
Especificación de requisitos de software

Proyecto: PLC 4 uni
Revisión [99.99]



Septiembre del 2025

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	7
1.6 Resumen	7
2 DESCRIPCIÓN GENERAL	8
2.1 Perspectiva del producto	8
2.2 Funcionalidad del producto	8
2.3 Características de los usuarios	8
2.4 Restricciones	9
2.5 Suposiciones y dependencias	9
2.6 Evolución previsible del sistema	9
3 REQUISITOS ESPECÍFICOS	10
3.1 Requisitos comunes de los interfaces	10
3.1.1 Interfaces de usuario	10
3.1.2 Interfaces de hardware	10
3.1.3 Interfaces de software	11
3.1.4 Interfaces de comunicación	11
3.2 Requisitos funcionales	11
3.2.1 Requisito funcional 1	12
3.2.2 Requisito funcional 2	12
3.2.3 Requisito funcional 3	12
3.2.4 Requisito funcional n	12
3.3 Requisitos no funcionales	12
3.3.1 Requisitos de rendimiento	12
3.3.2 Seguridad	12
3.3.3 Fiabilidad	12
3.3.4 Disponibilidad	12
3.3.5 Mantenibilidad	12

3.3.6	Portabilidad	12
3.4	Otros requisitos	13
4	APÉNDICES	13

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

PLC: Controlador Lógico Programable (Programmable Logic Controller). Dispositivo electrónico utilizado para la automatización de procesos electromecánicos.

ESP32-S3: Microcontrolador de bajo costo y alto rendimiento que incluye conectividad Wi-Fi y Bluetooth.

IDE: Entorno de Desarrollo Integrado (Integrated Development Environment). Aplicación de software que proporciona herramientas para el desarrollo de programas.

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

I/O: Entradas y Salidas (Inputs/Outputs).

IoT: Internet de las Cosas (Internet of Things).

1.5 Referencias

Referencia	Título	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

Esta es la sección más extensa y más importante del documento.

Debe contener una lista detallada y completa de los requisitos que debe cumplir el sistema a desarrollar. El nivel de detalle de los requisitos debe ser el suficiente para que el equipo de desarrollo pueda diseñar un sistema que satisfaga los requisitos y los encargados de las pruebas puedan determinar si éstos se satisfacen.

Los requisitos se dispondrán en forma de listas numeradas para su identificación, seguimiento, trazabilidad y validación (ej. RF 10, RF 10.1, RF 10.2,...).

Para cada requisito debe completarse la siguiente tabla:

Número de requisito	[Inserte aquí el texto]		
Nombre de requisito	[Inserte aquí el texto]		
Tipo	<input type="checkbox"/> Requisito	<input type="checkbox"/> Restricción	
Fuente del requisito	[Inserte aquí el texto]		
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional

y realizar la descripción del requisito

La distribución de los párrafos que forman este punto puede diferir del propuesto en esta plantilla, si las características del sistema aconsejan otra distribución para ofrecer mayor claridad en la exposición.

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.

[Inserte aquí el texto]

Indicar si hay que integrar el producto con otros productos de software.

Para cada producto de software debe especificarse lo siguiente:

- *Descripción del producto software utilizado*
- *Propósito del interfaz*
- *Definición del interfaz: contiendo y formato*

3.1.4 Interfaces de comunicación

[Inserte aquí el texto]

Describir los requisitos del interfaces de comunicación si hay comunicaciones con otros sistemas y cuales son las protocolos de comunicación.

3.2 Requisitos funcionales

[Inserte aquí el texto]

Definición de acciones fundamentales que debe realizar el software al recibir información, procesarla y producir resultados.

En ellas se incluye:

- *Comprobación de validez de las entradas*
- *Secuencia exacta de operaciones*
- *Respuesta a situaciones anormales (desbordamientos, comunicaciones, recuperación de errores)*
- *Parámetros*
- *Generación de salidas*
- *Relaciones entre entradas y salidas (secuencias de entradas y salidas, formulas para la conversión de información)*
- *Especificación de los requisitos lógicos para la información que será almacenada en base de datos (tipo de información, requerido)*

Las requisitos funcionales pueden ser divididos en sub-secciones.

3.2.1 Requisito funcional 1

3.2.2 Requisito funcional 2

3.2.3 Requisito funcional 3

3.2.4 Requisito funcional n

3.3 Requisitos no funcionales

3.3.1 Requisitos de rendimiento

[Inserte aquí el texto]

Especificación de los requisitos relacionados con la carga que se espera tenga que soportar el sistema. Por ejemplo, el número de terminales, el número esperado de usuarios simultáneamente conectados, número de transacciones por segundo que deberá soportar el sistema, etc.

Todos estos requisitos deben ser medibles. Por ejemplo, indicando "el 95% de las transacciones deben realizarse en menos de 1 segundo", en lugar de "los operadores no deben esperar a que se complete la transacción".

3.3.2 Seguridad

[Inserte aquí el texto]

Especificación de elementos que protegerán al software de accesos, usos y sabotajes maliciosos, así como de modificaciones o destrucciones maliciosas o accidentales. Los requisitos pueden especificar:

- *Empleo de técnicas criptográficas.*
- *Registro de ficheros con "logs" de actividad.*
- *Asignación de determinadas funcionalidades a determinados módulos.*
- *Restricciones de comunicación entre determinados módulos.*
- *Comprobaciones de integridad de información crítica.*

3.3.3 Fiabilidad

[Inserte aquí el texto]

Especificación de los factores de fiabilidad necesaria del sistema. Esto se expresa generalmente como el tiempo entre los incidentes permisibles, o el total de incidentes permisible.

3.3.4 Disponibilidad

[Inserte aquí el texto]

Especificación de los factores de disponibilidad final exigidos al sistema. Normalmente expresados en % de tiempo en los que el software tiene que mostrar disponibilidad.

3.3.5 Mantenibilidad

[Inserte aquí el texto]

Identificación del tipo de mantenimiento necesario del sistema.

Especificación de quien debe realizar las tareas de mantenimiento, por ejemplo usuarios, o un desarrollador.

Especificación de cuando debe realizarse las tareas de mantenimiento. Por ejemplo, generación de estadísticas de acceso semanales y mensuales.

3.3.6 Portabilidad

[Inserte aquí el texto]

Especificación de atributos que debe presentar el software para facilitar su traslado a otras plataformas u entornos. Pueden incluirse:

- *Porcentaje de componentes dependientes del servidor.*
- *Porcentaje de código dependiente del servidor.*
- *Uso de un determinado lenguaje por su portabilidad.*
- *Uso de un determinado compilador o plataforma de desarrollo.*
- *Uso de un determinado sistema operativo.*

3.4 Otros requisitos

[Inserte aquí el texto]

Cualquier otro requisito que no encaje en ninguna de las secciones anteriores.

Por ejemplo:

Requisitos culturales y políticos

Requisitos Legales

4 Apéndices

[Inserte aquí el texto]

Pueden contener todo tipo de información relevante para la SRS pero que, propiamente, no forme parte de la SRS.