

IMPLEMENTACIÓN DE UN SISTEMA PARA CONTROL DE SUPERVISIÓN Y  
ADQUISICIÓN DE DATOS DE LA CELDA MECATRÓNICA DE  
FISCHERTECHNIK DEL LABORATORIO DE AUTOMATIZACIÓN UTILIZANDO  
HARDWARE Y SOFTWARE LIBRE

NEIL SEBASTIAN CASTRO CAICEDO  
HUBERT ARMANDO DELGADO MAESTRE

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA  
FACULTAD DE INGENIERÍA  
PROGRAMA DE INGENIERÍA MECATRÓNICA  
BUCARAMANGA

2024

IMPLEMENTACIÓN DE UN SISTEMA PARA CONTROL DE SUPERVISIÓN Y  
ADQUISICIÓN DE DATOS DE LA CELDA MECATRÓNICA DE  
FISCHERTECHNIK DEL LABORATORIO DE AUTOMATIZACIÓN UTILIZANDO  
HARDWARE Y SOFTWARE LIBRE

NEIL SEBASTIAN CASTRO CAICEDO  
HUBERT ARMANDO DELGADO MAESTRE

TRABAJO DE GRADO

DIRECTOR  
NAYIBE CHIO CHO  
CODIRECTOR  
HERNÁN GONZÁLEZ ACUÑA

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA  
FACULTAD DE INGENIERÍA  
PROGRAMA DE INGENIERÍA MECATRÓNICA  
BUCARAMANGA

2024

## CONTENIDO

1 INTRODUCCIÓN .....	1
1.2 RESUMEN .....	1
1.3 PLANTEAMIENTO DEL PROBLEMA .....	2
1.4 CONTEXTO Y JUSTIFICACIÓN .....	3
1.5 ANTECEDENTES .....	6
1.6 OBJETIVOS .....	9
1.6.1 GENERAL .....	9
1.6.2 ESPECIFICOS .....	9
2 MARCO TEÓRICO .....	1
2.1 CONTROLADOR LÓGICO PROGRAMABLES .....	1
2.2 HARDWARE LIBRE Y MICROCONTROLADORES .....	2
2.3 INDUSTRIA 4.0 Y TENDENCIAS ACTUALES .....	3
2.4 PROTOCOLOS DE COMUNICACIÓN .....	4
2.5 SISTEMAS SCADA .....	5
2.6 TECNOLOGÍAS WEB.....	5
2.6.1 FRONTEND.....	5
2.6.2 BACKEND Y BASES DE DATOS .....	6
2.7 CELDAS FISCHERTECHNIK .....	7
2.8 NORMATIVIDAD .....	8
2.8.1 DISEÑO DE PCB .....	8
2.8.2 LICENCIAS DE SOFTWARE LIBRE .....	9
3. METODOLOGÍA .....	1
4. RECOPILACIÓN DE INFORMACIÓN Y ANALISIS DE REQUERIMIENTOS .....	1

4.1 RECOLGIDA DE INFORMACIÓN CONTROLADOR COMERCIALES ....	1
4.2 RECOLGIDA DE INFORMACIÓN PLATAFORMAS SCADA .....	2
4.3 REQUERIMIENTOS CELDA MECATRÓNICA FISCHERTECHNIK .....	3
4.4 REQUERIMIENTOS DEL CONTROLADOR LÓGICO PROGRAMABLE.....	4
4.5 REQUERIMIENTOS DE LA PLATAFORMA SCADA WEB.....	5
4.6 ESPECIFICACIONES DEL SISTEMA .....	6
4.6.1 ESPECIFICACIONES DEL CONTROLADOR .....	6
4.6.2 ESPECIFICACIONES DE LA PLATAFORMA.....	7
5. DISEÑO Y FABRICACIÓN CONTROLADOR .....	1
5.1 SELECCIÓN DE COMPONENTES .....	1
5.2 DISEÑO DE CIRCUITOS ELECTRONICOS .....	2
5.2.1 ENTRADAS DIGITALES .....	3
5.2.2 ENTRADAS ANALÓGICAS .....	6
5.2.3 SALIDAS ANALÓGICAS .....	8
5.2.4 SALIDAS MOTORES PWM.....	10
5.2.4 SALIDAS DIGITALES RELÉ.....	12
5.5 FABRICACIÓN PCB.....	13
6. DISEÑO 3D Y FABRICACIÓN .....	1
6.1.1 FABRICACIÓN CARCASA CONTROLADOR.....	2
6.2 DISEÑO Y VALIDACIÓN CELDAS FISCHERTECHNIK.....	4
6.3 INTEGRACIÓN DEL SISTEMA .....	6
7. DESARROLLO PLATAFORMA SCADA WEB .....	1
7.1 METODOLOGÍA DE DESARROLLO .....	1
7.2 ARQUITECTURA DEL SISTEMA .....	2

7.3 COMPONENTES DEL SISTEMA .....	3
7.4 BASE DE DATOS.....	4
7.5 CONEXIÓN CON EL CONTROLADOR LÓGICO PROGRAMABLE .....	5
7.6 ESTRUCTURA DE ARCHIVOS.....	7
7.7 FLUJO GENRAL DE DATOS DEL SISTEMA .....	7
7.8 FUNCIONALIDADES IMPLEMENTADAS .....	8
8. VALIDACIÓN Y RESULTADOS.....	1
8.1 VALIDACIÓN DE COMPONENTES FISCHERTECHNIK .....	1
8.2 VALIDACIÓN DEL CONTROLADOR LÓGICO PROGRAMABLE .....	2
8.3 VALIDACIÓN DEL SISTEMA INTEGRADO .....	3
8.4 VALIDACIÓN DE GUÍAS DE LABORATORIO .....	7
8.5 CONCLUSIONES DE LA VALIDACIÓN .....	8
CONCLUSIONES .....	1
RECOMENDACIONES.....	1
BIBLIOGRAFIA.....	1
ANEXOS .....	1
ANEXO A: RECOPILACIÓN INFORMACIÓN CONTROLADORES COMERCIALES .....	1
ANEXO B: RECOPILACIÓN INFORMACIÓN PLATAFORMAS IOT .....	1
ANEXO C: ESPECIFICACIONES TECNICAS CELDAS FISCHERTECHNIK.....	1
ANEXO D: SELECCIÓN DE COMPONENTES ELECTRONICOS .....	1
ANEXO E: REGLAS DE DISEÑO APLICADAS .....	1
ANEXO F: FABRICACIÓN PCB SHIELD ADAPTADOR .....	1
ANEXO G: DISEÑO Y FABRICACIÓN DE LA CARCASA.....	1

ANEXO H: RECONSTRUCCIÓN PIEZAS FALTANTES .....	1
ANEXO I: CONEXIONES DEL SISTEMA.....	1
ANEXO J: TECNOLOGÍAS Y CONSIDERACIONES DE DISEÑO .....	1
ANEXO K: ARQUITECTURA DETALLADA DE LA PLATAFORMA.....	1
ANEXO L: MODELOS DE BASE DE DATOS.....	1
ANEXO M: IMPLEMENTACIÓN DE PROTOCOLOS DE COMUNICACIÓN .....	1
ANEXO N: ESTRUCTURA DE LA RAÍZ DE ARCHIVOS .....	1
ANEXO O: VALIDACIÓN DE PIEZAS FISCHERTECHNIK .....	1
ANEXO P: VALIDACIÓN DEL CONTROLADOR LÓGICO PROGRAMABLE.....	1
ANEXO Q: VALIDACIÓN DEL SISTEMA INTEGRADO .....	1
ANEXO R: VALIDACIÓN DE GUÍAS DE LABORATORIO.....	1

## LISTA DE TABLAS

Tabla 1 Componentes electrónicos de la planta FISCHERTECHNIK. ....	4
Tabla 2 Entradas/Salidas FISCHERTECHNIK vs Proyecto. ....	5
Tabla 3 Dimensiones finales carcasa controlador. ....	2
Tabla 4 Matriz pugh selección de fabricación/material. ....	3
Tabla 5 Resumen componentes principales del sistema. ....	4
Tabla 6 Resultados comparaciones de errores. ....	2
Tabla 7 Resultados pruebas de validación. ....	5
Tabla 8 Comparación de especificaciones técnicas. ....	1
Tabla 9 Capacidad de procesamiento. ....	1
Tabla 10 Capacidad de entradas/salidas. ....	2
Tabla 11 Capacidad de conectividad y comunicaciones. ....	2
Tabla 12 Comparación de especificaciones técnicas. ....	1
Tabla 13 I/O Celda Brazo neumático FISCHERTECHNIK. ....	1
Tabla 14 Especificaciones I/O Celda Brazo neumático FISCHERTECHNIK. ....	2
Tabla 15 I/O Celda Banda Transportadora FISCHERTECHNIK. ....	2
Tabla 16 Especificaciones I/O Celda Banda Transportadora FISCHERTECHNIK. ....	3
Tabla 17 Selección de microcontrolador [43] ....	1
Tabla 18 Selección módulo de comunicación. ....	3
Tabla 19 Selección entradas digitales – Optoacoplador. ....	5
Tabla 20 Selección entradas digitales – Inversor. ....	7
Tabla 21 Selección entradas analógicas. ....	8
Tabla 22 Selección salidas motor PWM. ....	10
Tabla 23 Selección salidas analógicas. ....	12
Tabla 24 Selección salidas relé. ....	13
Tabla 25 Selección fuente de voltaje. ....	15
Tabla 26 Elaboración de piezas. ....	2
Tabla 27 Piezas CAD faltantes y su función. ....	2
Tabla 28 Información de conexiones de banda transportadora. ....	2

Tabla 29 Información de conexiones de brazo neumático. ....	5
Tabla 30 Información conexiones de neumática del Brazo neumático. ....	7
Tabla 31 Comparación tecnologías web. ....	2
Tabla 32 Librerías usadas en cada capa. ....	3
Tabla 33 Validación de piezas Fischertechnik. ....	2
Tabla 34 Mediciones experimentales entradas digitales. ....	3
Tabla 35 Mediciones experimentales entradas analógicas. ....	4
Tabla 36 Mediciones experimentales salidas analógicas. ....	7
Tabla 37 Validación fototransistor. ....	3
Tabla 38 Validación valores sensor de color. ....	4

## LISTA DE FIGURAS

Figura 1 Procesos de la celda mecatrónica marca Fischertechnik.....	2
Figura 2 Requisitos ROBO Pro [1].....	3
Figura 3 Diagrama de bloques del sistema.....	5
Figura 4 Metodología V.....	1
Figura 5 Composición modular PLC proyecto de grado.....	3
Figura 6 Circuito entradas digitales.....	4
Figura 7 Simulación entradas digitales .....	5
Figura 8 Circuito entradas analógicas.....	6
Figura 9 Simulación entradas analógicas.....	7
Figura 10 Circuito salidas analógicas.....	8
Figura 11 Simulación salidas analógicas .....	10
Figura 12 Circuito salidas motores PWM.....	11
Figura 13 Simulación salidas motores PWM.....	11
Figura 14 Circuito salidas digitales relé.....	12
Figura 15 Simulación salidas digitales relé .....	13
Figura 16 Modelo 3D capa superior módulo I/O.....	14
Figura 17 Ruteado capa superior módulo I/O .....	15
Figura 18 Modelo 3D capa inferior módulo I/O.....	15
Figura 19 Ruteado capa inferior módulo I/O. ....	16
Figura 20 Módulo I/O y módulo shield adaptador fabricados. ....	16
Figura 21 Vista isométrica CAD carcasa controlador.....	1
Figura 22 Vista isométrica carcasa fabricada.....	4
Figura 23 Modelo banda transportadora en ft Desing. ....	5
Figura 24 Modelo brazo neumático en ft Desing.....	5
Figura 25 Boceto ubicación del sistema.....	6
Figura 26 Patrón MVC.....	2
Figura 27 Arquitectura del sistema general, incluido controlador lógico programable proyecto.....	3

Figura 28 Flujo manipulación actuadores.	5
Figura 29 Flujo de lectura de sensores.	6
Figura 30 Flujo general de datos del sistema.	7
Figura 31 Montaje del sistema completo.	3
Figura 32 Diagrama de flujo del proceso del sistema.	4
Figura 33 Pruebas del funcionamiento SCADA y celdas.	6
Figura 34 Modelo 3D capa superior módulo shield adaptador.	1
Figura 35 Componentes módulo shield adaptador.	2
Figura 36 Plano general.	3
Figura 37 Plano bisagra.	4
Figura 38 Plano tapa inferior.	4
Figura 39 Plano tapa superior.	5
Figura 40 Diagrama de conexiones del controlador y la banda transportadora.	1
Figura 41 Diagrama de conexiones del controlador y el brazo neumático.	4
Figura 42 Diagrama de conexiones neumáticas del brazo.	6
Figura 43 Octoverse 2023 Github [46].	2
Figura 44 Modelo board.	1
Figura 45 Modelo datavar.	2
Figura 46 Modelo project.	3
Figura 47 Interconexión componentes del sistema y controlador lógico programable.	1
	1
Figura 48 Estructura trama protocolo serial.	2
Figura 49 Estructura trama protocolo MIDI.	3
Figura 50 Entrada analógica/controlador sin compensador [V] vs Entrada analógica/fuente [V].	4
Figura 51 Entrada analógica/controlador con compensador [V] vs Entrada analógica/fuente [V].	5
Figura 52 Diagrama de flujo del proceso del brazo neumático.	1
Figura 53 Diagrama de flujo del proceso de la banda transportadora.	2
Figura 54 Posición correcta lectura de color pieza.	5

Figura 55 Resultados conexión Gateway-Controlador .....	1
Figura 56 Resultados Comparación con otras plataformas SCADA.....	2
Figura 57 Resultados de la interfaz .....	2
Figura 58 Resultados aspectos a mejorar.....	2
Figura 59 Resultados conocimiento de JavaScript. ....	3
Figura 60 Resultados satisfacción con el editor de código.....	3
Figura 61 Resultados eficiencia de lectura/escritura de GPIO. ....	3
Figura 62 Resultados diferencia frente a C/C++ de Arduino. ....	4
Figura 63 Resultados satisfacción con contador de piezas y banderas. ....	4
Figura 64 Resultados representatividad de procesos industriales.....	4
Figura 65 Resultados interrupciones o errores durante ejecución.....	5
Figura 66 Resultados fluidez entre SCADA y celda física. ....	5
Figura 67 Resultados frecuencia de consulta al tutor.....	5
Figura 68 Resultados relevancia para redes y automatización. ....	6
Figura 69 Resultados relevancia claridad de las guías. ....	6
Figura 70 Resultados potencial de uso futuro. ....	6

## 1 INTRODUCCIÓN

### 1.2 RESUMEN

Este trabajo de grado explora la viabilidad de diseñar un controlador lógico programable (PLC) de hardware libre para la celda mecatrónica Fischertechnik del laboratorio de automatización de la UNAB, complementado con una plataforma SCADA web para su monitoreo y control. Partiendo de la obsolescencia del software propietario original y los estándares de la Industria 4.0, se propone un PLC basado en Arduino Mega (ATmega2560) que iguala o supera las capacidades de entrada/salida del controlador comercial, y una plataforma web implementada desde cero con tecnologías open-source.

Los objetivos incluyen: (1) diseñar el esquema electrónico del PLC, (2) desarrollar la plataforma SCADA web, (3) fabricar la PCB y su carcasa, y (4) validar el sistema mediante guías de laboratorio.

La metodología siguió el modelo en V, abarcando especificación de funciones, definición de subsistemas y componentes, prototipado, pruebas unitarias e integración, hasta la validación completa del sistema.

Como resultados, se obtuvo un esquema electrónico reproducible, dos controladores autónomos de hardware libre, una plataforma SCADA funcional y tres guías de laboratorio que demuestran la usabilidad y pertinencia académica del proyecto. La validación práctica con la banda transportadora y el brazo neumático confirmó el correcto funcionamiento de entradas digitales, analógicas, salidas PWM y relés, así como la interacción en tiempo real con la interfaz web.

Palabras clave: controlador lógico programable, hardware libre, scada, celda mecatrónica, industria 4.0.

### 1.3 PLANTEAMIENTO DEL PROBLEMA

En el laboratorio de automatización de la UNAB, actualmente se encuentra una celda mecatrónica de la marca Fischertechnik que no está siendo utilizada. Esta celda está diseñada para ejecutar tres procesos distintos a escala reducida: el transporte de piezas, la manipulación mediante un brazo robótico y la clasificación. Su potencial para experimentar con estos procesos la convierte en una herramienta valiosa para el aprendizaje y la investigación de los estudiantes.

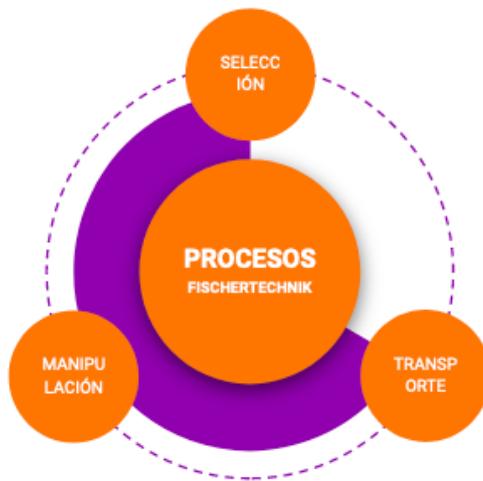


Figura 1 Procesos de la celda mecatrónica marca Fischertechnik.

Aunque la celda cuenta con un controlador propio de la marca Fischertechnik, el software necesario para programarla es propietario, lo que limita su flexibilidad. Este software, que requiere Windows Vista como sistema operativo, se ha vuelto obsoleto debido a su antigüedad de más de 10 años. Esta situación dificulta su integración con tecnologías modernas lo cual resulta poco atractiva en el ámbito educativo, especialmente en el contexto actual de la Industria 4.0.

### **1.1 Instalación de ROBO Pro**

Los requisitos mínimos para la instalación de ROBO Pro son los siguientes:

- Un PC compatible IBM con procesador Pentium II a 500 MHz como mínimo, 64 MB RAM y unos 40 MB de memoria libre en el disco duro



- Un monitor y una tarjeta gráfica con resolución mínima de 1024x768 píxeles. En monitores con tubo de imagen la frecuencia de refresco de imagen debería ser como mínimo de 85 Hz para obtener una imagen sin parpadeo. Las pantallas planas TFT ofrecen siempre una imagen sin parpadeo, de modo que la frecuencia de refresco no tiene importancia en este tipo de pantallas.
- Microsoft Windows XP o Vista
- Un puerto USB libre para la conexión de ROBO TX Controller. Para el ROBO Interface necesitas un puerto USB libre o un puerto RS232 libre de COM1 a COM4.

**Figura 2 Requisitos ROBO Pro [1]**

Por este motivo se plantea en el proyecto la siguiente pregunta : ¿Es viable diseñar un controlador lógico programable basado en hardware libre provisto de funcionalidades similares a las de uno comercial conectado a una plataforma desarrollada en tecnologías web para el monitoreo y control de la celda mecatrónica Fischertechnik de la UNAB?

## **1.4 CONTEXTO Y JUSTIFICACIÓN**

La celda mecatrónica Fischertechnik, con sus secuencias operativas a escala, proporciona una plataforma ideal para simular procesos industriales en entornos educativos [1]. Sin embargo, la obsolescencia de su controlador propietario representa una valiosa oportunidad para desarrollar un nuevo controlador basado en hardware libre, que ofrezca funcionalidades similares a las de un PLC comercial y esté alineado con las tendencias de la industria.

En este contexto, la relevancia de un programa orientado a la ingeniería radica en su capacidad para adaptarse a las tendencias emergentes y las demandas del mercado laboral. Esto implica la incorporación de tecnologías innovadoras,

metodologías ágiles y prácticas sostenibles. Por ejemplo, las tendencias en controladores, junto con los principios de la Industria 4.0, pueden preparar a los estudiantes para enfrentar los desafíos de la industria actual y futura [2].

Se espera que los PLC evolucionen en respuesta a los avances tecnológicos en comunicaciones, hardware y software, especialmente en el contexto del crecimiento de la Industria. Este fenómeno se detalla en el análisis de tendencias y pronósticos del mercado de PLC para 2024-2029 realizado por Mordor Intelligence, que resulta particularmente revelador. Según el informe, "...la creciente necesidad de personalización por parte de los usuarios finales ha llevado a procesos de fabricación más refinados y complejos, con requerimientos de ajustes frecuentes. Esto ha impulsado a los usuarios a adoptar e invertir en sistemas más flexibles, como los controladores basados en PC, en lugar de los PLC tradicionales..." [3].

El informe también destaca "el desarrollo de controladores de automatización programables (PAC), que integran los PLC con las PC" como un factor clave en esta evolución. Sin embargo, aunque muchos proveedores de controladores industriales enfatizan las diferencias entre PLC y PAC, los futuros ingenieros de automatización podrían no centrarse tanto en la nomenclatura, sino en el rendimiento y las características disponibles al seleccionar sus sistemas [3].

Esta transformación de los PLC refleja una creciente demanda por sistemas de automatización más adaptables, que no solo reduzcan costos, sino que también ofrezcan mayor flexibilidad. Los controladores basados en hardware libre, al facilitar una integración más sencilla con otras tecnologías modernas mediante software libre, emergen como alternativas viables a los PLC tradicionales. Además, estos dispositivos pueden ser actualizados y mantenidos con mayor facilidad, lo que contribuye a una reducción en los costos operativos a largo plazo [4].

El desarrollo de controladores basados en hardware abierto ha sido objeto de estudio en múltiples universidades a nivel global, incluyendo instituciones en Perú [5], Noruega [6] y Malasia [7]. Esto resalta la creciente importancia de esta área, que promueve conceptos innovadores para los estudiantes orientados hacia la industria inteligente. La implementación de laboratorios para aplicar estas nuevas tecnologías, apoyados por un módulo controlador desarrollado en la UNAB, permitiría a la universidad posicionarse como una de las instituciones interesadas en este campo emergente.

Por esta razón, este proyecto de grado pretende desarrollar un controlador lógico programable basado en hardware libre conectado a una plataforma para control de supervisión y adquisición de datos basada en tecnologías web, permitiendo que los procesos industriales a escala de la celda mecatrónica Fischertechnik de la UNAB cumplan con los estándares de la Industria 4.0.

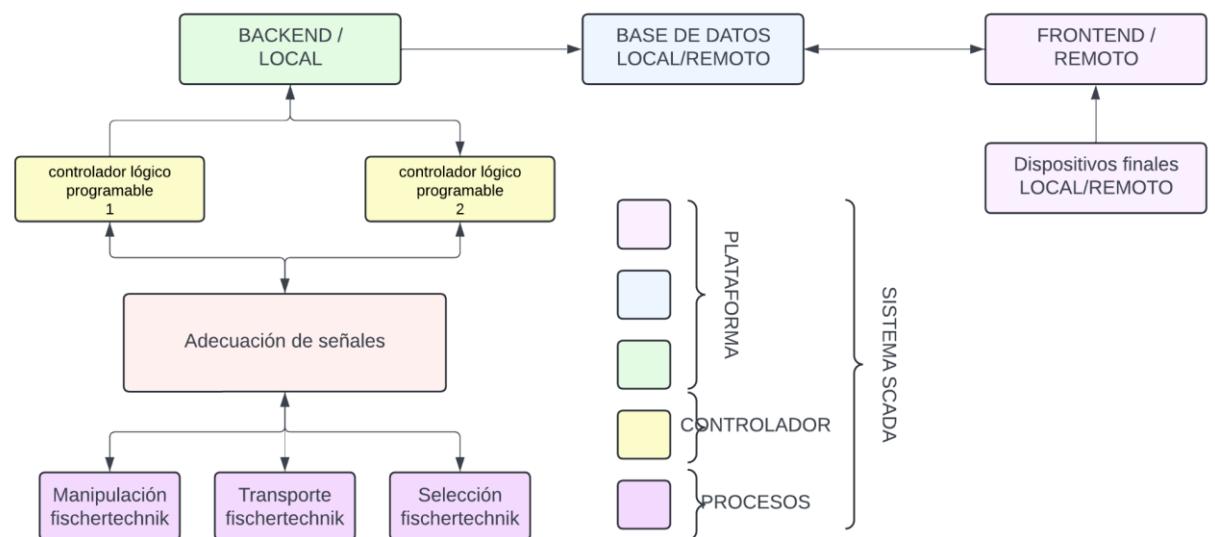


Figura 3 Diagrama de bloques del sistema.

## 1.5 ANTECEDENTES

En el ámbito de la automatización industrial, ha crecido el interés por alternativas más accesibles y flexibles a los Controladores Lógicos Programables (PLC) tradicionales. Esta tendencia responde a la demanda de soluciones económicas para aplicaciones educativas y de pequeña escala industrial, así como a la integración de tecnologías emergentes como el Internet de las Cosas (IoT) y la realidad aumentada. En este contexto, plataformas de hardware libre como Arduino y Raspberry Pi han surgido como potenciales sustitutos de los PLC en ciertas aplicaciones, desafiando el paradigma tradicional.

La literatura reciente presenta un panorama variado de enfoques y soluciones que abordan esta necesidad. Investigadores han explorado el desarrollo de alternativas de bajo costo basadas en hardware libre, con resultados prometedores pero también con desafíos significativos. Proyectos como "PLCduino", basado en Arduino Uno [7], y un controlador industrial utilizando Raspberry Pi [8], han demostrado la viabilidad de realizar muchas funciones de un PLC convencional a un costo considerablemente menor. Estos desarrollos han sido especialmente exitosos en entornos educativos y en pequeñas aplicaciones industriales, aunque enfrentan limitaciones en capacidad de procesamiento y memoria.

Asimismo, se han realizado esfuerzos para crear soluciones de automatización industrial basadas en hardware y software de código abierto. Por ejemplo, se ha utilizado Raspberry Pi como servidor OPC-UA y Arduino para módulos de entrada/salida [6]. Estas iniciativas han mostrado resultados alentadores en términos de funcionalidad y respuesta rápida, aunque su implementación efectiva en entornos industriales ha requerido ajustes adicionales.

La integración de tecnologías web en sistemas SCADA ha sido otra área de investigación destacada. Estudios como el desarrollo de un sistema SCADA basado

en la web para el control de procesos químicos [9] y la implementación de soluciones de monitoreo y control basadas en IoT [10] han evidenciado las ventajas de estas tecnologías en términos de flexibilidad y accesibilidad. Sin embargo, estos proyectos también han enfrentado desafíos relacionados con la integración de componentes tecnológicos diversos y limitaciones en la capacidad de los servidores web.

La incorporación de tecnologías emergentes, como la realidad aumentada mediante PLC comerciales, ha sido explorada por investigadores como Rueda y Juan [11], quienes desarrollaron una aplicación para mejorar el soporte técnico y el control de sistemas industriales. A pesar de su eficacia y soporte, estas soluciones también enfrentan limitaciones técnicas y desafíos en la calidad de los modelos 3D de realidad aumentada.

Los estudios comparativos entre soluciones de bajo costo y PLC comerciales han arrojado luz sobre las fortalezas y debilidades de cada enfoque. Investigaciones de Zaragoza y Kim [12] destacan que, aunque plataformas como Arduino son más accesibles, los PLC tradicionales siguen siendo superiores en robustez para aplicaciones industriales a gran escala. Sin embargo, a menudo estos estudios carecen de datos cuantitativos sobre el rendimiento a largo plazo en condiciones reales.

El desarrollo de controladores industriales modulares basados en hardware abierto, como el propuesto por Kairuz Cabrera [13], busca cerrar la brecha entre las soluciones de bajo costo y los PLC comerciales. Estos enfoques han cumplido requisitos técnicos comparables a los controladores comerciales a un costo reducido, aunque aún enfrentan desafíos en la robustez del hardware en entornos industriales exigentes.

En conjunto, estos estudios demuestran que las alternativas de bajo costo a los PLC tradicionales son viables y económicamente ventajosas, especialmente en aplicaciones educativas y de pequeña escala industrial. La integración de tecnologías web, IoT y realidad aumentada ha ampliado las capacidades de estos sistemas, ofreciendo mayor flexibilidad y accesibilidad. No obstante, persisten retos en términos de robustez, capacidad de procesamiento y compatibilidad con estándares industriales [14].

De cara al futuro, existen áreas prometedoras para la investigación y el desarrollo. Es esencial explorar métodos que incrementen la robustez y fiabilidad de las soluciones basadas en hardware libre en entornos industriales rigurosos, así como técnicas para mejorar su rendimiento. También es urgente el desarrollo de estándares que faciliten la integración y compatibilidad de diferentes módulos en sistemas de automatización de bajo costo [15].

Adicionalmente, la optimización de protocolos de comunicación entre PLC y sistemas SCADA, la mejora de la seguridad en sistemas SCADA basados en web, y la realización de estudios comparativos a largo plazo entre soluciones de bajo costo y PLC comerciales en entornos industriales son áreas que requieren atención continua. Estos esfuerzos podrían contribuir significativamente a la evolución de las soluciones de automatización industrial de bajo costo, mejorando su viabilidad y adopción en una gama más amplia de aplicaciones.

En conclusión, el campo de la automatización industrial está experimentando una transformación impulsada por la búsqueda de alternativas más accesibles y flexibles a los PLC tradicionales. Aunque las soluciones basadas en hardware libre y tecnologías emergentes han demostrado un potencial significativo, aún queda trabajo por hacer para abordar los desafíos de robustez, rendimiento y compatibilidad. El futuro de la automatización industrial probablemente verá una convergencia continua entre soluciones de bajo costo y tecnologías industriales.

establecidas, resultando en sistemas más accesibles y potentes para diversas aplicaciones.

## 1.6 OBJETIVOS

### 1.6.1 GENERAL

Implementar un sistema para control de supervisión y adquisición de datos de la celda mecatrónica Fischertechnik del laboratorio de automatización utilizando hardware y software libre.

### 1.6.2 ESPECIFICOS

- Diseñar el esquema electrónico de un PLC (controlador lógico programable) utilizando hardware libre, que ofrezca funcionalidades similares a las de un dispositivo comercial.
- Desarrollar la plataforma SCADA (control de supervisión y adquisición de datos) basada en tecnologías web, destinada a gestionar los procesos de la celda mecatrónica Fischertechnik.
- Desarrollar la placa de circuito impreso y su carcasa protectora, basándose en el esquema electrónico del controlador lógico programable.
- Validar el funcionamiento de los procesos de la celda mecatrónica Fischertechnik mediante el desarrollo y aplicación de guías de laboratorio basadas en el sistema SCADA (control de supervisión y adquisición de datos).

## 2 MARCO TEÓRICO

Este marco teórico examina aspectos fundamentales como: los controladores lógicos programables (PLCs) de hardware libre, los protocolos de comunicación, los sistemas SCADA basados en tecnologías open-source, las tecnologías web para el desarrollo de interfaces de usuario y la integración de celdas Fischertechnik en entornos educativos. El propósito es proporcionar una visión integral de las alternativas y herramientas disponibles para la automatización moderna.

### 2.1 CONTROLADOR LÓGICO PROGRAMABLES

Un Controlador Lógico Programable (PLC) es un dispositivo electrónico diseñado para automatizar procesos industriales. Funciona como un pequeño computador robusto capaz de ejecutar programas en tiempo real. Los PLC procesan señales de entrada, toman decisiones basadas en su programación y controlan salidas para operar máquinas o procesos. Su versatilidad, confiabilidad y capacidad de soportar entornos industriales hostiles los han convertido en componentes esenciales de la automatización moderna [16].

El PLC clásico es la versión tradicional y ampliamente utilizada en la industria. Se caracteriza por su robustez, diseño modular y capacidad de expansión. Estos dispositivos ofrecen una amplia gama de entradas y salidas, tanto digitales como analógicas, y pueden comunicarse con otros equipos mediante diversos protocolos industriales [17].

La adecuación de señales en PLCs involucra el acondicionamiento de entradas y salidas para garantizar su compatibilidad y seguridad. Para señales digitales, se

utilizan optoacopladores que aíslan eléctricamente el circuito de entrada del PLC. En señales analógicas, se emplean amplificadores operacionales para ajustar niveles de voltaje o corriente [18].

Los PLC clásicos se programan utilizando lenguajes estandarizados por la norma IEC 61131-3. Estos incluyen el Diagrama de Escalera (LD), ampliamente usado por su similitud con los esquemas eléctricos; el Diagrama de Bloques de Función (FBD), que permite una programación gráfica basada en bloques; el Lista de Instrucciones (IL), un lenguaje de bajo nivel similar al ensamblador; y el Texto Estructurado (ST), que se asemeja a lenguajes de alto nivel como Pascal [19].

## 2.2 HARDWARE LIBRE Y MICROCONTROLADORES

El hardware libre en el contexto de la automatización industrial se refiere a plataformas de desarrollo de código abierto que permiten la creación de sistemas de control personalizados. Estas plataformas ofrecen una alternativa flexible y económica a los PLC comerciales, especialmente para aplicaciones de pequeña y mediana escala [20].

