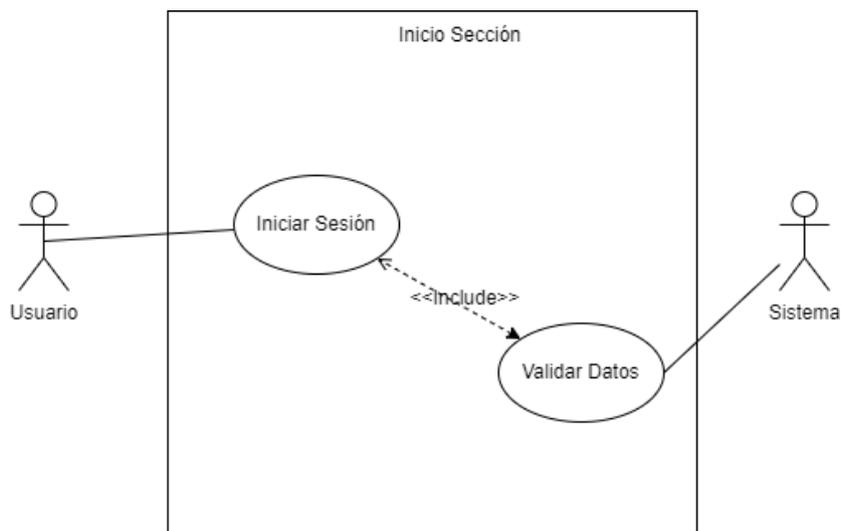
Fuente: Elaboración propia

Esta interfaz permite agregar, editar o eliminar los datos de simulación, en la opción de editar, los parámetros se cargan automáticamente al dar clic en un registro de la interfaz anterior.

2.4.3.4. Diagramas UML

2.4.3.4.1. Diagramas de casos de uso

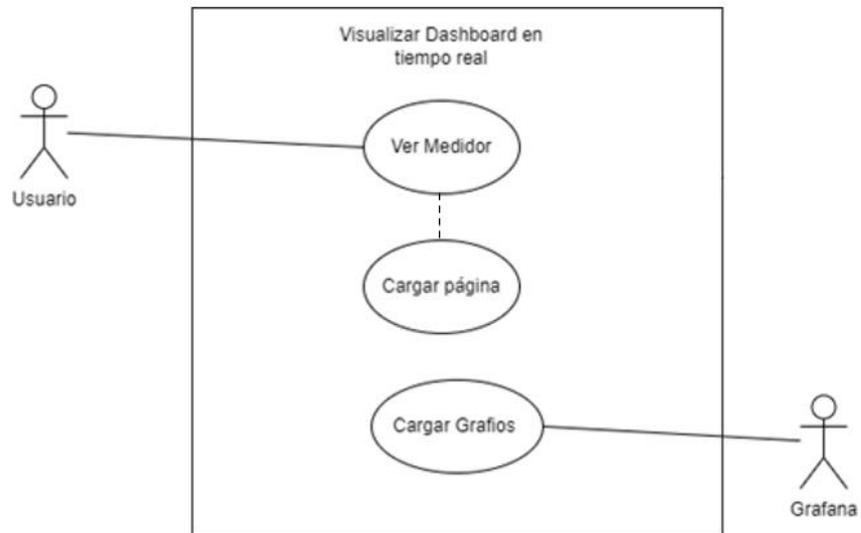
Ilustración 15. Caso de uso Inicio de sección



Fuente: Elaboración propia

El usuario ingresa sus credenciales, el sistema valida si los datos son correctos, autoriza o no el ingreso al sistema.

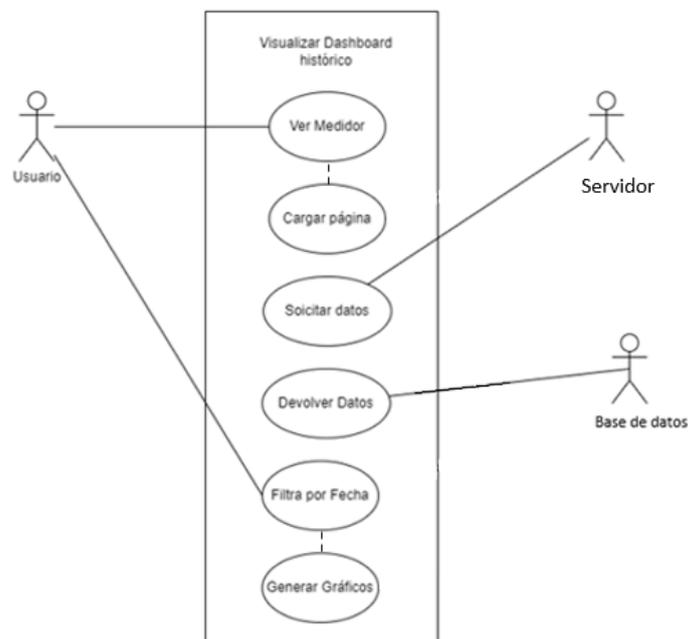
Ilustración 16. Caso de uso Visualizar dashboard en tiempo real



Fuente: Elaboración propia

El usuario escoge que medidor quiere observar, sistema renderiza la página y se carga los gráficos de Grafana.

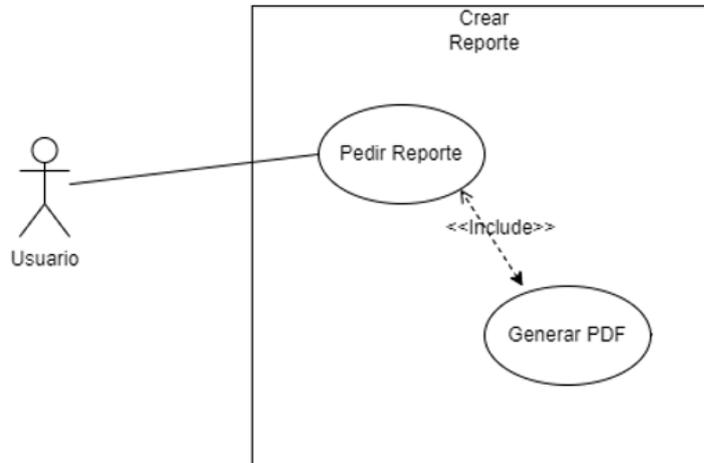
Ilustración 17. Caso de uso visualizar dashboard histórico



Fuente: Elaboración propia

El usuario escoge el medidor que quiere analizar, el sistema carga la página y solicita los datos al servidor, el servidor devuelve los datos necesarios y el usuario puede filtrar por fecha los gráficos, el sistema se encarga de generar los gráficos.

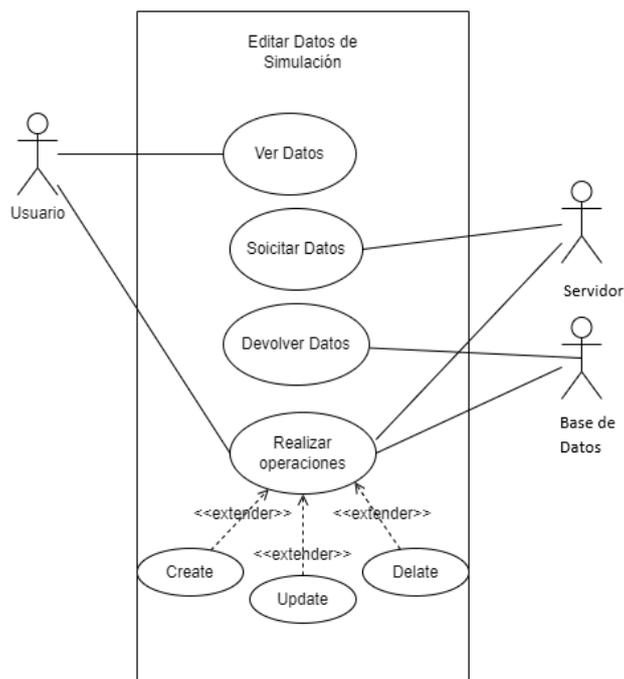
Ilustración 18. Caso de uso Crear reporte



Fuente: Elaboración propia

El usuario pide un reporte, el sistema con los datos ya cargados en la página genera un reporte en PDF que se descarga automáticamente.

Ilustración 19. Caso de uso Editar datos de simulación

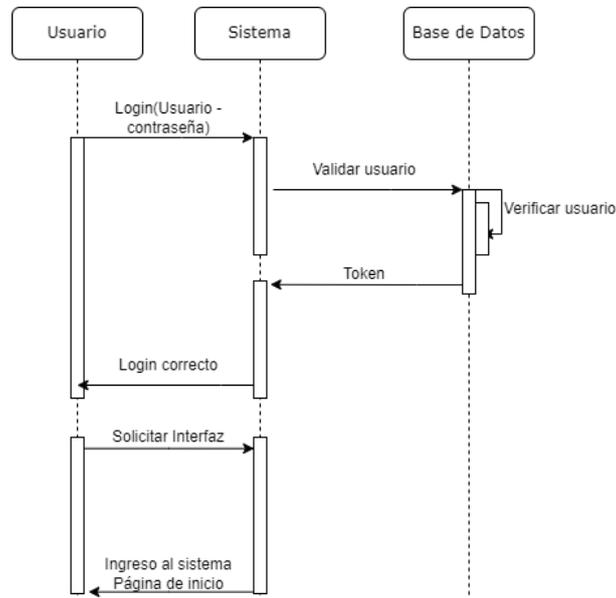


Fuente: Elaboración propia

El usuario pide visualizar los datos utilizados para la simulación, el sistema solicita los datos al servidor y este se los devuelve, el usuario puede añadir nuevos, actualizar los existentes o eliminar los datos visualizados.

