

**Diseño e integración de un entorno web orientado a la gestión energética y análisis
predictivo de consumo eléctrico en plantas cerveceras**

Brayan Rueda Mayorga

Asesor

Luis Angel Anillo Arrieta

Tutor

Brayan Andru Montenegro Embus

Universidad Nacional Abierta y a Distancia UNAD
Escuela de Ciencias Básicas Tecnológicas e Ingenierías
Especialización en Ciencia de Datos y Analítica

2026

Resumen

El presente proyecto tiene como objetivo diseñar e implementar una solución para el manejo eficiente de los datos de energía en la Cervecería Barranquilla, con posibilidad de ser escalada a otras plantas o industrias con infraestructura similar. La electricidad constituye uno de los recursos de mayor costo en la producción de bienes y servicios, lo que hace indispensable contar con herramientas confiables para gestionar el consumo energético de procesos y equipos.

Aunque la mayoría de las plantas dispone de medidores de energía, no es común contar con software especializado que recopile automáticamente dichos consumos. Los programas ofrecidos por los fabricantes suelen requerir licencias costosas para realizar acciones básicas o integrar nuevos equipos, lo que incrementa la inversión y los gastos de mantenimiento. Además, presentan limitaciones en el manejo de datos masivos, entornos variables e infraestructuras virtualizadas, lo que genera la necesidad de personal adicional para resolver fallas.

Este desarrollo propone una interfaz adaptable y sencilla, capaz de gestionar datos energéticos en tiempo real y de predecir consumos con base a valores de producción. Con ello se busca ofrecer una alternativa más flexible y eficiente frente a las soluciones tradicionales, contribuyendo a la optimización de recursos y a la sostenibilidad industrial.

Palabras clave: Energía, gestión de datos, eficiencia energética, software industrial, sostenibilidad.

Abstract

The objective of this project is to design and implement a solution for the efficient management of energy data at Cervecería Barranquilla, with the possibility of scaling it up to other plants or industries with similar infrastructure. Electricity is one of the costliest resources in the production of goods and services, making it essential to have reliable tools to manage the energy consumption of processes and equipment.

Although most plants have energy meters, it is not common to have specialized software that automatically collects this consumption data. The programs offered by manufacturers often require expensive licenses to perform basic actions or integrate new equipment, which increases investment and maintenance costs. In addition, they have limitations in handling massive data, variable environments, and virtualized infrastructures, which creates the need for additional personnel to resolve failures.

This development proposes an adaptable and simple interface capable of managing energy data in real time and predicting consumption based on production values. The aim is to offer a more flexible and efficient alternative to traditional solutions, contributing to resource optimization and industrial sustainability.

Keywords: Energy, data management, energy efficiency, industrial software, sustainability.

Tabla de Contenido

Justificación	9
Planteamiento del Problema	11
Objetivos.....	12
Objetivo General	12
Objetivos Específicos.....	12
Marco de Referencia	13
Gestión y Eficiencia Energética Industrial.....	13
Infraestructura OT y Ciberseguridad Industrial	14
EMS (Energy Management System).....	14
Funciones de Un EMS: Adquisición de Datos, Alarmas, Análisis, Reportes y Predicción.....	15
Limitaciones Típicas de los EMS Comerciales	15
Estado del Arte.....	16
Marco Teórico.....	22
Automatización Industrial y Captura de Datos	22
OPC UA (Open Platform Communications Unified Architecture)	22
Bases de Datos NoSQL: InfluxDB	22
Desarrollo de APIs Web con Python	23
Integración de Tecnologías para la Industria 4.0	23
Docker	24
Contenedores y Virtualización	24
Funcionalidades Principales	24
Relevancia en la Industria 4.0.....	25

React.....	25
Vite	26
TypeScript.....	26
Marco Conceptual.....	28
La Ciencia de Datos en la Industria	28
Sistemas de Monitoreo Energético.....	28
Automatización de Informes	28
Indicadores Clave de Desempeño Energético (KPI).....	29
Manejo de Datos en Entornos Industriales.....	29
Metodología	30
Fase 1 (Diagnóstico Técnico y Levantamiento de Requerimientos)	30
Fase 2 (Estructuración y Modelado de Datos)	32
Figura 1 Estructura de Base de Datos	32
Arquitectura Basada en Contenedores Docker	33
Backend	33
Frontend	34
PostgreSQL.....	34
InfluxDB.....	35
Fase 3 (Desarrollo de Algoritmos de Procesamiento y Análisis Automático).....	35
Funcionamiento del Backend	36
Fase 4 (Validación, Despliegue y Monitoreo Continuo)	38
Modelo de Rendimiento del Sistema Backend.....	39
Variables del Modelo.....	39

Rendimiento Modbus Tasa de Solicitudes y Carga Sobre InfluxDB	40
Rendimiento OPC UA: Periodo Efectivo	40
Rendimiento del Proceso Keepalive	41
Carga Generada por SSE (Server-Sent Events)	41
Ejemplo Numérico	41
Cálculos de Ejemplo	42
Funcionamiento del Frontend	43
Tecnologías Principales	43
Punto de Entrada	43
Componente Raíz y Ruteo	44
Rutas Activas	44
Layout Principal	44
Comunicación Entre el Navegador y el Backend: HTTP y SSE	45
Resultado Final	47
Dashboard	47
Equipos	49
Informes	50
Predictivo	51
Cronograma Ejecutado	54
Recursos Usados	56
Conclusiones	57
Mejoras Futuras	59
Bibliografía	60

Lista de Figuras

Figura 1 <i>Estructura de Base de Datos</i>	32
Figura 2 <i>Esquema de Funcionamiento del Proyecto</i>	33
Figura 3 <i>Vista de Funcionamiento en Docker</i>	35
Figura 4 <i>Página/Dashboard</i>	48
Figura 5 <i>Página de Manejo de Equipos</i>	49
Figura 6 <i>Página/informes</i>	50
Figura 7 <i>Modal de Selección de Equipos y Modal de Selección de Variables</i>	50
Figura 8 <i>Página/Predictivo</i>	52
Figura 9 <i>Gráfico de predictivo</i>	52

Lista de Tablas

Tabla 1 *Cronograma Ejecutado*..... 54

Tabla 2 *Recursos Usados*..... 56

Justificación

El sector industrial colombiano enfrenta un incremento sostenido en las tarifas de energía eléctrica, lo que ha generado una presión significativa sobre los costos operativos. Entre 2021 y 2024, las tarifas aumentaron 15,3% anual, el mayor incremento registrado en más de dos décadas, según un análisis de Corficolombiana. Este comportamiento tarifario ha llevado a que Colombia presente precios más altos que otros países latinoamericanos como México, Ecuador y Brasil, afectando directamente la competitividad industrial. En regiones como la Costa Atlántica, donde opera Cervecería Barranquilla, las tarifas son 18% más altas que en el interior del país, debido principalmente a mayores pérdidas eléctricas y costos asociados a la infraestructura regional. Esta situación incrementa la necesidad de contar con sistemas de monitoreo energético precisos, integrados y en tiempo real. (Medina, 2024).

Actualmente, la planta cuenta con un software especializado para la captura de datos, sin embargo, debido a las políticas internas de comunicación a internet y los altos costos de mantenimiento del software, la versión actualmente funcional ya no cuenta con soporte técnico del fabricante y presenta comúnmente fallos que llevan a volver manual la captura de datos de los medidores, lo que genera errores de transcripción, retrasos en la toma de decisiones y dificultades para identificar ineficiencias en los procesos productivos. La falta de confiabilidad en los datos limita la identificación de oportunidades de ahorro energético.

Las soluciones comerciales disponibles en el mercado, como las ofrecidas por Schneider Electric, Siemens o ABB, están orientadas a la digitalización y monitoreo inteligente del consumo energético. Schneider Electric, por ejemplo, destaca que sus plataformas permiten visualizar consumos en tiempo real, detectar ineficiencias y anticipar fallas, generando ahorros de hasta 20% en costos energéticos para las empresas que adoptan sistemas avanzados de

gestión. (Schneider Electric, 2024). No obstante, estas soluciones suelen implicar costos elevados en licencias y servicios de integración, además de depender de proveedores externos para la incorporación de nuevos equipos o modificaciones en la infraestructura.

Ante este panorama, el desarrollo de una solución propia, abierta y escalable representa una alternativa estratégica para Cervecería Barranquilla. Una plataforma interna permitiría eliminar la dependencia de licencias externas, reducir costos asociados al monitoreo energético y consolidar la información en tiempo real con capacidades analíticas y predictivas adaptadas a la operación específica de la planta. Además, facilitaría la identificación de ineficiencias, la optimización del consumo energético y la generación de ahorros económicos sostenibles. Finalmente, una solución de este tipo podría replicarse en otras plantas del grupo o en industrias con necesidades similares, contribuyendo a la competitividad y a los objetivos corporativos de sostenibilidad.

Planteamiento del Problema

Actualmente, Cervecería Barranquilla utiliza el software PME (Power Monitoring Expert) para la captura automática de datos energéticos. Aunque esta herramienta es robusta en la adquisición y almacenamiento de información, ha dejado de recibir soporte técnico por parte del fabricante. Además, al igual que la mayoría de los sistemas comerciales de este tipo, requiere licencias costosas para habilitar la medición de nuevos equipos, lo que limita la velocidad con la que puede ampliarse la capacidad de detección de fugas de energía, ineficiencias y oportunidades de mejora.

El principal factor que impulsa la necesidad de desarrollar una arquitectura propia de monitoreo y control energético es la incompatibilidad operativa de PME con las políticas de ciberseguridad e infraestructura OT de la compañía. El software depende de activadores y mecanismos de licenciamiento complejos que no soportan adecuadamente la restauración de máquinas virtuales en caso de fallas. Esto provoca que, ante incidentes, la recuperación del sistema pueda tardar varios días, generando pérdida de datos y afectando la continuidad del seguimiento energético.

En respuesta a estas limitaciones, se propone el desarrollo de un software propio basado en Django y React, accesible desde cualquier navegador dentro de la red industrial de la compañía. El uso de estos frameworks, junto con librerías de JavaScript y la integración nativa con herramientas de Python como Scikit-learn, permitirá incorporar modelos de inteligencia artificial, escalar la solución a dispositivos móviles y garantizar una plataforma flexible, segura y alineada con las necesidades operativas de la planta.

Objetivos

Objetivo General

Implementar un sistema de procesamiento que integre la captura, estructuración y análisis de datos de consumo energético provenientes de medidores Schneider, permitiendo la generación de informes estratégicos y la implementación de modelos predictivos de consumo eléctrico.

Objetivos Específicos

Analizar los requisitos técnicos y operativos necesarios para la captura automática de datos de consumo energético mediante los medidores Schneider, garantizando su integración eficiente al sistema de procesamiento.

Desarrollar un modelo de estructuración de datos que permita organizar la información obtenida de los medidores Schneider en un formato comprensible y adecuado para la toma de decisiones.

Implementar algoritmos de procesamiento automático que optimicen el análisis de los datos de consumo energético, facilitando la generación de informes detallados y estratégicos.

