

Documento de requerimientos de software

*Desarrollo de un sistema web para la gestión de sensores
lot incorporando algoritmos de machine learning para la
Predicción de variables objetivo*

Fecha: *[06/07/2025]*

Tabla de contenido

Historial de Versiones	4
Información del Proyecto	4
Aprobaciones	4
1. Propósito	5
2. Alcance del producto / Software	5
3. Referencias	6
4. Funcionalidades del producto	6
5. Clases y características de usuarios	7
6. Entorno operativo	7
7. Requerimientos funcionales	9
7.1. Monitoreo en tiempo real de variables eléctricas	9
7.2. Generación automática de reportes.....	10
7.3. Predicción de fallos mediante Machine Learning	10
8. Reglas de negocio	13
9. Requerimientos de interfaces externas.....	14
9.1. Interfaces de usuario	14
9.2. Interfaces de hardware	15
9.3. Interfaces de software.....	16
9.4. Interfaces de comunicación	16
10. Requerimientos no funcionales	16
10.1. Rendimiento.....	16
10.2. Usabilidad	17
10.3. Confiabilidad y Disponibilidad	17

10.4.	Seguridad	17
10.5.	Mantenibilidad y Escalabilidad	18
11.	Otros requerimientos	18
11.1.	Requerimientos de Base de Datos.....	18
11.2.	Internacionalización y Localización ¡Error! Marcador no definido.	
11.3.	Requerimientos Legales y de Cumplimiento	18
11.4.	Reutilización de Componentes.....	19
12.	Glosario	19

Historial de Versiones

Fecha	Versión	Autor	Organización	Descripción
16/04/2025	1.0	Jorge Reyes	Espoch	Versión inicial del documento
10/06/2025	1.1	Jorge Reyes	Espoch	Modificaciones

Información del Proyecto

Empresa / Organización	Escuela Superior Politécnica de Chimborazo
Proyecto	Administración de sensores IoT en sistemas fotovoltaicos
Fecha de preparación	16/04/2025
Cliente	Facultad de Informática y Electrónica
Patrocinador principal	Ing. Isaac Torres
Gerente / Líder de Proyecto	Jorge Reyes
Gerente / Líder de Análisis de negocio y requerimientos	Jorge Reyes

Aprobaciones

Nombre y Apellido	Cargo	Departamento u Organización	Fecha	Firma
Ing. Julio Santillán	Profesor/Revisor técnico	ESPOCH	06/07/2025	

1. Propósito

El presente documento especifica los requerimientos funcionales y no funcionales del sistema web NODALIS versión 1.0. Este software está diseñado para la administración eficiente de sensores IoT en diversos entornos empresariales, integrando algoritmos de Machine Learning para el análisis inteligente de variables físicas como temperatura, humedad, consumo energético u otras según el caso de uso.

El sistema cubre módulos como: gestión de sensores, monitoreo en tiempo real, procesamiento y análisis predictivo, visualización mediante dashboards, generación de reportes y alertas, y administración de usuarios con distintos niveles de acceso.

2. Alcance del producto / Software

NODALIS es una solución integral orientada al monitoreo, análisis y optimización del comportamiento de infraestructuras que dependen de variables físicas relevantes, mediante la integración de sensores IoT y técnicas de Machine Learning. El sistema está diseñado para recopilar datos de variables físicas en tiempo real, analizarlos de forma automatizada, predecir fallos potenciales y facilitar la toma de decisiones mediante visualizaciones gráficas y alertas inteligentes.

Beneficios:

- Mejora en la eficiencia del mantenimiento preventivo y correctivo de infraestructuras en entornos industriales.
- Detección temprana de condiciones anómalas que afecten la operación.
- Centralización del monitoreo de múltiples sensores IoT en una sola plataforma accesible vía web.
- Generación de reportes analíticos y alertas automatizadas para decisiones técnicas más rápidas y precisas.

