



Pontificia Universidad  
**JAVERIANA**  
Cali

**Facultad de Ingeniería  
y Ciencias**  
Ingeniería Electrónica

MONOGRAFÍA DE TRABAJO DE GRADO

Desarrollo de sistema de control de un robot delta  
lineal basado en la norma ANSI/RIA R15.06-2012

Juan Sebastián Pinzón Cortés  
Jorge Alejandro Ángel Moreno

*Director*

M.Sc. Juan David Contreras

*Codirector*

B.Sc. Bleider Johan Arizala

15 de febrero de 2024





## **Acta de Correcciones al Proyecto de Grado Ingeniería Electrónica**

**Fecha:** 13/02/2024

**Autores:** Juan Sebastian Pinzon Cortes  
Jorge Alejandro Angel Moreno

**Nombre del Proyecto de Grado:**

Desarrollo de sistema de seguridad de control de un robot delta lineal basado en la norma ANSI/RIA R15.06-2012

**Director:** Juan David Contreras

Como indica el artículo 2.27 de las Directrices de Trabajo de Grado, he verificado que los estudiantes indicados arriba han implementado todas las correcciones que los Jurados del Proyecto de Grado definieron que se efectuaran, como consta en el Acta de Calificación correspondiente.

Firma de Director(a) del Proyecto de Grado

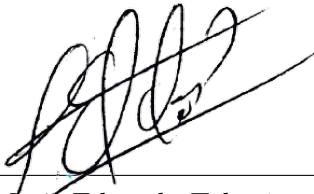
Nota de Aceptación

Aprobado por el Comité de Trabajo de Grado en cumplimiento de los requisitos exigidos por la Pontificia Universidad Javeriana para optar al título de Ingeniero Electrónico.

*Camilo Rocha*

---

PhD. Hernán Camilo Rocha Niño  
Decano Facultad de Ingeniería y Ciencias



---

PhD. Luis Eduardo Tobon  
Director Carrera Ingeniería Electrónica



---

M.Sc. Juan David Contreras  
Director Trabajo de Grado



---

B.Sc. Bleider Johan Arizala  
Co-Director Trabajo de Grado

*Alexander Martínez A.*

---

PhD. Alexander Martínez Álvarez  
Jurado 1

*L.A.S. Plata E.*

---

B.Sc. Simon Plata  
Jurado 2



Santiago de Cali, 15 de febrero de 2024

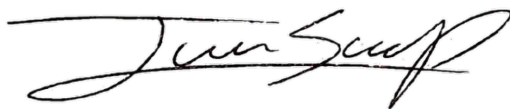
Señores  
Pontificia Universidad Javeriana – Cali  
PhD. Hernán Camilo Rocha Niño  
Decano  
Facultad de Ingeniería y Ciencias  
Ciudad

Cordial Saludo.

Por medio de la presente nos permitimos presentarle el Trabajo de Grado titulado “Desarrollo de sistema de control de un robot delta lineal basado en la norma ANSI/RIA R15.06-2012”.

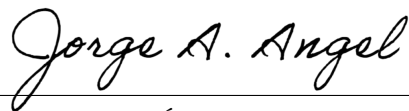
Esperamos que este trabajo reúna todos los requisitos académicos, cumpla el propósito para el cual fue creado y sirva de apoyo para futuros proyectos relacionados con la materia.

Atentamente,



---

Juan Sebastián Pinzón Cortés



---

Jorge Alejandro Ángel Moreno



Santiago de Cali, 15 de febrero de 2024

Señores

**Pontificia Universidad Javeriana – Cali**

PhD. Hernán Camilo Rocha Niño

Decano

Facultad de Ingeniería y Ciencias

Ciudad

Cordial Saludo.

Certificamos que el presente Trabajo de Grado titulado “Desarrollo de sistema de control de un robot delta lineal basado en la norma ANSI/RIA R15.06-2012”, realizado por Juan Sebastián Pinzón Cortés y Jorge Alejandro Ángel Moreno, estudiantes de Ingeniería Electrónica, se encuentra terminado y puede ser presentado para su sustentación.

Atentamente,



M.Sc. Juan David Contreras  
Director Trabajo de Grado



B.Sc. Bleider Johan Arizala  
Co-Director Trabajo de Grado



# Agradecimientos

El agradecimiento principal va dirigido a Dios, quien nos ha guiado y bendecido en cada etapa de nuestras vidas. Su grandeza se refleja en la sabiduría e inteligencia que nos ha brindado para cursar esta maravillosa carrera, la cual culminamos hoy con excelentes resultados y logros personales. Le agradecemos también, por poner en nuestro camino a personas tan importantes y por cuidar de ellas, quienes han sido pilares fundamentales en nuestra formación tanto personal como profesional. A nuestros padres, quienes siempre han estado con su apoyo incondicional hacia nosotros. Sus sacrificios, amor, paciencia y orientación han sido pilares fundamentales en nuestra vida que nos han permitido llegar aquí. A nuestra familia que siempre está atenta, compartiendo nuestras alegrías y logros, deseándonos siempre lo mejor y ayudándonos en todo lo que han podido.

A nuestro director, que a pesar de sus ocupaciones nos ha apoyado desde la distancia, orientándonos y aconsejándonos con paciencia y dedicación. Su guía ha sido de gran valor en cada paso de nuestro proyecto, brindándonos la claridad y la dirección necesarias para superar obstáculos. A nuestro codirector, que desde su área y especialidad, nos ayudó a superar gran cantidad de retos y que con sus valiosas contribuciones y compromiso, se logró elevar la calidad de nuestro proyecto.

A nuestros compañeros que siempre fuimos un equipo de los que ahora muchos son nuestros más cercanos amigos, a nuestro director de Carrera, quien nos ha ayudado a tomar importantes decisiones, y siempre está pendiente de los problemas y requerimientos de todos. Y a nuestros profesores, que con su dedicación y talento, han logrado compartir con nosotros su conocimiento y experiencia de manera excepcional. Su compromiso con la enseñanza ha sido una fuente constante de inspiración en todo nuestro proceso de aprendizaje.



# Glosario

## Símbolos

$\delta_i$  Se

## Acrónimos y Abreviaturas

*ANSI* Instituto Nacional Estadounidense de Estándares  
*CAP* Centro de automatización de procesos  
*HMI* Interfaz Hombre-Máquina  
*ISO* Organización Internacional de Normalización  
*RDL* Robot delta lineal  
*PLC* Controlador lógico programable

## Términos

*PYME* Empresas pequeñas y medianas con un número limitado de empleados e ingresos.  
*Pendant* Dispositivo de entrada para controlar máquinas, robots y equipos.



# Resumen

La robótica industrial es una tecnología que ha logrado darle valor agregado a la producción en masa de artículos en diferentes campos a nivel mundial. Esta ha logrado que los productos sean fabricados en un tiempo menor, debido a la rapidez en la ejecución de tareas; con una mayor precisión, debido al bajo error de un robot al ejecutar una orden y de una forma más segura, puesto que un robot puede trabajar en ambientes hostiles, como altas temperaturas, lo cual podría ser perjudicial para un operario humano. Sin embargo, la implementación de estos sistemas robóticos comerciales es muy costosa, por lo que empresas en vía de desarrollo no tienen el poder económico para adquirirlos.

Una solución este problema es la implementación de robots hechos a la medida, los cuales serían más accesibles para las empresas y potenciar así el desarrollo de nuevos negocios. A pesar de tener un costo menor, estos robots deben cumplir con estándares de seguridad, que garanticen la integridad de los operarios dentro de las instalaciones donde se utilicen. El objetivo de este trabajo es diseñar e implementar un sistema de seguridad industrial en un robot Delta lineal, ubicado en el Centro de Automatización de Procesos, en la Universidad Javeriana Cali. Se llevó a cabo una evaluación de riesgos para determinar los potenciales peligros a los que los operarios o la infraestructura del dispositivo podrían estar expuestos. Con base a estos resultados, se determinaron los requisitos que el sistema debe cumplir. Posteriormente, se diseñó una arquitectura de seguridad capaz de mitigar los riesgos y cumplir con los requerimientos establecidos. Esta arquitectura se implementó en el robot y se evaluó para validar su seguridad. Durante el desarrollo de este proyecto, se siguieron los lineamientos y estándares establecidos por una entidad internacional reguladora, como ANSI RIA.

