Especificación de requisitos de software

Proyecto: PLC 4 uni
Revisión [99.99]

Septiembre del 2025

De la plantilla de formato del documento © & Coloriuris http://www.qualitatis.org
.

Ficha del documento

Fecha	Revisión	Autor	Verificado dep. calidad.
17/09/2025	01.00	Josue Ismael Suriel Paulino	**Carlos Antonio Pichardo Viuque**

Documento validado por las partes en fecha: [Fecha]

Por el cliente	Por la empresa suministradora
Fdo. D./ Dña [Nombre]	Fdo. D./Dña Josue Imsale Suriel Paulino

Contenido

FICHA DEL DOCUMENTO **3**

CONTENIDO **4**

1 INTRODUCCIÓN **6**

1.1 Propósito **6**

1.2 Alcance **6**

1.3 Personal involucrado **6**

1.4 Definiciones, acrónimos y abreviaturas **7**

1.5 Referencias **9**

1.6 Resumen **9**

2 DESCRIPCIÓN GENERAL **9**

2.1 Perspectiva del producto **9**

2.2 Funcionalidad del producto **9**

2.3 Características de los usuarios **10**

2.4 Restricciones **10**

2.5 Suposiciones y dependencias **10**

2.6 Evolución previsible del sistema **11**

3 REQUISITOS ESPECÍFICOS **11**

3.1 Requisitos comunes de los interfaces **12**
3.1.1 Interfaces de usuario 12
3.1.2 Interfaces de hardware 13
3.1.3 Interfaces de software 13
3.1.4 Interfaces de comunicación 13

3.2 Requisitos funcionales **15**
3.2.1 Requisito funcional 1 16
3.2.2 Requisito funcional 2 16
3.2.3 Requisito funcional 3 16
3.2.4 Requisito funcional n 16

3.3 Requisitos no funcionales **16**
3.3.1 Requisitos de rendimiento 16
3.3.2 Seguridad 16
3.3.3 Fiabilidad 17
3.3.4 Disponibilidad 18
3.3.5 Mantenibilidad 18

3.3.6 Portabilidad 18

3.4 Otros requisitos 19

4 APÉNDICES 20

1 Introducción

Esta sección proporciona una vista general de la Especificación de Requisitos de Software (SRS) para el proyecto

PLC 4 UNI. Incluye el propósito, alcance, definiciones, acrónimos, referencias y un resumen del documento.

1.1 Propósito

El propósito de este documento es definir de manera clara y completa todos los requisitos funcionales y no funcionales para el desarrollo del software (firmware y entorno de programación) del proyecto **PLC 4 UNI**.

Este documento está dirigido a:

- **Equipo de desarrollo:** Para que comprendan las especificaciones técnicas y los objetivos a cumplir.

- **Gestores del proyecto:** Para facilitar el seguimiento, la validación y las pruebas del sistema.

1.2 Alcance

El proyecto **PLC 4 UNI** consiste en el desarrollo de un sistema de control lógico programable (PLC) de bajo costo basado en el microcontrolador **ESP32-S3**. El objetivo es ofrecer una alternativa económica y accesible a los PLC comerciales, orientada principalmente a estudiantes universitarios de ingeniería, automatización y áreas afines.

El sistema permitirá a los usuarios programar lógica de control, interactuar con sensores y actuadores del mundo real y explorar conceptos de automatización e IoT (Internet de las Cosas) gracias a la conectividad nativa del ESP32-S3. Se incluirá tanto el desarrollo del firmware para el microcontrolador como una aplicación de software para que los usuarios puedan crear y cargar sus programas.

1.3 Personal involucrado

Nombre	Carlos Antonio Pichardo Viuque	Josue Ismael Suriel Paulino
Rol	Jefe de Proyecto	Desarrollador hardware/Firmware
Categoría profesional	Ingeniero Senior	Ingeniero mecatrónico
Responsabilidades	Gestión, planificación y supervisión.	Diseño y programación del firmware del ESP32-S3.
Información de contacto		
Aprobación		

1.4 Definiciones, acrónimos y abreviaturas

- **API** (**I**nterfaz de **P**rogramación de **A**plicaciones): Un conjunto de reglas y herramientas que permiten que diferentes aplicaciones de software se comuniquen entre sí.

