

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

Implementación de un Sistema de Monitoreo para la Infraestructura Operativa de un Data Center Utilizando Herramientas de Desarrollo de Código Abierto.

PROYECTO DE TITULACIÓN

Previo la obtención del Título de:

Magister en Automatización y Control

Presentado por:

Fernando Enrique Orbea Guerrero

Mario Andrés Salazar Merchán

GUAYAQUIL - ECUADOR

Año: 2024

DEDICATORIA

El presente proyecto se lo dedico a Dios y a mi madre Dora Guerrero Morán con mucho amor y gratitud, por darme el apoyo incondicional, a mi esposa Jessica Salinas González por siempre impulsarme a ser mejor, enseñándome a valorar todo lo que tengo, y lograr con éxito mi carrera. A mi hija Doménica Orbea Salinas como motivación de superación y de triunfo en la vida.

Fernando Orbea Guerrero.

Quiero dedicar este proyecto de grado a mi padre Mario Alberto Salazar Cortéz el cual ya no pertenece a este mundo, quisiera con todo mi corazón que hayas podido estar presente a lado mío compartir y celebrar juntos éste logro; agradecido siempre por la buena educación brindada, a quien fue mi gran maestro y amigo de vida, gracias a Él por ser quien soy como persona y cómo profesional. A mi madre Carolina quien también me ha apoyado incondicionalmente durante el transcurso de toda mi vida.

A mi esposa Rebeca y a mi hijo Mario Israel que han sido pilares fundamentales para seguir creciendo y alcanzar nuevas metas; a mis hermanas Ana y Mayra, amigos y compañeros.

Mario Salazar Merchán.

AGRADECIMIENTOS

Nuestro más sincero agradecimiento a nuestro Director de tesis MSc. Richard Sánchez, por brindarnos su apoyo total al desarrollo de este proyecto final de Maestría. A nuestros profesores de la FIEC que nos compartieron sus conocimientos para fortalecer nuestra carrera técnica y así permitirnos ser un gran aporte a la industria ecuatoriana. Agradecimiento especial a nuestras familias, por su apoyo durante todo este proceso de la MACI X.

Fernando Orbea Guerrero.

Quiero agradecer a Dios por permitirme llegar hasta esta etapa de mi vida, a mi familia en especial a mi papá: Mario Alberto Salazar Cortéz por su buena labor como padre a pesar de que ya no se encuentra conmigo él me brindó todo su apoyo incondicional.

Mi madre Carolina por estar siempre a mi lado compartiendo su amor, sabiduría y tiempo, mi esposa e hijo Rebeca y Mario Israel siendo ellos mi pilar fundamental, a mis hermanas Ana y Mayra.

A nuestros maestros por impartir sus conocimientos y experiencias acogiéndonos en las aulas con la mejor disposición del caso para poder llevar a cabo este proyecto de titulación.

Mario Salazar.

DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Fernando Enrique Orbea Guerrero y Mario Andrés Salazar Merchán damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”

Ing. Fernando Enrique
Orbea Guerrero

Ing. Mario Andrés
Salazar Merchán

COMITÉ EVALUADOR

Msc. Richard Sánchez Rosado
PROFESOR TUTOR

P.H.D. Dennys Paillacho Chiliza
PROFESOR EVALUADOR

RESUMEN

Para un Centro de Datos, la implementación de un Sistema de Gestión de Edificios es crucial para asegurar el monitoreo eficiente de la infraestructura eléctrica y mecánica. Un BMS bien diseñado no solo facilita la supervisión continua, sino que también mejora la capacidad de respuesta ante eventos o incidentes, minimizando tiempos de inactividad y optimizando el rendimiento general de la infraestructura operativa del mismo.

Para la implementación de este sistema de monitoreo se utilizaron plataformas de software libre que permitieron desplegar servicios contenerizados de Node-RED para poder alcanzar mediante el desarrollo de programación visual la adquisición, tratamiento, almacenamiento y presentación de la data que generan los equipos existentes en la red del centro de datos, a través de los diferentes protocolos de comunicación industriales.

Los resultados obtenidos fueron enfocados en conseguir una solución robusta y flexible para gestionar la infraestructura de un centro de datos, mejorando la capacidad de respuesta y la eficiencia operativa.

El uso de software libre y herramientas de contenedores permiten reducir costos de licencias y facilitar la implementación del desarrollo de sistemas de monitoreo.

Palabras Clave: Node-RED, Contenedores, Protocolos de comunicación, Monitoreo.

ABSTRACT

For a Data Center, implementing a Building Management System (BMS) is crucial to ensure efficient monitoring of electrical and mechanical infrastructure. A well-designed BMS not only facilitates continuous supervision but also enhances the response capability to events or incidents, minimizing downtime and optimizing the overall performance of the operational infrastructure.

For the implementation of this monitoring system, open-source platforms were used to deploy containerized Node-RED services. This approach allowed for the visual programming development needed to acquire, process, store, and present the data generated by the existing equipment on the data center network through various industrial communication protocols.

The results focused on achieving a robust and flexible solution for managing a data center's infrastructure, improving response capability and operational efficiency.

The use of open-source software and container tools helps reduce licensing costs and facilitates the implementation of monitoring system development.

Keywords: *Node-RED, Containers, Communication Protocols, Monitoring.*

ÍNDICE GENERAL

COMITÉ EVALUADOR.....	7
RESUMEN.....	8
<i>ABSTRACT</i>	9
ÍNDICE GENERAL	10
ABREVIATURAS.....	13
SIMBOLOGÍA.....	14
ÍNDICE DE FIGURAS	15
ÍNDICE DE TABLAS.....	17
CAPÍTULO 1.....	18
1. Introducción	18
1.1 Descripción del problema	19
1.2 Justificación del problema	20
1.3 Objetivos	20
1.3.1 Objetivo General.....	20
1.3.2 Objetivos Específicos.....	20
1.4 Marco teórico.....	21
1.4.1 Protocolos de Comunicación de Infraestructura de soporte de un Centro de Datos	21
1.4.2 Pasarelas de Conversión de Protocolo Modbus RTU a Modbus TCP ...	21
1.4.3 Contenerización	22
1.4.4 Contenerización de aplicaciones	23
1.4.5 Ventajas y beneficios de la contenerización	23
1.4.6 Contenerización Docker.....	25
1.4.7 SOFTWARE OPEN SOURCE	26

1.4.8	Node-RED	27
1.4.9	INKSCAPE.....	28
1.4.10	CMDER.EXE.....	28
1.4.11	NODE.JS	29
1.4.12	Grafana.....	29
1.4.13	Paneles de control	30
1.4.14	VISUAL STUDIO CODE	31
CAPÍTULO 2.....		32
2.	Metodología	32
2.1	Adquisición y procesamiento de data	33
2.1.1	Cómo instalar Docker en CentOS 7	33
2.1.2	Cómo instalar Node-Red en Docker	34
2.1.3	Actualizar	36
2.1.4	Introducción a NodeRed	36
2.2	Almacenamiento de la data:	44
2.3	Diseño de dashboards.....	44
2.3.1	Diseñar bajo INKSCAPE.....	44
2.3.2	Desarrollo del dashboard.....	52
CAPÍTULO 3.....		77
3.	Resultados Y ANÁLISIS	77
3.1	Comparativa cualitativa del sistema de monitoreo	77
3.2	Costos de Implementación Sistema de monitoreo.	78
3.3	Análisis de los resultados.	80
CAPÍTULO 4.....		81
4.	Conclusiones Y Recomendaciones	81

Conclusiones.....	81
Recomendaciones.....	82
BIBLIOGRAFÍA.....	84
APÉNDICES.....	86

ABREVIATURAS

ESPOL	Escuela Superior Politécnica del Litoral
SCADA	Supervisory Control and Data Acquisition,
IOT	Internet of things
BMS	Building Management System
PLC	Programmable Logic Controller

SIMBOLOGÍA

V	Voltio
A	Amperio
W	Watt
KVA	Kilovoltamperio
PUE	Power Usage Effectiveness
Hz	Hertz

ÍNDICE DE FIGURAS

Figura 1-1 Arquitectura del Docker Engine (IKONO, 2017)	25
Figura 1-2 Arquitectura Docker (DZONE, 2019).....	26
Figura 1-3 Internet de las cosas (Molina, 2019)	28
Figura 1-4 Cómo funciona Grafana (Asturias, 2023)	31
Figura 2-1 Interfaz de usuario (Los autores, 2024).....	37
Figura 2-2 Nodos por default (Lea, 2023).....	37
Figura 2-4 Manage palette (Los autores, 2024).....	39
Figura 2-5 Nodos descargados (Los autores, 2024)	40
Figura 2-6 Nodo MODBUS (Los autores, 2024)	41
Figura 2-7 Nodo Snmp (Los autores, 2024)	42
Figura 2-8 Nodo Bacnet (Los autores, 2024).....	43
Figura 2-9 Nodo Ethernet-IP (Los autores, 2024).....	44
Figura 2-10 Proyecto nuevo (Los autores, 2024)	46
Figura 2-11 Hoja Del Proyecto (Los autores, 2024)	46
Figura 2-12 Paneles futuristas inkscape (Google, s.f.)	47
Figura 2-13 Capa: Fondos y Marcos (Los autores, 2024)	48
Figura 2-14 Capas: Fondos y Marcos más Texto fijo (Los autores, 2024)	48
Figura 2-15 <i>Fondos y Marcos más Variables</i> (Los autores, 2024)	49
Figura 2-16 Fondos y Marcos más Imágenes (Los autores, 2024)	49
Figura 2-17 Diseño de pantalla completa (Los autores, 2024)	50
Figura 2-18 Diseño de MENU para navegación entre pantallas (Los autores, 2024)....	50
Figura 2-19 Imagen a modo de plantilla (Los autores, 2024)	51
Figura 2-20 Imagen de equipo medidor de energía (Los autores, 2024).....	51
Figura 2-21 Árbol de Proyecto Principal (Los autores, 2024)	56
Figura 2-22 Transform.tools (Los autores, 2024)	57
Figura 2-23 Insertar imagen desarrollada en INKSCAPE (Los autores, 2024).....	57
Figura 2-24 Imagen transformada a líneas de código (Los autores, 2024)	58
Figura 2-25 Programa generado para variables (Los autores, 2024)	59
Figura 2-26 Exportando Interface (Los autores, 2024)	60
Figura 2-27 Ejemplo asignar el tipo de variable a la imagen (Los autores, 2024)	61

Figura 2-28 Ejemplo asignar el tipo de variable a la imagen (Los autores, 2024)	62
Figura 2-29 Código generado de la imagen (Los autores, 2024)	63
Figura 2-30 Código texto fijo (Los autores, 2024).....	64
Figura 2-31 Variables por segmentos (Los autores, 2024).....	65
Figura 2-32 Archivo estados (Los autores, 2024).....	65
Figura 2-33 Archivo dataPrincipal (Los autores, 2024).....	66
Figura 2-34 Interface con los parámetros de las variables (Los autores, 2024)	66
Figura 2-35 Asignación de variable (Los autores, 2024)	67
Figura 2-36 Lectura de la variable (Los autores, 2024)	67
Figura 2-37 Programación en el Panel de Grafana (Los autores, 2024)	68
Figura 2-38 Data Sources (Los autores, 2024).....	69
Figura 2-39 Variables comunicadas o leídas en el Grafana (Los autores, 2024)	69
Figura 2-40 Presentación del Dashboard en Grafana (Los autores, 2024)	70
Figura 2-41 Dashboard Equipo Medidor con datos reales (Los autores, 2024)	71
Figura 2-42 Dashboard Equipo Rectificador con datos reales (Los autores, 2024).....	71
Figura 2-43 Dashboard Equipos Generadores (Los autores, 2024)	72
Figura 2-44 Dashboard Resumen (Los autores, 2024)	73
Figura 2-45 Dashboard Equipo PDI's (Los autores, 2024).....	74
Figura 2-46 Dashboard Equipo PDI's breakers de distribución (Los autores, 2024).....	74
Figura 2-47 Menú de navegación en Grafana (Los autores, 2024)	75

ÍNDICE DE TABLAS

- Tabla 3.1-1 Comparativa cualitativa del sistema de monitoreo (Los autores, 2024)77
- Tabla 3.2-1 Costos sistema de monitoreo en el mercado actual (Los autores, 2024) ...78
- Tabla 3.2-2 Costos de la implementación del proyecto (Los autores, 2024)79
- Tabla 2.3-1 Protocolos de comunicación de los equipos y cantidad de variables (Los autores, 2024)86
- Tabla 2.3-2 Variables de Generadores (Los autores, 2024).....87
- Tabla 2.3-3 Variables de UPS (Los autores, 2024)95
- Tabla 2.3-4 Variables de manejadoras de aire (Los autores, 2024)96
- Tabla 2.3-5 Variables de Rectificadores (Los autores, 2024)97
- Tabla 2.3-6 Variables de Medidor Ecoluz (Los autores, 2024)97
- Tabla 2.3-7 Variables de PDU (Los autores, 2024)98
- Tabla 2.3-8 Variables de ATS (Los autores, 2024)..... 100
- Tabla 2.3-9 Variables de Breakers-Spectra (Los autores, 2024)..... 100
- Tabla 2.3-10 Variables de Breakers-PowerBreak (Los autores, 2024)..... 103

CAPÍTULO 1

1. INTRODUCCIÓN

Un centro de datos es una instalación física centralizada que las organizaciones utilizan para alojar sus aplicaciones y datos críticos, se encuentra compuesto por:

- Una infraestructura de red que permite la conectividad de servidores (físicos y virtualizados), servicios del centro de datos, almacenamiento y conectividad externa a las ubicaciones del usuario final.
- Una infraestructura de almacenamiento en donde los datos son el combustible de los centros de datos modernos, ya que los sistemas de almacenamiento se utilizan para contener este valioso producto. (EVCON, 2024)
- Los recursos de computación son el motor de un centro de datos en donde los servidores ofrecen la capacidad de procesamiento, la memoria, el almacenamiento local y la conectividad de red que impulsan las aplicaciones. (EVCON, 2024)

Dada la criticidad que conllevan este tipo de instalaciones físicas, los componentes de un centro de datos requieren de una importante infraestructura de soporte que contribuye al funcionamiento eficaz de los equipos de TI (Tecnología de Información), garantizando el uso de las instalaciones sin interrupciones. Entre estos, se incluyen los subsistemas de alimentación, fuentes de alimentación ininterrumpidas (UPS), sistemas de climatización, sistemas de supresión de incendios, generadores de respaldo y conexiones a redes externas.

Un Centro de Datos al ser considerado un sitio de alta tecnología, de misión crítica y con requerimientos muy demandantes de servicios principalmente de climatización y energía eléctrica de precisión, su monitoreo se vuelve vital. (Alcocer, 2020)

1.1 Descripción del problema

La complejidad del sistema de monitoreo de un edificio va de la mano con las características del mismo, ya que cada edificio es único. Sin embargo, los centros de datos representan un reto mayor debido a la criticidad de su operación ya que cuentan con una infraestructura de soporte compleja para garantizar siempre la mayor disponibilidad. (Alcocer, 2020)

En el mercado industrial encontramos diferentes fabricantes de software SCADA (Supervisión, Control y Adquisición de Datos) para la gestión de edificios, que de ahora en adelante lo llamaremos BMS (sistema de gestión de edificaciones). Estos sistemas combinan software y hardware de supervisión y control especializados, que son licenciados y su costo depende de la capacidad de recursos a adquirir en base a la magnitud del proyecto de monitoreo que estaría definido por los protocolos de comunicación a manejar, cantidad de equipos y/o cantidad de variables a monitorear.

En cuanto al hardware, el uso de pasarelas de conversión de protocolo simple o multiprotocolo utilizadas para realizar la adquisición de la data de uno o varios equipos que forman parte de un sistema crítico en un centro de datos, suponen el riesgo de que alguno de estos dispositivos pueda quedar fuera de servicio por daño permanente y por ende perder parte o la totalidad del monitoreo de la infraestructura de soporte.

Esta situación se agravaría si no se cuenta con la disponibilidad de otra pasarela de respaldo para su reemplazo, con lo que se incurre de forma adicional en costos y tiempos de adquisición que pudieran ser elevados si no se encuentra el repuesto en el mercado local y la compra se la realiza por importación.

Por otro lado, la integración de un nuevo equipo al sistema de monitoreo muchas veces se vuelve compleja y suele estar ligada al uso de software propietario que requiere de una licencia adicional para la administración y adquisición de una determinada cantidad de variables de acuerdo al tipo de protocolo de comunicación que posea.

1.2 Justificación del problema

El auge del IoT (Internet de las Cosas) ha dado paso a que la revolución para el sector industrial 4.0 experimente una transformación digital que continúa evolucionando de acuerdo a las nuevas necesidades del mercado. (VERITAS, 2024)

Aprovechando esta tendencia y de la mano de plataformas de desarrollo de arquitectura abierta se busca ofrecer otra alternativa a la hora de planificar la implementación nueva o migración de un BMS que permita minimizar la dependencia del uso de pasarelas industriales como hardware especializado y de software licenciado que permitan realizar la adquisición y presentación de la data de la infraestructura de soporte del centro de datos.

1.3 Objetivos

1.3.1 Objetivo General

Utilizar plataformas de desarrollo IoT para implementar un sistema de monitoreo, que realice la adquisición, procesamiento y presentación de datos de los equipos que conforman la infraestructura de soporte de un Centro de Datos.

1.3.2 Objetivos Específicos

1. Identificar y reconocer el protocolo de comunicación con los que cuentan los equipos que conforman la infraestructura de soporte de un Centro de Datos.

2. Establecer la red de comunicaciones que se implementará, incluyendo en ella las pasarelas de conversión de protocolos seriales a TCP.
3. Implementar y levantar plataforma Node-RED como servicio contenerizado con Docker, utilizando una máquina virtual (servidor en la nube) con sistema operativo Linux, para realizar la adquisición y procesamiento de la data de los equipos según el protocolo de comunicación que posea.
4. Utilizar plataforma Grafana para desarrollar dashboard de la infraestructura de soporte de un centro de datos que facilite al operador de monitoreo el reconocimiento del estado operativo de los equipos de la infraestructura de forma intuitiva.

1.4 Marco teórico

1.4.1 Protocolos de Comunicación de Infraestructura de soporte de un Centro de Datos

1.4.2 Pasarelas de Conversión de Protocolo Modbus RTU a Modbus TCP

Son puertas de enlace Modbus estándar que convierten entre los protocolos Modbus TCP y Modbus RTU/ASCII. Se admiten hasta 16 maestros Modbus TCP simultáneos, con hasta 31 esclavos RTU/ASCII por puerto serie. Para maestros RTU/ASCII, se admiten hasta 32 esclavos TCP. El enrutamiento a través de los puertos serie se puede controlar mediante la dirección IP, el número de puerto TCP o la asignación de ID. El enrutamiento del puerto serie por puerto TCP y dirección IP permite el acceso de hasta 4 clientes/maestros TCP (modelos MB3280/MB3480), mientras que el enrutamiento por asignación de ID permite el acceso de hasta 16 clientes/maestros TCP (MB3180/MB3280/MB3480).

Integración de red Modbus estándar Los tres modelos MGate™ estándar (MB3180, MB3280 y MB3480) están diseñados para una fácil integración de redes Modbus TCP y RTU/ASCII. Con estos modelos, los dispositivos esclavos serie Modbus se pueden incorporar sin problemas a una red Modbus TCP existente, y los esclavos Modbus TCP pueden hacerse accesibles a los maestros serie. Los MB3180, MB3280 y MB3480 ofrecen funciones que hacen que la integración de

la red sea fácil, personalizable y compatible con casi cualquier red Modbus. Gateways rentables y de alta densidad Los gateways MGate™ MB3000 pueden conectar eficazmente una alta densidad de nodos Modbus a la misma red. El MB3280 puede gestionar hasta 62 nodos esclavos en serie y el MB3480 puede gestionar hasta 124 nodos esclavos en serie. Cada puerto serie RS-232/422/485 se puede configurar individualmente para operación Modbus RTU o Modbus ASCII y para diferentes velocidades en baudios, lo que permite integrar ambos tipos de redes con Modbus TCP a través de una puerta de enlace Modbus. Enrutamiento automático de dispositivos para una configuración sencilla (patentado) La función de enrutamiento automático de dispositivos de Moxa ayuda a eliminar muchos de los problemas e inconvenientes que encuentran los ingenieros que necesitan configurar una gran cantidad de dispositivos Modbus. Un solo clic del mouse es todo lo que se necesita para configurar una tabla de enrutamiento de ID esclavo y configurar puertas de enlace Modbus para detectar automáticamente solicitudes Modbus desde un sistema de control de supervisión y adquisición de datos (SCADA). Al eliminar la necesidad de crear manualmente la tabla de enrutamiento de ID de esclavo, la función de enrutamiento automático de dispositivos ahorra a los ingenieros una cantidad significativa de tiempo y costos.

1.4.3 Contenerización

Es una forma de virtualización del SO en la que se ejecutan aplicaciones en espacios de usuario aislados llamados contenedores, que utilizan el mismo SO compartido. Un contenedor de aplicaciones es un entorno informático portátil y totalmente empaquetado:

- Tiene todo lo que una aplicación necesita ejecutar, incluidos sus binarios, bibliotecas, dependencias y archivos de configuración, todo encapsulado y aislado en un contenedor.

- La contenerización de una aplicación abstrae el contenedor del sistema operativo host, y proporciona acceso limitado a los recursos subyacentes, un proceso similar al de una máquina virtual ligera.
- Puede ejecutar la aplicación en contenedores en varios tipos de infraestructuras, como de forma completa, en la nube o dentro de máquinas virtuales, sin tener que refactorizarla para cada entorno.

Con la contenerización, tiene menos gastos generales durante el inicio y no necesita configurar sistemas operativos invitados separados para cada aplicación, ya que todas comparten un kernel de SO. Debido a esta alta eficiencia, los desarrolladores de software comúnmente usan la contenerización de aplicaciones para empaquetar varios de los microservicios individuales que componen las aplicaciones modernas. (VERITAS, 2024)

1.4.4 Contenerización de aplicaciones

La abstracción del SO host hace que las aplicaciones en contenedores sean portátiles y les permite ejecutarse de manera consistente y uniforme en cualquier plataforma o nube. Los desarrolladores pueden transportar fácilmente contenedores de una plataforma a otra, por ejemplo del SO Windows a Linux. También se ejecutarán de forma coherente en servidores tradicionales "sin sistema operativo" o en infraestructuras virtualizadas, ya sea en las instalaciones o en la nube. Por lo tanto, los desarrolladores pueden seguir utilizando los procesos y las herramientas que deseen. (VERITAS, 2024)

1.4.5 Ventajas y beneficios de la contenerización

La contenerización ofrece beneficios significativos para los desarrolladores de software y los equipos de desarrollo, que van desde mayor agilidad y portabilidad hasta mejores controles de costos. A continuación, se muestran las ventajas:

- Portabilidad

Un contenedor de aplicaciones crea un paquete de software ejecutable que se abstrae del SO host. Por lo tanto, al no depender ni estar vinculado al SO host, lo hace portátil y le permite ejecutarse de manera consistente y uniforme en cualquier plataforma o nube.

- Velocidad

Los desarrolladores se refieren a los contenedores como "livianos" porque comparten el kernel del SO de la máquina host y no están sujetos a una sobrecarga adicional. Su función ligera impulsa una mayor eficiencia de los servidores y reduce los costos de licencia y de servidores. También acelera el tiempo de inicio, ya que no hay que hacer arrancar ningún SO.

- Escalabilidad

La tecnología del contenedor de aplicaciones ofrece alta escalabilidad. Un contenedor de aplicaciones puede manejar cargas de trabajo crecientes mediante la reconfiguración de la arquitectura existente para permitir que los recursos utilicen un diseño de aplicación orientado al servicio. Como alternativa, un desarrollador puede agregar más contenedores dentro de un clúster de máquinas distribuidas.

- Eficiencia

Dado que el software que se ejecuta en entornos en contenedores comparte el kernel del SO de la máquina host, los desarrolladores pueden compartir las capas de la aplicación entre contenedores. Además, los contenedores tienen una capacidad inherentemente menor que las máquinas virtuales. Necesitan tiempos de inicio mínimos, lo que permite a los desarrolladores ejecutar más contenedores

en la misma capacidad de cómputo que una máquina virtual. En consecuencia, esto genera una mayor eficiencia del servidor al tiempo que reduce los costos de licencias y del servidor. (VERITAS, 2024)

1.4.6 Contenerización Docker

El motor de Docker es, quizás, la tecnología de motores de contenedores más conocida y utilizada en todo el mundo. Como pieza principal en una arquitectura de contenedores, un Docker se refiere a un código abierto basado en un kernel de Linux que es responsable de crear contenedores en un SO.

Cuando un Docker accede a un solo kernel de SO, puede administrar múltiples aplicaciones distribuidas que se ejecutan en sus respectivos contenedores. La base de la contenerización es el paquete de software que los desarrolladores implementan en un único envío virtual. (VERITAS, 2024)

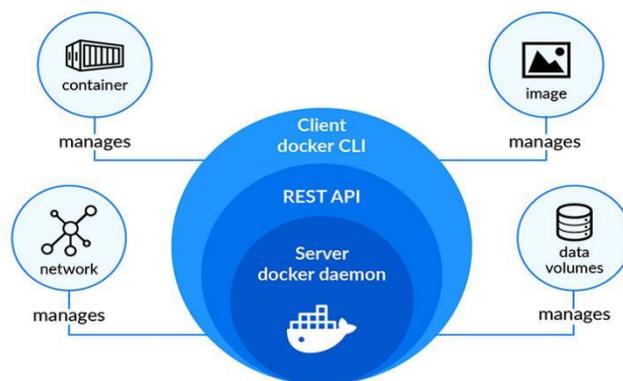


Figura 1-1 Arquitectura del Docker Engine (IKONO, 2017)



Figura 1-2 Arquitectura Docker (DZONE, 2019)

1.4.7 SOFTWARE OPEN SOURCE

El software open source o de código abierto se lanza con una licencia específica que pone su código fuente a disposición de los usuarios finales de forma legal. Existen muchas licencias de este tipo, pero normalmente el software se considera open source bajo las siguientes condiciones:

Está disponible en forma de código fuente sin costo adicional, lo cual significa que los usuarios pueden visualizar el código del software y hacer todos los cambios que deseen.

El código fuente se puede reutilizar en un software nuevo, así que cualquier persona puede usar el código fuente para desarrollar su propio programa y distribuirlo.

El hecho de que sea open source no significa que el software ejecutable se distribuya sin costo. Sin embargo, sí significa que su código fuente está disponible sin costo.

El software libre y el software open source comparten ciertos valores y suelen combinarse en la frase popular "software open source y libre" (FOSS). (REDHAT, 2022)

Dentro de los programas elegidos para nuestro proyecto de monitoreo de la infraestructura operativa de un Centro de Datos tenemos los siguientes: INKSCAPE, CMDER, DOCKER, Node.js, NodeRed, Visual Studio Code, Grafana.

1.4.8 Node-RED

Node-RED es herramienta de desarrollo open-source basada en programación visual que fue creada por IBM para conectar dispositivos de hardware, APIs y servicios en línea.