**Palabras Clave:** Robot Delta Lineal, HMI, Dispositivos de protección, Requerimientos.



# Abstract

Industrial robotics is a technology that has managed to add value to the mass production of items in different fields worldwide. Thanks to the robotics integration to mass production products are manufactured in a shorter time, due to the speed in the execution of tasks; with greater precision, due to the low error of robot's when executing a tasks, and in a safer way, because robot's can work in hostile environments, such as high temperatures, which could be harmful for humans. However, the implementation of commercial robotic systems is very expensive, making it imposible for small or growing companies to acquire them.

In order to help small or growing companies to integrate robotics in their processes, custom-made robots are a solution, this are more accessible, thus helping them to enhance their development. Despite having a lower cost, custom-made robots must comply with safety standards that guarantee operators and machinery safety within the facilities where they are integrated. The purpose of this work is to design and implement an industrial safety system in a Delta Lineal robot, located in the Process Automation Center, at the Pontificia Javeriana Cali University. This robot will follow some of the guidelines and standards established by international regulatory entities such as ISO, IEC and ANSI RIA. It is expected that by integrating this safety system to the robot, it will be able to maintain its functionality and at the same time guarantee the safety of the operators and itself.

**Keywords:** Delta Linear Robot, HMI, Security devices, Requirements.



# Índice general

<b>1. Introducción</b>	<b>1</b>
<b>2. Planteamiento del problema</b>	<b>3</b>
<b>3. Justificación</b>	<b>5</b>
<b>4. Objetivos</b>	<b>7</b>
4.1. Objetivo General	7
4.2. Objetivos Específicos	7
<b>5. Marco de referencia</b>	<b>9</b>
5.1. Áreas Temáticas	9
5.2. Marco Teórico	9
5.2.1. Conceptos	9
5.2.2. Teoría	11
5.3. Trabajos Relacionados	13
<b>6. Requisitos del sistema</b>	<b>15</b>
6.1. Estado de software	15
6.2. Estado de hardware	15
6.3. Problemas encontrados	16
6.3.1. Puesta a punto	16
6.4. Matriz de evaluación de riesgos	17
6.5. Definición de requerimientos del sistema	19
6.6. Requerimientos específicos del Teach pendant	24
<b>7. Diseño de la arquitectura de control</b>	<b>25</b>
7.1. Diseño de software	25
7.1.1. Modo Manual baja velocidad	25
7.1.2. Modo Manual alta velocidad	26
7.1.3. Modo Automático	27
7.1.4. Modo Remoto	27
7.1.5. Estado de seguridad	28
7.1.6. Programación del PLC	28
7.2. Diseño de hardware	28
7.2.1. Guarda de encierro	28
7.2.2. Teach Pendant	29
7.2.3. Parada de emergencia	30

7.3. Integración de la arquitectura . . . . .	31
<b>8. Implementación del sistema de control</b>	<b>33</b>
8.1. Implementación de software . . . . .	33
8.1.1. Interfaz gráfica de usuario (GUI) . . . . .	33
8.1.2. Modo Manual baja velocidad . . . . .	36
8.1.3. Modo Manual alta velocidad . . . . .	40
8.1.4. Modo Automático . . . . .	43
8.1.5. Modo Remoto . . . . .	45
8.1.6. Estado de seguridad . . . . .	48
8.1.7. Programación del PLC . . . . .	49
8.2. Implementación de hardware . . . . .	61
8.2.1. Guarda de encierro . . . . .	61
8.2.2. Teach Pendant . . . . .	63
8.2.3. Parada de emergencia . . . . .	64
8.3. Implementación de la arquitectura . . . . .	65
<b>9. Resultados y Discusión</b>	<b>71</b>
9.1. Evaluación del Cumplimiento de Requisitos y Medidas de Protección . . . . .	71
9.2. Pruebas . . . . .	71
<b>10. Conclusiones</b>	<b>73</b>
<b>11. Recomendaciones</b>	<b>75</b>
<b>Bibliografía</b>	<b>77</b>
<b>12. Anexos</b>	<b>81</b>
<b>Anexos</b>	<b>81</b>
<b>A. Diseño del software de la arquitectura de control</b>	<b>83</b>
A.1. Modo Manual Baja Velocidad . . . . .	83
A.2. Modo Manual Alta Velocidad . . . . .	84
A.3. Modo Automático . . . . .	85
A.4. Modo Remoto . . . . .	86
A.5. Estado de Seguridad . . . . .	87
A.6. Programación del PLC . . . . .	88
<b>B. Diseño del hardware de la arquitectura de control</b>	<b>89</b>
B.1. Guarda de encierro . . . . .	89
B.2. Componentes del Teach Pendant . . . . .	92
B.3. Diseño del Teach Pendant . . . . .	93

---

B.4. Parada de emergencia interna . . . . .	95
<b>C. Implementación del hardware de la arquitectura de control</b>	<b>99</b>
C.1. Guarda de encierro . . . . .	99
C.2. Teach Pendant . . . . .	102
C.3. Parada de emergencia interna . . . . .	105



# Índice de figuras

5.1. Configuración robot delta lineal tipo Keops o inclinado	9
6.1. Patín de tensión instalado en la mitad de los rieles	17
7.1. Diseño de estructura ensamblada del teach pendant con componentes	31
8.1. Interfaz gráfica de usuario (GUI) sección izquierda	34
8.2. Interfaz gráfica modo Manual Baja velocidad (Control ejes individuales)	36
8.3. Interfaz gráfica modo Manual Baja velocidad (Control por teclado)	38
8.4. Teclado utilizado para el control	39
8.5. Interfaz gráfica modo Manual alta velocidad (Control ejes individuales)	40
8.6. Interfaz gráfica modo Manual alta velocidad (Control por posición)	41
8.7. Interfaz gráfica modo Automático (ENSEÑAR POSICIONES)	43
8.8. Interfaz gráfica modo Automatico (EJECUTAR RUTINA)	44
8.9. Interfaz gráfica modo Remoto (Visualización)	45
8.10. Interfaz gráfica modo Remoto (Control por posición)	46
8.11. Interfaz gráfica modo Manual alta velocidad (por eje)	47
8.12. Interfaz gráfica Estado de seguridad	49
8.13. Código de la función DO_HOMING	50
8.14. Código de la función DO_MOVE	51
8.15. Código de la función DO_RESET	52
8.16. Código de la función DO_POWER	53
8.17. Código de la función DO_SLOW_STOP	54
8.18. Código de la función MONITOREO_SWITCH	55
8.19. Código de la función MONITOREO_PUERTA	57
8.20. Código de la función BALIZA	59
8.21. Código principal del PLC	60
8.22. Proceso de instalación de guardas de seguridad	62
8.23. Elementos de interacción en el panel electrico	62
8.24. Teach pendant completo	63
8.25. Diagrama de conexiones pendant	64
8.26. Botones de emergencia instalados en parte exterior, teach pendant y dentro de zona de peligro, respectivamente.	64
8.27. Diagrama de integración de la arquitectura	66
A.1. Diagrama de flujo de modo <i>Manual Low</i>	83
A.2. Diagrama de flujo de modo <i>Manual High</i>	84
A.3. Diagrama de flujo de modo <i>Automático</i>	85