- **CE** (**C**onformité **E**uropéenne): Marcado que indica la conformidad con las normas de salud, seguridad y protección ambiental para productos vendidos dentro del Espacio Económico Europeo.

- **CRC32** (**C**hequeo de **R**edundancia **C**íclica de 32 bits): Un tipo de función de verificación de errores utilizada para detectar cambios accidentales en datos digitales.

- **ESP-IDF** (**E**spressif **I**oT **D**evelopment **F**ramework): El kit de desarrollo de software (SDK) oficial de Espressif para programar sus microcontroladores, incluido el ESP32-S3.

- **ESP32-S3**: Un microcontrolador de bajo costo y alto rendimiento que incluye conectividad Wi-Fi y Bluetooth, siendo el cerebro del proyecto PLC 4 UNI.

- **FCC** (**F**ederal **C**ommunications **C**ommission): Agencia de los Estados Unidos que regula las comunicaciones por radio, televisión, cable y satélite.

- **Firmware**: Software que reside de forma permanente en un dispositivo de hardware, en este caso, en el microcontrolador ESP32-S3.

- **GUI** (**G**raphical **U**ser **I**nterface / Interfaz Gráfica de Usuario): Un tipo de interfaz de usuario que permite la interacción a través de elementos gráficos como ventanas, iconos y botones.

- **HTTP/HTTPS** (**H**yper**T**ext **T**ransfer **P**rotocol / **S**ecure): Protocolo para la transferencia de información en la web. HTTPS es la versión segura y cifrada.

- **I/O** (**I**nputs/**O**utputs / Entradas y Salidas): Se refiere a los terminales físicos del PLC que leen señales de sensores (entradas) y envían señales a actuadores (salidas).

- **IDE** (**I**ntegrated **D**evelopment **E**nvironment / Entorno de Desarrollo Integrado): La aplicación de software que proporciona las herramientas para escribir, compilar y cargar programas en el PLC.

- **IEEE 830**: El estándar del Instituto de Ingenieros Eléctricos y Electrónicos para la creación de Especificaciones de Requisitos de Software (SRS).

- **IoT** (**I**nternet **o**f **T**hings / Internet de las Cosas): La red de dispositivos físicos que se conectan e intercambian datos a través de internet.

- **JSON** (**J**ava**S**cript **O**bject **N**otation): Un formato de texto ligero para el intercambio de datos, utilizado en la comunicación MQTT.

- **LED** (**L**ight-**E**mitting **D**iode / Diodo Emisor de Luz): Componente electrónico utilizado como indicador visual en el hardware del PLC.

- **MQTT** (**M**essage **Q**ueuing **T**elemetry **T**ransport): Un protocolo de mensajería ligero diseñado para la comunicación entre dispositivos IoT.

- **MTBF** (**M**ean **T**ime **B**etween **F**ailures / Tiempo Medio Entre Fallos): Una métrica de fiabilidad que predice el tiempo promedio que un sistema operará antes de fallar.

- **NTP** (**N**etwork **T**ime **P**rotocol / Protocolo de Tiempo de Red): Un protocolo para sincronizar los relojes de los sistemas informáticos a través de una red.

- **OTA** (**O**ver-**T**he-**A**ir): Se refiere a la capacidad de actualizar el firmware de un dispositivo de forma inalámbrica.

- **PCB** (**P**rinted **C**ircuit **B**oard / Placa de Circuito Impreso): La base física donde se montan y conectan los componentes electrónicos del PLC.

- **PLC** (**P**rogrammable **L**ogic **C**ontroller / Controlador Lógico Programable): Un dispositivo de computación industrial robusto adaptado para el control de procesos de fabricación.

- **QoS** (**Q**uality **o**f **S**ervice / Calidad de Servicio): Un parámetro en MQTT que define el nivel de garantía de entrega de un mensaje.

- **RoHS** (**R**estriction **o**f **H**azardous **S**ubstances / Restricción de Sustancias Peligrosas): Una directiva que restringe el uso de ciertos materiales peligrosos en aparatos eléctricos y electrónicos.

