|  |
| --- |
|  |

Especificación de requisitos de software

Proyecto: PLC4UNI

Revisión 1.0

|  |  |
| --- | --- |
|  | Septiembre |

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

.

Ficha del documento

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Revisión** | **Autor** | **Verificado dep. calidad.** |
| 19/9/2025 | 1 | Mario Melvyn Ramírez Mendoza | Ing. Carlos Pichardo |

Documento validado por las partes en fecha: 19/09/2025

|  |  |
| --- | --- |
| Por el cliente | Por la empresa suministradora |
|  |  |
| Fdo. D./ Dña Ing. Carlos Pichardo | Fdo. D./Dña Equipos de Estudiantes Mecatrónica - Proyecto: PLC for uni |

Contenido

[Ficha del documento 3](#_Toc33411057)

[Contenido 4](#_Toc33411058)

[1 Introducción 6](#_Toc33411059)

[1.1 Propósito 6](#_Toc33411060)

[1.2 Alcance 6](#_Toc33411061)

[1.3 Personal involucrado 6](#_Toc33411062)

[1.4 Definiciones, acrónimos y abreviaturas 7](#_Toc33411063)

[1.5 Referencias 7](#_Toc33411064)

[1.6 Resumen 8](#_Toc33411065)

[2 Descripción general 8](#_Toc33411066)

[2.1 Perspectiva del producto 8](#_Toc33411067)

[2.2 Funcionalidad del producto 9](#_Toc33411068)

[2.3 Características de los usuarios 9](#_Toc33411069)

[2.4 Restricciones 9](#_Toc33411070)

[2.5 Suposiciones y dependencias 9](#_Toc33411071)

[2.6 Evolución previsible del sistema 9](#_Toc33411072)

[3 Requisitos específicos 10](#_Toc33411073)

[3.1 Requisitos comunes de los interfaces 10](#_Toc33411074)

[3.1.1 Interfaces de usuario 10](#_Toc33411075)

[3.1.2 Interfaces de hardware 10](#_Toc33411076)

[3.1.3 Interfaces de software 10](#_Toc33411077)

[3.1.4 Interfaces de comunicación 10](#_Toc33411078)

[3.2 Requisitos funcionales 11](#_Toc33411079)

[3.2.1 Requisito funcional 1 11](#_Toc33411080)

[3.2.2 Requisito funcional 2 11](#_Toc33411081)

[3.2.3 Requisito funcional 3 11](#_Toc33411082)

[3.2.4 Requisito funcional n 11](#_Toc33411083)

[3.3 Requisitos no funcionales 11](#_Toc33411084)

[3.3.1 Requisitos de rendimiento 11](#_Toc33411085)

[3.3.2 Seguridad 11](#_Toc33411086)

[3.3.3 Fiabilidad 11](#_Toc33411087)

[3.3.4 Disponibilidad 12](#_Toc33411088)

[3.3.5 Mantenibilidad 12](#_Toc33411089)

[3.3.6 Portabilidad 12](#_Toc33411090)

[3.4 Otros requisitos 12](#_Toc33411091)

[4 Apéndices 12](#_Toc33411092)

# Introducción

El presente documento constituye la Especificación de Requisitos de Software (SRS) para el proyecto PLC for Uni, un controlador lógico programable diseñado con fines académicos y formativos. Este documento se elabora siguiendo el estándar IEEE 830-1998, con el objetivo de definir de manera clara, estructurada y verificable los requisitos del sistema, garantizando que el producto final cumpla con las necesidades de los usuarios y pueda ser utilizado como plataforma de aprendizaje en el área de mecatrónica, automatización y control industrial.

## Propósito

El propósito de este documento es establecer los requisitos funcionales y no funcionales que debe cumplir el PLC for Uni, asegurando que su diseño, implementación y validación respondan a los objetivos del proyecto.

Este documento servirá como referencia para:

Estudiantes y docentes universitarios, quienes utilizarán el PLC como herramienta de aprendizaje en automatización, programación en lenguajes de control y prácticas de laboratorio.

Equipo de desarrollo, quienes deberán contar con lineamientos claros para la construcción del hardware y software del sistema.

Instituciones académicas, como base para futuros proyectos de investigación, docencia y extensión en el área de control industrial.