Evaluar la precisión y confiabilidad de los informes generados a partir del sistema de procesamiento automático, asegurando que los resultados sean representativos y útiles para la toma de decisiones energéticas.

Marco de Referencia

El consumo de energía eléctrica constituye uno de los indicadores más relevantes en la gestión de recursos de cualquier empresa, ya que es el insumo fundamental que sostiene el funcionamiento de la infraestructura moderna. Por esta razón, las organizaciones deben asumirlo como un gasto fijo mensual; sin embargo, también resulta indispensable implementar estrategias de modulación, optimización y mejora continua que permitan reducir dicho costo operativo.

Además, la energía eléctrica está asociada a procesos de generación que pueden implicar distintos niveles de impacto ambiental. En consecuencia, disminuir el consumo no solo representa un beneficio económico, sino que contribuye directamente a la reducción de emisiones y a la mitigación de la contaminación, fortaleciendo el compromiso de la empresa con la sostenibilidad y la protección del entorno.

Gestión y Eficiencia Energética Industrial

La eficiencia energética se entiende como el uso óptimo de la energía para mantener o mejorar la productividad reduciendo consumos innecesarios, un concepto promovido en Colombia por la UPME dentro de sus programas de evaluación industrial. (SIMEC, 2024). La industria manufacturera es especialmente sensible a este indicador, ya que consume el 26,39% de la energía total del país, por lo que cualquier mejora tiene un impacto directo en la competitividad y en la reducción de costos operativos.

Entre los indicadores clave utilizados en la gestión energética se encuentran el consumo específico (kWh/hl), el factor de potencia y la demanda máxima, los cuales permiten identificar ineficiencias y priorizar intervenciones. En el contexto colombiano, la eficiencia energética industrial se ha convertido en un pilar estratégico debido al aumento de los precios de la energía

y a las metas nacionales de descarbonización, lo que incentiva a las empresas a optimizar su consumo para reducir costos y mejorar su desempeño ambiental. (ERCO, 2025).

Infraestructura OT y Ciberseguridad Industrial

La infraestructura OT (Operational Technology) se diferencia de IT (Information Technology) en que está orientada al control de procesos físicos, mientras que IT gestiona información digital. Debido a la creciente digitalización industrial, los sistemas OT enfrentan riesgos cibernéticos cada vez mayores, lo que exige estándares especializados de protección.

La norma ISA/IEC 62443 es el estándar internacional más reconocido para la ciberseguridad en sistemas de control industrial, proporcionando un marco estructurado para proteger la integridad, disponibilidad y confidencialidad de los sistemas OT. (Wallarm, 2025)

En ABI, la ciberseguridad industrial de cada planta se arma basados en normativas internas y externas, estas normativas establecen niveles de acceso a la información diseñados de tal forma que los procesos productivos no sean fácilmente accesibles desde fuera de las instalaciones, por lo tanto, se impide completamente el acceso a internet sobre las máquinas virtuales que componen el control y la automatización de los procesos, dando lugar a escenarios de difícil actualización de sistemas y software, por lo cual es normal encontrar tecnologías ya obsoletas pero que aún cumplen las funciones para las cuales fueron adquiridas.

EMS (Energy Management System)

Un Energy Management System (EMS) es un sistema de software y hardware diseñado para monitorear, analizar y optimizar el consumo energético en instalaciones industriales o comerciales. Su función principal es gestionar de forma inteligente el uso de la energía, integrándose con medidores, sensores y sistemas de automatización para mejorar la eficiencia y reducir costos. Según Industronic, un EMS actúa como el “cerebro” que coordina el uso de

energía en tiempo real, apoyándose en estándares como ISO 50001 para garantizar una gestión eficiente. (Clymer, 2025)

Funciones de Un EMS: Adquisición de Datos, Alarmas, Análisis, Reportes y Predicción

Un EMS moderno incorpora funciones avanzadas que permiten recopilar datos energéticos en tiempo real mediante medidores inteligentes y sensores. De acuerdo con ICO Energía, estas plataformas realizan monitorización, medición, análisis de datos, detección de picos de demanda, integración con sistemas industriales, y generación de reportes para auditorías y cumplimiento normativo.(Icoenergia, 2024).

Limitaciones Típicas de los EMS Comerciales

Aunque los EMS comerciales ofrecen herramientas robustas, suelen presentar limitaciones importantes. Estos sistemas requieren infraestructura especializada, integración compleja y cumplimiento estricto de normativas, lo que incrementa costos y dificulta su escalabilidad en entornos industriales dinámicos. Además, plataformas propietarias como las descritas por Industronic pueden generar dependencia del proveedor (vendor lock-in), ya que su expansión o personalización depende de licencias adicionales, servicios externos y compatibilidad limitada con tecnologías abiertas. Estas restricciones afectan la capacidad de las empresas para integrar nuevos equipos, escalar puntos de medición o adaptar el sistema a arquitecturas OT modernas.(Icoenergia, 2024)

Estado del Arte

En plantas de producción cerveceras, el manejo de indicadores es una actividad fundamental para garantizar eficiencia operativa. Una correcta gestión de los KPI permite obtener ahorros económicos significativos y mejorar los retornos de inversión. Por ello, las empresas suelen apoyarse en herramientas de gestión y desarrollos tecnológicos tanto internos como comerciales para optimizar sus procesos. Dentro de los indicadores claves en las industrias se encuentra los consumos de energía eléctrica, dado que no solo representan costos altos, sino que también hacen parte de requisitos legales ambientales que se deben monitorear y cumplir.

Con el propósito de mejorar la competitividad de las compañías cerveceras se han desarrollado sistemas de gestión energética que monitorean los consumos en los diferentes sistemas y etapas del proceso y a través de la virtualización de esta información permiten gestionar de forma oportuna este KPI. Uno de los pioneros en desarrollo de herramientas de monitoreo de energía es la compañía Schneider, con aportes en este campo desde el 2001 con el lanzamiento de ION Enterprise 3.0 que fue un software de medición y análisis de consumos de energía que ha evolucionado a través de los años hasta llegar al EcoStruxure Power Monitoring Expert (PME) que permite gestionar los consumos de energía eléctrica a nivel industrial y comercial mediante la captura de datos en tiempo real a través de sensores y medidores y además brinda la posibilidad de generar informes alineados al estándar ISO 50001. Otra de las ventajas de esta herramienta es que permite relacionar consumo con producción, es decir kWh/Hl de cerveza y además detecta picos de carga y optimizar la demanda para evitar sanciones.

Otra de las soluciones comerciales más utilizadas en la industria es SIMATIC Energy Management desarrollada por Siemens a mediados del 2010 la cual permite registrar y analizar consumos de energía en cada uno de los procesos de la planta de producción mediante IoT y

analítica avanzada (Siemens, 2026). Esta herramienta integra datos de máquinas, SCADA y sistemas de edificios lo que brinda una información global de los focos de consumo en tiempo real, además permite realizar análisis comparativos entre el volumen producido, con respecto al consumo en cada etapa del proceso. Gracias a este control, la empresa puede evitar sanciones económicas que pueden llegar hasta los €100.000, una cifra que refleja el impacto real que tienen los picos de demanda mal gestionados en los costos operativos industriales.

La gestión energética en la industria cervecera no solo ha sido abordada por grandes compañías de tecnología, sino que también es motivo de estudio y desarrollo en entorno académicos. En 2009 estudiantes de la universidad de Antioquia en su tesis “Sistema de gestión integral de energía para Owens Illinois Peldar” diseñaron un modelo integral de gestión energética para una planta industrial de envases de vidrio, integrando monitoreo en tiempo real, impacto de producción y mantenimiento en la eficiencia. Incluye software para recopilación automática de datos de medidores, reduciendo costos operativos en un 15-20%. Escalable a otras industrias con infraestructuras virtualizadas. (Berrío et al., 2009)

Támara Guevara (2024), en su trabajo *Revisión sistemática de literatura de las innovaciones que favorecen la eficiencia energética en sectores industriales de América Latina y el Caribe*, analiza la eficiencia energética en sectores industriales a partir de la adopción de herramientas digitales sostenibles en América Latina, con énfasis en Colombia, orientadas a la reducción de emisiones y costos en los procesos productivos. A través de la revisión de múltiples estudios, el autor identifica el uso de modelos predictivos, sistemas de monitoreo y dashboards para datos en tiempo real, y plantea una crítica a los altos costos de licenciamiento del software propietario, proponiendo alternativas más flexibles y escalables para infraestructuras industriales, como las del sector cervecero. Asimismo, expone que diversos estudios reportan reducciones en

los costos energéticos asociadas a la digitalización, con rangos que oscilan entre 5 % y 25 %, dependiendo del contexto industrial y del tipo de tecnología implementada; sin embargo, resalta que dichos resultados provienen de estudios individuales y no se consolidan en una clasificación sectorial uniforme, lo que evidencia la necesidad de desarrollar soluciones energéticas adaptadas a las particularidades de cada proceso productivo.

En el informe de práctica empresarial desarrollado por Aguilar Betancur (2019) en la Cervecería Unión, se realizó un diagnóstico energético enfocado en el área de embotellado, con el objetivo de disminuir el consumo de vapor, uno de los principales insumos energéticos del proceso productivo. El estudio identificó que los procesos de lavado de botellas y pasteurización eran los mayores consumidores de vapor, con pérdidas significativas de energía atribuibles a deficiencias en el aislamiento de tuberías y fallas en el sistema de trampas de vapor. Se aplicaron conceptos de termodinámica y transferencia de calor para evaluar estas pérdidas, y se diseñó un experimento multifactorial que analizó variables como temperatura, concentración de soda cáustica, tipo de botella y uso de aditivos. Se determinó que la temperatura y el tipo de botella eran los factores más influyentes en la eficiencia del lavado, mientras que el aditivo tenía un impacto mínimo. A través de la implementación de aislamiento térmico en las tuberías de recirculación de soda y la corrección de problemas en el sistema de calentamiento del tanque 1, se logró un ahorro energético mensual de 63.366,85 MJ, lo que representó una mejora en el indicador de consumo energético de la planta de 0.235 MJ/hL y un ahorro económico mensual de \$1.928.886,91. El proyecto evidenció la importancia de una adecuada metrología, mantenimiento preventivo y análisis detallado de los procesos para optimizar el uso de la energía en entornos industriales, especialmente en la industria cervecera.