2.4.3.4.2. Diagrama de secuencias

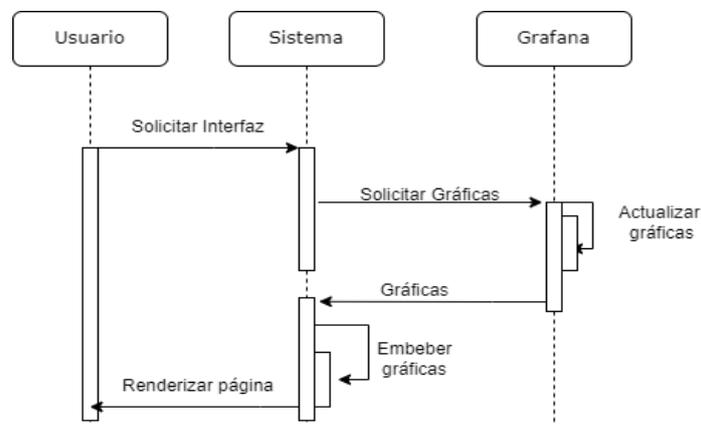
Ilustración 20. Diagrama de secuencia - Inicio de sesión



Fuente: Elaboración propia

El diagrama de secuencia nos muestra el proceso necesario para realizar el ingreso al sistema. El usuario ingresa sus credenciales, el sistema ejecuta una consulta a la base de datos para realizar una validación, si los datos existen se envía un token y se procede al sistema, este realiza el login, por último, el usuario solicita la interfaz de inicio y el sistema renderiza la página.

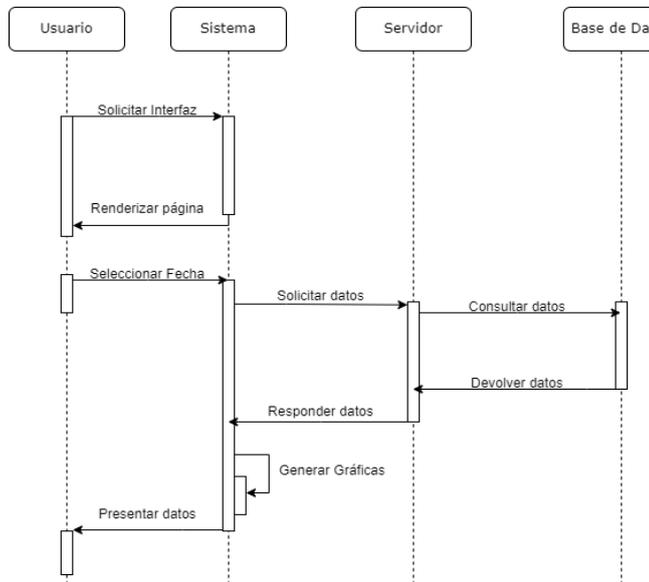
Ilustración 21. Diagrama de secuencia - Cargar Dashboard Grafana



Fuente: Elaboración propia

El usuario efectúa la solicitud de la interfaz para visualizar el dashboard en tiempo real, el sistema solicita los gráficos estadísticos a Grafana, Grafana actualiza los gráficos y los envía al sistema, el sistema carga los gráficos en el HTML, renderiza la página y la envía al usuario.

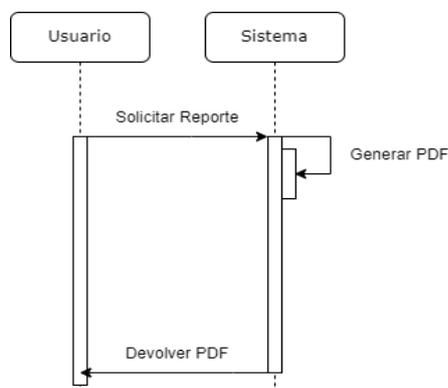
Ilustración 22. Diagrama de secuencia - Dashboard histórico



Fuente: Elaboración propia

El usuario solicita visualizar la interfaz gráfica correspondiente al medidor con data histórica. El sistema renderiza la página, el usuario escogerá el periodo de tiempo del que desea revisar datos, el sistema realiza una petición al servidor, esta consulta los datos en la base de datos y devuelve los mismos. El servidor prepara los datos y responde la solicitud del sistema, el sistema genera los gráficos y presenta los datos al usuario.

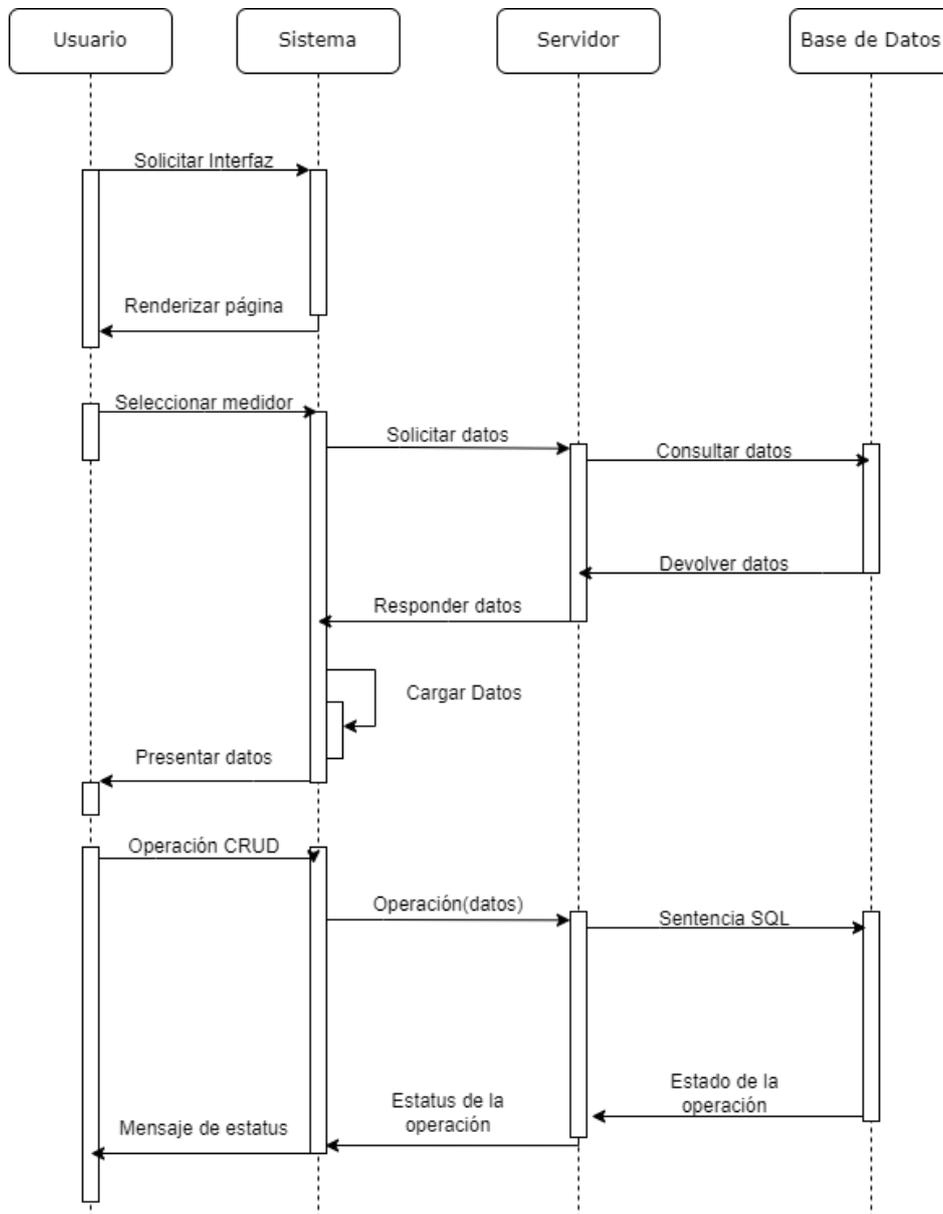
Ilustración 23. Diagrama de secuencia - Generar Reporte



Fuente: Elaboración propia

El usuario realiza la petición al sistema de generar un reporte, el sistema crear el reporte en formato PDF y lo envía al usuario listo para descargar.

Ilustración 24. Diagrama de secuencia - CRUD datos simulación



Fuente: Elaboración propia

El usuario ejecuta la petición de la interfaz correspondiente a los datos de simulación, el sistema carga la página y el usuario escoge el medidor que desea observar, el sistema efectúa una petición al servidor y este a la base de datos, la base de datos responde con los datos y son enviados al servidor y este a su vez los envía al sistema para ser mostrados al usuario, el usuario

elige si desea agregar, editar o eliminar algún registro y de ser el caso se realiza el proceso correspondiente.

2.5. Ejecución y/o ensamble del prototipo

2.5.1. Proceso ETL

En primera instancia se procedió a realizar un proceso de ETL con los datos, este proceso se llevó a cabo con Python y Google Colab, ya que se manejan volúmenes grandes de datos.

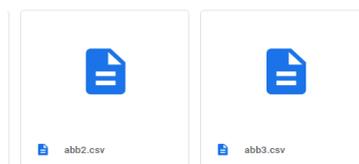
2.5.1.1. Extracción

El proceso de extracción fue facilitado por el docente encargado de administrar el servidor que almacena los datos

2.5.1.2. Transformación

Para la transformación se realizó la carga de los datos al drive.

