

---

## Especificación de requisitos de software

Proyecto: [Nombre del proyecto]  
Revisión [99.99]

---

[Mes de año]



**INSTITUTO TECNOLÓGICO DE LAS AMÉRICAS (ITLA)**

Mecatrónica

**MANUAL DE USUARIO**

**PLC 4 UNI**

**Controlador Lógico Programable Educativo IoT**



**Modelo:** P4U-ESP32-S3-V1 **Revisión del Documento:** 1.0 **Fecha:** Septiembre 2025

**Desarrollado por:** equipo PLC4UNI

## CAPITULO I. PUESTA EN MARCHA DEL PROYECTO

## 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.
17/09/2025	17/09/2025	Franelis Luciano	Carlos Pichardo (Gerente de Proyecto)

Documento validado por las partes en fecha: 18/09/2025

Por el cliente	Por la empresa suministradora
Instituto Tecnologico de Las Americas (ITLA)	Equipo de Desarrollo
Profesor Carlos Antonio Pichardo	SR. Franelis Luciano

# Contenido

<b>FICHA DEL DOCUMENTO</b>	<b>5</b>
<b>CONTENIDO</b>	<b>6</b>
<b>1 INTRODUCCIÓN</b>	<b>8</b>
<b>1.1 Propósito</b>	<b>9</b>
<b>1.2 Alcance</b>	<b>9</b>
<b>1.3 Personal involucrado</b>	<b>10</b>
<b>1.4 Definiciones, acrónimos y abreviaturas</b>	<b>10</b>
<b>1.5 Referencias</b>	<b>11</b>
<b>1.6 Resumen</b>	<b>12</b>
<b>2 DESCRIPCIÓN GENERAL</b>	<b>13</b>
<b>2.1 Perspectiva del producto</b>	<b>13</b>
<b>2.2 Funcionalidad del producto</b>	<b>14</b>
<b>2.3 Características de los usuarios</b>	<b>15</b>
<b>2.4 Restricciones</b>	<b>16</b>
<b>2.5 Suposiciones y dependencias</b>	<b>17</b>
<b>2.6 Evolución previsible del sistema</b>	<b>18</b>
<b>3 REQUISITOS ESPECÍFICOS</b>	<b>19</b>
<b>3.1 Requisitos comunes de los interfaces</b>	<b>19</b>
3.1.1 Interfaces de usuario	19
3.1.2 Interfaces de hardware	20
3.1.3 Interfaces de software	21
3.1.4 Interfaces de comunicación	21
<b>3.2 Requisitos funcionales</b>	<b>22</b>
3.2.1 Requisito funcional 1	22
3.2.2 Requisito funcional 2	26
3.2.3 Requisito funcional 3	28
3.2.4 Requisito funcional n	¡Error! Marcador no definido.
<b>3.3 Requisitos no funcionales</b>	<b>29</b>
3.3.1 Requisitos de rendimiento	29
3.3.2 Seguridad	30

3.3.3	Fiabilidad	30
3.3.4	Disponibilidad	31
3.3.5	Mantenibilidad	31
3.3.6	Portabilidad	32
<b>3.4</b>	<b>Otros requisitos</b>	<b>32</b>
<b>4</b>	<b>APÉNDICES</b>	<b>70</b>

## 1 Introducción

En el ámbito de la educación tecnológica contemporánea, la formación en sistemas de automatización y control constituye un eje estratégico para preparar a los futuros profesionales que habrán de desenvolverse en industrias cada vez más digitalizadas y competitivas. Sin embargo, el acceso a los Controladores Lógicos Programables (PLC), herramienta fundamental en el mundo de la automatización industrial, continúa siendo limitado en el espacio académico, ya sea por su elevado costo, su orientación eminentemente productiva o la falta de modelos específicamente diseñados para fines pedagógicos.

El proyecto **PLC4UNI** surge como respuesta a esta necesidad, con la intención de diseñar y desarrollar un PLC orientado a la educación superior. Su propósito no se restringe únicamente a suplir la ausencia de un dispositivo didáctico en el mercado, sino que se proyecta como una iniciativa transformadora que busca integrar la teoría con la práctica, fomentar la innovación en el aula y ofrecer un recurso que acerque al estudiante al entorno industrial real sin sacrificar la accesibilidad ni la simplicidad de uso.

En este sentido, **PLC4UNI** no pretende ser una mera réplica económica de un PLC comercial, sino un dispositivo concebido con criterios pedagógicos, capaz de adaptarse a diferentes niveles de enseñanza y de promover el aprendizaje significativo. El trasfondo del proyecto radica en comprender que la educación en automatización no debe limitarse al entrenamiento técnico, sino que debe estimular la capacidad crítica, la resolución de problemas y la creatividad de los futuros ingenieros y tecnólogos.

Así, la creación de un PLC para la educación universitaria se plantea como una apuesta estratégica para cerrar la brecha entre la academia y la industria, dotando a los estudiantes de un recurso práctico que no solo refuerce sus competencias técnicas, sino que también les permita experimentar con escenarios de innovación, investigación aplicada y desarrollo de proyectos interdisciplinarios.

[Inserción aquí el texto]

*La introducción de la Especificación de requisitos de software (SRS) debe proporcionar una vista general de la SRS. Debe incluir el objetivo, el alcance, las definiciones y acrónimos, las referencias, y la vista general del SRS.*

## 1.1 Propósito

Este documento, denominado Especificación de Requisitos de Software (SRS), tiene como propósito fundamental definir de manera clara, exhaustiva y estructurada los requisitos funcionales y no funcionales del sistema **PLC4UNI**. El objetivo central de este proyecto es concebir y desarrollar un Controlador Lógico Programable (PLC) de código abierto, accesible y pedagógicamente orientado, capaz de responder a las necesidades de enseñanza y aprendizaje en el ámbito universitario de la automatización y el control.

La intención de este documento no se limita únicamente a establecer lineamientos técnicos, sino que también busca sentar las bases para una innovación educativa que trascienda la simple replicación de dispositivos industriales. Se persigue un enfoque transformador, donde el PLC se convierta en un instrumento de integración entre la teoría y la práctica, permitiendo que los estudiantes experimenten con un recurso auténtico, adaptado a su nivel de formación, y que al mismo tiempo constituya un puente hacia el mundo profesional.

Este documento está dirigido a los stakeholders clave del proyecto, entre ellos:

- El equipo de desarrollo, encargado de materializar las especificaciones técnicas.
- Los docentes e investigadores, quienes encontrarán en el PLC4UNI un recurso versátil para el diseño de prácticas, proyectos y entornos de simulación aplicados.
- Las autoridades universitarias y gestores académicos, que podrán evaluar el impacto del proyecto en la calidad de la enseñanza y en la preparación de los futuros profesionales.

En este sentido, el SRS de **PLC4UNI** no solo servirá como base contractual y guía técnica para el diseño, implementación y validación del sistema, sino también como una declaración de intenciones estratégicas que refuerza el compromiso de la universidad con la innovación, la accesibilidad tecnológica y la formación integral de sus estudiantes.

[Inserte aquí el texto]

- *Propósito del documento*
- *Audiencia a la que va dirigido*

## 1.2 Alcance

El proyecto PLC4UNI tiene como alcance el diseño, desarrollo e implementación de un Controlador Lógico Programable (PLC) de código abierto y bajo costo, concebido específicamente para fines educativos en el nivel universitario. A diferencia de los PLC comerciales, cuyo diseño responde a criterios de eficiencia industrial y aplicaciones de producción en masa, este sistema busca priorizar la funcionalidad didáctica, la simplicidad operativa y la adaptabilidad curricular.

El sistema abarcará las siguientes dimensiones:

- Entorno de hardware: un dispositivo compacto, robusto y modular que permita la conexión de entradas y salidas digitales y analógicas, facilitando la experimentación con sensores, actuadores y pequeños sistemas de automatización.
- Entorno de software: una plataforma de programación accesible e intuitiva, con soporte para lenguajes estándar de automatización (por ejemplo, diagrama de escalera y texto estructurado), además de interfaces gráficas que permitan la comprensión gradual de los conceptos por parte del estudiante.
- Material didáctico complementario: guías de uso, manuales y prácticas integradas que orienten el aprendizaje autónomo y colaborativo, reforzando el vínculo entre la teoría y la práctica.

En términos de cobertura, el alcance del sistema se centra en el contexto educativo y no en la sustitución de los PLC industriales en ambientes productivos. No obstante, se

busca garantizar que el estudiante adquiera competencias que puedan trasladarse sin fricciones al manejo de controladores comerciales, logrando un aprendizaje significativo y transferible.

De manera estratégica, el proyecto busca:

1. Reducir la brecha entre el acceso académico y la realidad industrial en materia de automatización.
2. Proveer a las universidades de una herramienta de formación práctica sostenible, que pueda actualizarse y escalarse en función de las necesidades curriculares.
3. Fomentar la investigación aplicada y el desarrollo de proyectos interdisciplinarios, aprovechando la apertura y flexibilidad del sistema.

En síntesis, el alcance de PLC4UNI se define no solo por la entrega de un dispositivo físico y su software asociado, sino también por la creación de un ecosistema educativo integral que promueva la innovación, la equidad en el acceso tecnológico y la formación de competencias críticas para el futuro profesional de los estudiantes.

[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	Gerente de Proyecto
Categoría profesional	Profesor de Diseño para mecatronico
Responsabilidades	Tomar decisiones
Información de contacto	cpichardo@itla.edu.do
Aprobación	[Inserte aquí el texto]

*Relación de personas involucradas en el desarrollo del sistema, con información de contacto. Esta información es útil para que el gestor del proyecto pueda localizar a todos los participantes y recabar la información necesaria para la obtención de requisitos, validaciones de seguimiento, etc.*

## 1.4 Definiciones, acrónimos y abreviaturas

Con el fin de garantizar la claridad conceptual en la interpretación de este documento, se definen a continuación los términos, acrónimos y abreviaturas más relevantes en el contexto del proyecto PLC4UNI:

- **PLC (Controlador Lógico Programable):** Dispositivo electrónico de control utilizado en la automatización de procesos industriales, diseñado para recibir señales de entrada, procesarlas mediante un programa lógico y generar señales de salida que actúan sobre dispositivos externos.
- **PLC4UNI:** Nombre asignado al proyecto de creación de un PLC didáctico, de código abierto y bajo costo, orientado a la educación universitaria.
- **SRS (Software Requirements Specification):** Documento de Especificación de Requisitos de Software, basado en la norma IEEE Std 830-1998, cuyo propósito es definir de manera detallada y estructurada los requisitos del sistema a desarrollar.
- **Stakeholder:** Cualquier persona, grupo u organización que tiene interés en el desarrollo, implementación o resultado del sistema. En el caso de este proyecto, incluye profesores, estudiantes, investigadores, desarrolladores y autoridades universitarias.

- **Lenguaje Ladder (LD):** Lenguaje de programación gráfico estándar utilizado en PLCs, basado en esquemas que emulan diagramas eléctricos de control por contactos.
- **Texto Estructurado (ST):** Lenguaje de programación textual utilizado en PLCs, similar a lenguajes de alto nivel, que facilita la implementación de algoritmos más complejos.
- **Open Source (Código abierto):** Modelo de desarrollo en el cual el software y, en algunos casos, el hardware, se liberan bajo licencias que permiten su estudio, modificación y redistribución.
- **Automatización educativa:** Enfoque pedagógico que integra tecnologías de automatización y control en procesos de enseñanza-aprendizaje, con el fin de fortalecer competencias prácticas y teóricas en el área de la ingeniería.
- **IEEE Std 830-1998:** Estándar internacional que establece las directrices para la elaboración de documentos de especificación de requisitos de software.
- **ESP32:** Microcontrolador con Wi-Fi y Bluetooth integrados, base del hardware del sistema.
- **I/O:** Input/Output (Entradas y salidas digitales o analógicas).
- **ADC:** Analog-to-Digital Converter (convertidor analógico-digital).
- **HMI:** Human Machine Interface (Interfaz hombre-máquina).
- **Modbus RTU/TCP:** Protocolos de comunicación industrial.
- **RS-485:** Bus serial diferencial para comunicaciones industriales.
- **BOM:** Bill of Materials (lista de materiales)

[Inserte aquí el texto]

*Definición de todos los términos, abreviaturas y acrónimos necesarios para interpretar apropiadamente este documento. En ella se pueden indicar referencias a uno o más apéndices, o a otros documentos.*

## 1.5 Referencias

Referencia	Título	Ruta	Fecha	Autor
[Ref.]	[Título]	[Ruta]	[El texto]	[Autor]
IEEE Std 830-1998.	<i>IEEE Recommended Practice for Software Requirements Specification</i>	Institute of Electrical and Electronics Engineers	1998	IEEE
IEC 61131-3	<i>Programmable Controllers – Part 3: Programming Languages</i>	International Electrotechnical Commission	2013	IEC
González & Sánchez (2019)	Automatización industrial y control programable	Editorial Alfaomega, México	2019	González, A. & Sánchez, R.
Siemens Documentation	PLC Programming Basics	Siemens AG Documentation Center	2021	Siemens AG

Rockwell Technical Paper	Introduction to Programmable Logic Controllers	Rockwell Automation Technical White Paper	2020	Rockwell Automation
OpenPLC Project	Open Source PLC	<a href="http://www.openplcproject.com">http://www.openplcproject.com</a>	2025	OpenPLC Project
Qualitatis Template	Plantilla de especificación de requisitos de software conforme al estándar IEEE 830	<a href="http://www.qualitatis.org">http://www.qualitatis.org</a>	-	Qualitatis.org

*Relación completa de todos los documentos relacionados en la especificación de requisitos de software, identificando de cada documento el título, referencia (si procede), fecha y organización que lo proporciona.*

## 1.6 Resumen

El presente documento constituye la **Especificación de Requisitos de Software (SRS)** para el proyecto **PLC4UNI**. En él se establecen los fundamentos conceptuales, los objetivos, el alcance y los actores involucrados en el desarrollo del sistema, además de detallar los requisitos funcionales y no funcionales que guiarán la implementación del mismo.

El documento se encuentra organizado de la siguiente manera:

- **Capítulo 1 – Introducción:** Presenta el propósito del documento, el alcance del sistema, el personal involucrado, las definiciones y referencias clave, así como un panorama general del proyecto.
- **Capítulo 2 – Descripción general:** Ofrece una visión global del producto, incluyendo su perspectiva, funcionalidades principales, características de los usuarios, restricciones, suposiciones y posibles evoluciones del sistema.
- **Capítulo 3 – Requisitos específicos:** Detalla los requisitos funcionales y no funcionales del sistema, organizados de manera estructurada y con trazabilidad, para orientar el desarrollo técnico y las pruebas de validación.
- **Capítulo 4 – Apéndices:** Contiene información complementaria que respalda el contenido principal, como referencias técnicas adicionales, diagramas, ejemplos de casos de uso y consideraciones académicas.

De esta forma, el documento asegura una estructura lógica y progresiva, que permite a los distintos stakeholders comprender, validar y dar seguimiento al desarrollo del proyecto **PLC4UNI**, garantizando tanto la claridad técnica como la coherencia pedagógica del mismo.

[Inserte aquí el texto]

- *Descripción del contenido del resto del documento*
- *Explicación de la organización del documento*

## 2 Descripción general

### 2.1 Perspectiva del producto

El sistema PLC4UNI se concibe como un producto **independiente** diseñado específicamente para el ámbito educativo universitario, pero con una arquitectura que facilita su **integración** en ecosistemas de aprendizaje y laboratorios más amplios. Desde su naturaleza física —un controlador compacto y modular con entradas/salidas— hasta su componente lógico —firmware abierto y un entorno de programación accesible—, PLC4UNI actúa como el núcleo práctico donde convergen la experimentación, la enseñanza y la investigación aplicada en automatización y control.

Operativamente, PLC4UNI puede desplegarse en dos modos principales: (a) **modo autónomo**, donde el dispositivo funciona conectado directamente a sensores y actuadores en una bancada de prácticas y es programado desde un ordenador local; y (b) **modo integrado**, en el que el PLC forma parte de un conjunto mayor compuesto por estaciones de trabajo, un servidor de prácticas, sistemas de adquisición y plataformas de gestión del aprendizaje (LMS). Esta doble modalidad permite que el producto cubra tanto actividades presenciales de laboratorio como escenarios de enseñanza remota o semi-presencial, facilitando prácticas dirigidas, telelaboratorios y evaluación a distancia.

En cuanto a fronteras y puntos de unión, PLC4UNI expone interfaces claramente definidas que permiten su interoperabilidad con otros elementos del entorno educativo y con herramientas industriales de referencia. Entre las interfaces previstas se incluyen:

- **Entradas/Salidas físicas**: bloques digitales y analógicos con posibilidad de expansión mediante módulos, destinados a la conexión de sensores, pulsadores, relés y actuadores de laboratorio.
- **Interfaces de comunicación**: puerto USB para programación y registro de datos; interfaz Ethernet (y, opcionalmente, Wi-Fi) para comunicación local, administración remota y acceso a prácticas en red; puerto RS-485 para protocolo Modbus RTU cuando se deseé interoperar con equipos industriales de referencia.
- **Interfaz de programación**: soporte para lenguajes educativos estandarizados (por ejemplo, diagrama de escalera y texto estructurado compatibles con IEC 61131-3), a través de un IDE amigable de escritorio y/o una alternativa web.
- **Integración con software**: capacidad para exportar e importar programas y registros en formatos comunes (p. ej. JSON/CSV para telemetría, repositorios Git para versiones de proyectos), y mecanismos para interoperar con simuladores y plataformas LMS.

Es importante delimitar lo que PLC4UNI **no** pretende ser: no se concibe como un sustituto directo de PLC industriales certificados para entornos productivos críticos ni como un dispositivo para control de procesos en instalaciones con requisitos de seguridad funcional avanzados. Su diseño prioriza la seguridad didáctica, el modularidad y la facilidad de uso, por encima de características industriales como certificaciones SIL/PL o envolventes para condiciones extremas.