Objetivos y metas:

- Desarrollar una plataforma web que centralice la recolección y análisis de datos provenientes de sensores IoT.
- Implementar modelos de Machine Learning para analizar patrones de funcionamiento y predecir fallos.
- Proporcionar visualizaciones claras y herramientas de alerta automática para múltiples tipos de variables físicas.

- Facilitar la administración de alertas y reportes automáticos sobre el estado y comportamiento de los sistemas monitoreados.
- Establecer una arquitectura escalable y segura que permita la implementación en diversos escenarios del sector energético.

Este documento se complementa con otros entregables como el documento de casos de uso, donde se especifican los escenarios operativos y funcionales del sistema.

3. Referencias

- Angelino dos Santos, T., Gomes de Freitas, F., Carvalho Gonçalves, D. L., & Fernández-Ramírez, L. M. (2022). Design and validation of IoT measurement system for photovoltaic generation. Ingenius, 28, 44–52. <https://doi.org/10.17163/ings.n28.2022.04>
- (S/f). Edu.co. Recuperado el 16 de abril de 2025, de <https://repository.ucatolica.edu.co/server/api/core/bitstreams/604e030e-b281-4cac-aea4-3bf94648e246/content>
- (S/f-b). Edu.ec. Recuperado el 16 de abril de 2025, de <http://dspace.esPOCH.edu.ec/handle/123456789/20473>
- (S/f-c). Edu.ec. Recuperado el 16 de abril de 2025, de <http://dspace.esPOCH.edu.ec/handle/123456789/2897>

4. Funcionalidades del producto

A continuación, se listan las principales funcionalidades que el sistema ofrecerá a sus usuarios:

1. Registro y configuración de sensores IoT.
2. Visualización en tiempo real de variables eléctricas (voltaje, corriente, potencia, etc.).
3. Procesamiento de datos eléctricos y almacenamiento histórico.
4. Análisis predictivo mediante algoritmos de Machine Learning.
5. Generación de reportes automáticos de rendimiento y diagnóstico.
6. Emisión de alertas por fallos, anomalías o comportamiento anómalo.
7. Gestión de usuarios, roles y niveles de acceso.

8. Administración de dispositivos conectados y configuración de parámetros del sistema.
9. Visualización gráfica e interactiva de métricas energéticas.
10. Descarga de reportes en formatos estándar (PDF, CSV).

5. Clases y características de usuarios

El sistema está diseñado para ser utilizado por diferentes tipos de usuarios, según su rol, experiencia técnica, y nivel de interacción con el sistema. A continuación, se describen los principales perfiles de usuario:

- Administrador del Sistema
 - Funcionalidades relevantes: 1, 7, 8, 11
 - Descripción: Encargado de la configuración general del sistema, administración de usuarios y dispositivos, y monitoreo de la operatividad general.
 - Frecuencia de uso: Alta
- Usuario
 - Funcionalidades relevantes: 9, 10
 - Descripción: Acceso restringido para visualización de datos o reportes puntuales sin posibilidad de realizar configuraciones o ajustes.
 - Frecuencia de uso: Baja

6. Entorno operativo

El sistema se desarrollará como una aplicación web que permita la administración, monitoreo y análisis de sensores IoT. Su entorno operativo abarcará tanto componentes cliente (front-end) como servidor (back-end), así como la integración con dispositivos físicos IoT.

Plataforma de Hardware:

- Servidores:
 - Procesador Intel Xeon o AMD EPYC, mínimo 8 núcleos
 - Memoria RAM mínima de 16 GB
 - Almacenamiento SSD (mínimo 500 GB), escalable en la nube
 - Conectividad a red estable (mínimo 100 Mbps)
- Dispositivos Cliente (usuarios):

- Computadoras de escritorio, laptops, tablets o smartphones con navegador web actualizado
- Resolución mínima recomendada: 1366x768
- Sensores IoT compatibles:
 - Dispositivos con conectividad Wi-Fi o LoRaWAN
 - Protocolos de comunicación MQTT, HTTP
 - Equipos capaces de medir variables físicas (voltaje, corriente, potencia y temperatura)

Sistemas Operativos Compatibles:

- Servidor web:
 - Linux (Ubuntu Server 22.04 LTS o superior)
 - Alternativamente, servidores con Windows Server 2019 o superior (según configuración)
- Cliente (usuarios):
 - Compatible con navegadores modernos (Chrome, Firefox, Edge, Safari) en sistemas.