El trabajo de grado desarrollado por Pérez Rodríguez (2024) en la Universidad Nacional Abierta y a Distancia (UNAD) se centró en el diseño de un mecanismo para la medición del consumo energético en los equipos y procesos productivos de la empresa Itacol de Occidente, dedicada a la fabricación de alimentos concentrados. La investigación surgió ante la ausencia de políticas específicas en Colombia sobre eficiencia energética, lo que resalta la necesidad de abordar esta problemática a nivel organizacional. El estudio se enfocó en caracterizar los procesos productivos, realizar un inventario de carga de los equipos energéticos y diseñar una matriz energética para evaluar el consumo por procesos. Se utilizaron herramientas como diagramas de Pareto para identificar los equipos con mayor consumo energético en cada etapa del proceso productivo, incluyendo dosificación, molienda, mezcla, peletizado, postengrase, enmelazadora, extrusión y ensacado. El análisis de consumo energético se realizó durante los meses de septiembre, octubre y noviembre, evidenciando una mejora en la eficiencia energética en noviembre, cuando se logró una mayor producción con un menor consumo de energía por tonelada (21.32 kWh/ton frente a 24.4 kWh/ton en septiembre). Además, se diseñó una matriz de oportunidades de eficiencia energética que incluyó propuestas como la instalación de motores de alta eficiencia, medidores de energía por proceso, cambio de PLCs en peletizadoras y mejoras en tableros eléctricos. Estas acciones fueron evaluadas en términos de costo, tiempo, personal requerido y beneficios esperados, permitiendo priorizar las iniciativas con mayor impacto. El estudio concluyó que los procesos de peletizado, molienda y extrusión son los principales consumidores de energía, por lo que se recomendó enfocar los esfuerzos de optimización en estas áreas. Asimismo, se propuso implementar un sistema de monitoreo energético continuo para identificar tendencias y tomar decisiones informadas que contribuyan a la sostenibilidad y competitividad de la empresa.

Campos Avella et al. (2008) en su trabajo “*Modelo de gestión energética para el sector productivo nacional*” proponen el Modelo de Gestión Integral de la Energía (MGIE) como una estrategia orientada a la reducción del consumo energético y de los costos asociados, mediante la adopción de una cultura energética y la implementación de procesos de mejora continua en el sector productivo nacional. El modelo se estructura en tres etapas consecutivas: decisión estratégica, instalación y operación, las cuales guían a la empresa en la implementación de un sistema de gestión que reduce costos operativos e impactos ambientales mediante el ciclo de mejora continua (PHVA). Los autores destacan que el MGIE es una tecnología de baja inversión y alto impacto que permite monitorear indicadores en tiempo real y alinear la estructura organizacional con el uso racional de la energía.

Chaljub Rojas y Troncozo Arenales (2022) desarrollan, en su investigación Sistema de monitoreo de consumo de energía (kWh) y control remoto a través de aplicaciones de mensajería, un prototipo residencial para el monitoreo del consumo eléctrico y el control de cargas mediante tecnologías IoT y la aplicación de mensajería Telegram. El sistema utiliza un medidor multi-función PZEM-004T y un microcontrolador NodeMCU ESP8266 para capturar variables como voltaje y energía activa, permitiendo al usuario visualizar en tiempo real el costo estimado de su factura basado en la regulación colombiana de la CREG. Los resultados experimentales demostraron una alta precisión del dispositivo, con un margen de error del -1.37% en comparación con los medidores comerciales, validando su eficacia para la gestión económica del hogar y la detección de ineficiencias.

Díaz Gómez (2018) presenta el desarrollo de una solución de hardware y software basada en un Lenguaje de Dominio Específico (DSL) para la gestión de los servicios de agua y energía en el hogar. La plataforma integra tecnologías como Arduino, RFID y sensores de efecto Hall

para permitir al usuario administrador programar hábitos de consumo, visualizar gráficas de gasto y recibir alertas de voz educativas sobre el ahorro de recursos. Las pruebas de implementación en entornos residenciales evidenciaron reducciones significativas en el consumo de energía, logrando ahorros de entre el 27% y el 30% al identificar dispositivos de alto gasto, como refrigeradores, y optimizar su uso.

Finalmente, la Cervecería Barranquilla cuenta con un equipo que se encarga de realizar proyectos de transformación en la recolección, manejo y uso de la información de producción, entre sus desarrollos está la integración de los 3 software: Node-Red + InfluxDB + Grafana. La unión de estas tecnologías ha permitido que la planta gestione su información y desarrolle dashboard similares a los planteados en esta Tesis, sin embargo, no están hechos para esta tarea específica y requieren alto conocimiento de programación para realizar seguimientos a la energía. A pesar de lo poderoso que es Node Red, este también es especialmente peligroso para la arquitectura de la red industrial, ya que puede saturar los enlaces de comunicación al generar múltiples enlaces y peticiones a los PLC, por tal motivo la compañía desarrolló SODA. La nueva arquitectura de manejo de información orientada a “One Single Source” permite que pueda ser utilizada para que, en el desarrollo de este proyecto, también puedan extraerse información de los PLC de forma segura, para la realización de análisis y seguimiento de variables además de solo energía.

Marco Teórico

Automatización Industrial y Captura de Datos

La automatización industrial implica el uso de tecnologías para controlar y monitorear procesos sin intervención humana directa. Uno de los pilares de esta automatización es la captura de datos en tiempo real, que permite registrar variables críticas como temperatura, presión, consumo energético o velocidad de producción. Estos datos, cuando se integran adecuadamente en sistemas de información, permiten optimizar procesos, reducir tiempos de respuesta y mejorar la toma de decisiones operativas y estratégicas. (TEGA, 2023).

OPC UA (Open Platform Communications Unified Architecture)

Es un estándar de comunicación industrial diseñado para la interoperabilidad entre dispositivos y sistemas de diferentes fabricantes. A diferencia de versiones anteriores como OPC DA, OPC UA es independiente de la plataforma, seguro, escalable y orientado a objetos. Permite acceder a datos de sensores, PLCs y SCADAs de forma estructurada y segura, lo que lo convierte en una herramienta clave para la digitalización industrial.(OPC Foundation, 2026). En el contexto de este proyecto, OPC UA actúa como puente entre los dispositivos de campo (como medidores o controladores) y el sistema de captura de datos, permitiendo extraer información en tiempo real para su posterior procesamiento.

Bases de Datos NoSQL: InfluxDB

InfluxDB es una base de datos NoSQL diseñada para manejar grandes volúmenes de datos generados en entornos industriales, especialmente aquellos provenientes de sensores y dispositivos que producen información en tiempo real. A diferencia de las bases de datos relacionales, no requiere esquemas rígidos, lo que la hace ideal para datos heterogéneos y en

constante evolución.

Sus principales ventajas incluyen:

- Escalabilidad horizontal: permite almacenar grandes cantidades de datos sin comprometer el rendimiento.
- Alta disponibilidad y consultas flexibles: optimizada para entornos donde se requiere acceso rápido y confiable. (InfluxData Inc, 2026)

Desarrollo de APIs Web con Python

Una API web (Application Programming Interface) permite exponer funcionalidades o datos a través de internet o redes internas, facilitando la comunicación entre sistemas. En este proyecto, se utiliza Python como lenguaje principal y el framework Django para desarrollar una API robusta que conecte los datos capturados vía OPC UA con la base de datos InfluxDB, especializada en el manejo de series temporales. Django, por su arquitectura modular y su sistema integrado de seguridad, es ideal para entornos industriales donde se requiere escalabilidad y control de acceso (Django Software Foundation, 2024). Además, la integración con InfluxDB garantiza un manejo eficiente de grandes volúmenes de datos en tiempo real, optimizando el análisis histórico y la generación de reportes energéticos.

Integración de Tecnologías para la Industria 4.0

La combinación de OPC UA, InfluxDB y APIs web desarrolladas en Python representa un enfoque moderno y flexible para la digitalización industrial. Esta arquitectura permite capturar datos directamente desde el proceso, almacenarlos de forma eficiente y exponerlos en tiempo real para su análisis o visualización. Según Siemens (2026), estas tecnologías son fundamentales para la Industria 4.0, que promueve la interconexión de sistemas, la automatización inteligente y la toma de decisiones basada en datos.

Docker

Docker es un proyecto de software libre creado en 2013 por Solomon Hykes y desarrollado por Docker Inc. Su objetivo es automatizar el despliegue de aplicaciones dentro de contenedores de software. Los contenedores de Docker se basan en las funciones de aislamiento del kernel de Linux, como cgroups y namespaces, que permiten ejecutar procesos de manera independiente sin necesidad de iniciar máquinas virtuales completas. (Red Hat, 2023).

Contenedores y Virtualización

A diferencia de las máquinas virtuales tradicionales, los contenedores de Docker no requieren un sistema operativo independiente, sino que aprovechan el kernel del anfitrión para ejecutar aplicaciones de forma aislada. Esto reduce la sobrecarga de recursos y mejora la eficiencia. Los contenedores incluyen:

- Código de la aplicación.
- Bibliotecas necesarias.
- Configuraciones específicas.

De esta manera, garantizan que la aplicación se ejecute de forma consistente en cualquier entorno. (Stephanie Susnjara & Ian Smalley, 2025).

Funcionalidades Principales

Docker permite a los desarrolladores crear, desplegar, ejecutar, actualizar y administrar contenedores de manera sencilla, lo que lo convierte en una herramienta clave en la arquitectura de microservicios y en entornos de nube híbrida.

Las principales funcionalidades son:

- Portabilidad: ejecutar aplicaciones en cualquier servidor o nube.
- Escalabilidad: desplegar múltiples instancias rápidamente.

- Automatización: simplificar la entrega continua (CI/CD).
- Flexibilidad: integración con Kubernetes y otras plataformas de orquestación.

(Stephanie Susnjara & Ian Smalley, 2025).

Relevancia en la Industria 4.0

Docker se ha convertido en un pilar de la digitalización industrial, ya que facilita la integración de aplicaciones de monitoreo, análisis de datos y sistemas de control en entornos heterogéneos. Su capacidad de empaquetar aplicaciones en contenedores portables lo hace esencial para proyectos de automatización, IoT y ciencia de datos. (Stephanie Susnjara & Ian Smalley, 2025).

React

React es una biblioteca de JavaScript de código abierto creada por Facebook en 2013. Su propósito es facilitar la construcción de interfaces de usuario interactivas y escalables mediante un enfoque basado en componentes reutilizables. Una de sus innovaciones clave es el DOM virtual, que optimiza el rendimiento al actualizar solo las partes necesarias de la interfaz. Además, React adopta un enfoque declarativo y un flujo de datos unidireccional, lo que simplifica la gestión del estado y reduce errores en aplicaciones complejas. (Daniella Coleman, 2023; Jose Angel Saavedra, 2023)

Ventajas Principales:

- Modularidad y reutilización de componentes.
- Mejor rendimiento gracias al Virtual DOM.
- Amplia comunidad y soporte continuo de Meta.
- Flexibilidad para integrarse con otras tecnologías.

Vite

Vite es una herramienta de construcción frontend creada por Evan You (autor de Vue.js). Se lanzó para resolver los problemas de lentitud en compilación y recarga de herramientas tradicionales como Webpack. Vite aprovecha los ESM (ECMAScript Modules) nativos del navegador para ofrecer un entorno de desarrollo ultrarrápido, con Hot Module Replacement (HMR) casi instantáneo. Además, incluye soporte listo para usar para TypeScript, JSX y CSS, y utiliza Rollup para la compilación optimizada en producción.(Barragán, 2024; Vite, 2026)

Ventajas principales:

- Inicio instantáneo y recarga rápida.
- Soporte nativo para TypeScript.
- Arquitectura modular y flexible con plugins.
- Ideal para proyectos modernos de gran escala.