Por último, desde la perspectiva del ciclo de vida del producto, PLC4UNI se plantea como una plataforma **abierta y extensible**: su firmware y herramientas de programación estarán disponibles bajo licencias abiertas que permitan su adaptación, mantenimiento y ampliación por parte de la comunidad académica. Esta apertura facilita, además, la integración con herramientas de simulación y marcos de trabajo ya existentes en el ámbito educativo (por ejemplo, proyectos OpenPLC y repositorios institucionales), respetando siempre la trazabilidad y la reproducibilidad de las prácticas docentes.

[Inserte aquí el texto]

*Indicar si es un producto independiente o parte de un sistema mayor. En el caso de tratarse de un producto que forma parte de un sistema mayor, un diagrama que sitúe el producto dentro del sistema e identifique sus conexiones facilitan la comprensión.*

## 2.2 Funcionalidad del producto

El sistema **PLC4UNI** debe proporcionar un conjunto de funcionalidades básicas y complementarias que aseguren su valor pedagógico en la enseñanza universitaria de la automatización y el control. Estas funcionalidades se agrupan en cuatro ejes principales:

### a) Adquisición y control de señales

- Procesamiento de **entradas digitales y analógicas** provenientes de sensores y dispositivos didácticos.
- Generación de salidas digitales y analógicas para el control de actuadores de baja potencia (relés, LEDs, motores, solenoides, etc.).
- Posibilidad de ampliación modular mediante tarjetas de expansión de E/S.

### b) Programación y ejecución

- Disponibilidad de un **entorno de programación accesible**, con soporte para lenguajes estándar IEC 61131-3 (diagrama de escalera y texto estructurado).
- Capacidad de cargar, ejecutar y detener programas de usuario en tiempo real.
- Herramientas de monitoreo y depuración, que permitan observar el estado de las variables y señales durante la ejecución.

### c) Comunicación e integración

- Conectividad básica mediante **USB** para programación y alimentación.
- Conectividad extendida mediante **Ethernet y Wi-Fi**, orientada a prácticas en red y acceso remoto.
- Soporte para **protocolos industriales (Modbus RTU/TCP)**, con el fin de facilitar la interoperabilidad con otros equipos de referencia.
- Opciones de integración con **plataformas educativas** (LMS y simuladores).

### d) Gestión y registro de datos

- **Almacenamiento y exportación** de registros en formatos estándar (CSV, JSON) para su análisis y elaboración de informes.
- Integración con **repositorios de control de versiones** para la gestión de proyectos de programación.

En términos generales, el **PLC4UNI** está concebido para recibir señales de entrada, procesarlas mediante un programa lógico definido por el usuario y generar salidas físicas o virtuales en un entorno didáctico controlado. Con ello, se asegura que el estudiante pueda experimentar con las mismas dinámicas de la automatización industrial, en un marco adaptado a la enseñanza universitaria.

[Inserte aquí el texto]

*Resumen de las funcionalidades principales que el producto debe realizar, sin entrar en información de detalle.*

*En ocasiones la información de esta sección puede tomarse de un documento de especificación del sistema de mayor nivel (ej. Requisitos del sistema).*

*Las funcionalidades deben estar organizadas de manera que el cliente o cualquier interlocutor pueda entenderlo perfectamente. Para ello se pueden utilizar métodos textuales o gráficos.*

## 2.3 Características de los usuarios

Tipo de usuario	Estudiantes de pregrado, Estudiantes de posgrado, Docentes e instructores, Investigadores y desarrolladores
Formación	cursan programas de ingeniería, tecnología o carreras afines (electrónica, mecatrónica, energía, automatización, informática industrial), cursan programas de especialización o maestría en automatización, control, sistemas embebidos o áreas afines. profesionales con experiencia en ingeniería, automatización y control, con capacidad de diseñar planes de estudio y prácticas formativas., profesionales o estudiantes avanzados en áreas de innovación tecnológica, electrónica y software educativo.
Habilidades	conocimientos básicos en electricidad, programación y sistemas digitales; capacidad para comprender instrucciones técnicas y realizar prácticas guiadas., dominio intermedio o avanzado en programación y modelado de sistemas; experiencia en investigación aplicada., elaboración de guías pedagógicas, supervisión de proyectos, manejo de entornos de programación y didáctica de la enseñanza técnica., experiencia en programación avanzada, integración de sistemas, desarrollo de hardware y software de código abierto.
Actividades	desarrollo de prácticas de laboratorio, elaboración de proyectos académicos, simulación de procesos básicos de automatización y validación de programas en hardware real., diseño de proyectos interdisciplinarios, integración del PLC con otras plataformas, validación experimental de modelos y exploración de nuevas metodologías de enseñanza., diseño de prácticas de laboratorio, seguimiento de proyectos estudiantiles, evaluación de competencias y gestión de entornos de aprendizaje presenciales o remotos., ampliación de las funcionalidades del PLC4UNI, creación de módulos complementarios, pruebas de compatibilidad con estándares industriales y publicaciones académicas derivadas del uso del sistema.

En términos generales, los usuarios del PLC4UNI presentan un nivel educativo medio a avanzado, con competencias técnicas proporcionales a su etapa formativa. El

sistema ha sido diseñado con un enfoque escalable y progresivo, de modo que sea accesible para principiantes, pero también suficientemente robusto para usuarios avanzados que deseen profundizar en entornos de investigación y desarrollo.

*Descripción de los usuarios del producto, incluyendo nivel educacional, experiencia y experiencia técnica.*

## 2.4 Restricciones

El desarrollo y utilización del sistema **PLC4UNI** está sujeto a un conjunto de restricciones que delimitan su diseño, implementación y aplicación. Estas restricciones responden tanto a consideraciones técnicas como pedagógicas y normativas:

### a) Restricciones técnicas

**Recursos de hardware limitados:** el PLC está orientado a prácticas académicas, por lo que su capacidad de procesamiento, memoria y número de entradas/salidas será menor que la de un PLC industrial de gama alta.

**Compatibilidad con hardware didáctico:** el diseño se centra en la interacción con sensores y actuadores de baja potencia; no se contempla el control directo de cargas industriales de gran escala sin dispositivos intermedios.

**Lenguajes de programación restringidos:** se soportarán principalmente los lenguajes de la norma IEC 61131-3 más utilizados en el ámbito educativo (Ladder Diagram y Texto Estructurado), quedando fuera lenguajes avanzados menos comunes.

### b) Restricciones de software

**Sistema operativo base:** el firmware del PLC se diseñará para plataformas abiertas y ligeras (ej. RTOS embebido o Linux reducido), limitando la portabilidad directa a sistemas cerrados o propietarios.

**Dependencia de entornos abiertos:** el software de programación y simulación estará basado en herramientas de código abierto, lo que restringe la integración inmediata con IDEs propietarios de algunos fabricantes.

### c) Restricciones pedagógicas

**Orientación académica:** el producto no debe emplearse en entornos productivos industriales donde se requiera certificación de seguridad funcional (SIL/PL) o normativas estrictas de operación.

**Nivel de complejidad limitado:** el sistema está pensado para actividades de laboratorio y proyectos universitarios, evitando sobrecargar a los usuarios con configuraciones excesivamente técnicas o procesos de instalación complejos.

### d) Restricciones normativas y de seguridad

**Cumplimiento eléctrico básico:** el diseño se ajustará a normas de bajo voltaje y seguridad eléctrica aplicables a dispositivos educativos, evitando riesgos de electrocución o daños por sobrecarga.

**Licencias de software:** al basarse en un modelo de código abierto, los componentes deberán respetar licencias libres (GPL, MIT, etc.), lo que limita la inclusión de módulos con restricciones propietarias.

En síntesis, el PLC4UNI estará condicionado por la  **simplicidad, la seguridad y la accesibilidad**, lo que implica dejar fuera funcionalidades avanzadas de entornos industriales. Estas limitaciones no representan una debilidad, sino una estrategia para asegurar que el producto cumpla su objetivo principal: ser un recurso didáctico, económico y adaptable al contexto universitario.

[Inserte aquí el texto]

*Descripción de aquellas limitaciones a tener en cuenta a la hora de diseñar y desarrollar el sistema, tales como el empleo de determinadas metodologías de desarrollo, lenguajes de programación, normas particulares, restricciones de hardware, de sistema operativo etc.*

## 2.5 Suposiciones y dependencias

El desarrollo y correcta operación del sistema **PLC4UNI** se sustenta en un conjunto de suposiciones y dependencias que deben considerarse como condiciones externas al propio producto, pero que influyen directamente en su diseño, implementación y utilización:

### a) Suposiciones

**Disponibilidad de infraestructura educativa:** se asume que las universidades cuentan con laboratorios básicos de electricidad y electrónica donde el dispositivo pueda ser instalado y utilizado.

**Acceso a ordenadores personales:** se presupone que los estudiantes y docentes dispondrán de equipos de cómputo compatibles para instalar o ejecutar el entorno de programación asociado al PLC4UNI.

**Conectividad mínima:** se asume la existencia de redes locales o acceso a internet para habilitar funciones de integración, actualizaciones de software y prácticas remotas.

**Competencias previas de los usuarios:** se considera que los estudiantes poseen conocimientos básicos en programación estructurada y fundamentos de sistemas digitales, necesarios para comprender el funcionamiento del PLC.

### b) Dependencias

**Dependencia de hardware externo:** el uso del sistema requiere sensores, actuadores y módulos didácticos complementarios para el desarrollo de prácticas de laboratorio.

**Dependencia de software educativo:** el PLC4UNI se apoyará en entornos de programación de código abierto, lo que implica que su funcionamiento depende de la estabilidad y continuidad de dichos proyectos comunitarios.

**Dependencia de normas técnicas:** la compatibilidad con lenguajes de programación está condicionada a la vigencia de la norma **IEC 61131-3**, utilizada como referencia para los entornos de automatización.

**Dependencia de soporte institucional:** el éxito del proyecto depende del compromiso de las universidades para incorporar el dispositivo en sus planes de estudio, capacitar a los docentes y mantener actualizado el software.

En conjunto, estas suposiciones y dependencias constituyen el **marco de viabilidad del sistema**. Si alguna de ellas se ve modificada (por ejemplo, la falta de conectividad o la discontinuidad de herramientas de software abierto), será necesario ajustar los requisitos del producto para garantizar su funcionamiento en el contexto real de uso.

[Inserte aquí el texto]

*Descripción de aquellos factores que, si cambian, pueden afectar a los requisitos. Por ejemplo una asunción puede ser que determinado sistema operativo está disponible para*

*el hardware requerido. De hecho, si el sistema operativo no estuviera disponible, la SRS debería modificarse.*

## 2.6 Evolución previsible del sistema

El sistema **PLC4UNI** ha sido concebido como una plataforma **abierta, modular y escalable**, lo cual permite prever un conjunto de mejoras y evoluciones que podrán implementarse en fases posteriores, de acuerdo con las necesidades académicas y tecnológicas emergentes. Entre las principales líneas de evolución se destacan:

1. **Ampliación de módulos de hardware:** inclusión de nuevas tarjetas de expansión que integren más entradas/salidas, soporte para señales industriales de mayor potencia o módulos especializados (p. ej., comunicación inalámbrica avanzada, control de motores trifásicos o integración con sistemas de energía renovable).
2. **Extensión de lenguajes de programación soportados:** incorporación progresiva de otros lenguajes de la norma IEC 61131-3 (como Function Block Diagram – FBD o Sequential Function Chart – SFC), para ampliar las posibilidades pedagógicas y acercar aún más a los estudiantes a entornos industriales reales.
3. **Integración con tecnologías emergentes:** desarrollo de interfaces que permitan la conexión con entornos de simulación de gemelos digitales, plataformas de IoT educativo, inteligencia artificial aplicada al control de procesos y análisis de datos en la nube.
4. **Evolución pedagógica:** creación de repositorios abiertos de prácticas, librerías de proyectos compartidos y comunidades académicas que retroalimenten el uso del PLC4UNI, promoviendo un aprendizaje colaborativo y la generación de nuevo conocimiento.
5. **Certificación académica y estandarización:** en fases avanzadas, se contempla la posibilidad de desarrollar una certificación institucional de competencias adquiridas mediante el uso del PLC4UNI, estandarizando prácticas y asegurando la calidad formativa a nivel interuniversitario.

En suma, la evolución del sistema no se limita a la mejora técnica, sino que responde a una visión de sostenibilidad e innovación educativa, donde el **PLC4UNI** se consolide como una plataforma viva, capaz de adaptarse a los cambios de la industria y a las necesidades de la educación superior en el ámbito de la automatización y el control.

[Inserte aquí el texto]

*Identificación de futuras mejoras al sistema, que podrán analizarse e implementarse en un futuro.*

### 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

[Inserte aquí el texto]

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

El sistema **PLC4UNI**, basado en un microcontrolador ESP32 y diseñado para entornos académicos universitarios, debe integrar un conjunto de interfaces robustas que permitan la interacción entre el usuario, el hardware, el software y otros sistemas externos. Estos interfaces aseguran la usabilidad pedagógica, la escalabilidad del sistema y la compatibilidad con entornos industriales y educativos.

##### 3.1.1 Interfaces de usuario

[Inserte aquí el texto]

*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.*

El usuario interactuará con el PLC4UNI principalmente mediante un IDE multiplataforma (Windows y Linux), compatible con los lenguajes de programación Ladder Diagram (LD) y Texto Estructurado (ST), de acuerdo con el estándar IEC 61131-3.

El entorno de programación deberá incluir:

- Editor visual para creación de programas.
- Monitoreo en tiempo real de entradas, salidas y variables internas.
- Herramientas de depuración con mensajes claros en **español e inglés**.
- Documentación y ayuda contextual integrada.

Además, se implementará una **interfaz web ligera**, accesible mediante navegador, que permita programar y supervisar el dispositivo a través de conectividad Wi-Fi, ampliando las opciones de acceso remoto y prácticas de laboratorio virtual.

### 3.1.2 Interfaces de hardware

[Inserte aquí el texto]

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

El PLC4UNI deberá incluir un conjunto de interfaces de hardware adaptadas al entorno académico y compatibles con prácticas industriales básicas. En cuanto a entradas, dispondrá de **16 canales de entrada en total**, ubicados en la parte superior del módulo. De estos, **2 pines estarán dedicados a la entrada de alimentación principal de 24 VDC y 0 V**, asegurando la correcta alimentación del sistema. Además, contará con **una entrada de 4–20 mA** para sensores industriales normalizados (2 pines), **una entrada de 0–10 VDC** para prácticas de control de procesos (1 pin), **una entrada de 0–10 VAC** con terminales de fase, neutro y tierra (3 pines), **una entrada para transformador de corriente (CT)** utilizada en medición de corriente alterna (2 pines), **cuatro entradas digitales tipo NPN de 24 VDC** (4 pines) y **dos entradas PNP con optoacoplador** (2 pines), garantizando aislamiento eléctrico y compatibilidad con distintos tipos de sensores discretos. De esta forma, se completa el total de **16 entradas**, que abarcan señales digitales, analógicas, alternas y de corriente de proceso.

Respecto a las salidas, el PLC4UNI dispondrá de **16 canales de salida**, diseñados para ofrecer flexibilidad tanto en aplicaciones educativas como industriales. Entre ellas se incluye **un puerto RS-485 bidireccional** (3 pines) para comunicación y prácticas con protocolos como Modbus RTU, **tres salidas tipo PNP de 24 VDC** (3 pines) destinadas al control de actuadores de baja potencia, **un relé con contactos común, normalmente abierto y normalmente cerrado** (3 pines) y **un relé con contactos común y normalmente abierto** (2 pines), que permiten la comutación de cargas AC/DC. Además, incorpora **una salida analógica de 4–20 mA** (2 pines) y **una salida analógica de 0–10 VDC** (1 pin) para control de variadores, actuadores o simuladores de proceso, junto con **cuatro salidas tipo NPN (24 VDC)**, de las cuales **dos incluyen optoacopladores** para aislamiento de protección (4 pines en total). Esta configuración completa las **16 salidas del sistema**, proporcionando opciones de control digital, analógico y de potencia.

El dispositivo incluirá un **puerto USB 2.0** destinado exclusivamente a la **programación, transferencia de datos y alimentación auxiliar del microcontrolador** durante pruebas o mantenimiento. Finalmente, la **alimentación principal del PLC4UNI** se realizará mediante una fuente de **24 VDC**, con protección interna contra sobrecargas, polaridad inversa y cortocircuitos, garantizando la seguridad operativa en entornos académicos y de laboratorio.

El PLC4UNI deberá incluir un conjunto de interfaces de hardware adaptados al entorno académico y compatibles con prácticas industriales básicas. En cuanto a entradas, dispondrá de un mínimo de 16 entradas en total, de las cuales 12 serán digitales transistorizadas (24 VDC, tipo NPN/PNP, con aislamiento básico), distribuidas en 8 entradas estándar de 24 VDC, 2 entradas configuradas para señales de 0–5 V (compatibles con sensores y señales de laboratorio de bajo voltaje) y 2 entradas configuradas para 0–

10 V (para prácticas con variadores y procesos industriales). Las 4 entradas restantes serán analógicas, con resolución mínima de 12 bits, configurables en 2 de 0–10 V (sensores de proceso como temperatura, presión o caudal), 1 de 0–5 V (señales de laboratorio) y 1 de 4–20 mA (para sensores industriales normalizados). Respecto a las salidas, el sistema contará con un total de 12 salidas, de las cuales 11 serán a relé (AC/DC, hasta 250 VAC / 30 VDC, 5A), aptas para cargas resistivas e inductivas como lámparas, contactores, bombas pequeñas o solenoides, y 1 salida a TRIAC (AC, hasta 10A), diseñada para cargas alternas de mayor potencia (ejemplo: calefactores y lámparas). Además, se incorporarán 2 salidas analógicas (0–10 V, resolución mínima de 8 bits) destinadas al control de variadores, actuadores y prácticas de control de procesos, y hasta 8 canales PWM configurables en frecuencia y ciclo útil, orientados al manejo de motores DC, servos y otros actuadores. El sistema también integrará puertos de expansión estándar I<sup>2</sup>C, SPI y UART, lo que permitirá la conexión de periféricos académicos como pantallas, memorias, módulos HMI y sensores adicionales. Finalmente, la alimentación eléctrica podrá realizarse mediante una entrada estándar de 24 VDC, con protección interna frente a sobrecargas, polaridad inversa y cortocircuitos, garantizando la seguridad en prácticas de laboratorio.

