

---

## Especificación de requisitos de software

Proyecto: Desarrollo de un Sistema Web para la  
Gestión de Sensores IoT Incorporando Algoritmos de  
Machine Learning para la Predicción de Variables

Objetivo

Revisión: 1.0

---

# Ficha del documento

Fecha	Revisión	Autor	Verificado dep. calidad.
05/10/2025	1.0	Paulina Peláez	

Documento validado por las partes en fecha: 05/102025

Por el cliente	Por la empresa suministradora
Fdo. D./ Dña GITEA	Fdo. D./Dña NODALIS



## Contenido

<b>FICHA DEL DOCUMENTO</b>	<b>2</b>
<b>CONTENIDO</b>	<b>3</b>
<b>1 INTRODUCCIÓN</b>	<b>5</b>
1.1 Propósito	5
1.2 Alcance	5
1.3 Personal involucrado	5
1.4 Definiciones, acrónimos y abreviaturas	6
1.5 Referencias	6
1.6 Resumen	7
<b>2 DESCRIPCIÓN GENERAL</b>	<b>7</b>
2.1 Perspectiva del producto	7
2.2 Funcionalidad del producto	8
2.3 Características de los usuarios	8
2.4 Restricciones	8
2.5 Suposiciones y dependencias	9
2.6 Evolución previsible del sistema	9
<b>3 REQUISITOS ESPECÍFICOS</b>	<b>9</b>
<b>3.1 Requisitos comunes de los interfaces</b>	<b>9</b>
3.1.1 Interfaces de usuario	9
3.1.2 Interfaces de hardware	9
3.1.3 Interfaces de software	10
3.1.4 Interfaces de comunicación	10
<b>3.2 Requisitos funcionales</b>	<b>10</b>
3.2.1 Requisito funcional 1	10
3.2.2 Requisito funcional 2	10
3.2.3 Requisito funcional 3	11
3.2.4 Requisito funcional 4	11
3.2.5 Requisito funcional 5	11
3.2.6 Requisito funcional 6	11
3.2.7 Requisito funcional 7	11
3.2.8 Requisito funcional 8	12
3.2.9 Requisito funcional 9	12
3.2.10 Requisito funcional 10	12



---

3.2.11	Requisito funcional 11	12
3.2.12	Requisito funcional 12	12
3.2.13	Requisito funcional 13	13
3.2.14	Requisito funcional 14	13
3.2.15	Requisito funcional 15	13
3.2.16	Requisito funcional 16	13
<b>3.3</b>	<b>Requisitos no funcionales</b>	<b>13</b>
3.3.1	Requisitos de rendimiento	13
3.3.1.1	Requisito no funcional 1	14
3.3.1.2	Requisito no funcional 2	14
3.3.1.3	Requisito no funcional 3	14
3.3.2	Seguridad	14
3.3.2.1	Requisito no funcional 4	14
3.3.2.2	Requisito no funcional 5	14
3.3.2.3	Requisito no funcional 6	15
3.3.3	Fiabilidad	15
3.3.3.1	Requisito no funcional 7	15
3.3.3.2	Requisito no funcional 8	15
3.3.4	Disponibilidad	15
3.3.4.1	Requisito no funcional 9	15
3.3.5	Mantenibilidad	16
3.3.5.1	Requisito no funcional 10	16
3.3.5.2	Requisito no funcional 11	16
3.3.5.3	Requisito no funcional 12	16
3.3.6	Portabilidad	16
3.3.6.1	Requisito no funcional 13	16
3.3.6.2	Requisito no funcional 14	16
3.3.6.3	Requisito no funcional 15	17
<b>3.4</b>	<b>Otros requisitos</b>	<b>17</b>
3.4.1	Otro Requisito 1	17
3.4.2	Otro Requisito 2	17
3.4.3	Otro Requisito 3	17
<b>4</b>	<b>APÉNDICES</b>	<b>18</b>
<b>4.1</b>	<b>Glosario de términos</b>	<b>18</b>
<b>4.2</b>	<b>Referencias técnicas</b>	<b>18</b>
<b>4.3</b>	<b>Consideraciones éticas y sostenibles</b>	<b>19</b>



# 1 Introducción

La presente Especificación de Requisitos de Software (SRS) describe de manera detallada las funcionalidades, restricciones y características que debe cumplir el sistema web para la gestión de sensores IoT, el procesamiento de datos y la predicción de variables eléctricas mediante algoritmos de Machine Learning. Este documento proporciona una visión integral del producto, su propósito, alcance, definiciones, participantes, referencias y estructura general, conforme a la norma IEEE 830-1998.