Una herramienta robusta, de fácil aprendizaje, y que no requiere muchos conocimientos de programación y que se ha consolidado como una de las principales aplicaciones de gestión y transformación de datos en tiempo real para soluciones IoT e Industria 4.0.

Node-RED permite conectar gráficamente bloques predefinidos, llamados nodos, para desarrollar una tarea concreta. La conexión de los nodos, habitualmente una combinación de nodos de entrada, nodos de procesamiento y nodos de salida, forman lo que conocemos como flow.

Entre los nodos disponibles podemos encontrar desde protocolos estándar como MQTT, REST, Modbus, OPC-UA, Bacnet, SNMP hasta integraciones a APIs de terceros como Microsoft Azure, Amazon Web Services, Twitter, Facebook y muchas más. (PICKDATA, 2020)

Node-RED está creado a partir de Node.js, lo cual proporciona a la herramienta una potencia suficiente para que sea escalable y fiable, con unos requerimientos de computación muy bajos. Dicha característica ha propiciado que Node-RED pueda ser ejecutado en ordenadores domésticos, en servidores cloud y en dispositivos embebidos de bajas prestaciones.

1.4.11 NODE.JS

Node.js es un entorno de tiempo de ejecución de JavaScript (de ahí su terminación en .js haciendo alusión al lenguaje JavaScript). Este entorno de tiempo de ejecución en tiempo real incluye todo lo que se necesita para ejecutar un programa escrito en JavaScript. También aporta muchos beneficios y soluciona muchísimos problemas, por lo que sería más que interesante realizar nuestro curso de Node.js para obtener las bases, conceptos y habilidades necesarias que nos motiven a profundizar en sus opciones e iniciar la programación.

Node.js fue creado por los desarrolladores originales de JavaScript. Lo transformaron de algo que solo podía ejecutarse en el navegador en algo que se podría ejecutar en los ordenadores como si de aplicaciones independientes se tratara. (OPENWEBINARS, 2019)

1.4.12 Grafana

Grafana es una plataforma interactiva y open source de visualización de datos desarrollada por Grafana Labs. Dicha plataforma permite a los usuarios ver sus datos a través de tablas y gráficos que se unifican en un panel de control (o en varios) para facilitar la interpretación y la comprensión. También permite realizar consultas y configurar avisos sobre la información y los indicadores desde el lugar en el que se almacena dicha información, ya sea en entornos de servidores tradicionales, clústeres de Kubernetes y varios servicios de nube, entre otros. De esta forma, podrá analizar los datos e identificar las tendencias y las inconsistencias con mayor facilidad, por lo que sus procesos serán más eficientes. Grafana se diseñó según los principios del open source y con la creencia de que los datos deberían ser accesibles para toda la empresa, no solo para unas pocas personas. Esto fomenta una cultura por la cual cualquier persona que necesite los datos puede encontrarlos y utilizarlos fácilmente, lo cual permite que los equipos sean más abiertos, innovadores y colaborativos. (REDHAT, 2022)

1.4.13 Paneles de control

Los paneles de control de Grafana transforman los datos recolectados de diversas fuentes al ofrecer una plataforma para compartir esta información con otros equipos o miembros. Esto facilita la colaboración y permite realizar análisis más profundos sobre los datos y sus implicaciones. Puede diseñar paneles de control personalizados para usted y su equipo, ajustándolos según sus necesidades específicas mediante funciones avanzadas de consulta y transformación.

Es fundamental entender los datos relevantes y sus relaciones cuando intenta encontrar la causa de un incidente o un comportamiento inesperado del sistema lo más rápido posible. Grafana permite que los datos se trasladen entre los equipos y sus miembros sin problemas para que puedan observarlos de forma óptima y, de esta manera, llegar a la raíz del problema rápidamente y resolverlo.

La información de los paneles de control de Grafana se puede compartir de las siguientes maneras:

- **En toda la empresa:** incluso con colegas que no utilizan la herramienta.
- **Con la comunidad de Grafana:** a nivel global, permitiendo la colaboración y el intercambio de conocimientos con usuarios de todo el mundo.
- **Desde cualquier lugar:** puede acceder a sus paneles de control desde cualquier dispositivo, sin importar su ubicación.

Características Clave

- **Paneles:** Visualice sus datos de diversas maneras, incluyendo histogramas, gráficos, mapas geográficos, mapas de calor, entre otros.
- **Plugins:** Procese datos de inmediato mediante una API intuitiva a través de plugins que se conectan directamente a las fuentes de datos, sin necesidad de trasladarlos. También puede desarrollar plugins personalizados para integrar indicadores desde cualquier API.
- **Alertas:** Cree, gestione y supervise todas sus alertas desde una interfaz centralizada.

- **Transformaciones:** Realice modificaciones como cambios de nombre, resúmenes, combinaciones y cálculos en todas sus fuentes de datos y consultas.
- **Anotaciones:** Añada eventos de diversas fuentes de datos a sus gráficos para proporcionar contexto adicional.
- **Editor de Paneles:** Ofrece una interfaz de usuario unificada para configurar y personalizar sus paneles de control de manera eficiente. (REDHAT, 2022)

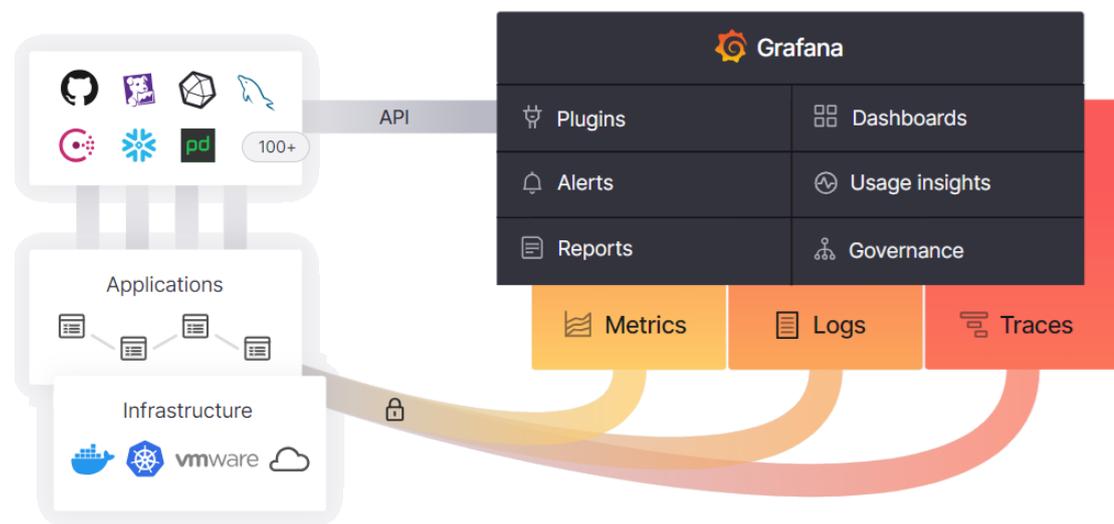


Figura 1-4 Cómo funciona Grafana (Asturias, 2023)

1.4.14 VISUAL STUDIO CODE

Visual Studio Code es una herramienta de edición de código fuente ligera pero eficaz, desarrollada por Microsoft. Es un software libre y multiplataforma, disponible para Windows, GNU/Linux y macOS.

Visual Studio Code ofrece una excelente integración con Git, soporte para depuración de código y una amplia gama de extensiones que permiten escribir y ejecutar código en prácticamente cualquier lenguaje de programación. Proporciona numerosas facilidades para escribir, depurar y probar código. Inicialmente, incluye un conjunto básico de componentes y funciones esenciales, con soporte nativo para JavaScript/TypeScript y Node.js. (Flores, 22)

CAPÍTULO 2

2. METODOLOGÍA

El uso de software se torna imprescindible, permitiendo ejecutar tareas de monitoreo en tiempo real de lo que sucede en campo a través de pantallas dinámicas, al mismo tiempo que cumplen funciones de almacenamiento de datos para futuras consultas a través de reportes.

Para las funciones de comunicación entre elementos de campo, controladores, computadoras (software), actuadores, sensores y demás se requieren tarjetas de comunicación (muchas veces integrados en los propios controladores) que permiten la transmisión de datos a través de cable o incluso a través de señales inalámbricas usando protocolos de comunicación.

Para lograr la adquisición de data de los equipos de la infraestructura de soporte del centro de datos se utilizaron recursos propios tales como la implementación de una máquina virtual, en los cuales se instalaron plataformas de código abierto que permiten levantar servicios contenerizados de herramientas de programación que permitirán comunicarse con los equipos de la infraestructura de soporte a través de la red.

Hoy en día se habla del término contenerización. Por lo general, casi todas las empresas utilizan un entorno de nube (público o privado), con instancias que ejecutan máquinas virtuales con funciones de escalabilidad y equilibrio de carga que representan su capa informática.

Los contenedores y las máquinas virtuales son tecnologías de virtualización de recursos muy similares. La virtualización es el proceso en el que un recurso singular del sistema, como la RAM, la CPU, el disco o la red, se puede “virtualizar” y representar como varios recursos. La diferencia clave entre los contenedores y las

máquinas virtuales es que estas virtualizan toda una máquina hasta las capas de hardware, y los contenedores solo virtualizan las capas de software por encima del nivel del sistema operativo.

2.1 Adquisición y procesamiento de data

Se implementará sobre una máquina virtual un servidor LINUX, distribución Centos 7, para instalar plataforma de contenerización Docker para levantar como servicio la herramienta IoT Node-Red, realizar la programación necesaria, adquirir y procesar la información de todos los equipos que conforman la infraestructura operativa de un centro de datos.

2.1.1 Cómo instalar Docker en CentOS 7

Conectarse al servidor utilizando SSH.

```
ssh tu-usuario@tu-servidor
```

Luego, tenemos que instalar una serie de paquetes antes de instalar Docker.

Necesitamos ser el usuario root y ejecutar la siguiente línea en la línea de comando:

```
yum install -y yum-utils device-mapper-persistent-data lvm2
```

La forma más fácil y segura de completar el proceso es a través de los repositorios oficiales de Docker. Para hacer esto, es necesario ejecutar esta línea de comando:

```
yum-config-manager--add-repo https://download.docker.com/linux/centos/docker-ce.repo
```

Después de esto, podemos instalar Docker en CentOS 7 ejecutando el siguiente comando:

```
yum install docker-ce
```

Para habilitar e iniciar Docker, ejecuta los siguientes comandos:

```
systemctl enable docker
```

```
systemctl start docker
```

Finalmente, verificar el estado del servicio para chequear que todo haya salido bien:

```
systemctl status Docker
```

Como resultado deberíamos ver una línea verde que indica que Docker está en funcionamiento.

2.1.2 Cómo instalar Node-Red en Docker

La instalación en Docker se realiza usando la imagen `nodered/node-red`.

La forma más sencilla es usar un fichero `docker-compose.yml` con el siguiente contenido:

```
version: "3"
```

```
services:
```

```
  node-red:
```

```
    image: nodered/node-red:latest
```

```
    container_name: node-red
```

```
    environment:
```

```
      - TZ=Europe/Madrid
```

```
    volumes:
```

```
      - /home/pi/volumes/node-red:/data
```

```
    ports:
```

```
- 1880:1880
restart: unless-stopped
```

Se puede usar `docker-compose up -d` o usar el contenido del fichero en Portainer.

Configuración de Node-Red

Una vez en marcha, se puede acceder a Node-RED a través del puerto 1880 (en este ejemplo `http://192.168.1.180:1880`) y comenzar la configuración.

Una opción recomendable, además de utilizar HTTPS, es utilizar la autenticación de usuarios.

Para ello hay que descomentar esta sección del fichero `settings.js` para disponer de un usuario admin con todos los privilegios:

```
adminAuth: {
  type: "credentials",
  users: [{
    username: "admin",
    password:
"$2a$08$zZWtXTja0fB1pzD4sHCMYOCMYz2Z6dNbM6tl8sJogENOMcxWV9DN.",
    permissions: "*"
  }]
},
```

Para generar la contraseña de este usuario hay que utilizar la herramienta `node-red` de la siguiente manera:

```
pi@pi4nas:~ $ docker exec -it node-red node-red admin hash-pw
```

Password:

```
$2a$08$zZWtXTja0fB1pzD4sHCMYOCMYz2Z6dNbM6tl8sJogENOMcxWV9DN.
```

A continuación se reinicia el contenedor para utilizar la nueva configuración.

2.1.3 Actualizar

Si ya se había instalado Node-RED anteriormente, se puede actualizar de la siguiente manera:

```
docker stop node-red  
docker rm node-red  
docker rmi nodered/node-red  
docker-compose up -d
```

2.1.4 Introducción a NodeRed

2.1.4.1 Que es y para qué sirve

Node-RED es una herramienta de desarrollo de código abierto basada en programación visual. Es robusta y fácil de aprender, y no requiere conocimientos previos de programación. Permite conectar bloques predefinidos, llamados nodos, para desarrollar tareas específicas mediante una interfaz gráfica.

Construida sobre Node.js, una plataforma basada en JavaScript diseñada para el desarrollo backend, el diseño web y la implementación de procesos, Node-RED ofrece una amplia gama de aproximadamente 2500 nodos disponibles. Además, soporta el manejo de diversos protocolos estándar, como Modbus, BACnet, SNMP, y MQTT, entre otros. (PICKDATA, 2020)

2.1.4.2 Interfaz

NodeRed cuenta con una interfaz amigable y entendible para el usuario la cual se compone en:

1. Nodos
2. Entorno de trabajo
3. Consola
4. Panel de ejecución

Al instalar Node-RED, se proporciona un conjunto inicial de nodos organizados en ocho categorías principales, tales como INPUT, OUTPUT, FUNCTION y SOCIAL. A medida que se requiera añadir nuevas funcionalidades, será necesario incorporar nodos adicionales para diversos protocolos de comunicación, los cuales están disponibles en la paleta de librerías.

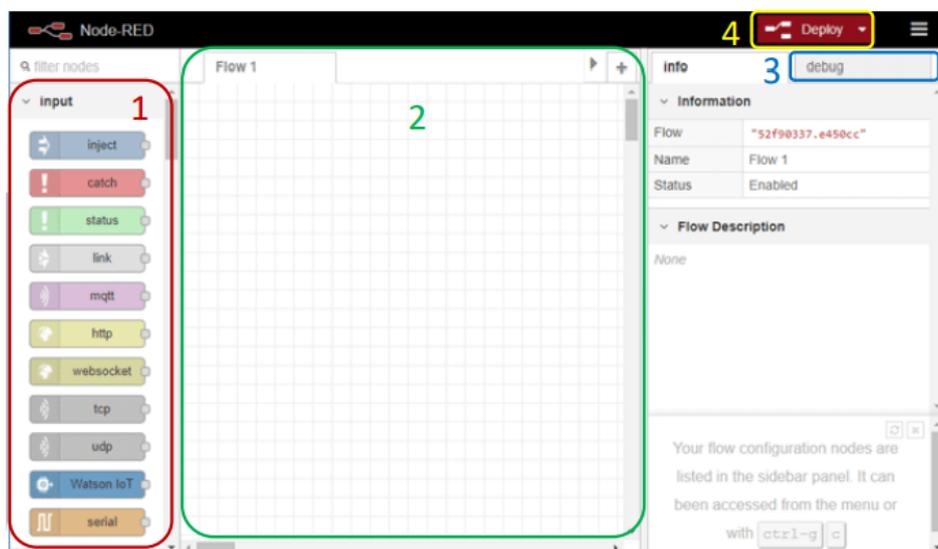


Figura 2-1 Interfaz de usuario (Los autores, 2024)



Figura 2-2 Nodos por default (Lea, 2023)

ENTRADAS

Hay 7 nodos de entrada básicos instalados de forma predeterminada. Cubren los mecanismos de comunicación básicos que probablemente utilicen las aplicaciones de IoT. Desde protocolos de Internet de nivel inferior, como UDP y TCP, hasta HTTP de nivel superior y MQTT de publicación/suscripción. (Lea, 2023)

SALIDAS

Los nodos de salida son esencialmente imágenes reflejadas del conjunto básico de nodos de entrada y proporcionan una forma de enviar datos en el mismo conjunto de protocolos, es decir, mqtt, http, udp, etc. (Lea, 2023)

FUNCION

La categoría de función contiene una variedad de nodos que llevan a cabo funciones de procesamiento específicas. Estos van desde los simples nodos de retardo y conmutación hasta el nodo de función programable que se puede adaptar a casi cualquier necesidad de programación. (Lea, 2023)

SOCIAL

Los nodos básicos de redes sociales soportan la interacción con el correo electrónico y con Twitter. Permiten flujos para enviar o recibir correo electrónico, o para enviar o recibir tweets. (Lea, 2023)

ALMACENAMIENTO

El nodo predeterminado configurado para el almacenamiento, debido a que está dirigido a dispositivos como Raspberry Pi, es bastante limitado y se centra en el almacenamiento basado en archivos. (Lea, 2023)

ANÁLISIS

Los nodos de análisis realizan análisis estándar de los mensajes entrantes. En el conjunto de nodos predeterminado, el único nodo proporcionado es el nodo de opinión, que se puede utilizar para intentar determinar la opinión de un mensaje entrante en

función de las palabras utilizadas en el mensaje, por ejemplo, un correo electrónico o un tweet. (Lea, 2023)

AVANZADO

Un conjunto de nodos diversos que ofrecen varios tipos de funcionalidad. (Lea, 2023)

RASPBERRY PI

Raspberry Pi es un ordenador de placa reducida, ordenador de placa única u ordenador de placa simple (SBC) de bajo costo muy popular en el la industria conectada y con gran apoyo de la comunidad. Es el complemento perfecto para Node-RED en el entorno de trabajo. (Lea, 2023)

Nos enfocaremos en descargar los nodos existentes en las librerías internas del Node-Red; para lo cual nos dirigimos a la parte superior derecha en los tres rayas indicadoras dándole click y desplegándose un listado en el cual seleccionamos Manage palette



Figura 2-3 Manage palette (Los autores, 2024)

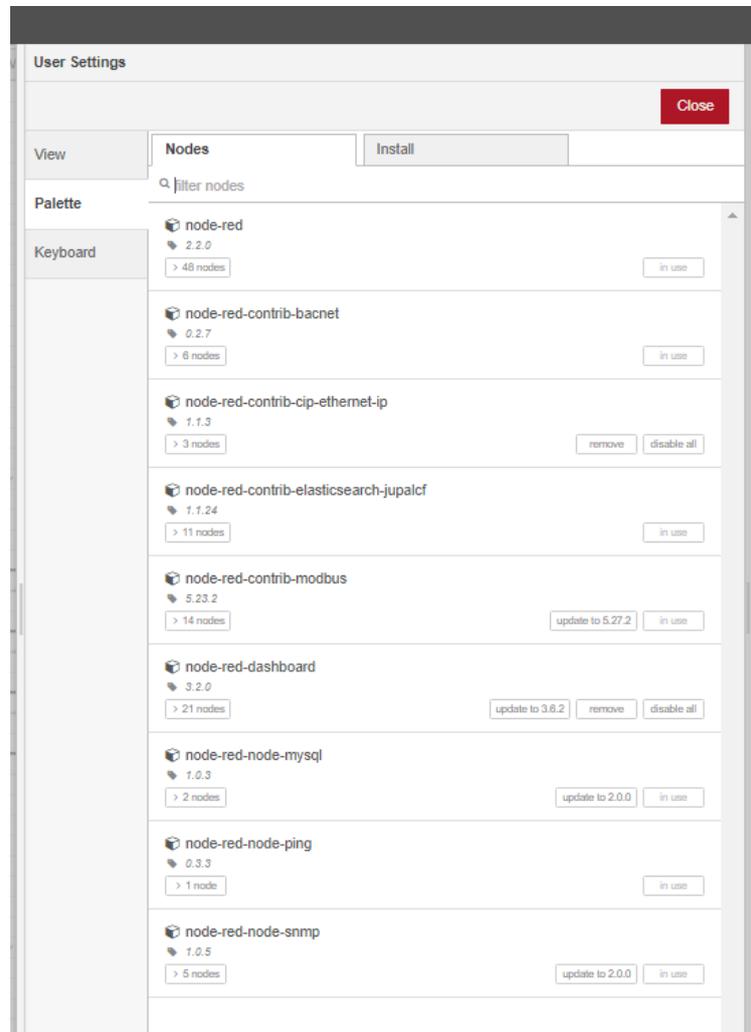


Figura 2-4 Nodos descargados (Los autores, 2024)

En esta imagen tenemos los nodos ya descargados correspondientes a los diferentes protocolos de comunicación utilizados en el proyecto tales como: Modbus, Snmp, Bacnet, Ethernet-IP

NODO MODBUS

Este nodo permite comunicarse con equipos con comunicación Modbus a través de redes TCP/IP. Se necesita de la configuración de la dirección IP del servidor y su número de puerto, el puerto predeterminado para esta comunicación es 502.

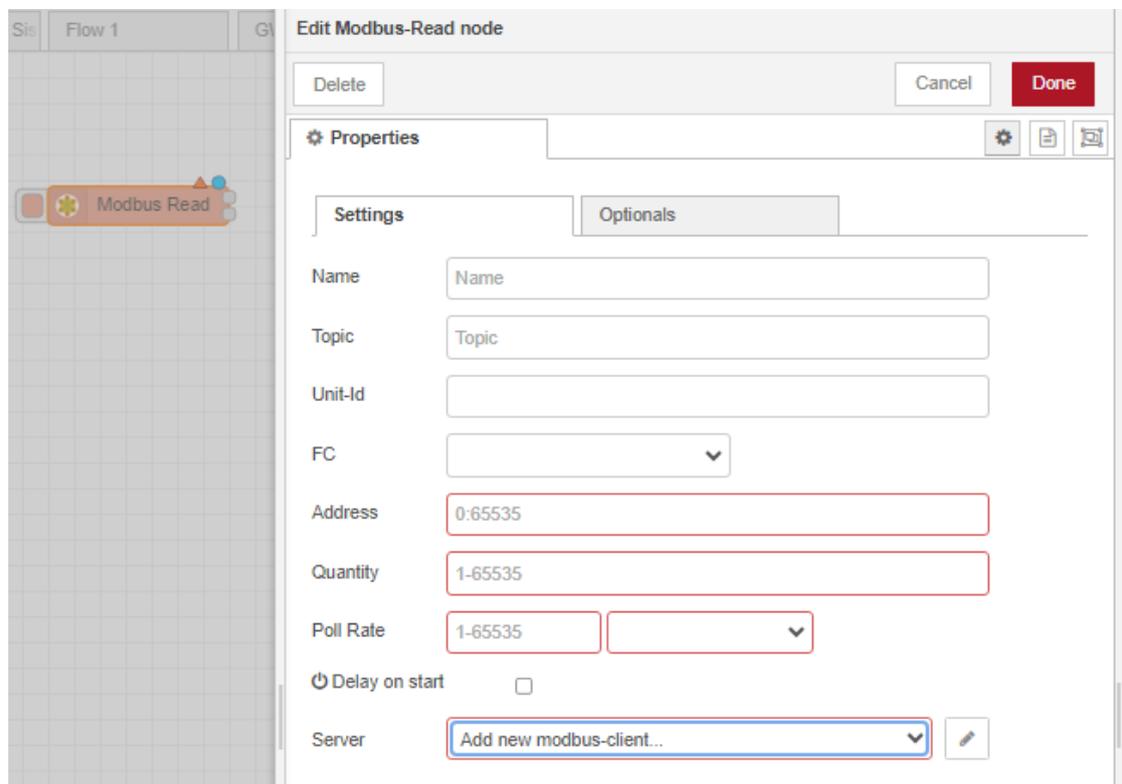


Figura 2-5 Nodo MODBUS (Los autores, 2024)

En este nodo debemos configurar o asignarle un nombre, el tipo de función si es Read Coil Status, Read Input Status, Read Holding Registers, Read Input Registers, ya que para la comunicación MODBUS se realiza mediante la ejecución de 'funciones'. Se pueden dividir en dos categorías principales: la primera para variables de bits (BOOL) y la segunda para WORD. Luego esa distinción se puede dividir nuevamente en variables, que son de solo lectura y aquellas que se pueden sobrescribir.

- Para nuestro caso vamos a usar la función de Read Holding Registers ya que vamos a leer los datos generados por el equipo.
- Luego de esto debemos tener la tabla propia del fabricante la cual nos indica en que posición o dirección se encuentran los registros o datos que vamos a obtener
- La cantidad de parámetros a leer
- Y por último el tiempo de la consulta que le vamos a realizar al dato para que lo lea

NODO SNMP

Este nodo permite comunicarse con equipos con comunicación SNMP. Se debe introducir la listas de OID u OIDs de SNMP.

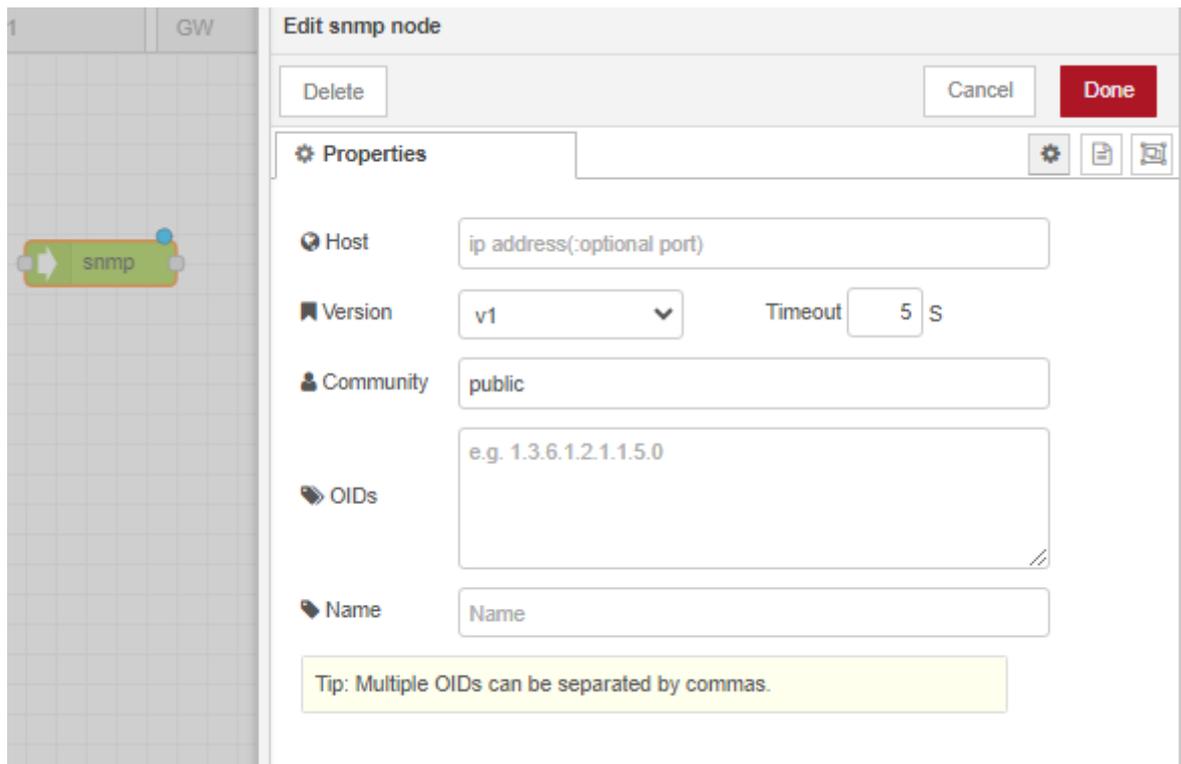


Figura 2-6 Nodo Snmp (Los autores, 2024)

Básicamente le damos la dirección IP del equipo al que deberemos adquirir los datos en el HOST, le asignamos la versión a la cual está configurada la tarjeta de comunicación snmp propia del equipo así como la comunidad, agregamos los Identificadores de Objetos (OID's); en este caso son los parámetros o los datos que envía el equipo. y le asignamos un nombre al nodo, en este caso deberá ser acorde al equipo al cual se le realizará la adquisición del dato.