### 3.1.3 Interfaces de software

[Inserción 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: contenido y formato*

El software asociado al PLC4UNI deberá contemplar las siguientes características:

- Compatibilidad con el estándar **IEC 61131-3** (LD y ST).
- Integración con **entornos de simulación** y repositorios abiertos (ej. OpenPLC, Git).
- Exportación e importación de proyectos en formatos estándar (ej. **CSV, JSON**) para análisis en Excel, Matlab o Python.
- Posibilidad de interacción con **plataformas LMS** (Moodle, Blackboard) mediante servicios web o intercambio de archivos.

### 3.1.4 Interfaces de comunicación

[Inserción aquí el texto]

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

USB 2.0: programación, transferencia de datos y alimentación del dispositivo.

Ethernet (mediante módulo externo ENC28J60 o W5500): para conexión estable en prácticas distribuidas y administración remota.

Wi-Fi integrado (802.11 b/g/n): comunicación inalámbrica con el entorno de programación y acceso remoto vía navegador.

Bluetooth/BLE: conexión con dispositivos móviles para prácticas educativas y supervisión básica.

UART (RS-232/RS-485 con transceptor externo): soporte de protocolos industriales como Modbus RTU.

I<sup>2</sup>C / SPI: conexión con periféricos académicos y módulos de expansión.

### 3.2 Requisitos funcionales (RF)

Esta sección contiene los requisitos funcionales del sistema PLC4UNI. Cada requisito se presenta en formato de tabla para garantizar trazabilidad, validación y claridad técnica.

[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, fórmulas para la conversión de información)*
- *Especificación de los requisitos lógicos para la información que será almacenada en base de datos (tipo de información, requerido)*

*Los requisitos funcionales pueden ser divididos en sub-secciones.*

#### 3.2.1 Requisito funcional (1-5)

RF 1

Número de requisito	RF 1
Nombre de requisito	Procesamiento de señales digitales
Tipo	Requisito
Fuente del requisito	Necesidad académica – prácticas básicas de automatización
Prioridad del requisito	Alta/Essencial
Descripción	El sistema deberá ser capaz de recibir y procesar entradas digitales (mínimo 8) y generar salidas digitales (mínimo 8) que permitan la interacción con sensores, pulsadores, LEDs y relés de baja potencia.

**RF 2**

Número de requisito	RF 2
Nombre de requisito	Procesamiento de señales analógicas
Tipo	Requisito
Fuente del requisito	Normas IEC 61131-3 y prácticas académicas en control de procesos
Prioridad del requisito	Alta/Esencial
Descripción	El PLC4UNI deberá soportar al menos 4 entradas analógicas (0–10 V o 4–20 mA) y 2 salidas analógicas, con el fin de realizar prácticas de control de variables continuas (temperatura, presión, velocidad, etc.).

**RF 3**

Número de requisito	RF 3
Nombre de requisito	Soporte para lenguajes de programación IEC 61131-3
Tipo	Requisito
Fuente del requisito	Estándar IEC 61131-3
Prioridad del requisito	Alta/Esencial
Descripción	El sistema deberá permitir la programación en al menos dos lenguajes estándar: Ladder Diagram (LD) y Texto Estructurado (ST), a través de un entorno de desarrollo integrado (IDE) accesible.

**RF 4**

Número de requisito	RF 4
Nombre de requisito	Conectividad USB y Ethernet
Tipo	Requisito
Fuente del requisito	Necesidades de interoperabilidad y prácticas en red
Prioridad del requisito	Alta/Esencial
Descripción	<b>Descripción:</b> <b>El PLC4UNI deberá incluir un total de 16 entradas y 16 salidas, además de un puerto USB destinado exclusivamente a la programación, alimentación y transferencia de datos. Este diseño simplificado busca mantener la robustez funcional necesaria para prácticas académicas e industriales básicas, reduciendo la complejidad de interconexión y priorizando la</b>

	<p><b>compatibilidad directa con dispositivos educativos.</b></p> <p><b>Entradas (16 en total):</b> La sección de entradas estará ubicada en la parte superior del dispositivo y contará con las siguientes conexiones y características:</p> <ul style="list-style-type: none"><li>• Entrada de alimentación principal: terminales de 24 VDC y 0 VDC, ocupando 2 pines de entrada.</li><li>• Entrada de corriente 4–20 mA: para sensores industriales, ocupa 2 pines.</li><li>• Entradas digitales tipo NPN (24 VDC): 4 canales dedicados, ocupando 4 pines.</li><li>• Entrada analógica de 0–10 VDC: para señales de control de proceso, ocupa 1 pin.</li><li>• Entrada de voltaje AC (0–10 VAC): con terminales de fase, neutro y GND, ocupando 3 pines.</li><li>• Entrada de transformador de corriente (CT): dedicada a medición de corriente alterna, ocupa 2 pines.</li><li>• Entradas tipo PNP con optoacoplador: 2 canales de aislamiento óptico, ocupando 2 pines.</li></ul> <p>Con esta configuración se alcanzan las 16 entradas totales, distribuidas entre señales digitales, analógicas, alternas y de corriente industrial, todas protegidas mediante aislamiento básico y acondicionamiento de señal.</p> <p><b>Salidas (16 en total):</b> El bloque de salidas estará diseñado para ofrecer compatibilidad tanto con cargas DC como AC, manteniendo la flexibilidad de uso didáctico e industrial. Su distribución será la siguiente:</p> <ul style="list-style-type: none"><li>• Puerto RS-485: interfaz bidireccional (entrada/salida) para comunicación con</li></ul>
--	--

	<p><b>dispositivos externos o prácticas Modbus RTU, ocupando 3 pines.</b></p> <ul style="list-style-type: none"><li>• <b>Salidas tipo PNP (24 VDC): 3 canales independientes, cada uno ocupando 1 pin.</b></li><li>• <b>Relé con contactos C, NA y NC: salida de control trifilar, ocupa 3 pines.</b></li><li>• <b>Relé con contactos C y NA: salida bifilar estándar, ocupa 2 pines.</b></li><li>• <b>Salida analógica de 4–20 mA: para control de actuadores industriales, ocupa 2 pines.</b></li><li>• <b>Salida analógica de 0–10 VDC: para control de variadores o simulación de procesos analógicos, ocupa 1 pin.</b></li><li>• <b>Salidas tipo NPN (24 VDC): 4 canales, de los cuales 2 estarán optoacoplados para protección adicional, ocupando 4 pines en total.</b></li></ul> <p><b>Esta disposición completa las 16 salidas, asegurando compatibilidad con señales de control analógicas y digitales, además de incluir elementos de conmutación electromecánica e interfaces industriales.</b></p> <p><b>Puerto de comunicación:</b></p> <ul style="list-style-type: none"><li>• <b>USB 2.0: único puerto de comunicación del sistema, utilizado para la programación, transferencia de datos y alimentación del microcontrolador durante pruebas y mantenimiento.</b></li></ul> <p><b>Condiciones de seguridad:</b> Todos los puertos deberán estar claramente identificados, documentados y protegidos contra sobrecargas, inversiones de polaridad y cortocircuitos. La alimentación principal será de 24 VDC, con fusibles de protección y optoaislamiento en los canales de entrada y salida más sensibles, garantizando la seguridad del usuario y del equipo durante las prácticas de laboratorio.</p>
--	--

--	--

**RF 5**

Número de requisito	RF 5
<b>Nombre de requisito</b>	Interfaz de usuario para depuración
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Necesidad académica – seguimiento de prácticas
<b>Prioridad del requisito</b>	Media/Deseado
<b>Descripción</b>	El entorno de programación deberá incluir herramientas de monitoreo en tiempo real, que permitan observar el estado de las variables y señales durante la ejecución del programa.

**Requisito funcional (6-10)**
**RF 6**

Número de requisito	RF 6
<b>Nombre de requisito</b>	Comunicación inalámbrica opcional
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Necesidad de prácticas remotas y enseñanza híbrida
<b>Prioridad del requisito</b>	Media/Deseado
<b>Descripción</b>	El sistema deberá permitir la incorporación de un módulo Wi-Fi opcional para habilitar la comunicación inalámbrica con el entorno de programación y plataformas educativas, facilitando el desarrollo de prácticas en red y tele laboratorios.

**RF 7**

Número de requisito	RF 7
<b>Nombre de requisito</b>	Soporte de protocolos industriales básicos
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Interoperabilidad académica con equipos de referencia
<b>Prioridad del requisito</b>	Alta/Esencial
<b>Descripción</b>	El PLC4UNI deberá soportar protocolos industriales de uso común, como Modbus RTU/TCP, con el fin de integrarse en prácticas donde se empleen dispositivos de referencia (sensores, variadores, HMI).

**RF 8**

Número de requisito	RF 8
<b>Nombre de requisito</b>	Registro y exportación de datos
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Necesidad académica – análisis de resultados
<b>Prioridad del requisito</b>	Alta/Esencial
<b>Descripción</b>	El sistema deberá ser capaz de registrar el estado de variables y eventos de ejecución, exportando los datos en formatos estándar (CSV, JSON) para análisis posterior en herramientas externas como Excel, Matlab o Python.

**RF 9**

Número de requisito	RF 9
<b>Nombre de requisito</b>	Integración con plataformas de gestión del aprendizaje (LMS)
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Estrategia pedagógica universitaria
<b>Prioridad del requisito</b>	Media/Deseado
<b>Descripción</b>	El entorno de programación deberá contar con mecanismos de integración con plataformas educativas (ej. Moodle, Blackboard), permitiendo la carga y validación de prácticas de los estudiantes de manera automática.

**RF 10**

Número de requisito	RF 10
<b>Nombre de requisito</b>	Ejecución y monitoreo en tiempo real
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Prácticas de automatización universitaria
<b>Prioridad del requisito</b>	Alta/Esencial
<b>Descripción</b>	El sistema deberá ejecutar los programas cargados en tiempo real, asegurando tiempos de ciclo compatibles con la simulación de procesos básicos de automatización (del orden de 1–10 ms), y permitiendo al usuario monitorear variables durante la ejecución.

### 3.2.2 Requisito funcional (11-15)

**RF 11**

Número de requisito	RF 11
<b>Nombre de requisito</b>	Mecanismos de seguridad didáctica
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Normativas de seguridad educativa
<b>Prioridad del requisito</b>	Alta/Esencial
<b>Descripción</b>	El sistema deberá incorporar protecciones contra sobrecargas y errores comunes de conexión, garantizando que los estudiantes puedan operar el dispositivo sin riesgo eléctrico ni daño al hardware.

**RF 12**

Número de requisito	RF 12
<b>Nombre de requisito</b>	Modularidad y escalabilidad del hardware
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Necesidad de flexibilidad académica
<b>Prioridad del requisito</b>	Alta/Esencial
<b>Descripción</b>	El hardware del PLC4UNI deberá estar diseñado de manera modular, permitiendo la adición de nuevas tarjetas de expansión o periféricos, con el fin de adaptarse a distintos niveles de complejidad académica.

**RF 13**

Número de requisito	RF 13
<b>Nombre de requisito</b>	Repositorio de prácticas y ejemplos
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Enfoque pedagógico universitario
<b>Prioridad del requisito</b>	Media/Deseado
<b>Descripción</b>	El sistema deberá contar con un repositorio de prácticas preconfiguradas y ejemplos de programas que permitan a los estudiantes iniciarse rápidamente en el uso del PLC.

**RF 14**

Número de requisito	RF 14
Nombre de requisito	Compatibilidad con simuladores externos
Tipo	Requisito
Fuente del requisito	Integración académica
Prioridad del requisito	Alta/Essencial
Descripción	El PLC4UNI deberá permitir la integración con simuladores educativos y de procesos industriales, de forma que los estudiantes puedan validar programas tanto en entornos virtuales como en hardware real.

**RF 15**

Número de requisito	RF 15
Nombre de requisito	Soporte comunitario y actualización abierta
Tipo	Requisito
Fuente del requisito	Modelo open source
Prioridad del requisito	Media/Deseado
Descripción	El proyecto deberá contar con un sistema de distribución abierto (repositorio Git) para actualizaciones de firmware, software de programación y librerías de prácticas, asegurando su evolución a largo plazo con apoyo de la comunidad académica.

### 3.3 Requisitos no funcionales (RNF)

#### 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”.*

**RNF 1**

Número de requisito	RNF 1
Nombre de requisito	Rendimiento
Tipo	Requisito
Fuente del requisito	Necesidad de prácticas en tiempo real
Prioridad del requisito	Alta/Essencial

<b>Descripción</b>	El sistema deberá procesar ciclos de ejecución en un tiempo máximo de 10 ms para garantizar la simulación de procesos en tiempo real.
--------------------	---

### 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.*

**RNF 2**

Número de requisito	RNF 2
Nombre de requisito	Seguridad
Tipo	Requisito
Fuente del requisito	Normativas educativas y de seguridad eléctrica
Prioridad del requisito	Alta/Essencial
Descripción	El PLC deberá cumplir con normativas de bajo voltaje y seguridad eléctrica para dispositivos educativos, minimizando riesgos de electrocución y daños por sobrecarga.

### 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.*

**RNF 3**

Número de requisito	RNF 3
Nombre de requisito	Fiabilidad
Tipo	Requisito
Fuente del requisito	Prácticas de laboratorio
Prioridad del requisito	Alta/Essencial
Descripción	El sistema deberá garantizar una disponibilidad de al menos 95% en sesiones de laboratorio, con mecanismos de recuperación frente a fallos menores de software.

**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.*

**RNF 4**

Número de requisito	RNF 4
Nombre de requisito	Disponibilidad
Tipo	Requisito
Fuente del requisito	Entornos académicos
Prioridad del requisito	Media/Deseado
Descripción	El software asociado deberá estar disponible para múltiples sistemas operativos (Windows, Linux), facilitando el acceso a estudiantes y docentes.

**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.*

**RNF 5**

Número de requisito	RNF 5
Nombre de requisito	Mantenibilidad
Tipo	Requisito
Fuente del requisito	Modelo open source
Prioridad del requisito	Media/Deseado

<b>Descripción</b>	El sistema deberá contar con documentación clara y repositorios abiertos que faciliten la corrección de errores, la actualización de funciones y la contribución comunitaria.
--------------------	---

### 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 independiente 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.*

### RNF 6

Número de requisito	RNF 6
Nombre de requisito	Portabilidad
Tipo	Requisito
Fuente del requisito	Entornos académicos diversos
Prioridad del requisito	Media/Deseado
Descripción	El software del PLC deberá ser portable a distintas plataformas, con un porcentaje mínimo de 80% de código independiente del sistema operativo, garantizando su uso en contextos diversos.

### 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*

**RNF 7**

Número de requisito	RNF 7
Nombre de requisito	Usabilidad del entorno de programación
Tipo	Requisito
Fuente del requisito	Necesidades pedagógicas universitarias
Prioridad del requisito	Alta/Essencial
Descripción	El IDE del PLC4UNI deberá permitir que un estudiante novato pueda crear su primer programa básico en menos de 15 minutos, con interfaz intuitiva, mensajes de error claros en español e inglés, y ayuda contextual integrada.

**RNF 8**

Número de requisito	RNF 8
Nombre de requisito	Escalabilidad del sistema
Tipo	Requisito
Fuente del requisito	Crecimiento institucional previsto
Prioridad del requisito	Media/Deseado
Descripción	El sistema deberá soportar la conexión simultánea de al menos 30 dispositivos PLC4UNI en una red local universitaria, manteniendo tiempos de respuesta inferiores a 500ms para operaciones básicas de programación y monitoreo.

**RNF 9**

Número de requisito	RNF 9
Nombre de requisito	Consumo energético eficiente
Tipo	Requisito
Fuente del requisito	Sostenibilidad educativa y costos operativos
Prioridad del requisito	Media/Deseado
Descripción	El PLC4UNI deberá operar con un consumo máximo de 5W en funcionamiento normal y entrar en modo de bajo consumo (menos de 1W) automáticamente tras 15 minutos de inactividad, con capacidad de despertar mediante comunicación USB o Ethernet.

**RNF 10**

Número de requisito	RNF 10
Nombre de requisito	Interoperabilidad con estándares industriales
Tipo	Requisito

<b>Fuente del requisito</b>	Transferencia de competencias al ámbito industrial
<b>Prioridad del requisito</b>	Alta/Esencial
<b>Descripción</b>	El sistema deberá ser compatible con al menos el 80% de las funciones básicas definidas en el estándar IEC 61131-3, permitiendo que los programas desarrollados puedan ser adaptados a PLCs comerciales con modificaciones mínimas (menos del 20% del código).

### Requisito Legal 1

<b>Número de requisito</b>	<b>Requisito Legal 1</b>
<b>Nombre de requisito</b>	Licenciamiento de Código Abierto
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Modelo de distribución
<b>Prioridad del requisito</b>	Alta/Esencial
<b>Descripción</b>	El proyecto PLC4UNI deberá distribuirse bajo una licencia de código abierto compatible con el ecosistema educativo (preferiblemente GPL v3 o MIT), garantizando que las instituciones educativas puedan modificar, distribuir y contribuir al desarrollo del sistema sin restricciones legales.

### Requisito Cultural 1

<b>Número de requisito</b>	<b>Requisito Cultural 1</b>
<b>Nombre de requisito</b>	Localización Educativa
<b>Tipo</b>	Requisito
<b>Fuente del requisito</b>	Entornos académicos internacionales
<b>Prioridad del requisito</b>	Alta/Esencial
<b>Descripción</b>	El sistema deberá soportar localización completa en español e inglés, incluyendo interfaz de usuario, mensajes de sistema, documentación técnica y materiales didácticos, facilitando su adopción en universidades hispanohablantes y su integración en programas de intercambio internacional.