## 1.1 Propósito

### •Propósito del documento

El propósito de este documento es definir los requisitos funcionales y no funcionales que regirán el diseño, desarrollo, implementación y validación del **Sistema Web para la Gestión de Sensores IoT Incorporando Algoritmos de Machine Learning para la Predicción de Variables Objetivo**.

Este documento servirá como base de comunicación entre el equipo desarrollador, los tutores académicos y los usuarios finales del sistema. Además, garantizará que las expectativas del cliente y las especificaciones técnicas estén alineadas antes del inicio de la fase de desarrollo.

### •Audiencia a la que va dirigido

Este documento está dirigido a:

- **Desarrolladores:** para comprender los requerimientos técnicos y funcionales.
- **Tutores y revisores académicos:** para evaluar la coherencia del diseño con los objetivos del proyecto.
- **Usuarios finales (investigadores y técnicos):** para validar la usabilidad y funcionalidad del sistema.
- **Gestores de proyecto:** para monitorear el cumplimiento de los hitos, riesgos y entregables.

## 1.2 Alcance

El producto para desarrollar es un **sistema web integral** que permitirá la **gestión, monitoreo y análisis de datos provenientes de sensores IoT instalados en sistemas fotovoltaicos**, aplicando algoritmos de **Machine Learning** para la predicción de variables eléctricas como voltaje, corriente, potencia e irradiancia.

Este sistema integrará cuatro componentes principales:

1. **Capa IoT:** adquisición de datos mediante sensores conectados a microcontroladores ESP32.
2. **Backend:** procesamiento, almacenamiento y gestión de la información usando Node.js y PostgreSQL.
3. **Módulo de Machine Learning:** análisis predictivo en Python con TensorFlow y Keras.
4. **Frontend:** visualización de datos, reportes y alertas en una interfaz web desarrollada con Angular.

El alcance del documento es consistente con la descripción general establecida en el **anteproyecto del sistema** y los lineamientos institucionales de la **ESPOCH**.

## 1.3 Personal involucrado

Nombre	Paulina Peláez
Rol	Desarrollador Principal
Categoría profesional	Estudiante de Ingeniería en Software
Responsabilidades	Desarrollo del frontend, integración de algoritmos de Machine Learning, documentación técnica.
Información de contacto	katherine.pelaez@epoch.edu.ec
Aprobación	si



Nombre	Jorge Reyes
Rol	Desarrollador Principal
Categoría profesional	Estudiante de Ingeniería en Software
Responsabilidades	Desarrollo del frontend, integración de algoritmos de Machine Learning, documentación técnica.
Información de contacto	jorge.reyes@epoch.edu.ec
Aprobación	si

## 1.4 Definiciones, acrónimos y abreviaturas

Internet of Things (IoT): Se refiere al conjunto de dispositivos físicos interconectados que recopilan, envían y reciben datos a través de Internet. En este proyecto, el término hace referencia a los sensores instalados en sistemas fotovoltaicos que monitorean variables eléctricas y climáticas en tiempo real.

Machine Learning (ML): Rama de la inteligencia artificial que permite a los sistemas aprender de los datos históricos y realizar predicciones o clasificaciones sin intervención humana directa. En el presente proyecto, se aplica para predecir el rendimiento energético de los paneles solares y detectar posibles anomalías.

API (Application Programming Interface): Conjunto de funciones, protocolos y herramientas que permiten la comunicación entre diferentes componentes de software. El sistema utilizará APIs RESTful para la comunicación entre el backend, el frontend y los servicios de Machine Learning.

MQTT (Message Queuing Telemetry Transport): Protocolo de mensajería ligero diseñado para dispositivos IoT. Su función principal es la transmisión eficiente de datos entre los sensores y el servidor central, garantizando bajo consumo de ancho de banda y energía.

REST (Representational State Transfer): Estilo de arquitectura de software que define cómo los recursos del sistema pueden ser accedidos y manipulados mediante peticiones HTTP. Se implementará para el diseño de los servicios web del sistema.

SRS (Software Requirements Specification): Documento formal que describe las funcionalidades, restricciones y características técnicas que debe cumplir un sistema de software. Este documento corresponde a la SRS del proyecto desarrollado.

