
Especificación de requisitos de software

Proyecto: PLC 4 uni
Revisión [99.99]



Septiembre del 2025

Instrucciones para el uso de este formato

Este formato es una plantilla tipo para documentos de requisitos del software.

Está basado y es conforme con el estándar IEEE Std 830-1998.

Las secciones que no se consideren aplicables al sistema descrito podrán de forma justificada indicarse como no aplicables (NA).

Notas:

Los textos en color azul son indicaciones que deben eliminarse y, en su caso, sustituirse por los contenidos descritos en cada apartado.

Los textos entre corchetes del tipo “[Inserte aquí el texto]” permiten la inclusión directa de texto con el color y estilo adecuado a la sección, al pulsar sobre ellos con el puntero del ratón.

Los títulos y subtítulos de cada apartado están definidos como estilos de MS Word, de forma que su numeración consecutiva se genera automáticamente según se trate de estilos “Titulo1, Titulo2 y Titulo3”.

La sangría de los textos dentro de cada apartado se genera automáticamente al pulsar Intro al final de la línea de título. (Estilos Normal indentado1, Normal indentado 2 y Normal indentado 3).

El índice del documento es una tabla de contenido que MS Word actualiza tomando como criterio los títulos del documento.

Una vez terminada su redacción debe indicarse a Word que actualice todo su contenido para reflejar el contenido definitivo.

Ficha del documento

Fecha	Revisión	Autor	Verificado dep. calidad.
11/09/2025	01.00	PLC4UNI	Carlos Antonio Pichardo Viuque

Documento validado por las partes en fecha: [Fecha]

Por el cliente	Por la empresa suministradora
Carlos Pichardo Viuque	Empresa de desarrollo PLC4UNI

Contenido

FICHA DEL DOCUMENTO	3
CONTENIDO	4
1 INTRODUCCIÓN	6
1.1 Propósito	6
1.2 Alcance	6
1.3 Personal involucrado	7
1.4 Definiciones, acrónimos y abreviaturas	7
1.5 Referencias	8
1.6 Resumen	8
2 DESCRIPCIÓN GENERAL	8
2.1 Perspectiva del producto	8
2.2 Funcionalidad del producto	8
2.3 Características de los usuarios	9
2.4 Restricciones	9
2.5 Suposiciones y dependencias	10
2.6 Evolución previsible del sistema	10
3 REQUISITOS ESPECÍFICOS	10
3.1 Requisitos comunes de los interfaces	12
3.1.1 Interfaces de usuario	12
3.1.2 Interfaces de hardware	12
3.1.3 Interfaces de software	12
3.1.4 Interfaces de comunicación	13
3.2 Requisitos funcionales	13
3.2.1 Requisito funcional 1	¡Error! Marcador no definido.
3.2.2 Requisito funcional 2	¡Error! Marcador no definido.
3.2.3 Requisito funcional 3	¡Error! Marcador no definido.
3.2.4 Requisito funcional n	¡Error! Marcador no definido.
3.3 Requisitos no funcionales	15
3.3.1 Requisitos de rendimiento	15
3.3.2 Seguridad	16
3.3.3 Fiabilidad	¡Error! Marcador no definido.
3.3.4 Disponibilidad	16
3.3.5 Mantenibilidad	17

3.3.6 Portabilidad	¡Error! Marcador no definido.
3.4 Otros requisitos	17
4 APÉNDICES	18

1 Introducción

Este documento ofrece una visión general de la **Especificación de Requisitos de Software (SRS)** para el proyecto **PLC 4 UNI**.

Describe su finalidad, el alcance, los términos clave, las abreviaturas, las fuentes de información y un resumen del contenido.

1.1 Propósito

El objetivo principal de este documento es detallar de forma exhaustiva los **requisitos funcionales y no funcionales** necesarios para crear el software (firmware y la interfaz de programación) del proyecto PLC 4 UNI.

Este documento es una guía esencial para:

- **El equipo de desarrollo:** Para que puedan entender a fondo las especificaciones técnicas y los objetivos del proyecto.
- **Los gestores del proyecto:** Para que puedan supervisar el progreso, validar los resultados y realizar las pruebas del sistema de manera efectiva.

1.2 Alcance

El proyecto **PLC 4 UNI** se enfoca en crear un **controlador lógico programable (PLC)** **económico** usando el microcontrolador **ESP32-S3**. Su meta es ser una alternativa a los PLC comerciales, especialmente para **estudiantes universitarios** de ingeniería, automatización y campos relacionados.