- **SDK** (**S**oftware **D**evelopment **K**it / Kit de Desarrollo de Software): Un conjunto de herramientas de software y programas utilizados por los desarrolladores para crear aplicaciones para una plataforma específica.

- **SRS** (**S**oftware **R**equirements **S**pecification / Especificación de Requisitos de Software): El documento formal que describe en detalle lo que un sistema de software debe hacer.

- **USB-C**: Un tipo de conector USB reversible utilizado para la transferencia de datos y la alimentación.

- **Wi-Fi**: Una tecnología de red inalámbrica que permite a los dispositivos conectarse a internet.

1.5 Referencias

Referencia	Titulo	Ruta	Fecha	Autor
[REF-01]	ESP32-S3 Series Datasheet	espressif.com	2022	Espressif Systems

1.6 Resumen

Este documento está organizado en tres secciones principales. La Sección 1 introduce el proyecto y el propósito de este documento. La Sección 2 ofrece una descripción general del producto, sus usuarios y las restricciones del entorno. Finalmente, la

Sección 3 detalla todos los requisitos específicos, tanto funcionales como no funcionales, que el sistema PLC 4 UNI debe cumplir.

2 Descripción general

2.1 Perspectiva del producto

PLC 4 UNI es un producto independiente y autocontenido, diseñado como una herramienta educativa. Aunque es autónomo, su principal valor reside en su capacidad para interactuar con un ecosistema más amplio de componentes electrónicos estándar, como sensores, relés, motores y otros actuadores, simulando el comportamiento de un sistema de control industrial a pequeña escala.

2.2 Funcionalidad del producto

PLC 4 UNI son:

1. **Ejecución de Lógica de Control:** Interpretar y ejecutar programas creados por el usuario (por ejemplo, en lógica de escalera) para tomar decisiones basadas en las entradas.
2. **Lectura de Entradas:** Monitorear el estado de sensores digitales y analógicos conectados a sus pines de entrada.

3. **Control de Salidas:** Activar o desactivar actuadores (motores, luces, relés) conectados a sus pines de salida.
4. **Programación y Depuración:** Permitir a los usuarios escribir, cargar y depurar sus programas a través de una conexión USB.
5. **Conectividad Inalámbrica:** Utilizar el Wi-Fi integrado para la monitorización remota o el envío de datos a plataformas IoT.

2.3 Características de los usuarios

Tipo de usuario	Estudiantes Universitarios	Profesores y Educadores	Aficionados (Hobbyists)
Formación	Carreras de Ingeniería (Eléctrica, Mecatrónica, Sistemas), Física o tecnología.	Docentes en áreas técnicas.	Autodidacta en electrónica y programación.
Habilidades	Conocimientos básicos de electrónica y programación	Experiencia en la enseñanza de conceptos de automatización y control.	Interés en la automatización del hogar y proyectos DIY (Hazlo tú mismo)
Actividades	Aprender sobre automatización, desarrollar proyectos académicos, practicar lógica de control.	Utilizar el PLC como herramienta didáctica en laboratorios y clases.	Crear prototipos y pequeños sistemas automatizados personales.

2.4 Restricciones

Hardware: El sistema debe estar basado exclusivamente en el microcontrolador ESP32-S3.

Costo: El costo total de los componentes (Bill of Materials) debe ser significativamente inferior al de un PLC de gama baja comercial para asegurar que sea una alternativa viable para el sector educativo.

Software de Programación: El IDE proporcionado debe ser de código abierto o gratuito y compatible con los principales sistemas operativos (Windows, macOS, Linux).

Alimentación: El dispositivo deberá ser alimentado a través de un puerto USB-C o una fuente de alimentación externa de bajo voltaje

2.5 Suposiciones y dependencias

- Se asume que los usuarios tienen acceso a una computadora personal para programar el PLC.