ISO/IEC 25010: Norma internacional que establece el modelo de calidad para productos de software, definiendo atributos como funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y seguridad.

## 1.5 Referencias

Referencia	Título	Ruta / Fuente	Fecha	Autor / Organización
[1]	<i>IEEE Std 830-1998: Software Requirements Specification</i>	IEEE Standards Association	1998	IEEE
[2]	<i>ISO/IEC 25010: Systems and Software Quality Requirements and Evaluation</i>	ISO	2011	ISO



[3]	<i>Documento: Alcance de un Proyecto</i>	Material académico ESPOCH	2025	ESPOCH
[4]	<i>Anteproyecto: Sistema Web IoT con Machine Learning</i>	Archivo institucional ESPOCH	2025	Reyes & Peláez
[5]	<i>TensorFlow Documentation</i>	<a href="https://www.tensorflow.org">https://www.tensorflow.org</a>	2024	Google AI

## 1.6 Resumen

El presente documento se organiza de la siguiente forma:

- Sección 1: Introducción general, propósito, alcance, personal involucrado y referencias.
- Sección 2: Descripción general del sistema, su entorno y funcionalidades principales.
- Sección 3: Requisitos funcionales y no funcionales detallados.
- Sección 4: Modelado del sistema y relaciones entre componentes.
- Sección 5: Criterios de aceptación, validación y aprobación del software.

En conjunto, este documento proporciona la base técnica necesaria para garantizar que el desarrollo del sistema cumpla con los objetivos funcionales, técnicos y de calidad definidos.

# 2 Descripción general

## 2.1 Perspectiva del producto

El sistema web para la gestión de sensores IoT es un **producto independiente**, diseñado como una solución integral que combina la adquisición de datos, el análisis predictivo y la visualización de información en tiempo real. No obstante, su arquitectura modular permite su futura integración con otros sistemas o plataformas de monitoreo energético.

El producto forma parte de un **ecosistema IoT-web-ML**, en el cual los sensores instalados en sistemas fotovoltaicos recolectan datos eléctricos y climáticos que se transmiten hacia un servidor central. Dichos datos son almacenados, procesados y analizados mediante algoritmos de *Machine Learning*, cuyos resultados se presentan en una interfaz web intuitiva.

En este contexto, el sistema se compone de los siguientes módulos interconectados:

- **Capa IoT:** conformada por sensores y microcontroladores ESP32 encargados de recolectar datos.
- **Capa de comunicación:** protocolo MQTT/MQTTS para el envío eficiente y seguro de la información.
- **Backend:** servidor desarrollado en Node.js que gestiona las solicitudes, el almacenamiento y las reglas del negocio.
- **Módulo de Machine Learning:** motor predictivo desarrollado en Python con TensorFlow, encargado de analizar y anticipar el rendimiento de los sistemas fotovoltaicos.
- **Frontend:** interfaz web en Angular que permite la interacción del usuario con los datos procesados.
- **Base de datos:** sistema PostgreSQL para el almacenamiento estructurado de información.

En resumen, el producto funciona como un **sistema web autónomo**, con potencial de ampliación hacia redes de monitoreo fotovoltaico de mayor escala.



## 2.2 Funcionalidad del producto

El sistema propuesto contará con las siguientes funcionalidades principales:

- **Gestión de usuarios:** registro, autenticación y control de roles (administrador, investigador y técnico).
- **Administración de sensores IoT:** registro, configuración y monitoreo del estado de los dispositivos.
- **Adquisición de datos:** recepción de lecturas en tiempo real de variables eléctricas y climáticas.
- **Procesamiento y almacenamiento:** limpieza, validación y registro de los datos en la base de datos PostgreSQL.
- **Análisis predictivo:** aplicación de algoritmos de *Machine Learning* para anticipar la generación de energía o detectar anomalías.
- **Visualización de información:** presentación de datos mediante dashboards, gráficos y reportes exportables.
- **Alertas automáticas:** notificación de fallas o desviaciones en el funcionamiento del sistema fotovoltaico.
- **Reportes personalizados:** generación de informes técnicos con indicadores de rendimiento.

Estas funcionalidades se organizarán de forma modular, garantizando escalabilidad, mantenibilidad y facilidad de actualización.