Este sistema permitirá a los usuarios programar sus propias lógicas de control, interactuar con sensores y actuadores, y experimentar con conceptos de **automatización e IoT**, aprovechando la conectividad del ESP32-S3. El proyecto incluye el desarrollo del **firmware** para el microcontrolador y una **aplicación de software** para que los usuarios puedan crear y cargar sus programas.

[Inserte aquí el texto]

- *Identificación del producto(s) a desarrollar mediante un nombre*
- *Consistencia con definiciones similares de documentos de mayor nivel (ej. Descripción del sistema) que puedan existir*

1.3 Personal involucrado

Nombre	Carlos Antonio Pichardo Viuque	
Rol		
Categoría profesional		
Responsabilidades		
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.
- **Automatización**
Conjunto de técnicas y herramientas que permiten operar procesos de manera automática, minimizando la intervención humana.
- **Entradas (Inputs)**
Señales recibidas por el PLC provenientes de sensores, pulsadores, interruptores u otros dispositivos que informan sobre el estado del proceso.
- **Salidas (Outputs)**
Señales enviadas desde el PLC hacia actuadores, motores, válvulas, lámparas, alarmas u otros dispositivos de control.
- **CPU (Unidad Central de Procesamiento)**
El “cerebro” del PLC, encargado de ejecutar el programa cargado y coordinar el procesamiento de las señales de entrada y salida.
- **Memoria del PLC**
Lugar donde se almacenan los programas de control y los datos temporales que utiliza la CPU para el procesamiento.
- **Lenguajes de Programación del PLC**
Conjunto de normas y representaciones gráficas o textuales usadas para programar. Los más comunes son:
 - **Diagrama de Escalera (Ladder Diagram, LD)**
 - **Lista de Instrucciones (IL)**
 - **Bloques de Funciones (FBD)**
 - **Texto Estructurado (ST)**
 - **Diagrama de Secuencias (SFC)**

- **Sensor**
Dispositivo que convierte una variable física (temperatura, presión, nivel, proximidad, etc.) en una señal eléctrica interpretable por el PLC.
- **Actuador**
Elemento que ejecuta la acción de control (por ejemplo: motores, cilindros neumáticos, contactores, relés).
- **Ciclo de Escaneo (Scan Time)**
Proceso repetitivo que realiza el PLC para leer entradas, ejecutar el programa, actualizar salidas y volver a comenzar.
- **HMI (Interfaz Hombre-Máquina)**
Medio de comunicación entre el operador y el PLC, generalmente a través de pantallas táctiles o software de supervisión.
- **Redes Industriales**
Protocolos de comunicación que permiten la interconexión entre PLCs, HMIs y otros dispositivos (ejemplo: Modbus, Profibus, Ethernet/IP).

1.5 Referencias

Referencia	Titulo	Ruta	Fecha	Autor
[REF-01]	PLC4UNI	espressif.com	202	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

El **PLC 4 UNI** es un dispositivo autónomo e independiente, concebido como una herramienta de apoyo educativo. Si bien funciona de manera individual, su mayor aporte se encuentra en la posibilidad de integrarse con un conjunto más amplio de componentes electrónicos convencionales —como sensores, relés, motores y actuadores— permitiendo así la simulación de un sistema de control industrial a escala reducida

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

D	Nombre del Requisito	Tipo	Fuente del Requisito	Prioridad	Descripción del Requisito
RF-001	Ejecución de Lógica de Control	Requisito	Funcionalidad del Producto	Alta/Essencial	<i>El firmware debe ser capaz de interpretar y ejecutar el programa de lógica de control cargado por el usuario (e.g., Lógica de Escalera, C++), escaneando las entradas, ejecutando el programa y actualizando las salidas de</i>