Ilustración 25. Carga de datos en Drive



Fuente: Elaboración propia

Ilustración 26. Importación de librerías

```
[ ] from google.colab import drive
drive.mount('/content/gdrive')

Mounted at /content/gdrive

[ ] import pandas as pd
import numpy as np
import os, cv2
from numpy import array

[ ] PATH = os.getcwd()
df1 = pd.read_csv(PATH + '/gdrive/My Drive/dash/abb2.csv', )
```

Fuente: Elaboración propia

Se importó las librerías necesarias para el procesamiento de los datos

Ilustración 27. Eliminación de datos no útiles

```
df2 = df1.drop(['v112', 'v123', 'v131', 'cos3ph', 'cos11', 'cos12', 'cos13', 'sp3ph', 'sp11', 'sp12', 'sp13', 'sel1', 'sel2', 'sel3' ], axis=1)
df2
```

Fuente: Elaboración propia

Luego procedió a eliminar los datos que no son de interés para la investigación y de esta forma reducir el volumen de datos.

Ilustración 28. Agregación de datos de identificador

```
df4 = df2
df4.insert(16, 'piso', 3)
df4
```

Fuente: Elaboración propia

Se realizó una agregación de parámetros para una mejor manipulación de los mismos.

Ilustración 29. Reorganización de datos

```
df5 = df4[['date', 'v3ph', 'v11', 'v12', 'v13', 'i3ph', 'i11', 'i12', 'i13', 'pf3ph', 'pf11',
          'pf12', 'pf13', 'ap3ph', 'ap11', 'ap12', 'ap13', 'rp3ph', 'rp11', 'rp12', 'rp13', 'freq', 'ae3ph',
          'ae11', 'ae12', 'ae13', 're3ph', 'rel1', 'rel2', 'rel3', 'piso']]
```

Fuente: Elaboración propia

Se procedió a la reorganización de los datos para que coincidan con el modelo de base de datos como se puede observar en la ilustración #6.

Ilustración 30. Formateo de Fecha

```
h = df4
h['date'] = h['date'].astype(str)
h.dtypes
h['date'] = ''+h["date"]+''
h
```

Fuente: Elaboración propia

Se realizó un formateo de datos correspondiente a las variables de fecha para el proceso de carga al gestor de base de datos.

Ilustración 31. Guardado de datos

```
df5.to_csv(PATH + '/gdrive/My Drive/dash/ETLab1.csv', index = False)
```

Fuente: Elaboración propia

Por último, se guardaron los datos en formato csv, listos para el proceso de carga a la base de datos

Febrero - 2018	27912	5316.00	22122	3167	2623	79.3%	11.3%	9.4%
Marzo - 2018	21224	3259.00	16517	2332	2375	77.8%	11.0%	11.2%
Abril - 2018	8067	947.00	5096	923	2048	63.2%	11.4%	25.4%
Mayo - 2018	14193	1410.00	10975	1050	2168	77.3%	7.4%	15.3%
Junio - 2018	23880	5091.00	19490	2343	2047	81.6%	9.8%	8.6%
Julio - 2018	21258	4240.00	16742	2258	2258	78.8%	10.6%	10.6%
Promedio	19422.3 3	3377.17	15157. 00	2012.1 7	2253.1 7			

Fuente: Elaboración propia

Tabla 19: Cálculo de consumo

	Activa	Reactiva
Consumo mes	19422.333	3377.167
Consumo día	647.411	112.572
Consumo hora	26.975	4.691
Consumo minuto	0.450	0.078

Fuente: Elaboración propia

Luego se calculó el porcentaje de consumo que registra cada medidor instalado, por consiguiente, encontramos el porcentaje de consumo que corresponde a los pisos que no poseen un medidor, como podemos observar en la ilustración #4. Equivale al 67% que corresponden al área administrativa (planta baja) y a la sala de profesores (piso 1).

Tabla 20: Porcentaje de Consumo por medidor

Medidor	Medición	Porcentaje (%)
General	293.9976	100

Medidor 2	62.6632	21.314187 6
Medidor 3	32.0874	10.914170 7

Fuente: Elaboración propia

Se calculó el porcentaje de consumo por línea eléctrica.

Tabla 21: Cálculo de porcentaje por línea de carga

Medidas	Energía Activa				Energía Reactiva			
	Trifásico	Línea 1	Línea 2	Línea 3	Trifásico	Línea 1	Línea 2	Línea 3
	293.90	95.38	108.28	90.49	98	29.73	41	27
Porcentaje	100.00	32.45	36.84	30.79	100	30.19	41.95	27.86

Fuente: Elaboración propia

Con estos datos y el muestreo realizado en la ilustración #33 se procedió a generar los datos que se utilizaron para la simulación.

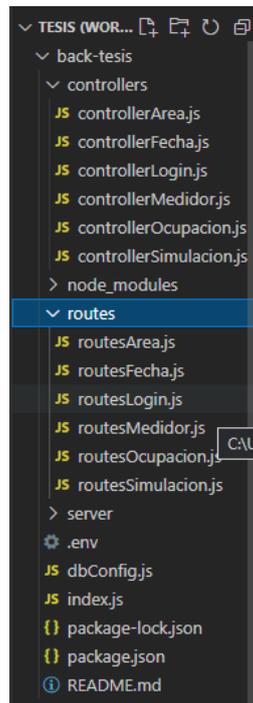
Ilustración 34. Datos correspondientes a la simulación

rpl1 double precision	rpl2 double precision	rpl3 double precision	freq double precision	ae3ph double precision	ael1 double precision	ael2 double precision	ael3 double precision	re3ph double precision
-2.04	-2.4	1.8	60	0.2247955	0.0729518	0.0828208	0.0692147	0.0390876
-2.04	-2.4	1.8	60.04	0.3371933	0.1094278	0.1242312	0.103822	0.0586314
-2.64	-2.64	1.68	59.97	0.449591	0.1459037	0.1656416	0.1384294	0.0781752
-2.76	3	1.8	60.02	0.5619888	0.1823796	0.2070521	0.1730367	0.0977189
-3.36	3.72	1.68	60.04	0.6743866	0.2188555	0.2484625	0.2076441	0.1172627
-2.88	3.72	1.68	60.04	0.2247955	0.0729518	0.0828208	0.0692147	0.0390876
-2.16	-2.76	-2.28	60.03	0.3371933	0.1094278	0.1242312	0.103822	0.0586314
-2.28	-2.76	-3	60.06	0.449591	0.1459037	0.1656416	0.1384294	0.0781752

Fuente: Elaboración propia

2.5.3. Desarrollo del Back-End

Ilustración 35. Estructura del Back-End



Fuente: Elaboración propia

Ilustración 36. Configuración del servidor Back-end

```
JS index.js X
back-tesis > JS index.js > ...
1  const express = require('express');
2  const conexion_pg = require('./dbConfig');
3  const cors = require('cors');
4
5  const app = express();
6
7  app.use(cors());
8
9  app.use(express.json());
10 app.use(express.urlencoded({extended: false}));
11
12 conexion_pg.connect();
13
14 app.use(require('./routes/routesArea'));
15 app.use(require('./routes/routesLogin'));
16 app.use(require('./routes/routesMedidor'));
17 app.use(require('./routes/routesOcupacion'));
18 app.use(require('./routes/routesFecha'));
19 app.use(require('./routes/routesSimulacion'));
20
21 var port=process.env.PORT || 4000;
22 app.listen(port);
```

Fuente: Elaboración propia

Ilustración 37. Configuración de conexión a la base de Datos

```
require("dotenv").config();

const { Pool, Connection } = require("pg");

const config = {
  user: process.env.DB_USER,
  password: process.env.DB_PASSWORD,
  database: process.env.DB_DATABASE,
  host: process.env.DB_HOST,
  port: process.env.DB_PORT,
  ssl: false
}

const conexion_pg = new Pool(config);

conexion_pg.connect()
  .then(db_pg => console.log('Conexión exitosa con base de datos PostgreSQL'))
  .catch(err => console.log('Error al conectar con base de datos PostgreSQL ' + err));

module.exports = conexion_pg;
```

Fuente: Elaboración propia

Ilustración 38. Codificación de rutas del servidor

```
const { Router } = require('express');
const router = Router();

const { getMedidor, createMedidor, createSimulacion, getMedidorDiagrafica,
  getMedidorDiakw, getMedidorDiaOcupacion, getMedidorDiaSuperficie, getMedidorMesgrafica,
  getMedidorAniografica, getMedidorAnioDatos, getMedidorAnioOcupacion } = require('../controllers/controllerMedidor');

router.get('/medidor/optener', getMedidor );
router.post('/medidor/ingresar', createMedidor );
router.get('/medidor/simular/:id', createSimulacion );
router.get('/medidor/dia/:id', getMedidorDiagrafica );
router.get('/medidor/diakw/:id', getMedidorDiakw );
router.get('/medidor/diakwocupacion/:id', getMedidorDiaOcupacion );
router.get('/medidor/diakwosuperf/:id', getMedidorDiaSuperficie );
router.get('/medidor/mes/:id', getMedidorMesgrafica );
router.get('/medidor/anio/:id', getMedidorAniografica );
router.get('/medidor/aniodatos/:id', getMedidorAnioDatos );
router.get('/medidor/anioocupacion/:id', getMedidorAnioOcupacion );

module.exports = router;
```