## 2.3 Características de los usuarios

Tipo de usuario	Administradores, investigadores y técnicos.
Formación	Los administradores e investigadores deberán contar con formación en ingeniería, preferentemente en software, eléctrica o electrónica. Los técnicos requerirán conocimientos básicos en sistemas fotovoltaicos y manejo de herramientas web.
Habilidades	Capacidad para interpretar gráficos y reportes de datos eléctricos. Conocimientos básicos sobre sensores IoT y comunicación de datos. Comprensión de conceptos de rendimiento energético y predicciones.
Actividades	Los administradores gestionan usuarios, proyectos y sensores. Los investigadores analizan datos, revisan predicciones y exportan reportes. Los técnicos supervisan el estado de los sensores y corrigen incidencias detectadas por el sistema.

## 2.4 Restricciones

El diseño y desarrollo del sistema deberá cumplir con las siguientes restricciones:

- **Metodología de desarrollo:** se empleará **SCRUM**, permitiendo entregas iterativas y control del progreso.
- **Lenguajes de programación:** Python para el módulo de Machine Learning, JavaScript/TypeScript para el frontend y backend, y SQL para la gestión de datos.
- **Frameworks definidos:** Angular (frontend), Node.js + Express (backend), TensorFlow (ML) y Sequelize (ORM).
- **Base de datos:** PostgreSQL como gestor de base de datos relacional.
- **Hardware:** microcontroladores ESP32 y sensores eléctricos de bajo consumo.
- **Conectividad:** disponibilidad de red Wi-Fi estable para la transmisión de datos IoT.
- **Normas de calidad:** el sistema será evaluado bajo la norma **ISO/IEC 25010**.
- **Recursos financieros:** presupuesto limitado financiado por los autores.





## 2.5 Suposiciones y dependencias

El desarrollo del sistema se basa en las siguientes suposiciones y dependencias:

- Se asume que la **infraestructura de red** disponible en los laboratorios de la ESPOCH será estable y suficiente para la comunicación IoT.
- Se considera que los **dispositivos IoT (ESP32 y sensores)** estarán operativos y correctamente calibrados.
- El **entorno de desarrollo** contará con compatibilidad entre las versiones de software seleccionadas (Node.js, Angular, TensorFlow, PostgreSQL).
- El sistema dependerá del acceso continuo a la base de datos y al broker MQTT para el flujo de datos en tiempo real.
- En caso de cambio en el hardware, versión de librerías o disponibilidad de red, los requisitos del sistema deberán ajustarse.

## 2.6 Evolución previsible del sistema

El diseño modular del sistema permite futuras ampliaciones y mejoras, entre las cuales se prevé:

- Integración con **plataformas de energía inteligente** o sistemas SCADA.
- Implementación de **modelos de Machine Learning avanzados** (Deep Learning o redes neuronales recurrentes).
- Desarrollo de una **aplicación móvil** para monitoreo remoto.
- Incorporación de un **sistema de mantenimiento predictivo** para anticipar fallas en sensores o paneles solares.
- Escalabilidad hacia **entornos industriales o empresariales**, extendiendo el uso más allá del ámbito académico.

# 3 Requisitos específicos

## 3.1 Requisitos comunes de los interfaces

El sistema web contará con varios tipos de interfaces que facilitarán la comunicación entre los diferentes componentes: usuarios, hardware IoT, software de análisis y bases de datos.

### 3.1.1 Interfaces de usuario

El sistema dispondrá de una **interfaz web responsiva**, desarrollada con **Angular**, accesible desde navegadores modernos (Chrome, Firefox, Edge).

Los requisitos principales son:

- La interfaz debe permitir **autenticación segura** (inicio y cierre de sesión).
- El usuario podrá acceder a un **panel principal** con visualización de datos en tiempo real provenientes de sensores IoT.
- Se incluirán **gráficas interactivas**, tablas y reportes exportables en formatos PDF o CSV.
- Los colores de la interfaz deberán seguir una paleta profesional (predominancia de tonos azules y blancos) para favorecer la legibilidad.
- El sistema debe ser **intuitivo y accesible**, de manera que un usuario con conocimientos técnicos básicos pueda operar sin entrenamiento previo.

### 3.1.2 Interfaces de hardware

El sistema se comunicará con los siguientes dispositivos de hardware:

- **Microcontroladores ESP32:** encargados de la adquisición de datos de los sensores (voltaje, corriente, temperatura e irradiancia).
- **Sensores eléctricos y ambientales:** conectados a los microcontroladores mediante pines analógicos o digitales.