A.4. Diagrama de flujo de modo <i>Remoto</i>	86
A.5. Diagrama de flujo de estado de seguridad	87
A.6. Diagrama de máquina de estados	88
B.1. Diseño de las guardas de seguridad	89
B.2. Diseño de las guardas de seguridad (frontal)	90
B.3. Diseño de las guardas de seguridad (izquierda)	90
B.4. Diseño de las guardas de seguridad (completa)	91
B.5. Diseño de la puerta de acceso a la celda	91
B.7. Diseño de caja de la estructura del teach pendant	93
B.6. Diseño de estructura ensamblada del teach pendant	94
B.8. Diseño de tapa la estructura del teach pendant	94
B.9. Diseño de estructura ensamblada del teach pendant con componentes	94
B.10. Diseño de estructura ensamblada para instalación de botón de emergencia interno	95
B.11. Diseño de caja para instalación de botón de emergencia interno (vista 1)	96
B.12. Diseño de caja para instalación de botón de emergencia interno (vista 2)	96
B.13. Diseño de tapa para instalación de botón de emergencia interno (vista 1)	97
B.14. Diseño de tapa para instalación de botón de emergencia interno (vista 2)	97
C.1. Celda robótica con las guardas y puerta implementadas (vista 1)	99
C.2. Celda robótica con las guardas y puerta implementadas (vista 2)	100
C.3. Celda robótica con las guardas y puerta implementadas (vista 3)	101
C.4. Estructura ensamblada del teach pendant	102
C.5. Caja para la estructura del teach pendant	102
C.6. Tapa para la estructura del teach pendant	103
C.7. Estructura con componentes del teach pendant	103
C.8. Teach pendant completo	104
C.9. Estructura de botón de emergencia interno ensamblada	105
C.10. Caja para la estructura de botón de emergencia interno (vista 1)	106
C.11. Caja para la estructura de botón de emergencia interno (vista 2)	106
C.12. Tapa para la estructura de botón de emergencia interno (vista 1)	107
C.13. Tapa para la estructura de botón de emergencia interno (vista 2)	107

# Índice de cuadros

6.1. Matriz de riesgo realizada al robot delta	18
6.2. Parte 1 de la tabla de requerimientos de la parte 1 de la norma ANSI/RIA R15.06	20
6.3. Parte 2 de la tabla de requerimientos de la parte 1 de la norma ANSI/RIA R15.06	21
6.4. Parte 1 de la tabla de requerimientos de la parte 2 de la norma ANSI/RIA R15.06	22
6.5. Parte 2 de la tabla de requerimientos de la parte 2 de la norma ANSI/RIA R15.06	23
6.6. Parte 3 de la tabla de requerimientos de la parte 2 de la norma ANSI/RIA R15.06	24
7.7. Tabla de especificaciones pendant	30
8.8. Colores de la baliza dependiendo de la entrada IN	59
8.9. Descripción de variables principales	61
9.10. Resultado de la evaluación realizada al sistema terminado	72
B.11. Comparación de dispositivos para selección del Computador	92
B.12. Comparación de dispositivos para selección de la Pantalla	92
B.13. Comparación de dispositivos para selección del Switch de seguridad	93



# Introducción

---

El uso de robots en los procesos de producción tiene una extensa trayectoria a nivel mundial. La introducción de la robótica industrial se dio a conocer en la década de los 30's, con la creación de los primeros dispositivos con la capacidad de realizar operaciones de pick & place. Estos, fueron presentados en la feria Anual de Nueva York en 1939, para su uso industrial, y de ocio, creando una de las industrias más grandes actualmente. Su desarrollo tuvo un despegue significativo en la Segunda Guerra Mundial, donde se emplearon autómatas de gran tamaño para tareas repetitivas de movimientos simples y cargas pesadas. Con la gran acogida de todo tipo de industrias por incorporar dispositivos mecatrónicos en sus procesos en la década de los 60's, estos empezaron a ser utilizados en grandes compañías. Se caracterizaron por programarse para realizar tareas que eran consideradas peligrosas para ser hechas de forma manual, como el manejo y templado de piezas metálicas a altas temperaturas [1].

A medida que el uso de la robótica en la industria creció, surgieron nuevas necesidades y desafíos gracias a la evolución de las tecnologías. Estos autómatas llegan a la industria de la mano con la industria 4.0, lo cual fusiona en un mismo ambiente laboral, una gran cantidad de dispositivos automatizados con operarios, que ha duplicado su número en los últimos diez años, todo esto con el objetivo de combinar las habilidades humanas y las capacidades de los robots, permitiendo así resolver en la industria aplicaciones que necesitarían maquinarias más complejas, caras y peligrosas. En 2011, la Organización Internacional de Estandarización (ISO) y otros organismos, ven necesario normalizar la seguridad de los operarios y todo quien interactúe con estos sistemas automatizados, para mitigar o minimizar el riesgo de un accidente, publicando así la norma ISO 10218 la cual especifica los requisitos de seguridad para robots industriales y la norma ISO/TS 15066 en el año 2015, la cual especifica los requisitos de seguridad adicionales para los "co-bots" y de asistencia, logrando encontrar versiones nacionales como la norma ANSI/RIA R 15.06-199, siendo esta la versión estadounidense de las ISO [2].

A pesar de las normas y estándares que fueron publicados, estos son difíciles de cumplir, en la implementación de sistemas robóticos a la medida, cuya demanda ha aumentado en los últimos años a causa de los altos precios los dispositivos en el mercado, es por esto, que se encuentran artículos, como el realizado por un grupo de investigadores de universidades de India, Bangladesh, Malasya y Arabia Saudí y publicado por Handiwi, editorial académica francesa, el cual se centra en el análisis de sistemas de seguridad diseñados e implementados en entornos de robótica industrial, con el objetivo de facilitar un trabajo seguro en conjunto con humanos. Además, se examinan las regulaciones y los conceptos introducidos en este ámbito hasta el año 2021, cuando este fue publicado.

La discusión abarca enfoques multidisciplinarios, como la estimación y evaluación de lesiones por colisiones con humanos, el desarrollo de equipos mecánicos y software para minimizar impactos, sistemas de detección de colisiones y estrategias de prevención, complementando su propuesta con un enfoque para la seguridad humana con robots en la Industria 4.0, utilizando redes definidas por software y CPS (sistemas ciberfísicos), los cuales están compuestos de los sistemas físicos que realizan algún proceso o función de interés, y los gemelos digitales que los modelan su comportamiento [3].

A medida que avanza la tecnología a un ritmo acelerado, se espera que la integración de la robótica en la industria siga en aumento considerable, como se ha venido dando en los últimos años. Este crecimiento no sólo implica una transformación significativa en los procesos productivos sino que también crea nuevos desafíos y oportunidades en eficiencia y productividad. Garantizar un entorno de trabajo seguro, se convierte en una prioridad a medida que las máquinas automatizadas se convierten en una parte esencial de la fuerza laboral. Por lo tanto, la colaboración y trabajo conjunto entre empresarios, científicos, ingenieros y responsables políticos es esencial para abordar estos desafíos y maximizar el potencial que ofrecen los robots en los futuros procesos de fabricación.

# Planteamiento del problema

---

Los procesos industriales le han brindado al ser humano la posibilidad de elaborar productos con alta calidad y de forma veloz, lo cual le genera mayores ganancias a las empresas, esto se ha logrado gracias a la revolución industrial y a la evolución tecnológica, siendo la robótica un claro ejemplo de esta innovación [4]. A pesar de sus beneficios, la adopción de esta tecnología se ve limitada por una variedad de aspectos en países como Colombia, donde no solo los elevados costos de implementación son un obstáculo, también la producción de bienes de bajo valor agregado y la superioridad de industrias que no dependen de la tecnología avanzada lo son. Las Pymes (pequeñas y medianas empresas), son vitales para la economía colombiana y estas encuentran en la robótica una herramienta para añadir valor a sus productos y entrar a competir en el mercado internacional. Sin embargo, los altos costos de los robots comerciales y la infraestructura para su implementación son grandes desafíos que afrontar.

Una solución esperanzadora es el desarrollo de robots a la medida, donde se busca cumplir con sus necesidades específicas, teniendo en cuenta las capacidades económicas de estas empresas. A pesar de que esta alternativa mitiga los costos y promueve el avance tecnológico, también genera un nuevo desafío relacionado con la seguridad. Los robots diseñados en el ámbito académico o de investigación, generalmente se centran en sus características de funcionamiento, como mayor precisión y versatilidad, pero carecen de las medidas de seguridad necesarias para su implementación en entornos industriales, lo cual limita su viabilidad en el mercado. El cumplimiento de las normativas internacionales como ISO 10218, ISO/TS 15066:2016 y especialmente la ANSI/RIA R15.06 es crucial para el desarrollo de robots con características que solucionen las necesidades de la empresa, que sean asequibles y sobre todo seguros para los operarios.