					<i>manera cíclica.</i>
<i>RF-002</i>	<i>Carga y Flasheo de Firmware</i>	Requisito	Jefe de Proyecto	Alta/Essencial	<i>El software IDE debe poder compilar el código del usuario y cargar el firmware resultante al microcontrolador ESP32-S3 a través de la conexión USB/Serial, verificando la integridad de la carga.</i>
		Requisito	Profesores	Alta/Essencial	<i>El firmware debe transmitir el estado actual de todas las Entradas/Salidas (I/O), así como el valor de las variables internas, a la computadora a través del puerto Serial para su depuración.</i>
<i>RF-004</i>	<i>Configuración Wi-Fi</i>	Requisito	Funcionalidad del Producto	Media/Deseado	<i>El usuario debe poder configurar las credenciales Wi-Fi (SSID y Contraseña) a través de una interfaz web integrada (Access Point) o por comandos seriales, para utilizar la conectividad IoT.</i>
<i>RF-005</i>	<i>Interacción con Actuadores Digitales</i>	Requisito	Funcionalidad del Producto	Alta/Essencial	<i>El PLC debe activar/desactivar el voltaje en sus pines de salida digital (I/O) basándose en la lógica de control del programa del usuario.</i>
<i>RF-006</i>	<i>Lectura de Sensores Analógicos</i>	Requisito	Funcionalidad del Producto	Media/Deseado	<i>El PLC debe poder leer y digitalizar las señales de los pines de entrada analógica (ADC) y poner estos valores a</i>

					disposición de la lógica del usuario.
--	--	--	--	--	--

3.1 Requisitos comunes de los interfaces

[Esta sección proporciona una descripción detallada de las entradas y salidas del sistema de software (IDE y Firmware del PLC 4 uni) y de su interacción con el entorno.]

3.1.1 Interfaces de usuario

El producto PLC 4 uni tendrá dos interfaces de usuario principales para atender las necesidades de programación y monitorización:]

1. Interfaz del IDE de Programación (Escritorio):

- RF-IUE.01: El IDE de escritorio debe proporcionar un área de edición de código con **resaltado de sintaxis** y numeración de líneas para el lenguaje de programación soportado.
- RF-IUE.02: Debe incluir una consola de salida o área de mensajes que muestre los resultados del proceso de compilación, incluyendo errores y advertencias.
- RF-IUE.03: Debe tener una opción visible para seleccionar el puerto USB/Serial de la computadora al que está conectado el PLC.
- Interfaz de Monitorización Serial o Web (Firmware):
- RF-IUE.04: El firmware debe enviar mensajes de monitorización y depuración a través del puerto Serial (visible en un Monitor Serial) en un formato legible.
- RF-IUE.05: Si se habilita la conectividad Wi-Fi (RF-COM.03), el firmware debe alojar una página web simple que muestre el estado actual de las I/O y la versión del firmware instalado.

3.1.2 Interfaces de hardware

[Especificación de las características lógicas que rigen la interacción entre el software (IDE/Firmware) y los componentes de hardware del sistema.]

RF-IHH.01 (Puerto USB/Serial): El software IDE debe ser capaz de inicializar y comunicarse con el chip USB-Serial del ESP32-S3 para la carga de código y la monitorización de depuración.

RF-IHH.02 (Pines I/O): El firmware debe poder leer las señales de los pines de entrada digital/análogica y manipular las señales de los pines de salida digital del microcontrolador según la lógica del programa del usuario.

RF-IHH.03 (LED de Estado): El firmware debe hacer parpadear un LED de estado (por ejemplo, el LED on-board del ESP32) para indicar que el ciclo de scan del programa de control se está ejecutando correctamente

3.1.3 Interfaces de software

[Indicar los productos de software de terceros con los que el sistema se integrará y el propósito de dicha integración.]

Software: Compilador GCC para ESP32-S3

Propósito del Interfaz: Traducir el código fuente del usuario (C/C++ o el lenguaje de control) a un archivo binario ejecutable para el microcontrolador.

RF-IHS.01: El IDE debe integrar y utilizar el toolchain de compilación recomendado por Espressif Systems (e.g., ESP-IDF) para generar firmware compatible.

Software: Sistema Operativo (Windows/macOS/Linux)

Propósito del Interfaz: Permitir la ejecución del IDE de escritorio y la comunicación a través del puerto Serial/USB.

RF-IHS.02: El IDE debe utilizar las librerías nativas del sistema operativo para enumerar y abrir los puertos de comunicación Serial disponibles en la PC del usuario

3.1.4 Interfaces de comunicación

[Descripción de los requisitos de los interfaces de comunicación y los protocolos utilizados.]

RF-COM.01 (Protocolo de Carga): El IDE debe utilizar el protocolo Serial de carga de bootloader nativo del ESP32-S3 para el proceso de flasheo del firmware.

RF-COM.02 (Monitorización Serial): La comunicación de depuración entre el firmware y el IDE (o Monitor Serial) debe usar el protocolo UART (Universal Asynchronous Receiver-Transmitter) a una velocidad estándar (e.g., 115200 baudios).