Los microcontroladores son componentes electrónicos programables que integran un procesador, memoria y periféricos en un solo chip. En el contexto de la automatización, se utilizan como alternativa de bajo costo y alta flexibilidad a los PLC tradicionales. Su programación generalmente se realiza en lenguajes de alto nivel como C o C++, lo que facilita el desarrollo rápido de aplicaciones de control [21].

Arduino es una plataforma de hardware y software libre ampliamente utilizada en proyectos de automatización a pequeña escala. Se basa en microcontroladores Atmel AVR y ofrece un entorno de desarrollo integrado (IDE) fácil de usar. Arduino

se destaca por su simplicidad, versatilidad y una extensa comunidad de usuarios [22].

El ESP32 es un microcontrolador de bajo costo y alto rendimiento desarrollado por Espressif Systems. Se destaca por su conectividad Wi-Fi y Bluetooth integrada, haciéndolo ideal para aplicaciones de IoT y automatización inalámbrica. Su capacidad para ejecutar sistemas operativos en tiempo real (RTOS) lo hace adecuado para tareas de cómputo más complejas [23].

### 2.3 INDUSTRIA 4.0 Y TENDENCIAS ACTUALES

Las nuevas tendencias en PLCs se enfocan en la integración de tecnologías avanzadas y la adaptación a la Industria 4.0. Se observa una evolución hacia dispositivos más compactos, con mayor capacidad de procesamiento y memoria. La conectividad mejorada, incluyendo Ethernet Gigabit y capacidades inalámbricas, facilita la integración con sistemas IoT industriales [3].

La Industria 4.0 representa la cuarta revolución industrial, caracterizada por la integración de tecnologías digitales en los procesos de fabricación. En este contexto, los PLC juegan un papel crucial como nodos de conectividad y control en sistemas. Se integran con tecnologías como IoT, big data y computación en la nube para facilitar la toma de decisiones en tiempo real [3].

Los Controladores de Automatización Programables (PAC) representan una evolución de los PLC tradicionales, combinando la robustez y confiabilidad de estos con las capacidades de procesamiento y flexibilidad de las PC industriales. Los PAC ofrecen un rendimiento superior en aplicaciones que requieren control avanzado, procesamiento de datos complejo e integración con sistemas de nivel superior [3].

## 2.4 PROTOCOLOS DE COMUNICACIÓN

TCP/IP (Transmission Control Protocol/Internet Protocol) es el conjunto de protocolos estándar para comunicaciones en redes, incluyendo Internet y muchos sistemas SCADA modernos. TCP proporciona una comunicación orientada a conexión, confiable y ordenada, ideal para transmisiones críticas de datos. La adopción de TCP/IP en SCADA facilita la integración con sistemas IT existentes [24].

El Protocolo de Datagramas de Usuario (UDP) es un protocolo de transporte sin conexión utilizado en sistemas SCADA para comunicaciones que priorizan la velocidad sobre la fiabilidad. UDP es ideal para transmisiones en tiempo real donde la pérdida ocasional de paquetes es preferible a la latencia introducida por la retransmisión [25].

Modbus es un protocolo de comunicación ampliamente utilizado en la industria de la automatización y sistemas SCADA. Desarrollado originalmente por Modicon, ahora es un estándar abierto. Modbus permite la comunicación cliente-servidor entre dispositivos conectados en diferentes tipos de buses o redes. Se caracteriza por su simplicidad, robustez y facilidad de implementación [26].

El Modelo OSI (Open Systems Interconnection) es un marco conceptual desarrollado por la ISO que describe las funciones de un sistema de comunicación en siete capas: física, enlace de datos, red, transporte, sesión, presentación y aplicación. Este modelo facilita la interoperabilidad y la resolución de problemas en sistemas complejos de comunicación [27].

## 2.5 SISTEMAS SCADA

SCADA (Supervisory Control and Data Acquisition) es un sistema de software utilizado en entornos industriales para monitorear y controlar procesos a gran escala. Permite la recolección de datos en tiempo real desde diversos dispositivos de campo, como sensores y PLCs, y proporciona una interfaz centralizada para la supervisión y control de operaciones [28].

El SCADA monolítico representa la primera generación de sistemas SCADA. Se caracteriza por una arquitectura centralizada donde todos los componentes del sistema se integran en un solo sistema computacional. Aunque robusto y confiable para aplicaciones de escala moderada, tiene limitaciones en términos de escalabilidad y flexibilidad [29].

El SCADA distribuido representa la segunda generación, diseñada para superar las limitaciones del enfoque monolítico. Las funciones del sistema se distribuyen entre múltiples estaciones conectadas a través de una red local (LAN). Esta distribución mejora la escalabilidad y redundancia del sistema [26].

## 2.6 TECNOLOGÍAS WEB

### 2.6.1 FRONTEND

HTML, CSS y JavaScript constituyen el núcleo del desarrollo frontend en sistemas SCADA modernos. HTML estructura el contenido, CSS maneja el diseño y la presentación visual, mientras que JavaScript aporta interactividad y dinamismo. Esta combinación permite crear interfaces de usuario sofisticadas y responsivas,

capaces de mostrar datos en tiempo real, gráficos interactivos y paneles de control personalizables [30].

React.js es una biblioteca de JavaScript para construir interfaces de usuario interactivas y eficientes en sistemas SCADA. Desarrollada por Facebook, React utiliza un enfoque basado en componentes que facilita la creación de aplicaciones web de una sola página (SPA). Su sistema de renderizado virtual optimiza el rendimiento, permitiendo actualizaciones rápidas de la interfaz de usuario [31].

AngularJS es un framework de JavaScript mantenido por Google, diseñado para crear aplicaciones web dinámicas de una sola página. Su arquitectura MVC (Modelo-Vista-Controlador) promueve una organización clara del código, mientras que su sistema de enlace de datos bidireccional simplifica la sincronización entre el modelo y la vista [32].

## 2.6.2 BACKEND Y BASES DE DATOS

El backend maneja la lógica del servidor, procesamiento de datos y comunicación con dispositivos de campo. Implementa protocolos industriales, gestiona bases de datos, procesa alarmas y ejecuta lógica de control. Tecnologías como Node.js, Python y Go son populares por su eficiencia y capacidad de manejar operaciones en tiempo real [33].

Node.js es un entorno de ejecución de JavaScript del lado del servidor, ideal para sistemas SCADA debido a su modelo de E/O no bloqueante y orientado a eventos. Permite manejar múltiples conexiones concurrentes eficientemente, crucial para la adquisición de datos en tiempo real [34].

Python es un lenguaje de programación versátil ampliamente utilizado en el backend de sistemas SCADA. Su sintaxis clara y su rica biblioteca de módulos lo hacen ideal para el procesamiento de datos, análisis estadístico y machine learning en entornos industriales [35].

Las bases de datos en sistemas SCADA son fundamentales para almacenar y gestionar grandes volúmenes de datos históricos y en tiempo real. MySQL es un sistema de gestión de bases de datos relacionales que ofrece un buen rendimiento para operaciones de lectura y escritura, esencial en la captura y recuperación de datos históricos [36]. MongoDB es una base de datos NoSQL orientada a documentos, cuyo modelo de datos flexible permite almacenar estructuras complejas y variables, ideal para manejar diversos tipos de datos industriales [37].

## 2.7 CELDAS FISCHERTECHNIK

Fischertechnik es un sistema de construcción y programación educativo utilizado en la enseñanza de robótica, mecánica y automatización. Desarrollado en Alemania, ofrece kits modulares que permiten a estudiantes y entusiastas crear modelos funcionales de máquinas y sistemas automatizados. Fischertechnik se destaca por su enfoque práctico en la enseñanza de conceptos STEM, combinando elementos mecánicos con componentes electrónicos programables [1].

El kit Fischertechnik STEAM Mechatronics es una solución educativa integral diseñada para la enseñanza de Ciencia, Tecnología, Ingeniería y Matemáticas. Este kit permite a los estudiantes construir y programar modelos mecatrónicos complejos, integrando conceptos de mecánica, electrónica y programación. Incluye componentes como motores, sensores, y controladores programables [1].

La cinta transportadora Fischertechnik es un módulo didáctico que simula el transporte automatizado de piezas mediante una banda impulsada por un motor de 9V. Integra sensores y actuadores, y permite el control mediante PLC o microcontroladores, facilitando la enseñanza de procesos industriales automatizados [1].

Fischertechnik permite y fomenta el diseño libre y personalizado dentro de sus sistemas educativos, al ofrecer kits modulares sin una configuración fija, lo cual impulsa la creatividad técnica y la solución de problemas en entornos de formación. A través de su plataforma digital "Fischertechnik Cloud", los usuarios pueden compartir, modificar y reutilizar modelos creados por otros, promoviendo un enfoque abierto al desarrollo de soluciones propias [1].

## 2.8 NORMATIVIDAD

### 2.8.1 DISEÑO DE PCB

La norma IPC-2221 establece los requisitos generales de diseño para placas de circuito impreso, incluyendo geometría de pistas, separación mínima entre conductores y consideraciones de apantallamiento para garantizar integridad de señal y manufacturabilidad. Su aplicación resulta esencial para proyectos de PCB de cualquier complejidad [38].

La norma IPC-610 especifica los requisitos de aceptación para montaje de placas de circuito impreso, englobando criterios de calidad para soldaduras, componentes y acabado superficial. Su cumplimiento asegura fiabilidad a largo plazo y reduce fallas en campo [39].

## 2.8.2 LICENCIAS DE SOFTWARE LIBRE

Las librerías de código abierto son colecciones de funciones y módulos distribuidos con licencias permisivas, que permiten su uso, modificación y redistribución sin coste. Facilitan el desarrollo eficiente al reutilizar componentes probados, fomentando la interoperabilidad y la innovación comunitaria [40].

La Licencia Pública General de GNU (GPL) garantiza que todo software derivado se mantenga libre, obligando a distribuir el código fuente bajo los mismos términos. Impulsa la colaboración comunitaria y evita que el software sea incorporado en productos propietarios sin otorgar los mismos derechos a los usuarios [41].

La Licencia MIT es una de las más sencillas y permisivas, permitiendo usar, copiar, modificar y distribuir el software con mínima restricción, únicamente exigiendo conservar el aviso de derechos de autor. Es ampliamente adoptada por su claridad y compatibilidad con la mayoría de proyectos [42].

### 3. METODOLOGÍA

En el desarrollo de este proyecto, se adoptó la metodología Devdas Shetty, la cual se caracteriza por su enfoque en la revisión y verificación continua en cada una de las etapas del proceso. Esta metodología permite identificar posibles desviaciones o errores en tiempo real, lo que facilita la implementación de correcciones y ajustes necesarios de manera oportuna. Su aplicación garantiza una mayor precisión y calidad en el desarrollo, asegurando que cada fase del proyecto se ejecute de acuerdo con los objetivos establecidos y que los posibles problemas sean solucionados antes de avanzar a etapas posteriores.

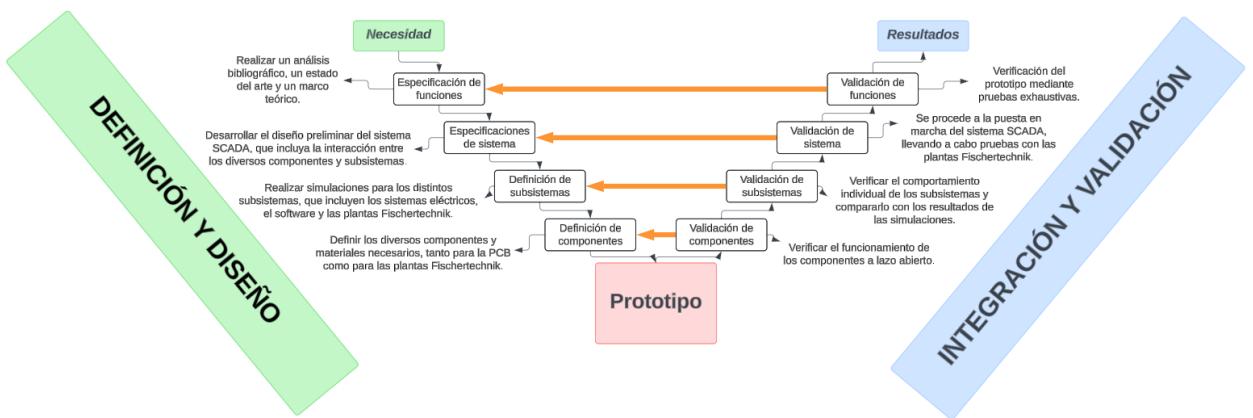


Figura 4 Metodología V.

1. **Especificación de funciones:** En esta primera etapa, se obtienen las especificaciones generales del proyecto. Se investiga sobre proyectos similares, analizando prototipos, componentes y software utilizados. Además, se realiza una lluvia de ideas para proponer y evaluar alternativas, eligiendo las soluciones más viables para el desarrollo del proyecto.

2. Especificaciones del sistema: En esta fase, se desarrolla el diseño preliminar del sistema SCADA, incluyendo la integración del hardware y software con sus principales características. Se detalla la interacción entre los distintos componentes y subsistemas. Para visualizar el funcionamiento global del proyecto, se elabora un bosquejo que ilustra la estructura del sistema, mostrando cómo se comunicarán los elementos y cómo se gestionan las funciones clave de monitoreo y control.
3. Definición de subsistemas: En este paso, el diseño general del sistema SCADA se descompone en subsistemas específicos, cada uno encargado de realizar tareas particulares o conjuntos de tareas. Se abordan tanto los sistemas eléctricos (incluyendo esquemas eléctricos, etapas de potencia y el controlador lógico programable), como los sistemas de software (protocolos de comunicación y el sistema SCADA web). Además, se considera el funcionamiento de las plantas Fischertechnik, para las cuales se elabora un bosquejo general que detalla el funcionamiento de cada planta y su integración dentro del sistema.
4. Definición de componentes: En esta etapa se realiza el diseño detallado de cada subsistema, definiendo los componentes necesarios para el sistema electrónico, los softwares requeridos para el sistema SCADA web, y los sensores y actuadores necesarios para el correcto funcionamiento de la planta Fischertechnik. Además, se identifican los elementos clave para asegurar la integración y comunicación eficiente entre todos los subsistemas.
5. Prototipo: Una vez completada la fase de diseño y definida la lista de componentes, se investiga la disponibilidad de los materiales y equipos en el mercado, como la impresión de la PCB, los diferentes módulos de comunicación, circuitos protectores y las piezas faltantes para las plantas Fischertechnik. Si se encuentran limitaciones en disponibilidad o costos, se

retroalimenta el diseño para realizar los ajustes necesarios. Posteriormente, se procede a la adquisición de los componentes definitivos para la construcción e implementación del sistema.

6. Validación de componentes: Una vez adquiridos los componentes, se llevan a cabo pruebas individuales de cada uno para asegurar su correcto funcionamiento y cumplimiento de los requisitos establecidos. Esto incluye la verificación del funcionamiento de los módulos de comunicación, la PCB con sus circuitos protectores, así como de los sensores y actuadores de las plantas Fischertechnik.
7. Validación de subsistemas: En esta etapa, el proyecto se divide en diferentes subsistemas y se realizan pruebas de cada subconjunto para evaluar su comportamiento antes del ensamblaje general. Se prestará especial atención a la funcionalidad y verificación del sistema SCADA web, así como al correcto funcionamiento de las plantas Fischertechnik.
8. Validación del sistema: Se procede a la puesta en marcha del sistema SCADA, realizando pruebas con las plantas Fischertechnik. Durante esta fase, se verifica el desempeño del sistema y se establece un proceso de retroalimentación hacia todas las etapas del proyecto, permitiendo la identificación y corrección de posibles inconvenientes y errores que puedan surgir.
9. Validación de especificaciones: Finalmente, se confirma que el prototipo cumple con todos los requisitos establecidos y produce los resultados esperados, según lo verificado a través de pruebas de funcionalidad. Asimismo, se elaboraron guías de laboratorio que permiten integrar el proyecto en un entorno educativo.

10. Resultados: Tras realizar pruebas de funcionalidad y obtener la retroalimentación de una encuesta a estudiantes, se registraron los siguientes resultados del proyecto: un esquema electrónico completo, que permite la replicación del sistema; una plataforma web SCADA funcional; dos controladores lógicos programables basados en hardware libre; y tres guías de laboratorio diseñadas específicamente para las plantas Fischertechnik. Estos logros garantizan la reproducibilidad del proyecto y refuerzan su valor educativo en áreas como la automatización, redes y comunicación industrial.

## 4. RECOPILACIÓN DE INFORMACIÓN Y ANALISIS DE REQUERIMIENTOS

La recopilación de información es una fase crucial para definir los parámetros técnicos y funcionales del sistema de propuesto. Este proceso se organiza en dos etapas interrelacionadas: el análisis de controladores lógicos programables comerciales y la evaluación de plataformas SCADA basadas en la nube.

### 4.1 RECOPILACIÓN DE INFORMACIÓN CONTROLADOR COMERCIALES

En esta sección, se comparan cuatro controladores lógicos programables comerciales: FISCHERTECHNIK Robo TX Controller, M-DUINO PLC de Industrial Shields, IONO UNO de Sfera Labs y Arduino OPTA. Cada uno representa una solución de diferente enfoque, el primero es una solución propietaria para aplicaciones educativas, mientras que los tres restantes están basados en arquitecturas de hardware libre adaptadas a entornos industriales. El análisis aborda las características de entrada y salida, especificaciones técnicas, arquitectura interna, certificaciones, interfaces de comunicación, estructura de precios y configuración modular de cada controlador. Este análisis busca establecer un marco comparativo objetivo que oriente la selección de componentes y determine el enfoque más apropiado para el controlador lógico programable del presente proyecto.

Las especificaciones técnicas detalladas de los controladores se presentan en las Tablas del Anexo A.

El análisis muestra una clara evolución tecnológica en la automatización industrial. El FISCHERTECHNIK Robo TX, como solución de la generación anterior, ofrecía un procesador ARM9 y conectividad limitada a USB y Bluetooth, siendo principalmente una herramienta educativa. Con el IONO UNO, se introduce la modularidad y relés industriales integrados, mientras que el M-DUINO PLC amplía

significativamente las capacidades de E/S y conectividad inalámbrica. Finalmente, el ARDUINO OPTA representa la generación más avanzada, con un procesamiento dual-core, conectividad IoT nativa, certificaciones industriales rigurosas y capacidades de seguridad avanzadas, reflejando la transición hacia sistemas distribuidos e interconectados, propios de la Industria 4.0.

#### 4.2 RECOPILACIÓN DE INFORMACIÓN PLATAFORMAS SCADA

Esta sección compara dos plataformas SCADA basadas en la nube: Ubidots y Arduino Cloud, que representan la evolución de los sistemas SCADA tradicionales hacia arquitecturas distribuidas. Ambas permiten el monitoreo remoto en tiempo real. El análisis aborda las características técnicas, compatibilidad con dispositivos y estructura de precios para guiar la elección del sistema de supervisión adecuado para el proyecto.

La tabla comparativa detallada de las plataformas SCADA se presenta en la Tabla del Anexo B.

Se observa una transición significativa en la evolución de las plataformas SCADA, desde los sistemas tradicionales hasta soluciones basadas en la nube. Ubidots se destaca por su madurez tecnológica en plataformas IoT industriales, con una integración avanzada, escalabilidad y capacidades de supervisión que superan las limitaciones geográficas de los sistemas convencionales. En comparación, Arduino Cloud democratiza el acceso a tecnologías de monitoreo, ofreciendo una interfaz intuitiva y costos reducidos. Ubidots, por su compatibilidad con más de 60 protocolos IoT y soporte nativo para PLCs con Modbus, se posiciona como la opción preferida para entornos industriales complejos.

#### 4.3 REQUERIMIENTOS CELDA MECATRÓNICA FISCHERTECHNIK

Para identificar las dos celdas a utilizar en el proyecto, se realizó un conteo tanto de las piezas disponibles en el kit como de las necesarias para el armado de distintas celdas. El criterio de selección incluyó la integración de componentes electrónicos y neumáticos, así como la búsqueda de variedad en entradas, salidas y niveles de voltaje para abordar un mayor rango de condiciones de control. A partir de este análisis, se seleccionaron una celda de banda transportadora con sistema de clasificación y un brazo neumático para la manipulación de piezas.

La celda de banda transportadora tiene como función principal detectar una pieza, identificar su color y, dependiendo de este, permitir su paso o realizar un desvío mediante una barrera. La celda del brazo neumático detecta una pieza en un punto A y la traslada a un punto B, mediante un sistema combinado de neumática y electrónica.

La planta Fischertechnik consta de tres procesos principales: manipulación, transporte y selección. La manipulación se lleva a cabo en la celda del brazo neumático, mientras que el transporte y la selección se realizan en la celda de la banda transportadora.

COMPONENTES ELECTRÓNICOS DE LA PLANTA			
CELDA	CANTIDAD	ELEMENTO	CONEXIÓN AL CONTROLADOR
Brazo neumático	1	Fototransistor	SI
	2	Switch	SI

	3	Electroválvula	SI
	1	Led	NO
	1	Mini Motor DC	SI
	1	Compresor	NO
Banda transportadora	1	Fototransistor	SI
	1	Switch	SI
	1	Led	NO
	1	Sensor de color	SI
	2	Motor DC	SI

Tabla 1 Componentes electrónicos de la planta FISCHERTECHNIK.

A partir del análisis de los componentes electrónicos de la planta, se identificaron los elementos conectados al controlador y se realizó un análisis detallado de entradas y salidas para cada celda.

Las especificaciones técnicas detalladas de ambas celdas se presentan en las Tablas del Anexo C.

#### 4.4 REQUERIMIENTOS DEL CONTROLADOR LÓGICO PROGRAMABLE

Después de identificar el número de entradas y salidas necesarias para cada celda mecatrónica de Fischertechnik, se analiza la información recopilada sobre el controlador original. El controlador desarrollado en este proyecto deberá contar,

como mínimo, con la misma cantidad de entradas y salidas que el equipo original y ser capaz de gestionar de forma independiente una celda completa.

I/O	Robo TX Controller 9V	Propuesta Controlador Lógico Programable Proyecto
Entradas análogas (compatible con señales digitales)	8	8
Entradas digitales	4	6
Salidas PWM (para motores)	4	4
Salidas análogas	0	2
Salidas rele	0	4

Tabla 2 Entradas/Salidas FISCHERTECHNIK vs Proyecto.

La propuesta satisface tanto las entradas y salidas mínimas requeridas para cada celda como las del controlador original de Fischertechnik. En los últimos años, los modelos ofrecidos por Fischertechnik para universidades e industrias han priorizado plantas y controladores que operan a 24V, alineándose con los estándares internacionales. Por esta razón, se ha adoptado este enfoque en el presente proyecto [3].

#### 4.5 REQUERIMIENTOS DE LA PLATAFORMA SCADA WEB

Fischertechnik no dispone actualmente de una solución SCADA o IoT compatible con su controlador original para la visualización, el control y el almacenamiento de datos. Por ello, los requerimientos de este proyecto se fundamentan en el análisis de plataformas SCADA/IoT existentes.

Al construir la plataforma SCADA web desde cero, se pretende ofrecer una base sólida para su evolución futura, incorporando las funcionalidades esenciales de cualquier SCADA: supervisión, control y adquisición de datos.

Se implementará una configuración de doble controlador: uno asignado a la celda del brazo y otro a la celda de la banda transportadora. Esta configuración permitirá mostrar la interconectividad entre ambos controladores a través del backend, como se ilustra claramente en la Figura 3.

## 4.6 ESPECIFICACIONES DEL SISTEMA

Tras recopilar y analizar en profundidad los requisitos de las celdas, controladores y sensores del sistema, se han definido las especificaciones técnicas que guiarán el desarrollo tanto del PLC como de la plataforma SCADA web.

### 4.6.1 ESPECIFICACIONES DEL CONTROLADOR

Cada celda mecatrónica de Fischertechnik dispondrá de un controlador lógico programable que iguala o supera el número mínimo de entradas y salidas del Robo TX Controller original, garantizando la independencia funcional de cada celda. La compatibilidad con los niveles de tensión industrial (24V) asegura la integración sin modificaciones adicionales de hardware.

Especificaciones técnicas del controlador:

- Alimentación DC 24V
- Entradas analogas 0-10V
- Entradas digitales 24V
- Salida motor PWM 0-10V
- Salida analoga 0-10V

- Salida de relé para el manejo de cargas de alta potencia
- Protección eléctrica contra inversión de polaridad, sobrecorriente y cortocircuitos
- Switch físico para frenar el proceso en caso de emergencia
- Disipación de calor pasiva
- Conectividad: USB y WiFi
- Protocolos: Serial, Virtual Serial y Variación MIDI
- Compatible con Arduino IDE

#### 4.6.2 ESPECIFICACIONES DE LA PLATAFORMA

Al no existir en la oferta de Fischertechnik una solución SCADA/IoT compatible con su controlador, la plataforma web se diseñará desde cero para incorporar las funcionalidades básicas de supervisión, control y adquisición de datos. Este desarrollo atenderá las tendencias de la Industria 4.0, habilitando la interconectividad de los dos controladores a través de un gateway.

Especificaciones técnicas de la plataforma:

- Basada en tecnologías web
- Arquitectura de backend, frontend y gateway desacoplado
- Protocolos: Serial, Variación MIDI
- Almacenamiento persistente de variables con su respectivo vector de tiempo
- Posibilidad de escribir secuencias en lenguaje de alto nivel
- Portabilidad entre sistemas operativos
- Librería para el manejo de entradas/salidas

En conjunto, estos elementos definen un sistema modular, independiente y preparado tanto para los requerimientos actuales de la planta como para su evolución hacia entornos industriales inteligentes.

## 5. DISEÑO Y FABRICACIÓN CONTROLADOR

Este capítulo presenta el desarrollo completo del controlador electrónico, desde la selección de componentes hasta la fabricación final de la tarjeta de circuito impreso. Se detallan los criterios técnicos empleados, los cálculos de diseño realizados y las validaciones mediante simulación que garantizan el correcto funcionamiento del controlador lógico programable del proyecto.

### 5.1 SELECCIÓN DE COMPONENTES

Tras analizar los requerimientos establecidos en la sección 4.6.1, se seleccionaron los siguientes componentes para el controlador:

Subsistema	Componente Seleccionado	Requerimiento Satisfecho
Microcontrolador	ATmega2560	Número de E/S, compatibilidad con Arduino IDE
Comunicación	ESP8266	Conectividad WiFi según especificación
Entradas Digitales	Optoacoplador: EL817S1 Inversor Schmitt: 74HC14	Entradas digitales 24V Protección eléctrica Compatibilidad de señales
Entradas Analógicas	Amplificador L358P	Entradas analógicas 0-10V
Salidas PWM	Driver L293D	Salida motor PWM 0-10V
Salidas Analógicas	Transistor TIP31C	Salida analógica 0-10V

Salidas Relé	Relé G5Q-1A-DC24	Salida de relé para cargas de alta potencia
Fuente	Módulo LM2596 Módulo LM7805	Alimentación DC 24V Protección contra inversión Protección contra sobrecorriente

El análisis detallado de la selección de componentes con las matrices Pugh comparativas y justificaciones se encuentra en el Anexo D: Selección de Componentes Electrónicos.

## 5.2 DISEÑO DE CIRCUITOS ELECTRONICOS

La implementación de los circuitos electrónicos se realiza siguiendo un enfoque sistemático que garantiza el correcto funcionamiento del sistema. El controlador se estructura en cinco módulos principales, cada uno correspondiente a una función específica:

- Módulo I/O: Circuitos de manipulación de entradas y salidas
- Módulo Shield Adaptador: Organización estratégica de pines mediante ruteo
- Módulo CPU: Arduino Mega 2560 R3
- Módulo Fuente: LM2596/LM7805
- Módulo WiFi: ESP8266

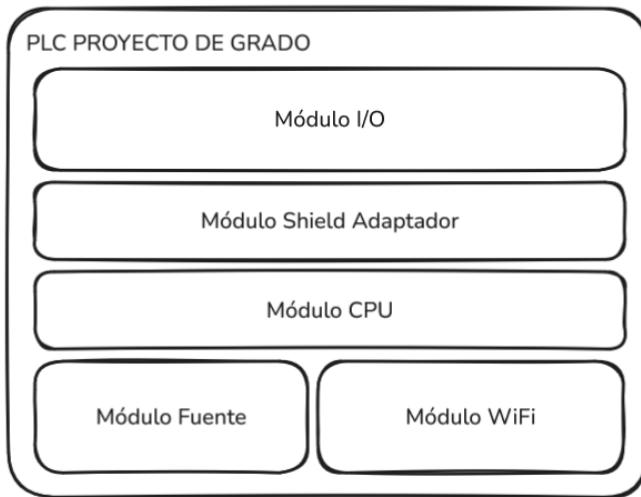


Figura 5 Composición modular PLC proyecto de grado.

El único módulo que requiere una etapa de diseño personalizado es el Módulo I/O, el cual se desarrolla siguiendo la metodología que se describe a continuación:

1. Presentación del diagrama esquemático del circuito
2. Cálculo detallado de parámetros críticos (voltaje, corriente y potencia)
3. Validación mediante simulación electrónica

Los módulos restantes se adquieren en presentaciones comerciales listas para montaje y funcionamiento, lo que simplifica su implementación en el controlador y reduce los tiempos de desarrollo.

### 5.2.1 ENTRADAS DIGITALES

El circuito de entradas digitales integra el optoacoplador EL817S1 y el inversor Schmitt 74HC14, incorporando además un LED indicador que permite visualizar el estado de la señal.

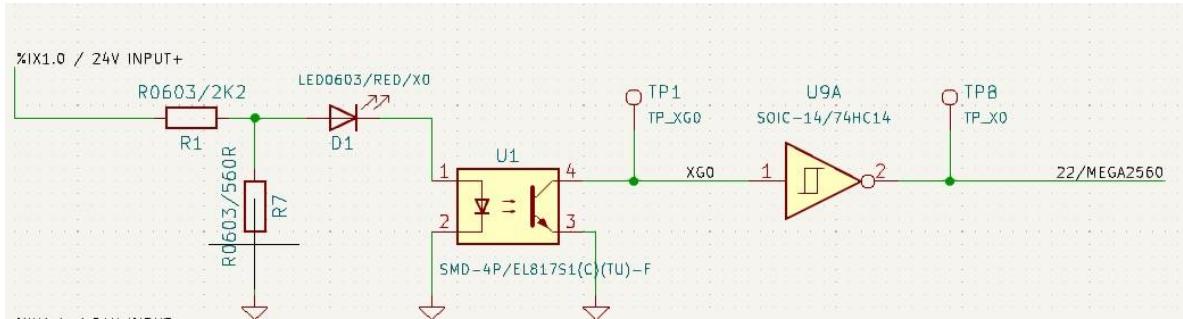


Figura 6 Circuito entradas digitales.

### Cálculos de parámetros eléctricos

La señal industrial de 24 V se conecta a un divisor de voltaje formado por resistencias en formato SMD 0603:

- $R1 = 2200 \Omega$
- $R7 = 560 \Omega$

El voltaje en el punto medio del divisor se calcula como:

- $V_{div} = V_{in} \times (R7 / (R1 + R7))$   $V_{div} = 24 V \times (560 \Omega / (2200 \Omega + 560 \Omega)) = 24 \times (560 / 2760) = 4.87 V$

Las caídas de tensión en los componentes LED según hojas de datos son [43]:

- LED rojo:  $V_f_{LED} \approx 1.8 V$
- LED optoacoplador:  $V_f_{opto} \approx 1.2 V$
- Caída total:  $V_f_{total} = 1.8 V + 1.2 V = 3.0 V$

El voltaje real en el nodo del divisor se limita a:

- $V_{div\_real} = 3.0 V$

Las corrientes en las resistencias y LED se calculan:

- $IR1 = (Vin - V_{div\_real}) / R1$   $IR1 = (24 \text{ V} - 3.0 \text{ V}) / 2200 \Omega = 21 \text{ V} / 2200 \Omega = 9.55 \text{ mA}$
- $IR2 = V_{div\_real} / R2$   $IR2 = 3.0 \text{ V} / 560 \Omega = 5.36 \text{ mA}$
- $I_{LED} = IR1 - IR2 = 9.55 \text{ mA} - 5.36 \text{ mA} = 4.19 \text{ mA}$

Las potencias disipadas en las resistencias son:

- $PR1 = VR1 \times IR1 = 21 \text{ V} \times 0.00955 \text{ A} = 0.2006 \text{ W}$   $PR2 = VR2 \times IR2 = 3.0 \text{ V} \times 0.00536 \text{ A} = 0.0161 \text{ W}$

A partir de los cálculos efectuados, se lleva a cabo la simulación para verificar los valores de corriente:

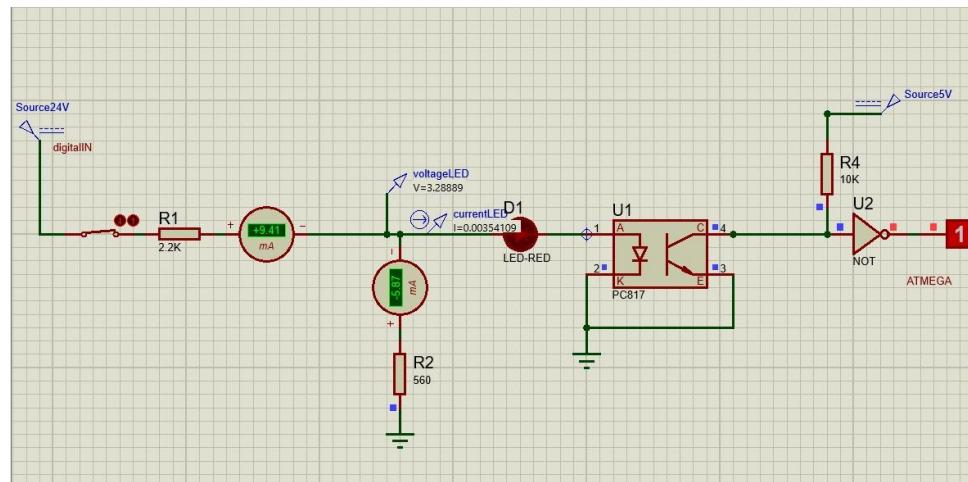


Figura 7 Simulación entradas digitales.