Fuente: Elaboración propia

Ilustración 39. Codificación de controller del servidor

```
const getMedidorDiaOcupacion = async(req, res) => {
  fInicio = req.params.id.split('n')[0];
  let piso = req.params.id.split('n')[1];
  const response = await conn.query(`SELECT MAX(num) FROM ocupacion
  where "time" between $1 and $2 and piso = $3`,
  [fInicio, fInicio + " 23:59:59", piso]);
  res.status(200).json(response.rows);
}

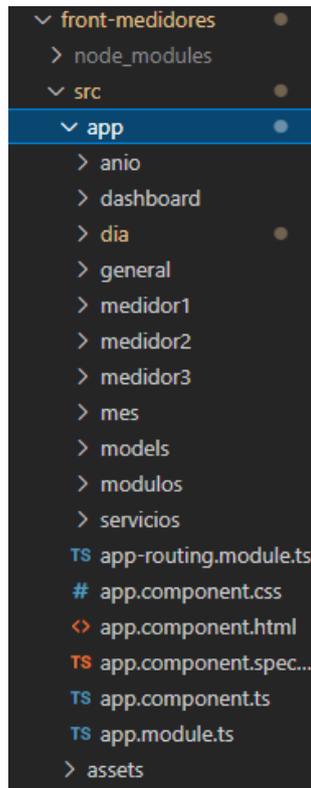
const getMedidorDiaSuperficie = async(req, res) => {
  const response = await conn.query(`select metros from superficie where piso = $1`,
  [req.params.id]);
  res.status(200).json(response.rows);
}

// graficos en mes
const getMedidorMesgrafica = async(req, res) => {
  fInicio = req.params.id.split('n')[0];
  fFinal = req.params.id.split('n')[1];
  let piso = req.params.id.split('n')[2];
  const response = await conn.query(`select to_char("time", 'YYYY/MM/DD'), v3ph, i3ph, pf3ph
  from public.medi where "time" between $1 and $2 and piso = $3;`,
  [fInicio, fFinal, piso]);
  res.status(200).json(response.rows);
}
```

Fuente: Elaboración propia

2.5.4. Desarrollo Front-End

Ilustración 40. Estructura del Front-End



Fuente: Elaboración propia

Ilustración 41. Maquetado del menú

```
dashboard.component.html X
front-medidores > src > app > dashboard > dashboard.component.html > mat-sidenav-container > mat-sidenav > mat-nav-list > mat-ac
1
2 <mat-sidenav-container autosize>
3   <mat-sidenav #sidenav mode = "side" opened="true" >
4     <mat-nav-list>
5
6       <mat-list-item>
7         <mat-icon mat-list-icon class="chevron_button sidenav-button" (click)="isExpanded = !isExpanded"
8           [ngClass]="{'rotated': isExpanded}" >chevron_right</mat-icon>
9       </mat-list-item>
10
11     <mat-accordion class="example-headers-align" multi>
12
13       <mat-expansion-panel>
14         <mat-expansion-panel-header>
15           <mat-panel-title>
16             <mat-list-item >
17               <mat-icon mat-list-icon >dashboard</mat-icon>
18               <span *ngIf="isExpanded" >General</span>
19             </mat-list-item>
20           </mat-panel-title>
21         </mat-expansion-panel-header>
22
23       <mat-list-item appearance="fill" [routerLink]="['medidores/general']">
24         <mat-icon mat-list-icon >R</mat-icon>
25         <span *ngIf="isExpanded" >Tiempo Real</span>
26       </mat-list-item>
```

Fuente: Elaboración propia

Ilustración 42. Hoja de Estilos del menú principal

```
mat-sidenav-container{
  height: 100%;
}

mat-sidenav-container mat-sidenav mat-nav-list .chevron_button{
  transition: 300ms ease-in-out;
  transform: rotate(0deg);
  right: 0;
  float: right;
  position: absolute;
}

mat-sidenav-container mat-sidenav mat-nav-list .chevron_button.rotated{
  transform: rotate(180deg);
}

.icon-simu{
  margin: auto
}

.mat-expansion-panel-header-title, .mat-expansion-panel-header-description{
  margin-right: 0px !important;
}
```

Fuente: Elaboración propia

Ilustración 43. Maquetado de Dashboard y carga de gráficos de Grafana

```
<div class="cuerpo">
  <h1>Medidor General</h1>

  <div class="seccion-uno">
    <div class="kw">
      <iframe src="http://localhost:3000/d-solo/ywx8b4b7k/real-time?orgId=1&panelId=24" frameborder="0"></iframe>
    </div>
    <div class="precio-kw">
      <iframe src="http://localhost:3000/d-solo/ywx8b4b7k/real-time?orgId=1&panelId=38" frameborder="0"></iframe>
    </div>
    <div class="ocupación">
      <iframe src="http://localhost:3000/d-solo/ywx8b4b7k/real-time?orgId=1&panelId=10" frameborder="0"></iframe>
    </div>
  </div>

  <br>

  <div class="seccion-dos">
    <div class="superficie">
      <iframe src="http://localhost:3000/d-solo/ywx8b4b7k/real-time?orgId=1&panelId=40" frameborder="0"></iframe>
    </div>
    <div class="kw-superficie">
      <iframe src="http://localhost:3000/d-solo/ywx8b4b7k/real-time?orgId=1&panelId=42" frameborder="0"></iframe>
    </div>
  </div>
</div>
```

Fuente: Elaboración propia

Ilustración 44. Codificación de Gráficos histórica

```
getMedidorDia(fecha: string){
  this._medidorService.getMedidorDia(fecha+"n"+this.piso).subscribe(data => {
    if(Object.keys(data).length === 0){
      console.log("no datos");
    }else{
      this.listMedidor = data;
      console.log(data)
      this.fecha = this.listMedidor.map((listMedidor: any) => listMedidor.time)
      this.voltaje = this.listMedidor.map((listMedidor: any) => listMedidor.v3ph)
      this.intencidad = this.listMedidor.map((listMedidor: any) => listMedidor.i3ph)
      this.pf3ph = this.listMedidor.map((listMedidor: any) => listMedidor.pf3ph)

      delay(300);

      this.graph = {
        data: [
          { x: this.fecha ,
            y: this.voltaje,
            type: 'line' },
        ],
        layout: {
          title: 'Voltaje Trifasico',
          yaxis: {
            range: [227.5, 230]
          }
        }
      };
    }
  });
}
```

Fuente: Elaboración propia

Ilustración 45. Codificación de generación de reporte

```
downloadPDF(){
  const DATA : any = document.getElementById('htmlData');
  const doc = new jsPDF('p' , 'pt', 'a4');
  const options = {
    background: 'white',
    scale: 3,
  };

  html2canvas(DATA, options).then((canvas) => {

    const img = canvas.toDataURL('image/PNG');

    // Add image Canvas to PDF
    const bufferX = 15;
    const bufferY = 15;
    const imgProps = (doc as any).getImageProperties(img);
    const pdfWidth = doc.internal.pageSize.getWidth() - 2 * bufferX;
    const pdfHeight = (imgProps.height * pdfWidth) / imgProps.width;
    doc.addImage(img, 'PNG', bufferX, bufferY, pdfWidth, pdfHeight, undefined, 'FAST');
    return doc;
  }).then((docResult) => {
    docResult.save(`${new Date().toISOString()}_tutorial.pdf`);
  });
}
```

Fuente: Elaboración propia

2.5.5. Desarrollo de gráficas en Grafana

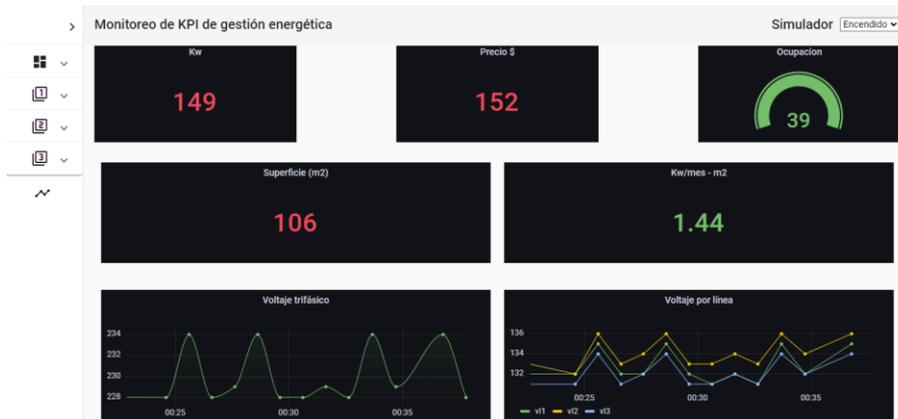
Ilustración 46. Gráficas generadas de Grafana



Fuente: Elaboración propia

2.5.6. Interfaces gráficas del sistema

Ilustración 47. Dashboard en Tiempo Real



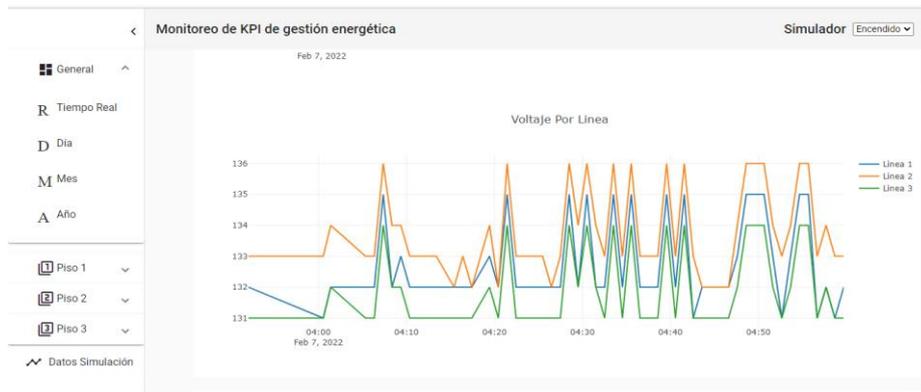
Fuente: Elaboración propia

Ilustración 48. Dashboard Histórico



Fuente: Elaboración propia

Ilustración 49. Gráficas históricas



Fuente: Elaboración propia

Ilustración 50. Datos de simulación

simulacion-list works!