El hardware deberá estar configurado para enviar los datos al servidor mediante protocolo **MQTT/MQTTS**, con un intervalo configurable (por defecto cada 60 segundos).

Cada sensor se identificará por un código único para facilitar su gestión y trazabilidad.

### 3.1.3 Interfaces de software

El sistema integrará diversos componentes de software interconectados:

- **Backend (Node.js + Express):** procesará solicitudes y gestionará los datos provenientes del broker MQTT y la base de datos.
- **Frontend (Angular):** permitirá la visualización de la información procesada.
- **Motor de Machine Learning (Python + TensorFlow):** analizará los datos e inferirá predicciones.
- **Base de datos (PostgreSQL):** almacenará lecturas históricas, configuraciones y resultados de análisis.

El intercambio entre backend y módulo de ML se realizará mediante **APIs RESTful** expuestas con **FastAPI**, utilizando el formato **JSON** para entrada y salida de datos.

### 3.1.4 Interfaces de comunicación

Protocolo: MQTT/MQTTS (seguro mediante TLS).

Broker: Eclipse Mosquitto, configurado para recibir mensajes desde los microcontroladores IoT.

Puertos utilizados: 1883 (MQTT) y 8883 (MQTTS).

Formato de mensaje: JSON con las variables eléctricas capturadas y metadatos (ID del sensor, marca temporal, unidad).

Comunicación entre software: REST/HTTP para el intercambio entre backend, frontend y servicios ML.

## 3.2 Requisitos funcionales

### 3.2.1 Requisito funcional 1

Número de requisito	RF1
Nombre de requisito	Gestión de usuarios
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Anteproyecto y alcance funcional del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

#### Descripción:

El sistema debe permitir registrar, autenticar, modificar y eliminar usuarios. Se implementarán roles diferenciados (administrador, investigador y técnico), controlando el acceso a las funciones del sistema según permisos asignados.

### 3.2.2 Requisito funcional 2

Número de requisito	RF2
Nombre de requisito	Registro de usuarios
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Alcance del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

#### Descripción:

El sistema permitirá registrar nuevos usuarios mediante un formulario que capture nombre, correo institucional, rol y contraseña, validando que el correo sea único.



### 3.2.3 Requisito funcional 3

Número de requisito	RF3
Nombre de requisito	Autenticación de usuarios
Tipo	<input type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Arquitectura backend
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

#### Descripción:

El sistema debe permitir el inicio de sesión mediante usuario y contraseña. Se implementará autenticación basada en tokens JWT y cifrado de contraseñas con bcrypt.

### 3.2.4 Requisito funcional 4

Número de requisito	RF4
Nombre de requisito	Gestión de sensores IoT
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Alcance del sistema e infraestructura IoT
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

#### Descripción:

El sistema permitirá registrar, modificar y eliminar sensores IoT, especificando tipo de medición, ubicación, frecuencia de muestreo y estado. Cada sensor tendrá un identificador único.

### 3.2.5 Requisito funcional 5

Número de requisito	RF5
Nombre de requisito	Registro de sensores
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Diseño de base de datos
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

#### Descripción:

Los administradores podrán registrar sensores especificando tipo (voltaje, corriente, irradiancia, temperatura) y asignarlos a un proyecto o ubicación.

### 3.2.6 Requisito funcional 6

Número de requisito	RF6
Nombre de requisito	Recepción de datos IoT
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Arquitectura de comunicación IoT
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

#### Descripción:

El sistema deberá recibir datos en formato JSON desde los sensores IoT a través del protocolo MQTT/MQTTS. Los datos incluirán identificador del sensor, variable, valor, unidad y marca temporal.

### 3.2.7 Requisito funcional 7

Número de requisito	RF7
Nombre de requisito	Validación de datos recibidos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Procesamiento backend
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

#### Descripción:

El sistema verificará que los datos recibidos estén completos y dentro de los



rangos permitidos antes de almacenarlos. Los datos anómalos serán descartados o marcados como inválidos.

### 3.2.8 Requisito funcional 8

Número de requisito	RF8
Nombre de requisito	Almacenamiento de datos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Requerimiento de persistencia
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

#### Descripción:

El sistema almacenará los datos válidos en una base de datos PostgreSQL. Los registros incluirán lecturas eléctricas, fecha, hora y origen del sensor, garantizando consistencia e integridad referencial.