TypeScript

TypeScript es un superconjunto de JavaScript desarrollado por Microsoft en 2012. Su característica más destacada es el tipado estático opcional, que permite detectar errores en tiempo de compilación en lugar de ejecución. También añade funcionalidades avanzadas como interfaces, clases, herencia, enums y genéricos, lo que mejora la organización y escalabilidad del código. TypeScript se compila a JavaScript, garantizando compatibilidad con cualquier navegador o entorno que soporte JS.(Jesús, 2024; typescriptutorial, 2024)

Ventajas principales:

- Tipado estático para mayor seguridad.
- Mejor mantenimiento en proyectos grandes.
- Compatibilidad total con JavaScript.

- Herramienta clave para equipos grandes y proyectos escalables.

Marco Conceptual

La Ciencia de Datos en la Industria

La ciencia de datos se ha convertido en una herramienta clave para la transformación digital de la industria, permitiendo extraer valor de grandes volúmenes de información generados por sensores, máquinas y sistemas de control. En el ámbito energético, su aplicación permite analizar patrones de consumo, detectar ineficiencias y tomar decisiones basadas en evidencia (Adarsh R., 2024)

Sistemas de Monitoreo Energético

Los sistemas de monitoreo energético permiten registrar, visualizar y analizar el consumo de energía en tiempo real. Estos sistemas se basan en la captura de datos provenientes de medidores inteligentes, como los dispositivos de Schneider Electric, que ofrecen información detallada sobre variables eléctricas como voltaje, corriente, potencia activa y factor de potencia. La correcta interpretación de estos datos es fundamental para implementar estrategias de eficiencia energética y cumplir con normativas ambientales y de producción sostenible como ISO 50001 (Schneider Electric, 2023)

Automatización de Informes

La automatización de informes consiste en el uso de herramientas tecnológicas para transformar datos crudos en reportes estructurados, visuales y comprensibles. Esta práctica reduce el error humano, mejora la velocidad de análisis y permite una toma de decisiones más ágil. Tecnologías como Python, bases de datos NoSQL y frameworks web permiten desarrollar aplicaciones que integran captura, procesamiento y visualización de datos en una sola plataforma (Jeremiah Oluseye, 2023).

Indicadores Clave de Desempeño Energético (KPI)

Los indicadores clave de desempeño energético (KPI) permiten evaluar la eficiencia de los procesos industriales en relación con el consumo de energía. Algunos de los KPI más comunes incluyen el consumo específico por unidad producida, el costo energético por la línea de producción y el porcentaje de pérdidas eléctricas. La correcta definición y seguimiento de estos indicadores es esencial para establecer metas de mejora continua y justificar inversiones en eficiencia energética (Siemens, 2014).

Manejo de Datos en Entornos Industriales

Actualmente, gran parte de la información se encuentra reunida en tablas de Excel, que, aunque efectivas, no son escalables para grandes volúmenes de datos. Por ello, se han desarrollado plataformas como Power BI, que permiten agrupar información de múltiples fuentes y facilitar la visualización de datos para la toma de decisiones. Sin embargo, su actualización y mantenimiento puede resultar costoso y limitado a largo plazo. (Microsoft, 2026).

Metodología

La metodología del proyecto se organizó en fases secuenciales e iterativas que abarcan desde el diagnóstico inicial hasta el despliegue y monitoreo continuo del sistema. Cada fase integró actividades técnicas y operativas que permitieron avanzar progresivamente en la construcción de una solución robusta para el manejo eficiente de datos energéticos en Cervecería Barranquilla, asegurando escalabilidad, precisión y adaptabilidad a las necesidades reales de la planta.

Fase 1 (Diagnóstico Técnico y Levantamiento de Requerimientos)

Esta fase se centró en comprender el estado actual del proceso de captura y gestión de datos energéticos. Incluye la revisión de los métodos existentes de medición y reporte, la identificación de los medidores Schneider involucrados —considerando modelos, protocolos de comunicación como Modbus o SNMP y frecuencias de muestreo— y la definición de los indicadores clave que deben ser monitoreados, tales como consumo energético en kWh, picos de demanda y desviaciones operativas. Asimismo, se validaron las necesidades de visualización de la información, la periodicidad de los informes y la compatibilidad con la arquitectura tecnológica futura.

El desarrollo de esta fase se fundamentó en la necesidad de superar las limitaciones del software actualmente utilizado en planta, Power Monitoring Expert (PME), el cual presenta restricciones técnicas y económicas significativas. Dicho sistema opera bajo un esquema de licenciamiento elevado, con costos estimados entre 800.000 y 1.000.000 de pesos por medidor, además de limitar el número de equipos que pueden integrarse por licencia. A nivel de almacenamiento, PME emplea una base de datos SQL que carece de mecanismos eficientes de gestión y depuración automática de datos históricos, lo que conduce a una saturación acelerada

del almacenamiento y obliga a realizar eliminaciones manuales frecuentes de información. En la actualidad, el sistema se ejecuta sobre un entorno Windows Server, con aproximadamente 500 GB de espacio dedicado exclusivamente al almacenamiento de datos energéticos, sin políticas avanzadas de retención.

Estas limitaciones justifican la necesidad de diseñar una arquitectura alternativa de gestión de datos energéticos, más ligera y eficiente, compatible con entornos industriales, sin restricciones en el número de equipos integrables y capaz de adaptarse a distintas plataformas de automatización. En este contexto, se diseñó una primera versión de la arquitectura técnica para la captura automática de datos desde los medidores, orientada a mejorar la escalabilidad, reducir costos operativos y garantizar una gestión más flexible y sostenible de la información energética. Este nuevo sistema para funcionar idealmente tiene como requisitos las siguientes especificaciones de hardware:

- RAM: 4 Gb
- Disco: 100 Gb
- Tarjeta gráfica: Intel UHD Graphics
- Wifi 6
- CPU: Intel Core I5

Como resultado del análisis técnico y de las limitaciones identificadas en esta fase del proyecto, se hizo necesario establecer una estructura de datos robusta que permitiera soportar la nueva arquitectura propuesta y garantizar la correcta gestión de la información capturada por los medidores. De esta forma, la transición hacia la siguiente fase del proyecto estuvo orientada a definir la infraestructura tecnológica requerida, así como también a desarrollar un modelo de datos escalable, flexible y compatible con diferentes protocolos y plataformas industriales. Esto

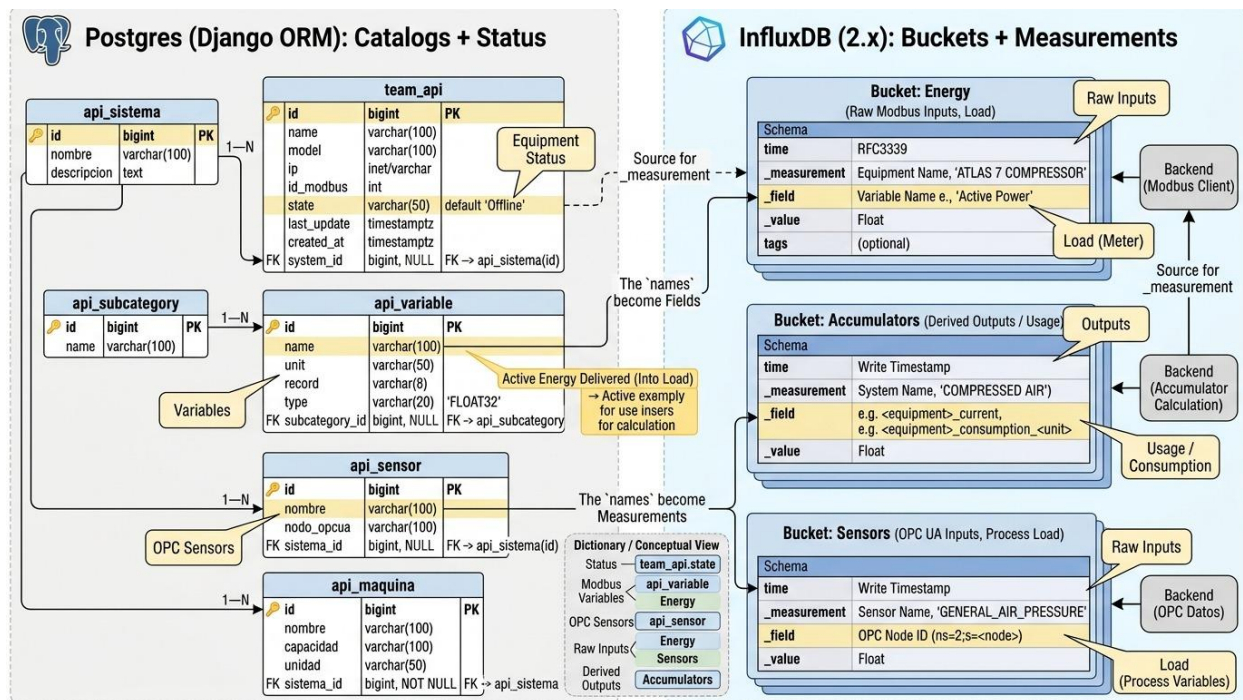
permite pasar de la conceptualización de la solución a la construcción de una base funcional para la captura, almacenamiento y procesamiento mejorado de la información energética.

Fase 2 (Estructuración y Modelado de Datos)

En esta fase se diseñó el modelo de datos que permitió organizar la información proveniente de los medidores en un formato útil para análisis y toma de decisiones. Se seleccionaron las tecnologías base (Django para backend, InfluxDB o PostgreSQL para almacenamiento), se definió la estructura de la base de datos para el registro histórico y se establecieron reglas de normalización y agregación. Posteriormente, se implementaron los scripts de captura automática mediante Modbus, SNMP o API local, validando conectividad, formato de datos y consistencia en la estructuración inicial.

Figura 1

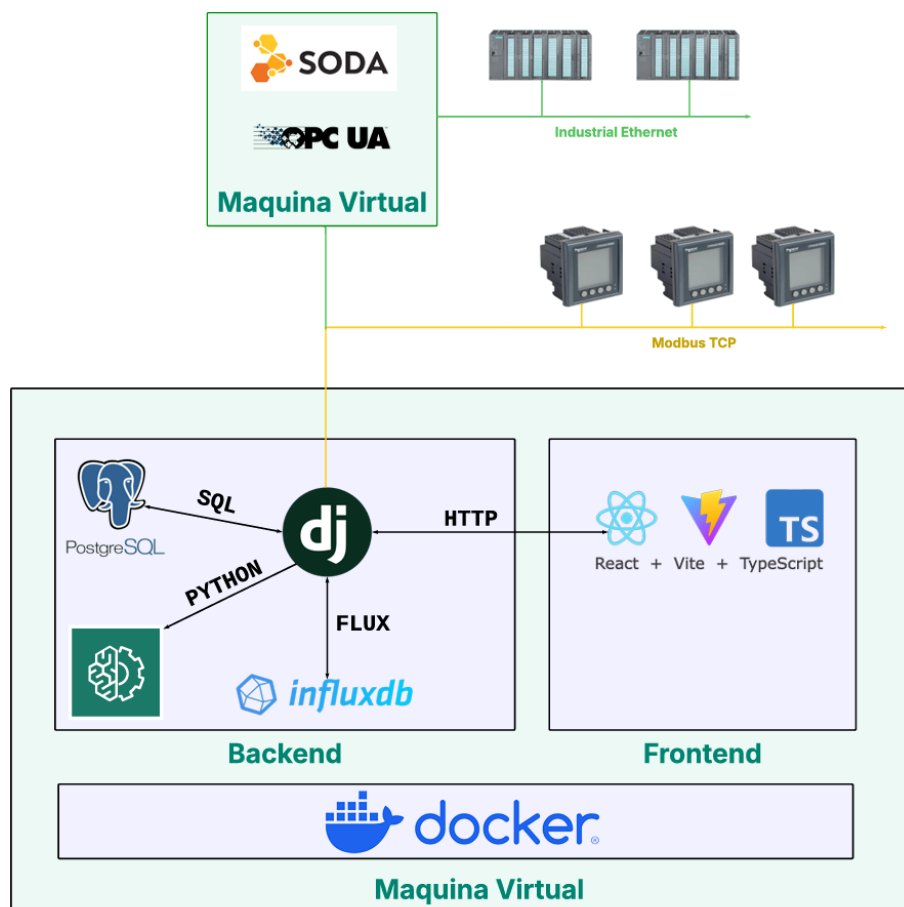
Estructura de Base de Datos



Nota. Se carga la información del archivo "DOCUMENTACION.md" en una plataforma IA y se generó la estructura del funcionamiento de la base de datos

Figura 2

Esquema de Funcionamiento del Proyecto



Arquitectura Basada en Contenedores Docker

El proyecto se ejecutó íntegramente sobre la plataforma Docker, empleando cuatro contenedores principales, cada uno con una función específica para garantizar el correcto funcionamiento y el intercambio eficiente de datos en el sistema.