La simulación determina los umbrales efectivos de la entrada digital: 0–11 V (nivel bajo) y 13,5–24 V (nivel alto), proporcionando un adecuado margen de ruido.

## 5.2.2 ENTRADAS ANALÓGICAS

Las entradas analógicas se diseñan con doble funcionalidad para procesar señales de 0–10 V analógicas y señales de 0–24 V digitales. El circuito incorpora diodos Zener para protección contra sobretensiones y un amplificador operacional configurado como seguidor de tensión.

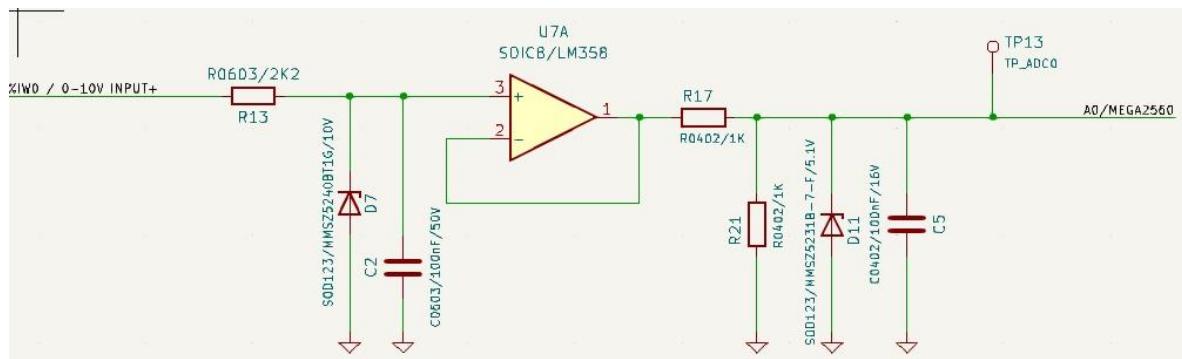


Figura 8 Circuito entradas analógicas.

### Cálculos de parámetros eléctricos

La corriente a través de la resistencia limitadora R1 se calcula para el caso de máxima tensión (24 V):

- $IR1 = (Vin - Vz) / R1$   $IR1 = (24 V - 10 V) / 2200 \Omega = 14 V / 2200 \Omega = 6.36 \text{ mA}$

La impedancia de entrada del LM358 es muy alta (típicamente  $> 1 \text{ M}\Omega$ ), por lo que:

$$I_{pin3} \approx 0$$

Por la ley de corrientes de Kirchhoff:

- $I_z = IR1 - I_{pin3} \approx 6.36 \text{ mA} - 0 = 6.36 \text{ mA}$

El pin 6 (entrada inversora) está conectado al pin 1 (salida) configurando un seguidor de voltaje que replica el voltaje de 10 V en la entrada no inversora.

El voltaje teórico en el nodo:

- $V_{nodo} = V_{out} \times (R_3 / (R_2 + R_3))$   $V_{nodo} = 10 \text{ V} \times (1000 \Omega / (1000 \Omega + 1000 \Omega)) = 10 \text{ V} \times (1000 / 2000) = 5 \text{ V}$

El diodo Zener de 5 V fija el voltaje en este nodo para proteger la entrada del microcontrolador.

Las potencias disipadas en los componentes son:

Resistencia 2,2 kΩ:

- $P_{R1} = I_{R1}^2 \times R_1 = (0.00636)^2 \times 2200 \approx 88.5 \text{ mW}$  (menor que los 250 mW nominales)

Diodo Zener 10 V:

- $P_{Dz} = V_z \times I_z = 10 \text{ V} \times 0.00636 \text{ A} = 63.6 \text{ mW}$  (menor que los 500 mW nominales)

Diodo Zener 5 V:

- $P_{Dz2} = V_z \times I_z = 5 \text{ V} \times 0.00004 \text{ A} = 0.2 \text{ mW}$  (valor despreciable)

A partir de los cálculos efectuados, se lleva a cabo la simulación para verificar los valores de corriente:

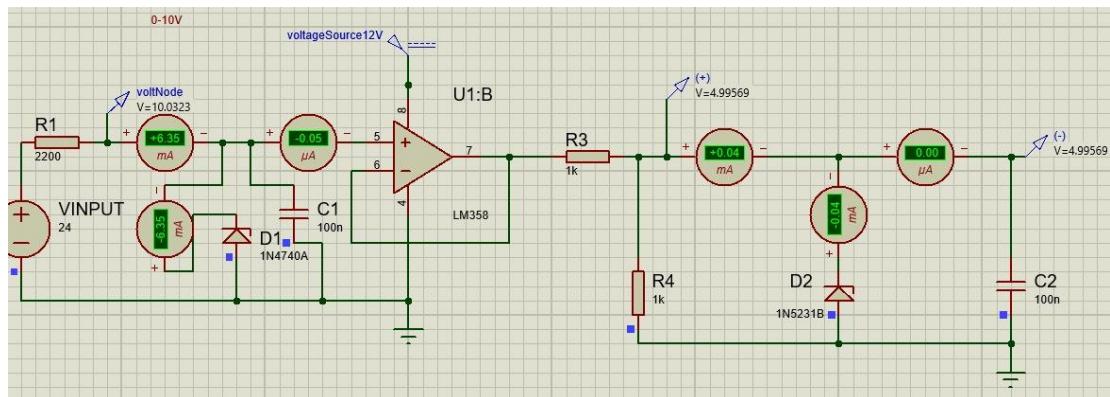


Figura 9 Simulación entradas analógicas.

### 5.2.3 SALIDAS ANALÓGICAS

El circuito de salidas analógicas convierte una señal PWM en una señal continua de 0-10V mediante un filtro RC seguido de un amplificador y un transistor de potencia.

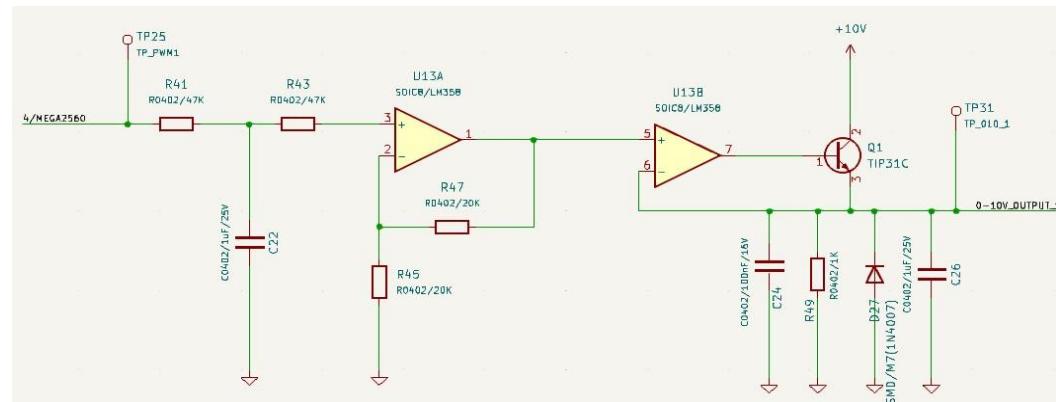


Figura 10 Circuito salidas analógicas.

### Cálculos de parámetros eléctricos

Los parámetros del filtro RC son:

- $R1 = 47 \text{ k}\Omega$
- $C1 = 1 \mu\text{F}$
- Frecuencia PWM = 980 Hz

Periodo de la señal PWM:

- $T = 1 / f_{\text{PWM}} = 1 / 980 \text{ Hz} = 1.02 \text{ ms}$

Constante de tiempo del filtro:

- $\tau = R1 \times C1 = 47000 \Omega \times 1 \times 10^{-6} \text{ F} = 0.047 \text{ s} = 47 \text{ ms}$

El rizado pico a pico para un filtro RC con entrada PWM se calcula:

- $V_{\text{ripple\_pp}} = (V_{\text{in}} \times D \times (1 - D)) / (f_{\text{PWM}} \times R1 \times C1)$

Para ciclo de trabajo 50% ( $D = 0.5$ ):

- $V_{\text{ripple\_pp}} = (5 \text{ V} \times 0.5 \times (1 - 0.5)) / (980 \text{ Hz} \times 47000 \Omega \times 1 \times 10^{-6} \text{ F}) = 27.1 \text{ mV}$

Para ciclo de trabajo 100% ( $D = 1$ ):

- $V_{\text{ripple\_pp}} = (5 \text{ V} \times 1 \times (1 - 1)) / (980 \text{ Hz} \times 47000 \Omega \times 1 \times 10^{-6} \text{ F}) = 0 \text{ V} (\text{sin rizado})$

El rizado es de 27.1 mV pico a pico al 50% de PWM (1.08% de  $V_{\text{avg}} = 2.5 \text{ V}$ ), resultando despreciable para esta aplicación ya que  $\tau \gg T$  ( $47 \text{ ms} \gg 1.02 \text{ ms}$ ).

El tiempo de respuesta del filtro se calcula mediante:

- $V_c(t) = V_{\text{final}} \times (1 - e^{-(t/\tau)})$

De 0 V a 5 V (PWM 0% - 100%):

- $t_{99\%} = -\tau \times \ln(1 - 4.95/5) = -0.047 \text{ s} \times \ln(0.01) \approx 0.2164 \text{ s} = 216.4 \text{ ms}$

El amplificador no inversor tiene una ganancia:

- $A_v = 1 + R_f / R_g = 1 + 20 \text{ k}\Omega / 20 \text{ k}\Omega = 2$

Salida para diferentes ciclos de trabajo:

- Al 50% de PWM:  $V_{\text{out1}} = 2 \times 2.5 \text{ V} = 5 \text{ V}$
- Al 100% de PWM:  $V_{\text{out1}} = 2 \times 5 \text{ V} = 10 \text{ V}$

Se seleccionan  $R_f = R_g = 20 \text{ k}\Omega$  como valores estándar para lograr ganancia 2 con corrientes bajas (microamperios) que aseguran bajo consumo.

El transistor TIP31C funciona como seguidor de emisor donde:

- $V_{\text{emisor}} = V_{\text{out1}}$  (la realimentación negativa compensa VBE)

Las potencias disipadas en los componentes son:

Resistencia de filtro ( $R_1 = 47 \text{ k}\Omega$ ):

- $\text{IRMS} \approx \text{Vripple\_pp} / (2 \times \sqrt{3} \times R_1) = 0.0271 \text{ V} / (2 \times 1.732 \times 47000 \Omega) \approx 0.16 \mu\text{A}$
- $P_{R1} = \text{IRMS}^2 \times R_1 = (0.16 \times 10^{-6} \text{ A})^2 \times 47000 \Omega \approx 0.0012 \text{ mW}$

Resistencias del amplificador ( $R_f = R_g = 20 \text{ k}\Omega$ ) para 100% PWM:

- $I = 5 \text{ V} / 20000 \Omega = 250 \mu\text{A}$
- $P = (250 \times 10^{-6} \text{ A})^2 \times 20000 \Omega = 1.25 \text{ mW}$

Resistencia de emisor ( $R_E = 1 \text{ k}\Omega$ ) para 100% PWM:

- $I_E = 10 \text{ V} / 1000 \Omega = 10 \text{ mA}$
- $P_{RE} = (10 \times 10^{-3} \text{ A})^2 \times 1000 \Omega = 100 \text{ mW}$

A partir de los cálculos efectuados, se lleva a cabo la simulación para verificar los valores de corriente:

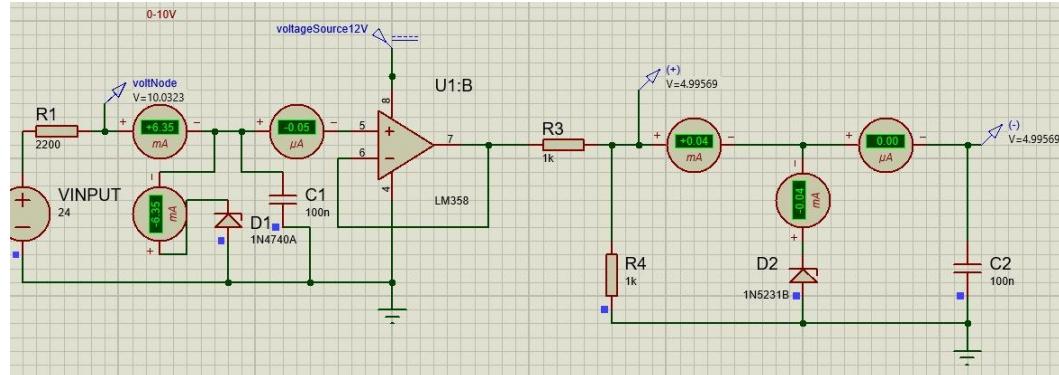


Figura 11 Simulación salidas analógicas

#### 5.2.4 SALIDAS MOTORES PWM

El circuito de control para motores utiliza el puente-H L293D que incorpora diodos de protección integrados, simplificando el diseño y reduciendo el número de componentes externos.

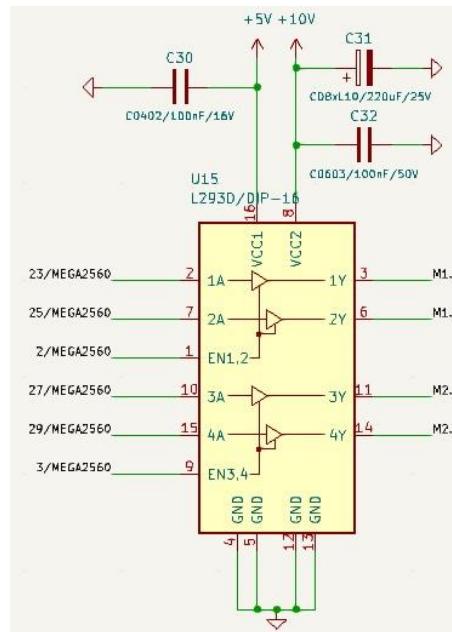


Figura 12 Circuito salidas motores PWM.

El circuito se complementa únicamente con los condensadores de desacoplo recomendados por el fabricante para garantizar la estabilidad de la alimentación durante los picos de corriente asociados a la conmutación de los motores.

Se lleva a cabo la simulación para verificar el funcionamiento:

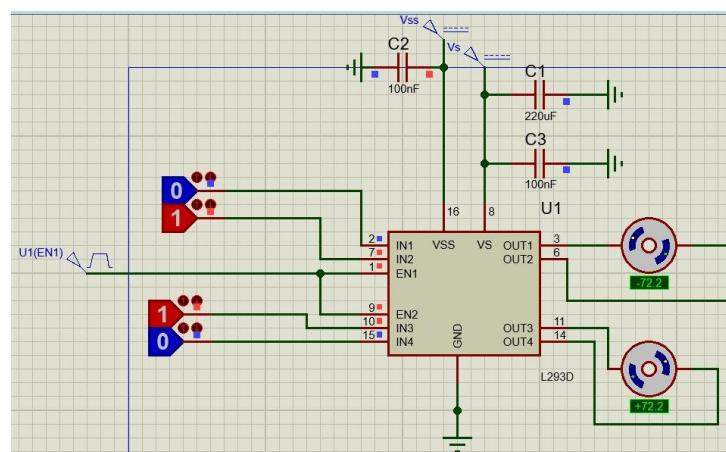


Figura 13 Simulación salidas motores PWM.

## 5.2.4 SALIDAS DIGITALES RELÉ

Para las salidas a relé se implementa un circuito que utiliza un arreglo de transistores Darlington. Esta configuración proporciona la ganancia de corriente necesaria para activar la bobina del relé G5Q-1A-DC24 de manera confiable.

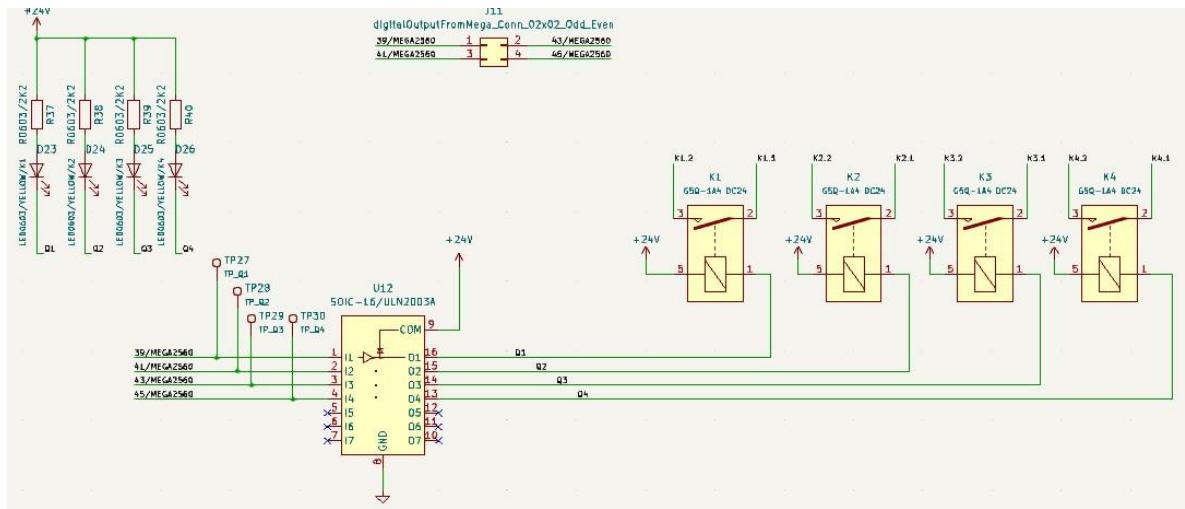


Figura 14 Circuito salidas digitales relé.

### Cálculos de parámetros eléctricos

La corriente en el circuito se calcula como:

- $I = (V_{total} - V_{LED}) / R = (24 \text{ V} - 2.0 \text{ V}) / 2200 \Omega = 22 \text{ V} / 2200 \Omega = 10 \text{ mA}$

Las potencias disipadas son:

Resistencia limitadora:

- $P_R = I^2 \times R = (0.01 \text{ A})^2 \times 2200 \Omega = 0.22 \text{ W} (220 \text{ mW})$

LED indicador:

- $P_{LED} = V_{LED} \times I = 2.0 \text{ V} \times 0.01 \text{ A} = 0.02 \text{ W} (20 \text{ mW})$

Todos los valores calculados se mantienen dentro de los márgenes de operación segura para los componentes seleccionados.

A partir de los cálculos efectuados, se lleva a cabo la simulación para verificar los valores de corriente:

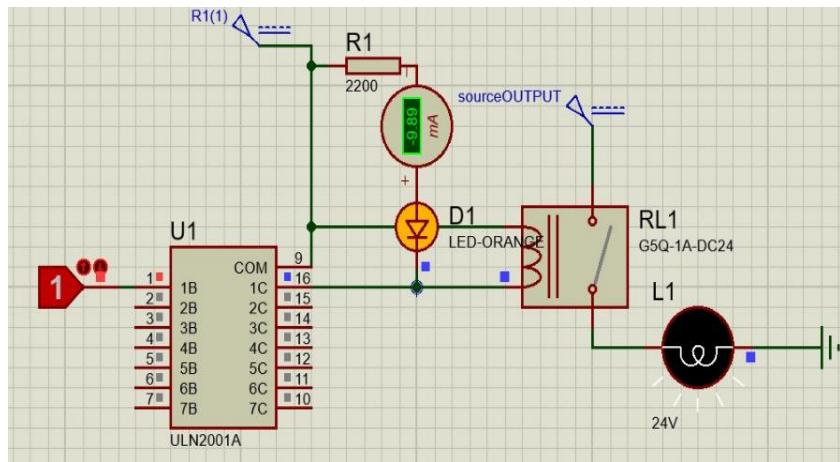


Figura 15 Simulación salidas digitales relé.

## 5.5 FABRICACIÓN PCB

En este apartado se describen, de manera detallada y sistemática, los criterios seguidos para la fabricación y montaje de la tarjeta de circuito impreso (PCB). El análisis incluye la selección de materiales, los parámetros técnicos y las limitaciones de fabricación establecidas por el proveedor JLCPCB, cuyas especificaciones técnicas se encuentran disponibles en su plataforma oficial [44].

Las reglas de diseño implementadas se describen detalladamente en el Anexo E: Reglas de diseño aplicadas.

A continuación se presentan las representaciones gráficas de ambas caras del módulo I/O. Estas ilustraciones muestran la distribución de componentes, trazados de señal, planos de cobre y detalles de interconexión que garantizan la funcionalidad eléctrica y térmica del sistema.

### **Capa superior**

La capa superior presenta los pads para componentes SMD y THT distribuidos según el esquema de diseño. Se observan las pistas de señal, los planos de cobre para alimentación y tierra, y la serigrafía blanca sobre máscara de soldadura negra que identifica componentes y polaridades. Las vías tentadas proporcionan interconexión con la capa inferior y mejoran la disipación térmica.

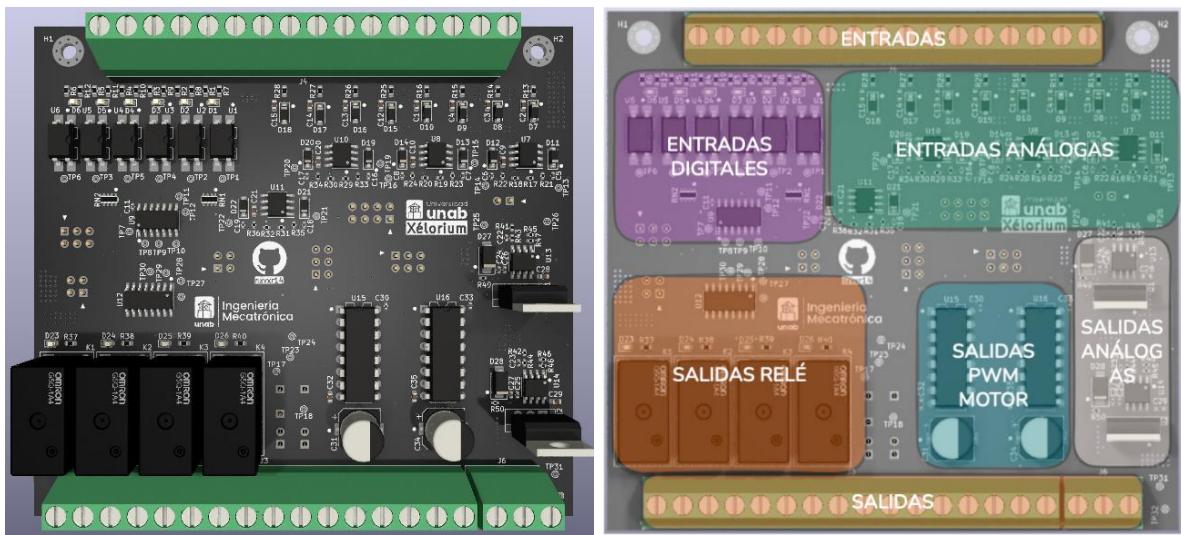


Figura 16 Modelo 3D capa superior módulo I/O.

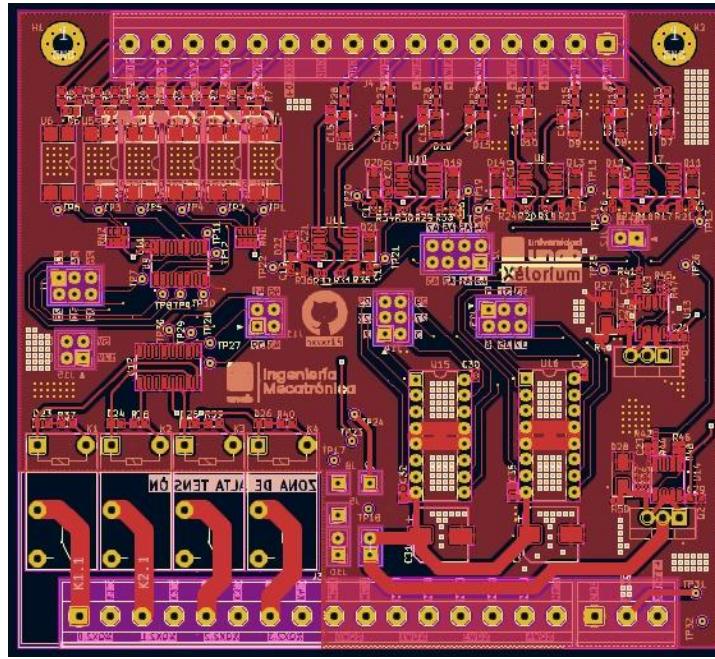


Figura 17 Ruteado capa superior módulo I/O.

### Capa inferior

La capa inferior integra principalmente el plano de tierra y las rutas de alimentación secundaria, con trazas de mayor anchura para asegurar la capacidad de corriente requerida.

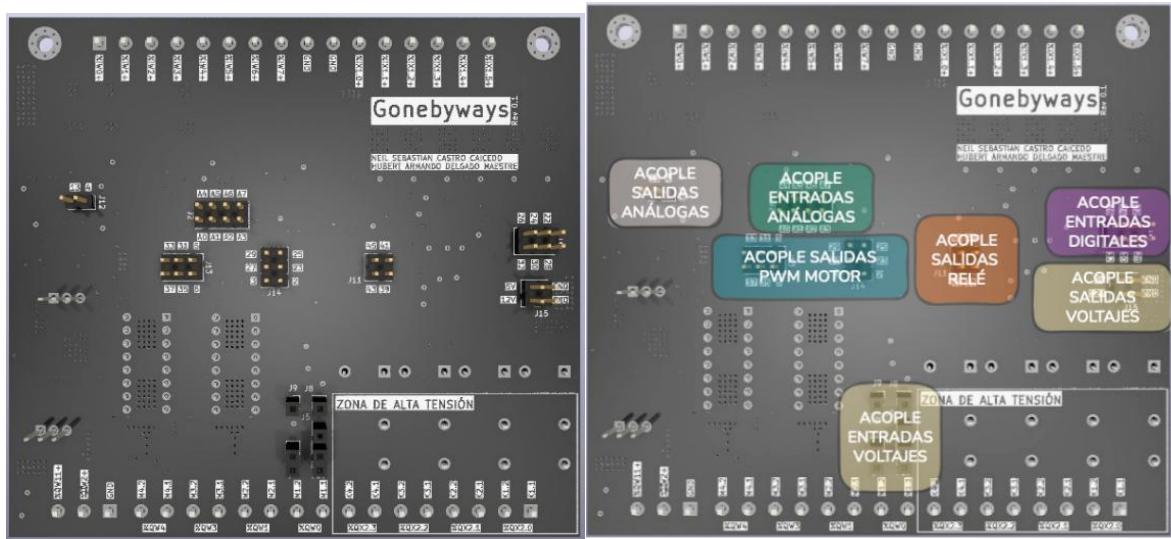


Figura 18 Modelo 3D capa inferior módulo I/O.

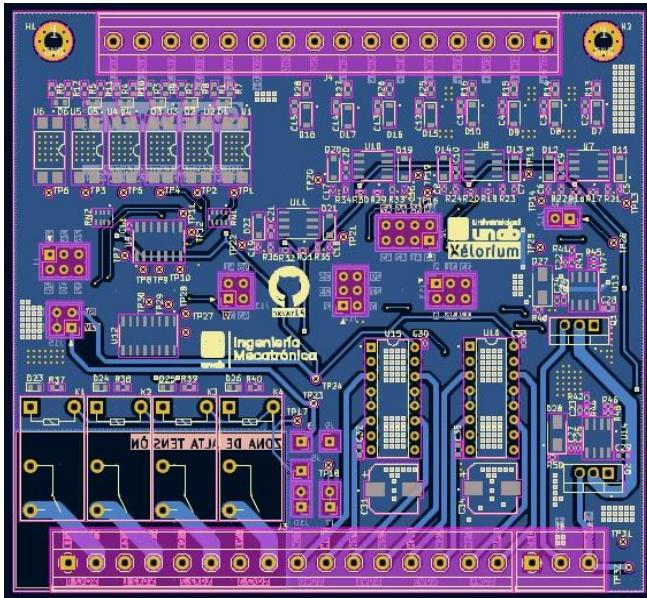


Figura 19 Ruteado capa inferior módulo I/O.

En el Anexo F: Fabricación PCB Shield Adaptador se presenta la representación gráfica del shield adaptador que acopla el microcontrolador Arduino Mega 2560 R3 con la tarjeta principal del controlador lógico programable desarrollado en el proyecto.

Como resultado del proceso de fabricación realizado con el proveedor JLCPCB mediante pedido en línea, se obtienen los módulos I/O y shield adaptador finales:

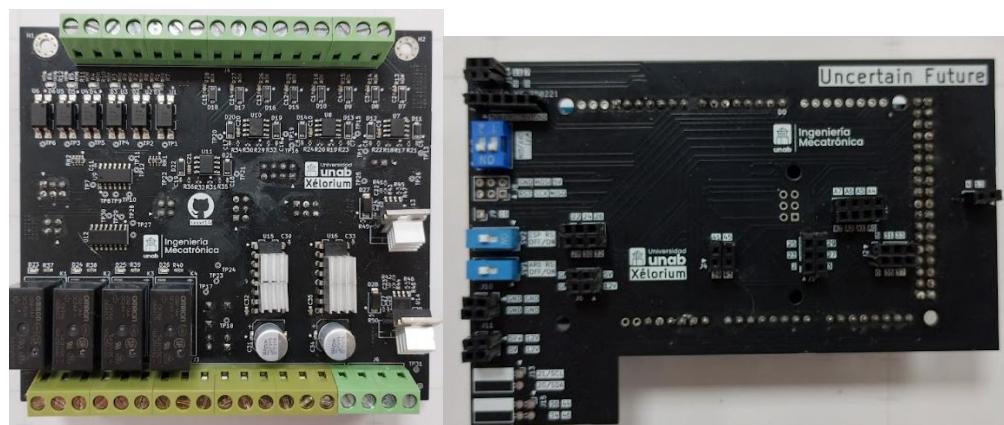


Figura 20 Módulo I/O y módulo shield adaptador fabricados.

## 6. DISEÑO 3D Y FABRICACIÓN

Este capítulo describe la construcción física del sistema, abarcando el diseño y fabricación de la carcasa del controlador, el armado de las celdas Fischertechnik y la integración final de todos los componentes. La fabricación se desarrolló con un enfoque modular que facilitó la organización y posterior mantenimiento del sistema.

### 6.1 CONSIDERACIONES CARCASA DEL CONTROLADOR

El diseño de la carcasa se realizó específicamente para este proyecto, adaptándose a la arquitectura modular del controlador y optimizando el espacio interior. La carcasa incorpora elementos inspirados en controladores industriales como el Siemens S7-1200, particularmente la implementación de compuertas frontales para facilitar acceso a conexiones.

El modelo CAD se desarrolló tomando como base las dimensiones exactas de los circuitos electrónicos, garantizando un ajuste preciso de todos los componentes.

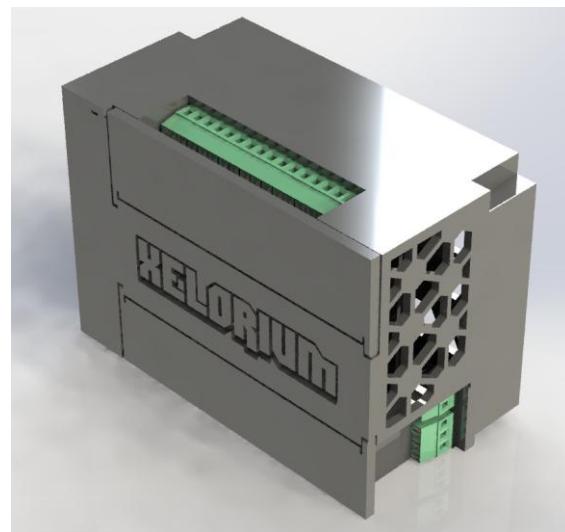


Figura 21 Vista isométrica CAD carcasa controlador.