NODO BACNET

Este nodo permite comunicarse con equipos con comunicación BaCnet. Se debe introducir la instancia, tipo, dispositivo.

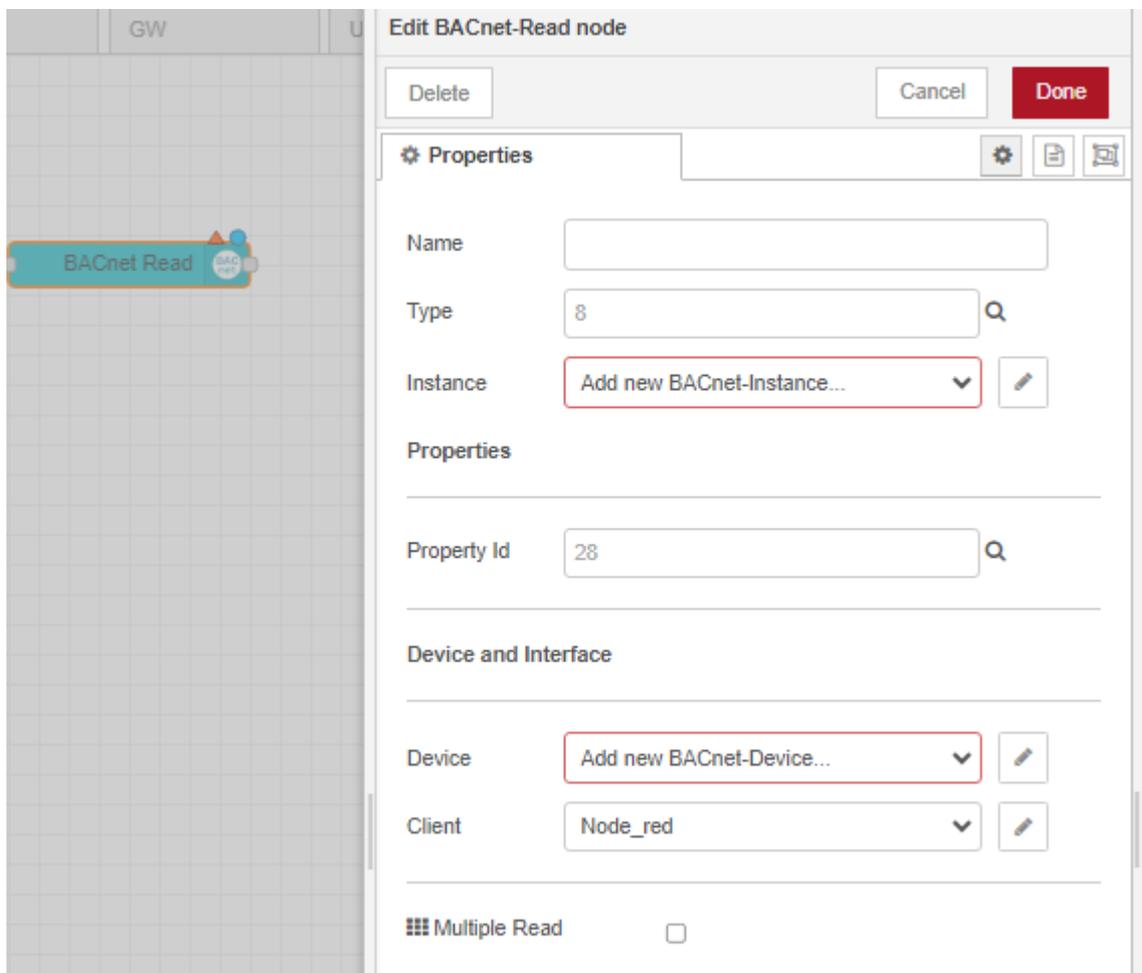


Figura 2-7 Nodo Bacnet (Los autores, 2024)

Para este caso debemos darle un nombre al nodo, acorde con el equipo debemos conocer la instancia, la propiedad, el dispositivo al cual apuntar, en este caso la dirección ip, y el cliente para nuestro caso la ip en dónde se encuentra levantado el NodeRed

NODO ETHERNET IP

Este nodo permite comunicarse con un PLC. Todos los datos se leen cíclicamente desde el PLC como está configurado en el punto final eth-ip, hay tres modos de hacerlos disponibles en un flujo:

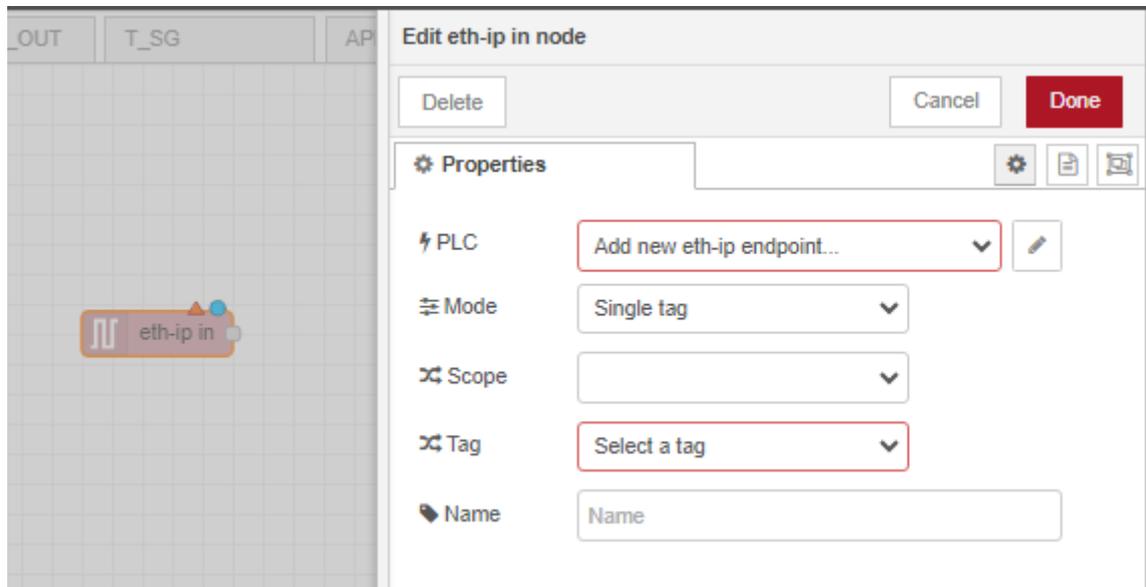


Figura 2-8 Nodo Ethernet-IP (Los autores, 2024)

Este nodo es más sencillo en el cual apuntamos directamente a la dirección IP con la cual se encuentra registrado el PLC a la red, escogemos la opción Mode --all tags-- y automáticamente nos lee las variables existentes en el PLC.

2.2 Almacenamiento de la data:

Parte del procesamiento de la data consiste en enviarla a una base de datos de alta disponibilidad existente en el centro de datos e implementada en un cluster.

2.3 Diseño de dashboards

Se realizan las pantallas en aplicativo o herramienta de diseño gráfico INKSCAPE, para después traducirlas a líneas de código.

2.3.1 Diseñar bajo INKSCAPE

En este aplicativo o editor se realizó el diseño de pantallas para la presentación del monitoreo de la infraestructura de soporte del centro de datos; en el cual se representa los equipos como Celda de Media Tensión, Transformador,

Generadores, UPS's, Rectificador, Chiller's, Bombas, Sensores de temperatura representados por medio de imágenes, estados operativos como el encendido, apagado y alertas por medios de círculos que simulan led's indicadores, valores y parámetros de voltaje y corrientes principales así como potencia total los cuáles son datos obtenidos de cada uno de los equipos ya nombrados.

El diseño se lo realizó de acuerdo a la necesidad y visualización apropiada para el operador quien va a estar a cargo del monitoreo de las pantallas.

En el diseño de las pantallas se trabajó de tal manera que los parámetros, valores o variables a cambiar sean agrupados por capas y los textos fijos e imágenes; en capas diferentes.

Se desarrolló un tipo plantilla en SVG para los diferentes equipos de la infraestructura operativa del CENTRO DE DATOS, cabe recalcar que en éste aplicativo solamente se desarrolló el diseño de las pantallas como imágenes para luego ser leídas como código y posteriormente asignarle la variable y las animaciones correspondientes a nuestra necesidad.

Para el desarrollo de las imágenes en SVG hemos creado un nuevo proyecto, y le asignamos las propiedades de la página para nuestro caso la hoja será de forma horizontal.

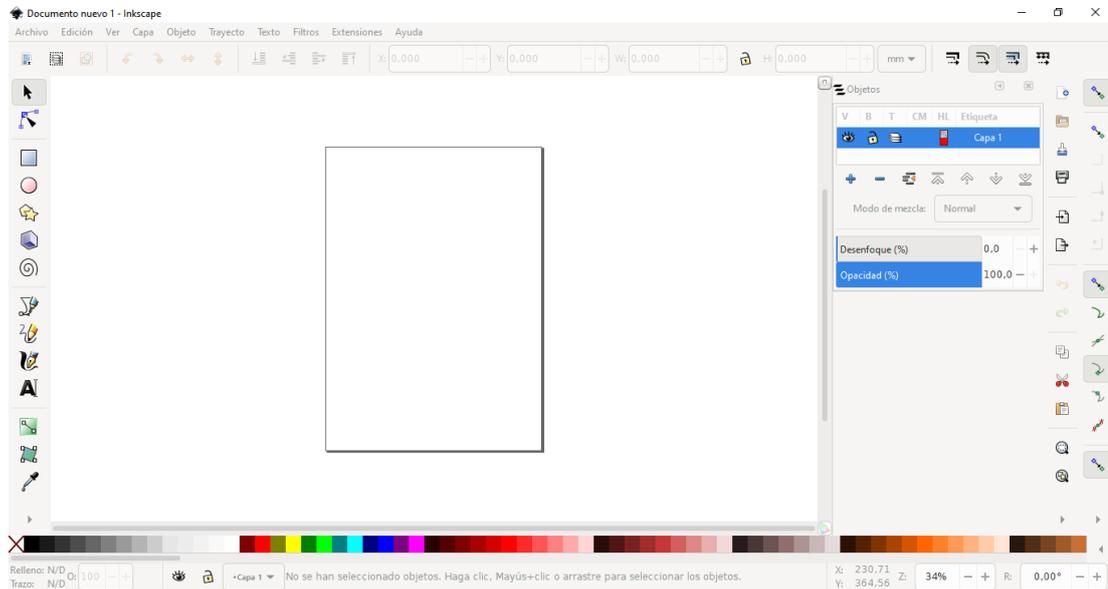


Figura 2-9 Proyecto nuevo (Los autores, 2024)

Las dimensiones de la hoja serán de acuerdo a la dimensión aproximada de la pantalla en la cual vamos a visualizar nuestros diseños.

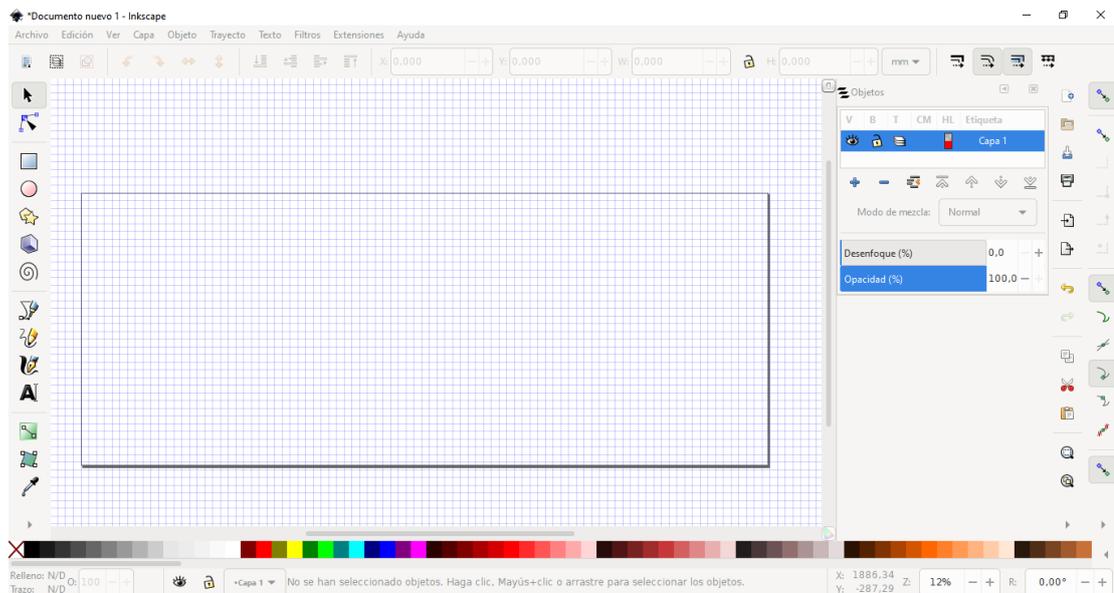


Figura 2-10 Hoja Del Proyecto (Los autores, 2024)

Haciendo una búsqueda en Google como fondos futuristas encontramos diversos fondos y diseños que fueron utilizados para el desarrollo de nuestro proyecto; los cuales se adaptaron descargándolas e importándolas en INKSCAPE para luego modificarlas y tener nuestra plantilla final.

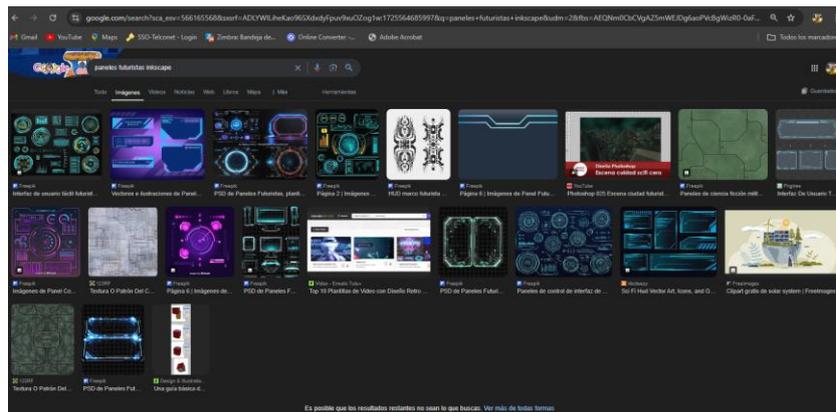


Figura 2-11 Paneles futuristas inkscape (Google, s.f.)

En este proyecto, nos centraremos en el diseño de una pantalla o dashboard que actúe como un resumen visual de los diversos equipos que conforman la infraestructura operativa del Centro de Datos. La pantalla estará diseñada para facilitar la comprensión del operador e incluirá parámetros básicos, como valores y estados operativos, así como alertas o alarmas representadas mediante LEDs y mensajes emergentes.

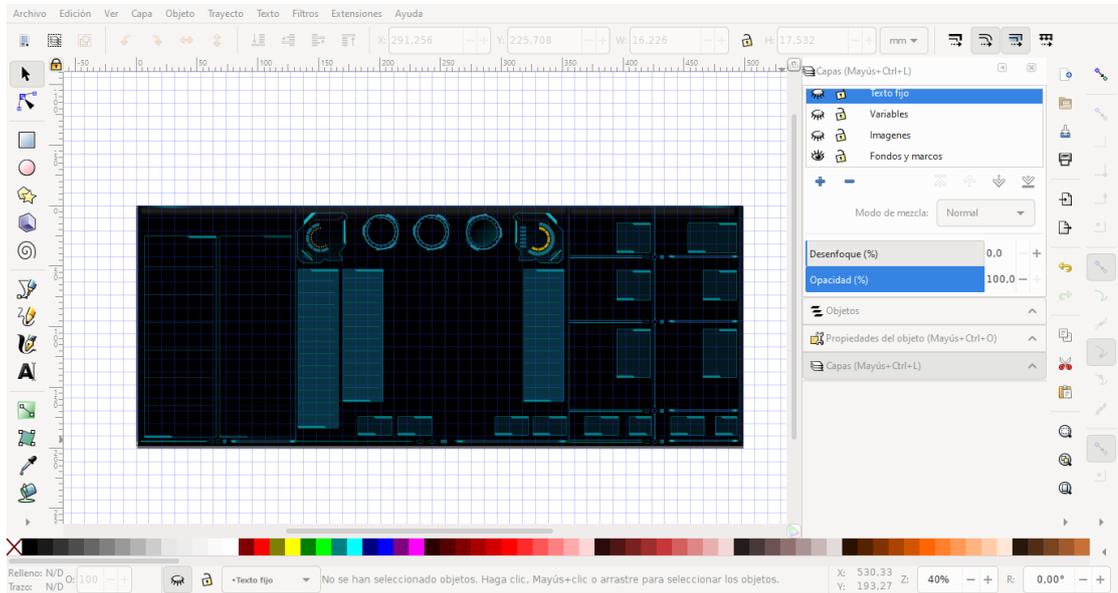


Figura 2-12 Capa: Fondos y Marcos (Los autores, 2024)

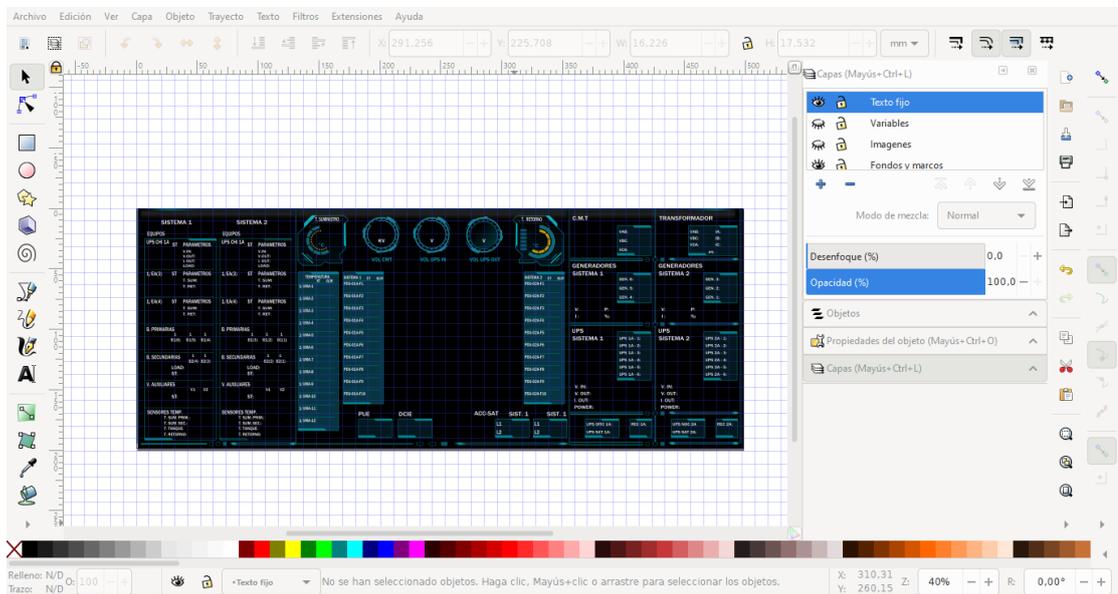


Figura 2-13 Capas: Fondos y Marcos más Texto fijo (Los autores, 2024)

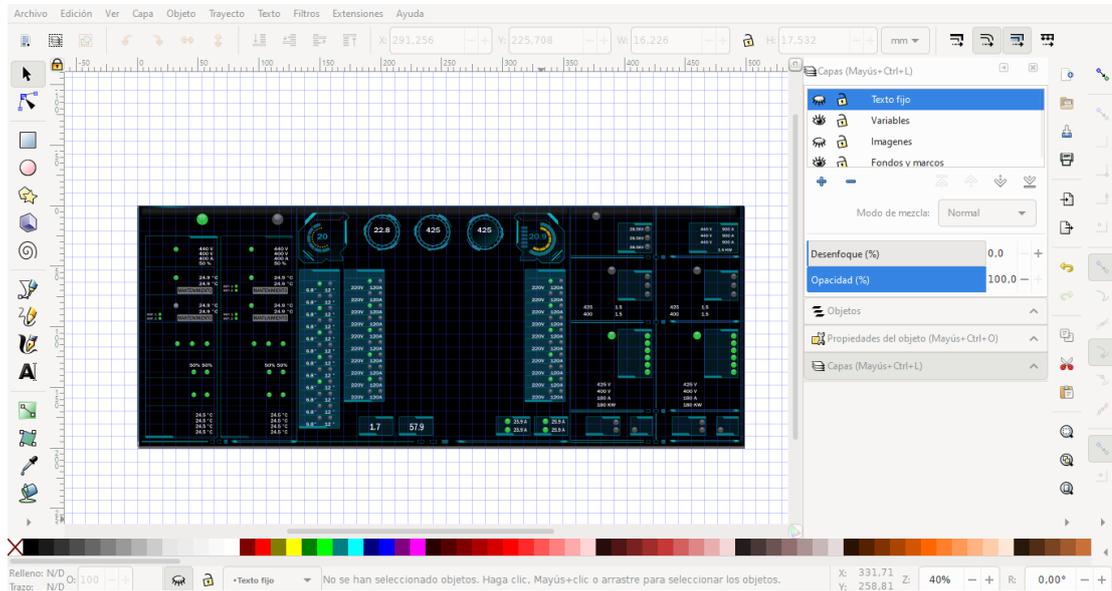


Figura 2-14 Fondos y Marcos más Variables (Los autores, 2024)

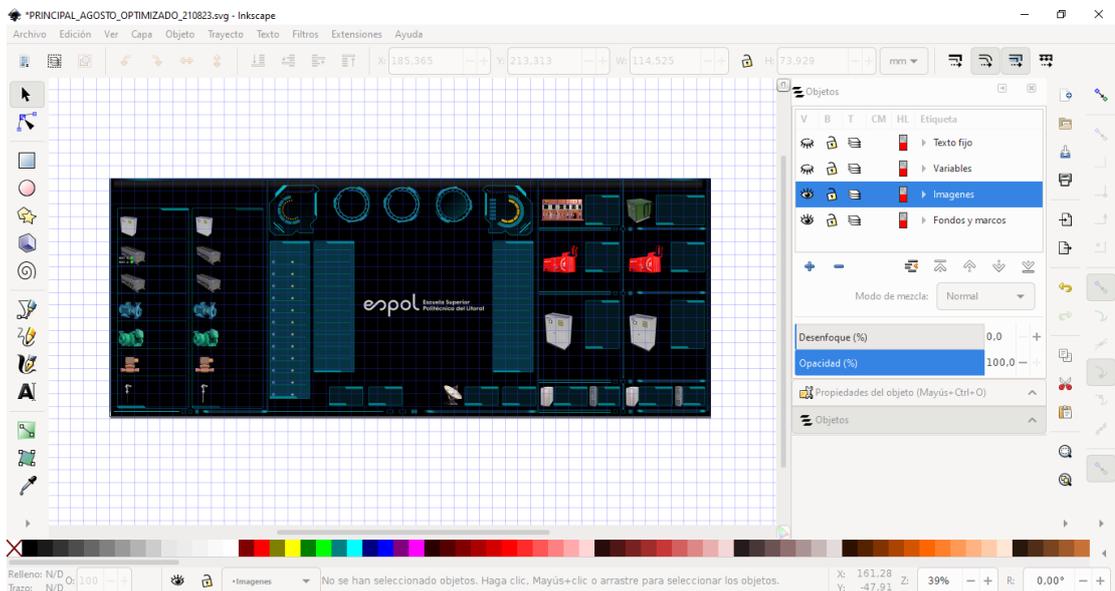


Figura 2-15 Fondos y Marcos más Imágenes (Los autores, 2024)

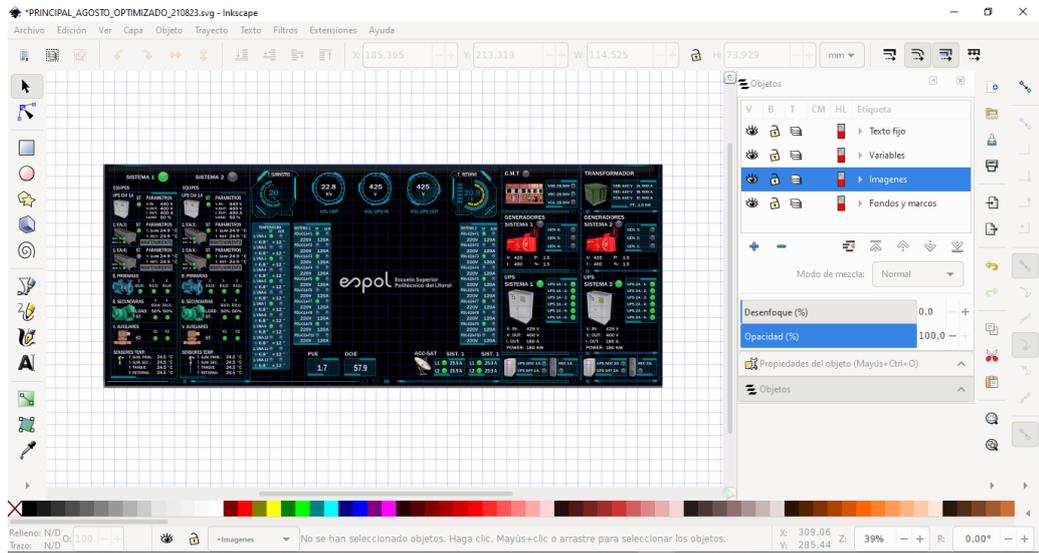


Figura 2-16 Diseño de pantalla completa (Los autores, 2024)

El proyecto incluye un encabezado de menú, diseñado como una imagen SVG en INKSCAPE. Este encabezado facilitará la navegación entre pantallas mediante enlaces interactivos.