**CAPITULO II. MANUAL DE INSTRUCCIONES**

**A - C**

- **ADC (Analog-to-Digital Converter):** Conversor Analógico-Digital. Circuito que convierte una señal analógica (voltaje continuo) en un valor digital numérico que el procesador puede leer.
- **ADE7953:** Circuito integrado especializado de *Analog Devices* utilizado en el PLC 4 UNI para la medición de energía eléctrica (voltaje, corriente, potencia activa) vía I2C.
- **AI (Analog Input / Entrada Analógica):** Señal de valor continuo empleada para representar magnitudes variables como temperatura, presión o nivel de voltaje (0-10V, 4-20mA).
- **AO (Analog Output / Salida Analógica):** Señal continua de salida que permite el control proporcional de equipos como variadores de frecuencia o válvulas.
- **API (Application Programming Interface):** Conjunto de reglas y herramientas que permiten que diferentes aplicaciones de software (como el IDE y el PLC) se comuniquen entre sí.
- **CT (Current Transformer):** Transformador de Corriente. Sensor utilizado para medir corriente alterna de forma no intrusiva, transformando una corriente alta en una señal pequeña segura para el PLC.

**D - F**

- **DI (Digital Input / Entrada Digital):** Señal binaria que solo admite dos estados lógicos: 1 (activo/24V) o 0 (inactivo/0V).
- **DO (Digital Output / Salida Digital):** Señal binaria de salida utilizada para accionar dispositivos externos en modo ON/OFF, como relés o luces piloto.
- **ESP-IDF (Espressif IoT Development Framework):** Kit de desarrollo oficial de Espressif para programar el microcontrolador ESP32-S3 a bajo nivel.
- **ESP32-S3:** Microcontrolador de alto rendimiento y bajo costo con conectividad Wi-Fi y Bluetooth integrada, que actúa como el "cerebro" del PLC 4 UNI.
- **Firmware:** Software que reside de forma permanente en el hardware del PLC, encargado de controlar los periféricos y ejecutar la lógica del usuario.

**G - L**

- **GPIO (General Purpose Input/Output):** Pin genérico en un chip cuyo comportamiento (entrada o salida) se puede controlar por software.
- **HMI (Human Machine Interface):** Interfaz Hombre-Máquina. Pantalla o software que permite a un operador interactuar con el PLC para visualizar datos o enviar comandos.
- **I2C (Inter-Integrated Circuit):** Protocolo de comunicación serial de corto alcance utilizado internamente para comunicar el ESP32 con el chip de energía (ADE7953).
- **IDE (Integrated Development Environment):** Entorno de Desarrollo Integrado. Software en PC donde se escribe, compila y carga el código al PLC (ej. Arduino IDE, PLC4uni Studio).
- **IoT (Internet of Things):** Internet de las Cosas. Red de dispositivos físicos conectados que intercambian datos a través de internet, característica clave de este PLC.
- **JSON (JavaScript Object Notation):** Formato de texto ligero y legible utilizado para empaquetar los datos enviados por el PLC vía MQTT.
- **Ladder (LD):** Lenguaje de programación gráfico (Escalera) basado en esquemas de contactos eléctricos, estándar en la industria de automatización.

**M - P**

- **Modbus RTU:** Protocolo de comunicación industrial estándar, robusto y sencillo, que funciona sobre la capa física RS-485 para conectar el PLC con otros equipos industriales.
- **MQTT (Message Queuing Telemetry Transport):** Protocolo de mensajería ligero diseñado para enviar datos de sensores a la nube con un consumo mínimo de ancho de banda.
- **NPN (Sink / Sumidero):** Configuración de sensor donde la salida conecta la carga a Tierra (GND/Negativo) cuando se activa. Requiere que la entrada del PLC provea la corriente (Source).
- **OTA (Over-The-Air):** Tecnología que permite actualizar el firmware del PLC de forma inalámbrica a través de Wi-Fi, sin conectar cables.

- **PCB (Printed Circuit Board):** Placa de Circuito Impreso. La base física donde se sueldan y conectan los componentes electrónicos.
- **PLC (Programmable Logic Controller):** Controlador Lógico Programable. Equipo electrónico robusto diseñado para controlar procesos industriales en tiempo real.
- **PNP (Source / Fuente):** Configuración de sensor donde la salida envía voltaje positivo (+24V) cuando se activa. Requiere que la entrada del PLC drene la corriente (Sink).

## R - Z

- **RS-485:** Estándar de transmisión de datos serial diferencial, muy inmune al ruido eléctrico, utilizado para largas distancias en entornos industriales.
- **S/S (Sink/Source):** Terminal común en las entradas del PLC que permite seleccionar la referencia de voltaje para hacerlo compatible tanto con sensores NPN como PNP.
- **Scan Cycle (Ciclo de Escaneo):** Proceso repetitivo donde el PLC lee entradas, ejecuta el programa de usuario y actualiza las salidas físicas.
- **USB-C:** Conector estándar reversible utilizado en el PLC para alimentación de 5V y programación de datos desde el PC.

### 3.5 ADVERTENCIAS Y PRECAUCIONES

Antes de instalar, cablear o poner en marcha el **PLC 4 UNI**, lea atentamente las siguientes advertencias. El incumplimiento de estas instrucciones puede resultar en daños irreparables al equipo, riesgos eléctricos o lesiones personales.

#### 3.5.1 ADVERTENCIAS DE PELIGRO ELÉCTRICO

- **Voltaje de Alimentación Crítico:** Este equipo está diseñado exclusivamente para alimentarse con **24 VDC**. Conectar directamente 110/220 VAC a las entradas de alimentación (24V+, GND) causará la destrucción inmediata de la fuente interna y riesgo de incendio. Verifique la polaridad antes de encender.
- **Separación de Circuitos:** Mantenga separados físicamente los cables de baja tensión (sensores, RS-485, USB) de los cables de potencia (Salidas a Relé, Alimentación AC). No los pase por las mismas canaletas para evitar interferencias electromagnéticas y riesgos de cortocircuito.

### 3.5.2 PRECAUCIONES DE HARDWARE (ESP32)

- **Niveles Lógicos de 3.3V:** El núcleo del sistema (ESP32-S3) opera a **3.3V**. Aunque las entradas digitales están protegidas y adaptadas para 24V, los pines expuestos de expansión o comunicación interna no toleran 5V ni 24V. **No inyecte señales de 5V directamente** en los pines del microcontrolador si decide soldar expansiones.
- **Entradas Analógicas (ADC):**
  - Para las entradas de **0-10V**, asegúrese de no exceder los 10V DC, ya que esto podría dañar el divisor de tensión interno y el pin del ADC.
  - Para las entradas de **4-20mA**, respete la polaridad del lazo de corriente. Invertir la conexión podría dañar la resistencia *shunt* de precisión interna.

### 3.6 USO EDUCATIVO Y LIMITACIONES

- **No Certificado para Soporte Vital:** El **PLC 4 UNI** es una herramienta educativa y de prototipado. **NO** debe utilizarse en aplicaciones críticas de seguridad, soporte vital, o control de maquinaria peligrosa donde una falla del software o hardware pueda resultar en lesiones graves o muerte.
- **Parada de Emergencia (E-STOP):** Para cualquier práctica que involucre partes móviles (motores, pistones), es obligatorio instalar un botón de **Parada de Emergencia** físico cableado en serie con la alimentación de los actuadores, o configurado en la entrada designada para cortar las salidas por software.
- **Manipulación de la PCB:** Si utiliza el equipo sin la carcasa superior, tome precauciones contra la **descarga electrostática (ESD)**. Toque una superficie metálica conectada a tierra antes de manipular la placa electrónica.

### 3.7 CONDICIONES AMBIENTALES

- **Humedad y Polvo:** La carcasa estándar tiene un grado de protección **IP20**. No exponga el equipo a goteos de agua, lluvia o ambientes con polvo metálico conductivo. Instálelo dentro de un gabinete si el entorno es hostil.
- **Ventilación:** Aunque el ESP32 es eficiente, los reguladores de voltaje y los relés generan calor. No cubra los orificios de ventilación de la carcasa.

### 3.8 CONEXIONES

El panel frontal del PLC 4 UNI dispone de diversas interfaces de conexión mediante borneras desmontables y puertos estándar. A continuación se detalla el pineado y la función de cada grupo:

#### 3.8.1 Alimentación (Power Supply):

- **V+ / GND:** Entrada de alimentación principal de **24 VDC**. El sistema cuenta con protecciones y reguladores internos para el funcionamiento del microcontrolador.
- **USB-C:** Puerto para programación del firmware, depuración serial y alimentación lógica (5V) durante el desarrollo.

#### 3.8.2 Entradas (Inputs):

- **Entradas Digitales (DI):** 8 entradas optoacopladas para señales de **24 VDC**. Admiten configuración **PNP** (lógica positiva) o **NPN** (lógica negativa) mediante un terminal común. Cuentan con filtrado por hardware para evitar rebotes.

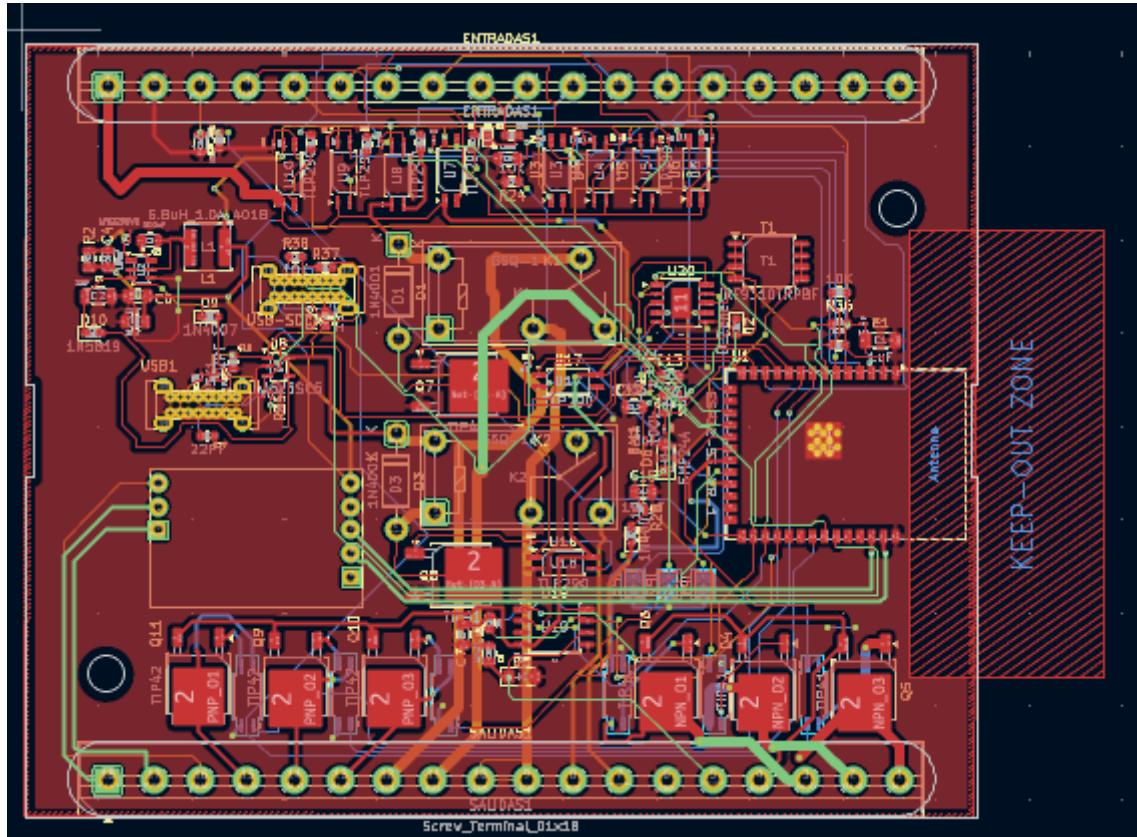
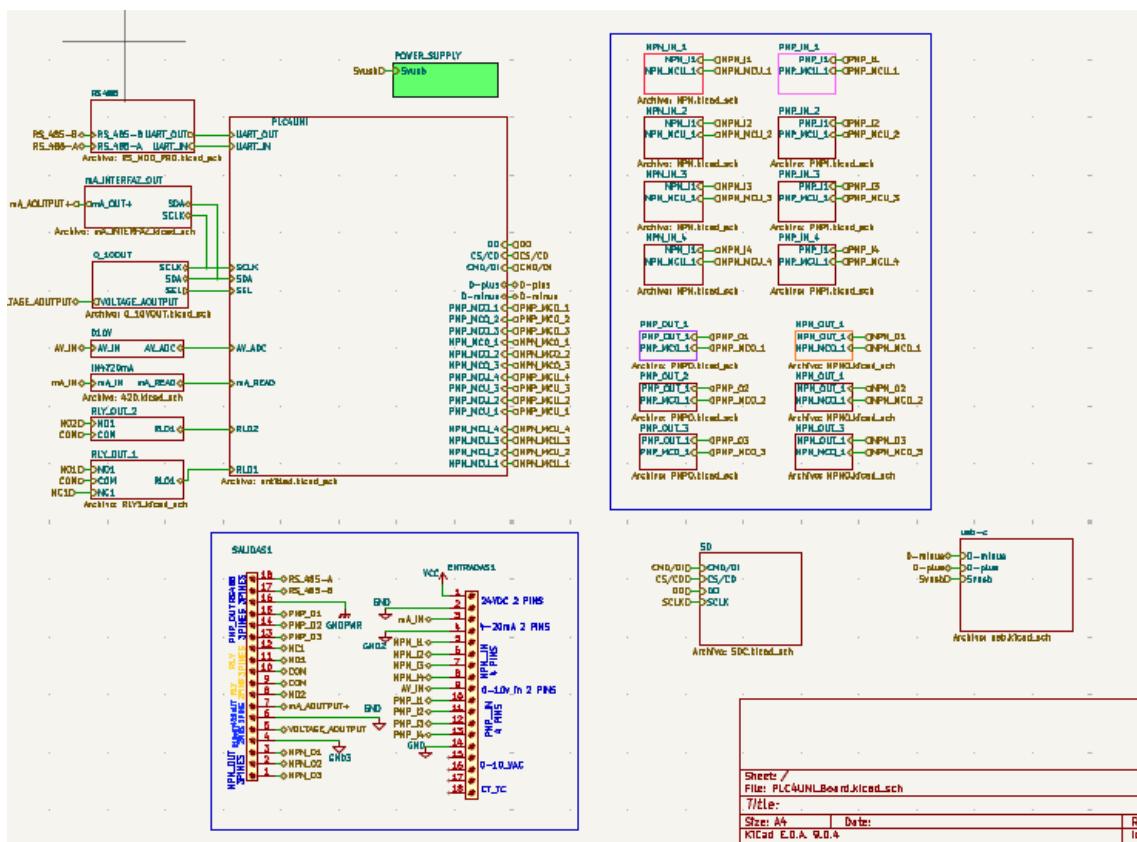
#### 3.8.3 Entradas Analógicas (AI):

- **0-10V:** Para sensores de voltaje estándar.
- **4-20mA:** Para instrumentación industrial (loops de corriente). Resolución mínima de 12 bits.
  - **Medición de Energía:** Interfaz para Transformador de Corriente (CT) y voltaje AC para el cálculo de potencia activa y consumo (kWh).

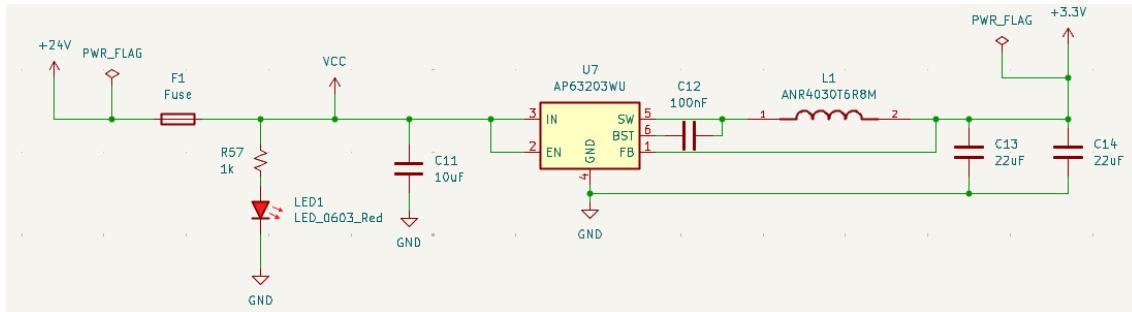
#### 3.8.4 Salidas (Outputs):

- **Salidas Digitales (DO):** 8 salidas para actuación.
  - **Relé:** Contactos secos para manejar cargas de hasta 5A (AC/DC).
  - **Transistor (NPN/PNP):** Salidas de alta velocidad para PWM o señales de control.
- **Salidas Analógicas (AO):** Salidas de 0-10V y 4-20mA para el control proporcional de actuadores o variadores de frecuencia.

## Diagramas



## Protección de la Fuente para ESP32



Este circuito es un sistema de **protección y regulación de fuente de alimentación** que convierte un voltaje industrial de 24V a un voltaje de 3.3V utilizable por el microcontrolador ESP32, incorporando múltiples capas de protección para garantizar un suministro eléctrico seguro y estable.

La primera línea de defensa es el fusible F1 ubicado justo después de la entrada de 24V. Este fusible actúa como protección contra sobrecorriente, cortando automáticamente el suministro si la corriente excede su valor nominal. Esto protege todo el circuito downstream de condiciones de falla como cortocircuitos, componentes defectuosos, o sobrecargas severas que podrían causar daños permanentes o incluso riesgos de incendio. El fusible es un componente sacrificial diseñado para fallar de manera controlada antes que cualquier otro elemento más costoso del sistema.