Este diseño resultó en las siguientes dimensiones finales:

Parámetro	Valor aproximado [cm]
Alto	75
Largo	145
Ancho	108

Tabla 3 Dimensiones finales carcasa controlador.

#### 6.1.1 FABRICACIÓN CARCASA CONTROLADOR

Para seleccionar el método de fabricación más adecuado, se evaluaron tres alternativas mediante una matriz Pugh considerando los siguientes criterios:

- Soporte para geometrías complejas: Capacidad del método para producir formas intrincadas como esquinas redondeadas, ranuras para ventilación y soportes internos para PCB.
- Repetibilidad: Consistencia en la producción de múltiples unidades con las mismas características y tolerancias.
- Tiempo de fabricación: Duración del proceso completo desde el diseño hasta la obtención de la pieza final.
- Accesibilidad: Disponibilidad de equipos y materiales.
- Post-procesado: Requerimientos de acabado adicional tras la fabricación inicial.
- Complejidad de uso: Nivel de experticia técnica necesaria para operar la tecnología.
- Costo: Inversión económica total considerando materiales, tiempo de máquina y mano de obra.

Criterio	Impresión 3D (PLA)	Corte láser (acrílico)	Mecanizado CNC
Soporte para geometrías complejas	1	-1	1
Repetibilidad	1	0	0
Tiempo de fabricación	1	1	-1
Accesibilidad	0	0	-1
Post-procesado	0	1	-1
Complejidad de uso	0	0	-1
Costo	1	-1	-1
TOTAL	4	0	-4

Tabla 4 Matriz pugh selección de fabricación/material.

La impresión 3D resultó el método óptimo, destacando por su bajo costo, capacidad para producir geometrías complejas como los soportes internos para placas, y su accesibilidad. Adicionalmente, este método permitió iterar rápidamente el diseño ante modificaciones requeridas durante el desarrollo del proyecto. Los detalles completos del proceso de fabricación se encuentran en el Anexo G: Diseño y fabricación de la carcasa.

Como resultado se obtuvo la siguiente carcasa:



Figura 22 Vista isométrica carcasa fabricada.

## 6.2 DISEÑO Y VALIDACIÓN CELDAS FISCHERTECHNIK

El diseño de las celdas se realizó utilizando el software Fischertechnik Design Studio [45], que permitió validar la configuración estructural antes del montaje físico. El análisis de componentes reveló que:

- La banda transportadora requirió 349 piezas
- El brazo neumático necesitó 213 piezas
- Se identificó un déficit de 19 piezas en el kit disponible

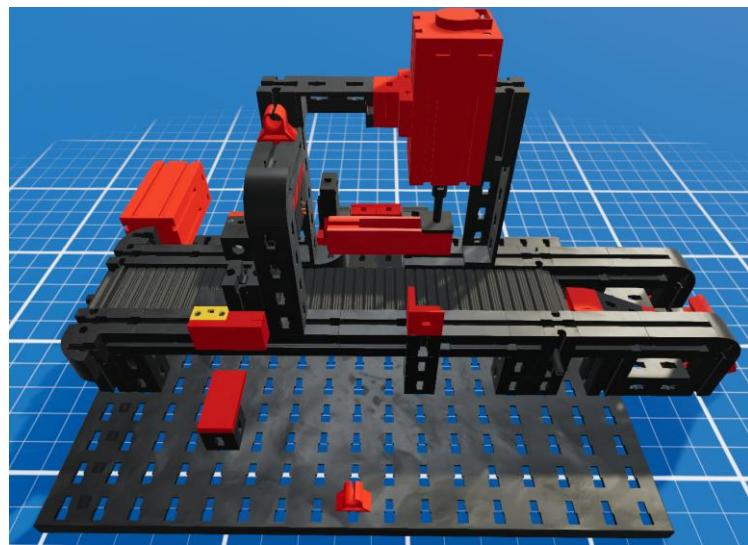


Figura 23 Modelo banda transportadora en ft Desing.

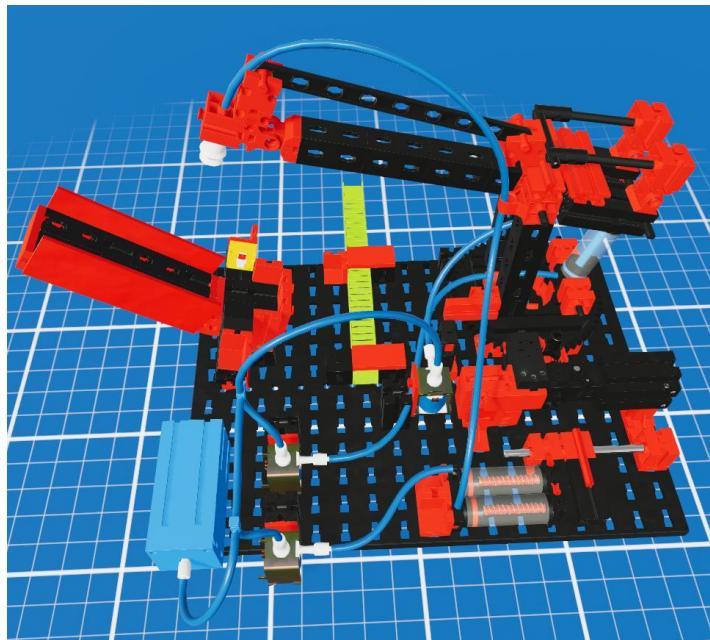


Figura 24 Modelo brazo neumático en ft Desing.

Las piezas faltantes se reconstruyeron digitalmente a partir de los modelos CAD disponibles en la biblioteca del software Fischertechnik, y se fabricaron mediante impresión 3D. La información detallada de cada componente reconstruido se encuentra en el Anexo H: Reconstrucción piezas faltantes.

### 6.3 INTEGRACIÓN DEL SISTEMA

La integración de los componentes se planificó mediante un esquema que definió la disposición física del sistema, optimizando el espacio y facilitando las conexiones entre controladores y celdas.

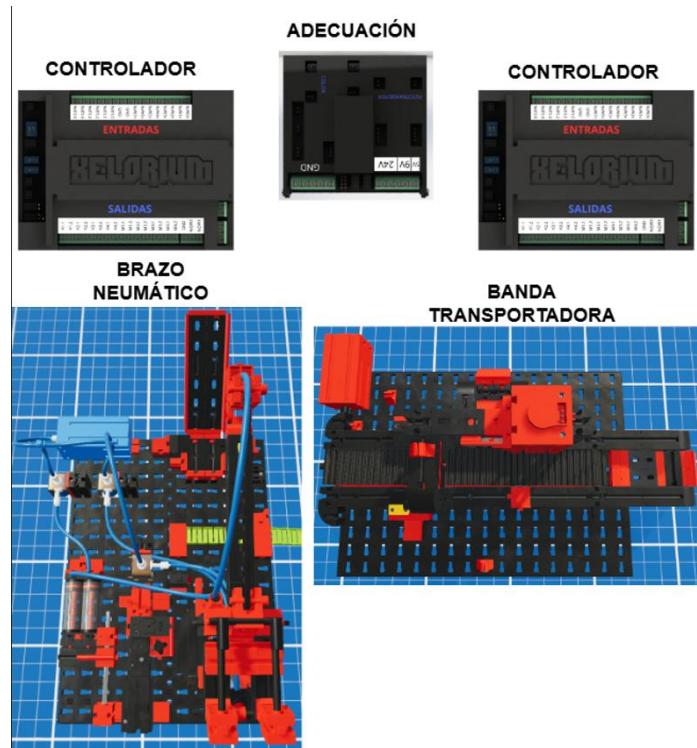


Figura 25 Boceto ubicación del sistema.

Las conexiones se organizaron en tres categorías principales:

1. Controlador - Banda transportadora
2. Controlador - Brazo neumático
3. Sistema neumático del brazo

Este enfoque estructurado facilitó tanto la instalación como el posterior diagnóstico y mantenimiento del sistema. Las tablas detalladas de conexiones y diagramas se presentan en el Anexo I: conexiones del sistema.

## 7. DESARROLLO PLATAFORMA SCADA WEB

La plataforma SCADA web desarrollada proporciona una solución moderna para la supervisión, control y adquisición de datos en entornos industriales educativos. Este sistema permite la monitorización en tiempo real de las celdas Fischertechnik, el control remoto de sus funciones y el almacenamiento de datos para análisis posterior. La plataforma se ha diseñado con tecnologías de código abierto para garantizar flexibilidad, escalabilidad e interoperabilidad entre diferentes sistemas. Las consideraciones detalladas sobre la selección de tecnologías se presentan en el Anexo J: Tecnologías y consideraciones de diseño.

### 7.1 METODOLOGÍA DE DESARROLLO

El desarrollo de la plataforma se realizó siguiendo el patrón arquitectónico Modelo-Vista-Controlador (MVC), que permite separar claramente las responsabilidades dentro del sistema:

- **Modelo:** Representa la estructura de datos y la lógica de negocio, implementada en el backend a través de MongoDB y Mongoose.
- **Vista:** Constituye la interfaz de usuario, desarrollada con React y TypeScript, proporcionando una experiencia interactiva y responsive.
- **Controlador:** Gestiona la comunicación entre el modelo y la vista, procesando solicitudes y orquestando el flujo de datos en todo el sistema.

Esta arquitectura se complementa con un enfoque de desarrollo modular que facilita la extensibilidad y el mantenimiento del código.

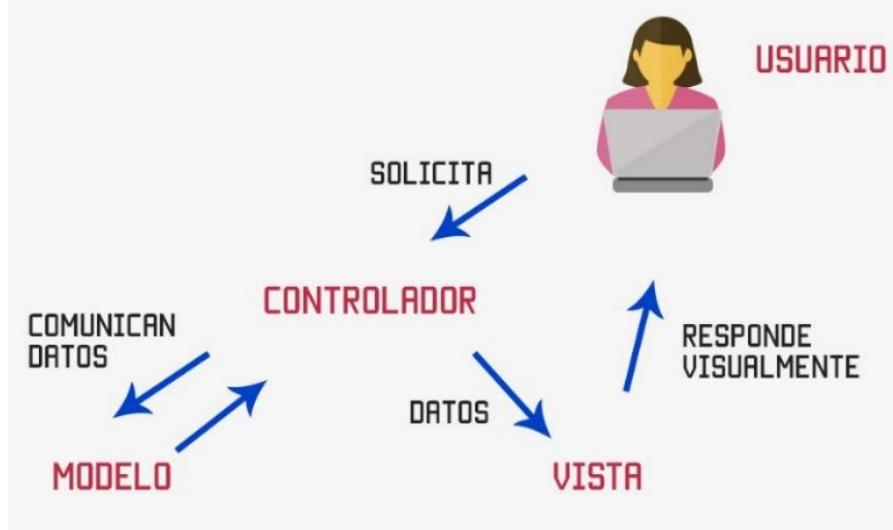


Figura 26 Patrón MVC.

## 7.2 ARQUITECTURA DEL SISTEMA

La plataforma se estructura en tres componentes principales interconectados:

1. Frontend: Interfaz de usuario desarrollada con React y TypeScript
2. Backend de datos: Servidor Node.js con Express y MongoDB
3. Gateway: Intermediario para comunicación con dispositivos físicos

Esta separación permite una mayor flexibilidad y escalabilidad, facilitando la adaptación a diferentes entornos industriales y educativos.

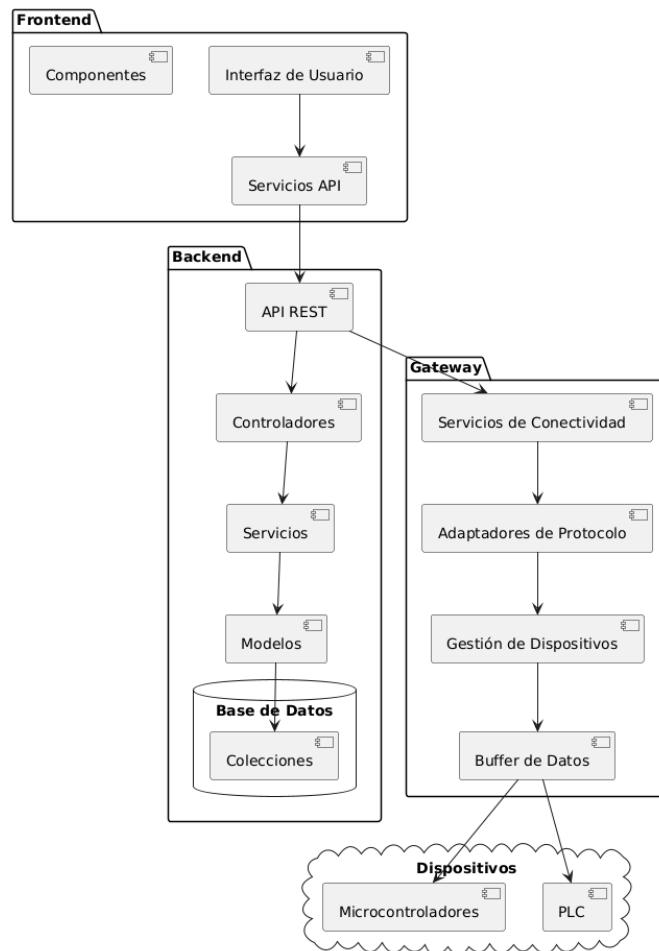


Figura 27 Arquitectura del sistema general, incluido controlador lógico programable proyecto.

### 7.3 COMPONENTES DEL SISTEMA

Como se observa en la figura 27, la arquitectura MVC está compuesta por componentes, entre los cuales destacan:

Componente	Función	Tecnología
Interfaz de Usuario	Presentación visual de datos y controles	React, TypeScript
API REST	Puntos de acceso para operaciones CRUD	Express

Base de Datos	Almacenamiento persistente de datos	MongoDB
Servicios de Conectividad	Comunicación con dispositivos	Socket.io, MQTT
Adaptadores de Protocolo	Traducción entre formatos de mensaje	Node.js
Dispositivos	Enviar y recibir datos del controlador lógico programable a la plataforma y viceversa	C

Tabla 5 Resumen componentes principales del sistema.

Todos los componentes del sistema observados en la figura 27 se detallan en el Anexo K: Arquitectura detallada de la plataforma.

## 7.4 BASE DE DATOS

La plataforma utiliza una base de datos NoSQL (MongoDB) con tres modelos principales: Project, Board y DataVar. Estos modelos permiten organizar la información de manera estructurada y eficiente, como se detalla en el Anexo L: Modelos de base de datos. La estructura de datos permite almacenar tanto la configuración de los controladores como los valores históricos de las variables monitoreadas.

- Un Project puede tener múltiples Boards (relación uno a muchos)
- Un Project puede tener múltiples DataVars (relación uno a muchos)
- Cada Board y DataVar pertenece a un único Project (relación muchos a uno)

## 7.5 CONEXIÓN CON EL CONTROLADOR LÓGICO PROGRAMABLE

El controlador lógico programable desarrollado en el capítulo 5 se integra con la plataforma SCADA a través del componente Gateway. Esta conexión se realiza mediante dos protocolos principales:

1. Protocolo Serial: Utiliza la comunicación USB-Serial del Arduino MEGA para transmitir datos y recibir comandos, ideal para configuraciones locales o de depuración.
2. Protocolo MIDI (variante): Implementado a través del módulo WiFi ESP8266, permite comunicación inalámbrica para supervisión y control remoto, utilizando una estructura de mensajes similar al protocolo MIDI pero adaptada para las necesidades específicas del proyecto.

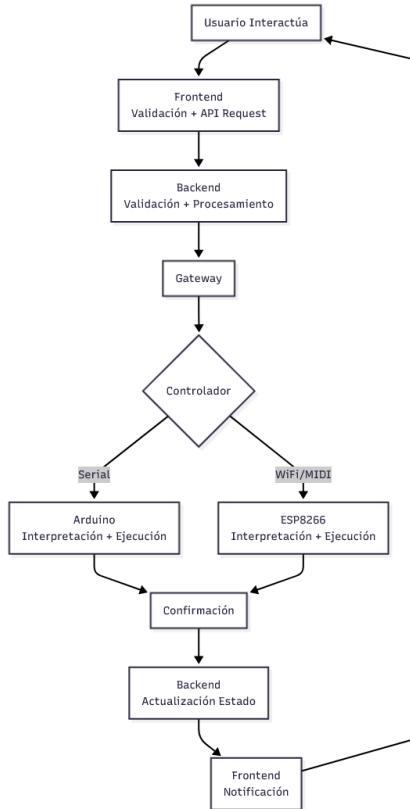


Figura 28 Flujo manipulación actuadores.

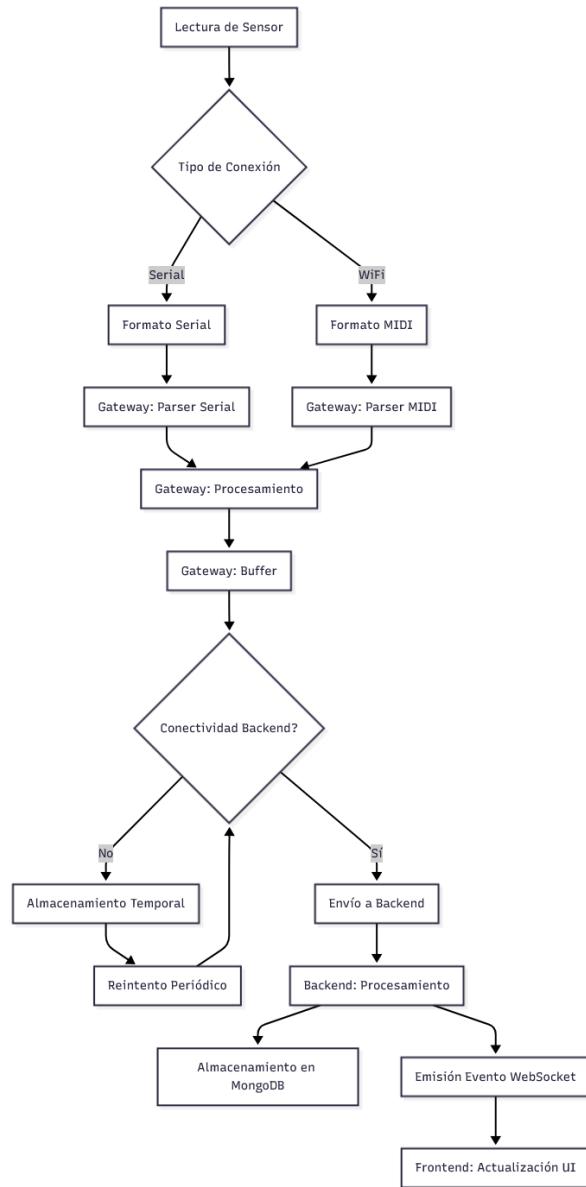


Figura 29 Flujo de lectura de sensores.

La implementación técnica detallada de la comunicación se puede consultar en el Anexo M: Implementación de protocolos de comunicación.

## 7.6 ESTRUCTURA DE ARCHIVOS

Cada capa principal (Backend, Frontend y Gateway) cuenta con una estructura de archivos específica que refleja sus responsabilidades en el sistema. La información detallada de la implementación se encuentra en el Anexo N: Estructura de la raíz de archivos.

## 7.7 FLUJO GENERAL DE DATOS DEL SISTEMA

El sistema implementa un flujo de datos bidireccional que permite la interacción entre usuario y dispositivos físicos:

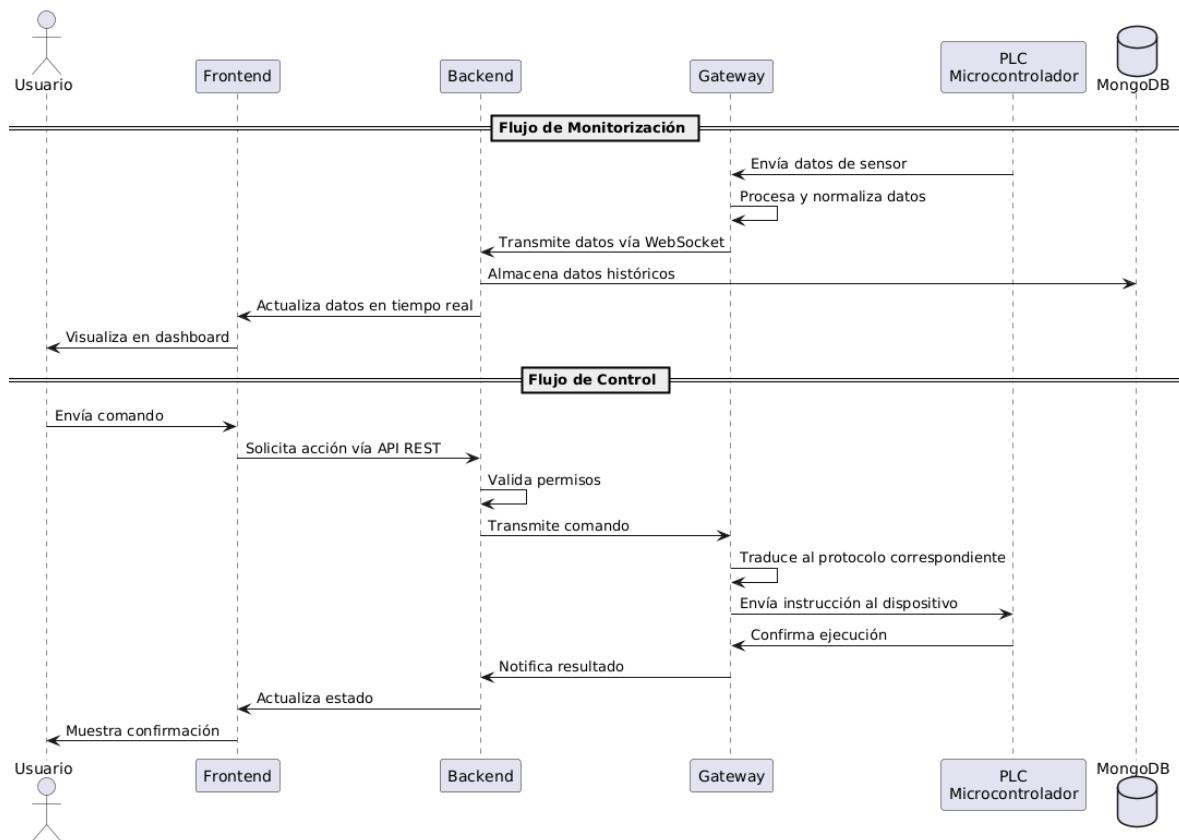


Figura 30 Flujo general de datos del sistema.

Como síntesis de lo expuesto en este capítulo, la figura 30 presenta una visión general del flujo de adquisición de datos, ilustrando las distintas conexiones entre las diferentes capas del sistema.

## 7.8 FUNCIONALIDADES IMPLEMENTADAS

La plataforma SCADA web ofrece las siguientes funcionalidades principales:

1. Monitorización en tiempo real: Visualización de estados de sensores y actuadores mediante gráficos dinámicos y tableros configurables.
2. Control remoto: Interfaz para envío de comandos a las celdas Fischertechnik, permitiendo controlar motores, electroválvulas y otros actuadores.
3. Registro histórico: Almacenamiento de datos para consulta posterior, análisis de tendencias y generación de informes.
4. Gestión de proyectos: Organización de dispositivos y variables por proyecto, con control de acceso basado en roles.

## 8. VALIDACIÓN Y RESULTADOS

Este capítulo presenta la validación del sistema desarrollado, abarcando las celdas Fischertechnik, el controlador lógico programable y el sistema SCADA web. Se verifican los componentes individuales y su funcionamiento integrado, concluyendo con la validación pedagógica de las guías de laboratorio desarrolladas.

### 8.1 VALIDACIÓN DE COMPONENTES FISCHERTECHNIK

Las piezas fabricadas mediante impresión 3D para complementar el kit Fischertechnik fueron evaluadas en su contexto de uso. Las pruebas confirmaron que todas las piezas se integraron correctamente en las celdas, cumpliendo adecuadamente sus funciones estructurales y mecánicas sin presentar problemas de ajuste o estabilidad. Los detalles de esta validación se presentan en el Anexo O: Validación de piezas fischertechnik.

Los principales aspectos validados fueron:

1. Ajuste dimensional: Todas las piezas se ajustaron correctamente a las estructuras existentes, con tolerancias dentro de los 0.2 mm requeridos.
2. Resistencia mecánica: Las piezas soportaron adecuadamente las fuerzas y tensiones aplicadas durante el funcionamiento normal de las celdas, sin presentar deformaciones o fallas.
3. Funcionalidad: Cada pieza cumplió su función específica (estructural, dinámica o de interfaz) de manera satisfactoria, permitiendo el correcto funcionamiento de los mecanismos.

4. Durabilidad: Durante las pruebas de validación del sistema completo, que implicaron múltiples ciclos de operación, las piezas mantuvieron su integridad y precisión.

Este proceso de validación confirma la viabilidad de utilizar impresión 3D como método para fabricar componentes de reemplazo o complementarios para el kit educativo Fischertechnik, ampliando sus posibilidades de uso en entornos académicos.

## 8.2 VALIDACIÓN DEL CONTROLADOR LÓGICO PROGRAMABLE

Para evaluar el desempeño del controlador desarrollado, se utilizó como referencia el análisis del controlador Fischertechnik realizado por David Adams en su libro Fischertechnik® TXT 4.0 Controller Internals and Programming [46]. La siguiente tabla compara los márgenes de error entre el controlador lógico programable desarrollado en el proyecto de grado contra el controlador de Fischertechnik:

Característica	Controlador Fischertechnik (Error)	Controlador Desarrollado (Error)	Resultado
Entradas analógicas	<6.8%	<5% (con compensador)	Comparable
Entradas digitales	50% frecuencias entre 500hz y 1 kHz	<0.05% hasta 9 kHz	Superior
Salidas analógicas / PWM	<5% por debajo del valor nominal	<1% en todo el rango	Superior
Control de motores	Tolerancias de $\pm 11\%$ a $\pm 12.5\%$	Tolerancia de <1%	Superior

Tabla 6 Resultados comparaciones de errores.

Los resultados demuestran que el controlador desarrollado alcanza niveles de precisión comparables o superiores al controlador Fischertechnik de referencia, cumpliendo satisfactoriamente con los requisitos para entornos educativos. Los detalles específicos de cada prueba y validación se presentan en el Anexo P: Validación del controlador lógico programable.

### 8.3 VALIDACIÓN DEL SISTEMA INTEGRADO

Se completó la integración de todos los componentes (celdas Fischertechnik, controlador lógico y tarjeta de adecuación de señales) siguiendo el esquema previamente diseñado. Para mejorar el orden y la operatividad, los cables y mangueras se condujeron por la parte inferior de la estructura, asegurando que no interfirieran con el funcionamiento del sistema.

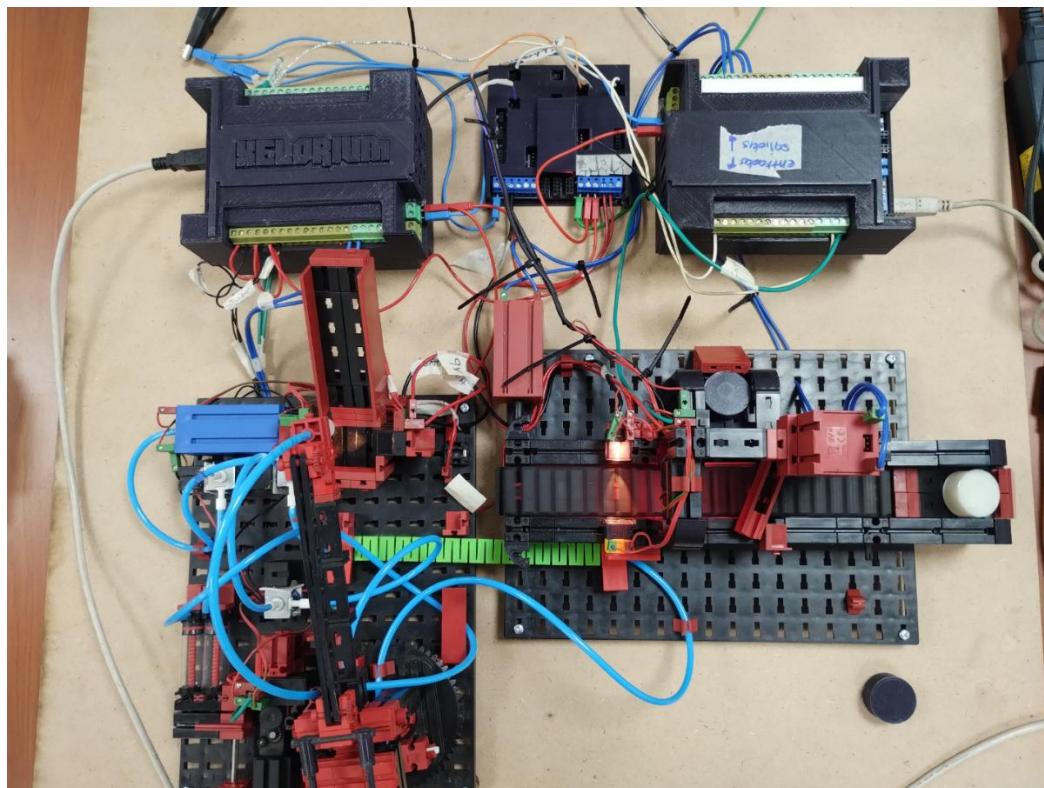


Figura 31 Montaje del sistema completo

El sistema implementa un proceso automatizado que inicia con la detección de una pieza, continúa con su manipulación en la primera celda (brazo neumático) y finaliza con su transporte y clasificación en la segunda celda (banda transportadora). Los datos generados en cada etapa se almacenan mediante variables globales y se visualizan en el sistema SCADA.

El siguiente diagrama representa el proceso completo del sistema, incluyendo la interacción entre ambas celdas y la plataforma SCADA:

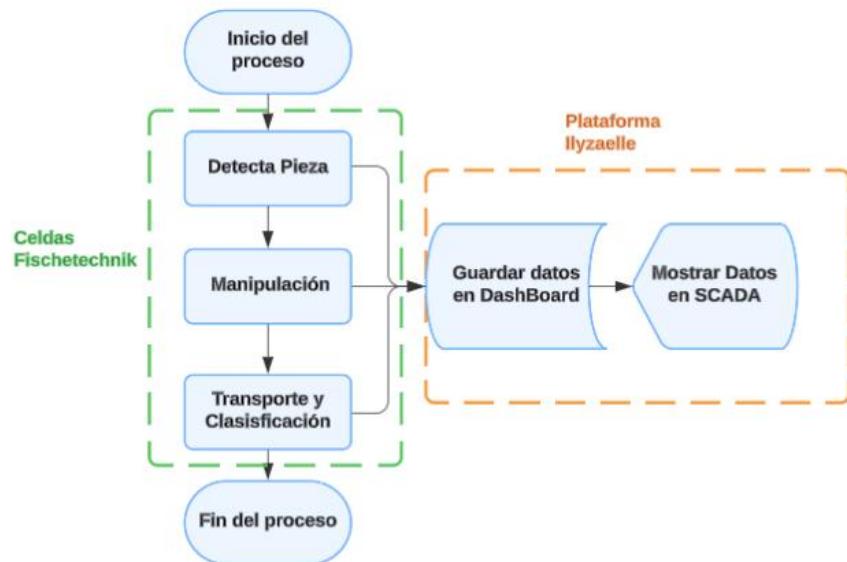


Figura 32 Diagrama de flujo del proceso del sistema.

Este diagrama muestra la secuencia de operaciones desde la detección inicial de una pieza hasta su clasificación final, así como el flujo de datos hacia el sistema SCADA.

Los diagramas de flujo detallados de cada proceso se presentan en el Anexo Q: Validación del sistema integrado.

Para validar el funcionamiento completo del sistema, se realizaron pruebas con 20 piezas (10 blancas y 10 negras) introducidas secuencialmente en el sistema. Los posibles errores se separaron en las categorías mostradas en el Anexo Q, Los resultados de las pruebas con las 20 piezas se presentan en la siguiente tabla:

Nº prueba	Color	Error de programación	Error al detectar pieza	Error al aspirar pieza	Error de movimiento	Error al dejar la pieza	Error en la banda	Error en la barreira	Error en interruptores
1-10	Blanco	No	No	No	No	No	No	No	No
11-20	Negro	No	No	No	No	No	No	No	No

Tabla 7 Resultados pruebas de validación.

No se registraron errores en ninguna de las categorías evaluadas, lo que confirma el correcto funcionamiento del sistema integrado.

Se realizó una prueba funcional utilizando la plataforma de supervisión SCADA, verificando en tiempo real el estado del proceso y la coordinación entre las celdas. La siguiente imagen muestra el sistema SCADA durante la operación:



Figura 33 Pruebas del funcionamiento SCADA y celdas.