Medidor	Id	voltaje trifasico	Voltaje linea 1	Voltaje linea 2	Voltaje linea 3	Intensidad trifasica	Energia Activa
1	1	228	132	133	131	49.2	0.2247955
1	2	228	132	133	131	49.2	0.3371933
1	3	228	132	132	131	54	0.449591
1	4	228	132	132	131	76.8	0.5619888
1	5	228	132	133	131	81.6	0.6743866
1	6	228	132	133	131	84	0.2247955
1	7	229	132	134	132	42	0.3371933
1	8	229	132	134	132	45.6	0.449591
1	9	229	132	134	132	51.6	0.5619888
1	10	228	131	133	131	36	0.6743866
1	11	228	131	133	131	43.2	0.2247955
1	12	228	131	133	131	48	0.3371933

Fuente: Elaboración propia

3. Capítulo III. Evaluación del prototipo

3.1. Plan de Evaluación

Para realizar la evaluación de calidad al sistema se evaluó según las métricas de la norma ISO/IEC 9126 que establece los siguientes parámetros:

- Funcionalidad
- Confiabilidad
- Usabilidad
- Eficiencia
- Mantenibilidad
- Portabilidad

Para facilitar su evaluación se emplea la escala de Likert puntuando desde 1 (Pésimo), hasta 5 (Excelente).

Tabla 22: Escala de Likert

Escala de Likert	
1	Pésimo
2	Malo
3	Regular
4	Bueno
5	Excelente

Fuente: Elaboración propia

3.2. Resultados de Evaluación

3.2.1. Evaluación de calidad

Tabla 23: Evaluación de calidad

característica	Sub característica	Criterio	Puntuación
Funcionalidad	Adecuación	¿El software tiene la capacidad de cumplir con las tareas por las cuales se desarrolló?	5
	Exactitud	¿El software cumple con los resultados solicitados de manera correcta y precisión?	5
	Interoperabilidad	¿El sistema logra interactuar con otro sistema?	5
Total			15/15
Eficiencia	Conducta en el tiempo	¿El tiempo de respuesta es el adecuado al usar el sistema?	3
	Consumo de recursos	¿Los recursos del sistema son ocupados de manera adecuada?	4
Total			7/10
Usabilidad	Operabilidad	¿El sistema es fácil de operarlo y controlarlo?	5
	Comprensibilidad	¿Es fácil de entender el funcionamiento del sistema?	4
	Facilidad de aprender	¿Es fácil de aprender a usar el sistema?	5
Total			14/15
Portabilidad	Coexistencia	¿El software interactúa y comparte recursos con otro?	4
	Adaptabilidad	¿Es fácil de adaptar el sistema en diversos entornos de trabajo?	4
Total			8/10
Mantenibilidad	Capacidad de análisis	¿Es fácil identificar fallas para ser corregidas o modificadas?	5
	Capacidad de modificación	¿Es fácil de modificar el sistema ante cualquier cambio del mismo?	5
	Facilidad de pruebas	¿Es fácil realizar pruebas del sistema?	5
Total			15/15
Confiabilidad	Tolerancia a fallos	¿El software es capaz de recuperar datos en caso de fallas?	2
	Recuperación	¿El software es capaz de recuperar datos en caso de fallas?	5
Total			12/15

Fuente: Elaboración propia

Después de analizar los resultados obtenidos al realizar la evaluación de calidad, según las métricas establecida en la norma ISO 9126, podemos resaltar el nivel de aceptación que el sistema alcanzo estando en excelente y bueno en casi todos los criterios de evaluación, del mismo modo podemos observar una penalización en los criterios referente a tiempos de carga, esto ocurre como consecuencia de usar gráficas de tercero que ralentiza la carga y renderizado del sistema.

3.2.2. Evaluación con herramientas

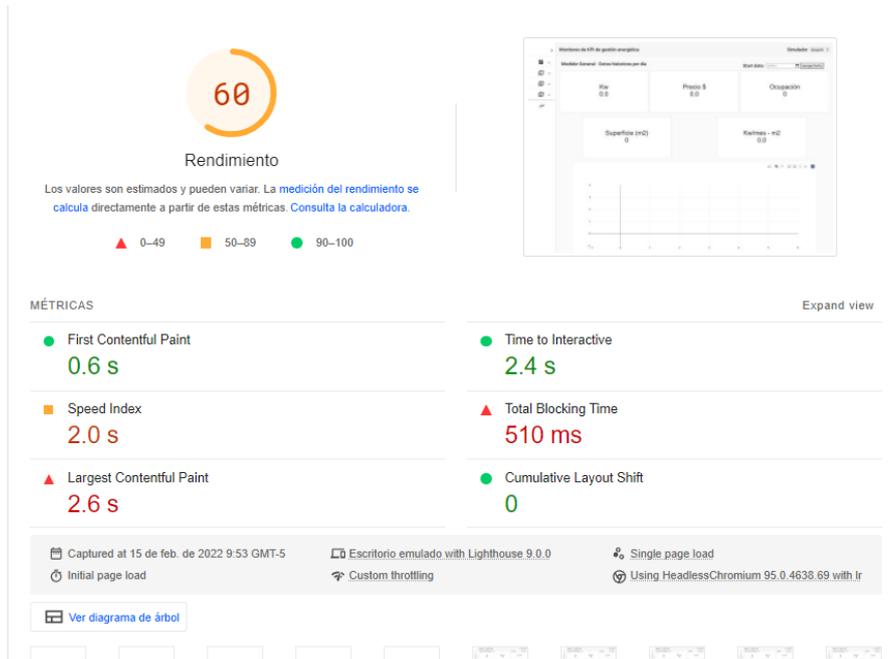
Se especifican los resultados obtenidos de la evaluación de la calidad realizada con diferentes herramientas web gratuitas que valoran términos como posicionamiento, usabilidad, rendimiento, entre otros.

Tabla 24: Herramientas de SEO

Análisis de herramientas Seo conforme a la Norma ISO/IEC 9126					
Herramienta	Funcionalidad	Eficiencia	Usabilidad	Accesibilidad	Portabilidad
Google Speed			X		X
GTMetrix	X	X			
Functional Accessibility Evaluator 2.2				X	

Fuente: Elaboración propia

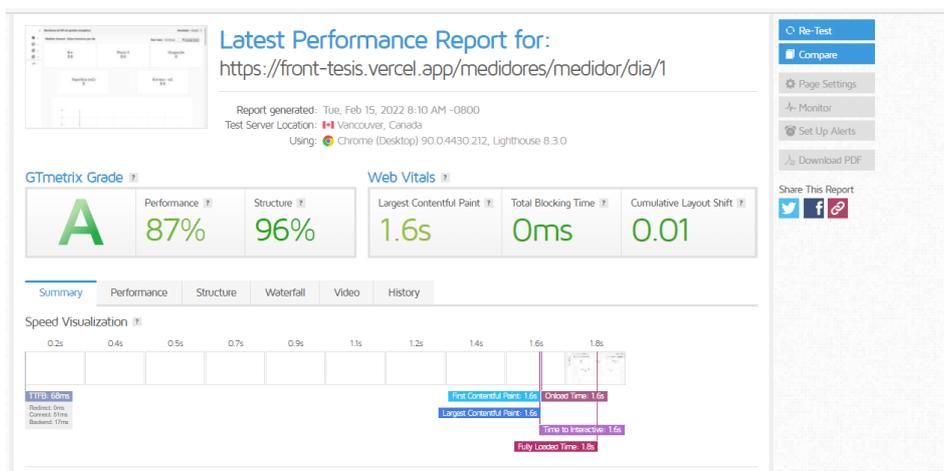
Ilustración 51. Google Speed



Fuente: Elaboración propia

Google Speed es una herramienta de Google que nos permite evaluar aspectos de páginas webs referentes a rendimiento, portabilidad y usabilidad. Según los resultados de la evaluación tenemos un nivel aceptable de rendimiento, con un tiempo de carga de 2.6 segundos, esto es debido a la gran cantidad de componentes gráficos que posee el sistema, aun así, está en niveles aceptables de rendimiento.

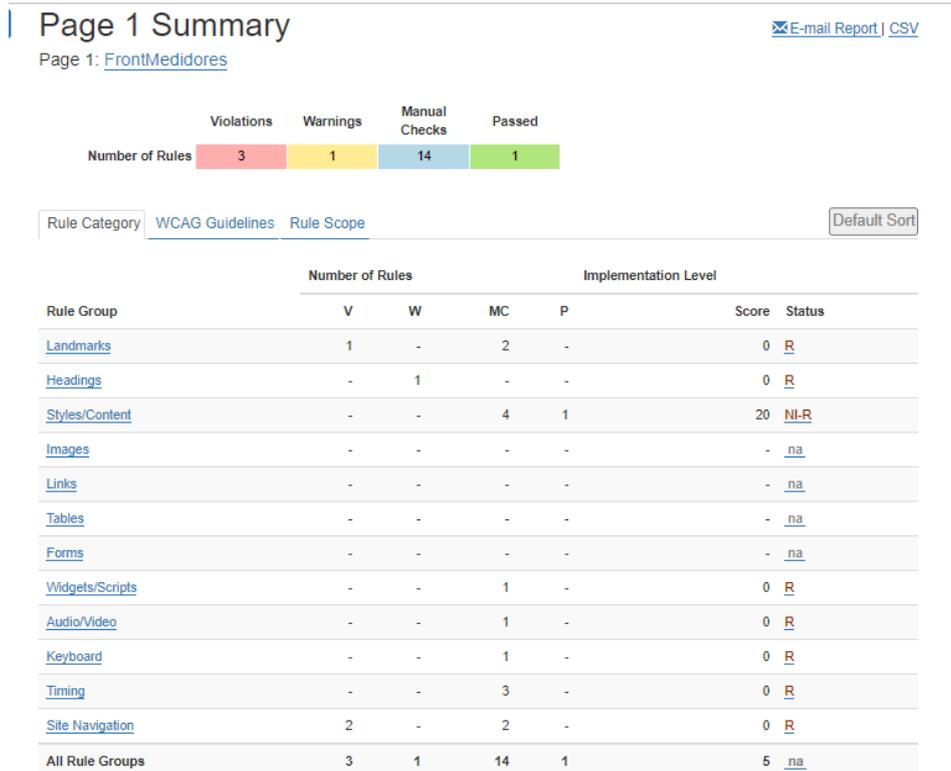
Ilustración 52. GTMetrix



Fuente: Elaboración propia

Por otro lado, al evaluar con la herramienta GTMetrix obtenemos una valoración superior al obtenido con la herramienta de Google, como resultado los aspectos de funcionalidad y eficiencia del contenido tienen una puntuación buena.