### 3.2.9 Requisito funcional 9

Número de requisito	RF9
Nombre de requisito	Procesamiento y limpieza de datos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Módulo de análisis backend
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

#### Descripción:

El backend aplicará algoritmos de validación, normalización y limpieza sobre los datos recolectados para mejorar la calidad del conjunto de entrenamiento de Machine Learning.

### 3.2.10 Requisito funcional 10

Número de requisito	RF10
Nombre de requisito	Análisis predictivo con Machine Learning
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Objetivos del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

#### Descripción:

El sistema aplicará modelos de Machine Learning (regresión, redes neuronales, clustering) implementados en Python/TensorFlow para predecir el comportamiento de variables eléctricas.

### 3.2.11 Requisito funcional 11

Número de requisito	RF11
Nombre de requisito	Entrenamiento de modelos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Módulo ML
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

#### Descripción:

Los modelos de predicción se entrenarán con datos históricos almacenados en la base de datos y podrán ser actualizados de forma periódica.

### 3.2.12 Requisito funcional 12

Número de requisito	RF12
Nombre de requisito	Visualización de datos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Interfaz de usuario
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional



**Descripción:**

El sistema mostrará los datos eléctricos y resultados de predicciones en tiempo real mediante gráficos y paneles interactivos en la interfaz web.

### 3.2.13 Requisito funcional 13

Número de requisito	RF13
Nombre de requisito	Reportes históricos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Requerimiento funcional
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Descripción:**

Los usuarios podrán visualizar series temporales, comparar períodos de rendimiento y generar reportes de comportamiento energético.

### 3.2.14 Requisito funcional 14

Número de requisito	RF14
Nombre de requisito	Generación de alertas
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Análisis del sistema
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Descripción:**

El sistema generará alertas automáticas cuando se detecten lecturas anómalas o caídas en el rendimiento, notificando mediante correo o panel web.

### 3.2.15 Requisito funcional 15

Número de requisito	RF15
Nombre de requisito	Reportes personalizados
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Alcance funcional
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Descripción:**

Los usuarios podrán generar reportes personalizados y exportarlos en formatos PDF o CSV, incluyendo gráficos y resúmenes estadísticos.

### 3.2.16 Requisito funcional 16

Número de requisito	RF16
Nombre de requisito	Seguridad y autenticación
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Restricciones del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Descripción:**

El sistema debe garantizar la seguridad mediante autenticación JWT, cifrado de contraseñas, uso de HTTPS, y control de acceso según roles de usuario.

## 3.3 Requisitos no funcionales

### 3.3.1 Requisitos de rendimiento



### 3.3.1.1 Requisito no funcional 1

Número de requisito	RNF1
Nombre de requisito	Rendimiento del sistema
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Especificación técnica del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Descripción:**

El sistema deberá procesar datos provenientes de hasta 50 sensores IoT simultáneamente, con una latencia máxima de 2 segundos entre la emisión del dato y su visualización en el dashboard. El 95% de las transacciones deberán completarse en menos de 1 segundo.

### 3.3.1.2 Requisito no funcional 2

Número de requisito	RNF2
Nombre de requisito	Capacidad concurrente de usuarios
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Escalabilidad del sistema
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Descripción:**

El sistema debe soportar mínimo 20 usuarios conectados simultáneamente sin degradación perceptible del rendimiento ni pérdida de datos en el flujo IoT-Web.

### 3.3.1.3 Requisito no funcional 3

Número de requisito	RNF3
Nombre de requisito	Tolerancia a la carga
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Requisitos de rendimiento del backend
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Descripción:**

El sistema deberá mantener un rendimiento estable ante un incremento del 30% de la carga de datos sin necesidad de detener el servicio ni reiniciar el servidor.

## 3.3.2 Seguridad

### 3.3.2.1 Requisito no funcional 4

Número de requisito	RNF4
Nombre de requisito	Seguridad de la información
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Restricciones del sistema y norma ISO/IEC 25010
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Descripción:**

El sistema garantizará la confidencialidad, integridad y disponibilidad de los datos mediante autenticación JWT, cifrado de contraseñas con bcrypt y comunicación bajo protocolo HTTPS.