Después del fusible viene el LED indicador LED1 con su resistencia limitadora R57 de 1kΩ. Este LED rojo proporciona indicación visual de que el sistema está recibiendo alimentación de 24V, lo cual es invaluable para diagnóstico rápido. Si el LED no enciende, sabes inmediatamente que hay un problema en la fuente de alimentación externa o que el fusible se ha abierto. Esta simple indicación puede ahorrar horas de troubleshooting al identificar rápidamente si el problema está en la alimentación o en otra parte del circuito.

El corazón del circuito es el regulador de voltaje buck (reductor) AP63203WU, identificado como U7. Este es un regulador comutado de alta eficiencia que convierte los 24V de entrada a los 3.3V que necesita el ESP32. Los reguladores buck funcionan mediante conmutación rápida de la corriente a través de un inductor, almacenando energía en el campo magnético del inductor y liberándola de manera controlada para mantener un voltaje de salida estable. Este tipo de regulador es mucho más eficiente que un regulador lineal cuando hay grandes diferencias entre voltaje de entrada y salida, ya que disipa mucho menos energía en forma de calor.

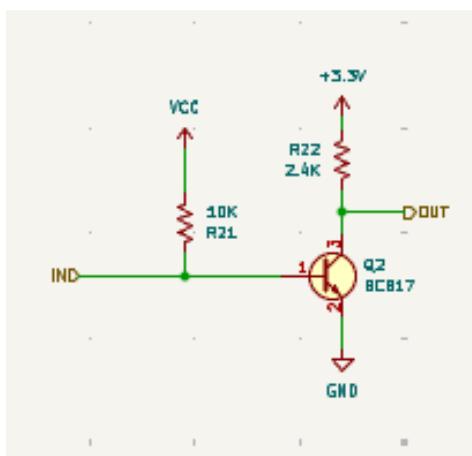
El inductor L1 de 6.8μH (modelo ANR4030T6R8M) es un componente crítico del regulador buck. Trabaja en conjunto con el circuito interno del AP63203WU para almacenar y liberar energía durante cada ciclo de conmutación. El valor del inductor está cuidadosamente seleccionado para el rango de corriente de salida esperado y la frecuencia de conmutación del regulador, optimizando el balance entre tamaño físico, eficiencia, y calidad de la regulación.

Los capacitores juegan roles fundamentales en el filtrado y estabilización del circuito. El capacitor C11 de 10μF en la entrada del regulador sirve como reservorio de energía y filtro de ruido de alta frecuencia proveniente de la fuente de 24V. Absorbe picos transitorios y proporciona la corriente instantánea que el regulador necesita durante sus ciclos de conmutación. El capacitor C12 de 100nF, conectado al pin FB (feedback), es probablemente parte del lazo de realimentación o bootstrap del regulador, ayudando a estabilizar la operación del circuito interno. En la salida, los capacitores C13 y C14, ambos de 22μF, trabajan en paralelo para proporcionar 44μF de

capacitancia total, lo que garantiza un voltaje de salida muy estable incluso ante cambios bruscos en la demanda de corriente del ESP32.

Este diseño de fuente de alimentación incorpora protecciones múltiples: protección contra sobrecorriente mediante el fusible, protección térmica interna en el regulador AP63203WU, y la estabilización mediante filtrado extensivo que protege contra picos y ruido. Todo esto asegura que el ESP32 reciba alimentación limpia, estable, y dentro de especificaciones, lo cual es absolutamente crítico para el funcionamiento confiable del microcontrolador. Una alimentación de mala calidad puede causar resets inesperados, comportamiento errático, corrupción de datos, o daño permanente al chip, por lo que este circuito de protección es tan importante como el propio microcontrolador.

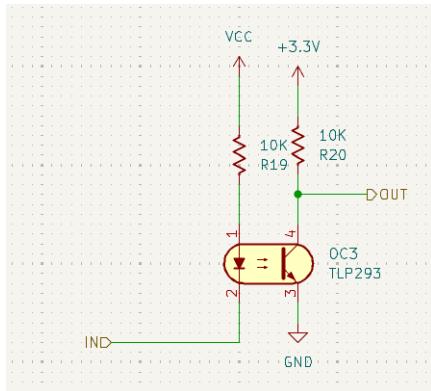
### Entradas NPN



El circuito tiene una señal de entrada llamada IND que controla un transistor BC817 (NPN) a través de una resistencia de base R21 de  $10\text{k}\Omega$ . Esta resistencia limita la corriente que llega a la base del transistor, protegiéndolo de sobrecorriente. El colector del transistor está conectado a la salida DOUT, y entre DOUT y la alimentación VCC (+3.3V) hay una resistencia pull-up R22 de  $2.4\text{k}\Omega$  que mantiene la salida en estado alto cuando el transistor está apagado.

El funcionamiento es el siguiente: cuando la señal IND está en bajo (0V), no hay corriente fluyendo hacia la base del transistor, por lo que éste permanece en corte. En este estado, el transistor se comporta como un circuito abierto entre colector y emisor, dejando que la resistencia pull-up R22 mantenga la salida DOUT en alto (3.3V). Por el contrario, cuando IND recibe una señal alta (3.3V), fluye corriente a través de R21 hacia la base del transistor, saturándolo. El transistor entonces conecta DOUT directamente a tierra a través de su colector-emisor, haciendo que DOUT baje a 0V.

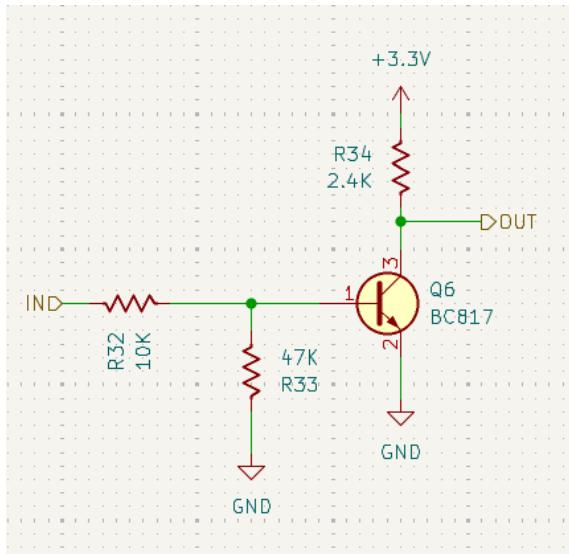
## Entrada NPN Opto acoplada



El opto acoplador TLP293 es el componente central de este diseño y funciona como un puente aislado entre dos circuitos eléctricos completamente separados. Por un lado tenemos el circuito de entrada donde llega la señal IND, que se conecta a través de una resistencia R19 de  $10\text{k}\Omega$  al LED interno del opto acoplador. Esta resistencia limita la corriente que circula por el LED para protegerlo de exceso de corriente. Cuando la señal IND está activa y hay una diferencia de potencial entre IND y GND, el LED interno del opto acoplador se enciende.

La magia del opto acoplador ocurre mediante transferencia óptica: cuando el LED interno se ilumina, la luz emitida incide sobre un fototransistor ubicado en el otro lado del componente, pero eléctricamente aislado del LED. Este fototransistor detecta la luz y comienza a conducir, conectando su colector con su emisor. Es importante destacar que entre el lado del LED y el lado del fototransistor no existe ninguna conexión eléctrica, solo transmisión de luz a través de un material aislante transparente, lo que proporciona aislamiento de varios miles de voltios.

## Entrada PNP



Este circuito es una **entrada digital para sensores PNP**, diseñada para leer señales de sensores industriales tipo PNP que funcionan entregando voltaje positivo cuando están activos.

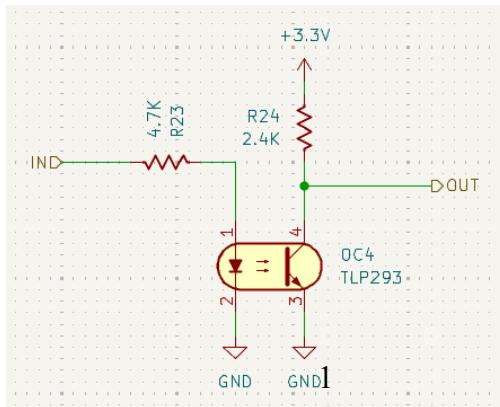
El componente principal es un transistor PNP BC817 (Q6) que procesa la señal de entrada IND. La señal llega a través de una resistencia R32 de  $10\text{k}\Omega$  que limita la corriente y protege tanto al sensor como al transistor. Entre la base del transistor y tierra hay una resistencia R33 de  $47\text{k}\Omega$  que funciona como pull-down, asegurando que la base se mantenga en un estado definido cuando no hay señal presente. En el colector del transistor hay una resistencia pull-up R34 de  $2.4\text{k}\Omega$  conectada a la alimentación de 3.3V, desde donde se toma la señal de salida DOUT.

El funcionamiento está diseñado específicamente para sensores PNP industriales. Estos sensores tienen una particularidad: cuando detectan algo, conectan su salida a su voltaje de alimentación positivo, y cuando no detectan nada, dejan la salida flotante o la conectan a través de alta impedancia. Cuando el sensor PNP está inactivo, la entrada IND no recibe voltaje o está en bajo. En este estado, la resistencia R33 mantiene la base del transistor cerca del potencial de tierra, mientras que el emisor está a un voltaje más positivo, lo que hace que el transistor PNP conduzca. Al conducir, el transistor conecta el colector prácticamente al voltaje del emisor, haciendo que DOUT suba a un nivel alto.

Cuando el sensor PNP se activa, entrega voltaje positivo a IND. Este voltaje positivo llega a la base del transistor a través de R32, elevando el potencial de la base hasta un valor cercano al del emisor. En un transistor PNP, cuando la base y el emisor están al mismo potencial o la base está más positiva que el emisor, el transistor deja de conducir y entra en corte. Con el transistor abierto, la resistencia pull-up R34 ya no tiene una trayectoria de baja resistencia hacia el emisor, y DOUT cae hacia tierra a través de las resistencias del circuito, llevando la salida a nivel bajo.

Este diseño es especialmente útil en automatización industrial donde los sensores PNP son estándar, como sensores de proximidad, fotoeléctricos, o finales de carrera electrónicos que operan a voltajes industriales superiores a los 3.3V del microcontrolador. El circuito adapta estos niveles y polaridades al rango que puede manejar el ESP32 de forma segura, además de proporcionar cierta protección contra sobre voltajes mediante las resistencias limitadoras.

### Entrada PNP Opto acoplada



Este circuito es una **entrada digital optoacoplada para sensores PNP**, que combina el aislamiento galvánico del optoacoplador con la capacidad de leer señales de sensores industriales tipo PNP.

El diseño utiliza un optoacoplador TLP293 (OC4) como elemento de aislamiento entre la señal de entrada IND y el circuito de control. La señal de entrada se conecta a través de una resistencia R23 de  $47\text{k}\Omega$  al ánodo del LED interno del optoacoplador, mientras que el cátodo del LED va directo a tierra. Esta resistencia limita la corriente que circula por el LED del optoacoplador,

protegiéndolo de sobrecorriente y permitiendo que funcione con los niveles de voltaje típicos de sensores PNP industriales.

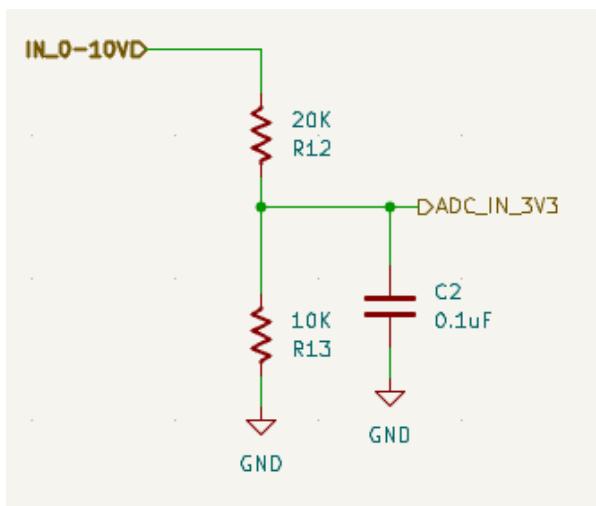
Los sensores PNP industriales tienen una característica específica en su operación: cuando están activos, conectan su salida al voltaje positivo de alimentación, y cuando están inactivos, dejan la salida flotante o en alta impedancia. Cuando el sensor PNP se activa, entrega voltaje positivo a la entrada IND. Este voltaje hace que fluya corriente desde IND, pasando por la resistencia R23, a través del LED del optoacoplador, y finalmente hacia tierra. La corriente que atraviesa el LED lo hace emitir luz, que es captada por el fototransistor ubicado en el lado aislado del optoacoplador.

Cuando el LED interno se ilumina, el fototransistor del optoacoplador comienza a conducir, conectando su colector con su emisor. Del lado de salida del optoacoplador, el colector del fototransistor está conectado a la alimentación de 3.3V a través de una resistencia pull-up R24 de  $2.4\text{k}\Omega$ , y desde ahí se toma la señal DOUT. El emisor del fototransistor está conectado a tierra. Cuando el fototransistor conduce debido a que el LED está encendido, conecta DOUT a tierra a través del colector-emisor, haciendo que la salida caiga a nivel bajo (0V).

Por el contrario, cuando el sensor PNP está inactivo, no hay voltaje en IND o está en un nivel muy bajo. Sin corriente fluyendo por el LED del optoacoplador, éste permanece apagado y el fototransistor queda en corte. Con el fototransistor abierto, la resistencia pull-up R24 mantiene la salida DOUT en nivel alto a 3.3V, lista para ser leída por el microcontrolador.

La ventaja fundamental de este diseño es el aislamiento galvánico completo que proporciona el optoacoplador. Esto significa que el circuito de entrada, donde puede haber voltajes industriales de 24V o más provenientes del sensor PNP, está completamente aislado eléctricamente del circuito de salida que opera a 3.3V y se conecta al ESP32. Esta separación física mediante transmisión óptica protege al microcontrolador de picos de voltaje, ruido eléctrico, diferencias de potencial de tierra, y corrientes de falla que podrían dañar componentes sensibles. Es ideal para aplicaciones industriales donde la confiabilidad y la protección del sistema de control son críticas.

## Entrada 0-10V.



Este circuito es una **entrada analógica de 0-10V** diseñada para adaptar señales industriales estándar de 0-10V a los niveles de voltaje que puede leer el ADC (Conversor Analógico-Digital) del ESP32, que opera en el rango de 0-3.3V.

El componente fundamental del circuito es un divisor de voltaje formado por dos resistencias en serie: R12 de  $20\text{k}\Omega$  y R13 de  $10\text{k}\Omega$ . Este divisor de voltaje funciona según el principio de que la corriente que fluye por ambas resistencias es la misma, y el voltaje total de entrada se divide proporcionalmente según los valores de cada resistencia. La señal de entrada IN\_0-10V puede variar desde 0V hasta 10V, dependiendo del sensor o dispositivo industrial conectado, y necesita ser reducida a un rango compatible con el ESP32.

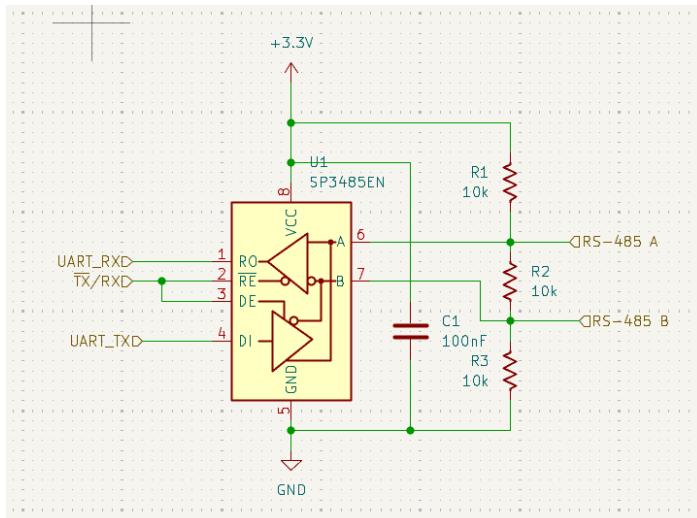
El funcionamiento matemático del divisor es directo. La resistencia total del circuito es  $20\text{k}\Omega + 10\text{k}\Omega = 30\text{k}\Omega$ . El voltaje de salida DADC\_IN\_3V3, tomado entre R13 y tierra, se calcula mediante la fórmula del divisor de voltaje:  $V_{out} = V_{in} \times (R13 / (R12 + R13))$ . Sustituyendo valores:  $V_{out} = V_{in} \times (10\text{k} / 30\text{k}) = V_{in} \times 0.333$ . Esto significa que cuando la entrada recibe 10V, la salida será aproximadamente 3.33V, justo en el límite superior del rango del ADC del ESP32. De manera similar, 0V en la entrada producirá 0V en la salida, y cualquier voltaje intermedio se escalará proporcionalmente.

El capacitor C2 de  $0.1\mu\text{F}$  ( $100\text{nF}$ ) conectado entre la salida DADC\_IN\_3V3 y tierra cumple una función crítica de filtrado. Las señales analógicas industriales a menudo vienen contaminadas con ruido eléctrico de alta frecuencia proveniente de motores, relés, interferencia electromagnética, o simplemente del cableado largo. Este capacitor actúa como filtro pasa-bajos, atenuando las componentes de alta frecuencia del ruido mientras deja pasar la señal real que varía más lentamente. Además, el capacitor ayuda a estabilizar la lectura del ADC, que toma muestras discretas del voltaje, proporcionando un reservorio de carga que mantiene el voltaje estable durante el proceso de muestreo.

Este tipo de circuito es estándar en automatización industrial donde muchos sensores y transductores generan señales analógicas de 0-10V para representar variables físicas como temperatura, presión, nivel, flujo, o posición. Ejemplos típicos incluyen transmisores de presión que dan 0V a presión cero y 10V a presión máxima, variadores de frecuencia que reportan su velocidad actual, o sensores de nivel que indican el porcentaje de llenado de un tanque. El circuito permite que el ESP32 lea estas señales directamente, convierta el voltaje a un valor digital

mediante su ADC interno, y luego lo procese para obtener el valor real de la variable física mediante una simple regla de tres o ecuación de escalado en el software.