Por lo tanto, el problema que se abordó con este proyecto es la identificación, análisis y mitigación de los riesgos relacionados con el funcionamiento del robot delta lineal del Centro de Automatización de Procesos de la Universidad Javeriana de Cali, resolviendo la pregunta de investigación ¿Cómo los robots diseñados en entornos académicos o de investigación y desarrollo podrían cumplir con los estándares de la normativa ANSI/RIA R15.06 y otras normativas internacionales de seguridad, para así aumentar su viabilidad en entornos industriales? Buscamos que el robot cumpla con los requerimientos de seguridad industrial establecidos en la normativa estadounidense ANSI/RIA R15.06 [5]. Dando como resultado un robot que cumpla con los requisitos técnicos y económicos para su viabilidad en la industria, pero que también garantice que la integridad de su estructura y de los operarios no será puesta en peligro. Este proyecto establecerá un precedente en la implementación de robótica segura, así como resaltará la importancia de la seguridad en el proceso de diseño de robots

en entornos académicos y de investigación, contribuyendo con la adopción de soluciones robóticas personalizadas en la industria del país.

# Justificación

---

En países en desarrollo, es frecuente encontrar que las empresas tienden a adquirir y utilizar maquinaria que no incorpora las últimas tecnologías, que les funciona bien para su producción, pero que no les permite ser competitivos contra industrias que cuentan con procesos altamente automatizados. Un país con empresas que no se destacan en innovación y eficiencia puede enfrentar varias desventajas, entre estas se encuentran la falta de competitividad internacional, productividad limitada, dependencia de bienes y servicios extranjeros, que lo llevan a un estancamiento económico. La actualización de una empresa en términos de tecnología, para adaptarse a la demanda y competencia actual, va ligada a una inversión económicamente razonable, que pueda ser pagada por la compañía, conllevando esto a una sola salida: el desarrollo de tecnología hecha a la medida, siendo esta una opción económicamente viable, aclarando que se deben tener en cuenta ciertos aspectos como la normativa internacional que rige la operación de máquinas dentro de la industria.

Para un robot estar en operación dentro de una planta, se debe tener en cuenta la implementación de un sistema de seguridad industrial, que garantice su uso de forma segura y sin perder calidad. Por otro lado, esto tiene un trasfondo, pues la falta de aplicaciones de robótica implementada en la educación universitaria genera que los estudiantes salgan al mundo laboral conociendo el manejo de las diferentes tecnologías, pero no con la capacidad de diseñarlas de forma correcta, lo cual dificulta la automatización en las empresas que no puedan costear una solución comercial. Debido a lo anterior, el desarrollo e implementación de un sistema de seguridad en el robot delta lineal del *CAP*, tendrá un impacto positivo en la tecnificación de empresas colombianas, brindándoles una opción viable de automatización a un menor costo, posicionando en el mercado robots a la medida que cumplan con los estándares de seguridad y dándole a los jóvenes universitarios una guía para el desarrollo de tecnologías robóticas de forma correcta.

En el año 2018, el 62% las *PYMES* en Colombia no tenían acceso a financiamiento, cifra que incrementó con la crisis económica generada por la llegada del COVID-19. Es por esta misma razón que cerca de la mitad de ellas quiebra después del primer año y solo 20% sobrevive al tercero [6]. Sin la capacidad inversora necesaria, es imposible la adquisición de la tecnología de vanguardia mencionada anteriormente, para ser competitiva en mercados internacionales. Esta situación ha creado una necesidad que, hasta ahora, no ha sido llevada a cabo de una manera simple por el requisito de cumplimiento de normas y estándares, que garanticen la seguridad de sus colaboradores. En esta industria, existen todo tipo de riesgos y accidentes, en los que se destacan accidentes por impacto o colisión, por aplastamiento y atrapamiento, por fugas y, de las piezas mecánicas que resultan de la avería de los componentes para el accionamiento del robot. Esto ha llevado a las

entidades responsables, a ser cada vez más estrictas, para minimizar el riesgo de cualquier tipo de estos accidentes [7]. Finalmente, también se presenta la problemática de los profesionales recién egresados, y estudiantes próximos a graduarse, pues en Colombia, tan solo el 30 % de jóvenes entre los 25 y 34 años cuenta con educación superior, con un pequeño porcentaje de estos en carreras profesionales de ingeniería, de los que la gran mayoría de estos, por falta de aplicaciones robóticas implementadas en sus respectivas instituciones, manejan la robótica, pero no saben desarrollarla. Sin dejar a un lado los investigadores, que en Colombia, tan solo el 2.5 % de ellos trabaja en la industria, mientras el 95.7 % se queda en la academia, frenando aún más el desarrollo de la robótica en el área laboral [8].

En conclusión, con este proyecto se busca dar un valor agregado a las empresas colombianas y a sus productos, permitiéndose crecer para entrar a competir en el mercado nacional e internacional. Además, se podrían posicionar en el mercado nacional robots hechos a la medida para las *PYMES* e incluso empresas grandes que deseen comenzar a incluir la robótica en sus procesos, sin realizar una inversión inicial excesiva. En adición a lo anterior, se impulsaría el desarrollo de tecnología a nivel nacional, con el objetivo de diversificar la economía de Colombia, y pasar de tener una economía basada en exportación de materia prima a una economía basada en la comercialización tecnológica.

# Objetivos

---

## 4.1. Objetivo General

Desarrollar el sistema de control de un robot delta lineal basado en la norma ANSI/RIA R15.06-2012.

## 4.2. Objetivos Específicos

- Especificar los requisitos de hardware y software pertinentes para que el robot pueda ser considerado seguro en la industria, basado en la norma ANSI/RIA R15.06-2012.
- Diseñar la arquitectura del sistema de control en hardware y software que dé cumplimiento a los requisitos planteados.
- Implementar el sistema de control.
- Evaluar el funcionamiento del sistema de control.



# Marco de referencia

---

## 5.1. Áreas Temáticas

- Robotics and automation - Robots - Robot control
- Robotics and automation - Robots - Robot programming
- Engineering management - management - Security management

## 5.2. Marco Teórico

### 5.2.1. Conceptos

- **Robot Delta lineal**

El robot delta lineal es un tipo de robot paralelo, que se caracteriza gracias a su base móvil, que siempre se desplaza en planos paralelos a sus guías, este está conformado por un conjunto de tres bielas, en la que cada una de estas cuenta con un sistema de actuación independiente, que cumplen la función de unir la base. La familia de robots delta lineal, se constituye de diferentes tipos de estructura, siendo las más comunes la estructura horizontal, vertical e inclinada, también conocida como Keops [\[9\]](#).



Figura 5.1: Configuración robot delta lineal tipo Keops o inclinado

El robot delta lineal tipo Keops, cuenta con ciertas características, que son claves a la hora de la evaluación de posibles soluciones para una aplicación industrial, como también algunas propiedades y casos en los que se destaca su uso donde se encuentra [\[10\]](#):

- Aplicaciones de cargas livianas y cortas distancias por su velocidad y flexibilidad de movimiento.
- Motores en bases fijas, que disminuyen el desgaste de cables y componentes, dándoles a estos una mayor vida útil y una necesidad de mantenimiento en mayor tiempo.
- Ideal para aplicaciones de Pick and Place a nivel industrial y de enseñanza

#### ■ Sistema de seguridad en robótica

La robótica experimenta un crecimiento notable debido a su eficiencia y precisión en los procesos. Estas características se deben a sus actuadores, tales como motores, sensores e infraestructura sólida. Sin embargo, estos elementos también plantean un problema potencial. Sin un control o limitación adecuados, podrían ocasionar daños tanto al propio sistema como al usuario que lo controle o se encuentre en su área de trabajo. Es por esta razón que surge el concepto de sistema de seguridad en la robótica [11].