Entorno de desarrollo y tecnologías utilizadas:

- Back-end:
 - Node.js / Python (Flask o Django)
 - Base de datos PostgreSQL
 - API RESTful
- Front-end:
 - Angular
 - Frameworks gráficos: Chart.js para visualización
- Machine Learning:
 - Modelos entrenados en Python (scikit-learn, TensorFlow o PyTorch)
 - Integración mediante servicios REST o microservicios
- Infraestructura de despliegue:
 - Servidor privado
 -
- Compatibilidad IoT:
 - Integración vía brokers MQTT
 - Middleware para captura y envío de datos de sensores hacia el sistema

7. Requerimientos funcionales

7.1. Monitoreo en tiempo real de variables físicas.

Descripción:

Permite visualizar en tiempo real los valores físicos (como temperatura, humedad, energía, presión, etc.) recolectados por sensores IoT en diversos entornos (industriales, agrícolas, ambientales, etc).

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado:

- El usuario inicia sesión en la plataforma.
- Selecciona el módulo de monitoreo.
- El sistema carga y actualiza constantemente los datos provenientes de los sensores.
- Los datos se muestran en dashboards gráficos y tablas actualizables.
- Si se detectan valores anormales, se activan alertas visuales y/o sonoras.

Requerimientos funcionales:

- **REQ-1:** El sistema debe conectarse a una base de datos de sensores IoT para obtener datos actualizados cada 10 segundos.
- **REQ-2:** El sistema debe mostrar los valores eléctricos en tiempo real mediante gráficos interactivos (líneas, barras, etc.).
- **REQ-3:** El sistema debe identificar y resaltar valores fuera de los umbrales definidos por el usuario.
- **REQ-4:** El sistema debe actualizar automáticamente el dashboard sin necesidad de recargar la página.
- **REQ-5:** El sistema debe permitir al usuario filtrar los datos por sensor, fecha y tipo de variable.

7.2. Generación automática de reportes

Descripción:

Permite generar reportes con datos históricos de sensores, gráficos de comportamiento, resumen de fallos detectados y predicciones generadas por los modelos de Machine Learning.

Prioridad: Media

Acciones iniciadoras y comportamiento esperado:

- El usuario accede al módulo de reportes.
- Selecciona el periodo de tiempo, sensores involucrados y el tipo de análisis requerido.
- El sistema procesa la información y genera un documento PDF o Excel.
- El reporte puede ser visualizado, descargado o enviado por correo.

Requerimientos funcionales:

- **REQ-6:** El sistema debe permitir al usuario seleccionar el intervalo de tiempo para el reporte.
- **REQ-7:** El sistema debe generar reportes en formato PDF y/o Excel.
- **REQ-8:** El sistema debe incluir gráficos, tablas y un resumen de fallos detectados.
- **REQ-9:** El sistema debe almacenar los reportes generados y permitir su consulta posterior.
- **REQ-10:** El sistema debe permitir enviar reportes automáticamente a direcciones de correo configuradas.

7.3. Predicción de fallos mediante Machine Learning

Descripción:

Utiliza modelos de Machine Learning entrenados con datos históricos para detectar patrones que indiquen probabilidad de fallo en los paneles solares o componentes eléctricos.