Backend

El contenedor de backend está desarrollado sobre Django y se encarga de gestionar todas las consultas automáticas a las bases de datos. Además, integra los modelos de inteligencia

artificial, los procedimientos de captura de datos mediante el protocolo Modbus TCP y las consultas a través de OPC. Este backend está optimizado para consumir la menor cantidad posible de recursos, con el objetivo de asegurar el funcionamiento constante de la aplicación.

Las capturas de datos a los medidores de energía se realizaron utilizando hilos (Threads), lo cual permitió efectuar múltiples consultas individuales y simultáneas. Esto previene interrupciones en el servicio y optimiza la recolección de datos. Sin embargo, el número de hilos está limitado y gestionado por procesos automáticos de inicio y cierre, evitando así la sobrecarga del procesador. La configuración óptima de este sistema se alcanzó mediante un proceso de prueba y error, ya que un exceso de hilos provocaba saturación y ralentizaba la aplicación en determinados momentos.

Frontend

El contenedor de frontend utiliza la combinación de ReactJs y ViteJs. Esta unión permite desarrollar aplicaciones escalables que pueden adaptarse a distintas plataformas, como teléfonos móviles, tabletas o televisores. De este modo, se garantizó tanto la fluidez en el funcionamiento como la adaptación automática de la interfaz gráfica, siguiendo además las buenas prácticas que exige TypeScript.

PostgreSQL

Para el almacenamiento de datos de funcionamiento, como nombres de equipos, sistemas, variables, máquinas, etc., se emplea un contenedor basado en PostgreSQL. Este contenedor es responsable de enviar todos los datos de las listas desplegadas al frontend y proporcionar la información requerida por el backend para consultar medidores y demás servicios asociados.

InfluxDB

Dada su alta fiabilidad en el almacenamiento de datos de procesos industriales, todos los datos capturados en planta se almacenaron en el contenedor basado en InfluxDB. Este contenedor se mantuvo en funcionamiento de forma continua y está diseñado para ser ligero y rápido, evitando que tanto el backend como el frontend experimenten demoras al acceder a los datos.

Figura 3

Vista de Funcionamiento en Docker

<input type="checkbox"/>	Name	Port(s)	Memory usage/limit	Image
<input type="checkbox"/>	energymonitorappv30	-	284.58MB / 61.76GB	-
<input type="checkbox"/>	frontend-1	3000:3000	46.54MB / 15.44GB	energymonitorappv30-frontend
<input type="checkbox"/>	backend-1	8000:8000	112MB / 15.44GB	energymonitorappv30-backend
<input type="checkbox"/>	dbInflux-1	8086:8086	94.86MB / 15.44GB	influxdb:2.7
<input type="checkbox"/>	dbPostgres-1	5432:5432	31.18MB / 15.44GB	postgres:17

Nota. Tomado de la pantalla “containers” en Docker desktop

Fase 3 (Desarrollo de Algoritmos de Procesamiento y Análisis Automático)

Esta fase aborda la construcción de la lógica que permitió transformar los datos en información estratégica. Se implementan procesos de normalización, agregación y detección de anomalías, junto con algoritmos de análisis automático para identificar patrones y comportamientos energéticos relevantes. Se desarrolló un modelo predictivo basado en machine learning para estimar consumos futuros en función de variables como la producción, evaluando diferentes configuraciones hasta obtener el menor porcentaje de error. Todo el procesamiento se integró en la plataforma web, junto con dashboards desarrollados en Bootstrap que permiten

visualizar datos históricos, análisis automáticos e informes generados por el sistema. Se realizan pruebas unitarias y de integración con datos reales para validar el desempeño.

Funcionamiento del Backend

El sistema backend desarrollado para la presente investigación implementa tareas de ingesta y monitoreo de datos dentro del mismo proceso de ejecución de Django, utilizando el servidor ASGI/Daphne y gestionando la concurrencia mediante hilos daemon. Esta arquitectura permite mantener procesos activos de captura y supervisión de información en tiempo real, alineados con las necesidades de entornos industriales digitalizados. La solución se estructuró en tres subsistemas principales que operan de manera concurrente:

- Ingesta Modbus, que emplea asincio dentro de un hilo dedicado, acompañado de un segundo hilo encargado de refrescar el catálogo de dispositivos.
- Ingesta OPC UA, que mantiene un bucle persistente en un hilo independiente, complementado por otro hilo para la actualización del catálogo de equipos conectados.
- Módulo de keepalive y estado de equipos, que ejecuta un bucle en un hilo principal y utiliza un ThreadPoolExecutor con capacidad de hasta veinte hilos para garantizar la verificación continua de la disponibilidad de los dispositivos.

En paralelo, los endpoints SSE (Server-Sent Events), implementados mediante StreamingHttpResponse, sostienen coroutines activas por cliente. Estas coroutines consultan la base de datos InfluxDB con una frecuencia controlada por la variable INTERVALO_REFRESCO, lo que asegura la entrega de datos en tiempo real y la actualización constante de las interfaces de usuario.

El proceso de creación y gestión de hilos dentro del sistema backend se inició durante la fase de arranque de la aplicación Django. Este comportamiento se articuló a través de un

mecanismo propio del framework denominado *application registry*, el cual ejecuta métodos especiales al cargar cada aplicación instalada. En particular, el método `ready()` de la clase `ApiConfig`, ubicada en el archivo `backend/api/apps.py`, se activa automáticamente cuando Django inicializa la aplicación correspondiente. Según la documentación oficial, este método está diseñado para ejecutar lógica de inicialización que no debe repetirse ni interferir con el ciclo de carga del servidor (Django Software Foundation, 2024).

Para evitar ejecuciones duplicadas, el sistema implementa controles que verifican:

1. Que el proceso no esté siendo ejecutado por el mecanismo de autorecarga de Django (`RUN_MAIN`).
2. Que la ejecución ocurra únicamente en el hilo principal del proceso.
3. Que el método no haya sido invocado previamente, mediante una bandera interna (`_ready_called`).

Una vez superadas estas validaciones, el método `ready()` invoca la función `iniciar_monitoreo_diferido()` definida en `backend/api/signals.py`, la cual actúa como orquestador del sistema de tareas en segundo plano.

La función `iniciar_monitoreo_diferido()` crea un hilo *daemon* denominado "**Monitoreo-KeepAlive**", que opera como hilo padre del subsistema de monitoreo. Este hilo espera aproximadamente tres segundos para permitir que Django complete su proceso de inicialización. Posteriormente, realiza importaciones diferidas de los módulos responsables de la ingesta y supervisión de datos:

1. `start_modbus_async`, encargado del monitoreo Modbus.
2. `start_opc_async`, encargado del monitoreo OPC UA.
3. `monitoreo_continuo`, encargado del mecanismo de *keepalive*.

Una vez cargados estos componentes, el hilo inicia los procesos Modbus y OPC, cada uno de los cuales genera sus propios hilos internos para mantener ciclos de lectura persistentes. Finalmente, el hilo padre entra en un bucle infinito que ejecuta de manera continua la función `monitoreo_continuo()`, la cual mantiene un ciclo permanente de verificación del estado de los dispositivos conectados. Este patrón de ejecución es consistente con el uso de hilos *daemon* y bucles de larga duración descritos en la biblioteca estándar de Python (Python Software Foundation, 2024).

El sistema también incorpora un mecanismo global para desactivar todas las tareas en segundo plano mediante la variable de entorno `ENABLE_BACKGROUND_TASKS`. Cuando esta se establece en `false`, el sistema evita la creación de hilos y detiene cualquier proceso de monitoreo, aceptando valores equivalentes como `1/true/yes/on` para habilitar su funcionamiento.

Fase 4 (Validación, Despliegue y Monitoreo Continuo)

En esta fase se evaluó la precisión, confiabilidad y utilidad de los informes generados por el sistema. Se realizaron pruebas con usuarios finales —operadores y técnicos de mantenimiento— para validar la calidad de los datos, la claridad de las visualizaciones y la respuesta del sistema ante eventos críticos como picos de carga o desviaciones. Con base en la retroalimentación, se realizaron ajustes funcionales y se documenta la solución. Posteriormente, se desplegó el sistema en un servidor local o en una nube industrial, configurando accesos por roles y capacitando a los usuarios operativos y administrativos. Finalmente, se estableció un proceso de monitoreo continuo para evaluar el impacto del sistema, medir mejoras en tiempos de análisis y detección de anomalías, y aplicar ajustes iterativos que garanticen la precisión y representatividad de los informes en la toma de decisiones energéticas.

Modelo de Rendimiento del Sistema Backend

El modelo de rendimiento propuesto constituye una herramienta analítica diseñada para estimar la carga operativa del sistema de monitoreo industrial. Debido a que el desempeño real depende de factores específicos del entorno (como la cantidad de equipos y sensores, las latencias de comunicación, la capacidad de CPU y memoria, y la velocidad de escritura en la base de datos InfluxDB) no es posible establecer un valor único y universal. En su lugar, se presenta un conjunto de fórmulas reproducibles acompañado de un ejemplo numérico basado en supuestos explícitos. Este enfoque permite que el modelo sea adaptable a distintos escenarios operativos.

Variables del Modelo

El modelo se fundamenta en un conjunto de variables que representan los elementos críticos del sistema:

- E: número de equipos en línea para lectura Modbus.
- N: número total de equipos supervisados mediante keepalive.
- S: número total de sensores OPC UA.

Intervalos de operación:

- T_m : intervalo de lectura Modbus (segundos).
- T_o : intervalo de lectura OPC UA (segundos).
- T_k : intervalo del proceso keepalive (segundos), con un valor por defecto de 8.
- T_r : intervalo de actualización de los endpoints SSE.