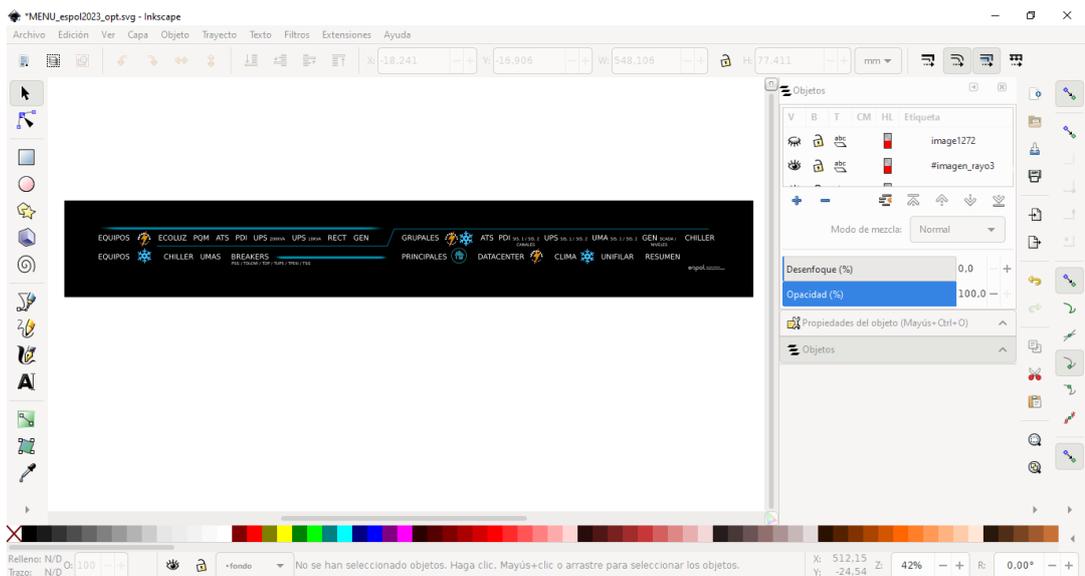


Figura 2-17 Diseño de MENU para navegación entre pantallas (Los autores, 2024)

Para la presentación de los diferentes equipos se realizó una imagen a modo de plantilla en la cual varían las imágenes de los diferentes equipos, los nombres, y las variables a mostrar.

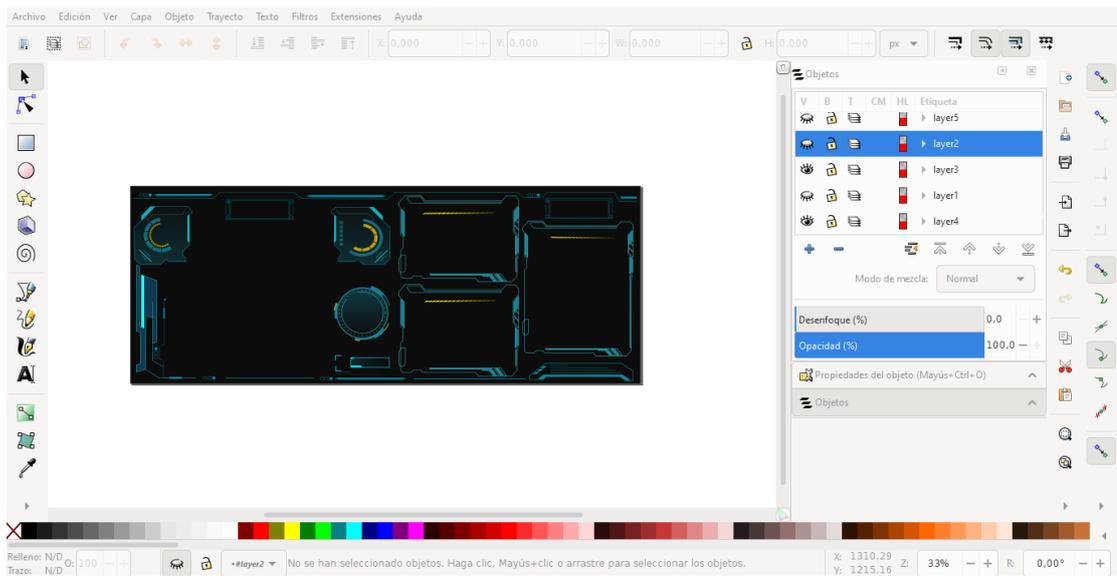


Figura 2-18 Imagen a modo de plantilla (Los autores, 2024)

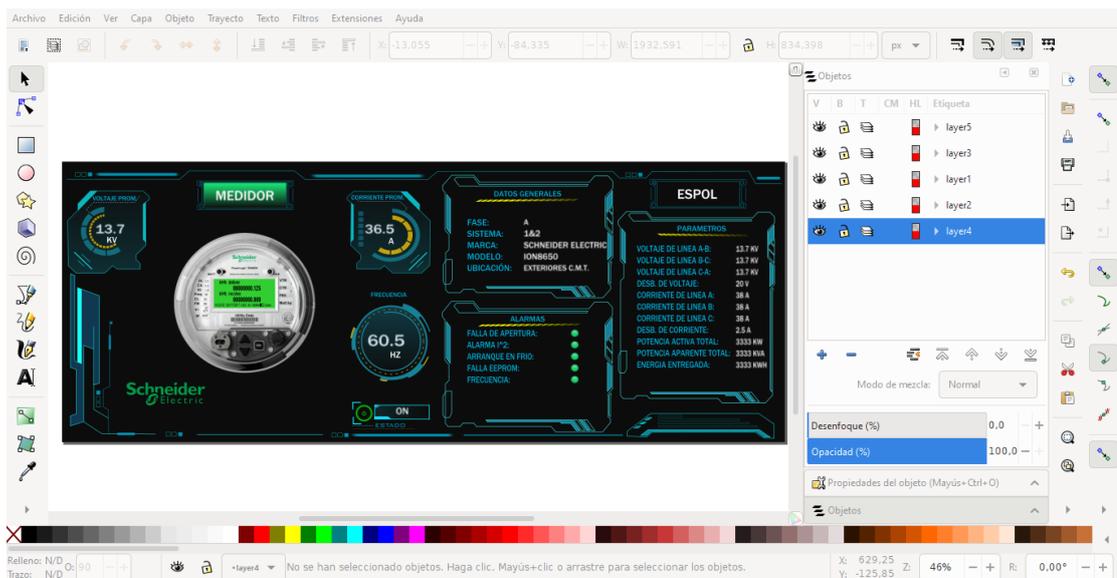


Figura 2-19 Imagen de equipo medidor de energía (Los autores, 2024)

2.3.2 Desarrollo del dashboard

Se empleará la plataforma de visualización de métricas Grafana, que ofrece la capacidad de representar gráficamente la información de los equipos mediante dashboards personalizados.

2.3.2.1 Creación de Grafana Plugins para Elastic

A continuación, se presenta una guía rápida para crear plugins adaptados a nuestras necesidades. Esta guía está diseñada para comenzar la creación de plugins para Elasticsearch utilizando Grafana y facilitará la configuración de un entorno de desarrollo local.

Estructura de Archivos

Se tiene que replicar la misma estructura de archivos ya que la carpeta grafana-data tiene que ser creada:

- /GRAFANA_ELASTIC
- /grafana-data
- /grafana-plugins
- docker-compose.yml
- README.md

Requisitos

- Se tiene que instalar Git: <https://git-scm.com/>
- Se recomienda instalar nvm si usan windows (<https://github.com/coreybutler/nvm-windows>) y si usan linux (<https://github.com/nvm-sh/nvm>) o Node.js directamente si tienen problemas con la instalación de nvm.
- Se tiene que instalar Yarn. <https://yarnpkg.com/>.
- Se tiene que instalar Docker. <https://www.docker.com/>. Si usan windows se sigue este documento: <https://docs.docker.com/docker-for-windows/install/>

Si usan linux escojan la distribucion y sigan el documento:
<https://docs.docker.com/engine/install/>

Descargar Proyecto

- Hacer una cuenta en github.
- Ir a la siguiente direccion:
https://github.com/nob3500/GRAFANA_PLUGINS.
- Realizar un Fork del proyecto.
- Hacer un directorio en la PC.
- Luego realizar un clon del proyecto con el siguiente comando: git clone
[https://github.com/\[USER\]/GRAFANA_PLUGINS.git](https://github.com/[USER]/GRAFANA_PLUGINS.git)

Gestion de git 1

- Crear un nuevo ramal (branch): git branch [nombre del nuevo ramal]
- Ir al nuevo ramal (branch): git checkout [nombre del nuevo ramal]

Arrancar grafana

- Dentro del directorio del proyecto ejecutar: docker-compose up -d

Inicializacion del Plugin

- Ir dentro del directorio grafana-plugins
- Correr el comando: npx @grafana/toolkit plugin:create my-plugin
- Correr el comando: cd my-plugin
- Correr el comando: yarn install
- Correr el comando: yarn dev
- En el arvhivo .gitignore borrar la linea con el nombre: dist/

Gestion de plugin

- Dentro de la carpeta del plugin.
- Cada vez que se haga un cambio correr el comando: yarn dev

- Para evitar compilar siempre y compilar automáticamente ejecutar: yarn watch
- Una vez concluido el plugin ejecutar: yarn build

Gestión de git 2

Desde el nuevo branch realizar lo siguiente: Correr estos comandos siempre que se quiera guardar el estado del trabajo realizado localmente:

- git add --all
- git commit -m "Mensaje de avance del proyecto"
- Para revisar los avances ejecutar lo siguiente: git log

Gestión de proyecto de git a github

- Para subir el proyecto a github se ejecuta el siguiente comando: git push
- Para actualizar el proyecto en otro host se usa: git pull

Juntar el proyecto al original

En su proyecto Fork de github hacer click en "pull request" y esperar el comentario del propietario original del proyecto, si se necesita hacer cambios hacer un git push nuevamente y pedir un "pull request".

Actualizar su copia del proyecto

- Agregar el repositorio original al proyecto fork (esto se hace una sola vez):
git remote add upstream
https://github.com/nob3500/GRAFANA_PLUGINS.git
- Buscar todos los cambios desde el repositorio principal: git fetch upstream
- Cambiarse al ramal maestro desde el fork: git checkout master
- Juntar los cambios del proyecto original desde del fork: git merge upstream/master

Actualizar el ramal

Desde el ramal que se encuentra o el nuevo creado hacer lo siguiente: git rebase master

Adicional a esta guía y para el desarrollo o la creación y la ejecución de los plugins en Grafana es necesario instalar dependencias.

Dependencias a instalar

Una dependencia de proyecto se define como la situación en la que un proyecto no puede iniciarse hasta que se haya completado otro proyecto anterior. En otras palabras, existe una dependencia cuando las tareas de un proyecto están interrelacionadas con las de otro en la gestión y ejecución de proyectos. Estas dependencias son cruciales para la correcta ejecución de nuestros dashboards en Grafana

- "@grafana/data": "^7.4.5",
- "@grafana/toolkit": "^7.4.5",
- "@grafana/ui": "^7.4.5",
- "emotion": "10.0.27"
- "@typescript-eslint/eslint-plugin": "^3.3.0",
- "sweetalert": "^2.1.2",
- "typescript": "^3.9.10"

2.3.2.2 Programación del Dashboard

Hemos elegido Visual Studio Code para programar nuestro dashboard y visualizar el código de manera clara y comprensible. Esto facilitará la realización de cambios futuros en el diseño, así como la implementación de mejoras y la integración de nuevos equipos o variables que puedan surgir con el tiempo.

Para nuestro proyecto de sistema de monitoreo, y siguiendo los pasos para la creación de un plugin, se generan por defecto varias carpetas con diversos componentes. En particular, nos centraremos en la carpeta src, donde crearemos una subcarpeta llamada componentes. En esta subcarpeta, incluiremos o escribiremos el código correspondiente a la imagen creada en INKSCAPE, un archivo para el texto fijo y un segundo archivo para las variables o datos que cambiarán en el tiempo añadiendo la extensión **.tsx**, archivos llamados principal2.tsx y variables2.tsx

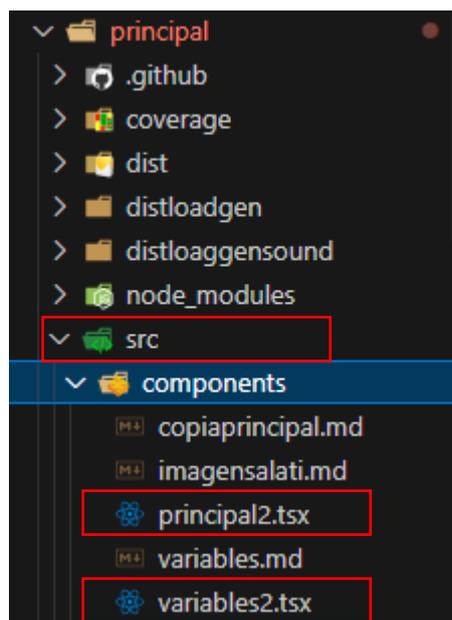


Figura 2-20 Árbol de Proyecto Principal (Los autores, 2024)

Para incorporar nuestra imagen en el código, utilizaremos una herramienta en línea accesible desde cualquier navegador web. Simplemente buscamos "Transform.tools" en el buscador, cargamos la imagen diseñada en INKSCAPE, y la herramienta generará automáticamente el código fuente correspondiente. Este código fuente debe ser integrado en nuestro proyecto, separándolo en texto fijo y variables según sea necesario.

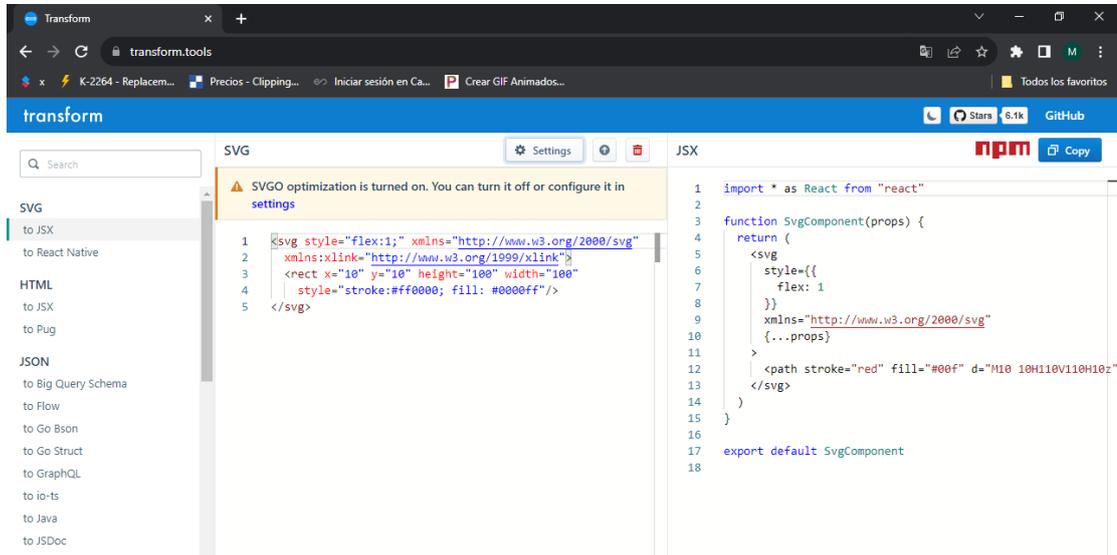


Figura 2-21 Transform.tools (Los autores, 2024)

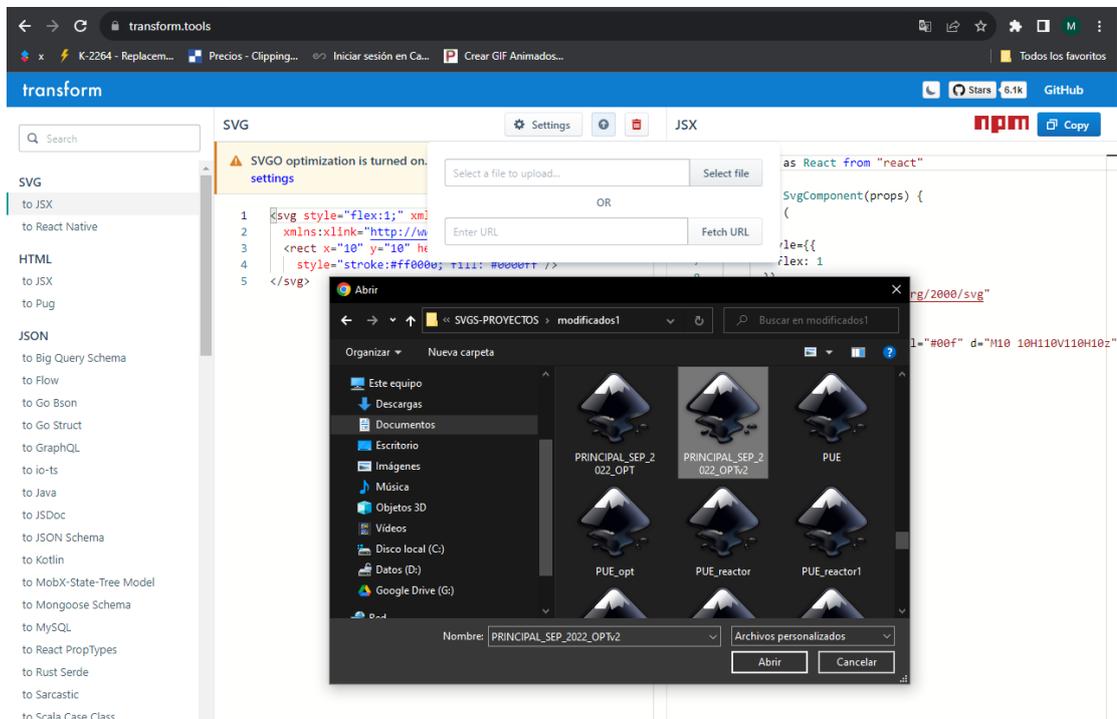


Figura 2-22 Insertar imagen desarrollada en INKSCAPE (Los autores, 2024)

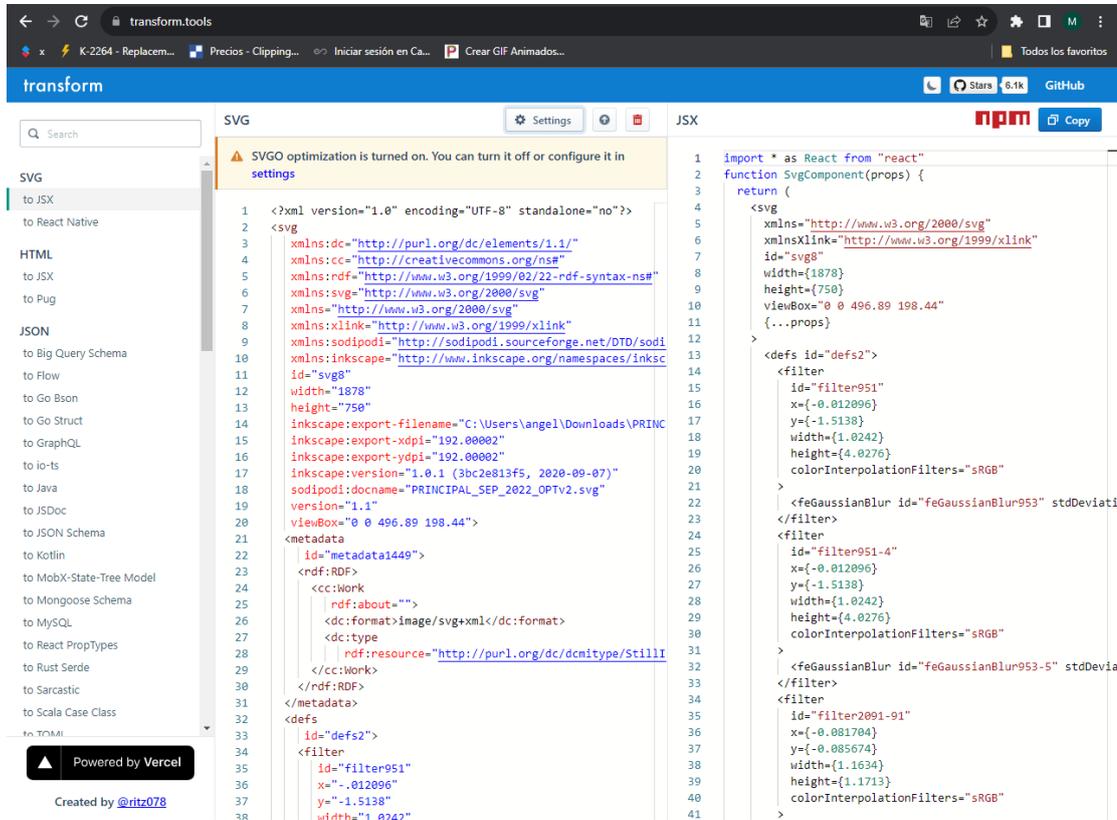


Figura 2-23 Imagen transformada a líneas de código (Los autores, 2024)

Una vez transformada nuestra imagen creada en INKSCAPE y traducidas a líneas de código, separamos los códigos en 2 archivos, uno para los textos fijos (no cambiarán con respecto al tiempo) y el segundo para las variables las cuales se actualizarán a medida de que el dato cambie. El texto fijo lo adjuntamos al archivo creado **principal2.tsx** y las variables al archivo creado **variables2.tsx**. En nuestro programa de Visual Studio Code realizamos la programación respectiva para el llamado de las variables y la asignación de las mismas; las cuales fueron asignadas y nombradas en NodeRed en la cual se realiza la adquisición del dato de cada uno de los equipos mediante la programación por bloques utilizando el bloque adecuado para cada protocolo de comunicación.

```
principal > src > components > variables2.tsx > ParametrosElec
1 import React from 'react';
2
3 type ParametrosElec = {
4   Vab_Tr01: number;
5   Vbc_Tr01: number;
6   Vca_Tr01: number;
7   Ia_Tr01: number;
8   Ib_Tr01: number;
9   Ic_Tr01: number;
10  Pt_Tr01: number;
11  V_CMTAVG: number;
12  V_CMTL1: number;
13  V_CMTL2: number;
14  V_CMTL3: number;
15  P_SIS1: number;
16  P_SIS2: number;
17  PUE: number;
18  DCIE: number;
19 };
20 type ParametrosClima = {
21   T_sumin: number;
22   T_ret: number;
23   uma1_TS: number; uma2_TS: number; uma3_TS: number; uma4_TS: number; uma5_TS: number; uma6_TS: number;
24   uma7_TS: number; uma8_TS: number; uma9_TS: number; uma10_TS: number; uma11_TS: number; uma12_TS: number;
25   uma1_TR: number; uma2_TR: number; uma3_TR: number; uma4_TR: number; uma5_TR: number; uma6_TR: number;
26   uma7_TR: number; uma8_TR: number; uma9_TR: number; uma10_TR: number; uma11_TR: number; uma12_TR: number;
27 };
28 type ParametrosUPS_SIS1 = {
29   V_in: number; V_out: number; I_out: number; P_out: number; Load: number;
30   V_inCHI: number; V_outCHI: number; I_outCHI: number; P_outCHI: number; LoadCHI: number;
31   V_PDI1A_F1: number; V_PDI1A_F2: number; V_PDI1A_F3: number; V_PDI1A_F4: number; V_PDI1A_F5: number;
32   V_PDI1A_F6: number; V_PDI1A_F7: number; V_PDI1A_F8: number; V_PDI1A_F9: number; V_PDI1A_F10: number;
```

Figura 2-24 Programa generado para variables (Los autores, 2024)

En la primera línea React es una biblioteca para construir interfaces de usuario. React es utilizado con otras bibliotecas para renderizar en ciertos entornos.

En nuestro código de variables declaramos tipos de variables a nuestro gusto las cuales estarán agrupadas y serán de tipo numéricas y tipo string, por ejemplo asignamos un ID como Vab_TR01 que significa Voltaje ab del transformador 01 y que es variable de tipo numérica esto es en types ParametrosElec, así mismo crearemos tipos de variables para los diversos equipos a monitorear en los tipos tenemos declarado como:

ParametrosElec, ParametrosClima, ParametrosUPS_SIS1,
ParametrosUPS_SIS2, ParametrosClima_SIS1, ParametrosClima_SIS2,
ParametrosGEN_SIS1, ParametrosGEN_SIS2, Estados_Principales,
Estados_SIS1, Estados_SIS2, Alarmas, Alarmas_SIS1, Alarmas_SIS2

```
principal > src > components > variables2 > Variables2
143 };
144
145 export interface DataPrincipal {
146   ParametrosElec: ParametrosElec;
147   ParametrosClima: ParametrosClima;
148   ParametrosUPS_SIS1: ParametrosUPS_SIS1;
149   ParametrosUPS_SIS2: ParametrosUPS_SIS2;
150   ParametrosClima_SIS1: ParametrosClima_SIS1;
151   ParametrosClima_SIS2: ParametrosClima_SIS2;
152   ParametrosGEN_SIS1: ParametrosGEN_SIS1;
153   ParametrosGEN_SIS2: ParametrosGEN_SIS2;
154   Estados_Principales: Estados_Principales;
155   Estados_SIS1: Estados_SIS1;
156   Estados_SIS2: Estados_SIS2;
157   Alarmas: Alarmas;
158   Alarmas_SIS1: Alarmas_SIS1;
159   Alarmas_SIS2: Alarmas_SIS2;
160 }
161 export const Variables2 = ({ParametrosElec, ParametrosClima, ParametrosUPS_SIS1, ParametrosUPS_SIS2, ParametrosCl
162 ParametrosGEN_SIS1, ParametrosGEN_SIS2, Estados_Principales, Estados_SIS1, Estados_SIS2, Alarmas, Alarmas_SIS1, A
163 return [
164   <g id="layer5" transform="translate(-14.288)">
165     <text
166       id="tsum"
167       transform="scale(1.0508 .95167)"
168       x={154.19633}
169       y={29.00918}
170       style={{
171         fontVariantCaps: "normal",
172         fontVariantEastAsian: "normal",
173         fontVariantLigatures: "normal",
174         fontVariantNumeric: "normal",
```

Figura 2-25 Exportando Interface (Los autores, 2024)

Creamos una interface que contendrá el tipo de variable declarada y se exportará como DataPrincipal

Adicional creamos una constante Variables2 para exportarla y contendrá los parámetros como los ya descritos.