- Se asume que el usuario tiene fuente para alimentar el PLC
- El proyecto depende fundamentalmente de la continua producción y disponibilidad del microcontrolador ESP32-S3 a un precio competitivo.
- Se asume que este framework seguirá recibiendo mantenimiento, será estable y no introducirá cambios drásticos que requieran una reescritura completa del código base.
- Se asume que los futuros sistemas operativos (Windows, macOS, Linux) mantendrán la compatibilidad con los drivers y las tecnologías subyacentes utilizadas por el IDE de programación. Una actualización mayor en un SO podría requerir un esfuerzo de desarrollo no planificado para mantener la funcionalidad.

2.6 Evolución previsible del sistema

Se prevén futuras mejoras que no están en el alcance de la primera versión:

Soporte para protocolos industriales: Implementación de Modbus TCP/IP para mejorar la interoperabilidad.

Expansión de I/O: Diseño de módulos apilables para aumentar el número de entradas y salidas.

IDE Avanzado: Migración a un entorno de desarrollo completamente basado en web o con capacidades de simulación.

3 Requisitos específicos

Número de requisito	RF-01
Nombre de requisito	Ejecución de Lógica de Escalera
Tipo	x☐ Requisito ☐ Restricción
Fuente del requisito	Definición del producto
Prioridad del requisito	X Alta/Esencial ☐ Media/Deseado ☐ Baja/ Opcional

***Descripción**: El firmware debe ser capaz de interpretar y ejecutar un programa de lógica de escalera cargado en su memoria. Debe realizar un ciclo de escaneo continuo (leer entradas, resolver la lógica, escribir en salidas) para controlar un proceso en tiempo real.*

Número de requisito	RF-02
Nombre de requisito	Manejo de Temporizadores (Timers)
Tipo	x☐ Requisito ☐ Restricción
Fuente del requisito	Estándar PLC (IEC 61131-3)

Prioridad del requisito	x☐ Alta/Esencial ☐ Media/Deseado ☐ Baja/ Opcional

***Descripción**: El sistema debe proveer bloques de función para temporizadores, como mínimo el **Temporizador con Retardo a la Conexión (TON)**. El usuario debe poder definir la base de tiempo (ej. milisegundos, segundos) y el valor de preselección desde el IDE*

Número de requisito	RF-03
Nombre de requisito	Manejo de Contadores (Counters)
Tipo	x☐ Requisito ☐ Restricción
Fuente del requisito	Estándar PLC (IEC 61131-3)
Prioridad del requisito	x☐ Alta/Esencial ☐ Media/Deseado ☐ Baja/ Opcional

***Descripción**: El sistema debe incluir bloques de función para contadores, como mínimo el **Contador Ascendente (CTU)**. El usuario debe poder definir el valor de preselección y asociar una entrada para el pulso de conteo*

Número de requisito	RF-04
Nombre de requisito	Lectura de Entradas Analógica
Tipo	☐x Requisito ☐ Restricción
Fuente del requisito	[Inserte aquí el texto]
Prioridad del requisito	☐ Alta/Esencial x☐ Media/Deseado ☐ Baja/ Opcional

***Descripción**: El firmware debe ser capaz de leer el valor de voltaje de las entradas analógicas y convertirlo a un valor numérico (ej. 0-1023) que pueda ser utilizado por la lógica de control para comparaciones o cálculos.*

3.1 Requisitos comunes de los interfaces

Esta sección describe detalladamente todas las entradas y salidas del sistema, definiendo cómo el producto interactuará con el usuario y con otros componentes de hardware y software.

3.1.1 Interfaces de usuario

- El IDE deberá presentar una interfaz gráfica intuitiva que permita la programación en **Lógica de Escalera (Ladder Logic)** mediante arrastrar y soltar componentes.

- El software debe proporcionar retroalimentación en tiempo real del estado de las I/O del PLC mientras está conectado.

- Deberá incluir botones claros para compilar, cargar y ejecutar el programa en el dispositivo.

3.1.2 Interfaces de hardware

Puerto de Programación: Un puerto USB-C para la carga de programas y la alimentación.

Terminales de Conexión: Bornes de tornillo para una conexión segura y fácil de cables de sensores y actuadores.

Indicadores LED: LEDs en la placa para indicar el estado de alimentación, conexión y el estado de las I/O.

3.1.3 Interfaces de software

El firmware debe soportar la conexión a redes Wi-Fi (2.4 GHz) utilizando seguridad WPA2/WPA3.