Parámetros de latencia:

- B: número promedio de bloques Modbus por equipo.
- L_{mr} : latencia promedio de lectura Modbus por bloque (segundos).

- Lmw: latencia promedio de escritura en InfluxDB por equipo (segundos).
- Lop: latencia promedio de lectura OPC por sensor (segundos).
- Low: latencia promedio de escritura OPC en InfluxDB por sensor (segundos).
- Lk: latencia promedio de verificación keepalive por equipo (segundos).
- Pk: número máximo de hilos disponibles para keepalive, con un valor estándar de

20.

Rendimiento Modbus Tasa de Solicitudes y Carga Sobre InfluxDB

- El número de solicitudes Modbus por ciclo se aproxima mediante:

$$Requests_Modbus_ciclo \approx E * B \quad (1)$$

- La tasa de solicitudes por segundo se estima como:

$$Requests_Modbus_seg \approx (E * B) / Tm \quad (2)$$

- La carga de escritura en InfluxDB se calcula considerando dos escrituras por equipo (energía y acumuladores):

$$Writes_Influx_ciclo \approx 2E \quad (3)$$

$$Writes_Influx_seg \approx (2E) / Tm \quad (4)$$

- El tiempo total del ciclo Modbus está dominado por el equipo más lento:

$$Ciclo_Modbus \approx \max_equipo(\sum Lmr) + overhead \quad (5)$$

Existe riesgo de saturación cuando la relación $2E/Tm$ descrita en la Ecuación 4 es elevada y la latencia de escritura en InfluxDB (Lmw) es significativa, lo que puede provocar acumulación de tareas en el threadpool.

Rendimiento OPC UA: Periodo Efectivo

- El tiempo de lectura y escritura por ciclo se aproxima mediante:

$$Duración_OPC \approx S * (Lop + Low) \quad (6)$$

- El periodo efectivo del ciclo OPC UA es:

$$Periodo_OPC_efectivo \approx Duraci3n_OPC + T_o \quad (7)$$

- La tasa de escrituras por segundo en InfluxDB se estima como:

$$Writes_OPC_seg \approx S / Periodo_OPC_efectivo \quad (8)$$

Rendimiento del Proceso Keepalive

- Con un m3ximo de Pk hilos, la duraci3n del ciclo se aproxima mediante:

$$Duraci3n_Keepalive \approx ceil(N / Pk) * Lk \quad (9)$$

- El periodo efectivo del ciclo es:

$$Periodo_Keepalive_efectivo \approx max(Tk, Duraci3n_Keepalive) \quad (10)$$

Carga Generada por SSE (Server-Sent Events)

Sea:

- C1: n3mero de conexiones SSE al endpoint /graficas_update (frecuencia: cada Tr).

- C2: conexiones SSE a /graficas_update_sistemas (frecuencia: cada 12Tr).

- C3: conexiones SSE para predicciones (frecuencia: cada 12Tr).

- La carga total de consultas a InfluxDB por segundo se aproxima mediante:

$$Qps_SSE \approx \left(\frac{C1}{Tr}\right) + \left(\frac{C2}{12Tr}\right) + \left(\frac{C3}{12Tr}\right) + \dots \quad (11)$$

Ejemplo Num3rico

Debido a que no se ha completado el cargue total de equipos, a3n no es posible calcular exactamente los datos, sin embargo, se ofrece este ejemplo de gu3a.

Se consideran los siguientes valores:

- E = 30 equipos en l3nea.

- B = 3 bloques Modbus por equipo.
- Tm = 5 segundos.
- S = 80 sensores OPC UA.
- To = 5 segundos.
- N = 50 equipos supervisados por keepalive.
- Pk = 20 hilos disponibles.

Si la latencia promedio por bloque Modbus es Lmr = 0.25 segundos, el tiempo por equipo es aproximadamente 0.75 segundos, pudiendo aumentar según el equipo más lento.

Cálculos de Ejemplo

- Solicitudes Modbus por segundo: $(30 * 3) / 5 = 18 \text{ req/s}$
- Escrituras Modbus en InfluxDB por segundo: $(2 * 30) / 5 = 12 \text{ writes/s}$
- Para OPC UA:
 - Si Lop + Low = 0.03 segundos:
 - Duración OPC = $80 * 0.03 = 2.4 \text{ s}$
 - Periodo OPC efectivo = $2.4 + 5 = 7.4 \text{ s}$
 - Writes OPC_seg $\approx \frac{80}{7.4} \approx 10.8 \frac{\text{writes}}{\text{s}}$
- Para keepalive:
 - Si Lk = 0.3 segundos:
 - Duración Keepalive = $\text{ceil}(50 / 20) * 0.3 = 0.9 \text{ s}$
 - Como Tk = 8 segundos, el sistema opera con un margen amplio, descansando

aproximadamente

Funcionamiento del Frontend

La arquitectura del frontend está diseñada para brindar una estructura sólida y flexible, facilitando tanto el desarrollo como el mantenimiento de la aplicación. A continuación, se describen los componentes clave y las tecnologías empleadas.

Tecnologías Principales

- **React + TypeScript (Vite):** La base de la aplicación está construida con React y TypeScript, utilizando Vite como herramienta de desarrollo para obtener una experiencia rápida y moderna.
- **React Router:** Se emplea para la gestión de rutas y el layout, permitiendo la navegación entre distintas secciones de la aplicación de manera eficiente.
- **Axios/Fetch para HTTP:** Estas librerías se utilizan para realizar peticiones HTTP al backend, asegurando la comunicación y transferencia de datos.
- **Server-Sent Events (EventSource):** Integrados para el manejo de eventos en tiempo real, proporcionando actualizaciones instantáneas desde el servidor.
- **Chart.js (+ chartjs-plugin-zoom):** Utilizado para la generación de gráficas interactivas, permitiendo a los usuarios visualizar datos y analizar información de manera dinámica.

Punto de Entrada

- **Archivo:** `frontend/src/main.tsx`
- Este archivo es responsable de montar la aplicación en el elemento `#root`, utilizando `StrictMode` para garantizar buenas prácticas y detectar posibles problemas durante el desarrollo.

Componente Raíz y Ruteo

- Archivo: *frontend/src/App.tsx*
- En este componente se crea el Router y se declaran las rutas principales de la aplicación.
- Se mantiene el estado de la interfaz de usuario, como el sidebar colapsado (*collapsed*), y se pasa dicho estado al componente de Layout.

Rutas Activas

Las rutas definidas en App.tsx permiten acceder a diferentes vistas:

/dashboard → *Dashboard*

/equipos → *Equipos*

/informes → *Informes*

/predictivo → *Predictivo*

/ → *Redirige a /dashboard*

*** → *“Página no encontrada”*

Layout Principal

- Archivo: *frontend/src/Pages/layout/Layout.tsx*
- Este componente se encarga de renderizar la Sidebar y el contenedor principal donde se muestran las vistas.
- Incluye un `Outlet` para renderizar las rutas hijas, permitiendo una estructura jerárquica y modular.
- Proporciona el contexto `{collapsed, setCollapsed}` a través del Outlet, facilitando la gestión del estado de la interfaz en las vistas secundarias.

Comunicación Entre el Navegador y el Backend: HTTP y SSE

La interacción entre el frontend y el backend, como se muestra en la Ilustración 1, se realiza principalmente a través de endpoints bajo la ruta **/api/**, empleando tanto el Django REST Framework como vistas personalizadas. Esta comunicación incluye tanto operaciones bajo el protocolo HTTP tradicional como actualizaciones en tiempo real mediante Server-Sent Events (SSE).

Endpoints REST/HTTP Principales. Los endpoints RESTful definidos en *backend/api/urls.py* permiten la gestión y consulta de los principales recursos del sistema, implementando operaciones CRUD (Crear, Leer, Actualizar y Eliminar) y funcionalidades especializadas. Entre los endpoints más relevantes se encuentran:

- **/api/csrf/**: obtiene la cookie CSRF necesaria para la seguridad de las peticiones.
- **/api/equipos/**: permite el CRUD de equipos utilizando el router de DRF.
- **/api/variables/**: operaciones CRUD sobre variables.
- **/api/subcategorias/**: CRUD de subcategorías.
- **/api/maquinas/**: CRUD de máquinas.
- **/api/sistemas/**: CRUD de sistemas.
- **/api/sensores/**: CRUD de sensores.

Además, existen endpoints personalizados para la obtención y consulta de datos específicos:

- **/api/get_data_influx/**: consulta el histórico de datos de InfluxDB para ser utilizados en gráficas.
- **/api/get_last_hour_initial/**: provee la semilla o los datos de la última hora para inicializaciones.

- **/api/get_query_inform/**: genera informes en formato JSON o archivo descargable.
- **/api/get_acumulados/**: entrega datos acumulados relevantes para análisis predictivo o series temporales.

Endpoints SSE (Tiempo Real). Para la comunicación en tiempo real, se utilizaron varios endpoints SSE definidos en *backend/api/urls.py*. Estos permiten el envío continuo de datos actualizados desde el backend al navegador, facilitando la actualización dinámica de la interfaz sin necesidad de recargar la página.

- **/api/graficas_update/**: envía puntos en vivo por equipo o variable.
- **/api/graficas_update_sistemas/**: transmite puntos por sistema, empleando el índice minuto del día.
- **/api/stream_predicciones/**: proporciona series predichas para cada sistema.
- **/api/stream_predicciones_dia/**: actualiza el total estimado del día por sistema.
- **/api/stream_sensores/**: transmite en tiempo real los valores de sensores por sistema.

Estructura de los Mensajes SSE en el Frontend. En el frontend, los mensajes SSE recibidos presentan una estructura típica donde **event.data** contiene un objeto JSON. El payload esperado tiene el siguiente formato:

```
{ "tipo": "grafico_actualizado", "contenido": ... }
```

Esta estructura permite identificar el tipo de evento recibido y acceder al contenido específico para su procesamiento en la interfaz.

Resultado Final

La implementación del sistema desarrollado culmina en una interfaz web que integra todas las funcionalidades necesarias para la captura, visualización, análisis y predicción del consumo energético en la planta. El resultado final se materializa en un conjunto de módulos diseñados para facilitar la interacción del usuario con los datos provenientes de los medidores Schneider, permitiendo no solo consultar información histórica y en tiempo real, sino también generar informes personalizados y realizar estimaciones predictivas del comportamiento energético diario. Cada sección de la plataforma fue construida para responder a necesidades operativas reales de la Cervecería Barranquilla, priorizando la claridad visual, la adaptabilidad a distintos dispositivos y la automatización de procesos que anteriormente requerían intervención manual. A continuación, se describen los componentes principales del sistema y su funcionamiento dentro del entorno de operación.

Dashboard

El dashboard es la página principal que se abre al iniciar la aplicación, en esta interfaz es posible visualizar los datos previamente registrados por el programa, por medio de la selección del equipo, de la variable y el rango de tiempo que se requiera ver, el cual va desde los últimos 5 minutos hasta días y semanas completas. Los dashboards están configurados para mostrar la información más legible posible, por tanto, cambiara automáticamente entre mostrar minutos, horas o días.