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado:

- El sistema ejecuta de forma periódica (por ejemplo, cada hora) el análisis predictivo.
- Se identifican comportamientos anómalos y se calcula un nivel de riesgo.
- Si el riesgo es alto, se genera una alerta.
- El usuario puede revisar las predicciones, ver la causa estimada y acceder a recomendaciones.

Requerimientos funcionales:

- **REQ-11:** El sistema debe contar con un modelo de ML entrenado para detectar patrones de fallo.
- **REQ-12:** El sistema debe analizar los datos recolectados cada hora y evaluar riesgos.
- **REQ-13:** El sistema debe notificar al usuario mediante una alerta visual y sonora cuando se detecte una predicción de fallo.
- **REQ-14:** El sistema debe registrar las predicciones realizadas y permitir su consulta histórica.
- **REQ-15:** El sistema debe mostrar al usuario recomendaciones asociadas al tipo de fallo probable.

7.4. Gestión de sensores y nodos

Descripción:

Permite al usuario administrador registrar, editar o eliminar sensores y nodos (ubicaciones físicas donde se instalan sensores).

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado:

- El administrador inicia sesión en el sistema.
- Accede al módulo de administración de sensores y nodos.
- Registra un nuevo nodo indicando su ubicación.
- Registra sensores indicando el tipo de variable, el nodo asociado y una etiqueta identificadora.
- El sistema valida los datos e inserta el nuevo sensor/nodo en la base de datos.

- El usuario puede editar o eliminar sensores existentes si es necesario.

Requerimientos funcionales:

- **REQ-16:** El sistema debe permitir registrar sensores indicando nombre, tipo de variable medida y nodo asociado.
- **REQ-17:** El sistema debe permitir registrar nodos con ubicación geográfica o nombre lógico.
- **REQ-18:** El sistema debe permitir editar y eliminar sensores o nodos registrados.
- **REQ-19:** El sistema debe validar que cada sensor esté asociado a un nodo activo.

7.5. Gestión de usuarios y roles

Descripción:

El sistema permite controlar el acceso mediante autenticación de usuarios con diferentes niveles de permisos.

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado:

- El usuario accede a la página de inicio de sesión.
- Ingresa sus credenciales y es autenticado.
- Según su rol, ve las funciones habilitadas (ej. un técnico no puede modificar sensores).
- El administrador puede registrar nuevos usuarios desde el módulo de administración.
- El sistema asigna el rol correspondiente y guarda el nuevo usuario en la base de datos.

Requerimientos funcionales:

- **REQ-20:** El sistema debe permitir registrar nuevos usuarios y asignarles un rol (Administrador, Técnico, Observador).
- **REQ-21:** El sistema debe autenticar usuarios mediante credenciales únicas.
- **REQ-22:** El sistema debe restringir funciones según el rol asignado.

7.6. Configuración de alertas por umbral

Descripción:

El usuario puede definir umbrales de variables físicas para que el sistema genere alertas automáticas si se superan.

Prioridad: Alta

Acciones iniciadoras y comportamiento esperado:

- El usuario con permisos accede al módulo de configuración de alertas.
- Selecciona el tipo de variable y define los valores mínimos y máximos aceptables.
- El sistema almacena estos valores y los aplica en el monitoreo en tiempo real.
- Cuando un dato excede el umbral, el sistema muestra una alerta en pantalla y/o la envía por correo.

Requerimientos funcionales:

- **REQ-23:** El sistema debe permitir definir umbrales mínimos y máximos por tipo de variable.
- **REQ-24:** El sistema debe generar alertas visuales o por correo si se exceden los umbrales.