En la línea 164 tendremos la imagen en líneas de código con respecto a las variables (datos que cambiarán con el tiempo)

```
principal2.tsx 9+ • variables2.tsx 4 X TS dataPrincipal.ts 1 SimplePanel.tsx TS alarmasStyles.ts TS estadosStyles.ts
principal > src > components > variables2.tsx > Variables2
404     {ParametrosElec.V_CMTL3}KV
405   </tspan>
406 </text>
407 <text
408   id="vab_tr01a"
409   transform="matrix(1.046 0 0 .95601 14.288 0)"
410   x={442.06537}
411   y={21.427717}
412   style={{
413     fontVariantCaps: "normal",
414     fontVariantEastAsian: "normal",
415     fontVariantLigatures: "normal",
416     fontVariantNumeric: "normal",
417     lineHeight: 1.25
418   }}
419   xmlSpace="preserve"
420   fill="#fff"
421   fontFamily="Franklin Gothic Medium"
422   fontSize="3.5278px"
423   letterSpacing={0}
424   strokeWidth={0.4333}
425   wordSpacing={0}
426 >
427   <tspan
428     id="tspan998-6-1-4-2-0-1-7-7-4-2"
429     x={442.06537}
430     y={21.427717}
431     style={{
432       fontVariantCaps: "normal",
433       fontVariantEastAsian: "normal",
434       fontVariantLigatures: "normal",
435       fontVariantNumeric: "normal",
436       lineHeight: 3.25
437     }}
438     fill="#fff"
439     fontFamily="Franklin Gothic Medium"
440     fontSize="3.5278px"
441     strokeWidth={0.4333}
442   >
443     {ParametrosElec.Vab_Tr01}V
444   </tspan>
445 </text>
446 <text
447   id="vbc_tr01a"
448   transform="matrix(1.046 0 0 .95601 14.288 0)"
449   x={442.06537}
```

Figura 2-26 Ejemplo asignar el tipo de variable a la imagen (Los autores, 2024)

Tomaremos el ejemplo anterior de Vab_Tr01, en este caso el en INKSCAPE nosotros le asignamos a los objetos un ID único el cual para mejor guiarnos deberemos de llamarle de igual manera o que tengan algo de similitud para poder apuntar más rápido, en la línea 408 del código de programación tenemos el id del objeto llamado vab_tr01a; y nos redirigimos a la línea 443 en la cual estaba un valor que le habíamos asignado a nuestra imagen para el diseño, esto se borra y se añade el tipo de variable en la cual se asignó: ParametrosElec.Vab_Tr01a

```
principal2.tsx 9+ • variables2.tsx 4 x TS dataPrincipal.ts 1 SimplePanel.tsx TS alarmasStyles.ts TS estadosStyles.ts
principal > src > components > variables2.tsx > Variables2
746   cx={433.22}
747   cy={58.482}
748   r={2.3465}
749   fill="#4d4d4d"
750   opacity={0}
751   paintOrder="markers stroke fill"
752   className={Alarmas_SIS1.gen1}
753 />
754 <circle
755   id="st_gen2"
756   cx={433.22}
757   cy={65.774}
758   r={2.3465}
759   fill="#4d4d4d"
760   paintOrder="markers stroke fill"
761   className={Estados_SIS1.gen2}
762 />
763 <ellipse
764   id="st_gen2brillo"
765   transform="matrix(.34362 0 0 .17988 419.74 32.847)"
766   cx={39.31}
767   cy={175.59}
768   rx={4.2297}
769   ry={4.0337}
770   fill="fff"
771   filter="url(#filter2091-9-7-7-5-54-9-8-4)"
772   opacity={0.29}
773   paintOrder="markers stroke fill"
774 />
```

Figura 2-27 Ejemplo asignar el tipo de variable a la imagen (Los autores, 2024)

Tenemos otro ejemplo similar en la línea 755 declaramos al objeto con id st_gen2 en este caso es un círculo o led indicador que representa el estado del generador 2 si está encendido o apagado aquí creamos una propiedad llamada className la cual contendrá el tipo de variable Estados_SIS1.gen2 de tipo string. Ésto es de manera resumida para poder declarar el tipo de variable a los objetos creados en la imagen INKSCAPE traducidos a líneas de código

```
principal2.tsx 9+ • TS dataPrincipal.ts 1 variables2.tsx 4 SimplePanel.tsx TS alarmasStyles.ts TS estadosStyles.ts stylesPo ...
principal > src > components > principal2.tsx > Principal2
1 import React from 'react'
2 import { Variables2, DataPrincipal } from './variables2';
3
4 const Principal2 = ({ParametrosElec, ParametrosClima, ParametrosUPS_SIS1, ParametrosUPS_SIS2,
5 ParametrosClima_SIS1, ParametrosClima_SIS2, ParametrosGEN_SIS1, ParametrosGEN_SIS2, Estados_Principales,
6 Estados_SIS1, Estados_SIS2, Alarmas, Alarmas_SIS1, Alarmas_SIS2 }: DataPrincipal) => {
7 //DECLARAR VARIABLES PARA EQUIPOS A VINCULAR
8 { ...
9 }
10 }
11
12 return (
13 <svg
14 id="svg8"
15 //width={1878}
16 //height={750}
17 viewBox="0 0 496.89 198.44"
18 >
19 <defs id="defs2">
20 <filter
21 id="filter951"
22 x={-0.012096}
23 y={-1.5138}
24 width={1.0242}
25 height={4.0276}
26 colorInterpolationFilters="sRGB"
27 >
28 <feGaussianBlur id="feGaussianBlur953" stdDeviation={0.68907906} />
29 </filter>
30 <filter
31 id="filter951-4"
32 x={-0.012096}
33 y={-1.5138}
34 >
```

Figura 2-28 Código generado de la imagen (Los autores, 2024)

En la primera línea React es una biblioteca para construir interfaces de usuario. React es utilizado con otras bibliotecas para renderizar en ciertos entornos. La segunda línea del código manda a importar constante declarada en el archivo variables2.tsx, esto con el fin de mantener varios archivos separados que dependan del otro. Y la interface DataPrincipal

En la tercera línea declaramos una constante Principal2 que contendrán los parámetros ya declarados en el archivo variables2

```
9067 <text
9068   id="text_trafo"
9069   transform="scale(1.046 .95601)"
9070   x={410.38412}
9071   y={10.256429}
9072   style={{
9073     fontVariantCaps: "normal",
9074     fontVariantEastAsian: "normal",
9075     fontVariantLigatures: "normal",
9076     fontVariantNumeric: "normal",
9077     lineHeight: 1.25
9078   }}
9079   xmlSpace="preserve"
9080   fill="#fff"
9081   fontSize="5.1949px"
9082   strokeWidth={0.4333}
9083 >
9084   <tspan
9085     id="tspan998-68"
9086     x={410.38412}
9087     y={10.256429}
9088     style={{
9089       fontVariantCaps: "normal",
9090       fontVariantEastAsian: "normal",
9091       fontVariantLigatures: "normal",
9092       fontVariantNumeric: "normal",
9093       lineHeight: 3.25
9094     }}
9095     fill="#fff"
9096     fontFamily="Franklin Gothic Medium"
9097     fontSize="5.1949px"
9098     strokeWidth={0.4333}
9099   >
9100     {"TR-01"}
9101   </tspan>
9102 </text>
```

Figura 2-29 Código texto fijo (Los autores, 2024)

Seguimos tomando como ejemplo transformador el objeto creado tiene un id de text_trafo el cual colocamos para nuestro entendimiento, en la línea 9100 tenemos el texto fijo el cual se mostrará en la pantalla; nosotros podemos modificarlo si así lo deseamos.

```
12569     >
12570         { "% LOAD:" }
12571     </tspan>
12572 </text>
12573 </g>
12574 <Variables2
12575   ParametrosElec={ParametrosElec}
12576   ParametrosClima={ParametrosClima}
12577   ParametrosUPS_SIS1={ParametrosUPS_SIS1}
12578   ParametrosUPS_SIS2={ParametrosUPS_SIS2}
12579   ParametrosClima_SIS1={ParametrosClima_SIS1}
12580   ParametrosClima_SIS2={ParametrosClima_SIS2}
12581   ParametrosGEN_SIS1={ParametrosGEN_SIS1}
12582   ParametrosGEN_SIS2={ParametrosGEN_SIS2}
12583   Estados_Principales={Estados_Principales}
12584   Estados_SIS1={Estados_SIS1}
12585   Estados_SIS2={Estados_SIS2}
12586   Alarmas={Alarmas}
12587   Alarmas_SIS1={Alarmas_SIS1}
12588   Alarmas_SIS2={Alarmas_SIS2}
12589 </>
12590 </svg>
12591 );
12592 };
12593 export default Principal2;
```

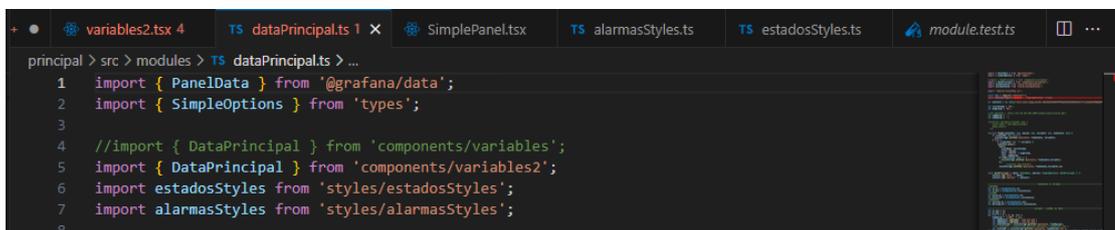
Figura 2-30 Variables por segmentos (Los autores, 2024)

Al final de nuestro código llamamos a nuestras variables con sus propiedades para después exportar la constante o el código llamado Principal2

```
principal > src > styles > TS estadosStyles.ts > [e] getStyles > [e] stylesFactory() callback > [e] on
1 import { stylesFactory } from '@grafana/ui';
2 import { css, keyframes } from 'emotion';
3
4 const parpadeo = keyframes`
5   0% { opacity: 1.0; };
6   100% { opacity: 0.0; };
7 `;
8
9 const getStyles = stylesFactory(() => {
10   return {
11     sinconexion: css`
12       fill: #999;
13     `,
14     on: css`
15       fill: #2fc43b;
16       //animation: ${parpadeo} 1s ease infinite;
17     `,
18     alarma: css`
19       fill: red;
20       animation: ${parpadeo} 1s ease infinite;
21     `,
22     warning: css`
23       fill: orange;
24       animation: ${parpadeo} 1s ease infinite;
25     `,
26   };
27 });
28
29 const styles = getStyles();
30
31 export default styles;
```

Figura 2-31 Archivo estados (Los autores, 2024)

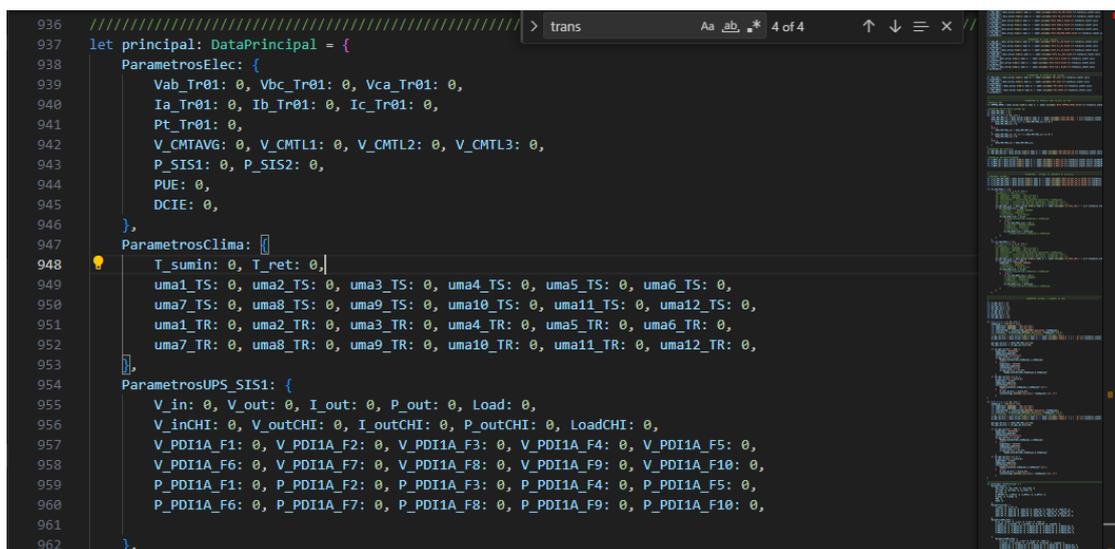
Adicional creamos un archivo llamado estadosStyles.ts en una subcarpeta llamada styles en la carpeta src para poder dar animación y estilizar en nuestro caso los led's indicadores de los diferentes equipos a mostrar, aquí se podrá darle color a través de la propiedad fill, algo de animación para que emita un parpadeo y tiempo. Debemos usar e importar las funciones como stylesFactory, css y keyframes.



```
principal > src > modules > TS dataPrincipal.ts > ...
1 import { PanelData } from '@grafana/data';
2 import { SimpleOptions } from 'types';
3
4 //import { DataPrincipal } from 'components/variables';
5 import { DataPrincipal } from 'components/variables2';
6 import estadosStyles from 'styles/estadosStyles';
7 import alarmasStyles from 'styles/alarmasStyles';
8
```

Figura 2-32 Archivo dataPrincipal (Los autores, 2024)

En este archivo realizaremos los cálculos, la asignación de la variable tal cual como fue nombrada en NodeRed para poder cargar el dashboard al grafana, aquí importamos propiedades de grafana y archivos como el de estados para la correcta asignación



```
936 /////////////////////////////////////////////////// > trans Aa _ab_* 4 of 4 ↑ ↓ ≡ × //
937 let principal: DataPrincipal = {
938   ParametrosElec: {
939     Vab_Tr01: 0, Vbc_Tr01: 0, Vca_Tr01: 0,
940     Ia_Tr01: 0, Ib_Tr01: 0, Ic_Tr01: 0,
941     Pt_Tr01: 0,
942     V_CMTAVG: 0, V_CMTL1: 0, V_CMTL2: 0, V_CMTL3: 0,
943     P_SIS1: 0, P_SIS2: 0,
944     PUE: 0,
945     DCIE: 0,
946   },
947   ParametrosClima: {
948     T_sumin: 0, T_ret: 0,
949     uma1_TS: 0, uma2_TS: 0, uma3_TS: 0, uma4_TS: 0, uma5_TS: 0, uma6_TS: 0,
950     uma7_TS: 0, uma8_TS: 0, uma9_TS: 0, uma10_TS: 0, uma11_TS: 0, uma12_TS: 0,
951     uma1_TR: 0, uma2_TR: 0, uma3_TR: 0, uma4_TR: 0, uma5_TR: 0, uma6_TR: 0,
952     uma7_TR: 0, uma8_TR: 0, uma9_TR: 0, uma10_TR: 0, uma11_TR: 0, uma12_TR: 0,
953   },
954   ParametrosUPS_SIS1: {
955     V_in: 0, V_out: 0, I_out: 0, P_out: 0, Load: 0,
956     V_inCHI: 0, V_outCHI: 0, I_outCHI: 0, P_outCHI: 0, LoadCHI: 0,
957     V_PDI1A_F1: 0, V_PDI1A_F2: 0, V_PDI1A_F3: 0, V_PDI1A_F4: 0, V_PDI1A_F5: 0,
958     V_PDI1A_F6: 0, V_PDI1A_F7: 0, V_PDI1A_F8: 0, V_PDI1A_F9: 0, V_PDI1A_F10: 0,
959     P_PDI1A_F1: 0, P_PDI1A_F2: 0, P_PDI1A_F3: 0, P_PDI1A_F4: 0, P_PDI1A_F5: 0,
960     P_PDI1A_F6: 0, P_PDI1A_F7: 0, P_PDI1A_F8: 0, P_PDI1A_F9: 0, P_PDI1A_F10: 0,
961   },
962 },
```

Figura 2-33 Interface con los parámetros de las variables (Los autores, 2024)

```

710 //-----PARAMETROS DE TRANSFORMADOR (PQM)-----
711 let VAB_TR01 = data.series.find(({ name }) => name?.includes('DATA.VOL_VAB.VALUE'))?.fields[1].state?.calcs
712   .lastNotNull;
713 let VBC_TR01 = data.series.find(({ name }) => name?.includes('DATA.VOL_VBC.VALUE'))?.fields[1].state?.calcs
714   ?.lastNotNull;
715 let VCA_TR01 = data.series.find(({ name }) => name?.includes('DATA.VOL_VCA.VALUE'))?.fields[1].state?.calcs
716   ?.lastNotNull;
717 let Ia_TR01 = data.series.find(({ name }) => name?.includes('DATA.CURR_A.VALUE'))?.fields[1].state?.calcs
718   ?.lastNotNull;
719 let Ib_TR01 = data.series.find(({ name }) => name?.includes('DATA.CURR_B.VALUE'))?.fields[1].state?.calcs
720   ?.lastNotNull;
721 let Ic_TR01 = data.series.find(({ name }) => name?.includes('DATA.CURR_C.VALUE'))?.fields[1].state?.calcs
722   ?.lastNotNull;
723 let Pt_TR01 = data.series.find(({ name }) => name?.includes('DATA.REALPOW_3PHAS.VALUE'))?.fields[1].state?.ca
724   ?.lastNotNull;
725 //

```

Figura 2-34 Asignación de variable (Los autores, 2024)

Seguimos con el ejemplo del transformador el cual le asignaremos las variables nombradas en el NodeRed.

```

1445 //-----PARAMETROS TRANSFORMADOR (PQM)-----
1446 principal.ParametrosElec.Vab_Tr01 = Number.parseFloat(VAB_TR01?.toFixed(2));
1447 principal.ParametrosElec.Vbc_Tr01 = Number.parseFloat(VBC_TR01?.toFixed(2));
1448 principal.ParametrosElec.Vca_Tr01 = Number.parseFloat(VCA_TR01?.toFixed(2));
1449 principal.ParametrosElec.Ia_Tr01 = Number.parseFloat(Ia_TR01?.toFixed(2));
1450 principal.ParametrosElec.Ib_Tr01 = Number.parseFloat(Ib_TR01?.toFixed(2));
1451 principal.ParametrosElec.Ic_Tr01 = Number.parseFloat(Ic_TR01?.toFixed(2));
1452 let Pot3F = Pt_TR01 / 100;
1453 principal.ParametrosElec.Pt_Tr01 = Number.parseFloat(Pot3F?.toFixed(1));
1454 //

```

Figura 2-35 Lectura de la variable (Los autores, 2024)

```
principal > src > SimplePanel.tsx > ...
> trans Aa_ab_* No results ↑ ↓ ≡ ×

1 import React from 'react';
2 import { PanelProps } from '@grafana/data';
3 import { SimpleOptions } from 'types';
4 import { css, cx } from 'emotion';
5 //import { stylesFactory, useTheme } from '@grafana/ui';
6 import { stylesFactory } from '@grafana/ui';
7 //import Principal from './components/principal';
8 import Principal2 from './components/principal2';
9
10 import dataPrincipal from 'modules/dataPrincipal';
11 interface Props extends PanelProps<SimpleOptions> {}
12
13 export const SimplePanel: React.FC<Props> = ({ options, data, width, height }) => {
14   //const theme = useTheme();
15   const styles = getStyles();
16
17   let principal = dataPrincipal(data, options);
18   return (
19     <div
20       className=cx(
21         styles.wrapper,
22         css`
23           width: ${width}px;
24           height: ${height}px;
25         `
26       )
27     >
28     <Principal2
29       ParametrosElec={principal.ParametrosElec}
30       ParametrosClima={principal.ParametrosClima}
31       ParametrosUPS_SIS1={principal.ParametrosUPS_SIS1}
32       ParametrosUPS_SIS2={principal.ParametrosUPS_SIS2}
33       ParametrosClima_SIS1={principal.ParametrosClima_SIS1}
34       ParametrosClima_SIS2={principal.ParametrosClima_SIS2}
35       ParametrosGEN_SIS1={principal.ParametrosGEN_SIS1}
36       ParametrosGEN_SIS2={principal.ParametrosGEN_SIS2}
37       Estados_Principales={principal.Estados_Principales}
38       Estados_SIS1={principal.Estados_SIS1}
39       Estados_SIS2={principal.Estados_SIS2}
40       Alarmas={principal.Alarmas}
41       Alarmas_SIS1={principal.Alarmas_SIS1}
42       Alarmas_SIS2={principal.Alarmas_SIS2}
43     />
44   </div>
```

Figura 2-36 Programación en el Panel de Grafana (Los autores, 2024)

Y para finalizar en la carpeta llamada SimplePanel.tsx creada por default, es dónde llamamos aquí a todo nuestro proyecto para que sea presentado en Grafana.

El proceso de la presentación del dato como tal no termina ahí; en nuestro aplicativo de grafana debemos crear el Data Source al cual debemos apuntar y obtener las variables para luego mostrarlas en nuestro DASHBOARD creado.

Nos dirigimos a nuestro panel de navegación de lado izquierdo de la pantalla y seleccionamos configuración se despliega una lista y seleccionamos Data

Sources, en este apartado debemos de configurar la IP del servidor y el índice en dónde se encuentran los equipos.

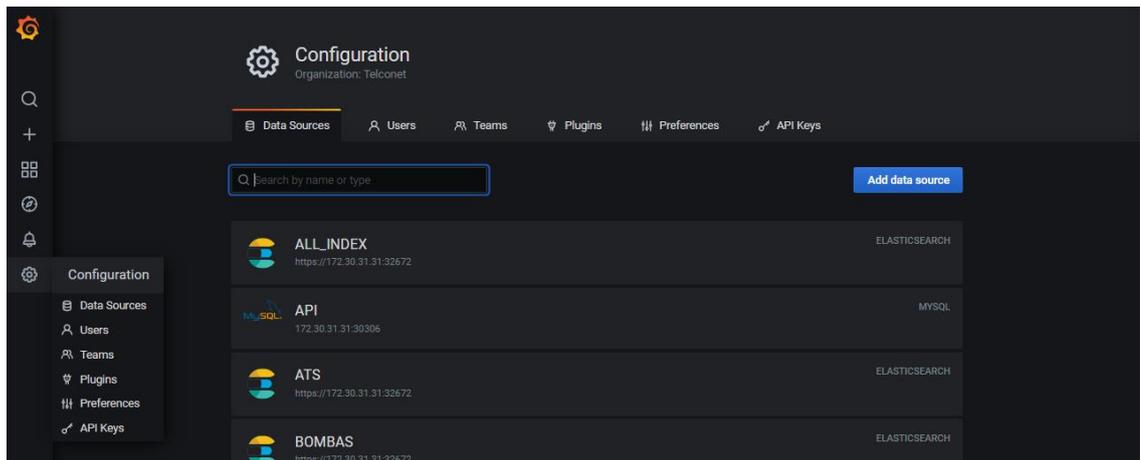


Figura 2-37 Data Sources (Los autores, 2024)

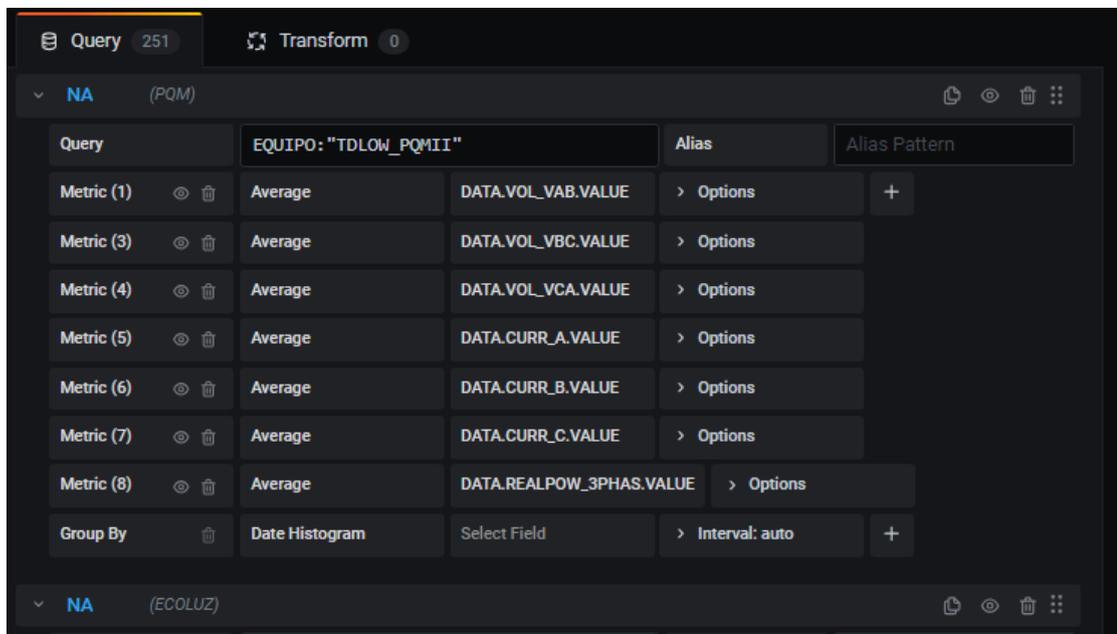


Figura 2-38 Variables comunicadas o leídas en el Grafana (Los autores, 2024)



Figura 2-39 Presentación del Dashboard en Grafana (Los autores, 2024)

La pantalla o dashboard presentada es a la cual hacemos referencia en el proyecto de titulación en la cual se pueden observar los diferentes equipos que se encuentran ubicados y forman parte de la infraestructura operativa del centro de datos, los indicadores verdes muestran equipos encendidos y los grises apagados, adicional muestran valores como potencia, voltaje, corriente, temperaturas en tiempo real lo cual indica que cambiarán a medida que el dato se actualice.



Figura 2-40 Dashboard Equipo Medidor con datos reales (Los autores, 2024)

Pantalla o DASHBOARD cargado en Grafana con datos reales del equipo medidor de energía, en éste dashboard se pueden visualizar parámetros de voltaje a nivel de 13.8KV corrientes en media tensión, datos básicos del medidor, frecuencia, potencia y energía, estado de las líneas activas.



Figura 2-41 Dashboard Equipo Rectificador con datos reales (Los autores, 2024)

Pantalla o DASHBOARD cargado en Grafana con datos reales del equipo rectificador, nivel de voltaje de entrada y salida, corriente, temperatura, datos básicos del equipo y estados operativos.



Figura 2-42 Dashboard Equipos Generadores (Los autores, 2024)

Pantalla o DASHBOARD cargado en Grafana con datos reales de los generadores en el cual se visualizan datos de temperatura de refrigerante, presión de aceite, nivel de combustible, voltaje de batería, horas de operación, voltaje de línea promedio, corriente de línea promedio, FP, Energía, Potencia activa y aparente



Figura 2-43 Dashboard Resumen (Los autores, 2024)

Pantalla o DASHBOARD cargado en Grafana con datos reales de las cargas existentes en el Centro de Datos, datos que se van actualizando cada 5 segundos a medida que la carga aumenta o disminuye están representadas las cargas totales por sistemas. (carga de servidores, ac/dc, carga de clima, pérdidas, servicios generales), así como también la energía total del día 1 hasta el presente los KVAs consumidos totales y el PUE que se calcula de la siguiente forma:

PUE = Potencia eléctrica total del centro / Potencia eléctrica total consumida por los sistemas. El valor del PUE puede estar en un rango que va desde 1.0 hasta infinito. Un 1.0 de valor indicaría una eficiencia del 100%.