Inicialmente se contempla la utilización de 4 graficas seleccionables, sin embargo, según se requiera por la empresa, en el futuro se pueden agregar mas gráficos ya sean del mismo tipo o de otro modelo de gráficos, pero ya se contaría como un desarrollo adicional, para efectos del presente proyecto se plantearon 4 dashboards lineales con los cuales se puede trabajar tanto en el

computador, como en un celular debido a la capacidad de redibujarse el layout según el tamaño de la ventana.

Figura 4

Página/Dashboard



Nota. Tomado de la ventana “dashboard” en el nuevo sistema desarrollado.

Componentes de la página:

1. Selector de equipo: Sirve para escoger el equipo que se quiere visualizar, este listado incrementa con cada equipo agregado.
2. Selector de Variable: Listado de variables que pueden escogerse, este listado no permite cambios.
3. Activar refresco automático: Al activar esta opción se recibirán datos cada 5 segundos y se irán agregando al final de la gráfica.

4. Seleccionar rango de fechas: En esta opción se despliega el calendario en donde se pueden escoger rangos de días o usar las opciones de tiempo preestablecidas.

Equipos

Figura 5

Página de Manejo de Equipos

The image shows two screenshots of a web application interface for managing equipment. The left screenshot displays a table with columns: Nombre, Modelo, IP, ID Modbus, and Estado. The right screenshot shows a modal window titled 'Agregar Equipo' with input fields for Nombre, Modelo, IP, and ID Modbus, and buttons for Guardar and Cancelar.

Nombre	Modelo	IP	ID Modbus	Estado
SUBESTACION	PM5500	10.0.7.1	1	Offline
COMPRESOR ATLAS 8	PM5500	10.2.2.30	1	Offline
PET 18T LINEA 1	PM5500	10.0.7.130	1	Offline
COMPRESOR LINEA PET	PM5500	10.0.7.123	1	Offline
POTENCIA 5	PM5500	10.0.7.126	1	Offline
POTENCIA 3 LINEA PET	PM5500	10.0.7.124	1	Offline
COM 48 COCINAS	PM5500	10.0.7.2	1	Offline
COMPRESOR CENTAC	PM5500	10.0.7.216	3	Offline
COMPRESOR KAESER 1	PM5500	10.0.7.216	1	Offline
TOTALIZADOR PET 440V	PM5500	10.0.7.122	1	Offline
CASA	PM5500	192.168.80.250	1	Offline
SOPLADOR LINEA PET	PM5500	10.0.7.125	1	Offline
COMPRESOR ATLAS 7	PM5500	10.0.7.187	1	Offline
COMPRESOR KAESER 2	PM5500	10.0.7.192	1	Offline

Nota. Tomado de la ventana “Equipos” en el proyecto desarrollado.

Componentes de la página:

Tabla de equipos: En esta tabla se imprimen todos los equipos agregados, se puede ver el nombre, el modelo del medidor, la IP, el ID modbus (Importante para comunicaciones con equipos que están conectados a otros por modbus RTU) y el estado el cual se cambia a “Online” o “Offline” dependiendo del estado del equipo.

1. Agregar equipo: Sirve para desplegar el formulario modal de agregar equipo.
2. Modal de agregar equipo: Acá se diligencian todos los datos necesarios para una correcta comunicación con el equipo en red.

Informes

Figura 6

Página/informes

Generar Informe

Nombre del Informe:

Rango de fechas:

Equipos: 0 seleccionados

Variables: 0 seleccionadas

Formato: XLSX CSV

Nota. Tomado de la ventana “Informes” del proyecto desarrollado.

Figura 7

Modal de Selección de Equipos y Modal de Selección de Variables

Equipos / Medidores

0 de 14 seleccionados

Nombre	Modelo	IP	ID Modbus	Estado
SUBESTACION	PM5000	10.0.7.1	OFFLINE	1
COMPRESOR ATLAS 8	PM5500	10.2.2.50	OFFLINE	1
PET TBT LINEA 1	PM5500	10.0.7.130	OFFLINE	1
COMPRESOR LINEA PET	PM5500	10.0.7.123	OFFLINE	1
POTENCIA 5	PM5500	10.0.7.126	OFFLINE	1
POTENCIA 3 LINEA PET	PM5500	10.0.7.124	OFFLINE	1
CCM 440 COCINAS	PM5000	10.0.7.2	OFFLINE	1
COMPRESOR CENTAC	PM5300	10.0.7.216	OFFLINE	3
COMPRESOR KAESER 1	PM5500	10.0.7.216	OFFLINE	1
TOTALIZADOR PET 440V	PM5500	10.0.7.122	OFFLINE	1
CASA	PM5000	192.168.80.250	OFFLINE	1

Variables por Subcategoría

0 de 61 seleccionadas

- Current ▶
- Voltage ▼
 - Power Factor A
 - Power Factor B
 - Power Factor C
 - Power Factor Total
 - Displacement Power Factor A
 - Displacement Power Factor B
 - Displacement Power Factor C
 - Voltage A-B
 - Voltage B-C
 - Voltage C-A
 - Displacement Power Factor Total
 - Voltage L-N Avg
 - Voltage C-N
 - Voltage B-N
 - Voltage A-N
 - Voltage L-L Avg
- Accumulated Energy ▶
- Power Factor ▶
- Frequency ▶

Nota. Tomado de las ventanas pantalla “Generación de informes” del proyecto desarrollado.

Componentes de la página:

1. Nombre del informe: Este componente sirve para establecer el nombre que tendrá el informe a generar.
2. Rango de fechas: Se despliega al dar click en el icono de calendario un rango de fechas para generar el informe.
3. Equipos: En este botón se despliega el modal de la Ilustración 7, en donde se puede seleccionar todos los equipos que harán parte del informe.
4. Variables: Con este botón se despliega el modal de la Ilustración 8, en donde se pueden seleccionar tantas variables como se requieran, todas estas separadas por categoría.
5. Formato: Es posible seleccionar si se requiere que el informe sea usado como una base de datos (CSV) o si se requiere que sea directamente una hoja de excel (XLSX).
6. Generar informe: Finalmente se da click en este botón para generar el informe con lo seleccionado anteriormente.

Predictivo

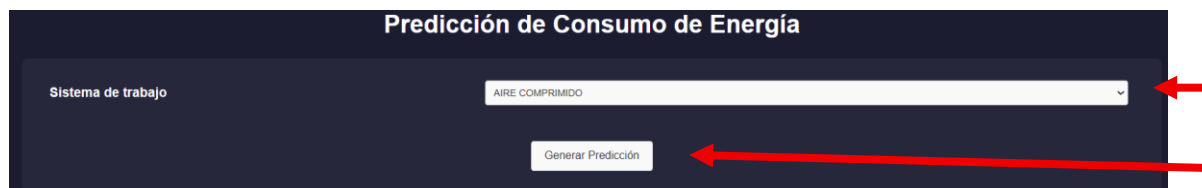
Finalmente, en la página “Predictivo” se puede hacer uso del modelo entrenado basado en RandomForest, para predecir un aproximado del consumo eléctrico total del sistema en el día, esto sirve para que se pueda ajustar la modulación del sistema con base a la meta diaria de consumo, por lo tanto, es posible ver como las decisiones de control del sistema pueden afectar en tiempo real el consumo final. Es importante tener en cuenta que para poder predecir se requiere información, por lo tanto, se requieren al menos 30 minutos de datos para intentar predecir un valor.

Para efectos del alcance del proyecto, se utilizó el sistema de aire comprimido de la planta, el cual cuenta con todos los equipos de medición necesarios para dar un aproximado del

consumo de todo el sistema, además de contar con una variable que es de la cual se desprende la utilización de los equipos, siendo perfecto para realizar análisis. En la medida que la planta lo requiera es posible entrar modelos para otro tipo de sistemas.

Figura 8

Página/Predictivo



Predicción de Consumo de Energía

Sistema de trabajo: AIRE COMPRIMIDO

Generar Predicción

Nota. Tomado de la página “Predicción de consumo de energía” del proyecto desarrollado.

Figura 9

Gráfico de predictivo



Nota. Tomado de la página “Generar predicción” del proyecto desarrollado.

Componentes de la página:

1. Sistema de trabajo: Este listado compone todos los sistemas entrenados para predicción de consumos de energía, para el alcance de este proyecto, solo se trabajó con el sistema de aire comprimido.
2. Generar predicción: Este botón desplegará no solo una gráfica de tiempo con los datos que se han guardado en todo el día, sino que establecerá enlaces automáticos con el baekend para en tiempo real ir acomodando al final de la gráfica todos los valores entrantes.
3. Gráfico de predicción: Una vez generado el grafico, la sumatoria de todos los consumos por minuto de los equipos que componen el sistema de aire comprimido se comenzará a imprimir automáticamente, y una vez el programa tenga suficientes datos dará un valor de consumo estimado para el resto de día y dibujará una gráfica azul que represan todos los valores predichos.

Cronograma Ejecutado

Tabla 1

Cronograma Ejecutado

Fase	Actividad	Mes				
		Sep	Oct	Nov	Dic	Ene
Diagnóstico y requerimientos	Revisión de procesos, entrevistas, identificación de equipos, definición de indicadores	x				
Diseño técnico de la solución	Arquitectura de la app, selección de tecnologías, diseño de endpoints y base de datos		x			
Desarrollo e integración	Scripts de captura, integración con medidores, interfaz web, pruebas unitarias		x	x		
Validación funcional	Pruebas con usuarios, simulación de eventos,			x	x	

Fase	Actividad	Mes				
		Sep	Oct	Nov	Dic	Ene
	ajustes, documentación técnica					
Despliegue y capacitación	Instalación en planta, configuración de accesos, capacitación, entrega de manuales					x
Monitoreo y mejora continua	Seguimiento de uso, recolección de métricas, propuestas de mejora, evaluación de impacto					x

Recursos Usados**Tabla 2***Recursos Usados*

Recurso	Descripción	Presupuesto
Equipo Humano	Estudiante de especialización en ciencia de datos	\$0
Equipos y Software	Computador, licencia vmware, licencia OPC.	\$7.000.000
Viajes y Salidas de Campo	Inventario de equipos de medición de energía.	\$0
Materiales y suministros	Documentación técnica de equipos schneider	\$0
Bibliografía	Acceso a las bases de datos Unad	\$0
Total		\$7.000.000

Conclusiones

El desarrollo del proyecto cumple con los objetivos planteados y genera el seguimiento de datos esperado, siendo una herramienta alternativa poderosa para el análisis de consumos de energía de la Cervecería, pero que además tiene el alcance de migrarse a cualquier operación que siga los mismos estándares de red y referencia de equipos. Adicionalmente, se obtienen los siguientes resultados y conclusiones:

El sistema desarrollado puede en el futuro reemplazar completamente el software de gestión de energía que utiliza la Cervecería actualmente.

La comunicación entre los contenedores Docker fue compleja de lograr, debido a que estas se comunican a través de diferentes lenguajes y protocolos (HTTP, Flux, SQL, Modbus), lo cual requirió la mayor cantidad de tiempo de desarrollo.