8. Reglas de negocio

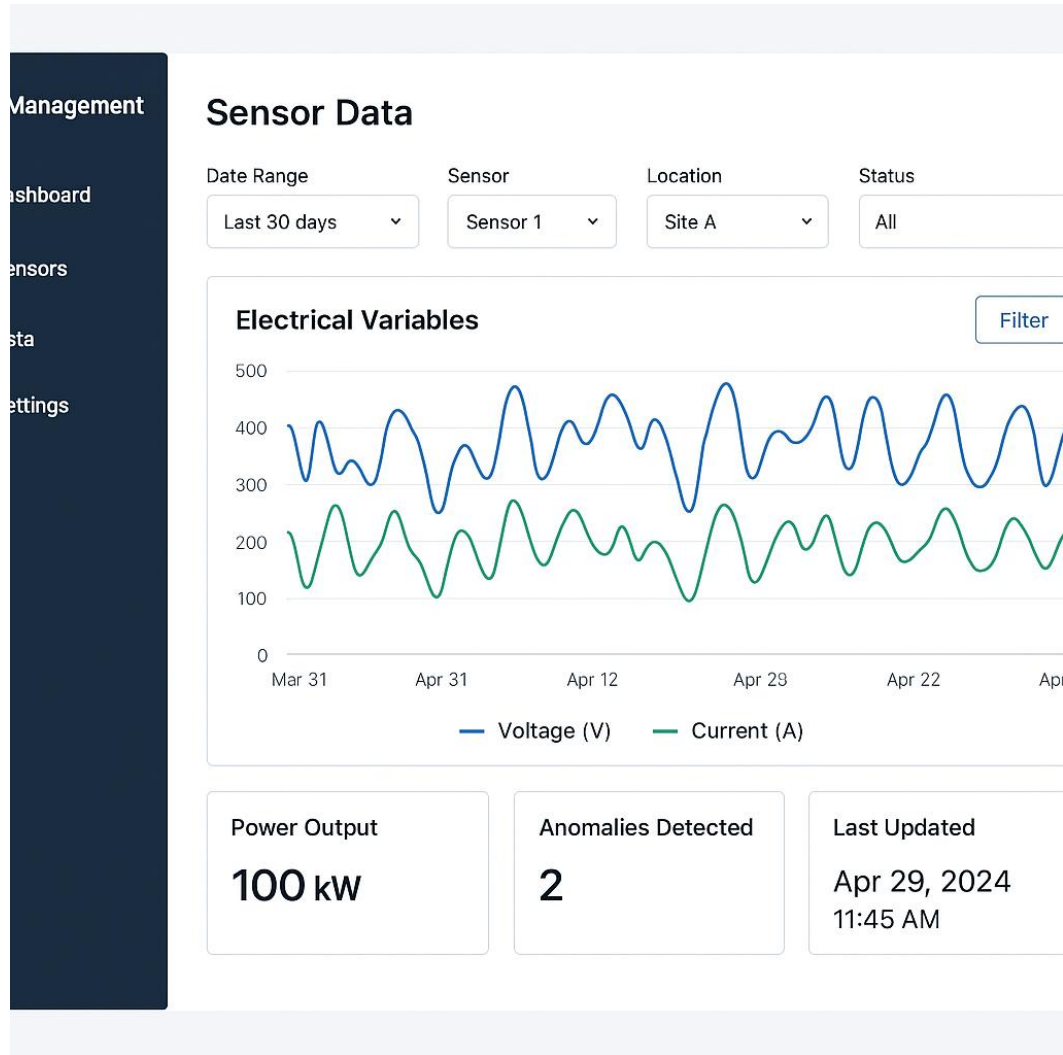
- **RN-1:** Solo los usuarios con rol de Administrador podrán configurar sensores IoT, gestionar usuarios, modificar umbrales de alerta y acceder al módulo de análisis predictivo.
- **RN-2:** Los usuarios con rol de Usuario solo podrán acceder a la visualización en tiempo real y a reportes pre-generados, sin modificar configuraciones.
- **RN-3:** Los usuarios deben autenticarse mediante credenciales válidas y únicas para acceder al sistema.
- **RN-4:** Los datos recolectados por los sensores deben almacenarse de forma encriptada y conservarse durante un mínimo de 12 meses para auditoría y análisis histórico.