La prueba confirmó que el sistema SCADA:

1. Muestra correctamente el estado de las celdas en tiempo real
2. Registra y visualiza adecuadamente el conteo de piezas
3. Permite el control remoto de los actuadores
4. Mantiene sincronizada la información entre el sistema físico y la plataforma web

Esta validación integrada demuestra que el sistema completo funciona correctamente, cumpliendo con los requisitos establecidos para el proyecto.

#### 8.4 VALIDACIÓN DE GUÍAS DE LABORATORIO

Con el propósito de garantizar la pertinencia y la usabilidad de las guías, se empleó un enfoque de laboratorios guiados, entendido como la combinación de una breve instrucción previa a la práctica (donde se exponen objetivos, fundamentos y procedimientos) seguida de la ejecución práctica bajo supervisión. Este modelo de aprendizaje en contextos tecnopedagógicos se respalda en la literatura especializada, donde se destaca que la orientación sistemática en el laboratorio facilita la internalización de habilidades técnicas y conceptuales [47].

El grupo de control estuvo conformado por un 92.3% de estudiantes activos de la UNAB y un 7.7% de participantes externos, realizándose un total de 25 pruebas individuales

Durante la fase de validación, se recopilaron datos de tiempo provenientes de diferentes estudiantes en cada laboratorio. Con base en dichas mediciones, se obtuvo lo siguientes tiempos promedio:

- Tiempo promedio registrado Laboratorio 1: 17.9 min
- Tiempo promedio registrado Laboratorio 2: 47.5 min
- Tiempo promedio registrado Laboratorio 3: 87.3 min

Estos tiempos demuestran que cada práctica puede realizarse en un bloque de clase estándar (45-90 minutos). Las métricas de validación de las encuestas se detallan en el Anexo R: Validación de guías de laboratorio.

La evaluación de las guías arrojó resultados muy positivos en cuatro áreas principales:

1. Uso general de la plataforma: El 100% de participantes logró establecer la conexión Gateway-controlador sin necesidad de ayuda externa, y más del 70% consideró la plataforma "mucho más fácil" que otros sistemas SCADA utilizados previamente.
2. Programación del controlador: El 81.8% se mostró "muy satisfecho" con el editor de código, y el 100% consideró que la eficiencia de lectura/escritura de GPIO era "eficiente" o "muy eficiente".
3. Interacción con celdas Fischertechnik: El 100% de los participantes calificó como "satisfactoria" o "muy satisfactoria" la implementación del contador de piezas y banderas de estado, y más del 90% no experimentó errores durante la ejecución.
4. Valoración general: El 90.9% calificó las guías como "claras" o "muy claras", y el 100% consideró que la práctica resultó motivante para diseñar funcionalidades adicionales y utilizar la plataforma en proyectos futuros.

Los resultados detallados por sección y la síntesis final de la validación se presentan en el Anexo R: Validación de guías de laboratorio.

## 8.5 CONCLUSIONES DE LA VALIDACIÓN

La validación integral del sistema ha demostrado que tanto los componentes individuales como su integración cumplen satisfactoriamente con los requisitos establecidos.

Estos resultados validan el cumplimiento de los objetivos del proyecto, confirmando la viabilidad técnica y pedagógica de la solución desarrollada para entornos educativos de automatización industrial.

## CONCLUSIONES

El desarrollo del proyecto permitió comprobar que la implementación del sistema propuesto logró cumplir los objetivos planteados de manera satisfactoria. La integración de los distintos componentes demostró ser funcional y coherente con las necesidades identificadas al inicio del trabajo. Los resultados alcanzados evidencian que las estrategias metodológicas empleadas facilitaron la construcción de un prototipo que responde a los requerimientos del entorno, aportando una solución innovadora y viable.

Durante el proceso se constató que el modelo diseñado posibilita un aprendizaje práctico y significativo, pues fomenta la interacción directa con la tecnología y fortalece competencias en áreas como la programación, la electrónica y el pensamiento lógico. Asimismo, el sistema desarrollado se convierte en un recurso de apoyo que contribuye al proceso de enseñanza-aprendizaje, ya que permite la experimentación y el análisis de resultados de forma tangible.

El proyecto también permitió verificar la pertinencia del diseño en términos de aplicabilidad, mostrando que es posible adaptar soluciones tecnológicas a contextos específicos con un costo accesible y con resultados eficientes. Esto evidencia que la propuesta tiene un potencial considerable para ser replicada y mejorada en escenarios similares, abriendo oportunidades para futuras investigaciones y aplicaciones prácticas.

Finalmente, se concluye que la experiencia adquirida a lo largo de la investigación favoreció no solo la validación del sistema, sino también el fortalecimiento de habilidades investigativas y de innovación en los participantes, reafirmando la importancia de integrar la tecnología como un medio formativo y transformador en diferentes contextos.

## RECOMENDACIONES

A partir de las limitaciones encontradas en el desarrollo del proyecto, se recomienda continuar con la optimización de los componentes técnicos, especialmente en lo relacionado con la calibración de sensores y la estabilidad del sistema en condiciones de mayor exigencia. Una mejora en estos aspectos incrementaría la precisión y confiabilidad de los resultados obtenidos.

Es aconsejable también ampliar las pruebas en distintos entornos y bajo diferentes condiciones de uso, de manera que se logre una validación más robusta que permita identificar nuevos factores de mejora. De igual manera, resulta pertinente realizar procesos de estandarización que garanticen la reproducibilidad del sistema en diferentes contextos educativos y técnicos.

Se sugiere considerar la incorporación de módulos adicionales que potencien las funcionalidades del prototipo, como la integración de nuevas interfaces de usuario, sistemas de comunicación más eficientes o la conexión con plataformas digitales que favorezcan la recopilación y análisis de datos en tiempo real. Estas mejoras no solo optimizarían el rendimiento, sino que ampliarían el campo de aplicación del sistema.

Finalmente, se recomienda fomentar la capacitación continua de los usuarios y facilitadores en el manejo del sistema, de manera que se asegure su correcto uso y se aproveche al máximo su potencial formativo. El fortalecimiento de estas competencias garantizará que la propuesta no solo se limite a un prototipo funcional, sino que se convierta en una herramienta sostenible y coden impacto a largo plazo.

## BIBLIOGRAFIA

- [1] Fischertechnik, «Apps & Software for the classroom,» Fischertechnik, [En línea]. Available: <https://www.fischertechnik.de/en/apps-and-software>. [Último acceso: 2024].
- [2] G. Garcés y C. Peña, «Ajustar la Educación en Ingeniería a la Industria 4.0: Una visión desde el desarrollo curricular y el laboratorio,» de Vol. 19, REXE, 2020, pp. 129-148.
- [3] Mordor Intelligence, «Análisis de participación y tamaño del mercado de PLC tendencias y pronósticos de crecimiento (2024-2029),» Mordor Intelligence, 2023. [En línea]. Available: <https://www.mordorintelligence.com/es/industry-reports/programmable-logic-controller-plc-market>. [Último acceso: 2024].
- [4] J. Pérez, «La relación del software libre con el hardware libre,» Somos Libres, 2023. [En línea]. Available: <https://somoslibres.org/index.php/bienvenido/101-hardware-libre/12842-la-relacion-del-software-libre-con-el-hardware-libre>. [Último acceso: 2024].
- [5] J. Rivera Vargas, L. M. Paucar Alcantara y A. J. Chafloque Moreno, «Controlador lógico programable de salida tipo relé, basado en Arduino para transmisión de datos entre etapas de procesos industriales,» *Universidad Nacional del Callao*, vol. I, 2019.
- [6] S. Rúnarsson, «Open source hardware and software alternative to industrial PLC,» *øgskolen i Sørøst-Norge*, vol. Master's thesis, 2016.
- [7] M. S. Mohd Samsudin, «Development of cost effective PLC training kit by using arduino platform (PLCDUINO),» *Universiti Tun Hussein Onn Malaysia*, vol. Doctoral dissertation, 2015.

- [8] G. Vieira, J. Barbosa, P. Leitão y L. Sakurada, «Low-cost industrial controller based on the raspberry pi platform,» *IEEE International Conference on Industrial Technology (ICIT)*, vol. I, pp. 292-297, 2020.
- [9] D. O. Kovaliuk, K. M. Huza y O. O. Kovaliuk, «Development of SCADA system based on web technologies,» *International Journal of Information Engineering and Electronic Business*, vol. 10, pp. 25-32, 2018.
- [10] A. A. Díaz, «Adaptación de los laboratorios planta piloto y laboratorio de automatización a la tecnología industria 4.0,» *Universidad Autónoma de Bucaramanga*, vol. I, 2020.
- [11] J. D. Rueda, «Gestión en la nube y modelo de realidad aumentada para el banco de pruebas de múltiples bombas centrífugas en configuración variable de la UNAB,» *Universidad Autónoma de Bucaramanga*, 2021.
- [12] M. G. Zaragoza y H. K. Kim, «Comparative Study of PLC and Arduino in Automated Irrigation System,» *International Journal of Control and Automation*, vol. 10, nº 6, pp. 207-218, 2017.
- [13] D. K. Cabrera, D. G. Garcia, A. B. Quirós, J. L. Ramos y A. M. Laguardia, «Proposal of a programmable logic controller based on open hardware,» *ITEGAM-JETIA*, vol. 9, nº 42, pp. 41-47, 2023.
- [14] A. Palanisamy, «Home automation using PLC and Arduino,» *Acta Mechanica Slovaca*, vol. 24, nº 1, pp. 56-68, 2020.
- [15] M. J. Rosales, «Automatización de la planta de Fischertechnik del laboratorio de automatización y control utilizando un PLC Siemens S7-1500,» *Universidad Industrial de Santander*, vol. 1, 2021.
- [16] W. Bolton, Programmable Logic Controllers, 6th ed., Newnes, 2015.
- [17] J. R. Hackworth y F. D. Hackworth Jr, Programmable Logic Controllers: Programming Methods and Applications, Pearson, 2003.
- [18] T. R. Kuphaldt, Lessons In Industrial Instrumentation, 2nd ed., Creative Commons Attribution, 2021.

- [19] K. H. John y M. Tiegelkamp, IEC 61131-3: Programming Industrial Automation Systems, 2nd ed., Springer, 2010.
- [20] A. Gibb, Building Open Source Hardware: DIY Manufacturing for Hackers and Makers, 1st ed., Addison-Wesley Professional, 2014.
- [21] M. A. Mazidi y S. Naimi, The AVR Microcontroller and Embedded Systems: Using Assembly and C, Pearson, 2010.
- [22] M. Margolis, Arduino Cookbook, 2nd ed., O'Reilly Media, 2010.
- [23] N. Kolban, Kolban's Book on ESP32, Leanpub, 2018.
- [24] E. D. Knapp y J. T. Langill, Industrial Network Security, 2nd ed., Syngress, 2014.
- [25] C. Kozierok, The TCP/IP Guide, Starch Press, 2005.
- [26] G. Clarke, D. Reynders y E. Wright, Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems, Newnes, 2004.
- [27] B. A. Forouzan, Data Communications and Networking, 5th ed., McGraw-Hill Education, 2017.
- [28] S. A. Boyer, SCADA: Supervisory Control and Data Acquisition, 4th ed., ISA, 2009.
- [29] D. Bailey y E. Wright, Practical SCADA for Industry, Newnes, 2003.
- [30] J. Duckett, HTML and CSS: Design and Build Websites, John Wiley & Sons, 2011.
- [31] A. Banks y E. Porcello, Learning React: Functional Web Development with React and Redux, O'Reilly Media, 2017.
- [32] B. Green y S. Seshadri, AngularJS, O'Reilly Media, 2013.
- [33] M. Richards, Software Architecture Patterns, O'Reilly Media, 2015.
- [34] T. Santana, Mastering Node.js, Packt Publishing, 2018.
- [35] M. Lutz, Programming Python, 4th ed., O'Reilly Media, 2010.
- [36] P. DuBois, MySQL Cookbook, 3rd ed., O'Reilly Media, 2014.

- [37] K. Chodorow, MongoDB: The Definitive Guide, 3rd ed., O'Reilly Media, 2019.
- [38] D. Brooks, Circuit Board Design for Manufacture, 2005.
- [39] D. L. Wingard, PCB Fabrication: Technology and Processes, O'Reilly Media, 2002.
- [40] O'Reilly Media, Understanding Open Source and Free Software Licensing, 1st ed., O'Reilly Media, 2004.
- [41] L. Rosen, Open Source Licensing: Software Freedom and Intellectual Property Law, O'Reilly Media, 2004.
- [42] J. Smith, Understanding MIT and BSD Open Source Licenses, O'Reilly Media, 2015.
- [43] JLCPCB, «Assembly Parts Lib For PCBA Only,» JLCPCB, 2024. [En línea]. Available: <https://jlpcb.com/parts>. [Último acceso: 2024].
- [44] JLCPCB, «PCB Manufacturing & Assembly Capabilities,» JLCPCB, 2024. [En línea]. Available: <https://jlpcb.com/capabilities/pcb-capabilities>. [Último acceso: 2024].
- [45] Fischertechnik, «ft Design Studio,» Fischertechnik, 2025. [En línea]. Available: <https://www.fischertechnik.de/en/ft-design-studio>. [Último acceso: 2025].
- [46] D. Adams, «Fischertechnik® TXT 4.0 Controller Internals and Programming,» *Tarma Software Research Ltd*, vol. I, p. 804, 2024.
- [47] S. K. Abell y N. G. Lederman, «Handbook of Research on Science Education,» *Lawrence Erlbaum Associates*, 2007.
- [48] Github blog, «Octoverse: The state of open source and rise of AI in 2023,» Github, 2023. [En línea]. [Último acceso: 2024].
- [49] B. Harkirat , K. TaranPreet y R. Yash , «The Better Comparison between PHP, Python-web & Node.js,» *International Journal of Research in Engineering and Science (IJRES)*, vol. I, p. 9, 2021.

## ANEXOS

### ANEXO A: RECOPILACIÓN INFORMACIÓN CONTROLADORES COMERCIALES

Parámetro	Robo TX	M-DUINO	IONO UNO	OPTA
Fabricante	Fischertechnik (Alemania)	Industrial Shields (España)	Sfera Labs	Arduino
Alimentación	9V DC	12-24V DC	24V DC	12-24V DC
Certificaciones	CE, FCC	CE, FCC, UKCA	CE	IP20, CE, FCC, RoHS
Formato	Monolítico	Modular	Modular	Monolítico
Aplicación Principal	Educativo	Industrial	Industrial	Educativo
Precio (USD)	200-300	250-350	135	146-193

Tabla 8 Comparación de especificaciones técnicas.

Parámetro	Robo TX	M-DUINO	IONO UNO	OPTA
Procesador	ARM9 @ 200 MHz	ATmega2560 @ 16 MHz	ATmega328p @ 16 MHz	Dual-core Cortex- M7+M4
Memoria	8 MB RAM, 2 MB Flash	8 KB SRAM, 256 KB Flash	2 KB SRAM, 32 KB Flash	1 MB RAM, 2 MB Flash
Velocidad de procesamiento	Alta	Media	Baja	Muy Alta

Tabla 9 Capacidad de procesamiento.

Parámetro	Robo TX	M-DUINO	IONO UNO	OPTA
Entradas	8 digital/análogo + 4 digitales	13 digitales + 7 análogas	6 digital/análogo	8 digital/análogo
Salidas	4 PWM motores DC	15 PWM + 4 relés	6 relés	4 relés

Tabla 10 Capacidad de entradas/salidas.

Parámetro	Robo TX	M-DUINO	IONO UNO	OPTA
Conectividad	Bluetooth, USB	Ethernet	Ethernet	Wifi, Bluetooth, Ethernet
Protocolos	RS-485, I <sup>2</sup> C	RS-485, TCP/IP, I <sup>2</sup> C, SPI	RS-485	RS-485, TCP/IP
IoT Ready	NO	PARCIAL	PARCIAL	SI

Tabla 11 Capacidad de conectividad y comunicaciones.

## ANEXO B: RECOPILACIÓN INFORMACIÓN PLATAFORMAS IOT

Parámetro	UBIDOTS	ARDUINO CLOUD
Enfoque Principal	Industrial	Educativo y desarrollo
Visualización	Dashboards avanzados	Dasboards básicos
Alertas	Email, SMS, WhatsApp, Telegram	Básicas en la plataforma
Protocolos	HTTP, MQTT, +60 integraciones	HTTP Web API
Dispositivos Compatibles	PLCs Modbus, gateways industriales, esp32 +20 dispositivos	Arduino y esp32
Escalabilidad	Alta	Limitada
Programación Remota	NO	SI
Curva de Aprendizaje	Alta	Baja
Precio (USD)	99-1200/mes	7-100/mes

Tabla 12 Comparación de especificaciones técnicas.

ANEXO C: ESPECIFICACIONES TECNICAS CELDAS FISCHERTECHNIK

I/O CELDA BRAZO NEUMÁTICO FISCHERTECHNIK			
ENTRADAS		SALIDAS	
DESCRIPCIÓN	TIPO	DESCRIPCIÓN	TIPO
2x Switch	Entrada digital (IX)	3x Electroválvula	Salida digital (QX)
1x Fototransistor	Entrada digital (IX)	1x Motor	Salida análoga (QW)
1x Sensor color	Entrada análoga (IW)		
COMBINACIÓN	3IX - 1IW - 3QX - 1QW		

Tabla 13 I/O Celda Brazo neumático FISCHERTECHNIK.

ESPECIFICACIONES TÉCNICAS CELDA BRAZO NEUMÁTICO FISCHERTECHNIK			
DESCRIPCIÓN	TIPO	ESPECIFICACIONES	Pines
Fototransistor	Entrada	Voltaje colector-emisor Vce:35V Corriente collector Ic: 15mA	1 salida, que proporciona una señal binaria (encendido/apagado)
Switch	Entrada	Voltaje máximo: 50V Corriente máxima: 2A	1 salida, que es la señal binaria (encendido/apagado)
Sensor de Color	Entrada	Voltaje: 6-10V Corriente: 15mA Valor de rango: 0-2000 mV	1 salida, proporciona una señal

Electrovalvulas	Salida	Tipo de válvula: 3/2 vías Voltaje 9V Corriente: 133 mA	Terminales para la conexión de alimentación.
Mini Motor Negro	Salida	Voltaje 6-9V Corriente máxima: 0.65A	Terminales para la conexión de alimentación.
Led	Externo	Voltaje 9V	Terminales para la conexión de alimentación.
Compresor	Externo	Voltaje 9v Corriente 200mA	Terminales para la conexión de alimentación.

Tabla 14 Especificaciones I/O Celda Brazo neumático FISCHERTECHNIK.

I/O CELDA BANDA TRANSPORTADORA FISCHERTECHNIK			
ENTRADAS		SALIDAS	
DESCRIPCIÓN	TIPO	DESCRIPCIÓN	TIPO
1x Fototransistor	Entrada digital (IX)	1x Motor selector	Salida analógica (QW)
1x Switch	Entrada digital (IX)	1x Motor banda	Salida analógica (QW)
COMBINACIÓN		2IX - 2QW	

Tabla 15 I/O Celda Banda Transportadora FISCHERTECHNIK.

ESPECIFICACIONES TÉCNICAS CELDA BANDA TRANSPORTADORA FISCHERTECHNIK			
DESCRIPCIÓN	TIPO	ESPECIFICACIONES	Pines
Fototransistor	Entrada	Voltaje colector-emisor Vce:35V Corriente collector Ic: 15mA	1 salida, que proporciona una señal binaria (encendido/apagado)

Switch	Entrada	Voltaje máximo: 50V Corriente máxima: 2A	1 salida, que es la señal binaria (encendido/apagado)
Motor DC Rojo	Salida	Voltaje: 9V DC Corriente nominal: 0.35A max	Cuenta con 2 pines los cuales son para la alimentación de 9V. Además, cuenta con un encoder: Un conector de 3 pines para el cable del encoder, que transmite las señales de posición.
Led	Externo	Voltaje 9V	Terminales para la conexión de alimentación.

Tabla 16 Especificaciones I/O Celda Banda Transportadora FISCHERTECHNIK.

## ANEXO D: SELECCIÓN DE COMPONENTES ELECTRONICOS

### D.1 SELECCIÓN DE MICROCONTROLADOR

Requerimiento a satisfacer: El controlador debe igualar o superar la capacidad de E/S del Robo TX Controller y ser compatible con Arduino IDE.

CRITERIOS	AT91SAM9260	ATmega328p	ATmega2560	ESP32
Pines GPIO (mínimo 20)	1	0	1	1
WiFi/Ethernet	0	-1	-1	1
Memoria Flash	1	-1	1	1
Memoria SRAM	0	-1	0	1
Frecuencia de Reloj	1	0	0	1
Compatibilidad voltaje	1	1	1	-1
Precio	-1	1	1	0
Dificultad implementación	-1	1	1	-1
TOTAL	2	0	4	<b>3</b>

Tabla 17 Selección de microcontrolador [43]

Pines GPIO:

- +1: >50 pines, suficientes para todas las E/S requeridas
- 0: 20-30 pines, cubre las necesidades básicas
- -1: <20 pines, insuficiente para los requerimientos

WiFi/Ethernet:

- +1: Integrado directamente en el chip

- 0: Disponible mediante conexión con módulos dedicados
- -1: Requiere módulos adicionales complejos

Memoria Flash:

- +1: >256KB, suficiente para aplicaciones complejas
- 0: 128-256KB, adecuada para aplicaciones de complejidad media
- -1: <128KB, limitada para aplicaciones industriales

Memoria SRAM:

- +1: >64KB, excelente para manejo de datos
- 0: 8-64KB, adecuada para aplicaciones comunes
- -1: <8KB, limitada para procesamiento de variables

Frecuencia de Reloj:

- +1: >100MHz, alto rendimiento
- 0: 16-20MHz, rendimiento estándar para controladores
- -1: <16MHz, rendimiento limitado

Compatibilidad voltaje:

- +1: Compatible con 5V, estándar en muchos sensores
- 0: Requiere adaptación simple
- -1: Requiere adaptación compleja

Precio:

- +1: <\$20 USD, económico
- 0: \$20-\$50 USD, precio medio
- -1: >\$50 USD, costoso

Dificultad implementación:

- +1: Baja dificultad, amplia documentación

- 0: Dificultad media, documentación suficiente
- -1: Alta dificultad, poca documentación

Se selecciona el microcontrolador ATmega2560 por su equilibrio entre capacidad y facilidad de implementación.

Especificaciones técnicas: 54 pines digitales I/O, 256KB de memoria flash, 8KB SRAM, velocidad de 16MHz y voltaje operativo de 5V. El microcontrolador se implementa mediante una placa Arduino MEGA, garantizando compatibilidad con Arduino IDE como se requiere en las especificaciones [43].

## D.2 SELECCIÓN MÓDULO DE COMUNICACIÓN

Requerimiento a satisfacer: El controlador debe tener conectividad USB y WiFi, soportando protocolo Serial y Virtual Serial.

CRITERIO	Ethernet W5500	ESP8266
Velocidad de comunicación	1	1
Costo	-1	1
Consumo de energía	-1	0
Facilidad de configuración	-1	1
Alcance/Distancia	1	0
Confiabilidad	1	0
Dificultad de implementación	-1	1
TOTAL	-1	4

Tabla 18 Selección módulo de comunicación.

Velocidad de comunicación:

- +1: >50 Mbps, excelente para transmisión de datos

- 0: 10-50 Mbps, adecuada para aplicaciones estándar
- -1: <10 Mbps, limitada para aplicaciones intensivas

Costo:

- +1: <\$10 USD, económico
- 0: \$10-\$20 USD, precio medio
- -1: >\$20 USD, costoso

Consumo de energía:

- +1: <100 mA, bajo consumo
- 0: 100-300 mA, consumo moderado
- -1: >300 mA, alto consumo

Facilidad de configuración:

- +1: Configuración plug & play, sin ajustes adicionales
- 0: Configuración moderada, pocos ajustes
- -1: Configuración compleja, múltiples ajustes

Alcance/Distancia:

- +1: >50m o sin limitación por cable
- 0: 10-50m de alcance
- -1: <10m de alcance

Confiabilidad:

- +1: Muy confiable, baja tasa de errores
- 0: Confiabilidad media, algunos errores ocasionales
- -1: Baja confiabilidad, errores frecuentes

Dificultad de implementación:

- +1: Baja dificultad, librerías completas disponibles

- 0: Dificultad media, implementación moderada
- -1: Alta dificultad, desarrollo Complejo

Se implementa el módulo ESP8266 como una solución que cumple con el requerimiento de conectividad.

Especificaciones técnicas: Velocidad de transmisión hasta 72Mbps, tensión de operación 3.3V (requiere adaptación desde 5V), formato SMD, consumo en transmisión ~170mA. Se comunica con el controlador principal por medio de TX/RX [43].

### D.3 SELECCIÓN ENTRADAS DIGITALES

Requerimiento a satisfacer: El controlador debe tener entradas digitales compatibles con señales de 24V y contar con protección eléctrica.

CRITERIOS	EL817S1	4N35	PC817
Aislamiento	1	1	1
Corriente de salida	1	1	1
Tiempo de respuesta	1	0	-1
Encapsulado	1	-1	-1
Disponibilidad	1	-1	1
TOTAL	5	0	1

Tabla 19 Selección entradas digitales – Optoacoplador.

Aislamiento:

- +1: >4 kV, excelente aislamiento
- 0: 2-4 kV, aislamiento adecuado
- -1: <2 kV, aislamiento insuficiente

Corriente de salida:

- +1: >40 mA, suficiente para activar etapas posteriores
- 0: 20-40 mA, adecuada para algunas aplicaciones
- -1: <20 mA, limitada para circuitos digitales

Tiempo de respuesta:

- +1: <5  $\mu$ s, respuesta rápida
- 0: 5-10  $\mu$ s, respuesta media
- -1: >10  $\mu$ s, respuesta lenta

Encapsulado:

- +1: SMD, ideal para montaje automatizado
- 0: DIP pequeño, adecuado para montaje manual o automatizado
- -1: DIP grande, menos eficiente para montaje

Disponibilidad:

- +1: Disponible en el proveedor seleccionado
- 0: Disponible con tiempos de entrega extendidos
- -1: No disponible fácilmente

Se selecciona el optoacoplador EL817S1 por su popularidad y amplia documentación.

Especificaciones técnicas: Aislamiento de 5kV, corriente de salida de 50mA, tiempo de respuesta de 4 $\mu$ s, encapsulado SMD-4P [43].

CRITERIOS	74HC14	CD40106B	TC74HC14AF
-----------	--------	----------	------------

Tensión de operación	0	1	0
Corriente de salida	0	1	0
Tiempo de respuesta	1	0	1
Encapsulado	1	0	1
Disponibilidad	1	1	-1
<b>TOTAL</b>	<b>3</b>	<b>3</b>	<b>1</b>

Tabla 20 Selección entradas digitales – Inversor.

Tensión de operación:

- +1: >6V, amplio rango de operación
- 0: 2-6V, rango estándar
- -1: <2V, rango limitado

Corriente de salida:

- +1: >8 mA, alta capacidad de conducción
- 0: 4-8 mA, capacidad media
- -1: <4 mA, baja capacidad

Tiempo de respuesta:

- +1: <20 ns, muy rápido
- 0: 20-50 ns, velocidad media
- -1: >50 ns, lento

Encapsulado:

- +1: Disponible en múltiples formatos
- 0: Un solo formato estándar
- -1: Formato especializado o difícil de soldar

Disponibilidad:

- +1: Disponible en el proveedor seleccionado

- 0: Disponible con tiempos de entrega extendidos
- -1: No disponible fácilmente

Se selecciona el inversor Schmitt 74HC14 como elemento para invertir la señal proveniente del optoacoplador.

Especificaciones técnicas: Tensión de operación 2-6V, corriente de salida 6mA, tiempo de respuesta 15ns, disponible en encapsulados DIP-6 y SO-8 [43].

#### D.4 SELECCIÓN ENTRADAS ANALÓGICAS

Requerimiento a satisfacer: El controlador debe tener entradas analógicas compatibles con señales de 0-10V.

CRITERIOS	L358P	LM324	TL082
Tensión de alimentación	1	1	0
Slew Rate	0	0	1
Rechazo al ruido	0	0	1
Encapsulado	1	1	0
Disponibilidad	1	0	0
TOTAL	3	2	2

Tabla 21 Selección entradas analógicas.

Tensión de alimentación:

- +1: >12V, compatible con fuentes industriales
- 0: 5-12V, requiere adaptación menor
- -1: <5V, requiere adaptación significativa

Slew Rate:

- +1: >10 V/ $\mu$ s, excelente respuesta dinámica
- 0: 0.5-10 V/ $\mu$ s, respuesta adecuada
- -1: <0.5 V/ $\mu$ s, respuesta lenta

Rechazo al ruido:

- +1: >90 dB, excelente inmunidad
- 0: 70-90 dB, inmunidad adecuada
- -1: <70 dB, susceptible al ruido

Encapsulado:

- +1: Disponible en formato DIP y SMD
- 0: Disponible en un solo formato estándar
- -1: Formato especializado

Disponibilidad:

- +1: Disponible en el proveedor seleccionado
- 0: Disponible con tiempos de entrega extendidos
- -1: No disponible fácilmente

Se selecciona el amplificador operacional L358P debido a su amplia popularidad y facilidad de uso.

Especificaciones técnicas: Alimentación de 3-32V, doble amplificador operacional, encapsulado DIP-8/SOIC-8, bajo consumo de corriente de 0.7mA típico, slew rate de 0.6 V/ $\mu$ s [43].

## D.5 SELECCIÓN SALIDAS MOTOR PWM

Requerimiento a satisfacer: El controlador debe tener salidas PWM de 0-10V para el control de motores.

CRITERIOS	L298N	L293D	TB6612FNG
Tensión de operación	1	1	0
Corriente por canal	1	0	0
Necesidad de disipador	-1	1	1
Tamaño	-1	0	1
Facilidad de uso	0	1	0
Disponibilidad	0	1	-1
TOTAL	0	4	1

Tabla 22 Selección salidas motor PWM.

Tensión de operación:

- +1: >12V, compatible con niveles industriales
- 0: 5-12V, requiere adaptación menor
- -1: <5V, requiere adaptación significativa

Corriente por canal:

- +1: >1A, capacidad para motores grandes
- 0: 0.5-1A, adecuada para motores medianos
- -1: <0.5A, limitada a motores pequeños

Necesidad de disipador:

- +1: No requiere disipador para la operación normal
- 0: Requiere disipador pequeño
- -1: Requiere disipador grande

Tamaño:

- +1: Pequeño (<2cm<sup>2</sup>), eficiente en espacio

- 0: Mediano ( $2-5\text{cm}^2$ ), tamaño razonable
- -1: Grande ( $>5\text{cm}^2$ ), ocupa mucho espacio

Facilidad de uso:

- +1: Fácil implementación, pocas conexiones externas
- 0: Implementación moderada, algunas conexiones adicionales
- -1: Implementación compleja, muchas conexiones adicionales

Disponibilidad:

- +1: Disponible en el proveedor seleccionado
- 0: Disponible con tiempos de entrega extendidos
- -1: No disponible fácilmente

Se selecciona el driver L293D debido a su tamaño y capacidades similares al L298N.

Especificaciones técnicas: Control de hasta 4 motores DC (2 bidireccionales), corriente máxima de 600mA por canal, tensiones de operación de 4.5-36V, encapsulado DIP-16 [43].

## D.6 SELECCIÓN SALIDAS ANALÓGICAS

Requerimiento a satisfacer: El controlador debe tener salidas analógicas de 0-10V.

CRITERIOS	TIP31C	2N2222	BC337
Tensión colector-emisor	0	0	1
Corriente de colector	1	0	0
Tensión base-emisor	0	1	1

Encapsulado	1	-1	-1
Disponibilidad	1	-1	1
TOTAL	3	-1	2

Tabla 23 Selección salidas analógicas.

Tensión colector-emisor:

- +1: >40V, mayor margen de seguridad
- 0: 30-40V, margen adecuado
- -1: <30V, margen limitado

Corriente de colector:

- +1: >1A, capacidad para cargas grandes
- 0: 0.5-1A, adecuada para cargas medianas
- -1: <0.5A, limitada a cargas pequeñas

Tensión base-emisor:

- +1: <0.8V, fácil de activar
- 0: 0.8-1.2V, activación estándar
- -1: >1.2V, requiere mayor tensión de control

Encapsulado:

- +1: TO-220, excelente disipación térmica
- 0: TO-126, disipación media
- -1: TO-92 o SMD, disipación limitada

Disponibilidad:

- +1: Disponible en el proveedor seleccionado
- 0: Disponible con tiempos de entrega extendidos
- -1: No disponible fácilmente

Se selecciona el transistor TIP31C debido a su fiabilidad y facilidad de uso.

Especificaciones técnicas: Vceo de 40V, Ic máxima de 3A, Vbe de 1.1V, encapsulado TO-220 que permite adecuada disipación térmica [43].

## D.7 SELECCIÓN SALIDAS RELÉ

Requerimiento a satisfacer: El controlador debe tener salidas de relé para manejo de cargas de alta potencia.