La implementación del sistema demostró que la combinación de tecnologías como OPC UA, InfluxDB y contenedores Docker permite una captura, almacenamiento y procesamiento eficiente de datos energéticos en tiempo real, alineándose con los principios de la Industria 4.0 y mejorando la digitalización en entornos industriales.

La incorporación de variables de proceso y el uso de algoritmos de aprendizaje automático, como RandomForest, permiten estimar con alta precisión (menor al 5% de error) el consumo energético, lo que abre la posibilidad de implementar estrategias proactivas para la eficiencia energética en la planta cervecera.

El diseño basado en contenedores y arquitecturas modulares garantiza la escalabilidad del proyecto, permitiendo futuras mejoras como la integración de más equipos, nuevas variables de proceso y funcionalidades avanzadas sin comprometer la estabilidad del sistema.

La adopción de este sistema favorece la cultura de mejora continua dentro de la Cervecería, al facilitar la identificación oportuna de desviaciones energéticas y promover la toma de decisiones basada en datos.

El enfoque modular y la interoperabilidad con diferentes protocolos y equipos posicionan la solución como una plataforma flexible, capaz de adaptarse a futuras tecnologías sin requerir grandes inversiones en infraestructura.

El proyecto sienta las bases para la integración de sistemas de gestión ambiental, permitiendo que la monitorización energética se vincule con objetivos de sostenibilidad y reducción de la huella de carbono.

Mejoras Futuras

El proyecto fue desarrollado basado en la ejecución de los objetivos planteados, sin embargo, tiene el potencial de crecer más allá de lo inicialmente planteado, por lo que se plantean las siguientes mejoras que podrían realizarse a futuro:

- **Ampliar la variedad de gráficos:** Actualmente se cuentan con 4 gráficos de líneas en la página de Dashboard, sin embargo, esto puede ser migrado a otro tipo basándose en los objetivos y necesidades del usuario final.
- **Alinear las variables con base al modelo:** Uno de los desafíos mas grandes en el desarrollo del proyecto, fue la comunicación eficiente con los medidores de energía de la serie PM5000 de la marca Schneider, pero adicionalmente, en el transcurso del proyecto se identificó que hay equipos de la misma marca que no tienen tablas modbus iguales, por lo tanto, la extracción de datos del programa actual está limitada a medidores entre las series PM5300 y PM5600, para lo cual, se requiere expandir las capacidades y configurar la aplicación para que incorpore o remueva variables con base a la serie y establezca tablas modbus según se requiera.
- **Mas modelos entrenados:** Para el alcance del proyecto se entrenó un modelo para predecir el consumo del sistema de aire comprimido, sin embargo, en el futuro puede entrenarse modelos para otros sistemas de la planta que requieran aplicar mejoras de modulación para lograr ahorros de energía.
- **Alertas:** La visualización automática y el envío de alertas funcionales mejorarían la capacidad de reacción ante eventos críticos, contribuyendo a la optimización de procesos y a la reducción de costos operativos.

Bibliografía

- Adarsh R. (2024, diciembre 18). *Digital Transformation in Energy: Top 10 Technologies [2025]*.
<https://www.startus-insights.com/innovators-guide/digital-transformation-in-energy/>
- Aguilar D. (2019, enero 14) *Diagnóstico para procesos de alto consumo energético en el área de embotellado de la cervecería unión*. Recuperado el 20 de febrero de 2026, de
<https://bibliotecadigital.udea.edu.co/entities/publication/cfe3aaa1-1684-4259-a909-2054f735c6cb>
- Alfredo Barragán. (2024, abril 24). *Vite: Revolucionando el desarrollo frontend de la web moderna | OpenWebinars*. <https://openwebinars.net/blog/vite-desarrollo-frontend/>
- Berrío, J., Felipe, R., & Botero, U. (2009). *Diseño e implementación de un sistema de gestión integral de energía para la empresa Owens Illinois Peldar planta Zipaquirá*.
- Campos Avella, J. C., Lora Figueroa, E. D., Tovar Ospino, I., Prias Caicedo, O. F., Quispe Oqueña, E. C., & Vidal Medina, J. R. (2007). *Modelo de gestión energética para el sector productivo nacional*. Universidad Tecnológica de Bolívar; Universidad del Atlántico; Universidad Autónoma de Occidente.
- Chaljub Rojas, E. O., & Troncozo Arenales, B. D. (2022). *Sistema de monitoreo de consumo de energía (KWH) y control remoto a través de aplicaciones de mensajería [Proyecto de grado, Universidad de la Costa]*.
- Daniella Coleman. (2023, agosto 30). *Revisión del framework React.js: ventajas, desventajas y casos de uso*. <https://serverspace.io/es/about/blog/review-of-the-react-js-framework-advantages-disadvantages-and-use-cases/>

- Díaz Gómez, L. (2018). *Prototipo para implementación de sistema de gestión y monitoreo de consumo de servicios domiciliarios*. SENA; Corporación de Educación Tecnológica Colsubsidio; Universidad Nacional de Colombia
- Django Software Foundation. (2024). *The web framework for perfectionists with deadlines | Django*. Recuperado el 9 de enero de 2026, de <https://www.djangoproject.com/>
- ERCO. (2025, diciembre 12). *Análisis Eficiencia Energética Industrial Colombia 2026. 7 estrategias clave para eficiencia energética industrial*. <https://blog.erc.energy/analisis-eficiencia-energetica-instalaciones-industriales-eficiencia-energetica/>
- Icoenergia. (2024, enero 2). *Sistemas de gestión de la energía sostenibles. Sistemas de Gestión de la Energía Energy Management System (EMS)*. Recuperado el 9 de enero de 2026, de <https://icoenergia.com/servicios/eficiencia-y-ahorro-energetico/sistemas-de-gestion-de-la-energia-energy-management-system-ems/>
- Clymer, S. (2025, septiembre 30). *Energy Management System: qué es, beneficios y aplicaciones*. Grupo Industronic; Industronic. <https://grupoindustronic.com/energy-managment-system/>
- InfluxData Inc. (2026). *InfluxDB Platform Overview*. <https://www.influxdata.com/products/influxdb-overview/>
- Jeremiah Oluseye. (2023, marzo 16). *How to Generate Automated Reports from a SQL Database Using Python*. <https://www.freecodecamp.org/news/automating-report-generation-from-sql-databases-using-python/>
- Jesús. (2024, octubre 18). *Ventajas y Desventajas de Typescript*. <https://www.dongee.com/tutoriales/ventajas-y-desventajas-de-typescript/>

- Jose Angel Saavedra. (2023, julio 17). *Qué es React y para qué sirve: ventajas y desventajas, casos de uso, características, quién lo utiliza*. <https://ebac.mx/blog/que-es-react>
- Mateo Medina Ariza. (2024, septiembre 30). *Tarifas de energía subieron 15,3% anual entre 2021 y 2024, la cifra más alta del siglo*. [larepublica.co. https://www.larepublica.co/economia/tarifas-electricas-aumentaron-15-3-anual-entre-2021-y-2024-la-cifra-mas-alta-del-siglo-3964636](https://www.larepublica.co/economia/tarifas-electricas-aumentaron-15-3-anual-entre-2021-y-2024-la-cifra-mas-alta-del-siglo-3964636)
- Microsoft. (2026). *Manufacturing | Power BI*. Recuperado el 9 de enero de 2026, de <https://www.microsoft.com/en-us/power-platform/products/power-bi/industry/manufacturing>
- OPC Foundation. (2026). *Unified Architecture - Landingpage - OPC Foundation. Unified Architecture – Landingpage*. Recuperado el 9 de enero de 2026, de <https://opcfoundation.org/about/opc-technologies/opc-ua/>
- Perez, J. (2024, enero 10). *Diseño de un mecanismo de medición sobre el consumo energético en los equipos y procesos de producción de la empresa Itacol de Occidente*. Edu.co Recuperado el 20 de febrero de 2026, de <https://repository.unad.edu.co/jspui/handle/10596/60603?locale=es>
- Red Hat. (2023, enero 20). *¿Qué es Docker y cómo funciona? Ventajas de los contenedores Docker*. <https://www.redhat.com/es/topics/containers/what-is-docker>
- Schneider Electric. (2023, julio 24). *ISO-50001 EN | Schneider Electric*. https://www.se.com/us/en/download/document/ISO-50001_EN/
- Schneider Electric. (2024, diciembre 4). *Optimización energética en la industria: cómo el monitoreo inteligente reduce costos y mejora la eficiencia | Schneider Electric Colombia. Optimización energética en la industria: cómo el monitoreo inteligente reduce costos y*

- mejora la eficiencia*. <https://www.se.com/co/es/about-us/newsroom/news/press-releases/optimizaci%C3%B3n-energ%C3%A9tica-en-la-industria-c%C3%B3mo-el-monitoreo-inteligente-reduce-costos-y-mejora-la-eficiencia-67508479a1839a3dd30b1088>
- Siemens. (2014). *Managing Energy Using Key Performance Indicators Siemens Retail & Commercial Systems*. www.siemens.com/sitecontrols
- Siemens. (2026). *¿Qué es la Industria 4.0? ¿Qué es la Industria 4.0?* Recuperado el 9 de enero de 2026, de <https://www.siemens.com/co/es/compania/areas-tematicas/industria-40.html>
- Siemens. (2026). *SIMATIC energy management software - Siemens Global. SIMATIC Energy Management software*. Recuperado el 9 de enero de 2026, de <https://www.siemens.com/global/en/products/automation/industry-software/automation-software/energymanagement.html>
- SIMEC. (2024, noviembre 25). *Eficiencia en la Industria*. UPME. Recuperado el 9 de enero de 2026, de <https://www.upme.gov.co/simec/eficiencia-energetica/eficiencia-en-la-industria/>
- Stephanie Susnjara, & Ian Smalley. (2025, noviembre 26). *¿Qué es Docker?*. IBM. Recuperado el 9 de enero de 2026, de <https://www.ibm.com/mx-es/think/topics/docker>
- Támara Guevara, C. A. (2024). *Revisión sistemática de literatura de las innovaciones que favorecen la eficiencia energética en sectores industriales de América Latina y el Caribe* [Trabajo de grado, Universidad Industrial de Santander]. Repositorio Institucional UIS. <https://noesis.uis.edu.co/items/e51d314a-12bc-4e32-b63a-b82bfc968513>
- TEGA. (2023, junio 5). *Qué es la automatización industrial*. TEGA Industrial. <https://tegaindustrial.com/blog/automatizacion-industrial/>

typescriptutorial. (2024, abril 10). *Ventajas y desventajas de TypeScript | Que Necesitas Saber!!!*

Recuperado el 9 de enero de 2026, de <https://typescriptutorial.com/es/ventajas-y-desventajas/>

Vite. (2019). *Herramienta frontend de próxima generación*. Recuperado el 9 de enero de 2026,

de <https://es.vite.dev/>

Wallarm. (2025, marzo 30). La serie de normas ISA/IEC 62443: guía completa ! — Wallarm.

La serie de normas ISA/IEC 62443: guía completa. <https://lab.wallarm.com/what/la-serie-de-normas-isa-iec-62443-guia-completa/?lang=es>