## Alcance

El proyecto PLC for Uni abarca el diseño y desarrollo de un controlador lógico programable de carácter académico, orientado a la enseñanza universitaria en mecatrónica y automatización. El sistema incluirá hardware con entradas y salidas básicas, comunicación estándar y software de programación sencillo en lenguajes de PLC, permitiendo a estudiantes y docentes crear, cargar y ejecutar programas en tiempo real con fines didácticos. Su uso estará limitado a prácticas de laboratorio y formación académica, no para aplicaciones industriales críticas.

## Personal involucrado

|  |  |
| --- | --- |
| Nombre | Carlos Pichardo v. |
| Rol | Director del Proyecto |
| Categoría profesional | Docente universitario |
| Responsabilidades | Supervisión, validación y aprobación del proyecto |
| Información de contacto | cpichardo@itla.edu.do |
| Aprobación |  |

|  |  |
| --- | --- |
| Nombre | Mario Melvyn Ramírez Mendoza |
| Rol | Integrante del equipo desarrollador del PLC |
| Categoría profesional | Estudiante |
| Responsabilidades | Desarrollo, diseño y documentación del proyecto |
| Información de contacto | 20231253@itla.edu.do |
| Aprobación |  |

## Definiciones, acrónimos y abreviaturas

**PLC (Programmable Logic Controller / Controlador Lógico Programable**): Dispositivo electrónico diseñado para automatizar procesos mediante la programación de entradas, salidas y funciones lógicas.

**IEEE 830-1998:** Norma internacional que define la estructura y contenido recomendado para documentos de especificación de requisitos de software.

**SRS (Software Requirements Specification):** Documento que detalla los requisitos funcionales y no funcionales que debe cumplir un sistema de software.

**Ethernet:** Tecnología de red que permite la comunicación del PLC con computadoras y otros dispositivos a través de protocolos estándar.

**PLC4Uni (Programmable Logic Controller for University):** Proyecto académico de desarrollo de un controlador lógico programable diseñado como herramienta didáctica para universidades.

## Referencias

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Referencia** | **Titulo** | **Ruta** | **Fecha** | **Autor** |
| [IEEE-830] | IEEE Std 830-1998, Recommended Practice for Software Requirements Specifications | IEEE Sanders Association | 1998 | IEEE |
| [IEC-61131-3] | IEC 61131-3: Programmable Controllers – Programming Languages | [https://webstore.iec.ch](https://webstore.iec.ch/) | 2013 | IEC |
| [ESP32-TD] | Technical Reference Manual – ESP32 Series | [https://www.espressif.com](https://www.espressif.com/) | 2022 | Espressif Systems |
| [OMRON-PLC] | Introducción a los Controladores Lógicos Programables | [https://www.omron.mx](https://www.omron.mx/) | 2022 | OMRON Corporation |
| [SIEMENS-S7] | PLC Programming Guide – SIMATIC S7 Series | [https://new.siemens.com](https://new.siemens.com/) | 2021 | Siemens AG |
| Mitsubishi MELSEC PLCs |  | <https://www.mitsubishielectric.com/fa> | 2023 | Mitsubishi Electric |
| [PLC4Uni-Doc] | Documento de Proyecto Académico PLC4Uni | Universidad ITLA / Estudiantes de Mecatrónica | 2025 | Equipo de Proyecto PLC4Uni |

## Resumen

Este documento detalla la Especificación de Requisitos de Software (SRS) para el proyecto PLC4Uni, un controlador lógico programable con fines académicos. La información se organiza en cuatro secciones principales, siguiendo las directrices del estándar IEEE 830-1998, para facilitar su comprensión y consulta.

Sección 1 - Introducción: Presenta el contexto general del proyecto, definiendo el propósito del documento, el alcance del PLC, el personal involucrado, las definiciones y las referencias utilizadas.

Sección 2 - Descripción general: Ofrece una visión de alto nivel del producto. Describe su perspectiva, las funcionalidades principales, las características de los usuarios finales (estudiantes y docentes), las restricciones de diseño y las posibles evoluciones futuras del sistema.

Sección 3 - Requisitos específicos: Constituye el núcleo del documento. Aquí se detallan exhaustivamente todos los requisitos que el sistema debe cumplir. Se divide en requisitos de interfaz (usuario, hardware, software y comunicaciones), requisitos funcionales que describen las operaciones del PLC , y requisitos no funcionales que definen atributos como rendimiento, seguridad, fiabilidad y mantenibilidad.