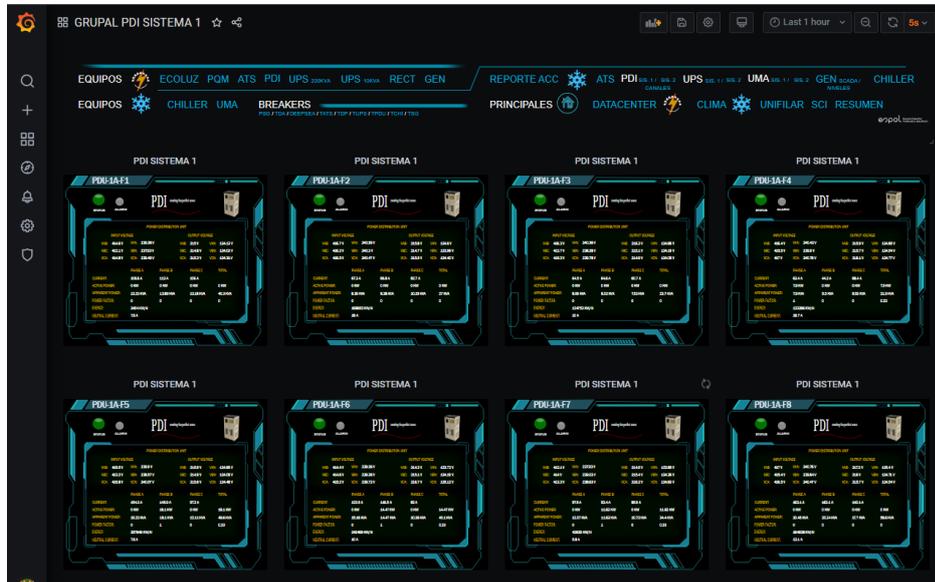


Figura 2-44 Dashboard Equipo PDI's (Los autores, 2024)

Pantalla o DASHBOARD cargado en Grafana de equipos PDI's con datos reales de las cargas existentes de la infraestructura de CENTRO DE DATOS las cuales están destinadas para reducir el voltaje del sistema de UPS's a nivel de 220 /120 Voltios y predispuestas para energizar los equipos de la sala de Servidores

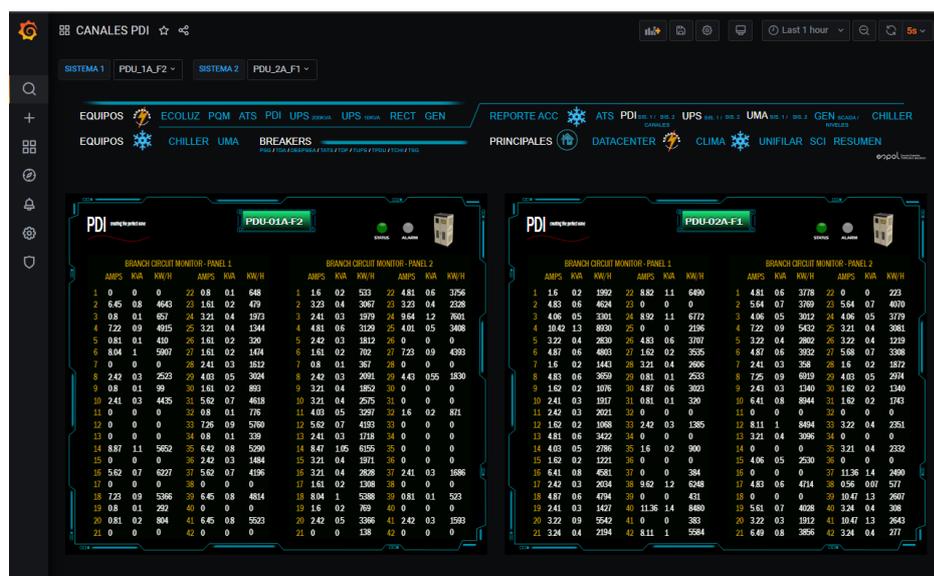


Figura 2-45 Dashboard Equipo PDI's breakers de distribución (Los autores, 2024)

Canales o Breakers, de las PDI's en los cuales se obtienen los datos de corriente, potencia y energía de las cargas de la sala de servidores.

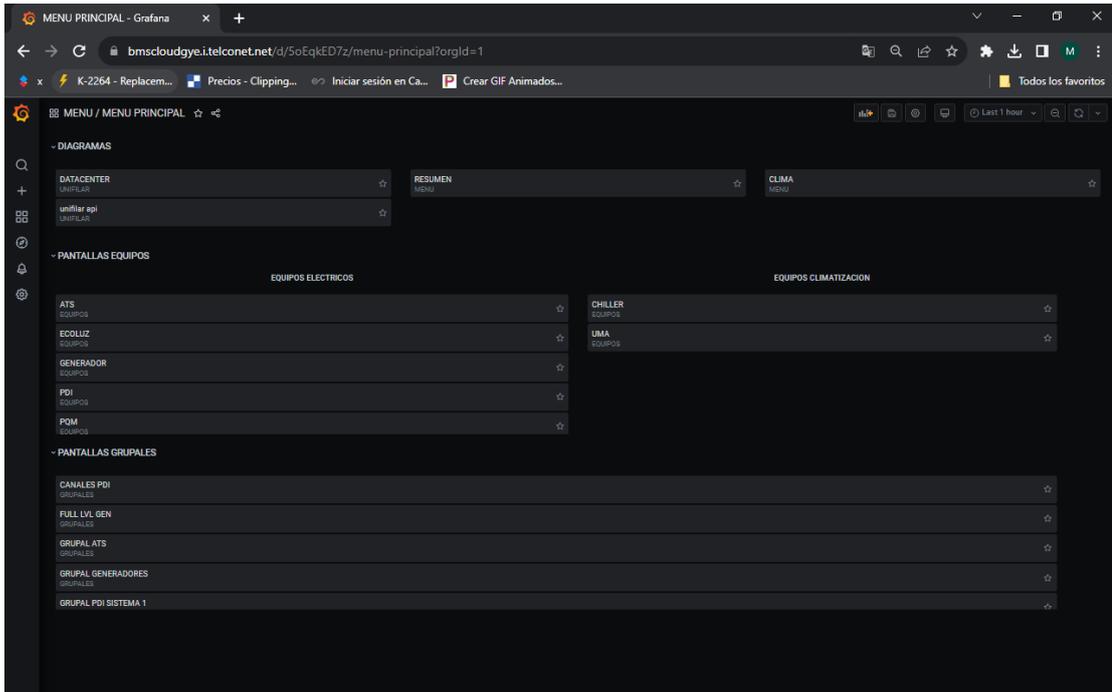


Figura 2-46 Menú de navegación en Grafana (Los autores, 2024)

Se desarrolla un dashboard en Grafana enlistando y agrupando los diferentes equipos de la infraestructura de soporte del CENTRO DE DATOS, en la cual con un click nos redirige al DASHBOARD mostrando la imagen del equipo y los datos mostrados.

2.3.2.3 Variables

La infraestructura del centro de datos está compuesta por diversos equipos organizados en dos sistemas principales: el sistema eléctrico y el sistema de climatización. Para cada uno de estos sistemas, fue necesario identificar las variables a muestrear y el protocolo de comunicación que utilizan. Esta información es crucial para seleccionar adecuadamente el bloque de programación en Node-

RED y las variables que se deben adquirir, así como para su correcta visualización en la pantalla de monitoreo.

CAPÍTULO 3

3. RESULTADOS Y ANÁLISIS

En esta sección se presentan los resultados de la implementación del sistema de monitoreo desarrollado utilizando plataformas de código abierto. Se abordarán los siguientes temas: una comparativa cualitativa del sistema de monitoreo respecto al anterior, los costos asociados a la ejecución del proyecto, y un análisis de los resultados obtenidos.

3.1 Comparativa cualitativa del sistema de monitoreo

- La implementación del proyecto ha minimizado la dependencia de licencias al aprovechar plataformas de código abierto. Esto ha permitido un alto grado de personalización de las herramientas, adaptándolas de manera precisa a las necesidades específicas del proyecto. En la tabla 3.1 se muestra las características del sistema de monitoreo anterior respecto a la del actual.

Tabla 3.1-1 Comparativa cualitativa del sistema de monitoreo (Los autores, 2024)

Características	Antes	Después
Protocolos de comunicación monitoreados bajo licencias	Si	No
Uso de licencias por puntos de monitoreo	Si	No
Uso de licencias por puntos historizados	Si	No
Pantallas de monitoreo desarrolladas	Limitadas	Ilimitadas
Uso de Servidores Físico/Virtuales	Virtuales	Virtuales
Costos de licencias	Si	No
Costos por renovación de soporte de licencias	Si	No
Suscripciones anuales del sistema operativo correspondiente al servidor	Si	No
Redundancia sobre la plataforma de monitoreo	No	Si

3.2 Costos de Implementación Sistema de monitoreo.

- Los costos de implementación de un sistema de monitoreo en el mercado actual, considerando el uso de plataformas licenciadas, se encuentran por alrededor de los \$79.938,00, se adjunta Tabla 3.2.1 con el detalle de los costos que se cotizan normalmente por empresas particulares.

Tabla 3.2-1 Costos sistema de monitoreo en el mercado actual (Los autores, 2024)

	Descripción	Cantidad	Unidad	Precio Unitario	Precio Total
Recursos	Aprovisionamiento de servidores virtual (Entregado por Centro de Datos) Procesador: 4-core Memoria RAM: 16GB Almacenamiento: 500GB	1	Un	\$ -	\$ -
	Licencia Sistema Operativo Windows Server 2019	1	Un	\$ 800,00	\$ 800,00
	Licencias de software de monitoreo	1	Un	\$ 44.971,00	\$ 44.971,00
	Soporte técnico con vigencia de 1 año	1	Un	\$ 8.992,00	\$ 8.992,00
	Adquisición de pasarela conversión protocolo Modbus RTU a TCP	4	Un	\$ 1.500,00	\$ 6.000,00
Mano de Obra	Ingeniería y desarrollo: - Ingeniería básica - Instalación y configuración servidor de sistema de monitoreo - Integración y validación de datos Sistema Eléctrico - Integración y validación de datos Sistema Mecánico (Climatización) - Construcción dashboards operativos y gerencial	1	Un	\$ 19.175,00	\$ 19.175,00
TOTAL					\$ 79.938,00

Consideraciones de la oferta particular

- ✓ Tiempo de ejecución: 5 meses
- ✓ Servidores Virtuales: Se utilizarán recursos propios de la empresa
- ✓ Red de infraestructura de comunicación (Lazos de red Modbus, Switches, cableado estructurado): Se utiliza la infraestructura existente
- ✓ Integración de equipos del sistema eléctrico y mecánico: Todos los que posean puerto y/o tarjeta de comunicación operativa y funcional
- ✓ Servidor de base de datos: Se utilizar Base de Datos existente

- Al utilizar soluciones de código abierto, se eliminó la necesidad de adquirir licencias de software comerciales, reduciendo significativamente los costos operativos, entre otros detalles en la integración de las variables de monitoreo, adicional a esto se trabajó con sistema operativo Linux en los servidores virtuales aprovisionados por el centro de datos.

Tabla 3.2-2 Costos de la implementación del proyecto (Los autores, 2024)

	Descripción	Cantidad	Unidad	Precio Unitario	Precio Total
Recursos	Aprovisionamiento de servidores virtuales (Entregado por Centro de Datos) Procesador: 4-core Memoria RAM: 16GB Almacenamiento: 500GB	2	Un	\$ -	\$ -
	Licencia Sistema Operativo Linux	1	Un	\$ -	\$ -
	Adquisición de pasarela MOXA MGATE MB3480 Conversión protocolo Modbus RTU a TCP	4	Un	\$ 1.350,00	\$ 5.400,00
Mano de Obra	Ingeniería y desarrollo: - Ingeniería básica - Instalación y configuración servidor de sistema de monitoreo - Integración y validación de datos Sistema Eléctrico - Integración y validación de datos Sistema Mecánico (Climatización) - Construcción dashboards operativos y gerencial	1	Un	\$ 10.800,00	\$ 10.800,00
TOTAL					\$ 16.200,00

Consideraciones del proyecto

- ✓ Tiempo de ejecución: 3 meses
- ✓ Servidores Virtuales: Se utilizarán recursos propios de la empresa
- ✓ Red de infraestructura de comunicación (Lazos de red Modbus, Switches, cableado estructurado): Se utiliza la infraestructura existente
- ✓ Integración de equipos del sistema eléctrico y mecánico: Todos los que posean puerto y/o tarjeta de comunicación operativa y funcional
- ✓ Servidor de base de datos: Se utilizar Base de Datos existente

3.3 Análisis de los resultados.

Analizando los costos de implementación del sistema de monitoreo, validamos que los recursos, como licencias de uso de software propietario, mano de obra de ingeniería de desarrollo, hace que los costos se eleven en aproximadamente un 79,7% del total de la oferta.

Analizando más a detalle vemos que la diferencia en costos de mano de obra es del 43,6% y en recursos el 91,1%, con lo que se ratifica una vez más que los rubros de licencias marcan la diferencia.

Otro de los puntos a considerar, es el tiempo de implementación del proyecto, ya que de esto depende los costos de mano de obra al ser directamente proporcionales.

El aprovechamiento de los recursos propios del Centro de Datos como el aprovisionamiento de servidores virtuales y el uso de sistemas operativos Linux fue pilar fundamental en la toma de decisiones para dar inicio a la ejecución del proyecto.

CAPÍTULO 4

4. CONCLUSIONES Y RECOMENDACIONES

Conclusiones

El uso de plataformas de código abierto proporcionó una gran flexibilidad y libertad en el desarrollo del proyecto. Al evitar las restricciones de software propietario, se pudo personalizar y adaptar el sistema de monitoreo según las necesidades específicas y permitió un ahorro significativo en licencias y mantenimiento de software.

El uso de pasarelas virtuales implementadas con Node-Red y utilizando tecnologías de contenerización permitieron frente a pasarelas físicas ser más eficientes en cuanto a velocidad de procesamiento, portabilidad, escalabilidad y modularidad.

La implementación del presente proyecto de titulación permitió poder realizar la adquisición, procesamiento y presentación de datos de los equipos de forma local y remota indiferente del protocolo de comunicación que maneje.

El proyecto se completó conforme a las metas establecidas, logrando los resultados esperados en términos de funcionalidad, rendimiento y usabilidad. Se optimizaron los procesos gracias a la automatización y la integración de diversas herramientas, reduciendo tiempos de respuesta y mejorando la eficiencia general.

La infraestructura desarrollada ofrece la capacidad de crecer y adaptarse a futuras necesidades, permitiendo la integración de nuevas funciones o módulos sin grandes cambios. La infraestructura desarrollada ofrece la posibilidad de adaptarse a cambios y escalas de manera sencilla, facilitando la adición de nuevas fuentes de datos o funcionalidades.

Gracias a la integración de sistemas personalizados, se obtuvo un mayor control y visibilidad sobre los datos, mejorando la capacidad de análisis y toma de decisiones, además de una mejor comprensión de los datos, lo que impulsó decisiones más informadas y precisas.

Los usuarios finales del sistema han reportado una experiencia positiva, destacando la facilidad de uso, la rapidez y la flexibilidad de la solución.

Estos resultados demuestran que la implementación fue exitosa y que el proyecto cumplió con los requisitos establecidos, proporcionando beneficios tangibles para la organización, así como la efectividad y beneficios de optar por soluciones de código abierto para este tipo de desarrollo.

Recomendaciones

Dentro de las recomendaciones para que el proyecto se mantenga en vanguardia de los avances tecnológicos, es analizar la posibilidad de utilizar otras plataformas de código abierto que nos permita tener más opciones de servicios de contenerización, inclusive hacer más robusto el sistema de adquisición de datos.

En vista de que la implementación de servidores es virtual, se recomienda mantener respaldada la configuración de las mismas usando servicios de Acronis que ofrece la posibilidad de generar respaldos periódicos de la máquina virtual, de tal forma que permita volver a levantar los servidores inclusive en otras instancias, minimizando de esta forma los tiempos de restablecimiento del sistema de monitoreo.

Para obtener una mayor velocidad de adquisición de datos en un protocolo de comunicación de red serial como Modbus RTU, se recomienda la redistribución de los lazos, respetando las especificaciones técnicas del mismo.

Mantener actualizadas las versiones de firmware de los equipos de la infraestructura eléctrica y mecánica del centro de datos, garantiza minimizar riesgos de vulnerabilidades informáticas que puedan ser aprovechadas por piratas informáticos.

Asegurar que el equipo esté capacitado en el uso de contenedores y microservicios. La formación continua puede ayudar a mantener al equipo actualizado con las mejores prácticas y nuevas tecnologías.

Fomentar una cultura DevOps para mejorar la colaboración entre desarrollo y operaciones, promoviendo la automatización y la entrega continua.

Mantener una documentación clara y actualizada, incluyendo detalles sobre su configuración y requisitos de despliegue.

Al seguir estas recomendaciones, se pueden aprovechar al máximo los beneficios de los contenedores y los microservicios, logrando una arquitectura de software flexible, escalable y mantenible.

BIBLIOGRAFÍA

- Alcocer, J. J. (13 de 10 de 2020). *DATACENTERDYNAMICS*. Obtenido de <https://www.datacenterdynamics.com/es/opinion/monitoreo-de-centros-de-datos/>
- Asturias, D. (23 de 10 de 2023). *WEBSENTRA*. Obtenido de <https://www.websentra.com/grafana-beginners-guide/>
- CODIGOELECTRONICA*. (03 de 08 de 2019). Obtenido de <http://codigoelectronica.com/blog/que-es-node-red>
- CONTANDO BITS*. (2024). Obtenido de <https://www.contandobits.com/redes/descargar-cmdr/>
- DZONE*. (20 de 09 de 2019). Obtenido de <https://dzone.com/articles/all-about-hibernate-manytomany-association>
- EDUCARCHILE*. (2024). Obtenido de <https://www.educarchile.cl/herramientas-tic/inkscape-editor-de-graficos-vectoriales>
- EVCON*. (2024). Obtenido de <https://evcon-group.com/servicio/data-center/>
- Flores, F. (22 de 07 de 22). *OPENWEBINARS*. Obtenido de <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>
- Google. (s.f.). Obtenido de https://www.google.com/search?sca_esv=566165568&sxsrf=ADLYWILiheKao96SXdxdyFpuv9xuOZog1w:1725564685997&q=paneles+futuristas+inkscape&udm=2&fbs=AEQNm0CbCVgAZ5mWEJDg6aoPVcBgWizR0-0aFOH11Sb5tINhd3zC4y7ZXTSrvvSBSNjw8fVX3G3tS3bGsqQeBBxb6Hy8PHLF19NkiQG5U-EnIjnm
- IKONO*. (18 de 09 de 2017). Obtenido de <https://ikono.co/tecnologia/que-es-docker/>
- Lea, R. (05 de 04 de 2023). *NODEREDGUIDE*. Obtenido de <https://noderedguide.com/node-red-lecture-4-a-tour-of-the-core-nodes/>
- Los autores. (2024).
- Molina, O. V. (26 de 07 de 2019). *MEDIUM*. Obtenido de <https://medium.com/@naxus1/iot-82622d5e5f5d>
- OPENWEBINARS*. (04 de 09 de 2019). Obtenido de <https://openwebinars.net/blog/que-es-nodejs/>

PICKDATA. (11 de 2020). Obtenido de https://www.pickdata.net/es/noticias/node-red-programacion-visual-iot?gclid=Cj0KCQjwx5qoBhDyARIsAPbMagAKUmf_mXfj26v2hBnbodroHZ2Q-il6JwAvw8zFp4j_i4m1lvdKVukaAoIHEALw_wcB

REDHAT. (11 de 2022). Obtenido de <https://www.redhat.com/es/topics/open-source/what-is-open-source-software>

REDHAT. (13 de 05 de 2022). Obtenido de <https://www.redhat.com/es/topics/data-services/what-is-grafana#:~:text=Grafana%20es%20una%20plataforma%20interactiva,la%20interpretaci%C3%B3n%20y%20la%20comprensi%C3%B3n.>

VERITAS. (2024). Obtenido de <https://www.veritas.com/es/mx/information-center/containerization>

APÉNDICES

APÉNDICE A

Tabla 0-1 Protocolos de comunicación de los equipos y cantidad de variables (Los autores, 2024)

Ubicación	Activo	Controlador (Marca, Modelo)	Protocolo de Comunicación	Monitoreo / Control	Cantidad de Datos I/O (Comunicación y Cableados)	Cantidad de variables		Cantidad de equipos
						Sistema 1	Sistema 2	
Guayaquil	Controlador FX-80	Jhonsen Control	BacNet	Monitoreo	1	180	180	2
Guayaquil	Generadores	Himoinsa, modulo de comunicación DSE 8610	Modbus TCP/RTU	Monitoreo	2	156	159	6
Guayaquil	UPS 200KVA	General Electric	SNMP	Monitoreo	1	182	182	14
Guayaquil	UPS 10KVA	General Electric	SNMP	Monitoreo	1	44	44	4
Guayaquil	Rectificador	Eltel	SNMP	Monitoreo	1	10	10	2
Guayaquil	Manejadora de aire	Uniflair	SNMP/BacNet	Monitoreo	2	234	0	13
Guayaquil	ATS	GE Zenith Controls	Lazo Modbus RTU	Monitoreo	1	50	40	9
Guayaquil	Power Break II	General Electric	Lazo Modbus RTU	Monitoreo	1	182	140	23
Guayaquil	Breaker Spectra	General Electric	Lazo Modbus RTU	Monitoreo	1	795	555	90
Guayaquil	PDU	PDI	Lazo Modbus RTU	Monitoreo	1	1820	1820	20
Guayaquil	DSE-7420	DSE	Modbus TCP	Monitoreo	1	50	0	1
Guayaquil	EPM 2200	General Electric	Modbus RTU	Monitoreo	1	30	20	5
Guayaquil	ION 8650	Schneider	Modbus TCP	Monitoreo	1	31	0	1

Tabla 0-2 Variables de Generadores (Los autores, 2024)

Dirección Modbus	# de Registros	Función	Tipo de Dato	Valores	Escala	Variable	Unidades	Equivalencia	Descripción
772	1	3	INT (16 Bit)	0 - 65535	1	Control Mode	N/A	0 -> Stop Mode (48654) 1 -> Auto Mode (48657) 2 -> Manual Mode (48655) 3 -> Test on load mode 4 -> Auto with manual restore mode/Prohibit Return 5 -> User configuration mode 6 -> Test off load mode 7 -> Off Mode 8 - 65534 -> Reserved 65535 -> Unimplemented Bus Breaker LED (48659) Gen Available LED (48661)	Modo de Control
1024	1	3	INT (16 Bit)	0 - 10000	1	Oil Pressure	Kpa	N/A	Presión de Aceite
1025	1	3	INT (16 S Bit)	(-50) - 200	1	Coolant Temperature	°C	N/A	Temperatura de refrigerante
1027	1	3	INT (16 Bit)	0 - 130	1	Fuel Level	%	N/A	Nivel de Combustible
1028	1	3	INT (16 Bit)	0 - 40	0,1	Charge Alternator Voltage	V	N/A	Voltaje de carga del alternador
1029	1	3	INT (16 Bit)	0 - 40	0,1	Engine Batery Voltage	V	N/A	Voltaje de batería del motor
1030	1	3	INT (16 Bit)	0 - 6000	1	Engine Speed	RPM	N/A	Velocidad del motor
1031	1	3	INT (16 Bit)	0 - 70	0,1	Generator Frequency	Hz	N/A	Frecuencia del Generador
1032	1	3	INT (16 Bit) H	0 - 18000	0,1	Generator L1-N voltage	V	N/A	Voltaje Generador Linea 1 - Neutro
1033	1		INT (16 Bit) L						
1034	1	3	INT (16 Bit) H	0 - 18000	0,1	Generator L2-N voltage	V	N/A	Voltaje Generador Linea 2 - Neutro
1035	1		INT (16 Bit) L						
1036	1	3	INT (16 Bit) H	0 - 18000	0,1	Generator L3-N voltage	V	N/A	Voltaje Generador Linea 3 - Neutro
1037	1		INT (16 Bit) L						
1038	1	3	INT (16 Bit) H	0 - 30000	0,1	Generator L1-L2 voltage	V	N/A	Voltaje Generador Linea 1 2
1039	1		INT (16 Bit) L						
1040	1	3	INT (16 Bit) H	0 - 30000	0,1	Generator L2-L3 voltage	V	N/A	Voltaje Generador Linea 23
1041	1		INT (16 Bit) L						
1042	1	3	INT (16 Bit) H	0 - 30000	0,1	Generator L3-L1 voltage	V	N/A	Voltaje Generador Linea 31

1043	1		INT (16 Bit) L						
1044	1	3	INT (16 Bit) H	0 - 99999,9	0,1	Generator L1 current	A	N/A	Corriente Generador Linea 1
1045	1		INT (16 Bit) L						
1046	1	3	INT (16 Bit) H	0 - 99999,9	0,1	Generator L2 current	A	N/A	Corriente Generador Linea 2
1047	1		INT (16 Bit) L						
1048	1	3	INT (16 Bit) H	0 - 99999,9	0,1	Generator L3 current	A	N/A	Corriente Generador Linea 3
1049	1		INT (16 Bit) L						
1052	1	3	INT (16 Bit) H	(- 9999999	1	Generator L1 watts	W	N/A	Potencia Activa Generador L1
1053	1		INT (16 Bit) L	9) - 99999999					
1054	1	3	INT (16 Bit) H	(- 99999999	1	Generator L2 watts	W	N/A	Potencia Activa Generador L2
1055	1		INT (16 Bit) L	9) - 999999999					
1056	1	3	INT (16 Bit) H	(- 999999999	1	Generator L3 watts	W	N/A	Potencia Activa Generador L3
1057	1		INT (16 Bit) L	9) - 9999999999					
1536	1	3	INT (16 Bit) H	(- 9999999999	1	Generator total watts	W	N/A	Potencia Total Activa Generador
1537	1		INT (16 Bit) L	9) - 99999999999					
1538	1	3	INT (16 Bit) H	0 - 9999999999	1	Generator L1 VA	VA	N/A	Potencia Aparente Generador L1
1539	1		INT (16 Bit) L	9					
1540	1	3	INT (16 Bit) H	0 - 9999999999	1	Generator L2 VA	VA	N/A	Potencia Aparente Generador L2
1541	1		INT (16 Bit) L	9					
1542	1	3	INT (16 Bit) H	0 - 9999999999	1	Generator L3 VA	VA	N/A	Potencia Aparente Generador L3
1543	1		INT (16 Bit) L	9					
1544	1	3	INT (16 Bit) H	0 - 9999999999	1	Generator Total VA	VA	N/A	Potencia Aparente Total Generador
1545	1		INT (16 Bit) L	9					
1554	1	3	INT (16 S Bit)	(-1) - 1	0,01	Generator Power Factor L1	N/A	N/A	Factor de Potencia Generador L1
1555	1	3	INT (16 S Bit)	(-1) - 1	0,01	Generator Power Factor L2	N/A	N/A	Factor de Potencia Generador L2
1556	1	3	INT (16 S Bit)	(-1) - 1	0,01	Generator Power Factor L3	N/A	N/A	Factor de Potencia Generador L3

1557	1	3	INT (16 S Bit)	(-1) - 1	0,01	Generator Average Power Factor	N/A	N/A	Factor de Potencia Generador Promedio
1650	1	3	INT (16 Bit) H	0 - 18000	0,1	Generator L-N voltage average	V	N/A	Voltaje Generador Linea - Neutro Promedio
1651	1		INT (16 Bit) L						
1658	1	3	INT (16 Bit) H	0 - 30000	0,1	Generator L-L voltage average	V	N/A	Voltaje Generador Linea - Linea Promedio
1659	1		INT (16 Bit) L						
1666	1	3	INT (16 Bit) H	0 - 99999,9	0,1	Generator current average	A	N/A	Corriente Generador Promedio
1667	1		INT (16 Bit) L						
1672	1	3	INT (16 Bit) H	0 - 99999,9	0,1	Generator current max	A	N/A	Corriente Generador Máxima
1673	1		INT (16 Bit) L						
1674	1	3	INT (16 S Bit)	(-999,9) - 999,9	0,1	Generator watts average	%	N/A	Porcentaje Potencia Activa Generador Promedio
1677	1	3	INT (16 S Bit)	(-999,9) - 999,9	0,1	Generator watts max	%	N/A	Porcentaje Potencia Activa Generador Máxima
1678	1	3	INT (16 S Bit)	0 - 999,9	0,1	Generator VA average	%	N/A	Porcentaje Potencia Aparente Generador Promedio
1681	1	3	INT (16 S Bit)	0 - 999,9	0,1	Generator VA max	%	N/A	Porcentaje Potencia Aparente Generador Máxima
1682	1	3	INT (16 S Bit)	(-999,9) - 999,9	0,1	Generator Var average	%	N/A	Porcentaje Potencia Reactiva Generador Promedio
1686	1	3	INT (16 S Bit)	(-999,9) - 999,9	0,1	Generator Power Factor Average	%	N/A	Porcentaje Factor de Potencia Generador Promedio
1798	1	3	INT (16 Bit) H	0 - 4.29 x10^9	1	Engine Run Time	Seconds	N/A	Tiempo de operación del generador
1799	1		INT (16 Bit) L						
1800	1	3	INT (16 Bit) H	0 - 4.29 x10^9	0,1	Generator Positive KW Hours	KW Hour	N/A	Energía Activa
1801	1		INT (16 Bit) L						
1804	1	3	INT (16 Bit) H	0 - 4.29 x10^9	0,1	Generator KVA Hours	KVA Hour	N/A	Energía Aparente
1805	1		INT (16 Bit) L						
1806	1	3	INT (16 Bit) H	0 - 4.29 x10^9	0,1	Generator KVAr Hours	KVAr Hour	N/A	Energía Reactiva
1807	1		INT (16 Bit) L						
1808	1	3	INT (16 Bit) H	0 - 99999	1	Number Of Starts	N/A	N/A	Número de arranque
1809	1		INT (16 Bit) L						
39425	1	3	INT (13/16 - 16/16 Bit)	0 - 15	1	Emergency stop	N/A	0 -> Disabled digital input 1 -> Not active alarm 2 -> Warning alarm	Paro de emergencia