### 3.3.2.2 Requisito no funcional 5

Número de requisito	RNF5
Nombre de requisito	Registro de actividad (logs)
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Módulo de auditoría
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional



**Descripción:**

El sistema deberá mantener un registro (log) de todas las acciones críticas realizadas por los usuarios y eventos del sistema, incluyendo inicios de sesión, modificaciones de datos y errores del servidor.

### 3.3.2.3 Requisito no funcional 6

Número de requisito	RNF6
Nombre de requisito	Copias de seguridad
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Gestión de datos críticos
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Descripción:**

Se deberán realizar copias de seguridad automáticas de la base de datos cada 24 horas, almacenadas de forma cifrada en un repositorio seguro.

### 3.3.3 Fiabilidad

#### 3.3.3.1 Requisito no funcional 7

Número de requisito	RNF7
Nombre de requisito	Fiabilidad del sistema
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Arquitectura y calidad del software
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Descripción:**

El sistema deberá mantener una tasa de error inferior al 1% en la transmisión y almacenamiento de datos. En caso de fallos de conexión, el módulo IoT deberá reintentar automáticamente la transmisión de datos hasta 3 veces antes de generar una alerta.

#### 3.3.3.2 Requisito no funcional 8

Número de requisito	RNF8
Nombre de requisito	Recuperación ante fallos
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Diseño de resiliencia del sistema
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Descripción:**

En caso de interrupción del servicio, el sistema deberá reiniciar automáticamente los procesos críticos (backend, broker MQTT, base de datos) sin pérdida de información.

### 3.3.4 Disponibilidad

#### 3.3.4.1 Requisito no funcional 9

Número de requisito	RNF9
Nombre de requisito	Disponibilidad
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Alcance del sistema
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Descripción:**

El sistema deberá estar disponible al menos el 98% del tiempo mensual, excluyendo periodos de mantenimiento programado. Los tiempos de inactividad deberán ser notificados a los usuarios con 24 horas de anticipación.





### 3.3.5 Mantenibilidad

#### 3.3.5.1 Requisito no funcional 10

Número de requisito	RNF10
Nombre de requisito	Mantenibilidad del sistema
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Arquitectura modular y metodología SCRUM
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Descripción:**

El sistema debe tener una arquitectura modular (frontend, backend, ML y base de datos) que permita realizar actualizaciones o correcciones sin afectar la operación de otros módulos.

#### 3.3.5.2 Requisito no funcional 11

Número de requisito	RNF11
Nombre de requisito	Documentación técnica
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Buenas prácticas de desarrollo
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Descripción:**

Cada módulo del sistema deberá estar documentado en archivos README y manuales técnicos, detallando dependencias, endpoints, variables de entorno y procesos de despliegue.

#### 3.3.5.3 Requisito no funcional 12

Número de requisito	RNF12
Nombre de requisito	Registro de mantenimiento
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Control de calidad
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Descripción:**

Se llevará un registro histórico de las actividades de mantenimiento, actualizaciones y correcciones aplicadas al sistema, con sus fechas y responsables.

### 3.3.6 Portabilidad

#### 3.3.6.1 Requisito no funcional 13

Número de requisito	RNF13
Nombre de requisito	Portabilidad
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Restricciones tecnológicas
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Descripción:**

El sistema deberá ser multiplataforma, con capacidad de ejecución en servidores Linux, Windows o macOS, y desplegable en entornos locales o en la nube (AWS, Azure o Google Cloud).

#### 3.3.6.2 Requisito no funcional 14

Número de requisito	RNF14
Nombre de requisito	Independencia del hardware





Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Diseño técnico
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Descripción:**

El sistema no dependerá de un hardware específico; podrá ejecutarse en cualquier servidor con los requisitos mínimos de CPU de 2 núcleos, 4 GB de RAM y conectividad estable a Internet.

### 3.3.6.3 Requisito no funcional 15

Número de requisito	RNF15
Nombre de requisito	Contenerización
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Fuente del requisito	Estrategia de despliegue
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Descripción:**

El sistema deberá poder ser desplegado en contenedores Docker, garantizando la portabilidad, aislamiento y fácil configuración en diferentes entornos.

## 3.4 Otros requisitos

### 3.4.1 Otro Requisito 1

Número de requisito	OR1
Nombre de requisito	Requisitos legales y éticos
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Políticas institucionales ESPOCH
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Descripción:**

El sistema deberá cumplir con las normas de propiedad intelectual y protección de datos de la ESPOCH, garantizando la confidencialidad de la información recolectada por los sensores.