- **RN-5:** El sistema debe validar que cada sensor esté correctamente registrado y asociado a un nodo antes de iniciar la recolección de datos.
- **RN-6:** Las alertas generadas (por superación de umbrales o por predicción de fallos) deben registrarse automáticamente y no podrán ser eliminadas por ningún tipo de usuario.
- **RN-7:** Cada reporte generado debe incluir metadatos (usuario que lo generó, fecha, rango analizado, sensores implicados) y contar con una firma digital o huella única que garantice su integridad.
- **RN-8:** El sistema solo permitirá modificar configuraciones o eliminar datos si el usuario tiene permisos explícitos asignados por el administrador.

9. Requerimientos de interfaces externas

9.1. Interfaces de usuario

- El sistema contará con una interfaz web responsive, accesible desde navegadores modernos (Chrome, Firefox, Edge).
- Se utilizarán menús laterales con iconografía clara para navegación principal.
- Las pantallas tendrán un diseño modular tipo dashboard, con tarjetas para datos y gráficas.
- Botones de acción siempre estarán ubicados en la parte inferior derecha, con colores estándar:
 - Verde: Confirmar/Guardar
 - Rojo: Eliminar
 - Azul: Ver más detalles
- Todas las pantallas incluirán encabezado con nombre del usuario, fecha, y botón de cerrar sesión.
- Se seguirá una guía de estilo uniforme para tipografía (Sans-serif), tamaño de letra mínimo legible (14px), y colores accesibles.



9.2. Interfaces de hardware

- Dispositivos soportados:
 - Computadoras de escritorio
 - Laptops
 - Tablets
- El sistema soportará comunicación con sensores IoT compatibles con protocolos como MQTT y HTTP RESTful.
- El backend debe ser capaz de interactuar con microcontroladores como ESP32 conectados a sensores eléctricos.

9.3. Interfaces de software

- El sistema se integrará con una base de datos PostgreSQL para almacenamiento estructurado de datos.
- La interfaz backend será una API RESTful desarrollada en Node.js.
- Se usará una librería de gráficos como Chart.js para visualizaciones.
- El módulo de Machine Learning será desarrollado en Python (scikit-learn o TensorFlow) y se expondrá mediante una API.
- El sistema debe ser compatible con sistemas operativos Linux y Windows Server en el entorno de producción.

9.4. Interfaces de comunicación

- Las comunicaciones se realizarán bajo el protocolo HTTPS para garantizar la seguridad.
- Los datos enviados desde los sensores usarán protocolos MQTT sobre TLS o HTTP con token de autenticación.
- Las notificaciones por correo electrónico serán enviadas mediante SMTP autenticado, con opción de integración con servicios como Gmail o SendGrid.
- Las credenciales y sesiones de usuario estarán protegidas mediante JWT (JSON Web Token).
- El sistema podrá recibir datos mediante formularios electrónicos o desde dispositivos vía API.

10. Requerimientos no funcionales

10.1. Rendimiento

- **RNF-1:** El sistema debe ser capaz de procesar e interpretar datos provenientes de al menos 100 sensores IoT simultáneamente, con una latencia máxima de 2 segundos para visualización en tiempo real.

- **RNF-2:** Las consultas de reportes históricos no deben superar los 5 segundos de respuesta en periodos de hasta 1 año.
- **RNF-3:** El sistema deberá manejar al menos 10 solicitudes concurrentes de usuarios sin degradación significativa del rendimiento.

10.2. Usabilidad

- **RNF-4:** El sistema deberá ser accesible desde navegadores web actuales sin requerir instalación de software adicional.
- **RNF-5:** El diseño de la interfaz deberá seguir principios de usabilidad y accesibilidad web (WCAG 2.1).
- **RNF-6:** El usuario debe poder realizar una tarea común (como generar un reporte o consultar un sensor) en no más de 5 pasos.
- **RNF-7:** El sistema deberá incluir mensajes de error claros y orientaciones contextuales para tareas críticas.