Ilustración 53. Functional Accessibility Evaluator 2.2



Fuente: Elaboración propia

Por último, la evaluación de la herramienta “Functional Accessibility Evaluator” evalúa los aspectos referentes a accesibilidad de contenido web.

3.3. Conclusiones

- El presente trabajo al ser implementado en un sistema web permite una gestión eficiente y multiplataforma de KPI's de gestión energética fácil de administrar y escalable.
- El uso de gráficos estadísticos permite una mayor comprensión de datos, y al ser interactivos ofrece la facilidad de realizar filtros o segmentación de las diferentes variables referentes a gestión energética.
- La generación de datos estadísticos ofrece una consolidación de la información que maximiza la eficiencia al analizar los datos medidos.

- El uso de reportes permite recibir información concreta, precisa y de utilidad en un formato portable para su posterior análisis.
- La implementación de un sistema de simulación de cargas eléctricas permitió el correcto desarrollo y pruebas del sistema Web, mitigando el inconveniente relacionado con la inaccesibilidad del servidor real.
- El Separar los servidores Back-End y Front-End siguiendo una arquitectura adecuada, nos permite obtener una escalabilidad y mantenimiento efectivo en el desarrollo de sistemas web.
- La utilización de versiones estables del software de desarrollo y codificación del sistema permite contar con gran cantidad de módulos o paquetes de utilidad con documentación amplia.
- Utilizar frameworks de desarrollo como lo son AngularJS o NodeJS nos ofrece una estructura base en la elaboración de proyectos, facilitando así la codificación y diseño de los mismos.

3.4. Recomendaciones

- Contar con mayor cantidad de datos para la creación del módulo de simulación nos permitiría tener datos más fiables al emular el comportamiento habitual de los medidores en un campus universitario.
- Desplegar una infraestructura para el acceso de remoto a los datos generados por los medidores, permitirá un mejor flujo de trabajo y reduciría la cantidad de código implementado para las futuras actualizaciones del sistema.
- Trabajar con un sistema de control de versiones como lo es Git, ayudará a la integración de nuevos colaboradores, al igual que reduciría el tiempo de pruebas antes del despliegue de futuras actualizaciones.

BIBLIOGRAFÍA

- [1] W. Villegas-Ch, X. Palacios-Pacheco, y S. Luján-Mora, «Application of a Smart City Model to a Traditional University Campus with a Big Data Architecture: A Sustainable Smart Campus», *Sustainability*, vol. 11, n.º 10, p. 2857, may 2019, doi: 10.3390/su11102857.
- [2] W. Villegas-Ch, J. Molina-Enriquez, C. Chicaiza-Tamayo, I. Ortiz-Garcés, y S. Luján-Mora, «Application of a Big Data Framework for Data Monitoring on a Smart Campus», *Sustainability*, vol. 11, n.º 20, p. 5552, oct. 2019, doi: 10.3390/su11205552.

- [3] Á. I. Torres Quijije, J. C. Pisco Vanegas, R. S. Pérez Parraga, y I. G. Vera García, «Monitoreo en tiempo real del consumo de energía eléctrica residencial que permita su apropiada gestión», *Revista Universidad y Sociedad*, vol. 12, n.º 2, pp. 218-222, jun. 2020.
- [4] D. Torregrossa, F. Hernández-Sancho, J. Hansen, A. Cornelissen, T. Popov, y G. Schutz, «Energy saving in wastewater treatment plants: A plant-generic cooperative decision support system», *Journal of Cleaner Production*, vol. 167, pp. 601-609, nov. 2017, doi: 10.1016/j.jclepro.2017.08.181.
- [5] OEIT, «The MIT-Microsoft Alliance», *iCampus*, 3 de noviembre de 2011. <http://icampus.mit.edu/about/> (accedido 13 de septiembre de 2020).
- [6] R. Gomes *et al.*, «Towards a Smart Campus: Building-User Learning Interaction for Energy Efficiency, the Lisbon Case Study», en *Handbook of Theory and Practice of Sustainable Development in Higher Education*, W. Leal Filho, L. Brandli, P. Castro, y J. Newman, Eds. Cham: Springer International Publishing, 2017, pp. 381-398. doi: 10.1007/978-3-319-47868-5_24.
- [7] D. Rico-Bautista, G. Maestre, y C. Guerrero, «Caracterización de la situación actual de las tecnologías inteligentes para una Universidad inteligente en Colombia/Latinoamérica», *Revista Iberica de Sistemas e Tecnologías de Informacao*, vol. 2020, pp. 484-501, 1 de marzo de 2020.
- [8] A. Recalde, I. Endara, M. Quimis, y C. Romero, «Operational framework proposal for ESPOL university 2.0 smart campus implementation», en *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, Salinas, oct. 2017, pp. 1-6. doi: 10.1109/ETCM.2017.8247523.
- [9] J. Novillo-Vicuña, D. Hernández Rojas, B. Mazón Olivo, J. Molina Ríos, y O. Cárdenas Villavicencio, *Arduino y el Internet de las cosas*, 1.ª ed. Editorial Científica 3Ciencias, 2018. doi: 10.17993/IngyTec.2018.45.
- [10] X.-B. Jin, X.-H. Yu, X.-Y. Wang, Y.-T. Bai, T.-L. Su, y J.-L. Kong, «Deep Learning Predictor for Sustainable Precision Agriculture Based on Internet of Things System», *Sustainability*, vol. 12, n.º 4, p. 1433, feb. 2020, doi: 10.3390/su12041433.
- [11] D. Hernández-Rojas, T. Fernández-Caramés, P. Fraga-Lamas, y C. Escudero, «Design and Practical Evaluation of a Family of Lightweight Protocols for Heterogeneous Sensing through BLE Beacons in IoT Telemetry Applications», *Sensors*, vol. 18, n.º 2, p. 57, dic. 2017, doi: 10.3390/s18010057.
- [12] D. Hernández Rojas, B. Mazon Olivo, y C. Escudero, *Internet de las cosas (IoT)*. Universidad Técnica de Machala, 2018.
- [13] D. L. Hernández Rojas, B. E. Mazón Olivo, y A. M. Campoverde Marca, «Cloud computing para el internet de las cosas. Caso de estudio orientado a la agricultura de precisión», *I Congreso Internacional de Ciencia y Tecnología UTMACH 2015*, pp. 47-53, 2015.
- [14] J. Berrú-Ayala, D. Hernandez-Rojas, P. Morocho-Díaz, J. Novillo-Vicuña, B. Mazon-Olivo, y A. Pan, «SCADA System Based on IoT for Intelligent Control of Banana Crop Irrigation», en *Applied Technologies*, vol. 1193, Cham: Springer International Publishing, 2020, pp. 243-256. doi: 10.1007/978-3-030-42517-3_19.
- [15] D. L. Hernández Rojas y H. Villegas, «Implementación de un sistema de monitoreo remoto de una red de impresoras multi-funcionales basado en SNMP y programado con Labview.», *Revista Tecnológica - ESPOL*, vol. 28,