El sistema de seguridad de un robot debe asegurar que el dispositivo se detenga en ciertas circunstancias, como un golpe a un operario, una colisión con un objeto o un fallo mecánico. En este contexto, es necesario definir criterios de seguridad, tales como limitaciones en el alcance del robot, para salvaguardar su estructura y prevenir situaciones que puedan generar una carga excesiva sobre los motores que lo impulsan. También, se deben establecer limitaciones en la velocidad de sus movimientos; en caso de que el robot trabaje a una velocidad muy alta, podría generar una gran inercia y provocar daños significativos en caso de colisión. Por esta razón, al limitar la velocidad de movimiento, se protege a los operarios que se encuentren en el área de trabajo del robot. De igual manera, la restricción de la intensidad de corriente contribuye a prevenir fallas técnicas en los motores [11].

En este contexto, es esencial establecer criterios de seguridad, como limitaciones en el alcance del robot, para salvaguardar su integridad y evitar situaciones que puedan ejercer una carga excesiva sobre los motores que lo impulsan. Además, resulta crucial imponer restricciones en la velocidad de sus movimientos; una velocidad demasiado alta podría generar una inercia considerable, provocando daños significativos en caso de colisión. Al limitar la velocidad de movimiento, se protege a los operarios presentes en el área de trabajo del robot.

#### ■ Interfaz Humano máquina

En cualquier línea de producción o planta industrial se debe tener control sobre ciertas variables específicas de un proceso para garantizar la calidad de la producción. La supervisión de la maquinaria que realiza los procesos y también de sus variables es realizada por los operarios de la planta y su método de interacción con la máquina es por medio de las Interfaces Humano Máquina (HMI). El HMI muestra estadísticas y métricas sobre los procesos que la máquina está realizando, por ejemplo, variables como la temperatura, humedad, o PH. Este dispositivo cuenta con una pantalla, esta varía sus características dependiendo de la gama a

la que pertenezca, puede ser un instrumento táctil o no, así como puede ser blanco y negro o a color. También, pueden poseer uno o varios métodos de intercambio de información, esto es de crucial importancia, ya que estos deben ser compatibles con el protocolo de comunicación utilizado por el PLC que lo controla [12].

En un proceso donde se utilice la robótica, la HMI muestrea variables de interés para el operario, como corriente, posición del punto central de la herramienta (TCP) o velocidad de funcionamiento. Además, se puede controlar estas variables según el criterio deseado, así como se deben limitar por seguridad para garantizar que no se vea afectada la integridad del personal adyacente y de la misma máquina. Así mismo, se logra controlar el modo de funcionamiento, generalmente pasando de parado, automático o manual. [13].

### 5.2.2. Teoría

#### ■ Estándar ISO 12100

ISO 12100 es una norma desarrollada por la Organización Internacional de Normalización (ISO) que proporciona directrices y principios generales para el diseño y la evaluación de riesgos de la maquinaria. El propósito de esta norma es garantizar la seguridad de los equipos a lo largo de todo su ciclo de vida, desde el concepto y el diseño hasta la eliminación.

El estándar ofrece un marco de trabajo para que los fabricantes de maquinaria identifiquen y evalúen los posibles riesgos asociados con sus productos y para que implementen las medidas adecuadas de reducción de riesgos. La norma abarca una amplia gama de maquinarias, desde maquinarias industriales hasta productos de consumo y vehículos. Esta es una herramienta importante para garantizar la seguridad de la maquinaria y para ayudar a los fabricantes a cumplir con los requisitos reglamentarios y de seguridad aplicables a cada caso. [14]

#### ■ Estándar ISO 14120

En el campo de la industria, no solo es importante garantizar el buen funcionamiento de la maquinaria y que estas ejecuten el proceso de manera satisfactoria, sino que también se debe asegurar que los operarios estén seguros y protegidos de cualquier riesgo que se genere por el uso de la máquina. El estándar *ISO 14120* es un estándar de la ISO (International Organization for Standardization). Este está clasificado como tipo B2, lo cual indica que trata sobre un aspecto de la seguridad o sobre uno o varios tipos de protecciones que se pueden utilizar en una amplia variedad de máquinas, específicamente dispositivos de seguridad como mandos manuales y equipos electro sensitivos [15].

Este estándar establece los requisitos generales para el diseño y construcción de guardas fijas y móviles en maquinaria industrial, centrándose específicamente en los dispositivos o materiales empleados para el resguardo de zonas peligrosas destinadas a los operarios [16].

La norma *ISO 14120* define características para el diseño y la construcción de protecciones de maquinaria como:

- Resistencia.
- Resistencia mecánica.
- Estabilidad.
- Accesibilidad.
- Visibilidad.
- Protección contra los peligros. [17]

#### ■ Estándar ISO 10218

La ISO, es una federación internacional que se encarga de establecer los estándares que toda empresa debe seguir para poner en el mercado un producto. El estándar *ISO 10218*, está bajo el tema *Robots and robotic devices — Safety requirements for industrial robots* y está compuesto por dos partes, la primera parte abarca el tema de robots, mientras que la segunda parte, trata sobre la integración del robot y sus sistemas. El estándar *ISO 10218* fue creado para reconocer los peligros que se presentan por el uso de robots industriales y sus sistemas, este es un estándar tipo C, que define los requerimientos de seguridad para una máquina en particular o un grupo de máquinas, en este caso se aplican a cualquier robot. Este estándar se divide en dos partes, donde la primera parte establece una guía para el diseño y construcción de un robot de forma segura, mientras que la segunda parte establece una guía para la seguridad de los operarios durante la integración, pruebas, funcionamiento y mantenimiento de un robot. [18]

#### ■ Estándar ANSI/RIA R 15.06-2012

La norma ANSI R15.06-1999, brinda pautas para la fabricación, reconstrucción e integración de robots industriales y sistemas robóticos para mejorar la seguridad del personal asociado con el uso y manipulación de los mismos, mediante distintos métodos ligados, como la evaluación de riesgos, con el fin de la implementación segura de la robótica colaborativa, la cual se define como un sistema robótico operado automáticamente que comparte el mismo espacio de trabajo con un humano.

Este estándar se encarga de repartir las responsabilidades de seguridad, resaltando que estas no solo dependen de un organismo. Para garantizar la integridad del usuario con los robots colaborativos, tanto el fabricante, los integradores, los instaladores y los usuarios finales deben comprometerse y cumplir con normas específicas. Se cuentan con 14 cláusulas que se deben poner en práctica para garantizar el menor riesgo posible, las cuales se nombran a continuación: [19]:

- Cláusula 1 - Ámbito, finalidad, aplicación y exclusiones.
- Cláusula 2 - Referencias normativas.

- Cláusula 3 - Definiciones.
- Cláusula 4 - Fabricación, re-manufacturación y reconstrucción de robots.
- Cláusula 5 - Requisitos de funcionamiento de los dispositivos de protección.
- Cláusula 6 - Instalación de robots y sistemas robóticos
- Cláusula 7 - Introducción a protección del personal
- Cláusula 8 - Protección del personal - Método prescrito.
- Cláusula 9 - Protección del personal - Método de evaluación de riesgos.
- Cláusula 10 - Implementación de protección del personal.
- Cláusula 11 - Dispositivos de protección - requisitos de aplicación.
- Cláusula 12 - Mantenimiento de robots y sistemas robóticos.
- Cláusula 13 - Pruebas y puesta en marcha de robots y sistemas robóticos.
- Cláusula 14 - Formación en seguridad del personal[19].