Debe ser posible enviar y recibir datos a través de Wi-Fi mediante protocolos estándar como HTTP o MQTT.

3.1.4 Interfaces de comunicación

Esta sección define todos los requisitos relacionados con las capacidades de comunicación del dispositivo, permitiendo su integración en redes locales y plataformas de Internet de las Cosas (IoT).

Conectividad Wi-Fi

Estándar y Frecuencia: El dispositivo debe ser compatible con el estándar IEEE 802.11 b/g/n y operar exclusivamente en la banda de frecuencia de 2.4 GHz. Esto maximiza la compatibilidad con la infraestructura de red existente en la mayoría de los entornos educativos y domésticos.

Seguridad de Red: Es obligatorio el soporte para los protocolos de seguridad WPA2-Personal. Adicionalmente, se implementará soporte para WPA3-Personal para asegurar la compatibilidad con redes más modernas y seguras. Los protocolos obsoletos y vulnerables como WEP o WPA no serán soportados.

Gestión de la Conexión: El firmware debe gestionar de forma autónoma el ciclo de vida de la conexión Wi-Fi. En caso de una desconexión accidental, el sistema intentará reconectarse automáticamente a la red previamente configurada en intervalos progresivos (por ejemplo, 10s, 30s, 60s) para no saturar la red. Este proceso no debe interrumpir la ejecución de la lógica de control principal del PLC.

Configuración de Credenciales: El usuario debe poder configurar el SSID (nombre de la red) y la contraseña a través de una interfaz dedicada en el IDE de escritorio. Esta información se enviará de forma segura al dispositivo a través de la conexión USB y se almacenará en la memoria no volátil del ESP32-S3.

Protocolo MQTT para Integración IoT

Para facilitar la integración nativa con plataformas de Internet de las Cosas, el firmware incluirá un cliente compatible con el protocolo MQTT v3.1.1.

Publicación de Datos (Publish): El PLC podrá enviar datos a un servidor central (broker) MQTT.

Formato de Datos: Los datos se empaquetarán en formato JSON para una fácil interpretación por parte de otros sistemas.

Tópicos (Topics): La estructura del tópico de publicación será configurable, siguiendo un patrón claro como plc4uni/<deviceID>/data, donde <deviceID> es un identificador único del dispositivo para evitar colisiones.

Calidad de Servicio (QoS): Las publicaciones se realizarán con QoS 0 (entregar como máximo una vez) para priorizar la velocidad y minimizar la sobrecarga en la red.

Suscripción a Comandos (Subscribe): El PLC podrá suscribirse a tópicos para recibir comandos de forma remota.

Tópicos (Topics): La estructura del tópico de suscripción seguirá un patrón similar, como plc4uni/<deviceID>/cmd.

Calidad de Servicio (QoS): La suscripción a comandos críticos se realizará con QoS 1 (entregar al menos una vez) para asegurar que las órdenes importantes sean recibidas por el dispositivo.

Seguridad en MQTT: Toda la comunicación con el broker MQTT deberá soportar cifrado a través de TLS/SSL, protegiendo la confidencialidad e integridad de los datos transmitidos.

Cliente HTTP/HTTPS

Peticiones HTTP/HTTPS: El firmware permitirá realizar peticiones HTTP/HTTPS GET y POST. Esto es útil para interactuar con APIs RESTful, permitiendo, por ejemplo, enviar una notificación a un servicio como IFTTT o recuperar un archivo de configuración de un servidor web.

Seguridad: El soporte para HTTPS es obligatorio para garantizar que toda la comunicación web se realice de forma segura y cifrada.

Sincronización Horaria (NTP)
Cliente NTP: El dispositivo implementará un cliente NTP (Network Time Protocol). Una vez conectado a internet, sincronizará su reloj interno con un servidor horario global.

Propósito: Esta funcionalidad es crucial para ejecutar tareas basadas en tiempo con alta precisión (ej. "activar una salida todos los días a las 8:00 AM") y para añadir timestamps exactos a los registros de eventos, lo cual es fundamental para la depuración y el análisis de datos.