CRITERIOS	G5Q-1A-DC24	JQC-3FF-S-Z	SRD-24VDC-SL-C
Tensión de operación	1	0	1
Corriente de carga	1	1	1
Tiempo de respuesta	1	1	1
Aislamiento	1	1	1
Tamaño	1	0	0
Disponibilidad	1	-1	1
TOTAL	6	2	5

Tabla 24 Selección salidas relé.

Tensión de operación:

- +1: 24V, compatible con nivel industrial
- 0: Amplio rango incluyendo 24V
- -1: No compatible con 24V

Corriente de carga:

- +1: >8A, capacidad alta
- 0: 5-8A, capacidad media
- -1: <5A, capacidad limitada

Tiempo de respuesta:

- +1: <10ms, respuesta rápida
- 0: 10-20ms, respuesta media
- -1: >20ms, respuesta lenta

Aislamiento:

- +1: >500V, alto aislamiento
- 0: 300-500V, aislamiento medio
- -1: <300V, aislamiento limitado

Tamaño:

- +1: Pequeño (<2cm<sup>2</sup>)
- 0: Mediano (2-4cm<sup>2</sup>)
- -1: Grande (>4cm<sup>2</sup>)

Disponibilidad:

- +1: Disponible en el proveedor seleccionado
- 0: Disponible con tiempos de entrega extendidos
- -1: No disponible fácilmente

Se selecciona el relé G5Q-1A-DC24 debido a su amplia aplicación en PLC comerciales de marcas reconocidas.

Especificaciones técnicas: Tensión de bobina 24VDC, capacidad de conmutación 10A a 250VAC/30VDC, aislamiento de 1000V, tiempo de respuesta 5ms, formato compacto [43].

## D.8 SELECCIÓN FUENTE DE VOLTAJE

Requerimiento a satisfacer: El controlador debe operar con alimentación DC 24V y contar con protección contra inversión de polaridad y sobrecorriente.

Tras el análisis de los requerimientos de voltaje de los componentes seleccionados, se identificó la necesidad de los siguientes niveles:

- 24V (nivel lógico industrial, entrada principal)
- 12V (alimentación Arduino)
- 10V (control de motores)
- 5V (circuitos digitales)

CRITERIOS	MC34063AD	LM2596	LM2940
Flexibilidad de salida	1	1	-1
Corriente de salida	0	1	1
Rango de entrada	1	1	0
Facilidad de uso	1	0	1
Eficiencia	1	0	-1
Tamaño	1	0	0
Disponibilidad	1	1	-1
<b>TOTAL</b>	<b>6</b>	<b>4</b>	<b>-1</b>

Tabla 25 Selección fuente de voltaje.

Flexibilidad de salida:

- +1: Salida ajustable en amplio rango
- 0: Salida ajustable en rango limitado
- -1: Salida fija

Corriente de salida:

- +1: >2A, alta capacidad
- 0: 1-2A, capacidad media
- -1: <1A, capacidad limitada

Rango de entrada:

- +1: >30V de entrada máxima
- 0: 20-30V de entrada máxima
- -1: <20V de entrada máxima

Facilidad de uso:

- +1: Pocos componentes externos
- 0: Cantidad moderada de componentes externos
- -1: Muchos componentes externos

Eficiencia:

- +1: >80%, alta eficiencia
- 0: 70-80%, eficiencia media
- -1: <70%, baja eficiencia

Tamaño:

- +1: Pequeño (<1cm<sup>2</sup>)
- 0: Mediano (1-3cm<sup>2</sup>)
- -1: Grande (>3cm<sup>2</sup>)

Disponibilidad:

- +1: Disponible en el proveedor seleccionado
- 0: Disponible con tiempos de entrega extendidos
- -1: No disponible fácilmente

Se selecciona el módulo LM2596 para generar 12V y 10V.

Especificaciones técnicas: Entrada 7-40V, salida ajustable 1.25-37V, corriente máxima 3A, eficiencia del 75% [43].

Para la regulación a 5V se utiliza el LM7805.

Especificaciones técnicas: Entrada máxima 35V, salida fija 5V  $\pm 4\%$ , corriente máxima 1A, protección térmica integrada, encapsulado TO-220 [43].

## ANEXO E: REGLAS DE DISEÑO APLICADAS

Durante la fase de diseño en KiCad se definieron reglas de diseño (DRC) que integran las restricciones específicas de JLCPCB junto con las recomendaciones de la normativa IPC. Las directrices principales son las siguientes:

Reglas de clearance y anchura de pista (basadas en diseño interno):

- Clearance mínimo configurado en KiCad: 0,127 mm.
- Anchura mínima de pista: 0,127 mm en todo el PCB, aunque el fabricante admite hasta 0,10 mm.
- Espacio vía-pista o vía-vía:  $\geq 0,25$  mm.
- Espacio pad (THT) a pad (THT):  $\geq 0,45$  mm, conforme a recomendaciones de anillo anular para orificios chapados.

Dimensionado de pads y anillos anulares:

- Pads para componentes SMD pequeños): se definieron pads con dimensiones mínimas de  $0,3 \times 0,3$  mm, manteniendo un clearance de 0,15 mm entre pads contiguos (para redes distintas).
- Anillo anular en PTH: mínimo de 0,25 mm para cumplir la norma IPC y las tolerancias de taladro ( $\pm 0,08$  mm).

Dimensionado y ubicación de vías:

- Diámetro de taladro en diseño: 0,3 mm, con pad drill de 0,6 mm.

- Clearance de vía a plano de cobre: 0,3 mm en capas internas y plan para minimizar riesgo de cortocircuito.

Separación entre silkscreen y pads:

- Gap entre texto o marcadores y almohadillas:  $\geq 0,15$  mm.
- Ancho de línea de silkscreen: 0,2 mm (superior al mínimo de 0,153 mm) para asegurar legibilidad en tinta blanca sobre máscara negra.

Distancia de cobre al borde del PCB:

- Margen mínimo: 0,3 mm (superior al mínimo requerido de 0,2 mm) para evitar exposiciones de cobre en contornos mecanizados.

Dimensionado de orificios no chapados (NPTH) para sujeción mecánica:

- Taladro para tornillos de montaje M3: 2,2 mm, con pad NPTH de 2,5 mm. Se verificó que la tolerancia de  $\pm 0,2$  mm en agujeros NPTH no impida el paso del tornillo ni genere interferencias con componentes cercanos.

Espaciado de grupos de pistas (misma red) para trazas de corriente fuerte:

- Señales de potencia (24 V, 12 V, 10 V, 5 V, GND): se agruparon con espaciado de 0,5 mm entre pistas paralelas, permitiendo disipación térmica.
- Red de tierra: se diseñó con relleno (flood) conectado a una vía cada 10 mm para minimizar inductancia.

## ANEXO F: FABRICACIÓN PCB SHIELD ADAPTADOR

Aplicando las mismas reglas de fabricación descritas en el Anexo E: Reglas de diseño aplicadas para el controlador lógico programable, se obtiene el diseño final del módulo shield adaptador que establece la comunicación entre el Módulo I/O con los Módulos CPU y WiFi.

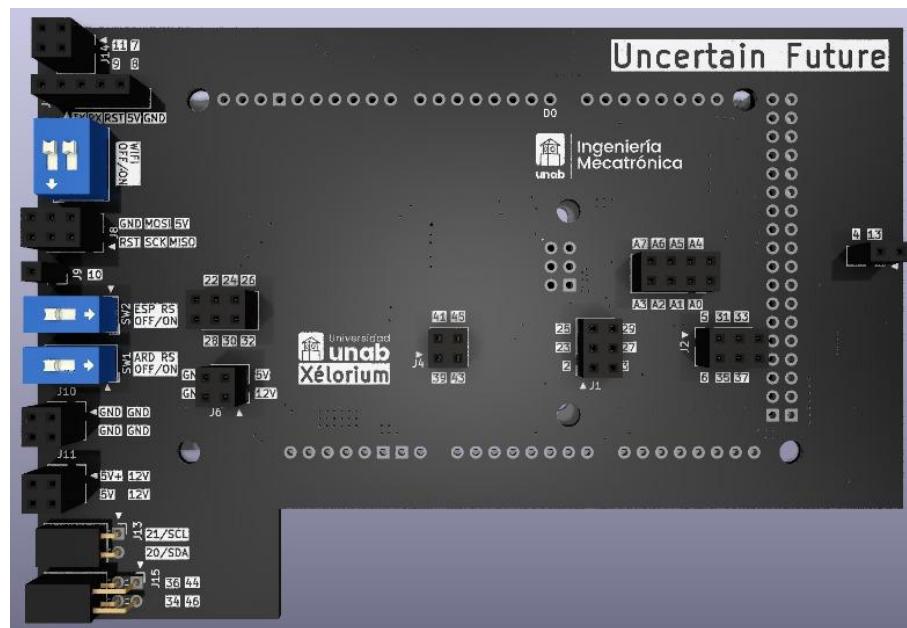


Figura 34 Modelo 3D capa superior módulo shield adaptador.

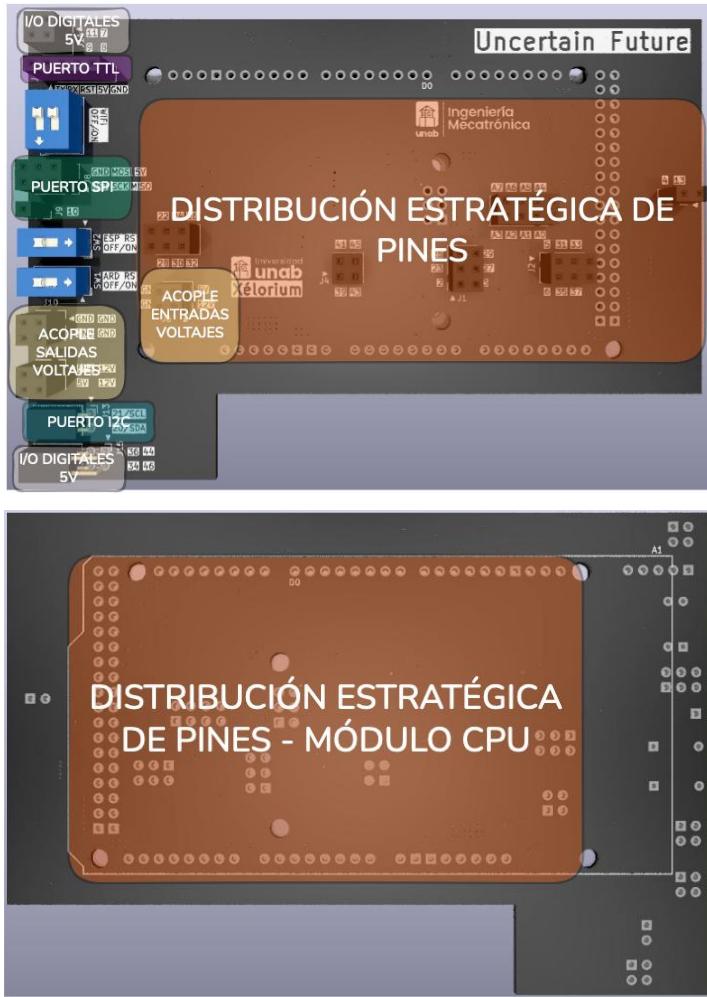


Figura 35 Componentes módulo shield adaptador.

## ANEXO G: DISEÑO Y FABRICACIÓN DE LA CARCASA

### G.1 PROCESO DE DISEÑO CAD

El proceso de diseño de la carcasa del controlador siguió un enfoque sistemático que comenzó con la medición precisa de todos los componentes electrónicos y la definición de sus requisitos de posicionamiento. Para este proceso se utilizó el software SolidWorks, que permitió desarrollar un modelo 3D completo.

El diseño incorporó las siguientes características clave:

- Estructura modular de fácil ensamblaje/desensamblaje
- Aberturas precisas para conexiones de entrada/salida
- Compuertas frontales para acceso rápido a los terminales
- Sistema de ventilación pasiva
- Puntos de montaje para fijación a superficies

### G.2 PIEZAS DE LA CARCASA

La carcasa se compone de múltiples piezas diseñadas para ensamblarse sin necesidad de adhesivos permanentes, únicamente 3 tornillos de sujeción, facilitando futuras modificaciones o reparaciones:

Pieza Solidworks	Pieza Fabricada	Función
------------------	-----------------	---------

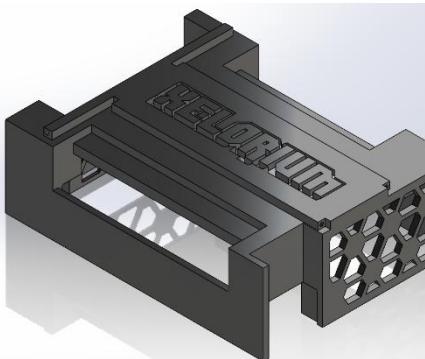
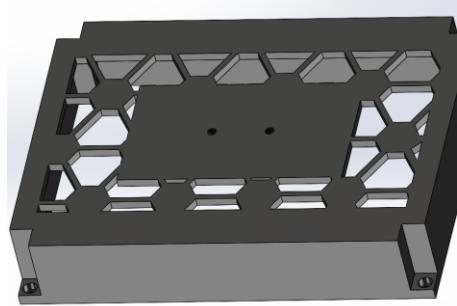
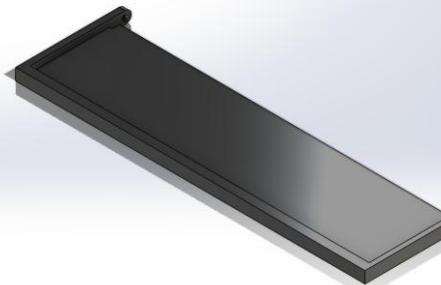
		Base principal que sostiene los circuitos y proporciona rigidez estructural
		Cubierta superior con rejillas de ventilación
		Panel con aberturas para conectores

Tabla 26 Elaboración de piezas

### G.3 PROCESO DE FABRICACIÓN 3D

La impresión 3D de las piezas se realizó con los siguientes parámetros:

- Material: PLA (Ácido Poliláctico)
  - Temperatura de extrusión: 210°C
  - Temperatura de cama: 60°C
  - Altura de capa: 0.2 mm
  - Relleno: 60% en patrón de panal
  - Velocidad de impresión: 60 mm/s
  - Soportes: Generados automáticamente para voladizos > 45°

El proceso de post-procesamiento incluyó:

1. Eliminación de soportes
  2. Lijado ligero de superficies visibles
  3. Verificación de ajustes y tolerancias
  4. Ensamblaje con tornillos M3

## G.4 PLANOS

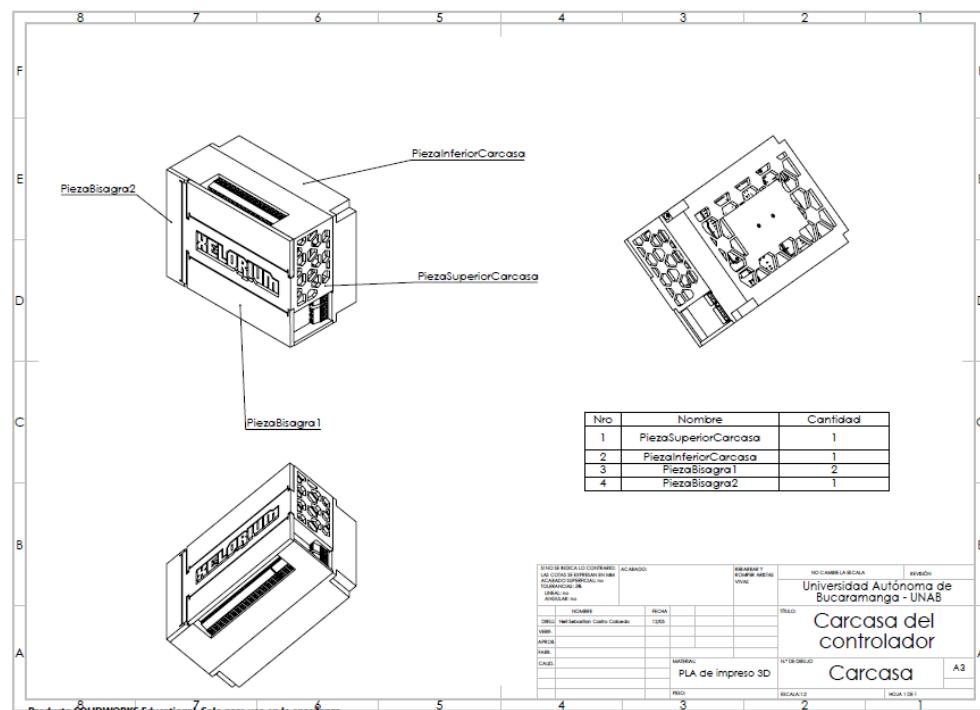


Figura 36 Plano general.

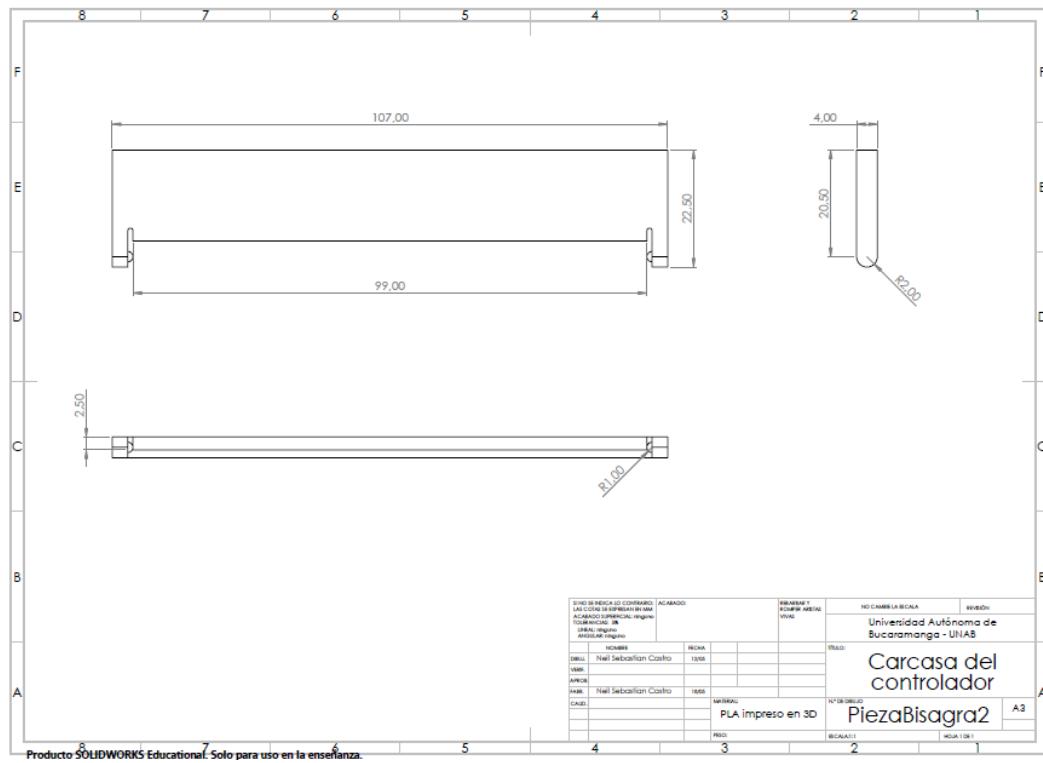


Figura 37 Plano bisagra.

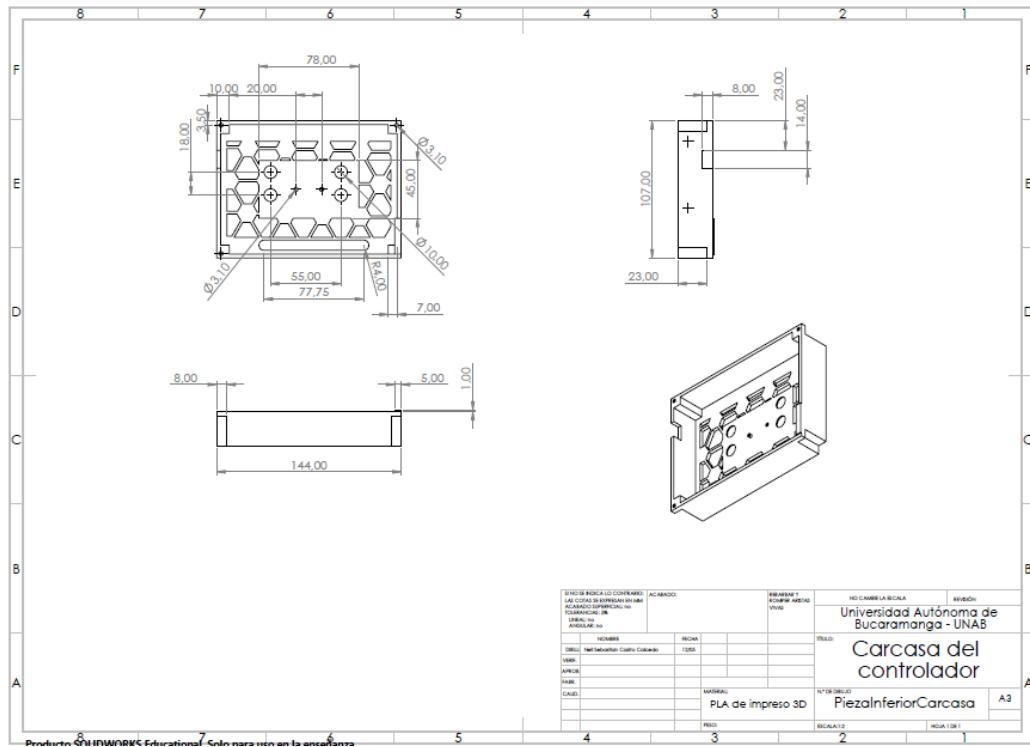


Figura 38 Plano tapa inferior.

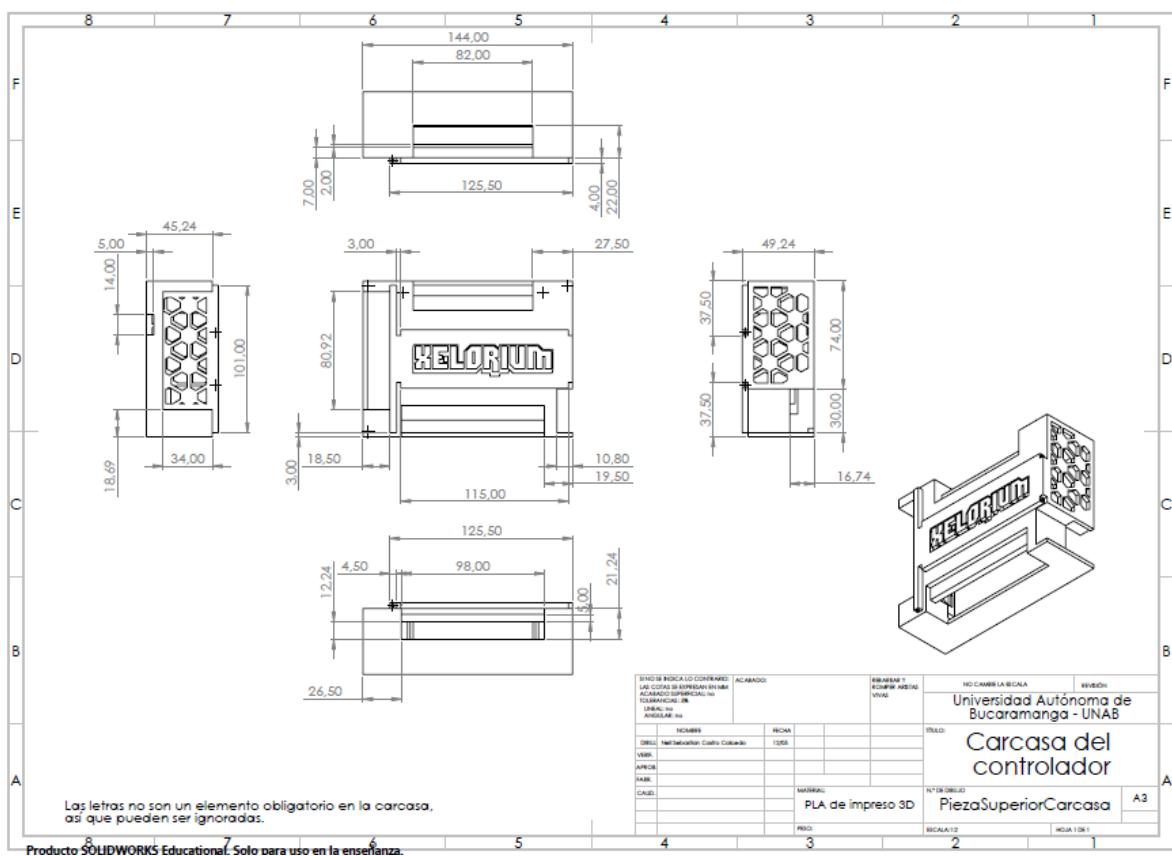


Figura 39 Plano tapa superior.

## ANEXO H: RECONSTRUCCIÓN PIEZAS FALTANTES

El análisis detallado del kit Fischertechnik disponible frente a los requerimientos de las celdas diseñadas identificó la ausencia de 19 piezas específicas. Estas piezas resultaron críticas para el correcto funcionamiento de los mecanismos y no podían ser reemplazadas por componentes alternativos del kit.

### G.1 PROCESO DE RECONSTRUCCIÓN DIGITAL

La reconstrucción digital de las piezas faltantes se realizó siguiendo estos pasos:

1. Extracción de modelos CAD desde la biblioteca del software Fischertechnik Design Studio [45]
2. Conversión de formatos a STL compatible con impresión 3D
3. Verificación de dimensiones críticas
4. Ajuste de tolerancias para ensamblaje con piezas originales
5. Preparación de archivos para impresión 3D

### G.2 COMPONENTES RECONSTRUIDOS

A continuación se detallan las piezas reconstruidas y su función específica en el sistema:

Referencia	Pieza Solidworks	Función
130593		Permite mediante un motor crear una barrera.

36532		Su función es bloquear la luz ambiental y canalizar un haz de luz concentrado a través del agujero, asegurando que solo la luz alineada pase.
172542		Pieza estructural
36227		Adaptador a presión
35065		Tensiona el brazo neumático
Propia		Pieza para el uso en las celdas
38414		Soporte para tensionar el brazo neumático

Tabla 27 Piezas CAD faltantes y su función.

Cada pieza reconstruida se sometió a pruebas de ajuste y funcionalidad con los componentes originales Fischertechnik. Las dimensiones críticas se verificaron con calibre digital, asegurando desviaciones menores a 0.1 mm respecto a las especificaciones originales.

## ANEXO I: CONEXIONES DEL SISTEMA

El sistema completo se estructura en tres subsistemas principales con sus respectivas conexiones:

1. Controlador - Banda transportadora
2. Controlador - Brazo neumático
3. Sistema neumático

Cada conexión se identifica mediante un código de referencia único que permite su rápida localización tanto en los diagramas como en el sistema físico.

### I.1 CONEXIONES CONTROLADOR - BANDA TRANSPORTADORA

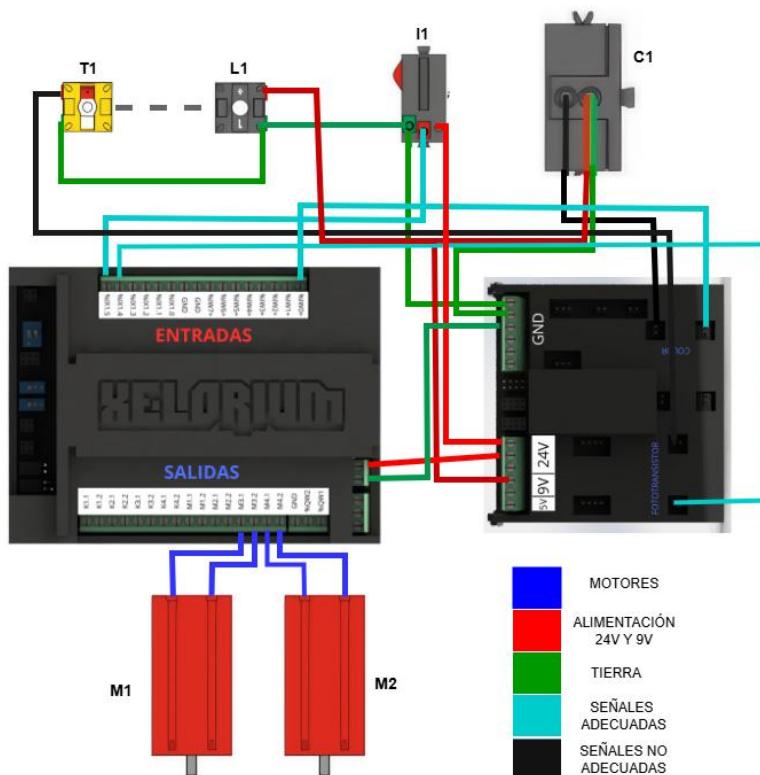


Figura 40 Diagrama de conexiones del controlador y la banda transportadora.

Referencia	Nombre	Conexión	Alimentación
M1	Motor Barrera	Los dos pines del motor están conectados al puente H del controlador ubicados en M3.1 y M3.2.	Se alimenta a través del controlador.
M2	Motor Banda transportadora	Los dos pines del motor están conectados al puente H del controlador ubicados en M4.1 y M4.2.	Se alimenta a través del controlador.
T1	Fototransistor	La señal del sensor fototransistor está conectada a la entrada del controlador IX1.4	Se alimenta con 9V
I1	Switch Barrera	La señal del Switch está conectada a la entrada del controlador IX1.5	Se alimenta con 24V
C1	Sensor Color	La señal del sensor de color está conectada a la entrada del controlador IW0	Se alimenta con 9V
L1	LED	No aplica	Se alimenta con 9V

Tabla 28 Información de conexiones de banda transportadora.

### Motor Barrera (M1)

- Tipo: Motor DC Fischertechnik
- Voltaje nominal: 9V
- Corriente nominal: 0.3A
- Función: Accionamiento de barrera para desvío de piezas

### Motor Banda transportadora (M2)

- Tipo: Motor DC Fischertechnik
- Voltaje nominal: 9V
- Corriente nominal: 0.5A
- Función: Movimiento continuo de la cinta transportadora

#### Fototransistor (T1)

- Tipo: Sensor óptico Fischertechnik
- Voltaje de operación: 9V
- Señal de salida: Digital (on/off)
- Función: Detección de presencia de pieza en la banda

#### Sensor Color (C1)

- Tipo: Sensor óptico de reflexión Fischertechnik
- Voltaje de operación: 9V
- Señal de salida: Analógica (0-9V)
- Función: Discriminación de color de la pieza

### I.2 CONEXIONES CONTROLADOR - BRAZO NEUMÁTICO

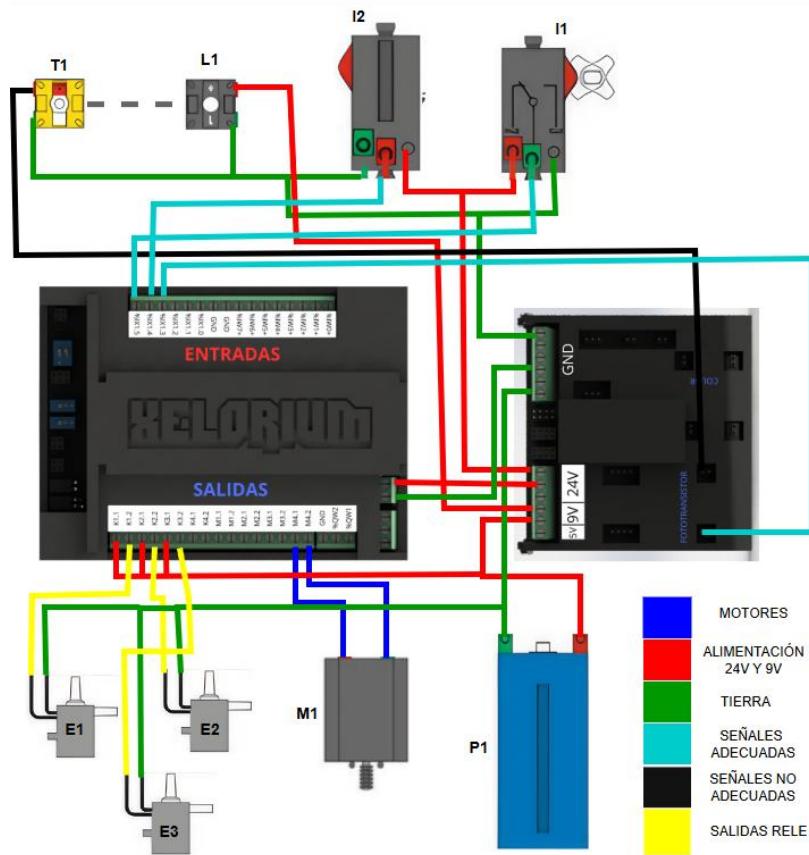


Figura 41 Diagrama de conexiones del controlador y el brazo neumático.