### 5.3. Trabajos Relacionados

El uso cada vez mayor de robots para realizar tareas junto con humanos plantea nuevos problemas de salud y seguridad en el trabajo. Será el nuevo lugar de trabajo del siglo XXI en el que la robótica profesional jugará un papel cada vez más importante. El artículo "*Working safely with robot workers: Recommendations for the new workplace*", publicado en la revista de *Higiene Ocupacional y Ambiental*, describe la creciente sofisticación de los robots y proporciona un conjunto de recomendaciones para prácticas seguras de robots profesionales. [20]

El desarrollo de sistemas de robótica industrial a medida es un problema complejo que suele resolverse mediante soluciones modulares, pero las plataformas tradicionales no aportan soluciones lo suficientemente flexibles para adaptarse a los entornos industriales. El capítulo escrito por Juan David Contreras, publicado en el libro *Avances en Investigación en Automatización y Robótica*, describe un método para implementar el control de movimiento de sistemas robóticos utilizando un PLC compatible con *PLCopen Motion*, donde se logra un alto nivel de modularidad en el software mientras se crea una solución independiente del hardware. Finalmente, el software de control se implementa en un robot delta lineal para demostrar las ventajas del enfoque. [21].

En otra perspectiva, el despliegue de sistemas robóticos presenta varios desafíos, siendo la seguridad de suma importancia debido a que estos robots comparten su entorno con los humanos hasta cierto punto. En el libro *Fundamentos de robótica*, específicamente en el capítulo escrito por Bruno Benzile, se brinda una descripción general de algunos estándares relevantes para los sistemas robóticos, los cuales están principalmente relacionados con su alcance y son emitidos por diversas organizaciones. Estos estándares, junto con otros documentos como las especificaciones técnicas, son cruciales para llevar a cabo la evaluación de riesgos de un nuevo sistema y la mitigación de

los peligros identificados, dos pasos críticos en el despliegue de células robóticas y manipuladores móviles. Además, en el área de los robots colaborativos (cobots), la seguridad de los operadores humanos se vuelve aún más crítica, considerando la proximidad intrínseca que se tiene con estos [22].

El artículo titulado *Emerging research fields in safety and ergonomics in industrial collaborative robotics: A systematic literature review* realiza una investigación exhaustiva sobre los avances actuales en seguridad y ergonomía en la robótica colaborativa industrial. Destaca la creciente relevancia de la colaboración humano-robot en el contexto de la Industria 4.0, que está transformando los entornos laborales en la industria. Los autores resaltan la importancia de asegurar la salud y seguridad ocupacional, centrándose en la implementación de la robótica colaborativa, enfatizando en la necesidad de diseñar celdas de trabajo que sean seguras y ergonómicas. A través de una revisión sistemática de la literatura científica reciente, el artículo identifica los principales temas de investigación y los desafíos emergentes en este campo, el cual resalta, que se encuentra en auge y constante evolución. [23]

# Requisitos del sistema

---

Antes de iniciar el proyecto, el robot delta lineal requería una inspección, tanto de condiciones y funciones de software como de hardware, necesario para determinar el estado actual del mismo, y poder identificar las posibles problemáticas con las que este cuenta, que impidieran o intervinieran en la aplicación de los requerimientos que expone la norma *ANSI RIA 15.06* [19].

## 6.1. Estado de software

El sistema robótico contaba con dos programas para su control. El primer programa utiliza el lenguaje de Texto Estructurado, el cual se emplea en la elaboración de scripts utilizados en un PLC. Para la realización y carga de este programa al PLC implementado en el robot, se utilizó la herramienta Connected Components Workbench (CCW). Este software contiene las rutinas para el movimiento del robot, haciendo uso de la función de Motion Control para el movimiento de los motores mediante sus respectivos drivers. Con esta función, se define la posición a la que debe moverse cada motor, además de indicar la velocidad y aceleración del movimiento. Por otro lado, la función Motion Control requiere la configuración previa de los ejes que se utilizarán. En esta fase, se definen parámetros del robot, como la velocidad del homing, dirección del homing y el uso de sensores de límite. Además, el programa establece otras funciones para el robot, como encendido/apagado, realización del homing, rutina de parada lenta y rutina de parada de emergencia.

El segundo programa utiliza el lenguaje de programación Python, que posee la capacidad de realizar operaciones matemáticas más complejas. Su amplia variedad de librerías le otorga una gran versatilidad. En este programa se llevan a cabo las operaciones de cinemática inversa y directa, calculando los movimientos necesarios para que el robot alcance las coordenadas X, Y, Z indicadas por el usuario. Estos parámetros se envían al PLC a través de la red a la que está conectado. Adicionalmente, el programa cuenta con una interfaz gráfica de usuario (GUI) para las funciones de encendido/apagado, homing y movimiento del robot. Esta revisión elimina algunas redundancias y ajusta la estructura para mejorar la claridad del texto.

## 6.2. Estado de hardware

El sistema robótico requiere ciertos elementos para los movimientos mecánicos del robot, entre ellos se encuentra el PLC, el motor de cada eje y su respectivo driver, quienes trabajan en conjunto para dirigir los movimientos del mismo. Estos movimientos son ordenados por el usuario mediante

el computador y transmitidos por el sistema mecánico conformado por deslizadores, rieles y correas. Adicionalmente, el sistema robótico cuenta con dispositivos cuyo propósito es facilitar al operario el manejo del robot, como el HMI y el Pendant para manejarlo de forma manual. No obstante, ninguno de estos sistemas se encontraba en funcionamiento ni cumplía con los requerimientos de seguridad, ya que solo contaban con su instalación en el panel.

Por otro lado, el sistema robótico, cuenta con un botón de paro de emergencia de fácil alcance para el usuario, indicadores de alimentación de voltaje de los tres motores, y botón de encendido del sistema.

### 6.3. Problemas encontrados

El estado del hardware y software en el que se recibió el sistema robótico presentaba problemas que impedían la implementación del sistema de seguridad. Entre estos problemas se encontraba el hecho de que el modelo no hacía uso del pendant y HMI integrados en el sistema, lo que resultaba en un desaprovechamiento de estos componentes. Además, el pendant no era adecuado para el control del robot, ya que contaba con un selector de tres posiciones para la selección del eje que el usuario deseaba mover y dos botones, permitiendo el movimiento en un solo eje. Como consecuencia, los parámetros importantes del robot, como la velocidad y la aceleración en tiempo real, no eran mostrados al usuario. Esto se agrava al no cumplir con los requerimientos de la norma [19], destacando la ausencia de un botón de paro de emergencia desde el Teach pendant y la falta de una pantalla de interfaz que impida funciones avanzadas, como el aprendizaje de rutinas, la inserción de puntos específicos a los cuales el usuario desea dirigir el robot y la selección de modos de operación.

Por otro lado, el robot no contaba con los brazos en su funcionamiento. Al ensamblarlos, el robot presentaba oscilaciones en sus movimientos laterales y aún más fuertes en el movimiento de bajada. Estas oscilaciones llegaban a un punto donde los brazos se desprendían de los deslizadores en el riel y se presentaban desajustes mecánicos de las piezas mecánicas cerca al área, por la alta vibración presentada. Además, se identificó poca tensión en las correas de transmisión de movimiento de los tres brazos, y valores desacertados de los constantes los controladores tipo proporcional-integral (PI) encontrados en los drivers de los motores.

#### 6.3.1. Puesta a punto

Iniciando con la solución de uno de los problemas encontrados, y mencionados en la sección anterior se empezó vinculando la HMI a la red inalámbrica propia del robot, con el objetivo de poder programar indicadores, pantallas y botones, y así darle funcionalidad. La programación de este dispositivo se realizó desde la herramienta *Connected Components Workbench* (CCW). Para darle funcionalidad al pendant, se le agregó la funcionalidad de control manual en el programa de Python, de esta forma se lograron hacer movimientos dirigidos por el usuario en los tres ejes.

Finalmente, las vibraciones en el robot se solucionaron variando el parámetro de ganancia proporcional del control de posición ( $P-09$ ). Se tomó el manual de los drivers que controlan los motores, los cuales son de referencia GXK SD300 [24], de esta forma se encontró que entre mayor sea el valor de este parámetro, mayor sería la rigidez y menor el retraso en la posición, pero a la vez, si este valor es muy alto causa oscilaciones o un sobre pico (*overshoot*) en el control de posición. Por esta razón, se procedió a disminuir paulatinamente el valor de esta ganancia y tensionar las correas de transmisión de fuerza con ayuda de patines de tensión en mitad de los rieles, como se muestra en la imagen 6.1, permitiendo disminuir la deflexión de las mismas, pudiendo solucionar el desajuste mecánico que se estaba presentando, el daño y desacople de piezas móviles del mismo.