RF-COM.03 (Wi-Fi): El firmware debe incluir librerías que permitan la conectividad TCP/IP (Wi-Fi) para enviar o recibir datos a través de protocolos como MQTT o HTTP a plataformas externas de IoT..

3.2 Requisitos funcionales

[Definición de las acciones fundamentales que debe realizar el software (IDE y Firmware) para el proyecto PLC 4 uni. Estas acciones incluyen la comprobación de validez de las entradas, la secuencia exacta de operaciones y la generación de salidas.]

Los requisitos funcionales (RF) se presentan en la siguiente tabla, donde cada uno detalla una capacidad específica del sistema, su fuente y su prioridad

ID	Nombre del Requisito	Tipo	Fuente del Requisito	Prioridad	Descripción del Requisito
RF-001	Ejecución de Lógica de Control	Requisito	Funcionalidad del Producto	Alta/Essencial	El firmware debe ser capaz de interpretar y ejecutar el programa de lógica de control

					cargado por el usuario (e.g., Lógica de Escalera, C++) de forma cíclica (scan), leyendo entradas y actualizando salidas.
RF-002	Carga y Flasheo de Firmware	Requisito	Jefe de Proyecto	Alta/Esencial	El software IDE debe poder compilar el código del usuario y cargar el firmware resultante al microcontrolador ESP32-S3 a través de la conexión USB/Serial, garantizando la integridad de la carga.
RF-003	Monitorización en Tiempo Real (Serial)	Requisito	Profesores	Alta/Esencial	El firmware debe transmitir el estado actual de todas las Entradas/Salidas (I/O) y las variables internas relevantes a la computadora a través del puerto Serial para propósitos de depuración.
RF-004	Configuración Wi-Fi	Requisito	Funcionalidad del Producto	Media/Deseado	El usuario debe poder configurar las credenciales Wi-Fi (SSID y Contraseña) mediante una interfaz de configuración (Access Point web o comandos seriales) para habilitar la conectividad IoT.
RF-005	Interacción con Actuadores Digitales	Requisito	Funcionalidad del Producto	Alta/Esencial	El PLC debe activar o desactivar el voltaje en sus pines de salida digital (I/O) basándose en el resultado de la

					lógica de control del programa del usuario.
RF-006	Lectura de Sensores Analógicos	Requisito	Funcionalidad del Producto	Media/Deseado	El PLC debe poder leer y digitalizar las señales de los pines de entrada analógica (ADC) y poner estos valores convertidos a disposición de la lógica de control del usuario.
RF-007	Manejo de Tiempos y Retardos (Timers)	Requisito	Funcionalidad del Producto	Alta/Essencial	El firmware debe proporcionar funciones de temporización (Timers) y conteo (Counters) precisas y estables que puedan ser utilizadas por la lógica del programa de control.
RF-008	Generación de Reporte de Compilación	Requisito	Estudiantes	Media/Deseado	El IDE debe generar un informe detallado post-compilación que muestre el uso de memoria (RAM y Flash) del firmware compilado en el ESP32-S3.

3.3 Requisitos no funcionales

Los Requisitos No Funcionales (RNF) definen las restricciones de calidad, rendimiento y diseño que el sistema PLC educativo debe cumplir.

A continuación, se presentan las tablas con los RNF, rellenando las subsecciones de la plantilla del documento ERS.

3.3.1 Requisitos de rendimiento

ID	Requisito de Rendimiento	Prioridad	Métrica de Aceptación
RNF-001	Latencia del Ciclo de Scan	Alta/Essencial	El tiempo de ciclo de ejecución de la lógica de control del firmware no debe exceder de

			100 milisegundos.
RNF-002	Uso de Memoria RAM	Media/Deseado	El firmware base (sin la lógica de usuario) no debe consumir más del 20% de la memoria RAM total disponible del ESP32-S3.
RNF-003	Velocidad de Carga del IDE	Media/Deseado	El IDE de programación de escritorio debe iniciarse y estar listo para la edición en menos de 5 segundos.

3.3.2 Seguridad

ID	Requisito de Seguridad	Prioridad	Métrica de Aceptación
RNF-007	Protección de Código Fuente	Media/Deseado	El firmware debe implementar la función de encriptación de flash del ESP32-S3 para proteger el código del programa de usuario contra la extracción o copia no autorizada.
RNF-008	Acceso a la Configuración Wi-Fi	Media/Deseado	La interfaz de configuración Wi-Fi (si es web) debe estar protegida con una contraseña predeterminada que el usuario pueda modificar inmediatamente.
RNF-009	Aislamiento de I/O	Alta/Esencial	La electrónica debe garantizar que un cortocircuito en las terminales de I/O no dañe permanentemente el microcontrolador ESP32-S3.