Referencia	Nombre	Conexión	Alimentación
M1	Motor Brazo	Los dos pines del motor están conectados al puente H del controlador ubicados en M4.1 y M4.2.	Se alimenta a través del controlador.
P1	Compresor	No aplica	Se alimenta con 9V
E1	Electroválvula Aspirar	El pin de activación está conectado al Relé del controlador ubicado en K1.2	Se alimenta a través del controlador.
E2	Electroválvula	El pin de activación está	Se alimenta a

	Sube	conectado al Relé del controlador ubicado en K2.2	través del controlador.
E3	Electroválvula Baja	El pin de activación está conectado al Relé del controlador ubicado en K3.2	Se alimenta a través del controlador.
T1	Fototransistor	La señal del sensor fototransistor está conectada a la entrada del controlador IX1.3	Se alimenta con 9V
I1	Switch Motor	La señal del Switch está conectada a la entrada del controlador IX1.5	Se alimenta con 24V
I2	Switch Barrera	La señal del Switch está conectada a la entrada del controlador IX1.4	Se alimenta con 24V
L1	LED	No aplica	Se alimenta con 9V

Tabla 29 Información de conexiones de brazo neumático.

### Electroválvulas (E1, E2, E3)

- Tipo: Electroválvula 3/2 vías Fischertechnik
- Voltaje nominal: 9V
- Corriente nominal: 0.2A
- Presión máxima: 0.5 bar

### Switch Motor/Barrera (I1, I2)

- Tipo: Microswitch de final de carrera
- Configuración: Normalmente abierto (NO)
- Voltaje máximo: 24V
- Función: Detección de posición final

### I.3 CONEXIONES NEUMÁTICAS

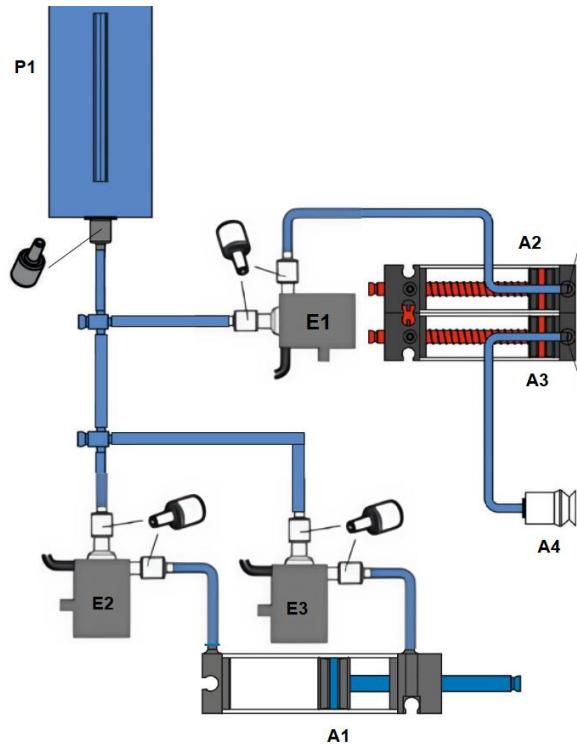


Figura 42 Diagrama de conexiones neumáticas del brazo.

Referencia	Nombre	Función
P1	Compresor	Genera aire comprimido para todo el sistema
E1	Electroválvula Aspirar	Al recibir señal, permite el paso de aire a A2
E2	Electroválvula Sube	Al recibir señal, dirige aire a A1 para extender el vástagos
E3	Electroválvula Baja	Al recibir señal, dirige aire a A1 para retraer el vástagos
A1	Pistón neumático doble efecto	Ejecuta movimiento lineal (subir/bajar) según la activación de E2 o E3

A2	Pistón neumático efecto simple con retorno	Realiza movimiento al recibir aire desde E1, retorno por resorte
A3	Pistón neumático efecto simple con retorno	Genera vacío para A4
A4	Ventosa-chupón de succión	Succiona piezas al recibir vacío desde A3

Tabla 30 Información conexiones de neumática del Brazo neumático.

- Presión de trabajo: 0.5 bar
- Diámetro de mangueras: 4 mm (estándar Fischertechnik)
- Caudal del compresor: 0.33 l/min
- Tiempo de respuesta electroválvulas: ~50 ms
- Configuración: Sistema cerrado con distribución en paralelo

## ANEXO J: TECNOLOGÍAS Y CONSIDERACIONES DE DISEÑO

El desarrollo de la plataforma SCADA Web se fundamenta en tecnologías de código abierto seleccionadas estratégicamente para maximizar la eficiencia y flexibilidad del sistema. La distribución del código fuente refleja esta orientación:

- TypeScript: 87,4% del código
- JavaScript: 12,5% del código
- Otros lenguajes: 0,1%

Esta elección permite aprovechar las ventajas de la tipificación estática ofrecida por TypeScript para mejorar la detección temprana de errores y facilitar el mantenimiento, manteniendo la flexibilidad y compatibilidad universal de JavaScript.

### J.1 JUSTIFICACIÓN DE TECNOLOGÍAS

La selección de JavaScript/TypeScript como base tecnológica se alinea con las tendencias actuales en desarrollo web e industrial, como lo demuestra el informe "Octoverse" 2023 de GitHub, que posiciona a JavaScript y TypeScript entre los lenguajes más utilizados globalmente [48].

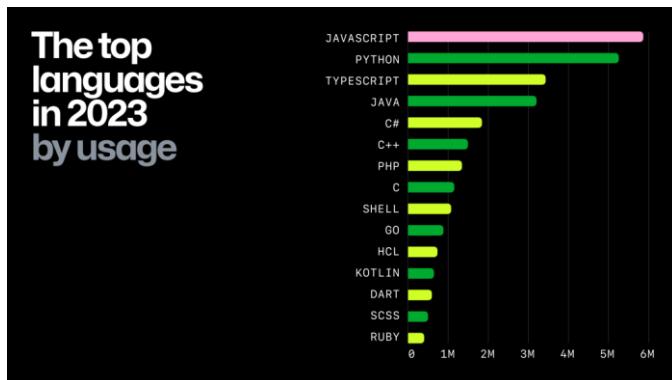


Figura 43 Octoverse 2023 Github [48].

## J.2 COMPARATIVA DE RENDIMIENTO

En escenarios de alta concurrencia, Node.js (plataforma basada en JavaScript) supera significativamente a alternativas como PHP, resultando ideal para aplicaciones intensivas en E/S. Mientras que PHP puede manejar solicitudes pequeñas, no se adapta bien a cargas elevadas. Del mismo modo, Python-Web, aunque cuenta con frameworks robustos, no es óptimo para aplicaciones web con alta demanda computacional [49].

Tecnología	Ventajas	Desventajas	Rendimiento en concurrencia
Node.js	Asíncrono, no bloqueante	Mayor complejidad	Excelente
PHP	Fácil implementación	Bloqueante	Limitado
Python-Web	Frameworks maduros	Limitaciones en GIL	Moderado

Tabla 31 Comparación tecnologías web.

### J.3 TECNOLOGÍAS UTILIZADAS EN LA PLATAFORMA SCADA

La siguiente tabla presenta las principales tecnologías empleadas en cada componente del sistema SCADA web desarrollado:

Categoría	Frontend	Backend	Gateway
Lenguaje principal	TypeScript (87.4%)	JavaScript/Node.js	JavaScript/Node.js
Framework/Librería base	React	Express	Node.js puro
Comunicación	Axios Socket.io-client	Socket.io Express REST API	Socket.io serialport midi
Gestión de datos	React Query Context API	MongoDB Mongoose	Buffer en memoria
Enrutamiento	React Router	Express Router	Express Router
Interfaz de usuario	Material-UI Recharts (gráficos)	N/A	N/A
Seguridad	JWT (cliente)	JWT (servidor) bcrypt	N/A
Depuración/Logs	React DevTools	Winston	Winston
Automatización de procesos	Webpack Babel	nodemon	nodemon
Protocolos soportados	HTTP/HTTPS WebSockets	HTTP/HTTPS WebSockets MQTT	Serial MIDI HTTP/HTTPS

Tabla 32 Librerías usadas en cada capa.

#### J.4 LIMITACIONES

El sistema SCADA Web desarrollado presenta ciertas limitaciones inherentes a su diseño y tecnologías:

- Dependencia de conectividad: Al tratarse de una plataforma web, la operabilidad del sistema está condicionada a la disponibilidad de conexión entre sus componentes.
- Latencia en tiempo real: Si bien el sistema implementa comunicación en tiempo real mediante WebSockets, existe una latencia inherente que puede no ser adecuada para procesos industriales que requieran tiempos de respuesta en escala de milisegundos críticos.
- Seguridad: Como toda aplicación web, requiere implementación de medidas robustas de seguridad para proteger tanto la información como el control de los sistemas físicos conectados.

## ANEXO K: ARQUITECTURA DETALLADA DE LA PLATAFORMA

La plataforma SCADA Web se estructura en un modelo de tres capas principales, organizadas para maximizar la escalabilidad y mantenibilidad del sistema como se observa en la figura 27.

### K.1 FRONTEND (CAPA DE PRESENTACIÓN)

#### Componentes

Elementos React que mantienen la cohesión de toda la parte visual de la plataforma, incluyendo dashboards, gráficos y paneles de control.

#### Interfaz de Usuario

Componente encargado de presentar la información y permitir la interacción del usuario con la plataforma SCADA. A través de esta capa, el usuario puede visualizar gráficos, estados de los dispositivos y emitir comandos.

#### Servicios API

Conjunto de módulos que abstraen las llamadas al backend, gestionando solicitudes HTTP y respuestas JSON. Actúan como puente entre la interfaz y la lógica del backend, garantizando que los datos se transmitan de manera ordenada y segura.

## K.2 BACKEND (CAPA DE LÓGICA DE NEGOCIO Y DATOS)

### API REST

Punto de entrada principal para todas las peticiones provenientes del frontend. Expone endpoints (rutas) que permiten consultar, crear, actualizar o eliminar recursos.

### Controladores

Módulos encargados de recibir las peticiones de la API REST, validar parámetros y coordinar la lógica necesaria para cada operación. Determinan qué servicio corresponde invocar según la ruta solicitada.

### Modelos, Servicios y Colecciones

Representan las estructuras de datos, esquemas y colecciones (datos NoSQL) que almacenan información histórica. Los modelos definen la forma en que se estructura la información y las relaciones entre entidades.

## K.3 GATEWAY (CAPA DE COMUNICACIÓN)

### Servicios de Conectividad

Componente responsable de orquestar la comunicación entre el backend y los dispositivos físicos. Define las rutinas que permiten enviar y recibir mensajes mediante distintos protocolos.

### Adaptadores de Protocolo

Módulos específicos que traducen los mensajes genéricos del sistema hacia el formato necesario para cada tipo de dispositivo. Facilitan la incorporación de protocolos nuevos sin modificar la lógica central.

### Gestión de Dispositivos

Encargada de llevar un registro de los equipos conectados, su estado de conexión y las características particulares de cada uno. Asegura que las peticiones se dirijan al dispositivo correcto.

### Buffer de Datos

Reserva temporal donde se almacenan los datos recibidos o pendientes de envío entre el sistema SCADA y los dispositivos. Permite manejar posibles retrasos o interrupciones en la red, evitando la pérdida de información.

## K.4 DISPOSITIVOS FÍSICOS

### PLC y Microcontroladores

Equipos de control que ejecutan la adquisición de señales analógicas y digitales, así como la ejecución de las acciones físicas. Reciben comandos desde el gateway y envían lecturas periódicas para su almacenamiento y análisis.

## ANEXO L: MODELOS DE BASE DE DATOS

El sistema utiliza MongoDB como base de datos NoSQL, permitiendo almacenar documentos con estructuras flexibles que se adaptan a diferentes tipos de datos generados por los dispositivos.

### L.1 MODELO BOARD

El modelo Board representa una placa o tarjeta electrónica asociada a un proyecto. Cada Board pertenece a un único proyecto y almacena información relevante sobre la placa.

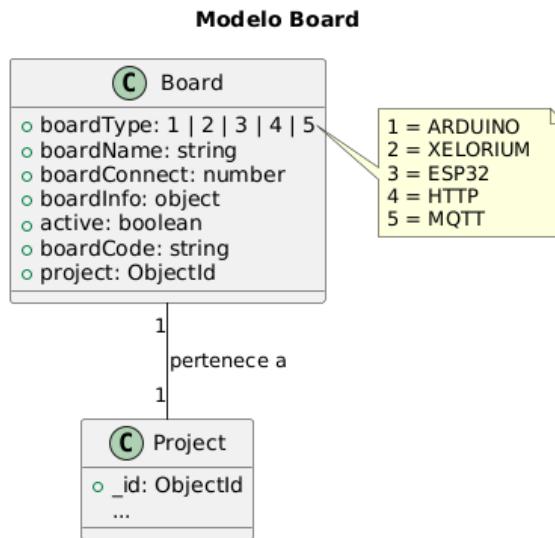


Figura 44 Modelo board.

```
interface Board {
    projectId: ObjectId; // Referencia al proyecto al que pertenece
    type: string; // Tipo de placa (Arduino, ESP32, etc.)
    name: string; // Nombre descriptivo de la placa
```

```

connections: number; // Número de conexiones disponibles
additionalInfo: string; // Información complementaria
active: boolean; // Estado de actividad de la placa
code: string; // Código identificador único
}

```

## L.2 MODELO DATAVAR

El modelo DataVar representa una entidad que almacena variables de datos asociadas a un proyecto. Cada DataVar tiene un nombre global, un nombre único para el dato, una descripción y un arreglo de valores.

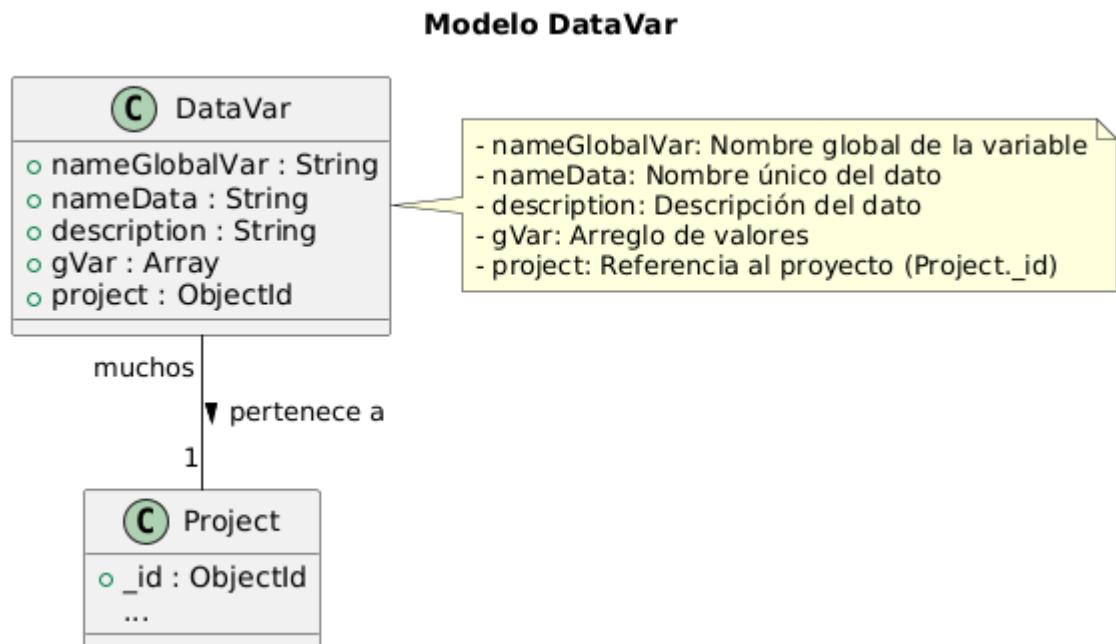


Figura 45 Modelo datavar.

```

interface DataVar {
    projectId: ObjectId; // Referencia al proyecto al que pertenece
    globalName: string; // Nombre global de la variable
}

```

```

dataName: string;      // Nombre único del dato
description: string;   // Descripción de la variable
gVar: Array<{
    value: any;
    timestamp: Date;
}>;
}

```

### L.3 MODELO PROJECT

El modelo Project representa un proyecto dentro de la plataforma. Cada proyecto tiene un nombre, descripción, estado, servidor, clave de API y relaciones con boards y dataVars.

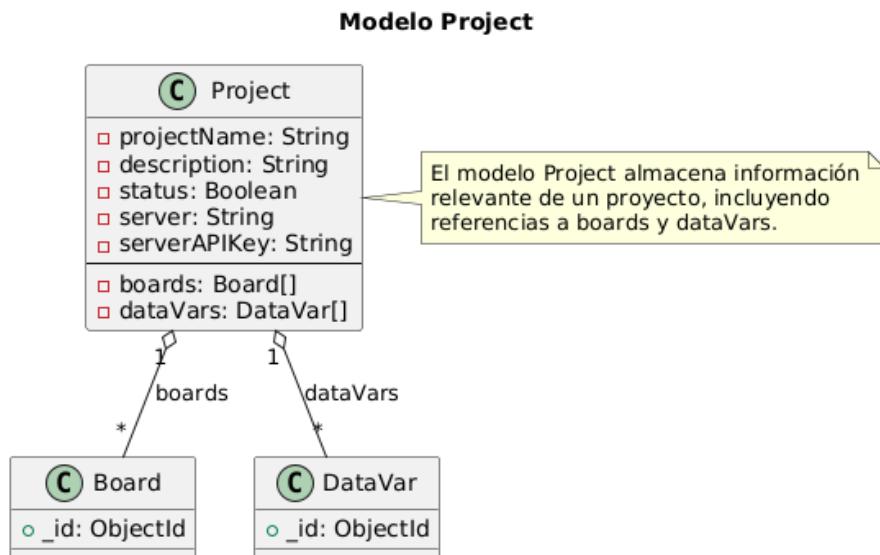


Figura 46 Modelo project.

```

interface Project {
    name: string;      // Nombre del proyecto
    description: string; // Descripción detallada
}

```

```
active: boolean;      // Estado de actividad
server: string;       // Servidor asociado
serverApiKey: string; // Clave de API para el servidor
boards: ObjectId[];   // Referencias a placas asociadas
dataVars: ObjectId[]; // Referencias a variables asociadas
}
```

## ANEXO M: IMPLEMENTACIÓN DE PROTOCOLOS DE COMUNICACIÓN

La interconexión entre el controlador lógico programable y los diferentes componentes del sistema se ve de la siguiente manera:

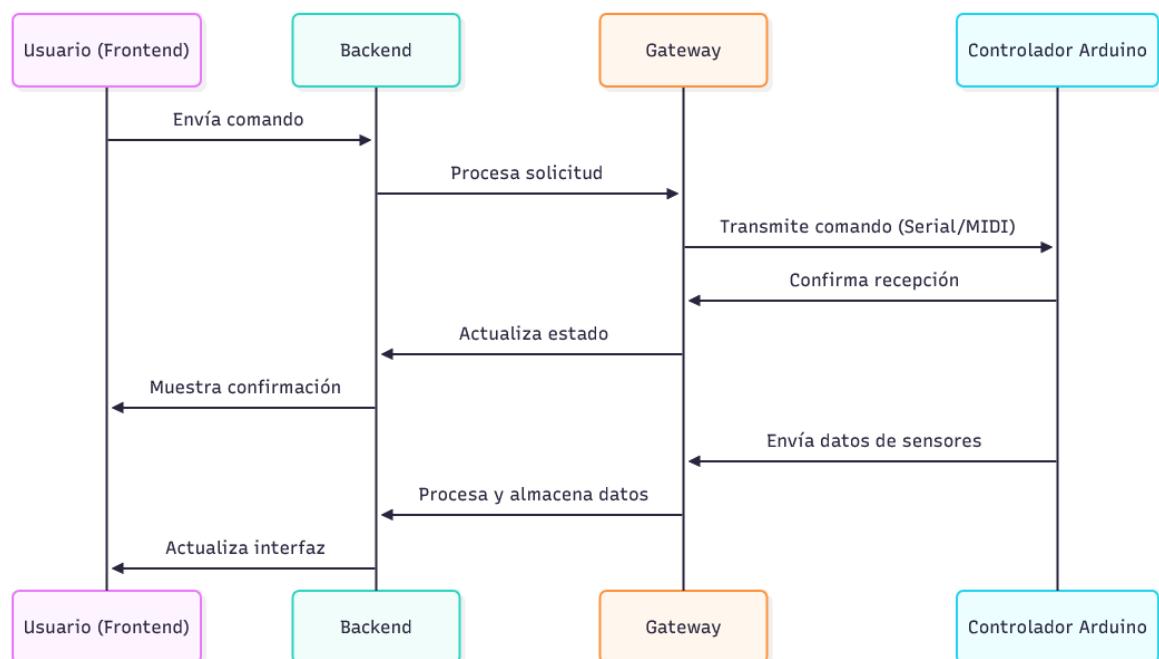


Figura 47 Interconexión componentes del sistema y controlador lógico programable.

### M.1 PROTOCOLO SERIAL

La comunicación serial con el controlador Arduino implementa un protocolo basado en tramas con la siguiente estructura:

[START\_BYTE][COMMAND][LENGTH][DATA...][CHECKSUM]

Donde:

- START\_BYTE: Byte fijo (0xAA) que indica el inicio de una trama

- COMMAND: Byte que indica el tipo de comando (lectura/escritura)
- LENGTH: Longitud de los datos
- DATA: Datos específicos del comando
- CHECKSUM: Verificación de integridad

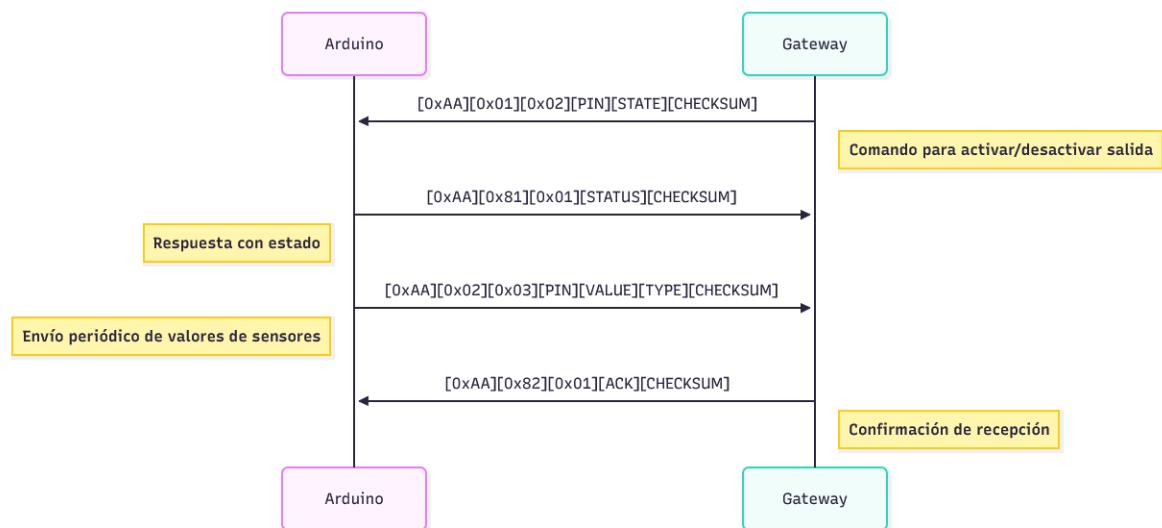


Figura 48 Estructura trama protocolo serial.

## M.2 PROTOCOLO MIDI (VARIANTE)

La comunicación WiFi mediante el ESP8266 utiliza una variante del protocolo MIDI, seleccionada por su eficiencia y baja latencia. El formato básico de los mensajes es:

[STATUS\_BYTE][DATA1][DATA2]

Donde:

- STATUS\_BYTE: Define el tipo de mensaje (0x9n = Note On, 0x8n = Note Off, etc.)
- DATA1: Primer byte de datos (nota/pin en nuestra implementación)

- DATA2: Segundo byte de datos (velocidad/valor en nuestra implementación)

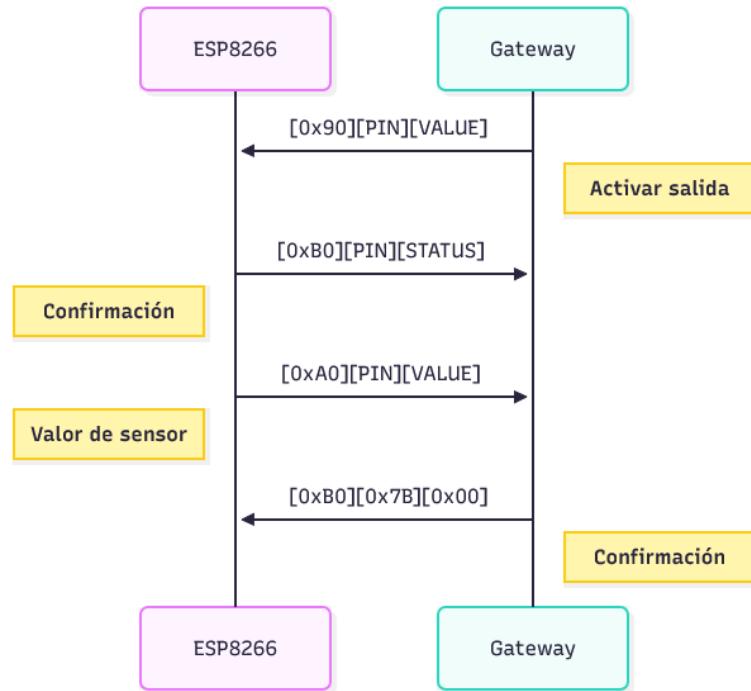


Figura 49 Estructura trama protocolo MIDI.

## ANEXO N: ESTRUCTURA DE LA RAÍZ DE ARCHIVOS

### N.1 ESTRUCTURA RAIZ DE ARCHIVOS BACKEND

El backend está implementado en Node.js utilizando Express como framework para la API REST. Su función principal es gestionar la comunicación entre el frontend y la base de datos, implementando la lógica de negocio del sistema SCADA.

Estructura de archivos raíz:

```
backend/
├── config/      # Configuraciones del servidor y base de datos
├── controllers/ # Controladores para cada recurso de la API
├── models/       # Modelos de datos de MongoDB
├── routes/       # Definición de rutas de la API
├── services/     # Servicios con lógica.
├── utils/        # Utilidades y funciones auxiliares
└── app.js        # Punto de entrada de la aplicación
└── package.json  # Dependencias del proyecto
```

Principales tecnologías utilizadas:

- Express: Framework para el desarrollo de la API REST
- Mongoose: ODM para interactuar con MongoDB
- JWT: Implementación de autenticación mediante tokens
- Socket.io: Para comunicación en tiempo real
- Winston: Sistema de logging

La API implementa un diseño RESTful con endpoints específicos para cada recurso, siguiendo las convenciones HTTP para operaciones CRUD y utilizando JWT para la autenticación y autorización de usuarios.

## N.2 ESTRUCTURA RAIZ DE ARCHIVOS FRONTEND

El frontend está desarrollado con React, utilizando TypeScript para garantizar un desarrollo más robusto y mantenible. Implementa una arquitectura basada en componentes que facilita la reutilización de código y la separación de responsabilidades.

Estructura de archivos raíz:

frontend/

```
|   └── public/      # Archivos estáticos y HTML base
|   └── src/
|       |   └── assets/    # Recursos estáticos (imágenes, estilos)
|       |   └── components/ # Componentes reutilizables
|       |   └── contexts/  # Contextos de React para estado global
|       |   └── hooks/     # Custom hooks
|       |   └── pages/     # Componentes de página
|       |   └── services/  # Servicios para API y WebSockets
|       |   └── types/     # Definiciones de tipos TypeScript
|       |   └── utils/     # Utilidades
|       |   └── App.tsx    # Componente principal
|       |   └── index.tsx  # Punto de entrada
|   └── package.json # Dependencias del proyecto
```

Principales tecnologías utilizadas:

- React: Biblioteca para construcción de interfaces
- TypeScript: Superset tipado de JavaScript
- Axios: Cliente HTTP para consumo de API
- React Router: Gestión de rutas
- React Query: Manejo de estado del servidor y caché
- Socket.io-client: Comunicación en tiempo real
- Recharts: Biblioteca para visualización de datos
- Material-UI: Framework de componentes UI

La interfaz de usuario se ha diseñado siguiendo principios de diseño responsive y usabilidad, implementando dashboards configurables que permiten la visualización de datos en tiempo real mediante gráficos y tablas interactivas.

### N.3 ESTRUCTURA RAÍZ DE ARCHIVOS GATEWAY

El gateway actúa como intermediario entre los dispositivos físicos y el backend de datos, gestionando la comunicación, transformación de datos y sincronización.

Estructura de archivos raíz:

```
gateway/
  ├── config/      # Configuraciones de conexión
  ├── controllers/ # Lógica de control para dispositivos y protocolos de
                   # comunicación
  ├── routes/      # Endpoints para transmisión de datos
  ├── utils/       # Utilidades
  ├── app.js        # Punto de entrada
  └── package.json # Dependencias
```

Principales tecnologías utilizadas:

- Node.js: Entorno de ejecución
- Socket.io: Comunicación en tiempo real con el backend
- mqtt.js: Cliente MQTT
- serialport: Comunicación serial
- Axios: Cliente HTTP para consumo de API

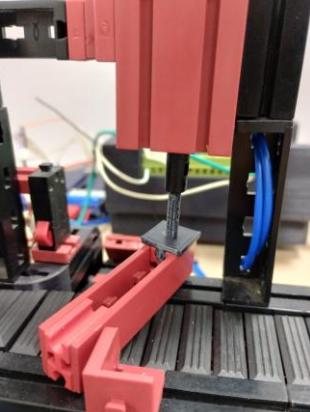
El gateway implementa un búfer de datos que permite gestionar la comunicación asíncrona entre dispositivos con diferentes velocidades y protocolos, garantizando que no se pierdan datos durante interrupciones temporales de la conexión.

## ANEXO O: VALIDACIÓN DE PIEZAS FISCHERTECHNIK

La validación de las piezas impresas en 3D para las celdas FischerTechnik siguió un proceso sistemático:

1. Identificación en el modelo CAD de la ubicación y función de cada pieza
2. Fabricación mediante impresión 3D con los parámetros establecidos
3. Colocación en las ubicaciones correspondientes según el modelo virtual
4. Verificación del ajuste dimensional y la funcionalidad mecánica
5. Documentación fotográfica del resultado

La siguiente tabla muestra los resultados de la validación de las piezas impresas en 3D en las celdas:

Imagen CAD	Pieza impresa	Ubicación en celda	Validación
		Barrera banda transportadora	Ajuste correcto, movimiento fluido

		Concentrador sensor óptico	Alineación adecuada, sin fugas de luz
		Elemento estructural	Resistencia adecuada, no presenta deformación
		Adaptador a presión	Encaje preciso, sin juego mecánico
		Tensor brazo neumático	Tensión correcta, sin deslizamiento
		Soporte de tensión	Estabilidad verificada bajo carga operativa

Tabla 33 Validación de piezas Fischertechnik.

## ANEXO P: VALIDACIÓN DEL CONTROLADOR LÓGICO PROGRAMABLE

### P.1 VALIDACIÓN DE ENTRADAS DIGITALES

La validación de las entradas digitales se basó en la comparación con la frecuencia de referencia del controlador Fischertechnik (1 kHz). Se utilizó un generador de señales Tektronix AFG1022 para producir ondas cuadradas con ancho de pulso del 50% a diferentes frecuencias (entre 20 Hz y 9000 Hz). Para cada frecuencia se realizaron diez mediciones, registrando los pulsos detectados por el controlador Arduino MEGA.

El siguiente código se implementó para la prueba:

```
volatile int pulseCount = 0; // Contador de pulsos
unsigned long previousMillis = 0; // Variable para controlar el tiempo
const long interval = 10000; // Intervalo de tiempo para imprimir (1000ms)
const int pinInterrupt = 2; // Pin de la interrupción (pin digital 2)

void setup() {
    // Inicialización del puerto serie
    Serial.begin(9600);

    // Configuración del pin de interrupción (pin 2 en este caso)
    pinMode(pinInterrupt, INPUT); // Configuramos el pin como entrada

    // Configuración de la interrupción:
    // La interrupción se activará cuando el pin 2 cambie de estado (flanco de subida o
    bajada)
    attachInterrupt(digitalPinToInterrupt(pinInterrupt), pulseDetected, RISING);
```

```

}

void loop() {
    // Verificamos si ha pasado el intervalo de 1000ms
    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis >= interval) {
        previousMillis = currentMillis; // Actualizamos el tiempo

        // Imprimimos el número de pulsos detectados
        Serial.print("Pulsos en el último segundo: ");
        Serial.println(pulseCount);

        // Reseteamos el contador de pulsos
        pulseCount = 0;
    }
}

// Función que se llama cada vez que se detecta una interrupción
void pulseDetected() {
    pulseCount++; // Incrementamos el contador de pulsos
}

```

Los resultados de las mediciones se resumen en la siguiente tabla:

Frecuencia Generador (Hz)	Pulsos Arduino IDE por segundo	Frecuencia Experimental (Hz)	Error (%)
20	20	20	0
100	100	100	0
5000	4997.93	4997.93	0.0414

9000	8996.6	8996.6	0.0378
------	--------	--------	--------

Tabla 34 Mediciones experimentales entradas digitales.