Figura 6.1: Patín de tensión instalado en la mitad de los rieles

## 6.4. Matriz de evaluación de riesgos

Teniendo el sistema robótico a punto, se procede a realizar la matriz de evaluación de riesgos, con el objetivo de identificar los peligros asociados al encendido, operación de movimiento automático, dirigido y general, como también el mantenimiento mecánico y eléctrico del robot. La tabla a realizar, tiene en cuenta la severidad del riesgo, como también la frecuencia en la que el operario se expone a cada uno de estos, los controles de ingeniería y administrativos que se realizan y los elementos de protección personal que se aplican actualmente a cada riesgo identificado, con la intención de mitigarlo. Esta se puede ver a continuación en el cuadro 6.1.



## 6.5. Definición de requerimientos del sistema

Los requerimientos que debe cumplir el sistema robótico son de gran importancia para garantizar que sea seguro. Para la elaboración de estos requerimientos se utilizó como base la norma ANSI/RIA R15.06, la cual fue desarrollada por la American National Standards Institute (ANSI), en colaboración con la Robotic Industries Association (RIA). Esta norma detalla las directrices para el diseño seguro y las medidas de protección de robots industriales.

Tras revisar la norma ANSI/RIA R15.06, se seleccionaron los requerimientos relacionados con aspectos específicos identificados durante la evaluación de riesgos, como modos de funcionamiento, límites de velocidad y tipos de paradas, así como aquellos vinculados al funcionamiento de la HMI. Los requerimientos se dividen en dos partes; los elaborados en la primera parte se presentan en los cuadros [6.2](#) y [6.3](#), mientras que los de la segunda parte se detallan en los cuadros [6.4](#), [6.5](#) y [6.6](#).

PARTE 1	
Item	Medidas o requerimientos de seguridad
5.2	<b>General requirements</b>
5.2.1	The safety-related control system performance of an interlocking system conforms to 5.4.
5.2.2	Re-initiation of power does not initiate motion.
5.3	<b>Actuating controls</b>
5.3.2	Actuating controls are constructed or located so as to prevent unintended operation.
5.3.3	Status of the actuating controls is clearly indicated.
5.3.3	If an indicator light is used, it is suitable for its installed location and its color meets the requirements of IEC 60204-1.
5.3.4	Actuating controls are labeled to clearly indicate their function.
5.3.5	While the robot is under local pendant control or other teaching device control, initiation of robot motion or change of local control selection from any other source is prevented.
5.4	<b>Safety-related control system performance (hardware/software)</b>
5.4.2	Single faults are detected at or before the next demand upon the safety function.
5.4.2	When single faults occur, the safety function is always performed and a safe state is maintained until the detected fault is corrected.
5.5	<b>Robot stopping functions</b>
5.5.1	Every robot has a protective stop function and an independent emergency stop function.
5.5.2	Robot has one or more emergency stop circuits.
5.5.2	Each control station has a manual initiated emergency stop function.
5.5.2	Emergency stop function can only be reset by manual action which enables a restart but does not cause the restart.
5.5.2	When the feature of an emergency stop output signal is provided, the output continues to function when the robot power is removed, or an emergency stop signal is generated if the output does not continue to function.
5.5.2	Emergency stop device complies with IEC 60204-1 and ISO 13850.
5.5.3	Robot has one or more protective stop functions with external connection capability.
5.5.3	The protective stop function performance complies with the requirements of 5.4.
5.5.3	This stop function causes a stop of all robot motion, removes or controls power to the robot drive actuators, and allows for the control of any other hazard controlled by the robot system.
5.5.3	At least one protective stop function is category 0 or 1.
5.5.3	If provided, safe stand-still and monitoring function performance complies with 5.4.
5.6	<b>Reduced speed control</b>
5.6.1	Speed of mounting flange and a selected TCP is controllable at selectable speeds.
5.6.1	An off-set feature is provided to allow the TCP speed to be controlled.
5.6.2	When operating under reduced speed control, the speed of the TCP does not exceed 250mm/s [10 in/s].
5.6.3.1	When provided, safety-rated reduced speed control complies with 5.4.2 to ensure reduced speed limit is not exceeded in a fault condition.
5.6.4	When provided, the speed of the TCP is monitored in accordance with 5.4.2.
5.6.4	If the speed exceeds the limit selected, a protective stop is issued.
5.7	<b>Operational modes</b>
5.7.1	Operational modes are selectable with a selector that can be locked in each position.
5.7.1	Each selector is clearly identifiable and allows only one mode be selected at one time.

Cuadro 6.2: Parte 1 de la tabla de requerimientos de la parte 1 de la norma ANSI/RIA R15.06

5.7.2	Safeguarding measures are functioning when robot task program is executed in automatic mode.
5.7.2	Automatic operation is prevented if any stop condition is detected.
5.7.2	Switching from automatic mode results in a stop.
5.7.3	Manual reduced-speed mode meets the requirements of 5.3.4 and 5.6.
5.7.3	Manual reduced-speed mode allows the robot to be operated by human intervention.
5.7.4	Selection requires a deliberate action and an additional confirming action.
5.7.4	Pendant provided conforming to 5.8 with a hold-to-run function in addition to the enabling device.
5.7.4	Means provided to incrementally adjust the speed from the initial value to the full programmed value in multiple steps.
5.7.4	Pendant indicates adjusted speed.
5.7.4	Optionally a separate deliberate action is required to return to the higher speed that was selected before the enabling device switch was released or compressed.
5,8	<b>Pendant controls</b>
5.8.2	Motion of the robot initiated from the pendant or teaching control device is under reduced speed control as described in 5.6.
5.8.2	When the controls contain provisions for selecting manual high-speed, the robot meets requirements of 5.7.4.
5.8.3	Pendant has a three-position enabling device.
5.8.3	When continuously held in a center-enabled position, the enabling device permits robot motion and any other hazards controlled by the robot.
5.8.3 a)	Enabling device operates independently from all other motion control functions/devices.
5.8.3 b)	Release of enabling device/compression past center position stops hazards.
5.8.3 c)	Going from fully compressed to the center position does not permit robot motion.
5.8.3 f)	Dropping an enabling devices does not result in a failure that allows motion to be enabled.
5.8.3 h)	Protective stop initiated when mode changes while enabling device is in center position.
5.8.4	Pendant has a stop function according to 5.5.2.
5.8.5	A means for separate confirmation action from outside the safeguarded space is provided.
5.8.6	Visual indication has been provided, identifying if the pendant is active.
5.12	<b>Axis limiting</b>
5.12.1	When the robot reaches an axis limit, the robot is stopped.
5.12.2	The robot control and task programs do not change electro-mechanical limit device settings.
5.12.3	Control programs for the safety-rated soft limiting are changeable only by authorized personee.
5.12.3	If the safety-rated soft limit is violated, a protective stop is initiated.
5.12.3	Safety-rated zone outputs for use in dynamic restricted space applications comply with 5.4.
5.12.3	Safety-rated soft limits, if used, always become activated upon power up.
5.13.	<b>Movement without drive power</b>
5.13 a	The robot is designed so that the axes are capable of being moved by a single person (if practicable) without the use of drive power in emergency or abnormal situations.
5.13 b	Controls are readily accessible but protected from unintended operation.
5.13 e	Where practicable, warning notices are posted near to the activating controls.