### Entrada y Salida RS485



Este circuito es una **interfaz de comunicación RS-485** que permite al ESP32 comunicarse a través del protocolo industrial RS-485, convirtiendo las señales UART estándar del microcontrolador en señales diferenciales robustas capaces de transmitir datos a largas distancias con alta inmunidad al ruido.

El componente central es el transceiver SP3485EN (U1), un chip especializado que hace la conversión bidireccional entre señales UART TTL de 3.3V y señales diferenciales RS-485. Este transceiver tiene dos funciones principales: recibir datos desde el bus RS-485 y convertirlos a niveles UART que el ESP32 puede leer, y transmitir datos desde el ESP32 hacia el bus RS-485 convirtiéndolos a señales diferenciales. El chip se alimenta con 3.3V desde VCC y tiene su pin de tierra (GND) conectado apropiadamente para referenciar todas las señales.

Las señales del ESP32 se conectan al transceiver de manera específica. La señal UART\_RXD del ESP32 se conecta al pin RO (Receiver Output) del SP3485EN, que es donde el chip entrega los datos recibidos desde el bus RS-485. La señal UART\_TXD del ESP32 se conecta al pin DI (Driver Input) del transceiver, que es donde el chip recibe los datos que debe transmitir al bus. El pin TX/RXD es especialmente importante porque controla la dirección de la comunicación: cuando está en alto, el transceiver habilita el driver para transmitir, y cuando está en bajo, habilita el receptor para recibir datos. Este control de dirección es necesario porque RS-485 es half-duplex, lo que significa que solo un dispositivo puede transmitir a la vez en el bus compartido.

Los pines RE (Receiver Enable) y DE (Driver Enable) del transceiver están conectados juntos al pin TX/RXD del ESP32. Esta configuración simplifica el control porque ambos pines se manejan con una sola señal: cuando TX/RXD está en bajo, RE está activo (habilitando el receptor) y DE está inactivo (deshabilitando el driver), permitiendo recibir datos. Cuando TX/RXD está en alto, RE se deshabilita y DE se habilita, permitiendo transmitir datos. Este esquema de control es estándar en implementaciones RS-485 con microcontroladores.

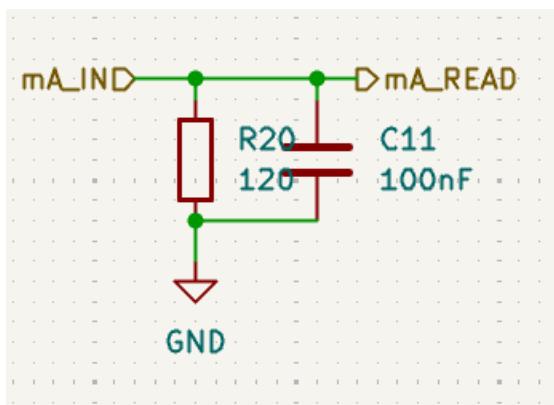
Las salidas diferenciales A y B del transceiver, conectadas a los pines 6 y 7 respectivamente, son el corazón de la robustez de RS-485. Estas dos líneas llevan la misma información pero con polaridades opuestas: cuando A está alto, B está bajo, y viceversa. El receptor en el otro extremo del cable lee la diferencia de voltaje entre ambas líneas en lugar del voltaje absoluto de cada una. Esto hace que el sistema sea extremadamente resistente al ruido electromagnético, porque cualquier interferencia afecta a ambas líneas por igual, y al restar una de la otra, el ruido se cancela mientras que la señal diferencial permanece intacta. Esta es la razón por la que RS-485 puede funcionar confiablemente en entornos industriales ruidosos y sobre cables de hasta 1200 metros.

Las resistencias de terminación R1, R2, y R3, todas de  $10\text{k}\Omega$ , junto con el capacitor C1 de  $100\text{nF}$ , forman la red de polarización y terminación del bus RS-485. Las resistencias R1 y R2 crean una red de bias que mantiene el bus en un estado definido cuando ningún dispositivo está transmitiendo, evitando estados flotantes que podrían interpretarse erróneamente como datos. R1 conecta la línea A hacia VCC mientras que R2 conecta la línea B hacia tierra a través de R3, estableciendo una diferencia de potencial pequeña pero suficiente para que el receptor detecte un estado idle válido. El capacitor C1 actúa como filtro de alta frecuencia entre las líneas A y B, ayudando a eliminar ruido de modo común y mejorando la calidad de la señal diferencial. En algunas configuraciones, especialmente en los extremos de un bus largo, se agregan resistencias de terminación de  $120\Omega$  entre A y B para igualar la impedancia característica del cable y prevenir reflexiones de señal, aunque en este esquema particular no aparecen incluidas.

Este circuito permite al ESP32 participar en redes industriales RS-485, que son omnipresentes en automatización para conectar PLCs, drives de motores, sensores inteligentes, medidores de energía, y otros dispositivos en redes multi-punto. Un solo par de cables puede conectar hasta 32 dispositivos en configuración estándar, o más con repetidores, todos compartiendo el mismo bus de comunicación mediante protocolos como Modbus RTU, que es uno de los más utilizados en la industria.

**NOTA:** Las entradas VAC 0-10V y GND, no se hicieron debido a limitaciones de diseño.

## Entrada 4-20mA



Este circuito convierte la corriente del lazo de 4-20mA en un voltaje que el ADC del ESP32 puede leer. La entrada mA\_IND recibe la corriente del sensor industrial, que fluye a través de la resistencia de sensado R20 de  $120\Omega$  hasta tierra. Según la ley de Ohm ( $V = I \times R$ ), cuando la corriente atraviesa esta resistencia, se desarrolla un voltaje proporcional a la corriente que está circulando.

El voltaje resultante se mide en el punto DmA\_READ, que se conecta directamente a un pin ADC del ESP32. Con una resistencia de  $120\Omega$ , los valores de voltaje para las corrientes típicas son: cuando el sensor entrega 4mA (valor mínimo del rango), el voltaje será  $4\text{mA} \times 120\Omega = 0.48\text{V}$ ; cuando el sensor entrega 20mA (valor máximo), el voltaje será  $20\text{mA} \times 120\Omega = 2.4\text{V}$ . Este rango de 0.48V a 2.4V está perfectamente dentro de los límites del ADC del ESP32, que puede leer de 0V a 3.3V, dejando un margen de seguridad adecuado.

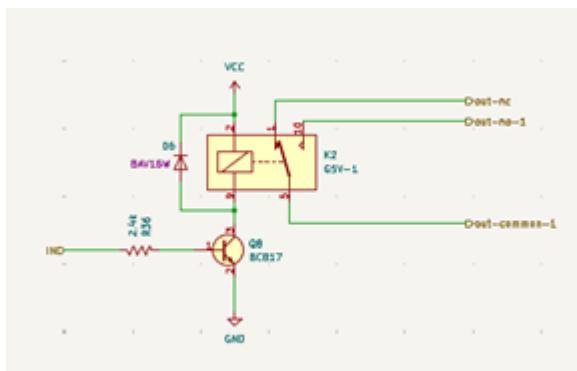
El capacitor C11 de 100nF está conectado en paralelo entre la salida DmA\_READ y tierra, actuando como filtro pasa-bajos. Las señales de 4-20mA en entornos industriales a menudo vienen acompañadas de ruido eléctrico de alta frecuencia causado por motores, contactores, interferencia electromagnética, o el propio cableado largo del lazo. El capacitor atenúa estas componentes de alta frecuencia mientras permite que la señal real, que varía lentamente, pase sin alteración. Además, estabiliza el voltaje durante el momento exacto en que el ADC toma la muestra, proporcionando una lectura más precisa y estable.

En el software del ESP32, debes leer el valor del ADC y convertirlo a corriente mediante una ecuación simple: Corriente(mA) = Voltaje\_leído /  $120\Omega$ . Luego, puedes escalar esta corriente al valor físico real que representa el sensor. Por ejemplo, si tienes un transmisor de presión de 0-100 PSI que envía 4-20mA, donde 4mA representa 0 PSI y 20mA representa 100 PSI, harías: Presión(PSI) =  $((\text{Corriente} - 4) / 16) \times 100$ .

Este diseño es estándar en la industria y funciona perfectamente con cualquier transmisor de 4-20mA, ya sea de temperatura, presión, nivel, flujo, humedad, o cualquier otra variable física medible.

**Nota: Los CT se diseñarán externamente.**

### Salidas a Relay



Este circuito representa un sistema de control de carga donde un microcontrolador ESP32 gobierna el encendido y apagado de un dispositivo de mayor potencia a través de un transistor. La idea fundamental es que el microcontrolador, que trabaja con señales digitales de bajo voltaje y corriente limitada, no puede alimentar directamente cargas que demandan más energía, por lo que necesita un intermediario que actúe como interruptor electrónico.

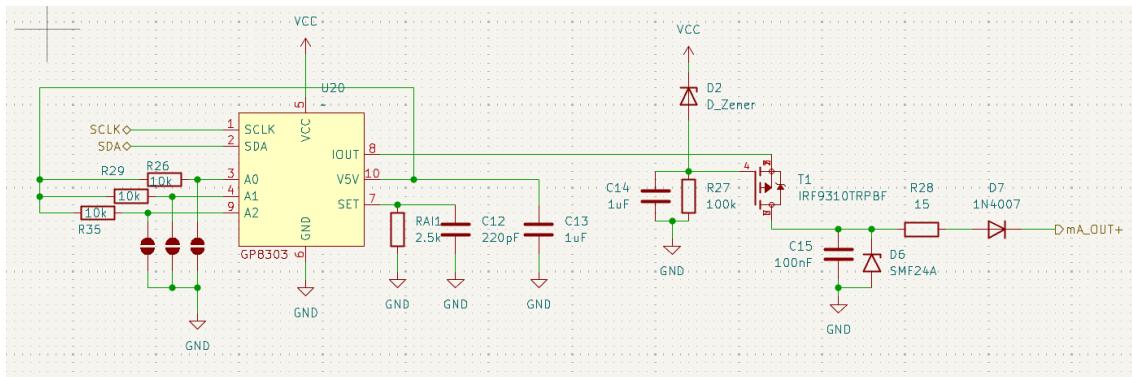
El corazón del control está en el ESP32, que genera una señal digital desde uno de sus pines GPIO. Esta señal viaja a través de una resistencia llamada  $R_{Service}$  hasta la base de un transistor NPN tipo 2N3904. La resistencia cumple una función crítica: limitar la corriente que entra a la base del transistor para evitar dañarlo, ya que la base es un componente delicado que solo necesita pequeñas corrientes para funcionar correctamente.

El transistor 2N3904 funciona como un interruptor controlado por corriente. Cuando el pin GPIO del ESP32 está en estado bajo (0V), no hay corriente fluyendo hacia la base del transistor, lo que mantiene al transistor en estado de corte, es decir, apagado. En esta condición, el transistor bloquea completamente el paso de corriente entre su colector y su emisor, dejando la carga desactivada. Por el contrario, cuando el GPIO pasa a estado alto (3.3V en el ESP32), comienza a fluir una pequeña corriente desde el pin hacia la base del transistor a través de la resistencia.

Esta corriente de base provoca que el transistor entre en saturación, lo que significa que se comporta como un interruptor cerrado. En este momento, el transistor permite el paso libre de corriente desde la fuente de alimentación VCC, a través de la carga conectada al colector, bajando por el colector-emisor del transistor, y finalmente llegando a tierra (GND). De esta manera, la carga recibe la energía necesaria para operar.

Como hemos mencionado en el manual, son dos entradas a Relay (una con 2 contactos, dígase NC, NO, y otra con un solo contacto NO), pero la hemos puesto así para que si en algún momento es necesario un cambio o 2 contactos, no halla que sustituir el Relay.

## Salida 4-20 mA.



Este es un circuito completo de **salida 4-20mA controlada digitalmente**, que permite al ESP32 generar corrientes precisas en el rango industrial estándar de 4-20mA para controlar actuadores, válvulas proporcionales, o enviar señales a otros dispositivos industriales.

### 3.1 Etapa de control digital (DAC)

El corazón del circuito es el DAC GP8303 (U20), un conversor digital-analógico de dos canales controlado mediante interfaz I2C. El ESP32 se comunica con este chip a través de las líneas SCLK (clock) y SDA (data), enviando comandos digitales que especifican el valor de corriente deseado. Las resistencias R29, R26, y R35 forman un divisor resistivo conectado a los pines de dirección A0, A1, y A2 del DAC, configurando la dirección I2C del dispositivo en el bus. Esto permite que múltiples DACs coexistan en el mismo bus I2C si fuera necesario. El DAC se alimenta desde VCC y tiene su pin de tierra correctamente conectado para referenciar todas las señales.

### 3.2 Conversión voltaje a corriente

El DAC genera un voltaje de salida proporcional al valor digital recibido desde el ESP32, que sale por el pin IOUT (pin 8). Este voltaje pasa por una red de filtrado formada por la resistencia RAI1 de 2.5kΩ y los capacitores C12 de 220pF y C13 de 1μF. Esta red elimina ruido de alta frecuencia y suaviza transiciones abruptas en el voltaje del DAC, produciendo una señal limpia y estable. El voltaje filtrado controla el gate del transistor MOSFET de canal P IRF9310TRPBF (T1).

### 3.3 Regulación activa de corriente

El transistor MOSFET T1 es el elemento activo que realmente controla la corriente de salida. Este es un transistor de canal P, lo que significa que conduce cuando su gate está a un voltaje más negativo que su source. El source del MOSFET está conectado a VCC a través del diodo Zener D2, mientras que el drain está conectado a la carga y luego a la resistencia de sensado R28. La resistencia R27 de 100kΩ entre gate y source mantiene el MOSFET apagado cuando no hay señal de control, y el capacitor C14 de 1μF en paralelo ayuda a estabilizar el voltaje de gate-source, previniendo oscilaciones.

El funcionamiento del control de corriente se basa en realimentación negativa. Cuando el voltaje del DAC aumenta, reduce la diferencia gate-source del MOSFET, permitiendo que fluya más corriente desde VCC, a través del MOSFET, pasando por la carga conectada en DmA\_OUT+, y finalmente a tierra a través de la resistencia de sensado R28 de 15Ω. Esta resistencia desarrolla

un voltaje proporcional a la corriente según  $V = I \times R$ . Si la corriente aumenta demasiado, el voltaje en R28 aumenta, lo que tiende a reducir la conducción del MOSFET, creando un lazo de realimentación que estabiliza la corriente en el valor deseado independientemente de cambios en la carga.

### 3.4 Protecciones del circuito

El diodo Zener D2 conectado entre VCC y el source del MOSFET limita el voltaje máximo que puede aparecer en el source, protegiendo al transistor de sobrevoltajes. El diodo D7 tipo 1N4007 en serie con la salida proporciona protección contra polaridad inversa, evitando que corriente fluya hacia atrás si la carga se conecta incorrectamente. El diodo TVS D6 modelo SMF24A protege contra transitorios de voltaje y picos que podrían dañar el circuito, especialmente importantes en entornos industriales donde los cables pueden captar descargas electrostáticas o interferencia electromagnética intensa. El capacitor C15 de 100nF en la salida filtra ruido residual de alta frecuencia que podría acoplarse al lazo de corriente.

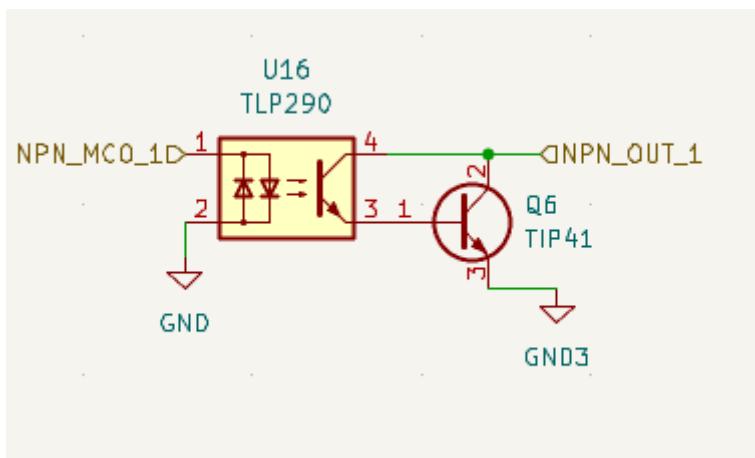
### 3.5 Rango de operación

Con este diseño, el ESP32 envía un valor digital al DAC GP8303 especificando la corriente deseada. El DAC convierte este valor a un voltaje analógico que ajusta la conducción del MOSFET, resultando en una corriente precisa entre 4mA y 20mA fluyendo a través de la carga conectada. La resistencia de sensado R28 de  $15\Omega$  permite monitorear la corriente: con 4mA, el voltaje será 60mV; con 20mA, será 300mV. Estos valores son pequeños, minimizando la pérdida de potencia en la resistencia. El rango de 4-20mA es estándar industrial porque el valor mínimo de 4mA permite detectar cables rotos o fallas, algo que no sería posible con un rango de 0-20mA donde 0mA podría significar tanto "sensor al mínimo" como "sensor desconectado".

### 3.6 Aplicaciones típicas

Este circuito permite controlar válvulas proporcionales que regulan el flujo de líquidos o gases según la corriente recibida, variadores de frecuencia que ajustan la velocidad de motores basándose en la señal de 4-20mA, posicionadores de actuadores que mueven válvulas o compuertas a posiciones específicas, o cualquier dispositivo industrial que acepte comandos analógicos en formato 4-20mA. Es el complemento perfecto para la entrada 4-20mA que vimos antes, permitiendo que el ESP32 no solo lea sensores industriales, sino que también controle actuadores en el mismo lenguaje eléctrico que habla toda la industria.

## Salida NPN Optoacoplada



Este es un circuito de **salida NPN optoacoplada**, diseñado para controlar cargas de corriente media-alta manteniendo aislamiento galvánico completo entre el circuito de control del ESP32 y la carga externa.

