|  |
| --- |
|  |

Especificación de requisitos de software

Proyecto:

Revisión

Logotipo, nombre de la empresa

El contenido generado por IA puede ser incorrecto.

|  |  |  |
| --- | --- | --- |
|  |  |  |



Ficha del documento

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Revisión** | **Autor** | **Verificado dep. calidad.** |
| **18-9-25** | **1.0** | **Omar Junior Ramos Rivera** | **Carlos Antonio Pichardo Viuque** |

Documento validado por las partes en fecha:

|  |  |
| --- | --- |
| Por el cliente | Por la empresa suministradora |
|  |  |
| Fdo. D./ Dña | Fdo. D./Dña |

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 6](#_Toc33411063)

[1.5 Referencias 6](#_Toc33411064)

[1.6 Resumen 6](#_Toc33411065)

[2 Descripción general 7](#_Toc33411066)

[2.1 Perspectiva del producto 7](#_Toc33411067)

[2.2 Funcionalidad del producto 7](#_Toc33411068)

[2.3 Características de los usuarios 7](#_Toc33411069)

[2.4 Restricciones 7](#_Toc33411070)

[2.5 Suposiciones y dependencias 7](#_Toc33411071)

[2.6 Evolución previsible del sistema 7](#_Toc33411072)

[3 Requisitos específicos 7](#_Toc33411073)

[3.1 Requisitos comunes de los interfaces 8](#_Toc33411074)

[3.1.1 Interfaces de usuario 8](#_Toc33411075)

[3.1.2 Interfaces de hardware 8](#_Toc33411076)

[3.1.3 Interfaces de software 8](#_Toc33411077)

[3.1.4 Interfaces de comunicación 8](#_Toc33411078)

[3.2 Requisitos funcionales 8](#_Toc33411079)

[3.2.1 Requisito funcional 1 9](#_Toc33411080)

[3.2.2 Requisito funcional 2 9](#_Toc33411081)

[3.2.3 Requisito funcional 3 9](#_Toc33411082)

[3.2.4 Requisito funcional n 9](#_Toc33411083)

[3.3 Requisitos no funcionales 9](#_Toc33411084)

[3.3.1 Requisitos de rendimiento 9](#_Toc33411085)

[3.3.2 Seguridad 9](#_Toc33411086)

[3.3.3 Fiabilidad 9](#_Toc33411087)

[3.3.4 Disponibilidad 9](#_Toc33411088)

[3.3.5 Mantenibilidad 10](#_Toc33411089)

[3.3.6 Portabilidad 10](#_Toc33411090)

[3.4 Otros requisitos 10](#_Toc33411091)

[4 Apéndices 10](#_Toc33411092)

# Introducción

El proyecto nace para solucionar la brecha de aprendizaje práctico que enfrentan los estudiantes de ingeniería y áreas técnicas, quienes a menudo carecen de acceso a costosos equipos industriales de automatización. PLC4Uni ofrecerá un entorno virtual seguro e intuitivo donde los estudiantes podrán diseñar, probar y depurar programas de control en lenguaje Ladder, desarrollando así habilidades prácticas y fundamentales en automatización industrial sin la necesidad de hardware físico.

## Propósito

## El propósito de este documento es definir y especificar todos los requisitos de software para el proyecto PLC4Uni. Servirá como guía principal para el equipo de desarrollo, el personal de pruebas y todas las partes interesadas (stakeholders) para asegurar que el producto final cumpla con los objetivos planteados

## . Alcance

El alcance del proyecto es el diseño y la fabricación de un Controlador Lógico Programable (PLC) físico, basado en el microcontrolador ESP32.

Una vez completado, este hardware será una herramienta educativa que permitirá a los usuarios crear, cargar y depurar programas de control utilizando lenguajes de programación estándar para PLC, como el Diagrama de Escalera (Ladder). El dispositivo simulará el comportamiento de un PLC industrial, permitiendo el control de procesos reales a través de sus entradas y salidas físicas, temporizadores y contadores.

**El hardware a desarrollar incluirá:**

* Una unidad central de procesamiento (ESP32) para ejecutar la lógica de control.
* Circuitos de entrada para leer señales de sensores y pulsadores.
* Circuitos de salida para accionar relés que controlan actuadores.
* Compatibilidad garantizada con plataformas de software de código abierto (como OpenPLC) para su programación.

**Fuera del alcance del proyecto está:**

* El desarrollo del software de programación para PC (IDE), ya que se utilizará una solución de código abierto existente.
* La fabricación en masa del dispositivo; el alcance se limita al diseño y la creación de prototipos funcionales.

## Personal involucrado

|  |  |
| --- | --- |
| Nombre | Pichardo, Omar |
| Rol | Líder del proyecto, equipo de desarrollo |
| Categoría profesional | Docente, estudiantes |
| Responsabilidades | - Definir la visión del producto, supervisar y validar requisitos.  - Coordinar al equipo de desarrollo y gestionar el proyecto. |
| Información de contacto | [cpichardo@itla.edu.do](mailto:cpichardo@itla.edu.do)  [20230846@itla.edu.do](mailto:20230846@itla.edu.do) |
| Aprobación | aprobado |

## Definiciones, acrónimos y abreviaturas

 PLC**:** Controlador Lógico Programable (Programmable Logic Controller).