3.2 Requisitos funcionales

Número de requisito	RF-01
Nombre de requisito	Ejecución de Lógica de Escalera
Descripción	El firmware deberá ser capaz de interpretar un programa en lógica de escalera, evaluando las condiciones de entrada y activando las salidas correspondientes en un ciclo de escaneo continuo.
Prioridad del requisito	Alta/Esencial

Número de requisito	RF-02
Nombre de requisito	Gestión de Entradas y Salidas Digitales
Descripción	El sistema deberá poder leer el estado (ALTO/BAJO) de al menos 8 entradas digitales y establecer el estado de al menos 8 salidas digitales.
Prioridad del requisito	Alta/Esencial

Número de requisito	RF-03
Nombre de requisito	Lectura de Entradas Analógicas
Descripción	El sistema deberá poder leer el valor de al menos 2 entradas analógicas con una resolución mínima de 10 bits.
Prioridad del requisito	Media/Deseado

Número de requisito	RF-04
Nombre de requisito	Conectividad Wi-Fi
Descripción	El usuario deberá poder configurar las credenciales de la red Wi-Fi desde el IDE y el firmware deberá poder establecer una conexión y enviar datos de telemetría a un servidor remoto.
Prioridad del requisito	Media/Deseado

3.2.1 Requisito funcional 1

Requisito funcional 1 (RF-01): Ejecución de Lógica de Escalera
Descripción: El firmware debe ser capaz de interpretar y ejecutar un programa de lógica de escalera. Realiza un ciclo de escaneo continuo: leer entradas, resolver la lógica y escribir en las salidas.

3.2.2 Requisito funcional 2

Requisito funcional 2 (RF-02): Manejo de Temporizadores (Timers)
Descripción: El sistema debe proveer bloques de función para temporizadores, como el Temporizador con Retardo a la Conexión (TON), permitiendo al usuario definir la base de tiempo y la preselección.

3.2.3 Requisito funcional 3

Requisito funcional 3 (RF-03): Manejo de Contadores (Counters)
Descripción: El sistema debe incluir bloques para contadores, como el Contador Ascendente (CTU), permitiendo al usuario definir el valor de preselección.

3.2.4 Requisito funcional 4

Requisito funcional n (RF-04): Lectura de Entradas Analógicas
Descripción: El firmware debe poder leer el voltaje de las entradas analógicas y convertirlo en un valor numérico utilizable en la lógica del programa.

3.3 Requisitos no funcionales

3.3.1 Requisitos de rendimiento

Tiempo de Ciclo de Escaneo (Scan Time): El 95% de los ciclos de escaneo, para un programa de lógica de escalera de tamaño promedio (aproximadamente 50 peldaños), deben completarse en menos de 20 milisegundos. Esto garantiza una respuesta rápida para aplicaciones de control en tiempo real.

Tiempo de Arranque (Startup Time): El sistema debe estar completamente operativo, ejecutando la lógica de control del usuario, en menos de 5 segundos desde que se le suministra energía.

Tiempo de Carga del Programa: La transferencia de un programa de tamaño promedio desde el IDE al PLC a través de la conexión USB debe completarse en menos de 30 segundos.

Latencia de Comunicación IoT: El tiempo desde que un evento ocurre en una entrada física hasta que su notificación es publicada en el broker MQTT debe ser, en promedio, inferior a 500 milisegundos bajo condiciones de red Wi-Fi óptimas.

3.3.2 Seguridad

Para el proyecto PLC 4 UNI, que combina control local con conectividad IoT, los siguientes requisitos de seguridad son fundamentales:

Comunicaciones Cifradas: Toda comunicación inalámbrica debe estar protegida. La conexión Wi-Fi deberá usar cifrado WPA2 como mínimo, y la comunicación

con el broker MQTT deberá realizarse obligatoriamente sobre TLS (Transport Layer Security) para garantizar la confidencialidad de los datos.

Carga Segura de Programa: La carga de un nuevo programa en el PLC solo será permitida a través de la conexión USB física. Se deshabilitará cualquier mecanismo de carga remota para prevenir modificaciones no autorizadas a la lógica de control.