Sección 4 - Apéndices: Contiene información complementaria que puede ser relevante para el proyecto, pero que no forma parte de los requisitos esenciales del sistema.

# Descripción general

## Perspectiva del producto

El PLC4UNI se define como un prototipo de hardware y software autónomo con fines didácticos. Su diseño y funcionamiento toman como referencia conceptual a los controladores lógicos programables de gamas de entrada de fabricantes reconocidos en la industria, como Siemens (línea LOGO! o S7-1200) y Allen-Bradley (línea Micro800).

Es fundamental aclarar que el propósito del PLC4UNI no es competir ni reemplazar a los equipos industriales certificados. Su objetivo principal es ofrecer una herramienta de aprendizaje accesible y de bajo costo que permita a los estudiantes de mecatrónica y áreas afines experimentar directamente con un dispositivo físico, cerrando la brecha entre la teoría de la automatización y la práctica.

Dentro del entorno de laboratorio del ITLA, el PLC4UNI funcionará como el núcleo de estaciones de trabajo o tableros de prueba. Se integrará en un ecosistema compuesto por:

* Un PC para la programación, carga y supervisión de la lógica de control.
* Componentes de entrada como sensores, pulsadores e interruptores.
* Componentes de salida como actuadores, contactores, luces piloto y pequeños motores..

## Funcionalidad del producto

El PLC4Uni ejecutará las siguientes funciones clave:

**Programación:** Permitirá diseñar y cargar programas de control, principalmente en lenguaje como Ladder, desde un PC.

**Ejecución:** Correrá la lógica cargada en tiempo real para leer entradas, procesar instrucciones y actualizar salidas de forma autónoma.

**Control Físico:** Interactuará con componentes de laboratorio como sensores y actuadores a través de sus puertos de entrada/salida.

**Monitoreo:** Ofrecerá visualización en línea del programa en ejecución para facilitar la depuración y el aprendizaje.

**Indicación**: Mostrará su estado operativo (encendido, en funcionamiento) mediante indicadores LED en el hardware.

## Características de los usuarios

|  |  |
| --- | --- |
| Tipo de usuario | Estudiante |
| Formación | Estudiante de nivel técnico o de ingeniería en Mecatrónica, Electrónica, Electricidad o áreas afines. |
| Habilidades | Conocimientos teóricos básicos de electricidad y lógica de programación. Típicamente, con poca experiencia práctica en la programación de PLCs industriales. |
| Actividades | Realizar prácticas de laboratorio, programar secuencias de control, conectar sensores/actuadores al PLC y depurar sus propios programas con supervisión. |

|  |  |
| --- | --- |
| Tipo de usuario | Usuario supervisor y preparador del entorno de aprendizaje (Docente). |
| Formación | Ingeniero, tecnólogo o técnico superior con experiencia profesional o académica en automatización y control industrial. |
| Habilidades | Dominio de lenguajes de programación de PLC y experiencia en el diseño de material didáctico y prácticas de laboratorio. |
| Actividades | Configurar los equipos, crear y proponer ejercicios, guiar a los estudiantes durante las prácticas, evaluar los resultados y realizar el mantenimiento básico del PLC. |

## Restricciones

**Restricciones de Hardware**

**Plataforma de Control:** El firmware del PLC se desarrollará sobre un microcontrolador específico de bajo costo y alto rendimiento, preferiblemente la plataforma ESP32, aprovechando su conectividad y capacidad de procesamiento.

**Componentes Electrónicos:** Se utilizarán componentes electrónicos de grado comercial y de propósito general, no de grado industrial. Esto se debe a restricciones presupuestarias y al enfoque educativo del dispositivo.

**Entradas y Salidas (I/O):** El prototipo contará con un número limitado de entradas y salidas digitales. No se contempla el manejo de I/O de alta velocidad o módulos de expansión complejos en esta primera versión.

**Restricciones de Entorno y Normativa**

**Entorno Operativo:** El PLC4Uni está diseñado exclusivamente para uso en entornos de laboratorio y con fines didácticos. No debe ser utilizado en aplicaciones industriales, procesos críticos o en entornos con altas exigencias de ruido electromagnético, vibración o temperaturas extremas.