- n.º 4, Art. n.º 4, dic. 2015, Accedido: 17 de febrero de 2022. [En línea]. Disponible en: <http://www.rte.espol.edu.ec/index.php/tecnologica/article/view/404>
- [16] J. J. Cartuche Calva, D. L. Hernández Rojas, R. F. Morocho Román, y C. D. Radicelli García, «Seguridad IoT: Principales amenazas en una taxonomía de activos», *HAMU*, vol. 7, n.º 3, p. 51, ene. 2021, doi: 10.21503/hamu.v7i3.2192.
- [17] W. Yaïci, K. Krishnamurthy, E. Entchev, y M. Longo, «Recent Advances in Internet of Things (IoT) Infrastructures for Building Energy Systems: A Review», *Sensors*, vol. 21, n.º 6, p. 2152, mar. 2021, doi: 10.3390/s21062152.
- [18] D. Hernandez-Rojas, B. Mazon-Olivo, J. Novillo-Vicuña, C. Escudero-Cascon, A. Pan-Bermudez, y G. Belduma-Vacacela, «IoT Android Gateway for Monitoring and Control a WSN», en *Technology Trends*, vol. 798, M. Botto-Tobar, N. Esparza-Cruz, J. León-Acurio, N. Crespo-Torres, y M. Beltrán-Mora, Eds. Cham: Springer International Publishing, 2018, pp. 18-32. doi: 10.1007/978-3-319-72727-1_2.
- [19] J. P. Novillo-Vicuña, D. L. Hernandez-Rojas, B. Mazon-Olivo, y K. D. Correa-Elizaldes, «Monitoreo inalámbrico de señales eléctricas de voltaje 110/220V a través de Arduino», *Alternativas*, vol. 19, n.º 1, jun. 2019, doi: 10.23878/alternativas.v19i1.198.
- [20] Y. Guaman, G. Ninahualpa, G. Salazar, y T. Guarda, «Comparative Performance Analysis between MQTT and CoAP Protocols for IoT with Raspberry PI 3 in IEEE 802.11 Environments», en *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, Sevilla, Spain, jun. 2020, pp. 1-6. doi: 10.23919/CISTI49556.2020.9140905.
- [21] S. Quincozes, T. Emilio, y J. Kazienko, «MQTT Protocol: Fundamentals, Tools and Future Directions», *IEEE Latin Am. Trans.*, vol. 17, n.º 09, pp. 1439-1448, sep. 2019, doi: 10.1109/TLA.2019.8931137.
- [22] B. Mazon-Olivo, D. Hernández-Rojas, J. Maza-Salinas, y A. Pan, «Rules engine and complex event processor in the context of internet of things for precision agriculture», *Computers and Electronics in Agriculture*, vol. 154, pp. 347-360, nov. 2018, doi: 10.1016/j.compag.2018.09.013.
- [23] D. Hernández-Rojas, T. Fernández-Caramés, P. Fraga-Lamas, y C. Escudero, «A Plug-and-Play Human-Centered Virtual TEDS Architecture for the Web of Things», *Sensors*, vol. 18, n.º 7, p. 2052, jun. 2018, doi: 10.3390/s18072052.
- [24] A. M. Campoverde, D. L. Hernández Rojas, y B. E. Mazón Olivo, «Cloud computing con herramientas open-source para Internet de las cosas», *Maskana*, vol. 6, pp. 173-182, dic. 2015.
- [25] D. C. Yacchirema Vargas y C. E. Palau Salvador, «Smart IoT Gateway For Heterogeneous Devices Interoperability», *IEEE Latin Am. Trans.*, vol. 14, n.º 8, pp. 3900-3906, ago. 2016, doi: 10.1109/TLA.2016.7786378.
- [26] R. Vasileva, L. Rodrigues, N. Hughes, C. Greenhalgh, M. Goulden, y J. Tennison, «What Smart Campuses Can Teach Us about Smart Cities: User Experiences and Open Data», *Information*, vol. 9, n.º 10, p. 251, oct. 2018, doi: 10.3390/info9100251.
- [27] L.-S. Huang, J.-Y. Su, y T.-L. Pao, «A Context Aware Smart Classroom Architecture for Smart Campuses», *Applied Sciences*, vol. 9, n.º 9, p. 1837, may 2019, doi: 10.3390/app9091837.

- [28] G. Guo, «Design and Implementation of Smart Campus Automatic Settlement PLC Control System for Internet of Things», *IEEE Access*, vol. 6, pp. 62601-62611, 2018, doi: 10.1109/ACCESS.2018.2877023.
- [29] T. M. Fernández-Caramés y P. Fraga-Lamas, «Towards Next Generation Teaching, Learning, and Context-Aware Applications for Higher Education: A Review on Blockchain, IoT, Fog and Edge Computing Enabled Smart Campuses and Universities», *Applied Sciences*, vol. 9, n.º 21, p. 4479, oct. 2019, doi: 10.3390/app9214479.
- [30] M. Collotta y G. Pau, «A Novel Energy Management Approach for Smart Homes Using Bluetooth Low Energy», *IEEE J. Select. Areas Commun.*, vol. 33, n.º 12, pp. 2988-2996, dic. 2015, doi: 10.1109/JSAC.2015.2481203.
- [31] N. T. Mbungu, R. M. Naidoo, R. C. Bansal, y V. Vahidinasab, «Overview of the Optimal Smart Energy Coordination for Microgrid Applications», *IEEE Access*, vol. 7, pp. 163063-163084, 2019, doi: 10.1109/ACCESS.2019.2951459.
- [32] H. Al Haj Hassan, A. Pelov, y L. Nuaymi, «Integrating Cellular Networks, Smart Grid, and Renewable Energy: Analysis, Architecture, and Challenges», *IEEE Access*, vol. 3, pp. 2755-2770, 2015, doi: 10.1109/ACCESS.2015.2507781.
- [33] R. Ullah, Y. Faheem, y B.-S. Kim, «Energy and Congestion-Aware Routing Metric for Smart Grid AMI Networks in Smart City», *IEEE Access*, vol. 5, pp. 13799-13810, 2017, doi: 10.1109/ACCESS.2017.2728623.
- [34] F. Divina, F. Vela, y M. Torres, «Biclustering of Smart Building Electric Energy Consumption Data», *Applied Sciences*, vol. 9, n.º 2, p. 222, ene. 2019, doi: 10.3390/app9020222.
- [35] C. Benavente-Peces, «On the Energy Efficiency in the Next Generation of Smart Buildings—Supporting Technologies and Techniques», *Energies*, vol. 12, n.º 22, p. 4399, nov. 2019, doi: 10.3390/en12224399.
- [36] I. Sittón-Candanedo, R. S. Alonso, Ó. García, L. Muñoz, y S. Rodríguez-González, «Edge Computing, IoT and Social Computing in Smart Energy Scenarios», *Sensors*, vol. 19, n.º 15, p. 3353, jul. 2019, doi: 10.3390/s19153353.
- [37] I. Dincer y C. Acar, «Smart energy systems for a sustainable future», *Applied Energy*, vol. 194, pp. 225-235, may 2017, doi: 10.1016/j.apenergy.2016.12.058.
- [38] Y. Li, R. García-Castro, N. Mihindikulasooriya, J. O'Donnell, y S. Vega-Sánchez, «Enhancing energy management at district and building levels via an EM-KPI ontology», *Automation in Construction*, vol. 99, pp. 152-167, mar. 2019, doi: 10.1016/j.autcon.2018.12.010.
- [39] E. Amrina y F. Imansuri, «Key Performance Indicators for Sustainable Campus Assessment: A Case of Andalus University», en *Industrial Engineering, Management Science and Applications 2015*, vol. 349, M. Gen, K. J. Kim, X. Huang, y Y. Hiroshi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 11-18. doi: 10.1007/978-3-662-47200-2_2.
- [40] M. A. Khan *et al.*, «Effective Demand Forecasting Model Using Business Intelligence Empowered With Machine Learning», *IEEE Access*, vol. 8, pp. 116013-116023, 2020, doi: 10.1109/ACCESS.2020.3003790.
- [41] B. Guo, Y. Liu, Y. Ouyang, V. W. Zheng, D. Zhang, y Z. Yu, «Harnessing the Power of the General Public for Crowdsourced Business Intelligence: A

- Survey», *IEEE Access*, vol. 7, pp. 26606-26630, 2019, doi: 10.1109/ACCESS.2019.2901027.
- [42] O. Azeroual, G. Saake, y M. Abuosba, «ETL Best Practices for Data Quality Checks in RIS Databases», *Informatics*, vol. 6, n.º 1, p. 10, mar. 2019, doi: 10.3390/informatics6010010.
- [43] Xindong Wu, Xingquan Zhu, Gong-Qing Wu, y Wei Ding, «Data mining with big data», *IEEE Trans. Knowl. Data Eng.*, vol. 26, n.º 1, pp. 97-107, ene. 2014, doi: 10.1109/TKDE.2013.109.
- [44] H. Liao, M. Tang, L. Luo, C. Li, F. Chiclana, y X.-J. Zeng, «A Bibliometric Analysis and Visualization of Medical Big Data Research», *Sustainability*, vol. 10, n.º 2, p. 166, ene. 2018, doi: 10.3390/su10010166.
- [45] Z. Wu, Y. Zhou, H. Wang, y Z. Jiang, «Depth prediction of urban flood under different rainfall return periods based on deep learning and data warehouse», *Science of The Total Environment*, vol. 716, p. 137077, may 2020, doi: 10.1016/j.scitotenv.2020.137077.
- [46] L. W. Santoso y Yulia, «Data Warehouse with Big Data Technology for Higher Education», *Procedia Computer Science*, vol. 124, pp. 93-99, 2017, doi: 10.1016/j.procs.2017.12.134.
- [47] F. Fuster Just, «Entorno con integración continua para aplicaciones web desarrolladas con AngularJS», *ing. agua*, vol. 18, n.º 1, p. ix, 2020, doi: 10.4995/ia.2014.3293.
- [48] A. M. Vukićević, I. Mačužić, M. Djapan, V. Milićević, y L. Shamina, «Digital Training and Advanced Learning in Occupational Safety and Health Based on Modern and Affordable Technologies», *Sustainability*, vol. 13, n.º 24, p. 13641, dic. 2021, doi: 10.3390/su132413641.
- [49] M. de la H. Navarro, «Servicio y Aplicación Móvil de prescripción y seguimiento de ejercicio físico usando Node.js, React Native y MQTT», 2021, p. 37.
- [50] J. E. M. Maya y F. S. S. Vega, «MÓDULO DE DESARROLLO DE PRUEBAS PSICOTÉCNICAS PARA LA PLATAFORMA MENTAL DATA», p. 56.
- [51] J.-M. Sierra-Fernández, O. Florencias-Oliveros, M.-J. Espinosa-Gavira, J.-J. González-de-la-Rosa, A. Agüera-Pérez, y J.-C. Palomares-Salas, «Online System for Power Quality Operational Data Management in Frequency Monitoring Using Python and Grafana», *Energies*, vol. 14, n.º 24, p. 8304, dic. 2021, doi: 10.3390/en14248304.
- [52] J. D. Pinzón C., A. Corredor R., F. Santamaría P., J. A. Hernández M., y C. L. Trujillo R., «Implementación de indicadores energéticos en centros educativos. Caso de estudio: Edificio Alejandro Suárez Copete-Universidad Distrital Francisco José de Caldas», *Revista EAN*, n.º 77, pp. 186-200, jul. 2014.
- [53] R. Fadhel Khelifa y K. Jelassi, «An energy monitoring and management system based on key performance indicators», en *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, Berlin, Germany, sep. 2016, pp. 1-6. doi: 10.1109/ETFA.2016.7733542.
- [54] D. Pramangioulis, K. Atsonios, N. Nikolopoulos, D. Rakopoulos, P. Grammelis, y E. Kakaras, «A Methodology for Determination and Definition of Key Performance Indicators for Smart Grids Development in Island Energy

- Systems», *Energies*, vol. 12, n.º 2, p. 242, ene. 2019, doi: 10.3390/en12020242.
- [55] J. Al Dakheel, C. Del Pero, N. Aste, y F. Leonforte, «Smart buildings features and key performance indicators: A review», *Sustainable Cities and Society*, vol. 61, p. 102328, oct. 2020, doi: 10.1016/j.scs.2020.102328.
- [56] J. Li, Q. Wang, y H. Zhou, «Establishment of Key Performance Indicators for Green Building Operations Monitoring—An Application to China Case Study», *Energies*, vol. 13, n.º 4, p. 976, feb. 2020, doi: 10.3390/en13040976.