El circuito comienza con el optoacoplador TLP290 (U16), que proporciona el aislamiento eléctrico esencial entre el lado de control y el lado de potencia. La señal de control NPN\_MCO\_ID viene desde un pin GPIO del ESP32 y se conecta al LED interno del optoacoplador en el pin 1, mientras que el pin 2 va a tierra. Cuando el ESP32 envía una señal alta en NPN\_MCO\_ID, fluye corriente a través del LED del optoacoplador, haciendo que este emita luz. Esta luz atraviesa el material aislante interno del optoacoplador y es detectada por el fototransistor ubicado en el lado completamente aislado del componente, en los pines 3 y 4.

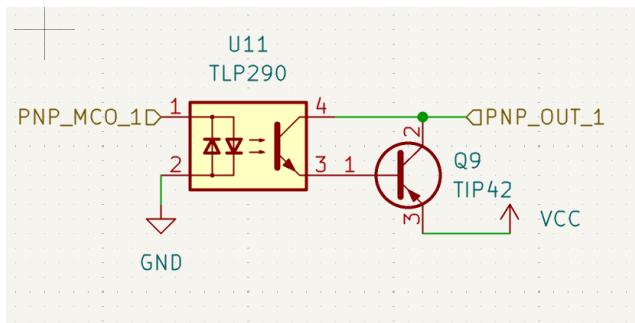
Cuando el fototransistor del optoacoplador detecta la luz del LED, comienza a conducir, permitiendo el paso de corriente desde su colector hacia su emisor. Este fototransistor no maneja directamente la carga final, sino que actúa como driver de un transistor de potencia más grande. El colector del fototransistor se conecta a la base del transistor NPN TIP41 (Q6), proporcionando la corriente de base necesaria para activarlo. El TIP41 es un transistor bipolar NPN de potencia capaz de manejar corrientes considerablemente mayores que el pequeño fototransistor del optoacoplador, típicamente hasta 6A de corriente de colector continua.

El funcionamiento del TIP41 es directo: cuando recibe corriente de base desde el fototransistor del optoacoplador, entra en saturación y se comporta como un interruptor cerrado entre su colector y su emisor. El colector está conectado a la salida GNPN\_OUT\_1, que es donde se conectaría la carga externa, mientras que el emisor está conectado a GND3, que es la tierra del lado de potencia. Cuando el transistor conduce, conecta efectivamente GNPN\_OUT\_1 a tierra a través del colector-emisor, permitiendo que fluya corriente desde la fuente de alimentación de la carga, a través de la carga misma, y finalmente hacia tierra a través del TIP41.

Este tipo de configuración emula exactamente el comportamiento de un sensor NPN industrial. Los sensores NPN funcionan conectando su salida a tierra cuando están activos, dejándola flotante cuando están inactivos. En este caso, el circuito permite que el ESP32 simule un sensor NPN de alta corriente, capaz de hundir corriente desde cargas que operan a voltajes diferentes al del microcontrolador. La ventaja crítica del aislamiento galvánico es que la tierra GND3 del lado de potencia puede estar a un potencial completamente diferente de la tierra GND del lado de control sin riesgo de daño. Esto es especialmente importante cuando se controlan cargas alimentadas por fuentes industriales de 24V o más, o cuando hay múltiples sistemas con diferentes referencias de tierra.

Las aplicaciones típicas de este circuito incluyen el control de relés, contactores, electroválvulas, luces de señalización industrial, entradas de PLCs que esperan señales NPN, arrancadores de motores, o cualquier dispositivo que requiera una señal de tipo sink (hundir corriente a tierra). El TIP41 puede manejar cargas que consumen varios amperios, y si se necesita aún más corriente, podría reemplazarse por transistores más robustos como el TIP120 o incluso MOSFETs de potencia. El aislamiento del optoacoplador protege al ESP32 de picos de voltaje, ruido eléctrico, y diferencias de potencial de tierra que son comunes en entornos industriales, garantizando operación confiable y segura del sistema de control.

## Salida PNP



Este es un circuito de **salida PNP optoacoplada**, diseñado para emular el comportamiento de un sensor PNP industrial de alta corriente, manteniendo el aislamiento galvánico entre el ESP32 y la carga externa.

El circuito utiliza el mismo optoacoplador TLP290 (U11) como elemento de aislamiento. La señal de control PNP\_MCO\_1D viene desde un pin GPIO del ESP32 y se conecta al ánodo del LED interno del optoacoplador en el pin 1, mientras que el cátodo en el pin 2 va a tierra. Cuando el ESP32 activa la señal enviando un nivel alto, fluye corriente a través del LED del optoacoplador, haciéndolo emitir luz. Esta luz atraviesa el material aislante interno y es captada por el fototransistor ubicado en el lado completamente aislado del componente.

Cuando el fototransistor detecta luz, comienza a conducir, permitiendo que fluya corriente desde su colector en el pin 4 hacia su emisor en el pin 3. Sin embargo, aquí viene la diferencia fundamental con el circuito NPN anterior: en lugar de usar un transistor NPN de potencia, se utiliza un transistor PNP TIP42 (Q9). El TIP42 es un transistor bipolar PNP de potencia capaz de manejar corrientes de hasta 6A, complementario al TIP41 que vimos en la configuración NPN.

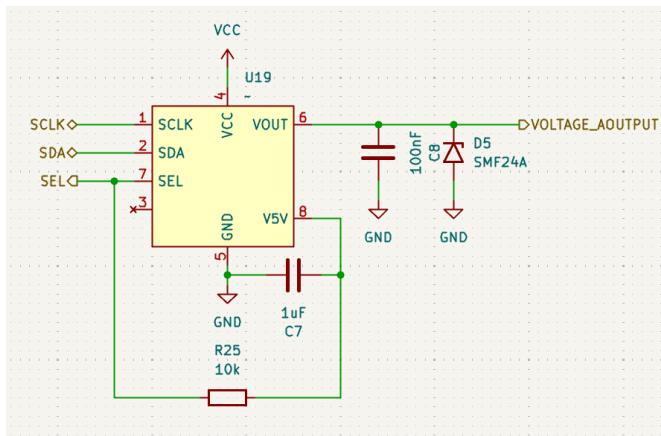
La configuración del transistor PNP es inversa a la del NPN. El emisor del TIP42 está conectado a VCC, que es el voltaje positivo de alimentación de la carga, mientras que el colector se conecta a la salida PNP\_OUT\_1. La base del TIP42 se conecta al colector del fototransistor del optoacoplador. Cuando el fototransistor conduce, drena corriente desde la base del TIP42 hacia tierra a través del emisor del fototransistor, creando una diferencia de potencial negativa entre la base y el emisor del PNP. Esta condición hace que el transistor PNP entre en saturación y conduzca.

Cuando el TIP42 conduce, conecta efectivamente su colector al voltaje del emisor, es decir, conecta PNP\_OUT\_1 a VCC a través del colector-emisor del transistor. Esto permite que fluya corriente desde VCC, a través del colector-emisor del TIP42, hacia la carga conectada en PNP\_OUT\_1, y finalmente hacia tierra a través de la carga misma. Este comportamiento replica exactamente la operación de un sensor PNP industrial, que proporciona voltaje positivo a su salida cuando está activo.

La diferencia fundamental entre este circuito y el NPN anterior radica en la dirección del flujo de corriente y la polaridad de la señal. Los sensores NPN hunden corriente hacia tierra cuando están activos, funcionando como sink. Los sensores PNP entregan corriente desde su alimentación positiva cuando están activos, funcionando como source. Esta configuración PNP es especialmente útil en sistemas industriales donde los dispositivos controlados esperan recibir voltaje positivo para activarse, como ciertos tipos de entradas de PLCs configuradas para lógica PNP, módulos de expansión, luces indicadoras que requieren alimentación positiva, o relés que necesitan voltaje en lugar de conexión a tierra.

El aislamiento galvánico proporcionado por el optoacoplador es igual de crítico aquí que en la configuración NPN, protegiendo al ESP32 de sobrevoltajes, ruido eléctrico, y diferencias de potencial entre las tierras del lado de control y el lado de potencia. Esto permite controlar cargas alimentadas por fuentes industriales de 24V o superiores con total seguridad para el microcontrolador. La combinación de ambos circuitos, NPN y PNP, en un mismo sistema proporciona máxima flexibilidad para interactuar con cualquier tipo de dispositivo industrial, sin importar si requiere lógica sink o source.

### Salida 0-10v



Un circuito de **salida 0-10V** permite al ESP32 generar señales analógicas en el rango industrial estándar de 0-10V para controlar actuadores, variadores de frecuencia, válvulas proporcionales, o sistemas de iluminación regulable (DALI, DMX).

#### 3.1 Componentes principales

El circuito típicamente utiliza un DAC (Conversor Digital-Analógico) controlado por el ESP32 mediante I2C o SPI, similar al GP8303 que viste en el circuito de salida 4-20mA. Sin embargo, en lugar de convertir a corriente, aquí se genera un voltaje directo. Como el DAC generalmente solo produce hasta 2.5V o 5V máximo, se necesita un amplificador operacional configurado como amplificador no inversor para escalar la señal hasta los 10V requeridos.

#### 3.2 Etapa de amplificación

El op-amp más común para esta aplicación es el TL081 o similar, que puede operar con alimentación de riel único hasta 15V o más. El DAC genera un voltaje de 0V a 2.5V proporcional al valor digital recibido del ESP32. Este voltaje se conecta a la entrada no inversora del op-amp, mientras que la entrada inversora se conecta a un divisor resistivo que forma la red de realimentación negativa. La ganancia del amplificador se ajusta mediante dos resistencias para multiplicar por 4 la señal de entrada, convirtiendo 0-2.5V en 0-10V a la salida. Por ejemplo, con una resistencia de realimentación de  $30\text{k}\Omega$  y una resistencia a tierra de  $10\text{k}\Omega$ , la ganancia es  $1 + (30\text{k}/10\text{k}) = 4$ .

### 3.3 Protecciones y filtrado

A la salida del op-amp se coloca una resistencia serie de  $100\Omega$ - $1k\Omega$  para limitar corriente ante cortocircuitos accidentales, seguida de un capacitor de  $100nF$  a tierra para filtrar ruido de alta frecuencia. Un diodo Zener de 10V puede agregarse entre la salida y tierra para recortar cualquier sobrevoltaje que pudiera generarse por mal funcionamiento del circuito. Adicionalmente, un diodo Schottky en antiparalelo previene voltajes negativos que podrían dañar equipos sensibles conectados.

### 3.4 Alimentación del circuito

El op-amp requiere alimentación de al menos 12V-15V para poder generar cómodamente los 10V de salida sin saturarse cerca del riel de alimentación. Esta alimentación puede venir de un regulador buck-boost si solo tienes 3.3V o 5V disponibles, o de una fuente dedicada de 15V. Es importante que esta fuente esté bien filtrada con capacitores de  $100\mu F$  y  $100nF$  en paralelo para evitar que el ruido de conmutación del op-amp afecte la precisión de la señal.

### 3.5 Calibración y linealidad

Para máxima precisión, el circuito puede incluir un potenciómetro multivuelta de ajuste de ganancia y otro de offset, permitiendo calibrar exactamente los puntos de 0V y 10V. Sin esta calibración, las tolerancias de las resistencias (típicamente  $\pm 1\%$  o  $\pm 5\%$ ) pueden causar errores de varios cientos de milivoltios. En aplicaciones críticas, se utilizan resistencias de precisión  $\pm 0.1\%$  y referencias de voltaje estables como la TL431 para garantizar que el DAC recibe un voltaje de referencia exacto.

### 3.6 Control desde el ESP32

En el software, el ESP32 envía comandos al DAC especificando un valor digital de 0 a 4095 (si es un DAC de 12 bits) o 0 a 65535 (si es de 16 bits). Este valor se traduce linealmente al voltaje de salida: valor 0 = 0V, valor máximo = 10V. Por ejemplo, para generar 5V con un DAC de 12 bits, enviarías el valor 2048. Para generar 7.3V, calcularías  $(7.3/10) \times 4095 = 2989$  y enviarías ese valor.

### 3.7 Aplicaciones típicas

Las salidas 0-10V son estándar para controlar variadores de frecuencia donde 0V significa motor detenido y 10V significa velocidad máxima, válvulas proporcionales que regulan flujo de líquidos o gases según el voltaje aplicado, sistemas de iluminación regulable donde el voltaje controla la intensidad luminosa, posicionadores de actuadores lineales, control de temperatura en sistemas HVAC, y cualquier equipo industrial que acepte comandos analógicos en formato 0-10V. Es particularmente popular en Europa, mientras que en Estados Unidos es más común el estándar 0-5V.

### 3.8 Capacidad de carga

Un op-amp típico puede entregar 10-25mA de corriente de salida, suficiente para la mayoría de entradas de control industrial que presentan impedancias de entrada superiores a  $10k\Omega$ . Si necesitas más corriente, puedes agregar un seguidor de voltaje (buffer) con transistores discretos o un op-amp de mayor corriente como el LM358 en configuración de driver. Para cargas capacitivas grandes, una resistencia serie pequeña de  $10-50\Omega$  en la salida del op-amp previene oscilaciones causadas por el lazo de realimentación interactuando con la capacitancia de la carga. ¿Tienes algún circuito de salida 0-10V específico que quieras que analice, o necesitas ayuda diseñando uno para tu aplicación?

### 3.9 Comunicaciones:

- **RS-485:** Terminales (A+, B-) para comunicación serial industrial bajo protocolo **Modbus RTU**.
- **Wi-Fi / Bluetooth:** Antena integrada en el módulo ESP32-S3 para conectividad inalámbrica (Modbus TCP, MQTT).

**Tabla de asignación de pines**

N. <sup>º</sup> Pin	Nombre GPIO	Etiqueta en Diagrama (Net Label)	Descripción / Función Probable
1	GND	GND	Tierra (Ground)
2	3V3	+3.3V	Alimentación 3.3V
3	EN	EN	Enable (Reset del Chip)
8	IO15	IN_NPN2D	Entrada NPN Directa 2
9	IO16	(Cableado sin etiqueta visible)	Possiblemente vinculado a entradas NPN
10	IO17	micro_rs485_tx	Transmisión Serial RS-485
11	IO18	micro_rs485_rx	Recepción Serial RS-485
12	IO8	IN_PNP1D	Entrada PNP Directa 1
13	IO19	USB-	Señal USB Diferencial Negativa (D-)
14	IO20	USB+	Señal USB Diferencial Positiva (D+)
16	IO46	Out_4-20mA_2	Salida de Corriente 2 (4- 20mA)

N. <sup>o</sup> Pin	Nombre GPIO	Etiqueta en Diagrama (Net Label)	Descripción / Función Probable
<b>17</b>	IO9	IN_PNP2D	Entrada PNP Directa 2
<b>18</b>	IO10	IN_NPN2_OptoD	Entrada NPN Optoacoplada 2
<b>19</b>	IO11	IN_NPN1_OptoD	Entrada NPN Optoacoplada 1
<b>20</b>	IO12	IN_PNP2_OptoD	Entrada PNP Optoacoplada 2
<b>21</b>	IO13	IN_PNP1_OptoD	Entrada PNP Optoacoplada 1
<b>22</b>	IO14	IN_NPND	Entrada NPN Directa 1
<b>23</b>	IO21	PNP_OUT1	Salida Transistor PNP 1
<b>24</b>	IO47	ADC_IN_3V3	Entrada Analógica (ADC)
<b>25</b>	IO48	micro1_rs485_control	Control de Flujo RS-485 (DE/RE)
<b>26</b>	IO45	Out_4-20mA	Salida de Corriente 1 (4-20mA)
<b>27</b>	IO0	SCL	Reloj I2C (Para sensores/energía)
<b>28</b>	IO35	PNP_OUT2	Salida Transistor PNP 2
<b>29</b>	IO36	PNP_OUT3	Salida Transistor PNP 3

N.º Pin	Nombre GPIO	Etiqueta en Diagrama (Net Label)	Descripción / Función Probable
30	IO37	Opto_NPN	Salida NPN Optoacoplada 1
31	IO38	Opto_NPN2	Salida NPN Optoacoplada 2
32	IO39	OUT_RELAY1	Control Relé 1
33	IO40	OUT_RELAY2	Control Relé 2
34	IO41	NPN_OUT1	Salida Transistor NPN 1
35	IO42	NPN_OUT2	Salida Transistor NPN 2
39	IO1	SDA	Datos I2C (Para sensores/energía)

### 3.10 CARACTERÍSTICAS TÉCNICAS

Parámetro	Especificación
Microcontrolador	Espressif <b>ESP32-S3</b> (Dual Core, 240 MHz)
Voltaje de Operación	24 VDC (Nominal)
Entradas Digitales	8 x Optoaisladas (Configurables PNP/NPN)
Salidas Digitales	8 x (Relé 5A / Transistor NPN-PNP)
Entradas Analógicas	4 x (0-10V / 4-20mA), Resolución 12-bits
Salidas Analógicas	0-10V / 4-20mA (Control proporcional)
Medición de Energía	IC dedicado (I2C/SPI) para medición de CT y Voltaje AC
Puertos de Comunicación	USB-C (Prog), RS-485 (Industrial), Wi-Fi (IoT)
Protocolos Soportados	Modbus RTU/TCP, MQTT, HTTP/HTTPS
Tiempo de Ciclo	< 10 ms (Ejecución de lógica)
Temperatura de Operación	Ambiente de laboratorio (aprox. -10°C a 60°C)

### 3.11 DIMENSIONES Y MECÁNICA

I **PLC 4 UNI** utiliza una carcasa industrial de perfil bajo diseñada para montaje en armarios de control.

- **Modelo de Carcasa:** Serie 23-156C (Estándar Industrial).
- **Dimensiones Totales:**
  - **Largo (L):** 107.0 mm
  - **Ancho (W):** 87.7 mm
  - **Profundidad (H):** 59.0 mm (Incluyendo clip de sujeción).
- **Sistema de Montaje:**
  - **Riel DIN:** Compatible con riel estándar de **35 mm** (EN 50022).
  - **Fijación Mural:** Dispone de pestañas para tornillos en las esquinas traseras (opcional si se retira el clip).
- **Material de la carcasa:** Plástico (PLA/ABS) impreso en 3D o inyectado.