Verificación de Integridad del Programa: Antes de ejecutar un programa recién cargado, el firmware del PLC deberá verificar una suma de comprobación (checksum), como un CRC32, que será calculada y enviada por el IDE. Si la verificación falla, el programa nuevo será descartado y se mantendrá el anterior para evitar la ejecución de código corrupto.

Aislamiento de Módulos: El núcleo del sistema que ejecuta la lógica de control en tiempo real (el ciclo de escaneo) deberá operar de forma aislada del módulo de comunicaciones de red. Un fallo o retraso en la conexión Wi-Fi o MQTT no debe interrumpir ni afectar el rendimiento del ciclo de control principal.

Registro Básico de Actividad: El sistema mantendrá un registro (log) simple en memoria de los eventos críticos, como reinicios inesperados o fallos de conexión recurrentes. Este log será accesible a través del puerto serie para facilitar el diagnóstico de problemas.

3.3.3 Fiabilidad

Para el proyecto PLC 4 UNI, la fiabilidad es clave para que sea una herramienta educativa útil y no una fuente de frustración. Los requisitos son:

Tiempo Medio Entre Fallos (MTBF): El sistema (hardware y firmware) deberá estar diseñado para un Tiempo Medio Entre Fallos (MTBF) de al menos 1000 horas de operación continua en condiciones de laboratorio (temperatura ambiente, humedad controlada). Esto asegura que el dispositivo pueda funcionar durante todo un semestre académico sin fallos de hardware o software críticos.

Recuperación de Errores de Conexión: El módulo de firmware encargado de la comunicación Wi-Fi/MQTT debe ser capaz de gestionar y recuperarse de pérdidas de conexión sin afectar al núcleo de control. La lógica de escalera del usuario debe continuar ejecutándose sin interrupciones incluso si la conexión a la red se pierde por un tiempo prolongado.

Integridad del Programa ante Fallo de Alimentación: El programa de usuario almacenado en la memoria no volátil del PLC debe ser inmune a la corrupción por fallos de alimentación inesperados. Tras un ciclo de apagado y encendido, el sistema debe cargar y ejecutar el último programa válido sin errores.

3.3.4 Disponibilidad

Para el PLC 4 UNI, es crucial que el dispositivo esté siempre listo para funcionar cuando un estudiante lo necesite en el laboratorio. Por ello, los requisitos de disponibilidad son:

Disponibilidad Operacional: El sistema deberá tener una disponibilidad operacional del 99.9% del tiempo mientras esté recibiendo alimentación eléctrica. Esto se traduce en un máximo de 8.76 horas de inactividad no planificada por año.

Paradas Planificadas: La única causa de inactividad planificada y aceptable será durante la carga de un nuevo programa desde el IDE. Durante este proceso, que no debe exceder los 30 segundos, el PLC no ejecutará la lógica de control.

Tiempo de Recuperación: Tras una parada no planificada (por ejemplo, un corte de energía), el sistema debe estar completamente disponible y operativo en menos de 5 segundos, minimizando así el tiempo de inactividad.

3.3.5 Mantenibilidad

Para el PLC 4 UNI, el enfoque de mantenimiento debe ser simple y centrado en la facilidad de actualización para un entorno educativo.

Tipo de Mantenimiento: El mantenimiento principal consistirá en actualizaciones de software (firmware y IDE). Estas actualizaciones serán de dos tipos:

Correctivas: Para solucionar errores (bugs) o vulnerabilidades de seguridad que se descubran.

Perfectivas: Para añadir nuevas funcionalidades (como nuevos bloques de lógica de escalera) o mejorar el rendimiento y la usabilidad del IDE.

Responsable del Mantenimiento:

El equipo de desarrollo será el único responsable de crear, probar y distribuir las actualizaciones de software.

El usuario final (estudiantes, profesores) será responsable de aplicar estas actualizaciones a su dispositivo y computadora, siguiendo un procedimiento simple y bien documentado.

Frecuencia del Mantenimiento: No se requiere un mantenimiento programado (semanal o mensual) por parte del usuario. Las actualizaciones se publicarán según sea necesario, en respuesta a la detección de errores críticos o al completar un ciclo de desarrollo de nuevas funcionalidades.