10.3. Confiabilidad y Disponibilidad

- **RNF-8:** El sistema debe tener una disponibilidad mínima del 99% mensual, excluyendo periodos programados de mantenimiento.
- **RNF-9:** En caso de pérdida de conexión con sensores, el sistema debe registrar el incidente y reintentar la conexión cada 30 segundos.
- **RNF-10:** El sistema debe contar con un mecanismo de respaldo automático de datos cada 24 horas.

10.4. Seguridad

- **RNF-11:** Todo acceso al sistema debe realizarse mediante autenticación segura con credenciales cifradas.
- **RNF-12:** Los datos en tránsito entre sensores, servidor y cliente deben estar cifrados mediante TLS 1.2 o superior.
- **RNF-13:** Se debe implementar un control de acceso basado en roles (RBAC) para limitar funcionalidades según el perfil del usuario.

- **RNF-14:** Todos los intentos de acceso fallido deben ser registrados con IP, usuario y timestamp.
- **RNF-15:** El sistema deberá cumplir con normativas de protección de datos personales aplicables, como la GDPR o su equivalente local.

10.5. Mantenibilidad y Escalabilidad

- **RNF-16:** El sistema deberá estar modularizado para permitir la actualización independiente de los módulos (por ejemplo: módulo de ML, módulo de sensores, módulo de usuarios).
- **RNF-17:** La arquitectura del sistema deberá ser escalable horizontalmente, permitiendo agregar más instancias del servidor en caso de crecimiento de la demanda.
- **RNF-18:** Se deberá mantener una documentación técnica y de usuario actualizada con cada nueva versión del sistema.

11. Otros requerimientos

11.1. Requerimientos de Base de Datos

- **ORQ-1:** El sistema deberá utilizar una base de datos relacional para almacenar registros de sensores, usuarios, reportes, alertas y configuraciones.
- **ORQ-2:** Se deberá garantizar la integridad referencial de los datos mediante el uso de llaves primarias y foráneas.
- **ORQ-3:** Se deben implementar respaldos automáticos diarios de la base de datos y almacenamiento redundante.
- **ORQ-4:** Las consultas a la base de datos deben estar optimizadas mediante índices para mejorar el rendimiento de lectura de datos históricos.

11.2. Requerimientos Legales y de Cumplimiento

- **ORQ-9:** Deben registrarse logs de acceso y actividad de usuarios durante un mínimo de 6 meses por temas de auditoría.

- **ORQ-10:** Toda la información de los usuarios y datos recolectados debe ser almacenada de forma segura y en servidores autorizados por la entidad propietaria del sistema.

11.3. Reutilización de Componentes

- **ORQ-11:** En el desarrollo del sistema se debe considerar la reutilización de componentes existentes como librerías de visualización de datos, frameworks web, y paquetes de Machine Learning previamente evaluados (por ejemplo, scikit-learn, TensorFlow, Plotly).

12. Glosario

Término / Sigla	Descripción
IoT (Internet of Things)	Red de dispositivos físicos interconectados que recopilan y comparten datos a través de Internet. En este proyecto, se refiere a los sensores que monitorean variables eléctricas.
ML (Machine Learning)	Rama de la inteligencia artificial que permite a los sistemas aprender automáticamente a partir de datos. Se utiliza en este sistema para analizar patrones y predecir fallas o comportamientos anómalos.
Variable Eléctrica	Parámetro eléctrico medido por sensores IoT, como voltaje, corriente, potencia o frecuencia.
Sensor	Dispositivo de IoT encargado de medir variables eléctricas en el sistema fotovoltaico.
Dashboard	Interfaz gráfica donde el usuario puede visualizar los datos en tiempo real y acceder a reportes.
Análisis Predictivo	Técnica basada en Machine Learning que permite anticipar eventos como fallos o sobrecargas en el sistema fotovoltaico.