El controlador demostró una precisión excepcional con errores menores al 0.05% incluso a frecuencias muy superiores al requisito de 1 kHz. Supera ampliamente los requisitos necesarios para compatibilidad con el sistema Fischertechnik y ofrece un rendimiento muy superior al esperado en entornos educativos (donde se aceptan errores de 5-10%) [46].

## P.2 VALIDACIÓN DE ENTRADAS ANALÓGICAS

Para validar las entradas analógicas, se compararon las lecturas del controlador con mediciones realizadas mediante un multímetro digital Unit UT33C. Se utilizó una fuente de laboratorio para generar voltajes entre 0 y 10 V, ajustados mediante un potenciómetro.

El siguiente código se implementó para la prueba:

```
void setup() {
    Serial.begin(9600);
}

void loop() {
    int sensorValue = analogRead(A0);
    float voltage = sensorValue * (5.0 / 1023.0);
    Serial.println(voltage);
}
```

Los resultados de las mediciones se resumen en la siguiente tabla:

Fuente (V)	Multímetro (V)	Controlador sin compensar (V)	Error (%)	Controlador con compensador (V)	Error (%)
1	1.05	0.9872	5.9810	1.033	1.6190
2	2.03	1.9845	2.2414	2.0109	0.9409
3	3.01	3.0313	0.7076	2.9985	0.3821
4	4.06	4.0469	0.3227	3.9958	1.5813
5	5.03	5.0146	0.3062	5.003	0.5368
6	6.09	6.0217	1.1215	5.9842	1.7373
7	7.07	6.9794	1.2815	7.0758	0.0820
8	8.03	7.8983	1.6401	8.1863	1.9465
9	9.08	8.7012	4.1718	9.1138	0.3722
9.5	9.56	9.1257	4.5429	9.4189	1.4759

Tabla 35 Mediciones experimentales entradas analógicas.

### ENTRADA ANALOGA/Controlador sin compensar V vs. ENTRADA ANALOGA/Fuente V

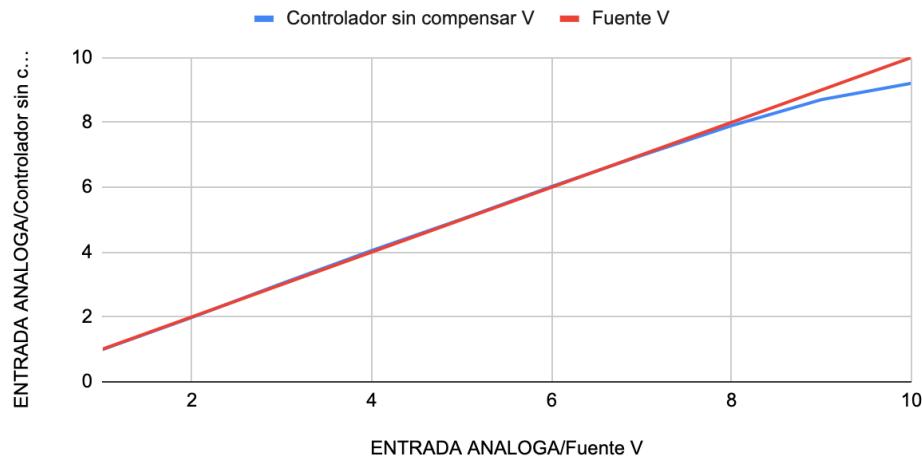


Figura 50 Entrada análoga/controlador sin compensador [V] vs Entrada análoga/fuente [V].

## ENTRADA ANALOGA/Controlador con compensador V vs. ENTRADA ANALOGA/Fuente V

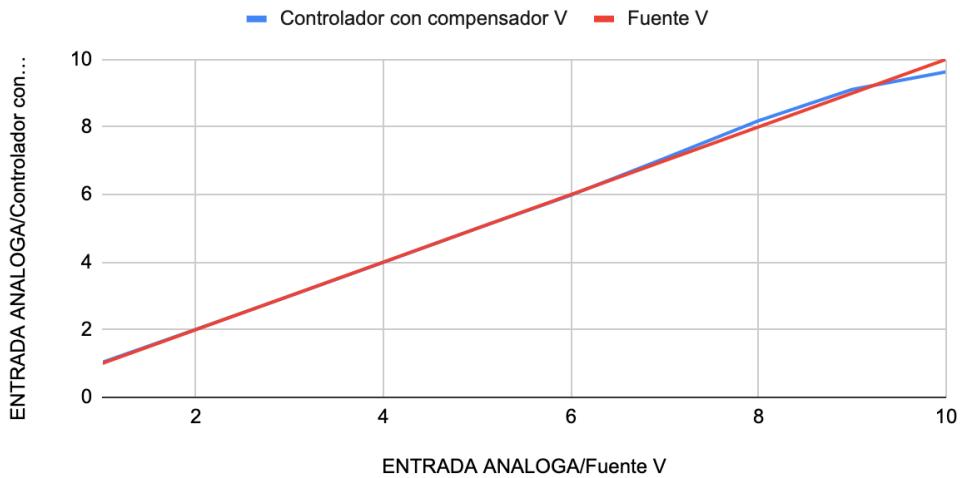


Figura 51 Entrada análoga/controlador con compensador [V] vs Entrada análoga/fuente [V].

El controlador Fischertechnik reporta errores de aproximadamente 6.8% en valores bajos y 4.5% en valores altos para entradas analógicas. El controlador desarrollado, sin compensación, muestra un comportamiento similar con errores que aumentan al acercarse a 10V (llegando hasta 9%) como se muestra en la figura 50, debido a la curva característica del diodo Zener de protección.

Con la implementación del compensador por software, el error máximo se reduce al 5%, lo que es comparable con el rendimiento del controlador Fischertechnik de referencia y cumple con los estándares aceptables para entornos educativos ( $\leq 5\%$  para proyectos básicos) [46]. La linealidad también mejora significativamente en el rango de 0-8V, donde el error se mantiene por debajo del 2%, superando al controlador de referencia como se observa en la figura 51.

### P.3 VALIDACIÓN DE SALIDAS ANALÓGICAS

Las salidas analógicas se validaron comparando la tensión programada en el controlador con mediciones realizadas mediante un multímetro digital Uni-T UT33C y un osciloscopio Tektronix TBS 1052B-EDU.

El siguiente código se implementó para la prueba:

```
const int pwmPin = 9;  
const int voltage_mV = 2500;  
  
void setup() {  
    pinMode(pwmPin, OUTPUT);  
}  
  
void loop() {  
    int dutyCycle = map(voltage_mV, 0, 5000, 0, 255);  
    analogWrite(pwmPin, dutyCycle);  
}
```

Los resultados de las mediciones se resumen en la siguiente tabla:

Controlador (V)	Multímetro (V)	Osciloscopio (V)	Error (%)
1	1.01	1.01	0.9901
2	2.03	2.02	0.9901
3	3	3.02	0.6623
4	4.01	4.04	0.9901
5	4.99	5.01	0.1996
6	6	6.03	0.4975

7	6.97	7.01	0.1427
8	7.99	8.04	0.4975
9	8.96	9.08	0.8811
10	9.98	10.1	0.9901

Tabla 36 Mediciones experimentales salidas analógicas.

El controlador Fischertechnik presenta una desviación aproximada del 5% por debajo del valor nominal en sus salidas PWM (entregando ~8.74V cuando debería proporcionar 9-9.25V). En contraste, el controlador del proyecto muestra errores inferiores al 1% en todo el rango de 1V a 10V, lo que representa un rendimiento significativamente mejor que el controlador de referencia.

Esta precisión superior es resultado de la implementación cuidadosa del filtro RC y la etapa de amplificación, que permiten una conversión más exacta de las señales PWM a valores analógicos estables.

#### P.4 VALIDACIÓN DE SALIDAS MOTORES PWM Y RELÉ

Las salidas para motores PWM y relés fueron validadas mediante pruebas funcionales con los actuadores de las celdas Fischertechnik. Al no contar con datos específicos sobre el rendimiento y los márgenes de error admisibles en las salidas del controlador Fischertechnik original, la validación se basó en la verificación del correcto funcionamiento con los motores y electroválvulas de las celdas.

Se verificaron los siguientes aspectos:

1. Control de velocidad de motores: Capacidad para variar la velocidad de los motores de la banda transportadora y barrera mediante señales PWM.
2. Control de dirección: Capacidad para invertir la dirección de rotación de los motores mediante el puente H L293D.

3. Activación de relés: Capacidad para activar y desactivar las electroválvulas neumáticas a través de los relés mecánicos.
4. Tiempo de respuesta: Verificación de que los tiempos de activación y desactivación son adecuados para las operaciones de las celdas.
5. Estabilidad: Verificación de la operación continua durante ciclos repetidos sin sobrecalentamiento o degradación del rendimiento.

## ANEXO Q: VALIDACIÓN DEL SISTEMA INTEGRADO

### Q.1 DIAGRAMA DE FLUJO DEL BRAZO NEUMÁTICO

El proceso de la primera celda (brazo neumático) sigue la siguiente secuencia:

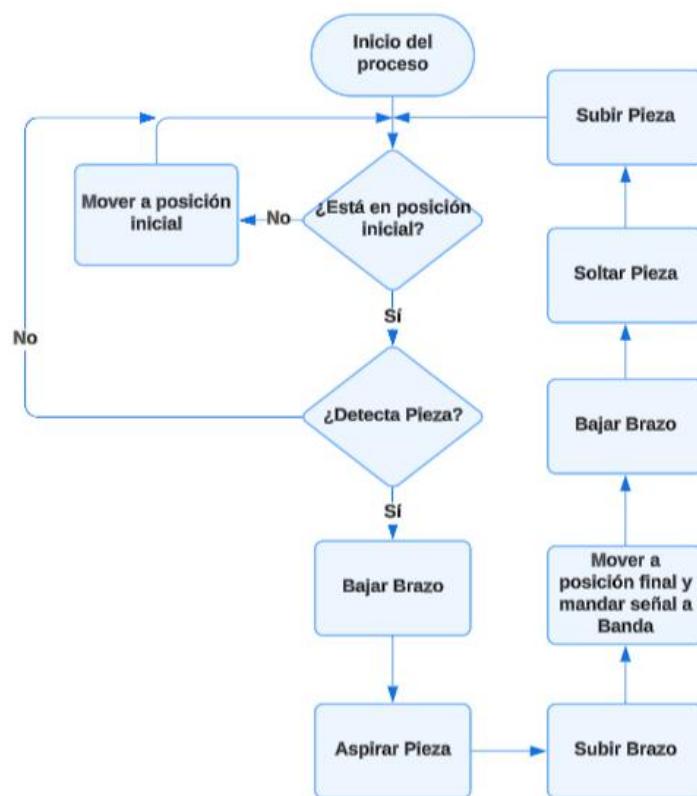


Figura 52 Diagrama de flujo del proceso del brazo neumático.

El proceso inicia con la detección de una pieza en la posición inicial del brazo neumático. Mediante el sistema neumático, el brazo toma la pieza y la transporta hasta una posición final, que corresponde al punto de inicio de la celda de banda.

transportadora. Al llegar a esta ubicación, se envía una señal a la siguiente celda y se deposita la pieza, tras lo cual el brazo regresa a su posición inicial para esperar una nueva pieza.

## Q.2 DIAGRAMA DE FLUJO DE LA BANDA TRANSPORTADORA

El proceso de la segunda celda (banda transportadora) sigue la siguiente secuencia:

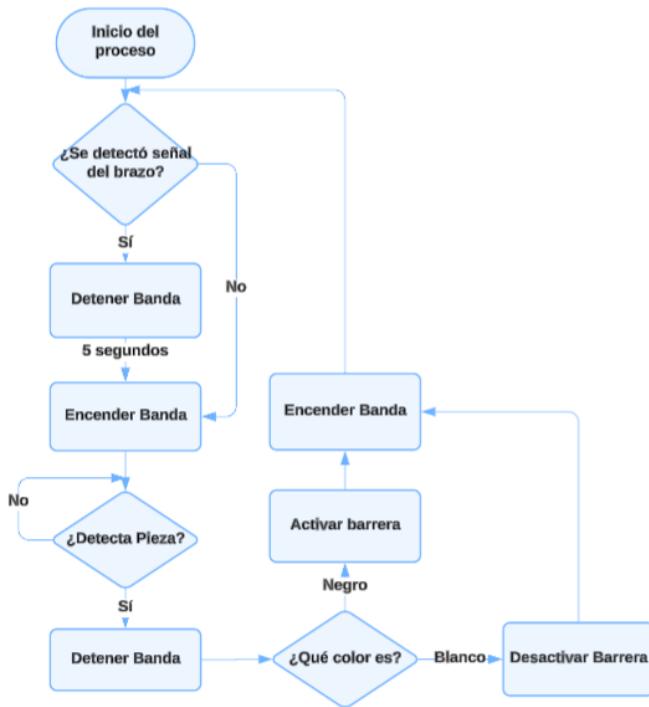


Figura 53 Diagrama de flujo del proceso de la banda transportadora.

Este es el segundo proceso del sistema, en el cual, tras recibir la señal desde la celda del brazo neumático, la banda transportadora se detiene momentáneamente para permitir que la pieza sea depositada. Luego, se activa el sistema de detección, que identifica el color de la pieza. Si la pieza es blanca, se le permite continuar su

trayecto; si es negra, se activa una barrera que la desvía hacia otra ruta. Este ciclo se repite a la espera de la siguiente pieza.

### Q.3 VALIDACIÓN DE SENSORES DE LAS CELDAS

#### Q.3.1 VALIDACIÓN DE FOTOTRANSISTORES

Los fototransistores presentes en ambas celdas fueron evaluados para verificar su capacidad de detección. Se realizaron 20 pruebas (10 con objeto presente y 10 con objeto ausente) con los siguientes resultados:

Estado	Detecciones correctas	Detecciones incorrectas	Precisión
Objeto presente (1)	10	0	100%
Objeto ausente (0)	10	0	100%

Tabla 37 Validación fototransistor.

Estos resultados confirman el correcto funcionamiento de los sensores fototransistores en todas las condiciones evaluadas.

#### Q.3.2 VALIDACIÓN DEL SENSOR DE COLOR

El sensor de color de la banda transportadora fue caracterizado con cuatro colores diferentes. Cada color se evaluó en 10 mediciones independientes, posicionando la pieza en el punto exacto de detección durante el proceso.

SENSOR COLOR BANDA	
COLOR	VOLTAJE AL CONTROLADOR
BLANCO	4.23
BLANCO	6.01
BLANCO	4.3
BLANCO	4.3
BLANCO	4.84
BLANCO	4.22
BLANCO	5.01
BLANCO	4.96
BLANCO	5.2
BLANCO	4.69
NEGRO	7.8
NEGRO	7.94
NEGRO	8.07
NEGRO	7.91
NEGRO	7.86
NEGRO	7.99
NEGRO	8.3
NEGRO	8.25
NEGRO	7.83
NEGRO	7.86
VERDE	6.91
VERDE	7.51
VERDE	6.83
VERDE	6.95
VERDE	7.4
VERDE	6.77
VERDE	7.01
VERDE	7.13
VERDE	6.77
VERDE	6.95
AZUL	5.91
AZUL	6.33
AZUL	5.89
AZUL	5.92
AZUL	5.95
AZUL	5.91
AZUL	6.95
AZUL	5.97
AZUL	5.93
AZUL	6.68

Tabla 38 Validación valores sensor de color.

La posición de lectura del color en la banda transportadora se muestra en la siguiente imagen:

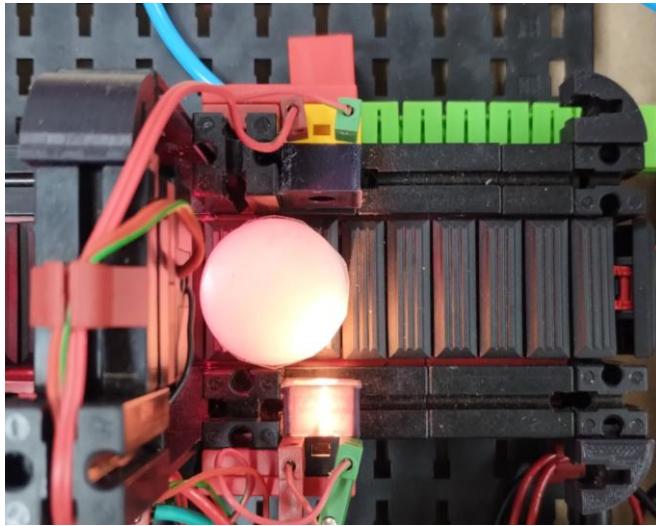


Figura 54 Posición correcta lectura de color pieza.

Los resultados muestran que los colores blanco y negro ofrecen la mejor diferenciación y la menor variabilidad en las mediciones, por lo que fueron seleccionados para la clasificación en el sistema final.

#### Q.4 CATEGORIAS POSIBLES ERRORES

Se registró la presencia de posibles errores en las siguientes categorías:

1. Error de programación: Fallos relacionados con el código, la conectividad o la plataforma.
2. Error al detectar pieza: El sistema no reconoce la presencia de una pieza.
3. Error al aspirar pieza: El brazo neumático no logra tomar correctamente la pieza o la suelta antes de llegar a su destino.
4. Error de movimiento: El brazo presenta trabas o no se desplaza correctamente.
5. Error al dejar la pieza: La pieza no es depositada correctamente o se cae durante la operación.
6. Error en la banda: La banda transportadora no se activa cuando debería.

7. Error en la barrera: La barrera no se activa o desactiva adecuadamente según el color de la pieza.
8. Error en interruptores: Los interruptores de las celdas no son leídos correctamente por el sistema.

## ANEXO R: VALIDACIÓN DE GUÍAS DE LABORATORIO

Se utilizaron las siguientes métricas para evaluar la eficacia de las guías:

1. Tiempo de instrucción por laboratorio: Tiempo dedicado a la exposición teórica inicial de los objetivos, la descripción de la plataforma y la demostración de los procedimientos básicos antes de iniciar la práctica. Se estandarizó en aproximadamente 5 minutos para cada sesión.
2. Tiempo de realización de cada laboratorio: Medido desde que los estudiantes iniciaron efectivamente la práctica hasta la entrega de los resultados obtenidos.
3. Encuestas de satisfacción: Aplicadas al finalizar cada práctica para evaluar aspectos como claridad, facilidad de uso, relevancia percibida y motivación generada.

### R.1 RESULTADOS POR SECCIÓN EVALUADA

#### R.1.1 USO GENERAL

Siguiendo la guía paso a paso, ¿pudo establecer la conexión entre su dispositivo y el Gateway sin inconvenientes?

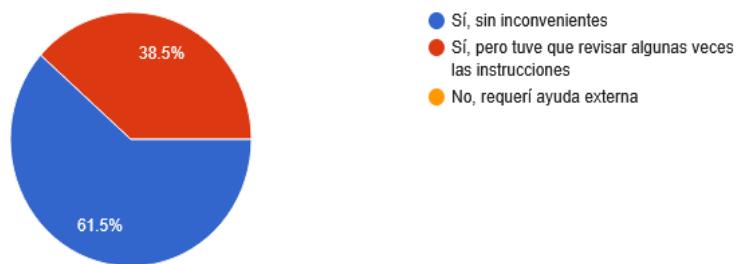


Figura 55 Resultados conexión Gateway-Controlador.

En comparación con otras plataformas SCADA que haya utilizado, ¿cómo evaluaría la facilidad de conexión entre el controlador, el Gateway y la plataforma Ilyzaelle, desde la apertura del gateway manager hasta la primera conexión del controlador?

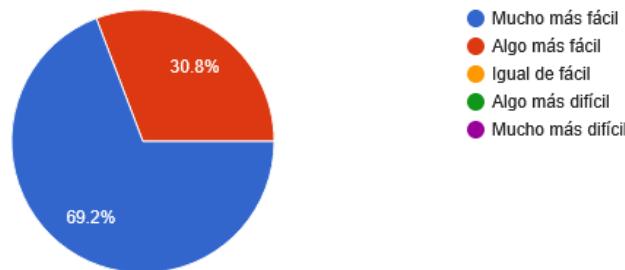


Figura 56 Resultados Comparación con otras plataformas SCADA.

¿Encontró intuitiva la interfaz para editar variables, agregar gráficos y paneles en la plataforma?

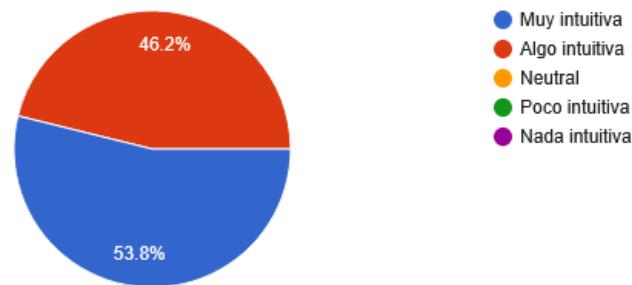


Figura 57 Resultados de la interfaz.

¿Qué aspecto del SCADA recomendaría mejorar o agregar?

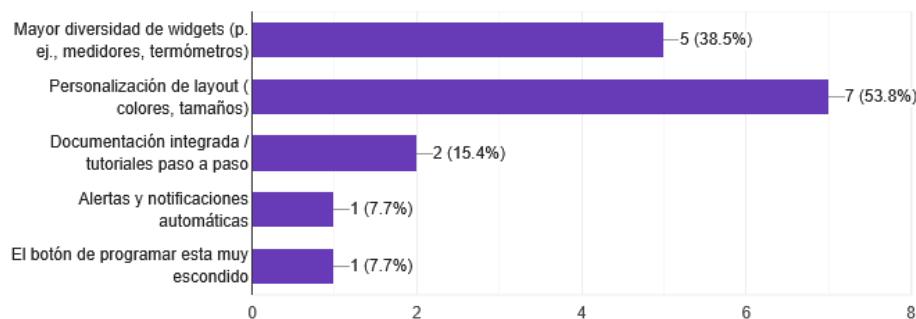


Figura 58 Resultados aspectos a mejorar.

## R.1.2 PROGRAMACIÓN DEL CONTROLADOR EN LA PLATAFORMA

¿Cómo califica su conocimiento de la sintaxis y funciones de JavaScript aplicadas en la plataforma?

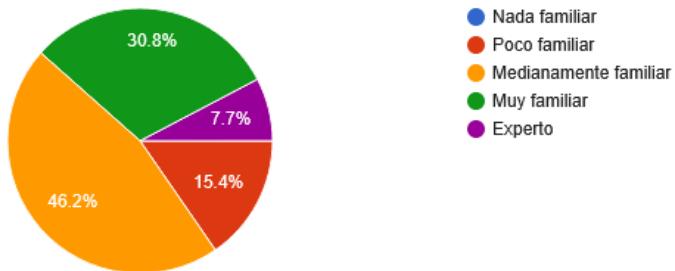


Figura 59 Resultados conocimiento de JavaScript.

En términos de satisfacción, ¿qué tan conforme quedó con el editor de código (funcionalidades Ejecutar/Guardar, resultado de sintaxis)?

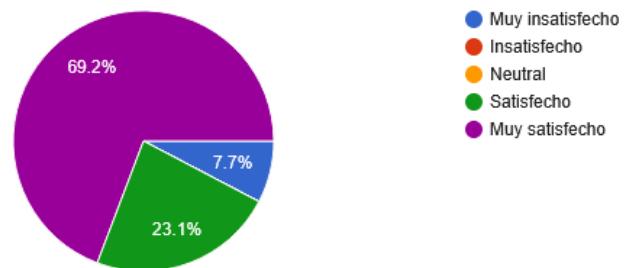


Figura 60 Resultados satisfacción con el editor de código.

¿Qué tan eficiente le pareció el uso de las funciones relacionadas con la lectura y escritura de los pines GPIO del controlador?

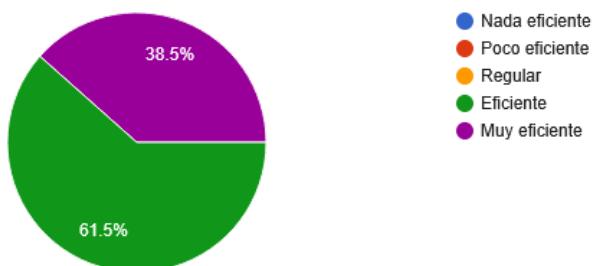


Figura 61 Resultados eficiencia de lectura/escritura de GPIO.

¿Qué tan diferente le pareció la experiencia de programar los controladores en JavaScript en comparación con el lenguaje nativo de Arduino (C/C++)?

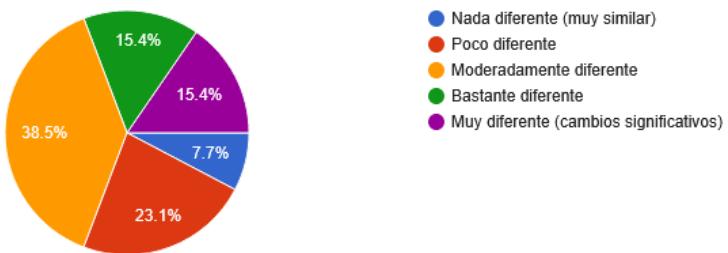


Figura 62 Resultados diferencia frente a C/C++ de Arduino.

### R.1.3 CELDA MECATRÓNICA FISCHERTECHNIK

¿Qué tan satisfecho quedó con la implementación del contador de piezas y las banderas de estado?

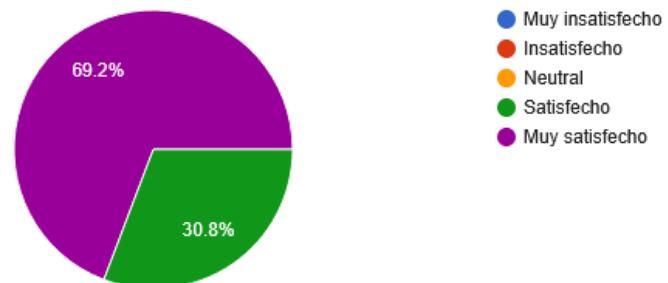


Figura 63 Resultados satisfacción con contador de piezas y banderas.

¿En qué medida las celdas representan procesos industriales reales como clasificación y transporte?

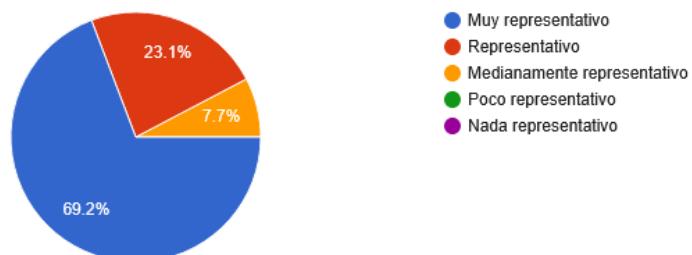


Figura 64 Resultados representatividad de procesos industriales.

Durante la ejecución de la celda, ¿experimentó interrupciones, errores o comportamientos inesperados en el sistema?

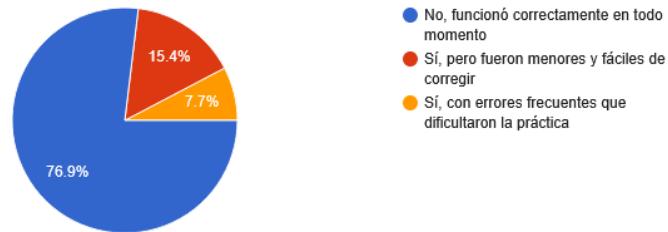


Figura 65 Resultados interrupciones o errores durante ejecución.

¿Sintió que la interacción entre el entorno virtual (dashboard SCADA) y la celda física fue fluida y coherente?

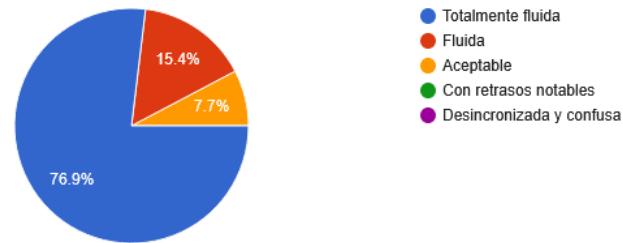


Figura 66 Resultados fluidez entre SCADA y celda física.

#### R.1.4 GENERALIDADES

¿Con qué frecuencia necesitaste ayuda del tutor para resolver dudas o problemas durante la realización de las guías?

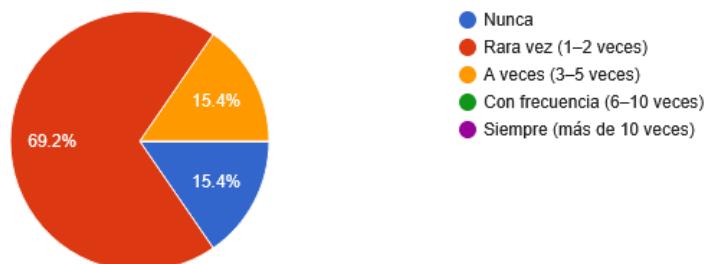


Figura 67 Resultados frecuencia de consulta al tutor.

¿Qué tan relevantes fueron los temas abordados para las asignaturas de redes de comunicación industrial y automatización industrial?

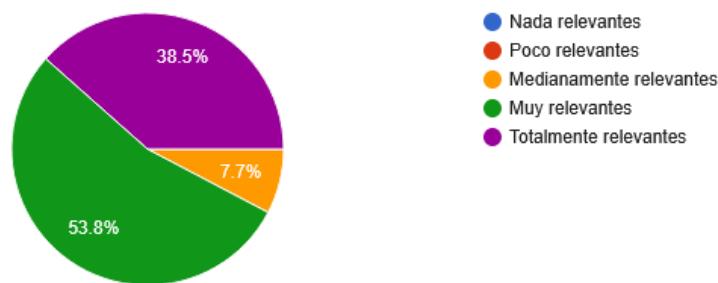


Figura 68 Resultados relevancia para redes y automatización.

¿Consideraste que las guías estaban organizadas de forma lógica y progresiva? [Copy](#)

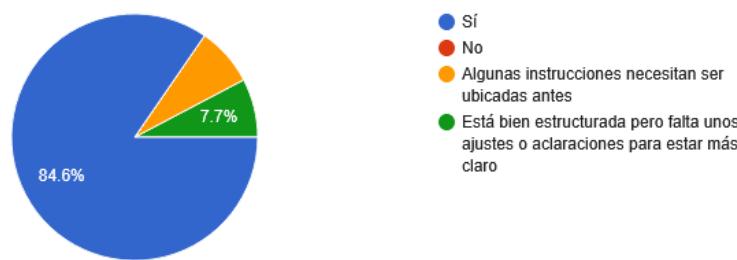


Figura 69 Resultados relevancia claridad de las guías.

Después de realizar las guías, ¿consideraste que existía la posibilidad de utilizar la plataforma para trabajos o proyectos futuros?

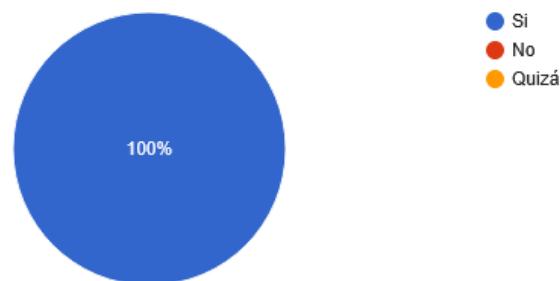


Figura 70 Resultados potencial de uso futuro.

El análisis de los datos cuantitativos (tiempos, porcentajes de satisfacción) y cualitativos (observaciones, comentarios abiertos) confirma que las guías de laboratorio cumplen con los objetivos de instruir, guiar y motivar a los estudiantes en un entorno IoT-SCADA/Arduino.

Se evidenció que los objetivos específicos de cada sección (facilidad de conexión, edición y visualización en SCADA, eficiencia en la programación de GPIO, representatividad de la celda mecatrónica y aprendizaje de JavaScript) se alcanzaron en la mayoría de casos, lo que respalda la eficacia de la metodología de laboratorios guiados.

Las áreas de mejora identificadas incluyen:

1. Optimizar la ubicación de controles clave en la interfaz
2. Ampliar la documentación integrada
3. Clarificar conceptos de programación asíncrona
4. Robustecer la estabilidad en la interacción real entre SCADA y la celda física

El elevado nivel de motivación reportado (100%) y el interés unánime en emplear la plataforma en proyectos futuros validan la escalabilidad de las guías para cursos de redes de comunicación industrial y automatización industrial.