 SRS**:** Especificación de Requisitos de Software (Software Requirements Specification).

 LAD **/ LD:** Diagrama de Escalera (Ladder Diagram).

 FBD**:** Diagrama de Bloques de Funciones (Function Block Diagram).

 UI**:** Interfaz de Usuario (User Interfaz).

 E**/S (I/O):** Entradas y Salidas (Inputs/Outputs).

## Referencias

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Referencia** | **Titulo** | **Ruta** | **Fecha** | **Autor** |
| [1] | IEEE Std 830-1998 | Estándar de la industria | 1998 | IEEE |
|  |  |  |  |  |

## Resumen

Este documento está organizado de la siguiente manera: La **Sección 2** proporciona una descripción general del producto, su propósito, sus usuarios y sus restricciones. La **Sección 3** detalla los requisitos específicos, tanto funcionales como no funcionales. Finalmente, la **Sección 4** contiene los apéndices.

# Descripción general

El propósito del proyecto **PLC4Uni** es diseñar y desarrollar una herramienta de software educativo, gratuita y accesible, que simule el funcionamiento de un Controlador Lógico Programable (PLC). El objetivo es solucionar la brecha de aprendizaje práctico que enfrentan los estudiantes de ingeniería y áreas técnicas, quienes a menudo carecen de acceso a costosos equipos industriales. PLC4Uni ofrecerá un entorno virtual seguro e intuitivo donde los estudiantes podrán diseñar y probar programas de control, desarrollando así habilidades prácticas en automatización industrial sin la necesidad de hardware físico.

## Perspectiva del producto

## El PLC4Uni es un dispositivo de hardware autónomo diseñado para la enseñanza de la automatización. Funciona como un PLC de nivel básico, que se programa desde una PC a través de un IDE de código abierto. Su principal diferenciador es el uso del ESP32, lo que le confiere una gran potencia de procesamiento y capacidades de comunicación inalámbrica.

## Funcionalidad del producto

El hardware del PLC deberá realizar las siguientes funciones principales:

* Leer el estado de entradas digitales optoacopladas.
* Controlar salidas de relé para activar actuadores externos.
* Ejecutar un programa de lógica de control cargado en su memoria.
* Comunicarse con una PC para recibir dicho programa.
* Proporcionar indicación visual clara del estado de las E/S.

## Características de los usuarios

|  |  |
| --- | --- |
| Tipo de usuario | **Estudiante, Profesor** |
| Formación | **Estudiante:** Cursando grado en ingeniería (eléctrica, mecatrónica, industrial), tecnología o formación técnica.  **Profesor:** Docente con experiencia en áreas de automatización y control. |
| Habilidades | Conocimientos básicos de electricidad y circuitos. • Capacidad para conectar cables a terminales de tornillo de forma segura. • Manejo de software en un PC para usar el entorno de programación. |
| Actividades | **Estudiante:** Conectar sensores y actuadores, cargar programas en el PLC, realizar prácticas de laboratorio y probar la lógica de control. • **Profesor:** Diseñar y proponer ejercicios prácticos, demostrar el funcionamiento del PLC y evaluar los proyectos de los estudiantes. |

## Restricciones

 Microcontrolador **Obligatorio:** El diseño debe estar centrado obligatoriamente en el microcontrolador **ESP32**.

 Compatibilidad **de Software:** El hardware debe ser diseñado para ser compatible con el runtime de OpenPLC.

 Costo**:** El costo total de los componentes por unidad debe mantenerse bajo para ser accesible en un entorno educativo.

 Seguridad**:** El diseño debe priorizar la seguridad del usuario, con aislamiento galvánico entre el MCU y las E/S.

 Alimentación**:** El dispositivo debe operar con una fuente de alimentación estándar de 24V DC.

## Suposiciones y dependencias

* Se asume la disponibilidad comercial de los módulos ESP32 y demás componentes electrónicos.
* El proyecto depende de la existencia y estabilidad de runtimes de PLC de código abierto que sean compatibles con la plataforma ESP32.

### 2.6 Evolución previsible del sistema

 Revisión **2:** Creación de módulos de expansión de E/S que se conecten al PLC principal.

 Revisión **3:** Integración de una pequeña pantalla LCD en la carcasa para diagnóstico y monitoreo.

 Revisión **4:** Aprovechar las capacidades Wi-Fi/Bluetooth del ESP32 para programación inalámbrica o monitoreo remoto (SCADA).