## Suposiciones y dependencias

**Dependencias de Hardware y Suministro**

Disponibilidad del Microcontrolador: Se asume la disponibilidad continua en el mercado del microcontrolador ESP32 y sus módulos de desarrollo a un costo accesible. Una interrupción en su suministro o un cambio drástico en su precio podría forzar un rediseño del hardware.

**Compatibilidad de Componentes:** El proyecto depende de que los componentes electrónicos seleccionados (relés, optoacopladores, fuentes de alimentación, etc.) sean compatibles entre sí y se mantengan disponibles para su adquisición.

**Infraestructura del Laboratorio:** Se supone que los laboratorios del ITLA cuentan con una infraestructura eléctrica estable y con bancadas de trabajo adecuadas para la conexión y prueba segura de los PLC y sus periféricos.

**Dependencias de Software y Tecnológicas**

**Estabilidad del Entorno de Desarrollo (SDK):** El desarrollo del firmware depende de las herramientas de software proporcionadas por el fabricante del microcontrolador (ej. ESP-IDF de Espressif y el framework de ESP32). Se asume que estas herramientas se mantendrán estables, funcionales y compatibles con los sistemas operativos de desarrollo.

**Permanencia del Sistema Operativo**: Se asume que los computadores de los laboratorios continuarán utilizando Windows 10 o una versión superior, y que futuras actualizaciones de este sistema operativo no generarán incompatibilidades críticas con el software de programación del PLC.

**Dependencias Institucionales y Académicas**

Integración Curricular: Se asume que el PLC4Uni será integrado en los planes de estudio y guías de laboratorio de al menos una de las asignaturas de la carrera de Mecatrónica, garantizando su uso y relevancia.

## Evolución previsible del sistema

Soporte para Entradas y Salidas Analógicas: Integrar puertos para leer señales analógicas (ej. sensores de temperatura o presión de 0-10V) y generar salidas analógicas (ej. para controlar la velocidad de un motor), permitiendo prácticas de control de procesos más avanzadas.

Módulos de Expansión: Diseñar un bus de comunicación que permita conectar módulos de expansión, como tarjetas con más entradas/salidas digitales, relés de mayor potencia o puertos de comunicación adicionales.

Soporte para más Lenguajes (IEC 61131-3): Ampliar el software de programación para incluir otros lenguajes estándar como el Diagrama de Bloques de Funciones (FBD) y el Texto Estructurado (ST).

Interfaz de Programación Web: Aprovechar la conectividad Wi-Fi del ESP32 para alojar un servidor web en el propio PLC. Esto permitiría a los usuarios programar y monitorear el dispositivo desde cualquier navegador web, sin necesidad de instalar software en el PC.

Integración con HMI/SCADA: Añadir la capacidad de comunicarse con software de Interfaz Hombre-Máquina (HMI) o sistemas SCADA básicos, introduciendo a los estudiantes en los conceptos de supervisión y control de procesos.

Considerar la obtención de certificaciones básicas, tanto educativas como industriales, que habiliten su aplicación en entornos formativos de mayor exigencia técnica.

# 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,...).

## Requisitos comunes de los interfaces

Descripción detallada de todas las entradas y salidas del sistema de software.

### Interfaces de usuario

Describir los requisitos del interfaz de usuario para el producto. Esto puede estar en la forma de descripciones del texto o pantallas del interfaz. Por ejemplo posiblemente el cliente ha especificado el estilo y los colores del producto. Describa exacto cómo el producto aparecerá a su usuario previsto.

### Interfaces de hardware

Especificar las características lógicas para cada interfaz entre el producto y los componentes de hardware del sistema. Se incluirán características de configuración.

### Interfaces de software

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

### Interfaces de comunicación

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

## Requisitos funcionales

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.

### Requisito funcional 1

### Requisito funcional 2

### Requisito funcional 3

### Requisito funcional n

## Requisitos no funcionales

### Requisitos de rendimiento

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 mesurables. 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”.

### Seguridad

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RNF-02 | | |
| Nombre de requisito | Protección de Integridad del PLC | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Director del proyecto | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

El sistema debe protegerse contra las dos amenazas más comunes en un entorno de laboratorio: la corrupción del software y el daño por errores de cableado. Para ello: 1. El PLC verificará la integridad del firmware (mediante checksum) antes de una actualización para evitar quedar inoperativo. 2. El hardware implementará aislamiento galvánico en todas sus entradas y salidas para proteger el circuito de control de sobretensiones o cortocircuitos accidentales.