3.3.3 Disponibilidad

ID	Requisito de Disponibilidad	Prioridad	Métrica de Aceptación

RNF-010	Disponibilidad del Firmware	Alta/Essencial	El firmware del PLC 4 UNI debe mostrar una disponibilidad del 99.9% durante su uso continuo en sesiones de laboratorio.
---------	-----------------------------	----------------	---

3.3.4 Mantenibilidad

ID	Requisito de Mantenibilidad	Prioridad	Métrica de Aceptación
RNF-011	Facilidad de Actualización del Firmware	Media/Deseado	El IDE debe incluir una función sencilla de "Actualización de Firmware Base" que permita al usuario final aplicar correcciones o mejoras sin perder su código de control.
RNF-012	Registro de Eventos (Logging)	Media/Deseado	El firmware debe emitir mensajes de error o advertencia específicos (e.g., "Error de conexión Wi-Fi") al puerto Serial para facilitar el mantenimiento correctivo

3.4 Otros requisitos

ID	Requisito de Portabilidad	Prioridad	Métrica de Aceptación
RNF-013	Compatibilidad con SO	Alta/Essencial	El IDE debe ser completamente funcional en las últimas tres versiones de Windows, macOS y la versión LTS de Ubuntu/Debian.
RNF-014	Lenguaje de Programación del IDE	Media/Deseado	El IDE debe ser desarrollado utilizando un lenguaje (framework) que asegure la portabilidad multiplataforma con un

			solo código base (e.g., Python/Tkinter, Java/Swing, Electron).
--	--	--	--

ID	Requisito Legal/Cultural	Prioridad	Descripción
RNF-015	Licencia de Código Abierto	Alta/Esencial	El código fuente del proyecto (firmware e IDE) debe ser liberado bajo una licencia permisiva de código abierto (e.g., MIT) para permitir su uso, modificación y distribución en el ámbito educativo.
RNF-016	Documentación en Español	Alta/Esencial	Toda la documentación técnica, manuales de usuario y comentarios del código deben estar escritos en español para uso exclusivo en la Universidad.

4 Apéndices

[Esta sección contiene información complementaria y de soporte que ayuda a la comprensión de los requisitos especificados en el documento. No forma parte de la especificación normativa.]

Glosario de Términos Específicos

Definición de términos técnicos o jerga utilizada específicamente en el ámbito de la automatización y el hardware del proyecto.

Término	Definición
BOM	Bill of Materials (Lista de Materiales). Documento que enumera todos los componentes, subcomponentes y piezas de la placa de hardware del PLC, junto con sus costos y proveedores.
Scan Time	Tiempo que tarda el PLC en ejecutar un ciclo completo de su programa de control: lectura de entradas, ejecución de lógica y actualización de salidas. Este tiempo es una métrica crítica de rendimiento.
GPIO	General Purpose Input/Output (Entrada/Salida de Propósito General). Se refiere a los pines físicos del microcontrolador ESP32-S3 que pueden ser configurados como entradas o salidas digitales/analógicas.
ADC	Analog-to-Digital Converter (Convertidor

	Analógico-Digital). Circuito integrado en el ESP32-S3 que permite leer voltajes analógicos de sensores y convertirlos en valores digitales para su procesamiento.
Bootloader	Programa de bajo nivel que se ejecuta al inicio del ESP32-S3 y es responsable de cargar el firmware principal desde la memoria flash a la RAM para su ejecución.

El sistema central ("PLC 4 uni Firmware/IDE") interactúa con cuatro entidades principales:

Programador (Estudiante/Profesor): Interactúa con el IDE de Escritorio para la escritura y carga de código. (Interfaz Serial/USB).

Actuadores/Sensores: Interactúa con los componentes físicos (Bombas, LEDs, Relés) a través de los Pines GPIO (Interfaz de Hardware).

Servidor IoT/Nube: Interactúa con plataformas externas para el envío de datos a través de Wi-Fi (Interfaz de Comunicación).

Base de Código del ESP-IDF: Es el framework que provee las funciones básicas para interactuar con el hardware (Interfaz de Software).