Mecanismo de Actualización: El IDE deberá incluir una función para notificar al usuario cuando haya una nueva versión disponible. La actualización del firmware del PLC se realizará a través de la conexión USB, utilizando una herramienta integrada en el propio IDE para garantizar un proceso simple y seguro.

3.3.6 Portabilidad

Para el proyecto PLC 4 UNI, los requisitos de portabilidad son diferentes para el software del computador (IDE) y el software del microcontrolador (firmware).

IDE de Escritorio (Alta Portabilidad)

Soporte Multiplataforma: El software del IDE deberá ser completamente funcional en los principales sistemas operativos de escritorio: Windows (10/11), macOS (últimas dos versiones) y Linux (específicamente en la última versión LTS de Ubuntu).

Tecnología de Desarrollo: Para cumplir con el requisito multiplataforma, el IDE se desarrollará utilizando un framework de desarrollo de aplicaciones de escritorio universal, como Electron (usando tecnologías web) o un lenguaje de programación con librerías de interfaz gráfica compatibles con todos los sistemas operativos objetivo (ej. Java, Python con Qt/PySide).

Firmware del PLC (Baja Portabilidad)
Dependencia del Hardware: El firmware está diseñado y optimizado específicamente para el microcontrolador ESP32-S3. No se considera un requisito su portabilidad a otras arquitecturas de microcontroladores.

Dependencia de la Plataforma: El desarrollo del firmware depende exclusivamente del ESP-IDF (Espressif IoT Development Framework) y su correspondiente cadena de herramientas. Cualquier intento de migración a otra plataforma de desarrollo requeriría una reescritura casi completa del código

3.4 Otros requisitos

Requisitos Legales
Licenciamiento de Código Abierto (Software): Todo el software desarrollado para el proyecto, incluyendo el firmware del PLC y el IDE de escritorio, deberá ser distribuido bajo una licencia de código abierto permisiva, como MIT o Apache 2.0. Esto permite su uso, modificación y distribución sin costo en el ámbito académico y comercial.

Licenciamiento de Hardware Abierto (Hardware): Los diseños de la placa de circuito impreso (esquemas, PCB) se publicarán bajo una licencia de hardware abierto, como la CERN Open Hardware Licence (CERN-OHL), para fomentar el estudio y la replicación del dispositivo.

Cumplimiento Normativo: El dispositivo de hardware final deberá cumplir con la directiva RoHS (Restriction of Hazardous Substances) para asegurar que no contiene materiales peligrosos. Además, se deberá evaluar el cumplimiento de las normativas de emisiones de radiofrecuencia (como FCC o CE) debido al uso de Wi-Fi.

Requisitos Culturales y de Usabilidad
Soporte Multi-idioma: La interfaz de usuario del IDE y toda la documentación oficial deberán estar disponibles, como mínimo, en Español e Inglés para facilitar su adopción a nivel internacional.

Accesibilidad: El diseño de la interfaz del IDE debe tener en cuenta principios de accesibilidad, como un contraste de color adecuado para usuarios con deficiencias visuales y un tamaño de fuente que sea fácil de leer.

4 Apéndices

Apéndice A: Diagrama de Arquitectura del Sistema
Este apéndice contendría un diagrama de bloques de alto nivel que ilustre la interacción entre los componentes clave: el IDE de escritorio, el PLC 4 UNI (hardware), los sensores/actuadores, la red Wi-Fi y los servicios en la nube (Broker MQTT).

Apéndice B: Casos de Uso Típicos
Aquí se describirían varios escenarios de uso prácticos para el entorno educativo. Por ejemplo:

Caso de Uso 1: Control de una secuencia de semáforos.

Caso de Uso 2: Automatización de un sistema de riego simple basado en un sensor de humedad.

Caso de Uso 3: Monitoreo de la temperatura de un laboratorio y envío de datos a una plataforma IoT cada 5 minutos.

Apéndice C: Glosario Extendido
Contendría un glosario detallado de términos técnicos relacionados con PLCs (ej. scan cycle, ladder logic, rung), microcontroladores y protocolos de comunicación utilizados en el proyecto, para apoyar a los estudiantes que no estén familiarizados con la materia.