### Fiabilidad

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RNF-03 | | |
| Nombre de requisito | Operación Continua en Sesión de Laboratorio | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Equipo de desarrollo | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

El PLC debe ser capaz de operar de forma ininterrumpida y sin bloqueos durante una sesión de laboratorio extendida (mínimo 8 horas) de uso continuo. En caso de un fallo de software imprevisto, el temporizador watchdog del microcontrolador forzará un reinicio automático. Crucialmente, el PLC deberá retener el último programa cargado por el usuario en su memoria no volátil, permitiendo reanudar la práctica sin necesidad de volver a cargar el código.

### Disponibilidad

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RNF-04 | | |
| Nombre de requisito | Arranque Seguro y Rápido | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Director del proyecto | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

El sistema debe estar listo para operar en menos de 10 segundos desde que se conecta a la alimentación. Como medida de seguridad fundamental en un entorno de aprendizaje, al arrancar o tras un reinicio, todas las salidas físicas deben inicializarse en un estado seguro (apagado), para evitar la activación accidental de actuadores.

### Mantenibilidad

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RNF-05 | | |
| Nombre de requisito | Diagnóstico y Extensibilidad para Fines Académicos | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Equipo de desarrollo / Fines didácticos. | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

El diseño del firmware debe ser modular y estar bien documentado, permitiendo que futuros estudiantes puedan entender y extender sus funcionalidades como parte de otros proyectos. El hardware debe incluir puntos de prueba (test points) claramente etiquetados para señales clave (ej. alimentación, comunicación) y un puerto de depuración accesible para facilitar el diagnóstico y el estudio de su funcionamiento interno.

### Portabilidad

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RNF-06 | | |
| Nombre de requisito | Portabilidad de la Aplicación de Programación | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Director del proyecto / Equipo de desarrollo | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

Aunque la versión inicial de la aplicación de programación se desarrollará para Windows, su arquitectura interna debe estar basada en tecnologías multiplataforma (ej. un framework como Qt, Electron o JavaFX). El objetivo es que, en el futuro, generar una versión compatible con otros sistemas operativos como macOS o Linux no requiera una reescritura completa del software, sino una simple recompilación y adaptación menor.

## Otros requisitos

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | OR-01 | | |
| Nombre de requisito | Licenciamiento y Carácter Abierto | | |
| Tipo | Requisito | Restricción | |
| Fuente del requisito | Director del proyecto / Equipo de desarrollo | | |
| Prioridad del requisito | Alta/Esencial | Media/Deseado | Baja/ Opcional |

Siendo un proyecto de carácter académico, todos los artefactos generados (esquemas, código fuente, diseños de PCB y documentación) deberán ser publicados bajo una licencia de código abierto y hardware libre (ej. MIT, Creative Commons BY-SA). Esto fomenta la colaboración y el uso del proyecto como una herramienta de aprendizaje para toda la comunidad. Toda la documentación principal deberá estar en español.

# Apéndices

Director del proyecto: Ing. Carlos Pichardo V.

Integrante del equipo de desarrollo: Mario Melvyn Ramírez Mendoza -20231253

*Espressif Systems. (2024). ESP-IDF programming guide. Retrieved September 22, 2025, from* [*https://docs.espressif.com/projects/esp-idf/en/latest/esp32/*](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/)

*OpenPLC Project. (2024). OpenPLC Project. Retrieved September 22, 2025, from* [*https://www.openplcproject.com/*](https://www.openplcproject.com/)

*PLC Mentor. (2024). PLC Mentor. Retrieved September 22, 2025, from* [*https://www.plcmentor.com/*](https://www.plcmentor.com/)

*RealPars. (2024). RealPars. Retrieved September 22, 2025, from* [*https://realpars.com/*](https://realpars.com/)

*Santos, R., & Santos, S. (2024). Random Nerd Tutorials. Retrieved September 22, 2025, from* [*https://randomnerdtutorials.com/*](https://randomnerdtutorials.com/)

*SoapBox Automation Inc. (2024). SoapBox Snap: The open-source industrial controller. Retrieved September 22, 2025, from* [*https://www.soapboxsnap.com/*](https://www.google.com/search?q=https://www.soapboxsnap.com/)