			INT (9/16 - 12/16 Bit)	0 - 15	1	Low oil pressure	N/A	3 -> Shutdown alarm 4 -> Electrical trip Alarm 5 - 7 -> Reserved 8 -> Inactive indication (no string) 9 -> Inactive indication (displayed string) 10 -> Active indication 11 - 14 -> Reserved 15 -> Unimplemented alarm	Baja presión de aceite
			INT (5/16 - 8/16 Bit)	0 - 15	1	High coolant temperature	N/A		Alta temperatura del refrigerante
			INT (1/16 - 4/16 Bit)	0 - 15	1	High oil temperature	N/A		
39426	1	3	INT (13/16 - 16/16 Bit)	0 - 15	1	Under Speed	N/A		Baja velocidad del motor
			INT (9/16 - 12/16 Bit)	0 - 15	1	Over Speed	N/A		Alta velocidad del motor
			INT (5/16 - 8/16 Bit)	0 - 15	1	Generator Under frequency	N/A		Baja frecuencia de generación
			INT (1/16 - 4/16 Bit)	0 - 15	1	Generator Over frequency	N/A		Alta frecuencia de generación
39427	1	3	INT (13/16 - 16/16 Bit)	0 - 15	1	Generator low voltage	N/A		Bajo voltaje de generación
			INT (9/16 - 12/16 Bit)	0 - 15	1	Generator high voltage	N/A		Alto voltaje de generación
			INT (5/16 - 8/16 Bit)	0 - 15	1	Battery low voltage	N/A		Bajo voltaje de batería
			INT (1/16 - 4/16 Bit)	0 - 15	1	Battery high voltage	N/A		Alto voltaje de batería
39428	1	3	INT (13/16 - 16/16 Bit)	0 - 15	1	Charge alternator failure	N/A		Falla de carga del alternador
			INT (9/16 - 12/16 Bit)	0 - 15	1	Fail to start	N/A		Falla en el arranque del generador
			INT (5/16 - 8/16 Bit)	0 - 15	1	Fail to stop	N/A		Falla en la parada del generador
			INT (1/16 - 4/16 Bit)	0 - 15	1	Generator breaker fail to close	N/A		El disyuntor del generador no se cerró
39429	1	3	INT (13/16 - 16/16 Bit)	0 - 15	1	Mains fail to close	N/A		
			INT (9/16 - 12/16 Bit)	0 - 15	1	Oil pressure sender fault	N/A	Fallo en el transmisor de presión de aceite	

			INT (5/16 - 8/16 Bit)	0 - 15	1	Loss of magnetic pick up	N/A		Perdida de captación magnética RPM
			INT (1/16 - 4/16 Bit)	0 - 15	1	Magnetic pick up open circuit	N/A		Circuito abierto de captación magnética RPM
39430	1	3	INT (13/16 - 16/16 Bit)	0 - 15	1	Generator high current	N/A		Corriente alta de generación
			INT (9/16 - 12/16 Bit)	0 - 15	1	Calibration lost	N/A		Perdida de calibración
			INT (5/16 - 8/16 Bit)	0 - 15	1	Low fuel level	N/A		Bajo nivel de combustible
			INT (1/16 - 4/16 Bit)	0 - 15	1	CAN ECU Warning	N/A		Advertencia por comunicación CAN con ECU (ADEC)
39431	1	3	INT (13/16 - 16/16 Bit)	0 - 15	1	CAN ECU Shutdown	N/A		Apagado por comunicación CAN con ECU (ADEC)
			INT (9/16 - 12/16 Bit)	0 - 15	1	CAN ECU Data fail	N/A		Falla de comunicación CAN con ECU (ADEC)
			INT (5/16 - 8/16 Bit)	0 - 15	1	Low oil level switch	N/A		Interruptor encendido por bajo nivel de aceite
			INT (1/16 - 4/16 Bit)	0 - 15	1	High temperature switch	N/A		Interruptor encendido por alta temperatura
39432	1	3	INT (13/16 - 16/16 Bit)	0 - 15	1	Low fuel level switch	N/A		Interruptor encendido por bajo nivel de combustible
			INT (9/16 - 12/16 Bit)	0 - 15	1	Expansion unit watchdog alarm	N/A		Alarma de vigilancia de la unidad de expansión
			INT (5/16 - 8/16 Bit)	0 - 15	1	kW overload alarm	N/A		Alarma de sobrecarga del generador
			INT (1/16 - 4/16 Bit)	0 - 15	1	Negative phase sequence current alarm	N/A		Alarma de corriente de secuencia de fase negativa
39433	1	3	INT (13/16 - 16/16 Bit)	0 - 15	1	Earth fault trip alarm	N/A		Alarma de disparo por fallo a tierra
			INT (9/16 - 12/16 Bit)	0 - 15	1	Generator phase rotation alarm	N/A		Alarma de rotación de fase del generador
			INT (5/16 - 8/16 Bit)	0 - 15	1	Auto Voltage Sense Fail	N/A		Falla en la detección automática de voltaje

			INT (1/16 - 4/16 Bit)	0 - 15	1	Maintenance alarm	N/A		Alarma de mantenimiento
39434	1	3	INT (13/16 - 16/16 Bit)	0 - 15	1	Loading frequency alarm	N/A		Alarma de frecuencia de carga
			INT (9/16 - 12/16 Bit)	0 - 15	1	Loading voltage alarm	N/A		Alarma de voltaje de carga
			INT (5/16 - 8/16 Bit)	0 - 15	1	Fuel usage running	N/A		Uso de combustible en funcionamiento
			INT (1/16 - 4/16 Bit)	0 - 15	1	Fuel usage stopped	N/A		Se detuvo el uso de combustible
39435	1	3	INT (13/16 - 16/16 Bit)	0 - 15	1	Protections disabled	N/A		Protecciones deshabilitadas
			INT (9/16 - 12/16 Bit)	0 - 15	1	Protections blocked	N/A		Protecciones bloqueadas
			INT (5/16 - 8/16 Bit)	0 - 15	1	Generator breaker failed to open	N/A		El disyuntor del generador no se abrió
			INT (1/16 - 4/16 Bit)	0 - 15	1	Mains breaker failed to open	N/A		
39436	1	3	INT (13/16 - 16/16 Bit)	0 - 15	1	Bus breaker failed to close	N/A		
			INT (9/16 - 12/16 Bit)	0 - 15	1	Bus breaker failed to open	N/A		
			INT (5/16 - 8/16 Bit)	0 - 15	1	Generator reverse power alarm	N/A		Alarma de potencia inversa del generador
			INT (1/16 - 4/16 Bit)	0 - 15	1	Short circuit alarm	N/A		Alarma de cortocircuito
39437	1	3	INT (13/16 - 16/16 Bit)	0 - 15	1	Air flap closed alarm	N/A		Alarma de trampilla de aire cerrada
			INT (9/16 - 12/16 Bit)	0 - 15	1	Failure to sync	N/A		Error al sincronizar
			INT (5/16 - 8/16 Bit)	0 - 15	1	Bus live	N/A		
			INT (1/16 - 4/16 Bit)	0 - 15	1	Bus not live	N/A		Barra no viva

39438	1	3	INT (13/16 - 16/16 Bit)	0 - 15	1	Bus phase rotation	N/A	Rotación de fase de barra
			INT (9/16 - 12/16 Bit)	0 - 15	1	Priority selection error	N/A	Error de selección de prioridad
			INT (5/16 - 8/16 Bit)	0 - 15	1	MSC data error	N/A	Error de datos MSC
			INT (1/16 - 4/16 Bit)	0 - 15	1	MSC ID error	N/A	Error de ID MSC
39439	1	3	INT (13/16 - 16/16 Bit)	0 - 15	1	Bus low voltage	N/A	Bajo voltaje de barra
			INT (9/16 - 12/16 Bit)	0 - 15	1	Bus high voltage	N/A	Alto voltaje de barra
			INT (5/16 - 8/16 Bit)	0 - 15	1	Bus low frequency	N/A	Baja frecuencia de barra
			INT (1/16 - 4/16 Bit)	0 - 15	1	Bus high frequency	N/A	Alta frecuencia de barra
39440	1	3	INT (13/16 - 16/16 Bit)	0 - 15	1	MSC failure	N/A	Falla MSC
			INT (9/16 - 12/16 Bit)	0 - 15	1	MSC too few sets	N/A	Muy pocos equipos MSC
			INT (5/16 - 8/16 Bit)	0 - 15	1	MSC alarms inhibited	N/A	Alarma inhibida MSC
			INT (1/16 - 4/16 Bit)	0 - 15	1	MSC old version units on the bus	N/A	
39441	1	3	INT (13/16 - 16/16 Bit)	0 - 15	1	Mains reverse power alarm/mains export alarm	N/A	
			INT (9/16 - 12/16 Bit)	0 - 15	1	Minimum sets not reached	N/A	Configuración mínima no alcanzada
			INT (5/16 - 8/16 Bit)	0 - 15	1	Insufficient capacity	N/A	Capacidad insuficiente
			INT (1/16 - 4/16 Bit)	0 - 15	1	Out of sync	N/A	Fuera de sincronismo
39452	1	3	INT (13/16 - 16/16 Bit)	0 - 15	1	High fuel level	N/A	Alto nivel de combustible

			INT (9/16 - 12/16 Bit)	0 - 15	1	Low kW Load (Wet Stacking)	N/A		Alarma de baja carga del generador
			INT (5/16 - 8/16 Bit)	0 - 15	1	Module Communication Fail (8661)	N/A		
			INT (1/16 - 4/16 Bit)	0 - 15	1	Bus Module Warning (8661)	N/A		
48640	1	3	INT (16 Bit)	0 - 1	1	Fuel Relay DO_A	N/A	N/A	Salida digital de relé de paso de combustible
48641	1	3	INT (16 Bit)	0 - 1	1	Start Relay DO_B	N/A	N/A	Salida digital de relé de arranque
48642	1	3	INT (16 Bit)	0 - 1	1	Audible Alarm DO_E	N/A	N/A	Salida Digital de Alarma Audible en generador
48643	1	3	INT (16 Bit)	0 - 1	1	Fuel Pump Control DO_F	N/A	N/A	Salida digital de rele control bomba combustible
48652	1	3	INT (16 Bit)	0 - 1	1	Close Gen Output DO_D	N/A	N/A	Salida digital de relé de cierre de breaker de salida del generador
48654	1	3	INT (16 Bit)	0 - 1	1	Stop Mode LED	N/A	N/A	Led Modo Stop
48655	1	3	INT (16 Bit)	0 - 1	1	Manual Mode LED	N/A	N/A	Led Modo Manual
48657	1	3	INT (16 Bit)	0 - 1	1	Auto Mode LED	N/A	N/A	Led Modo Auto
48659	1	3	INT (16 Bit)	0 - 1	1	Bus Breaker LED	N/A	N/A	Led breaker de barra cerrado
48661	1	3	INT (16 Bit)	0 - 1	1	Gen Available LED	N/A	N/A	Led de generador disponible

Tabla 0-3 Variables de UPS (Los autores, 2024)

Equipo	Registros	Fuccion	Tipo de Dato	Valores	Escala	Variable	Unidades	Equivalencia	Descripción	Descripción Item	Template Zbx
UPS-1-1A	N/A	N/A	N/A	N/A	1	Battery Status	N/A	N/A	Estado de la batería	Battery Status	UPS-1-1A GE
	N/A	N/A	N/A	N/A	1	Seconds On Battery	S	N/A	Segundos con batería	Time on Battery	UPS-1-1A GE
	N/A	N/A	N/A	N/A	1	Estimated Minutes Remaining	S	N/A	Minutos restantes estimados	Estimated minutes Remaining	UPS-1-1A GE
	N/A	N/A	N/A	N/A	1	Estimated Charge Remaining	A	N/A	Carga restante estimado	Estimated Charge Remaining	UPS-1-1A GE
	N/A	N/A	N/A	N/A	0,1	Battery Voltage	V	N/A	Voltaje de la batería	Battery Voltage	UPS-1-1A GE
	N/A	N/A	N/A	N/A	0,1	Input Voltage	V	N/A	Voltaje de entrada línea 1	Input Voltage L1	UPS-1-1A GE
	N/A	N/A	N/A	N/A	0,1	Input Voltage 2	V	N/A	Voltaje de entrada línea 2	Input Voltage L2	UPS-1-1A GE
	N/A	N/A	N/A	N/A	0,1	Input Voltage 3	V	N/A	Voltaje de entrada línea 3	Input Voltage L3	UPS-1-1A GE
	N/A	N/A	N/A	N/A	0,1	Input Voltage Min	V	N/A	Voltaje de entrada Min	Input Voltage Min L1	UPS-1-1A GE
	N/A	N/A	N/A	N/A	0,1	Input Voltage Max	V	N/A	Voltaje de entrada máx.	Input Voltage Max	UPS-1-1A GE
	N/A	N/A	N/A	N/A	0,1	Output Voltage	V	N/A	Voltaje de salida línea 1	Output Voltage L1	UPS-1-1A GE
	N/A	N/A	N/A	N/A	0,1	Output Voltage 2	V	N/A	Voltaje de salida línea 2	Output Voltage L2	UPS-1-1A GE
	N/A	N/A	N/A	N/A	0,1	Output Voltage 3	V	N/A	Voltaje de salida línea 3	Output Voltage L3	UPS-1-1A GE
	N/A	N/A	N/A	N/A	0,1	Output Current	A	N/A	Corriente de salida línea 1	Output Current L1	UPS-1-1A GE
	N/A	N/A	N/A	N/A	0,1	Output Current 2	A	N/A	Corriente de salida línea 2	Output Current L2	UPS-1-1A GE
	N/A	N/A	N/A	N/A	0,1	Output Current 3	A	N/A	Corriente de salida línea 3	Output Current L3	UPS-1-1A GE
	N/A	N/A	N/A	N/A	0,1	Output Power	W	N/A	Potencia de salida línea 1	Output Power	UPS-1-1A GE
	N/A	N/A	N/A	N/A	0,1	Output Power 2	W	N/A	Potencia de salida línea 2	Output Power2	UPS-1-1A GE
	N/A	N/A	N/A	N/A	0,1	Output Power 3	W	N/A	Potencia de salida línea 3	Output Power3	UPS-1-1A GE
	N/A	N/A	N/A	N/A	0,1	Output Percent Load	%	N/A	Porcentaje de carga de salida línea 1	Output Percent Load L1	UPS-1-1A GE
	N/A	N/A	N/A	N/A	0,1	Output Percent Load 2	%	N/A	Porcentaje de carga de salida línea 2	Output Percent Load L2	UPS-1-1A GE
	N/A	N/A	N/A	N/A	0,1	Output Percent Load 3	%	N/A	Porcentaje de carga de salida línea 3	Output Percent Load L3	UPS-1-1A GE
	N/A	N/A	N/A	N/A	1	Alarms Present	N/A	N/A	Alarmas presentes	Active Alarms Present	UPS-1-1A GE
	N/A	N/A	N/A	N/A	1	Inverter On Off	N/A	N/A	Estado de inversor	Inverter On Off	UPS-1-1A GE
N/A	N/A	N/A	N/A	1	Bypass On Off	N/A	N/A	Estado de Bypass	Bypass Status	UPS-1-1A GE	
N/A	N/A	N/A	N/A	1	Rectifier On Off	N/A	N/A	Estado de rectificador	Rectifier Status	UPS-1-1A GE	

Tabla 0-4 Variables de manejadoras de aire (Los autores, 2024)

Registros	Función	Tipo de Dato	Valores	Escala	Variable	Unidades	Equivalencia	Descripción	Descripción Item	Template Zbx
N/A	N/A	N/A	0-1	1	SYSTEM ON	N/A	0-Off 1-On	Estado ON OFF	System Status	Airco pCOWeb
N/A	N/A	N/A	0-1	1	ROOM TEMP AND HUMIDITY LIMITS	N/A	0-Off 1-On	Alarma de temp retorno		
N/A	N/A	N/A	0-1	1	CLOGGED FILTER ALARM	N/A	0-Off 1-On	Alarma de filtro		
N/A	N/A	N/A	0-1	1	FLOODING_ALARM	N/A	0-Off 1-On	Alarma de inundacion		
N/A	N/A	N/A	0-1	1	LOSS OF AIR FLOW ALARM	N/A	0-Off 1-On	Alarma perdida de flujo de aire		
N/A	N/A	N/A	0-1	1	HEATER OVERHEATING ALARM	N/A	0-Off 1-On	Alarma sobrecalentamiento		
N/A	N/A	N/A	0-1	1	SMOKE-FIRE ALARM	N/A	0-Off 1-On	Alarma de Humo		
N/A	N/A	N/A	0-1	1	HUMIDIFIER GENERAL ALARM	N/A	0-Off 1-On	Alarma de Humidificador		
N/A	N/A	N/A	0-1	1	UNIT ON ALARM	N/A	0-Off 1-On	Alarma general		
N/A	N/A	N/A	0-50	1	ROOM TEMPERATURE	°C	N/A	Temperatura retorno	Room Temperature (incoming air flow)	Airco pCOWeb
N/A	N/A	N/A	0-50	0,1	DELIVERY AIR TEMPERATURE	°C	N/A	Temperatura suministro	Supply Temperature (outgoing air flow)	Airco pCOWeb
N/A	N/A	N/A	0-100	0,1	ROOM RELATIVE HUMIDITY	rH%	N/A	humedad relativa	Room Humidity	Airco pCOWeb
N/A	N/A	N/A	0-100	0,1	COLD WATER VALVE RAMP	%	N/A	Rampa de valvula de agua		
N/A	N/A	N/A	0-100	0,1	EVAPORATING FAN SPEED	%	N/A	Velocidad de ventiladores	Supply Fan	Airco pCOWeb
N/A	N/A	N/A	0-50	0,1	COOLING SETPOINT RETURN AIR TEMP	°C	N/A	Setpoint	Setpoint Cooling Temperature	Airco pCOWeb
N/A	N/A	N/A	0-50	0,1	COOLING SENSITIVITY RETURN AIR TEMP	°C	N/A	Sensibilidad de enfriamiento		
N/A	N/A	N/A	0-10000	0,1	UNIT RUN HOURS	h	N/A	Horas trabajadas		
N/A	N/A	N/A	0-10000	0,1	HUMIDIFIER RUN HOURS	h	N/A	Horas trabajadas humidificador		

Tabla 0-5 Variables de Rectificadores (Los autores, 2024)

Registros	Funcion	Tipo de Dato	Valores	Escala	Variable	Unidades	Equivalencia	Descripción
N/A	N/A	N/A	0-1	1	POW_SYS_STAT	N/A		Estado ON OFF
N/A	N/A	N/A	0-1	1	MAIN_VOL_VALU	N/A		Alarma de temp retorno
N/A	N/A	N/A	0-1	1	REC_CUR_STAT	N/A		Alarma de filtro
N/A	N/A	N/A	0-1	1	REC_CUR_ENAB	N/A		Alarma de inundacion
N/A	N/A	N/A	0-1	1	REC_CUR_VALU	N/A		Alarma perdida de flujo de aire
N/A	N/A	N/A	0-1	1	REC_INP_VALU	N/A		Alarma sobrecalentamiento
N/A	N/A	N/A	0-1	1	BAT_VOL_STAT	N/A		Alarma de Humo
N/A	N/A	N/A	0-1	1	REC_TEMP_VALU	N/A		Alarma de Humidificador
N/A	N/A	N/A	0-1	1	BATT_VOL_VALU	N/A		Alarma general

Tabla 0-6 Variables de Medidor Ecoluz (Los autores, 2024)

Dirección Modbus	# de Registros	Función	Tipo de Dato	Valores	Escala	Variable	Unidades	Equivalencia	Descripción
18	1	3	UINT (16 Bit)		0,1	I a scaled	Amps	N/A	Corriente Media Tensión Linea A
19	1	3	UINT (16 Bit)		0,1	I b scaled	Amps	N/A	Corriente Media Tensión Linea B
20	1	3	UINT (16 Bit)		0,1	I c scaled	Amps	N/A	Corriente Media Tensión Linea C
21	1	3	UINT (16 Bit)		0,1	I avg scaled	Amps	N/A	Corriente Media Tensión Promedio
22	1	3	UINT (16 Bit)		0,1	V unbal	Volts	N/A	Voltaje de desbalance
23	1	3	UINT (16 Bit)		0,1	I unbal	Amps	N/A	Corriente de disbalance
24	1	3	UINT (16 Bit)		0,1	Freq	Freq	N/A	Frecuencia
27	1	3	UINT (16 Bit)		0,1	kW a scaled	Power	N/A	Potencia Activa Linea A
29	1	3	UINT (16 Bit)		0,1	kW b scaled	Power	N/A	Potencia Activa Linea B
31	1	3	UINT (16 Bit)		0,1	kW c scaled	Power	N/A	Potencia Activa Linea C
33	1	3	UINT (16 Bit)		0,1	kW tot scaled	Power	N/A	Potencia Activa Total
35	1	3	UINT (16 Bit)		0,1	kVAR a scaled	Power	N/A	Potencia Reactiva Linea A
37	1	3	UINT (16 Bit)		0,1	kVAR b scaled	Power	N/A	Potencia Reactiva Linea B
39	1	3	UINT (16 Bit)		0,1	kVAR c scaled	Power	N/A	Potencia Reactiva Linea C
41	1	3	UINT (16 Bit)		0,1	kVAR tot scaled	Power	N/A	Potencia Reactiva Total
43	1	3	UINT (16 Bit)		0,1	kVA a scaled	Power	N/A	Potencia Aparente Linea A
45	1	3	UINT (16 Bit)		0,1	kVA b scaled	Power	N/A	Potencia Aparente Linea B
47	1	3	UINT (16 Bit)		0,1	kVA c scaled	Power	N/A	Potencia Aparente Linea C
49	1	3	UINT (16 Bit)		0,1	kVA tot scaled	Power	N/A	Potencia Aparente Total
166	1	3	UINT (16 Bit)		1	VIn a scaled	Volts	N/A	Voltaje Linea A - Neutro

168	1	3	UINT (16 Bit)		1	VIn b scaled	Volts	N/A	Voltaje Linea B - Neutro
170	1	3	UINT (16 Bit)		1	VIn c scaled	Volts	N/A	Voltaje Linea C - Neutro
172	1	3	UINT (16 Bit)		1	VIn avg scaled	Volts	N/A	Voltaje Linea - Neutro Promedio
174	1	3	UINT (16 Bit)		1	VII ab scaled	Volts	N/A	Voltaje Linea AB
176	1	3	UINT (16 Bit)		1	VII bc scaled	Volts	N/A	Voltaje Linea BC
178	1	3	UINT (16 Bit)		1	VII ca scaled	Volts	N/A	Voltaje Linea CA
180	1	3	UINT (16 Bit)		1	VII avg scaled	Volts	N/A	Voltaje Linea - Linea Promedio
229	1	3	INT (16 Bit) H		1	kWh del	Volts	N/A	Energia Activa Entregada
230	1		INT (16 Bit) L						
231	1	3	INT (16 Bit) H		1	kWh rec	Volts	N/A	Energia Activa Recivida
232	1		INT (16 Bit) L						

Tabla 0-7 Variables de PDU (Los autores, 2024)

ID	Dirección	Registros	Funcion	Tipo de Dato	Valores	Escala	Variable	Unidades	Equivalencia	Descripción
0	0	1	3	INT (13/16 Bit)	0 - 1	N/A	Summary Alarm	N/A	0 = Normal, 1 = Alarm	Alarma General
1	0	1	3	INT (16 Bit)	0 - 65535	0.1	Input 1 AB Voltage	Volt	N/A	Voltaje entrada AB
	1	1	3	INT (16 Bit)	0 - 65535	0.1	Input 1 BC Voltage	Volt	N/A	Voltaje entrada BC
	2	1	3	INT (16 Bit)	0 - 65535	0.1	Input 1 CA Voltage	Volt	N/A	Voltaje entrada CA
	3	1	3	INT (16 Bit)	0 - 65535	0.1	Input 1 Frequency	Hz	N/A	Frecuencia entrada
	15	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 AB Voltage	Volt	N/A	Voltaje salida AB
	16	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 BC Voltage	Volt	N/A	Voltaje salida BC
	17	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 CA Voltage	Volt	N/A	Voltaje salida CA
	18	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 AN Voltage	Volt	N/A	Voltaje salida AN
	19	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 BN Voltage	Volt	N/A	Voltaje salida BN
	20	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 CN Voltage	Volt	N/A	Voltaje salida CN
	21	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 A Current	Amp	N/A	Corriente salida A
	22	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 B Current	Amp	N/A	Corriente salida B
	23	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 C Current	Amp	N/A	Corriente salida C
	24	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 A KW	KW	N/A	Potencia salida activa A
	25	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 B KW	KW	N/A	Potencia salida activa B
	26	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 C KW	KW	N/A	Potencia salida activa C
	27	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 Total KW	KW	N/A	Potencia salida activa Total