# Requisitos específicos

Esta sección desglosa en detalle todos los requisitos que el hardware del PLC4Uni debe cumplir. Se divide en los requisitos de las interfaces externas, los requisitos funcionales (lo que el hardware debe hacer) y los requisitos no funcionales (los atributos de calidad como la seguridad y el rendimiento).

|  |  |  |  |
| --- | --- | --- | --- |
| Número de requisito | RF-01 | | |
| Nombre de requisito | Procesamiento de Entradas Digitales | | |
| Tipo | ☑ Requisito | ☐ Restricción | |
| Fuente del requisito | Equipo de Proyecto / Función esencial de un PLC | | |
| Prioridad del requisito | ☑ Alta/Esencial | ☐ Media/Deseado | ☐ Baja/Opcional |

## Requisitos comunes de los interfaces

Esta subsección describe todos los puntos de conexión y de interacción que el PLC4Uni tendrá con el mundo exterior, incluyendo cómo se conecta con el usuario, con los sensores/actuadores del proceso a controlar y con el software de programación.

### Interfaces de usuario

La "interfaz de usuario" de este dispositivo de hardware se refiere a todos los componentes físicos que permiten a una persona entender su estado de un vistazo. No es una pantalla, sino un conjunto de indicadores visuales directos. Los requisitos son:

* **Indicadores LED de Estado:** El PLC debe incluir un conjunto de LEDs claramente visibles para informar sobre su estado operativo:
* **PWR (Power):** Un LED de color verde que permanecerá encendido de forma fija siempre que el dispositivo reciba alimentación correctamente.
* **RUN (Running):** Un LED de color ámbar que parpadeará para indicar que el microcontrolador (ESP32) está ejecutando el programa de lógica cargado.
* **Indicadores LED de Entradas y Salidas (E/S):** Cada terminal de entrada y salida debe tener un LED asociado para mostrar su estado lógico en tiempo real:
* **LEDs de Entrada:** 8 LEDs (ej. color amarillo) que se iluminarán cuando se detecte una señal de 24V en el borne de entrada correspondiente.
* **LEDs de Salida:** 6 LEDs (ej. color rojo) que se iluminarán cuando el programa active el relé de la salida correspondiente.
* **Etiquetado Claro:** Toda la serigrafía en la carcasa y/o el PCB debe ser clara, legible y utilizar identificadores estándar (ej. I1, I2... para entradas; O1, O2... para salidas) para evitar confusiones al realizar el cableado.

### Interfaces de hardware

Las interfaces de hardware para la conexión con el proceso a controlar serán:

* **Terminales de Entrada:** 8 bornes de tornillo para conectar sensores.
* **Terminales de Salida:** 8 bornes de tornillo para conectar actuadores.

### Interfaces de software

La interfaz para la carga de software (el programa de lógica) será:

* **Puerto de Programación:** Un conector USB-C que permita la comunicación directa con el IDE en la PC para cargar el programa en el ESP32.

### Interfaces de comunicación

 La principal interfaz de comunicación para programación es el puerto **USB-C**.

 El hardware debe exponer las capacidades **Wi-Fi y Bluetooth** del ESP32 en su diseño de PCB para un posible uso futuro.

## Requisitos funcionales

### Requisito funcional 1

* **Procesamiento de Entradas**

Descripción: El hardware debe leer 8 entradas de 24V DC, proteger el ESP32 mediante optoacopladores y entregar una señal lógica limpia (3.3V) a los pines GPIO del MCU.

Prioridad: Alta

### Requisito funcional 2

* **Accionamiento de Salidas**

Descripción: El hardware debe permitir que 6 pines GPIO del ESP32 controlen 6 relés, proporcionando salidas de contacto seco para manejar cargas externas.

Prioridad: Alta

### Requisito funcional 3

* **Ejecución de Firmware**

Descripción: El ESP32 debe ser capaz de arrancar y ejecutar de manera continua un firmware de PLC (runtime) almacenado en su memoria flash.

Prioridad: Alta

### Requisito funcional 4

* **Regulación de Voltaje**

Descripción: El circuito debe tomar una entrada de 24V DC y generar los voltajes regulados y estables necesarios para el ESP32 (3.3V) y otros componentes lógicos.

Prioridad: Alta

### Requisito funcional 5

* **Carga de Programa**

Descripción: El diseño debe incluir el circuito de auto-reseteo y arranque para que el ESP32 pueda ser programado directamente desde el IDE a través del USB-C.

Prioridad: Alta

## Requisitos no funcionales

### Requisitos de rendimiento

* El tiempo de respuesta total del hardware (desde que una entrada se activa hasta que la salida correspondiente reacciona) debe ser, en promedio, inferior a **15 milisegundos**

### Seguridad

* Debe existir un **aislamiento galvánico completo** (usando optoacopladores para entradas y relés para salidas) entre el circuito del ESP32 y todas las E/S.

### Fiabilidad

* El diseño debe utilizar componentes de grado industrial o de alta fiabilidad para garantizar una larga vida útil en un entorno de laboratorio.

### Disponibilidad

* El hardware debe estar diseñado para operación continua (24/7) sin sobrecalentamiento.

### Mantenibilidad

* Se deben utilizar zócalos para los componentes principales (como los optoacopladores) para facilitar su reemplazo en caso de falla.

### Portabilidad

* El peso total del dispositivo, incluyendo la carcasa, no debe superar los **500 gramos**.

## Otros requisitos

* **Documentación:** Se debe entregar un diagrama esquemático completo y los archivos de diseño del PCB.

# Apéndices

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