### 3.12 SOLUCIÓN DE PROBLEMAS (TROUBLESHOOTING)

Síntoma	Causa Probable	Solución Sugerida
<b>El LED PWR no enciende</b>	1. Fuente de alimentación desconectada.  2. Polaridad invertida.  3. Fusible PTC activado.	1. Verifique que lleguen 24VDC a las borneras 1 y 2.  2. Corrija el cableado (+/-).  3. Desconecte todo por 30 seg para rearmar el fusible.
<b>No hay comunicación Wi-Fi</b>	1. Credenciales incorrectas.  2. Señal débil.  3. PLC en modo AP.	1. Mantenga presionado BOOT 5 seg para reiniciar a fábrica.  2. Acerque el PLC al router o instale una antena externa.  3. Verifique si aparece la red PLC4UNI-SETUP.
<b>Las entradas digitales no detectan señal</b>	1. Configuración S/S incorrecta.  2. Sensor dañado.	1. Verifique si el terminal S/S está conectado a GND (para sensores PNP) o a 24V (para NPN).  2. Mida el voltaje en la bornera de entrada con un multímetro.

Síntoma	Causa Probable	Solución Sugerida
<b>Salida de Relé no activa la carga</b>	1. Lógica del programa errónea.  2. Carga desconectada.  3. Relé dañado por sobrecorriente.	1. Verifique que el LED de la salida (Qx) encienda en la placa.  2. Revise el cableado de la carga (110/220V).  3. Si el LED enciende pero el relé no suena ("clic"), el contacto puede estar soldado/dañado.
<b>Lectura de Corriente (CT) errónea</b>	1. CT mal conectado.  2. Rango excedido.	1. Asegúrese de que el transformador abrace <b>solo uno</b> de los cables de fase, no el cable completo (fase+neutro).

### 3.13 MANTENIMIENTO PREVENTIVO

Para garantizar la vida útil del equipo en el laboratorio:

- Limpieza:** Limpie la carcasa únicamente con un paño seco. No utilice solventes ni alcohol que puedan dañar el plástico ABS/PPO.
- Apriete de Borneras:** Las vibraciones pueden aflojar los tornillos. Verifique el apriete de las conexiones cada 6 meses.
- Inspección Visual:** Revise periódicamente que no haya cables pelados, signos de sobrecalentamiento (decoloración) en los conectores de relé o acumulación excesiva de polvo dentro de las ranuras de ventilación.

### 3.14 DISPOSICIÓN FINAL Y RECICLAJE (WEEE)

El **PLC 4 UNI** contiene componentes electrónicos que no deben desecharse en la basura común.

- **Símbolo del Contenedor Tachado:** Indica que al final de su vida útil, el producto debe ser llevado a un punto de recogida selectiva de residuos electrónicos (RAEE).
- **Componentes Reciclables:** La carcasa plástica (ABS/PC) es 100% recicitable. La PCB contiene metales valiosos (Cobre, Oro) recuperables en plantas especializadas.
- **Compromiso:** Como proyecto educativo, fomentamos la responsabilidad ambiental. Por favor, disponga de este equipo de manera responsable.

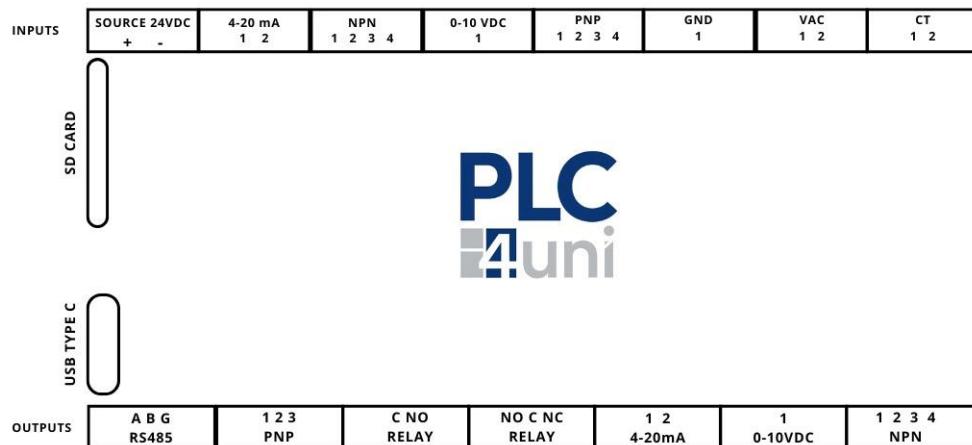
### 3.15 INFORMACIÓN LEGAL Y CRÉDITOS

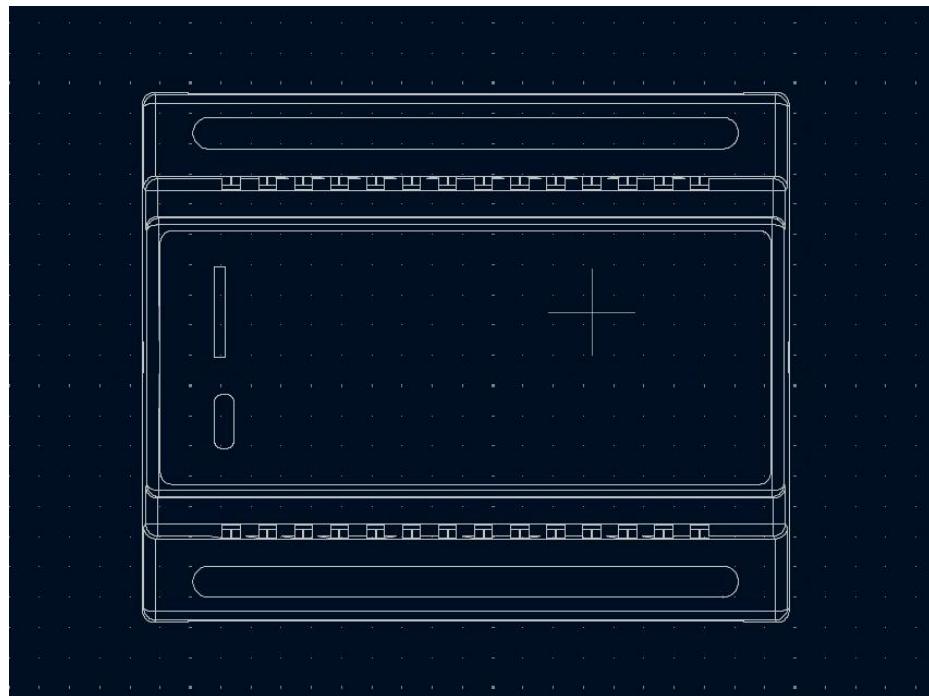
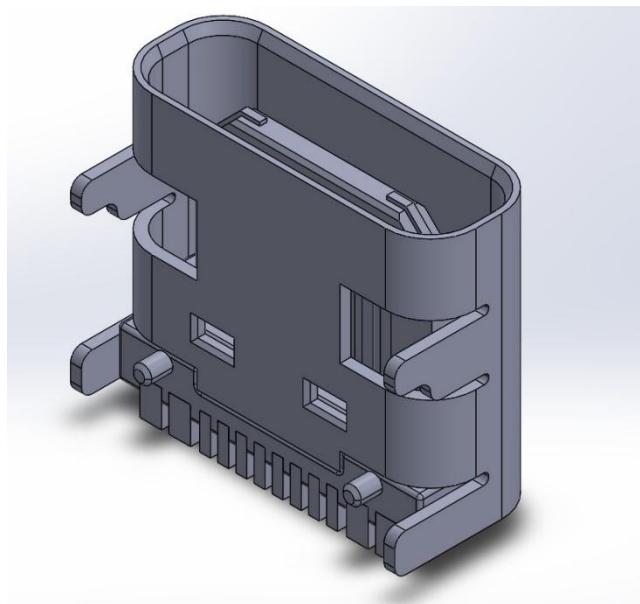
- **Licencia de Hardware:** Este hardware se distribuye bajo la licencia **CERN OHL v1.2**. Usted es libre de estudiar, modificar y fabricar este diseño.
- **Licencia de Software:** El firmware base y las librerías se distribuyen bajo licencia **MIT**.

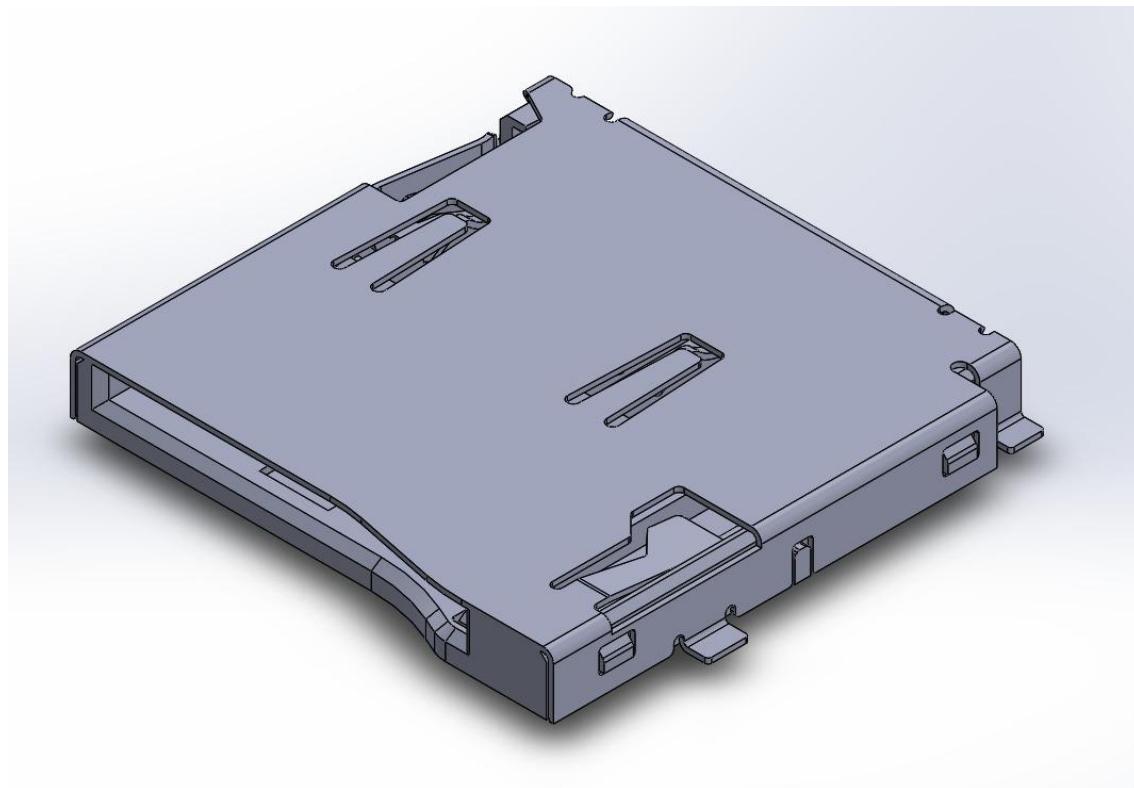
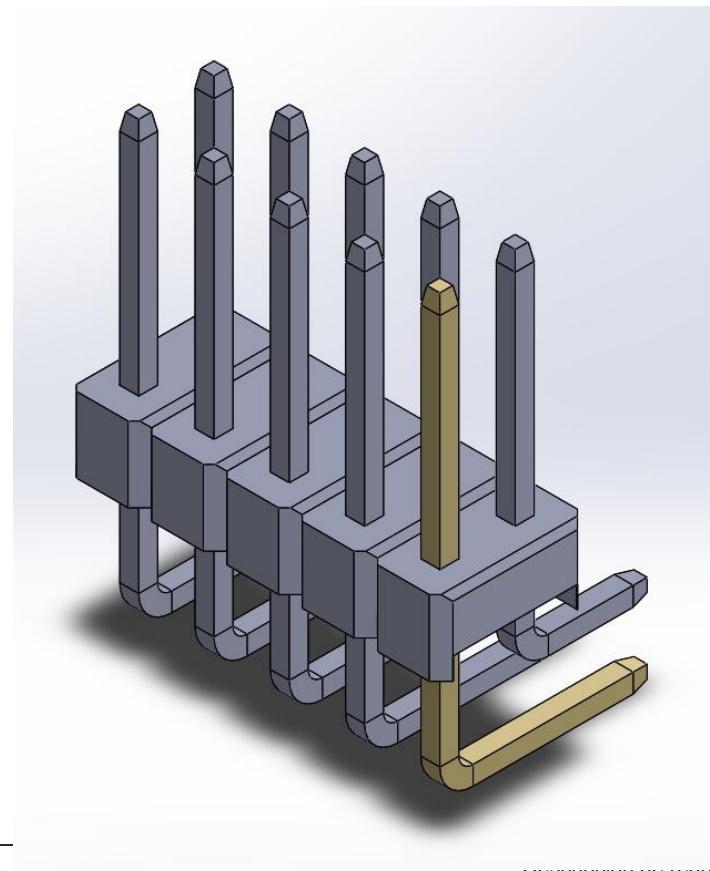
**Créditos:**

- Desarrollado por: [PLC4UNI]
- Institución: **Instituto Tecnológico de Las Américas (ITLA)**

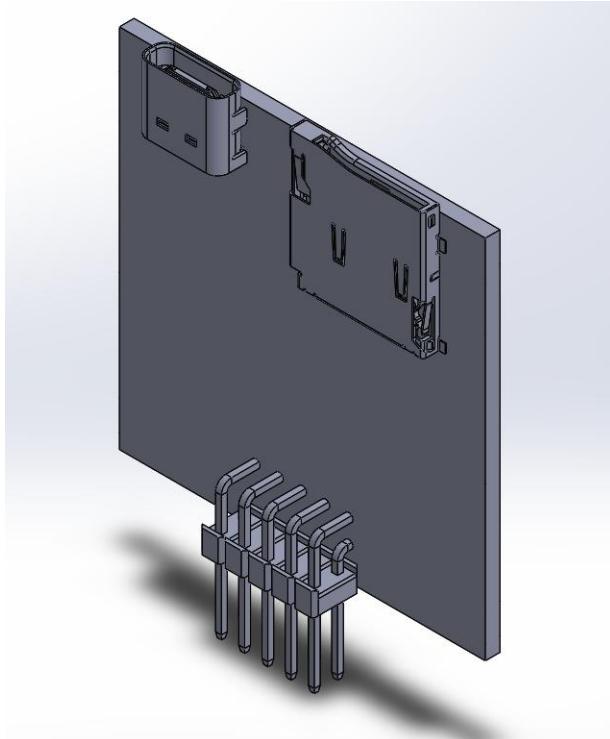
### CAPITULO III. Diseños

**Información del producto****Etiqueta**

*DXF**USB*

*SD CAR**Pin Headers para conexión de Módulos*

*USB, pin Headers y SD CAR*



## 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.*

### Apéndice A — Repositorio oficial del proyecto

Con el objetivo de garantizar la transparencia, la trazabilidad y la colaboración académica, el proyecto PLC4UNI dispone de un repositorio oficial en línea. En este espacio se almacenan los diagramas electrónicos, el firmware, la documentación técnica y las versiones sucesivas del desarrollo. De este modo, cualquier miembro de la comunidad educativa puede auditar los avances, proponer mejoras y contribuir activamente.

- Enlace al repositorio: mira el repositorio:  
[https://github.com/FranjelisLuciano/Franjelis\\_Luciano\\_PLC4UNI](https://github.com/FranjelisLuciano/Franjelis_Luciano_PLC4UNI)
- 

### Apéndice B — Canales de difusión y redes sociales

Para fortalecer la **visibilidad y el impacto académico del proyecto**, se han establecido distintos canales de difusión, orientados tanto a la comunidad universitaria como a espacios más amplios de intercambio:

- Página oficial de la asignatura en la plataforma de la universidad, donde se publicarán los avances formales y entregas.
- Grupo de WhatsApp del curso de Mecatrónica, con fines de **coordinación interna** y resolución rápida de incidencias.
- Publicaciones en redes sociales institucionales (Facebook, Instagram, LinkedIn académico), buscando **difundir logros y atraer colaboraciones externas**.

### Apéndice C — Cronograma resumido

El desarrollo del PLC4UNI sigue un cronograma progresivo, que articula tanto la planificación técnica como la pedagógica. A continuación, se muestra un esquema de referencia:

- **Semana 1:** Presentación del proyecto y establecimiento del objetivo general.
- **Semana 2:** Identificación de requisitos y entrega del documento SRS inicial.
- **Semana 3:** Selección de los componentes base y diseño preliminar de la arquitectura.
- **Semana 4:** Elaboración de los primeros esquemáticos y retroalimentación con el docente.
- **Semana 5:** Revisión de los esquemáticos y preparación del diseño de la PCB.
- **Semana 6:** Finalización de la PCB y validación previa a fabricación.
- **Semana 7:** Adquisición de componentes y gestión de la PCB.
- **Semana 8:** Montaje inicial del prototipo en laboratorio.
- **Semana 9:** Pruebas de hardware: validación de entradas, salidas y fuentes de alimentación.
- **Semana 10:** Desarrollo de firmware básico (gestión de E/S digitales y analógicas).
- **Semana 11:** Integración de firmware avanzado (comunicación RS-485, Wi-Fi).
- **Semana 12:** Validación de requisitos funcionales y no funcionales.
- **Semana 13:** Corrección de fallos, refinamiento de hardware/firmware y creación de manuales.
- **Semana 14:** Presentación final, entrega de documentación y defensa académica del proyecto.

#### Apéndice D — Contacto del equipo

Para fines de **coordinación académica y soporte institucional**, los responsables del proyecto PLC4UNI son:

- **Director del proyecto:** Profesor Carlos Antonio Pichardo Viuque — correo institucional: cpichardo@itla.edu.do
- **Integrante del equipo de desarrollo:** Franjelis Luciano — correo: 20231509@itla.edu.do