28	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 A KVA	KVA	N/A	Potencia salida aparente A
29	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 B KVA	KVA	N/A	Potencia salida aparente B
30	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 C KVA	KVA	N/A	Potencia salida aparente C
31	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 Total KVA	KVA	N/A	Potencia salida aparente Total
32	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 A KVAR	KVAR	N/A	Potencia salida reactiva A
33	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 B KVAR	KVAR	N/A	Potencia salida reactiva B
34	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 C KVAR	KVAR	N/A	Potencia salida reactiva C
35	1	3	INT (16 Bit)	0 - 65535	0.1	Output 1 Total KVAR	kVAR	N/A	Potencia salida reactiva Total
36	1	3	INT (16 Bit)	0 - 65535	0.01	Output 1 A PF	N/A	N/A	Factor de potencia Linea A
37	1	3	INT (16 Bit)	0 - 65535	0.01	Output 1 B PF	N/A	N/A	Factor de potencia Linea B
38	1	3	INT (16 Bit)	0 - 65535	0.01	Output 1 C PF	N/A	N/A	Factor de potencia Linea C
39	1	3	INT (16 Bit)	0 - 65535	1	Output 1 KWH	KWH	N/A	Energía Total
40	1		INT (16 Bit)	0 - 65535					
41	1	3	INT (16 Bit)	0 - 65535	0.1	Neutral Current	Amp	N/A	Corriente Neutro
42	1	3	INT (16 Bit)	0 - 65535	0.1	Ground Current	Amp	N/A	Corriente Tierra
70	1	3	INT (1/16 Bit)	0 - 1	N/A	Voltage Alarm	N/A	0 = Normal, 1 = Alarm	Alarma de Voltaje
			INT (2/16 Bit)	0 - 1	N/A	Current Alarm	N/A	0 = Normal, 1 = Alarm	Alarma de Corriente
			INT (3/16 Bit)	0 - 1	N/A	Power Alarm	N/A	0 = Normal, 1 = Alarm	Alarma de Potencia
			INT (4/16 Bit)	0 - 1	N/A	Building Alarm	N/A	0 = Normal, 1 = Alarm	Alarma Construcción
			INT (5/16 Bit)	0 - 1	N/A	Digital Alarm	N/A	0 = Normal, 1 = Alarm	Alarma Digital
			INT (6/16 Bit)	0 - 1	N/A	EPO	N/A	0 = Normal, 1 = Alarm	Alarma EPO
			INT (7/16 Bit)	0 - 1	N/A	Remote EPO	N/A	0 = Normal, 1 = Alarm	Alarma EPO remoto
			INT (8/16 Bit)	0 - 1	N/A	Thermal Overtemp	N/A	0 = Normal, 1 = Alarm	Alarma Sobretemperatura
			INT (9/16 Bit)	0 - 1	N/A	Thermal Hightemp	N/A	0 = Normal, 1 = Alarm	Alarma Alta temperatura
			INT (10/16 Bit)	0 - 1	N/A	Input Phase Rotation	N/A	0 = Normal, 1 = Alarm	Alarma de Rotación de Fase de Entrada
			INT (12/16 Bit)	0 - 1	N/A	Ground Current Trip	N/A	0 = Normal, 1 = Alarm	Disparo corriente a tierra
			INT (13/16 Bit)	0 - 1	N/A	Phase Rotation Trip	N/A	0 = Normal, 1 = Alarm	Disparo rotación de fase
			INT (14/16 Bit)	0 - 1	N/A	Over Voltage Trip	N/A	0 = Normal, 1 = Alarm	Disparo Sobrevoltaje
INT (15/16 Bit)	0 - 1	N/A	Under Voltage Trip	N/A	0 = Normal, 1 = Alarm	Disparo Bajo voltaje			

Tabla 0-8 Variables de ATS (Los autores, 2024)

Dirección	# de Registros	Función	Tipo de Dato	Valores	Escala	Variable	Unidades	Equivalencia	Descripción
0	1	3	INT (1/8 Bit)	0 - 1	N/A	Automatic Transfer Relay	N/A	1 = On, 0 = Off	Estatus de Transferencia Automática
			INT (2/8 Bit)	0 - 1	N/A	ATS Not in Auto Mode	N/A	1 = Not in Auto	Estatus de modo auto
			INT (3/8 Bit)	0 - 1	N/A	General Propose Alarm	N/A	0 = Normal, 1 = Alarm	Alarma de proposito general
			INT (7/8 Bit)	0 - 1	N/A	S2 Available	N/A	1 = S2 Available	Fuente 2 disponible
			INT (8/8 Bit)	0 - 1	N/A	S1 Available	N/A	1 = S1 Available	Fuente 1 disponible
3	1	3	INT (10/16 Bit)	0 - 1	N/A	S1 Position Status	N/A	1 = S1 Position	Status de posición fuente 1
			INT (11/16 Bit)	0 - 1	N/A	S2 Position Status	N/A	1 = S2 Position	Status de posición fuente 2
4	1	3	INT (16 Bit)	0 - 65535	1	Timer Countdown Value	N/A	N/A	Valor de tiempo cuenta regresiva
5	1	3	INT (16 Bit)	0 - 65535	1	S1 Voltage (Phase A-B)	Volt	N/A	Voltaje Fuente 1 Linea AB
6	1	3	INT (16 Bit)	0 - 65535	1	S1 Voltage (Phase B-C)	Volt	N/A	Voltaje Fuente 1 Linea BC
7	1	3	INT (16 Bit)	0 - 65535	1	S1 Voltage (Phase C-A)	Volt	N/A	Voltaje Fuente 1 Linea CA
8	1	3	INT (16 Bit)	0 - 65535	1	S2 Voltage (Phase A-B)	Volt	N/A	Voltaje Fuente 2 Linea AB
9	1	3	INT (16 Bit)	0 - 65535	1	S2 Voltage (Phase B-C)	Volt	N/A	Voltaje Fuente 2 Linea BC
10	1	3	INT (16 Bit)	0 - 65535	1	S2 Voltage (Phase C-A)	Volt	N/A	Voltaje Fuente 2 Linea CA
12	1	3	INT (16 Bit)	0 - 65535	1	S1 Frequency	Hz	N/A	Frecuencia Fuente 1
14	1	3	INT (16 Bit)	0 - 65535	1	S2 Frequency	Hz	N/A	Frecuencia Fuente 2
16	1	3	INT (16 Bit)	0 - 65535	1	Total Transfers to S1	N/A	N/A	Transferencias hacia fuente 1
48	1	3	INT (16 Bit)	0 - 65535	1	Total S1 Fails	N/A	N/A	Total de fallas de fuente 1
49	1	3	INT (16 Bit)	0 - 65535	1	Total Transfers to S2	N/A	N/A	Transferencias hacia fuente 2

Tabla 0-9 Variables de Breakers-Spectra (Los autores, 2024)

Dirección	Registros	Función	Tipo de Dato	Valores	Escala	Variable	Unidades	Equivalencia	Descripción
27	1	2	BOOL	0 - 1	N/A	Breaker position	N/A	0 - Open 1 - Closed	Posición de Breaker
51	1	2	BOOL	0 - 1	N/A	MET Tripped	N/A	0 - No 1 - Yes	Disparo de Breaker
56	1	2	BOOL	0 - 1	N/A	Voltage Unbalance Alarm Status	N/A	0 - Normal 1 - Alarm	Alarma Desbalance de Voltaje
57	1	2	BOOL	0 - 1	N/A	Under Voltage Alarm Status	N/A	0 - Normal 1 - Alarm	Alarma de Bajo Voltaje
58	1	2	BOOL	0 - 1	N/A	Over Voltage Alarm Status	N/A	0 - Normal 1 - Alarm	Alarma de Sobre Voltaje
59	1	2	BOOL	0 - 1	N/A	Current Unbalance Alarm Status	N/A	0 - Normal 1 - Alarm	Alarma Desbalance de Corriente
60	1	2	BOOL	0 - 1	N/A	Power Reversal Alarm Status	N/A	0 - Normal 1 - Alarm	Estado de alarma de inversión de energía
61	1	2	BOOL	0 - 1	N/A	GF Sum Alarm Status	N/A	0 - Normal 1 - Alarm	Alarma General
63	1	2	BOOL	0 - 1	N/A	Load Shedding Status	N/A	0 - Normal 1 - Alarm	Estado de desconexión de carga
14	1	4	INT (16 Bit) L	0 - 65535	1	Power Peak Demand Total	kW	N/A	Total demanda de Potencia pico

15	1		INT (16 Bit) H	0 - 65535					
18	1	4	INT (16 Bit)	0 - 65535	1	Voltage Phase A	Volt	N/A	Voltaje Fase A
19	1	4	INT (16 Bit)	0 - 65535	1	Voltage Phase B	Volt	N/A	Voltaje Fase B
20	1	4	INT (16 Bit)	0 - 65535	1	Voltage Phase C	Volt	N/A	Voltaje Fase C
21	1	4	INT (16 Bit)	0 - 65535	1	Current Phase A	Amp	N/A	Corriente Fase A
23	1	4	INT (16 Bit)	0 - 65535	1	Current Phase B	Amp	N/A	Corriente Fase B
25	1	4	INT (16 Bit)	0 - 65535	1	Current Phase C	Amp	N/A	Corriente Fase C
27	1	4	INT (16 Bit)	0 - 65535	1	Current Phase N	Amp	N/A	Corriente Neutro
31	1	4	INT (16 Bit) L	0 - 65535	1	Energy Total	kWH	N/A	Energia total
32	1		INT (16 Bit) H	0 - 65535					
36	1	4	INT (16 Bit)	0 - 65535	1	Power Factor Phase A	N/A	N/A	Factor de potencia Fase A
37	1	4	INT (16 Bit)	0 - 65535	1	Power Factor Phase B	N/A	N/A	Factor de potencia Fase B
38	1	4	INT (16 Bit)	0 - 65535	1	Power Factor Phase C	N/A	N/A	Factor de potencia Fase C
39	1	4	INT (16 Bit)	0 - 65535	1	Power Factor Total	N/A	N/A	Factor de potencia total
40	1	4	INT (16 Bit) L	0 - 65535	1	Power Real Phase A	kW	N/A	Potencia Activa Fase A
41	1		INT (16 Bit) H	0 - 65535					
42	1	4	INT (16 Bit) L	0 - 65535	1	Power Real Fase B	kW	N/A	Potencia Activa Fase B
43	1		INT (16 Bit) H	0 - 65535					
44	1	4	INT (16 Bit) L	0 - 65535	1	Power Real Fase C	kW	N/A	Potencia Activa Fase C
45	1		INT (16 Bit) H	0 - 65535					
46	1	4	INT (16 Bit) L	0 - 65535	1	Power Real Fase Total	kW	N/A	Potencia Activa Total
47	1		INT (16 Bit) H	0 - 65535					
54	1	4	INT (16 Bit) L	0 - 65535	1	Power Reactive Fase Total	kVAR	N/A	Potencia Reactiva Total
55	1		INT (16 Bit) H	0 - 65535					
62	1	4	INT (16 Bit) L	0 - 65535	1	Power Apparent Fase Total	kVAR	N/A	Potencia Aparente Total
63	1		INT (16 Bit) H	0 - 65535					
64	1	4	INT (16 Bit) L	0 - 65535	1	Power Demand Total	kW	N/A	Total de demanda de potencia
65	1		INT (16 Bit) H	0 - 65535					
66	1	4	INT (16 Bit)	0 - 65535	1	Frequency Measured	Hz	N/A	Frecuencia
215	1	3	INT (16 Bit)	0 - 11	1	Long Time Trip Pickup	N/A	1 -> 0.5 2 -> 0.55 3 -> 0.6	Ajuste de disparo largo

								4 -> 0.65 5 -> 0.7 6 -> 0.75 7 -> 0.8 8 -> 0.85 9 -> 0.9 10 -> 0.95 11 -> 1.0	
216	1	3	INT (16 Bit)	0 - 23	1	Long Time Trip Delay	N/A	0 -> Off 1 -> MVT1 2 -> MVT2 3 -> MVT3 4 -> MVT4 5 -> C- Min 6 - 14 -> C2..C10 15 -> C-Max 16 -> F-Min 17 - 22 -> F2..F7 23 -> F-Max	Retardo de disparo largo
222	1	3	INT (16 Bit)	1 - 16	1	Short Time Trip Pickup	N/A	1 - 1.5 2 - 2.0 3 - 2.5 4 - 3.0 5 - 3.5 6 - 4.0 7 - 4.5 8 - 5.0 9 - 5.5 10 - 6.0 11 - 6.5 12 - 7.0 13 - 7.5 14 - 8.0 15 - 8.5 16 - 9.0	Ajuste de disparo corto
223	1	3	INT (16 Bit)	0 - 12	1	Short Time Trip Delay	N/A	0 -> Off 1 -> Band1 2 -> Band2 3 -> Band3 4 -> Band4 5 -> Band5 6 -> Band6 7 -> Band7 8 -> Band8 9 -> Band9 10 -> Band10 11 -> Band11 12 -> Band12	Retardo de disparo corto
225	1	3	INT (16 Bit)	0 - 18	1	Instantaneous Trip Pickup	N/A	0 -> Off (For switchable Inst only) 1 -> Invalid 2 -> 2 3 -> 2.5 4 -> 3 5 -> 3.5 6 -> 4 7 -> 4.5 8 -> 5 9 -> 5.5 10 -> 6 11 -> 6.5	Ajuste de disparo instantaneo

									12 -> 7 13 -> 7.5 14 -> 8 15 -> 8.5 16 -> 9 17 -> 9.5 18 -> 10
--	--	--	--	--	--	--	--	--	--

Tabla 0-10 Variables de Breakers-PowerBreak (Los autores, 2024)

Dirección	Registros	Función	Tipo de Dato	Valores	Escala	Variable	Unidades	Equivalencia	Descripción
27	1	2	BOOL	0 - 1	N/A	Breaker position	N/A	0 - Open 1 - Closed	Posición de Breaker
51	1	2	BOOL	0 - 1	N/A	MET Tripped	N/A	0 - No 1 - Yes	Disparo de Breaker
61	1	2	BOOL	0 - 1	N/A	GF Sum Alarm Status	N/A	0 - Normal 1 - Alarm	Alarma General
62	1	2	BOOL	1 - 1	N/A	GF CT Alarm Status	N/A	1 - Normal 1 - Alarm	Alarma General
18	1	4	INT (16 Bit)	0 - 65535	1	Voltage Phase A	Volt	N/A	Voltaje Fase A
19	1	4	INT (16 Bit)	0 - 65535	1	Voltage Phase B	Volt	N/A	Voltaje Fase B
20	1	4	INT (16 Bit)	0 - 65535	1	Voltage Phase C	Volt	N/A	Voltaje Fase C
21	1	4	INT (16 Bit)	0 - 65535	1	Current Phase A	Amp	N/A	Corriente Fase A
23	1	4	INT (16 Bit)	0 - 65535	1	Current Phase B	Amp	N/A	Corriente Fase B
25	1	4	INT (16 Bit)	0 - 65535	1	Current Phase C	Amp	N/A	Corriente Fase C
27	1	4	INT (16 Bit)	0 - 65535	1	Current Phase N	Amp	N/A	Corriente Neutro
31	1	4	INT (16 Bit) LL	0 - 65535	1	Energy Total	kWH	N/A	Energía total
32	1		INT (16 Bit) L	0 - 65535					
33	1		INT (16 Bit) H	0 - 65535					
34	1		INT (16 Bit) HH	0 - 65535					
36	1	4	INT (16 Bit)	0 - 65535	1	Power Factor Phase A	N/A	N/A	Factor de potencia Fase A
37	1	4	INT (16 Bit)	0 - 65535	1	Power Factor Phase B	N/A	N/A	Factor de potencia Fase B
38	1	4	INT (16 Bit)	0 - 65535	1	Power Factor Phase C	N/A	N/A	Factor de potencia Fase C
39	1	4	INT (16 Bit)	0 - 65535	1	Power Factor Total	N/A	N/A	Factor de potencia total
40	1	4	INT (16 Bit) L	0 - 65535	1	Power Real Phase A	kW	N/A	Potencia Activa Fase A
41	1		INT (16 Bit) H	0 - 65535					
42	1	4	INT (16 Bit) L	0 - 65535	1	Power Real Fase B	kW	N/A	Potencia Activa Fase B
43	1		INT (16 Bit) H	0 - 65535					
44	1	4	INT (16 Bit) L	0 - 65535	1	Power Real Fase C	kW	N/A	Potencia Activa Fase C
45	1		INT (16 Bit) H	0 - 65535					

46	1	4	INT (16 Bit) L	0 - 65535	1	Power Real Fase Total	kW	N/A	Potencia Activa Total
47	1		INT (16 Bit) H	0 - 65535					
54	1	4	INT (16 Bit) L	0 - 65535	1	Power Reactive Fase Total	kVAR	N/A	Potencia Reactiva Total
55	1		INT (16 Bit) H	0 - 65535					
62	1	4	INT (16 Bit) L	0 - 65535	1	Power Apparent Fase Total	kVAR	N/A	Potencia Aparente Total
63	1		INT (16 Bit) H	0 - 65535					
64	1	4	INT (16 Bit) L	0 - 65535	1	Power Demand Total	kW	N/A	Total de demanda de potencia
65	1		INT (16 Bit) H	0 - 65535					
66	1	4	INT (16 Bit)	0 - 65535	1	Frequency Measured	Hz	N/A	Frecuencia
215	1	3	INT (16 Bit)	1 - 11	1	Long Time Trip Pickup	N/A	1 -> 0.5 2 -> 0.55 ... 10 -> 0.95 11 -> 1.0	Ajuste de disparo largo
216	1	3	INT (16 Bit)	0 - 44	1	Long Time Trip Delay	N/A	0 -> Off 1 -> C- Min ... 24 -> F-2 44 -> F-Max	Retardo de disparo largo
222	1	3	INT (16 Bit)	1 - 16	1	Short Time Trip Pickup	N/A	1 - 1.5 2 - 2.0 ...	Ajuste de disparo corto
223	1	3	INT (16 Bit)	0 - 17	1	Short Time Trip Delay	N/A	0 -> Off 1 -> Band1 2 -> Band2 ... 16 -> Band16 17 -> Band17	Retardo de disparo corto
225	1	3	INT (16 Bit)	0 - 44	1	Instantaneous Trip Pickup	N/A	0 -> Off (For switchable Inst only) 1 -> Invalid 2 -> 2 ... 43 -> 29 44 -> 30	Ajuste de disparo instantaneo

Tabla 0-11 Lazos de redes MODBUS (Los autores, 2024)

Equipos en red Modbus RTU					
SISTEMA 1					
PLC/RED	DISPOSITIVO	EQUIPO / REGISTRO	DESCRIPCION	ID. MODBUS	
PLC1_G1A		68			
RED 1	ATS-TDP	1			
		ATS - 03A		1	
	TABLERO-ATS-TDP	3			
		T_ATS - 03A-1 (EEE->Sis 1) (MPACT-Plus)		3	
		T_ATS - 03A-3 (G1->Sis 1) (MPACT-Plus)		5	
		T_ATS - 03A-2 (G2->Sis 1) (MPACT-Plus)		7	
	TABLERO-TDP	9			
		TDP-01A-0 (Power Break II)		9	
		TDP-01A-1 (Spectra RMS)		11	
		TDP-01A-2 (Power Break II)		13	
		TDP-01A-3 (Spectra RMS)		15	
		TDP-01A-4 (Spectra RMS)		17	
		TDP-01A-5 (Spectra RMS)		19	
		TDP-01A-6 (Spectra RMS)		21	
		TDP-01A-7 (Spectra RMS)		23	
		Medidor Parametros Digital EPM 2000		25	
	TABLERO-TD	4			
		TD-01A-0 (Sis 1 + Sis 2) (Power Break II)		27	
		TD-01A-1 (Sis 1) (Power Break II)		29	
		TD-01A-2 (Sis 2) (Power Break II)		31	
		PQMII 1A (Fase A)		33	
	TABLERO-PSG	3			
		PSG-01A-1 (G1->Sis 2) (Power Break II)		35	
		PSG-01A-2 (G1->Sis 1) (Power Break II)		37	
		PSG-01A-3 (G1->TPDU-01A-12) (Power Break II)		39	
	RED 2	TABLERO-UPS-CHILLER	20		
			T-CHI-01A-0 (Spectra RMS)		49
		T-CHI-01A-1 (Spectra RMS)		51	
		T-CHI-01A-2 (Spectra RMS)		53	
		T-CHI-01A-3 (Spectra RMS)		55	
	T-CHI-01A-4 (Spectra RMS)		57		

		T-CHI-01A-5 (Spectra RMS)		59	
		T-CHI-01A-6 (Spectra RMS)		61	
		T-CHI-01A-7 (Spectra RMS)		63	
		T-CHI-01A-8 (Spectra RMS)		65	
		T-CHI-01A-9 (Spectra RMS)		67	
		T-CHI-01A-10 (Spectra RMS)		69	
		T-CHI-01A-11 (Spectra RMS)		71	
		T-CHI-01A-12 (Spectra RMS)		73	
		T-CHI-01A-13 (Spectra RMS)		75	
		T-CHI-01A-14 (Spectra RMS)		77	
		T-CHI-01A-15 (Spectra RMS)		79	
		T-CHI-01A-16 (Spectra RMS)		81	
		T-CHI-01A-17 (Spectra RMS)		83	
		T-CHI-01A-18 (Spectra RMS)		85	
		T-CHI-01A-19 (Spectra RMS)		87	
RED 3	ATS-CHI	1			
		ATS-CHI-7A		89	
	TABLERO-PDU	14			
		T-PDU-01A-0 (Power Break II)		91	
		T-PDU-01A-1 (Spectra RMS)		93	
		T-PDU-01A-2 (Spectra RMS)		95	
		T-PDU-01A-3 (Spectra RMS)		97	
		T-PDU-01A-4 (Spectra RMS)		99	
		T-PDU-01A-5 (Spectra RMS)		101	
		T-PDU-01A-6 (Spectra RMS)		103	
		T-PDU-01A-7 (Spectra RMS)		105	
		T-PDU-01A-8 (Spectra RMS)		107	
		T-PDU-01A-9 (Spectra RMS)		109	
		T-PDU-01A-10 (Spectra RMS)		111	
		T-PDU-01A-11 (Spectra RMS)		113	
		T-PDU-01A-12 (Power Break II)		115	
		Medidor Parametros Digital EPM 2000		117	
		ATS-PDU	1		
		ATS-PDU-9A			119
	RED 4	TABLERO-CHILLER	1		
		Medidor Parametros Digital PM 710		129	
RED 4	TABLERO-UPS-IN	2			
		T-UPS-IN-1A-0 (Power Break II)		131	

		Medidor Parametros Digital EPM 2000		145
	TABLERO-UPS-OUT	2		
		T-UPS-OUT-1A-0 (Power Break II)		147
		T-UPS-OUT-1A-7 (Power Break II)		161
	TABLERO-SG	4		
		T-SG-01A-0 (Spectra RMS)		163
		T-SG-01A-1 (Spectra RMS)		165
		T-SG-01A-2 (Spectra RMS)		167
		T-SG-01A-3 (Spectra RMS)		169
	ATS-SG	1		
		ATS-SG-11A		171
	ATS-REC	1		
		ATS-REC-05A		173
	ATS-SAT	1		
		ATS-SAT-01A		175
PLC2_G1A		10		
RED 1	PDU	10		
		PDU-01A-1		100-103
		PDU-01A-2		90-93
		PDU-01A-3		80-83
		PDU-01A-4		70-73
		PDU-01A-5		60-63
		PDU-01A-6		50-53
		PDU-01A-7		40-43
		PDU-01A-8		30-33
		PDU-01A-9		20-23
		PDU-01A-10		10-13
SISTEMA 2				
PLC/RED	DISPOSITIVO	EQUIPO / REGISTRO	DESCRIPCION	ID. MODBUS
PLC1_G2A		58		
RED 1	ATS-TDP	1		
		ATS - 04A (Breaker ABB->EEE)		1
		ATS - 04A (Breaker ABB->GEN)		2
	TABLERO-ATS-TDP	3		
		T_ATS - 04A-1 (EEE->Sis 2) (MPACT-Plus)		3
		T_ATS - 04A-2 (G1->Sis 2) (MPACT-Plus)		5
		T_ATS - 04A-3 (G2->Sis 2) (MPACT-Plus)		7

	TABLERO-TDP	9		
		TDP-02A-0 (Power Break II)		9
		TDP-02A-1 (Spectra RMS)		11
		TDP-02A-2 (Power Break II)		13
		TDP-02A-3 (Spectra RMS)		15
		TDP-02A-4 (Spectra RMS)		17
		TDP-02A-5 (Spectra RMS)		19
		TDP-02A-6 (Spectra RMS)		21
		TDP-02A-7 (Spectra RMS)		23
		Medidor Parametros Digital EPM 2000		25
	TABLERO-PSG	3		
		PSG-02A-1 (G2->Sis 1) (Power Break II)		27
		PSG-02A-2 (G2->Sis 2) (Power Break II)		29
		PSG-02A-3 (G2->TPDU-02A-12) (Power Break II)		31
RED 2	TABLERO-UPS-CHILLER	20		
		T-CHI-02A-0 (Spectra RMS)		39
		T-CHI-02A-1 (Spectra RMS)		41
		T-CHI-02A-2 (Spectra RMS)		43
		T-CHI-02A-3 (Spectra RMS)		45
		T-CHI-02A-4 (Spectra RMS)		47
		T-CHI-02A-5 (Spectra RMS)		49
		T-CHI-02A-6 (Spectra RMS)		51
		T-CHI-02A-7 (Spectra RMS)		53
		T-CHI-02A-8 (Spectra RMS)		55
		T-CHI-02A-9 (Spectra RMS)		57
		T-CHI-02A-10 (Spectra RMS)		59
		T-CHI-02A-11 (Spectra RMS)		61
		T-CHI-02A-12 (Spectra RMS)		63
		T-CHI-02A-13 (Spectra RMS)		65
		T-CHI-02A-14 (Spectra RMS)		67
		T-CHI-02A-15 (Spectra RMS)		69
		T-CHI-02A-16 (Spectra RMS)		71
		T-CHI-02A-17 (Spectra RMS)		73
		T-CHI-02A-18 (Spectra RMS)		75
	T-CHI-02A-19 (Spectra RMS)		77	
RED 3	ATS-CHI	1		
		ATS-CHI-8A		79
	ATS-PDU	1		

		ATS-PDU-10A (Deep Sea 7420 RTU o TCP)		81
	TABLERO-UPS-IN	2		
		T-UPS-IN-2A-0 (Breaker ABB)		83
		Medidor Parametros Digital EPM 2000		97
	TABLERO-UPS-OUT	2		
		T-UPS-OUT-2A-0 (Power Break II)		99
		T-UPS-OUT-2A-7 (Power Break II)		113
	ATS-SAT	1		
		ATS-SAT-02A		115
	ATS-REC	1		
		ATS-REC-06A		116
RED 4	TABLERO-CHILLER	1		
		Medidor Parametros Digital PM 710		125
	TABLERO-PDU	13		
		T-PDU-02A-0 (Power Break II)		127
		T-PDU-02A-1 (Spectra RMS)		129
		T-PDU-02A-2 (Breaker ABB)		131
		T-PDU-02A-3 (Spectra RMS)		133
		T-PDU-02A-4 (Spectra RMS)		135
		T-PDU-02A-5 (Spectra RMS)		137
		T-PDU-02A-6 (Spectra RMS)		139
		T-PDU-02A-7 (Breaker ABB)		141
		T-PDU-02A-8 (Spectra RMS)		143
		T-PDU-02A-9 (Spectra RMS)		145
		T-PDU-02A-10 (Spectra RMS)		147
		T-PDU-02A-11 (Spectra RMS)		149
		T-PDU-02A-12 (Power Break II)		151
		Medidor Parametros Digital EPM 2000		153
PLC2_G2A		10		
RED 1	PDU	10		
		PDU-02A-1		100-103
		PDU-02A-2		90-93
		PDU-02A-3		80-83
		PDU-02A-4		70-73
		PDU-02A-5		60-63
		PDU-02A-6		50-53

		PDU-02A-7		40-43
		PDU-02A-8		30-33
		PDU-02A-9		20-23
		PDU-02A-10		10-13