### 3.4.2 Otro Requisito 2

Número de requisito	OR2
Nombre de requisito	Requisitos ambientales
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Objetivos de sostenibilidad
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Descripción:**

El desarrollo y operación del sistema deben promover la eficiencia energética, priorizando la optimización de recursos de hardware y fomentando el uso responsable de la energía eléctrica en entornos fotovoltaicos.

### 3.4.3 Otro Requisito 3

Número de requisito	OR3
Nombre de requisito	Requisitos culturales e institucionales
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Fuente del requisito	Contexto educativo
Prioridad del requisito	<input type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input checked="" type="checkbox"/> Baja/ Opcional

**Descripción:**

El sistema deberá estar adaptado al entorno académico de la ESPOCH, utilizando lenguaje técnico en español y una interfaz accesible para usuarios con formación diversa.



## 4 Apéndices

Esta sección presenta información complementaria que apoya la comprensión, validación y futura extensión del sistema, pero que no forma parte directa de los requisitos funcionales o no funcionales del documento principal.

### 4.1 Glosario de términos

- IoT (Internet of Things): Red de dispositivos físicos interconectados capaces de recopilar, transmitir y procesar datos mediante Internet.
- Sensor: Dispositivo que detecta variaciones físicas (voltaje, corriente, irradiancia, temperatura) y las convierte en señales digitales.
- ESP32: Microcontrolador de bajo consumo con conectividad Wi-Fi y Bluetooth, utilizado para la adquisición de datos en sistemas IoT.
- MQTT (Message Queuing Telemetry Transport): Protocolo ligero de mensajería utilizado para la comunicación entre dispositivos IoT y servidores, eficiente en redes con ancho de banda limitado.
- Machine Learning (Aprendizaje Automático): Rama de la inteligencia artificial que permite a los sistemas aprender patrones a partir de datos para realizar predicciones o clasificaciones.
- TensorFlow: Biblioteca de software de código abierto desarrollada por Google para el desarrollo y entrenamiento de modelos de Machine Learning.
- Backend: Parte lógica del sistema que gestiona el procesamiento, almacenamiento y comunicación con la base de datos.
- Frontend: Interfaz de usuario del sistema web, accesible desde un navegador.
- API (Application Programming Interface): Conjunto de reglas que permite la comunicación entre diferentes componentes de software.
- SCRUM: Metodología ágil de gestión de proyectos enfocada en la entrega incremental y continua de valor.
- ISO/IEC 25010: Norma internacional que define los estándares de calidad del software en dimensiones como funcionalidad, fiabilidad, usabilidad y seguridad.

### 4.2 Referencias técnicas

Referencia	Título	Ruta / Fuente	Fec ha	Autor / Organización
[Ref. 1]	TensorFlow: Open Source Machine Learning Framework	<a href="https://www.tensorflow.org">https://www.tensorflow.org</a>	2024	Google
[Ref. 2]	MQTT v3.1.1 Specification	<a href="https://mqtt.org/">https://mqtt.org/</a>	2023	OASIS
[Ref. 3]	PostgreSQL 16 Documentation	<a href="https://www.postgresql.org/docs/">https://www.postgresql.org/docs/</a>	2024	PostgreSQL Global Dev. Group
[Ref. 4]	Node.js Documentation	<a href="https://nodejs.org/en/docs">https://nodejs.org/en/docs</a>	2024	Node.js Foundation
[Ref. 5]	Angular Developer Guide	<a href="https://angular.dev/guide">https://angular.dev/guide</a>	2024	Google Developers
[Ref. 6]	ISO/IEC 25010:2011 Software Engineering – System and Software Quality	<a href="https://www.iso.org">https://www.iso.org</a>	2024	ISO



### 4.3 Consideraciones éticas y sostenibles

- Se promueve el **uso responsable de los recursos energéticos** mediante el monitoreo eficiente de sistemas fotovoltaicos.
- Se garantiza la **confidencialidad y seguridad de los datos recolectados** por los sensores.
- El software y la documentación se liberarán bajo licencia **MIT**, fomentando la investigación abierta y colaborativa.
- El sistema contribuirá a los **Objetivos de Desarrollo Sostenible (ODS)**, específicamente al ODS 7 (Energía asequible y no contaminante) y ODS 9 (Industria, innovación e infraestructura).