Anexos

Anexo 1: Análisis de Riesgos

Tabla 25: Matriz de riesgos

Matriz de Riesgos										
N o.	EDT: Componente/ Entregable	Tipo de riesgo	Riesgo	Efecto	Impac to	Probabilid ad	Evaluación del Riesgo			Estrategia de Mitigación/ Respuesta a los riesgos
							Calificac ión	Severidad		
								Val or	Nive l	
1	1.3 Modelado	Alcance	Dificultad al diseñar la distribución de elementos del Dashboard	Incumplien do con los plazos al rectificar distribución	2	1	2	1	Bajo	Realizar la correcta adquisición de los requisitos del proyecto.
2	1.4.2.2 Desarrollo Función Carga de Datos	Operacional es	Error al realizar la carga de datos, o carga de datos incompleta	Resultados erróneos	3	2	6	3	Alto	Asegurarse de depurar correctamente los datos y la estructura de los mismos.

3	1.4.2.1 Desarrollo Función Procesamiento de Datos	Operacionales	Error al manipular datos o implementar métodos que ejecuten cálculos	Gráficos sin valor al presentar datos erróneos	3	1	3	2	Medio	Realizar Pruebas unitarias.
4	1.4.2.2 Desarrollo Función Presentación de Datos	Operacionales	Escasez de gráficos para la correcta presentación de los datos	Dificultad de interpretar los gráficos al no ser los adecuados	2	2	4	2	Medio	Realizar Pruebas unitarias.
5	1.5.1 Pruebas	Operativo	Error en la implementación de pruebas	Errores en el programa en producción	3	1	3	2	Medio	Verificar la confiabilidad de las pruebas unitarias.

6	1.2.4 Búsqueda bibliográfica	Normativa	No encontrar artículos que cumplan con los requisitos de la revista	Rechazo de tesis por parte del comité evaluador.	3	1	3	2	Medio	Buscar artículos en revistas de alto impacto y de reciente publicación.
7	1.2.4 Redacción del Marco teórico	Normativa	Error en la elaboración de redacción y citas	Tesis enviada a revisión.	3	2	6	3	Alto	Verificar el correcto uso de sintaxis y citas bibliográficas.
8	1.5.5 Trabajo final entregado a Tutor	Normativa	Errores en redacción o estructura	Tesis Aprobada por el tutor	3	1	3	2	Medio	Verificar la estructura de artículos según la revista elegida.

Anexo 2: Gestión de la calidad

Políticas de calidad del Programa

- Fiabilidad: Capacidad de operar sin errores.
- Modificable: Capacidad de hacer los cambios necesarios de una forma sencilla.
- Comprensible: Capacidad de comprender el software operativo, de cara a un cambio o arreglo.
- Rendimiento: Velocidad y compacidad del software.
- Utilizable: Capacidad de uso sencillo del software.
- Probable: Capacidad de construir y ejecutar fácilmente casos de prueba.

Interesados de la calidad (Clientes internos y externos)

Tabla 26: Interesados de la calidad

Nombre	Tipo	Satisfacción
Pedro Ricardo Cruz Zambrano	Interno	Por analizar
Ing. Dixcys Hernández	Interno	Por analizar
Ing. Bertha Mazón	Externo	Por analizar
Ing. Johnny Novillo	Externo	Por analizar

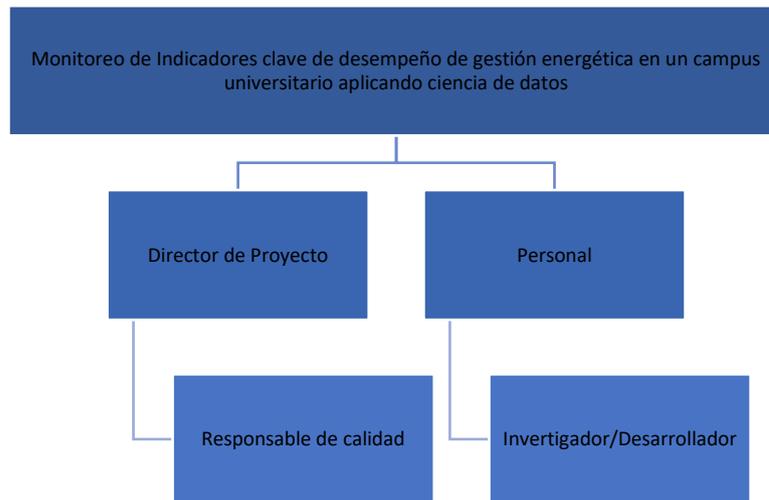
Fuente: Elaboración propia

Expectativas y requerimientos de calidad

- Presentación de variables energéticas
- Gráficos limpios y legibles
- Aplicación de diversos filtros para las diferentes vistas
- Rapidez y fluidez de ejecución
- Generación de gráficos dinámicos

Equipo de calidad, roles y responsabilidades de la gestión de calidad

Ilustración 54. Equipo de calidad



Fuente: Elaboración propia

Tabla 27: Roles y Responsabilidades en la calidad

Roles	Responsabilidades
Responsable de calidad	<ul style="list-style-type: none"> ● Planificar y establecer los procedimientos necesarios para el cumplimiento de la calidad del proyecto. ● Asegurarse que el proyecto cumpla con los requerimientos y lineamientos obtenidos en la fase de planeación y diseño.
Investigador/Desarrollador	<ul style="list-style-type: none"> ● Cumplir con los estándares de Usabilidad y IX del proyecto. ● Implementar estándares de calidad en el código del proyecto. ● Realizar pruebas de calidad en el código.

Fuente: Elaboración propia

Normas, guías, estándares o modelo de calidad acordes al proyecto, los entregables y su proceso de desarrollo.

Según ISO/IEC 2500

- Funcionalidad
- Fiabilidad
- Usabilidad
- Eficiencia
- Mantenibilidad
- Portabilidad

Anexo 3: Acta de constitución

Tabla 28: Acta de constitución

Acta de Constitución del Proyecto (Project Charter)		
Proyecto:		Monitoreo de Indicadores clave de desempeño de gestión energética en un campus universitario aplicando ciencia de datos
Código:	P-001	
Fecha de inicio:	08-11-2021	Duración (Meses o días)
Fecha de cierre:	18-02-2022	3 meses
Tipo de proyecto:	Artículo científico	
Director del Proyecto:	Ing. Dixys Hernandez	
Propósito o resumen del Proyecto:		
Este documento describe el desarrollo de un panel de control o Dashboard para el análisis de datos referentes a consumo energético generados por medidores ABB M2M instalados en un Smart Campus. El panel de control implementado presenta gráficas estadísticas de variables eléctricas en periodos de tiempo específicos y en tiempo real.		
Objetivos:		
Meta (Objetivo General):		Implementar un sistema web que permita el monitoreo de indicadores clave de desempeño de gestión energética utilizando business Intelligence
Específicos:		
<ul style="list-style-type: none"> • Obtener estadísticas del consumo de las diferentes variables en periodos de tiempo determinados. • Presentar gráficos estadísticos en una interfaz web gráfica amigable. 		

- Desarrollar un sistema de generación de datos que simule el consumo energético habitual de un campus universitario.
- Generar reportes con datos estadísticos de las diferentes variables de gestión energética.

Estrategias y criterios de éxito (Beneficios):

El proyecto enfoca los objetivos en la implementación de un Dashborad que procese y presente gráficos estadísticos para su posterior análisis. El proyecto a identificado al componente de eficiencia como estrategia para lograr la aceptación y el apoyo al proyecto por los stakeholders.

Estructura - Miembros del proyecto

Código	Apellidos y nombres	Rol que desempeña
I-001	Pedro Ricardo Cruz Zambrano	Investigador
I-002	Ing. Sist. Mazon Olivo Bertha Eugenia, Mg	Consultor
I-003	Ing. Dixys Hernandez	Director de proyecto
I-004	Ing. Johnny Novillo	Consultor

Gerencia (Organigrama) del Proyecto



Entregables e hitos importantes (Líneas Base)

Entregable	Fecha
Borrador del Artículo	30-01-2022
Dashboard	15-01-2022
Artículo Culminado	16-02-2021

Control de Cambios

Versión	Fecha	Responsable	Descripción del cambio
Revisión			

Elaborado por:		Aprobado por:	
Pedro Ricardo Cruz Zambrano		Ing. Sist. Mazon Olivo Bertha Eugenia, Mg	

Fuente: Elaboración propia