PARTE 2	
Item	Medidas o requerimientos de seguridad.
<b>5.2</b>	<b>Safety-related control system performance (hardware/software)</b>
5.2.3	Results of risk assessment used to determine performance.
<b>5.3</b>	<b>Design and intallation</b>
5.3.2	Automatic mode selection shall be outside the safeguarded area.
5.3.3	Robot system shall not respond to any external remote control commands or conditions causing hazardous situations.
5.3.6	Hazardous energy isolation means provided.
5.3.6	Hazardous energy isolation means shall be clearly marked.
5.3.8.1	Robot system is provided with a protectice stop.
5.3.8.1	Robot system is provided with a emergency stop.
5.3.8.2	Each control station capable of initiating robot motion or other hazardous functions shall have an emergency stop.
5.3.8.2	Actuation stops all robot motion and other hazardous functions in the cell.
5.3.8.2	Has a single emergency stop for the system or span of control for multiple functions.
5.3.8.2	All emergency stop devices in a single workspace have the same span of control.
5.3.8.2	Emergency stop function shall be either stop category 0(parada por desconexion inmediata de la alimentacion) or categoria 1 (parada controlada con energia disponible para los actuadores y quitar la energia cuanod se detienen).
5.3.11	Emergency recovery procedures signs or labels are affixed to the robot system or instructions for affixeing are provided.
5.3.15	Additional enabling devices and their integration comply with Part 1, Clause 5.8.3.
5.3.15	Sufficient additional enabling devices for thr expected number of personnel within the safeguarded space are provided.
5.3.15	Enabling devices are interlocked in areas of cell where personnel are exposed to a hazard from additional robot or other equipment.
<b>5.4</b>	<b>Limiting robot Motion</b>
5.4.2	The safeguarded space has been established by perimeter guarding with due considerations of location and layout and hazards within.
5.4.2	The restricted space of the robot has been established by means which limit the motion of the robot, end-effector, fixture and workpiece.
5.4.2.	The restricted space lies within the safeguarded space.
5.4.2	The perimeter safeguards are not installed closer to the hazard than the restricted space or the perimeter safeguard is designed to be the limiting device in accordance with 5.4.3.
5.4.3	Mechanical robot limiting means comply with [part 1, clause 5.12.2].
5.4.3	Non-mechanical robot limiting means comply with [Part 1, clause 5.12].
5.4.3	Control systems used for non-mechanical robot limiting means comply with [Part 1, 5.2.2] unless a risk assessment detemrines another performance.
5.4.3	The limiting devices have been correctly adjusted and secured.
<b>5.5</b>	<b>Layout</b>
5.5.2	Where practicable layout design allows tasks to be performed outside the safeguarded space.
5.5.2	Safe and adequate access to task locations within safe guarded space has been provided.
5.5.2	Access path and means are free of hazards.
5.5.2	Permanent means of access provided for high frequency tasks.
5.5.2	Access means provided to electrical equipment mounted above normal reach.

Cuadro 6.4: Parte 1 de la tabla de requerimientos de la parte 2 de la norma ANSI/RIA R15.06

5.5.4	Operation modes in accordance with 5.6.4.2, 5.6.4.3 or a separate control mode is provided for process observation inside safeguarded space.
<b>5.6</b>	<b>Robot system operational mode application</b>
5.6.2	Unauthorized and/or inadvertent mode selection is prevented.
5.6.2	Selection means only enables selected mode and not operation.
5.6.2	Separate actuation required to initiate robot operation.
5.6.2	Unambiguous indication of operation mode provided.
5.6.2	Changing mode of operation does not create hazards.
5.6.3.1	Entering safeguarded space while in automatic mode causes a protective stop.
5.6.3.2	Selection of automatic mode does not override protective stop or emergency stop condition.
5.6.3.2	Selection of automatic mode is located outside safeguarded space .
5.6.3.2	Switching from automatic mode causes a stop.
5.6.3.3	Automatic operation is initiated from outside the safeguarded space.
5.6.3.3	Initiation of automatic mode only possible when all safeguards are active.
5.6.3.4	Start and restart is a simple operation.
5.6.3.4	Protective measures are functional prior to start and restart.
5.6.3.4	Safety-related control function complies with 5.2.2.
5.6.3.4	Manually actuated interlocks for start and restart provided.
5.6.3.4	Personnel are protected from start and restart while in the safeguard space.
5.6.3.4	Manually actuated start and restart controls can not be activated from inside safeguard space.
5.6.3.4	Functional safety and protective measures provided to ensure no one in in the safeguard space prior to start and restart.
5.6.3.4	Presence sensing or audio-visual pre-start warning signal provided when necessary.
5.6.3.4	Accessible emergency stop devices are provided within safeguarded spaces.
5.6.4.1	Local control during manual intervention uses pendant.
5.6.4.1	Location of control devices allows observation of work area.
5.6.4.1	Stop control device placed near each start control device.
5.6.4.1	Control prevented from other sources during local control.
5.6.4.2	Adjustment of TCP using offset feature has been accomplished.
5.6.4.2	Maximum velocity of TCP less than 250 mm/s [10 in/s].
5.6.4.2	Determination if slower speed required.
5.6.4.2	Functional enabling device required for motion.
5.6.4.3	Requirements of [Part 1] and of pendants met.
5.6.5	Only a single control source during remote control is active at one time.
5.6.5	No overriding of local selection or causing hazardous conditions due to remote control.
5.6.5	Activating Remote function only possible from local operation control.
5.6.5	Restrictions exist for remotely changing safety-rated soft axis and space limiting parameters.
5.6.5	Indication of active remote control exists.
5.6.5	Remote control is prevented when in manual high-speed.
<b>5.7</b>	<b>Pendant</b>
5.7.1	Pendant emergency stop complies with 5.3.8.2.
5.7.1	Cabled pendants have sufficient cable length and durability.
5.7.1	Pendant storage provided to prevent damage.
5.7.2	Emergency stop and enabling device comply with [Part 1, 5.8].
5.7.2	Unintentional control is avoided
5.7.2	Single point of control is provided.
<b>5.8</b>	<b>Maintenance and repair</b>

Cuadro 6.5: Parte 2 de la tabla de requerimientos de la parte 2 de la norma ANSI/RIA R15.06

5.8.2	System designed for safe access for maintenance.
5.8.3	The guards are of sufficient size to allow easy access.
<b>5.10</b>	<b>Safeguarding</b>
5.10.4.1	Interlocking devices associated with guards meet applicable requirements.
5.10.4.1	Fixed guards require tools to remove.
5.10.4.2	The gap between the bottom of the guard and the floor or the gap in the guard cannot allow a person to reach over, under, around or through the guard and reach a hazard.
5.10.4.3	Movable guards at their closed position prevent operators reaching hazardous areas.
5.10.4.3	Movable guards open laterally or away from the hazard, and not into the safeguarded space.
5.10.4.3	Interlocking is provided to bring any hazards to a safe state before an operator can gain access.
5.10.4.3	Movable guards used to initiate starting on closure (control guards) meet applicable
5.10.4.4	Where required, guard locking is provided in addition to the control interlock.
5.10.4.4	Guard locking complies with 5.10.4.4 a) and b).
5.10.4.4	Safeguarded space has a means to prevent a person from being trapped inside.
5.10.5.2	Reset of the sensitive protective equipment does not by itself initiate hazardous conditions.
5.10.5.2	Resetting of a restart interlock requires a deliberate human action.
5.10.6.2	The gap between a moving station and any fixed elements does not exceed 120 mm.
5.10.7	The muting function fulfills the requirements of ISO 13849-1.

Cuadro 6.6: Parte 3 de la tabla de requerimientos de la parte 2 de la norma ANSI/RIA R15.06

## 6.6. Requerimientos específicos del Teach pendant

El Teach pendant permite el manejo del sistema por parte del operario, y este debe cumplir los siguientes requerimientos:

- Permitir el cambio de modo de operación por parte del operario.
- Contar con un dispositivo que monitoree que el pendant se encuentre en las manos del operario, con el propósito de no accionar accidentalmente un comando en caso de caída.
- Emplear un dispositivo que detecte cuando el operario entra en pánico, para realizar una parada de emergencia.
- Mostrar datos en tiempo real al usuario de las variables dinámicas del sistema.
- Su diseño debe ser ergonómico y cómodo para el usuario.

# Diseño de la arquitectura de control

---

El inicio del desarrollo del sistema de control implica la fase de diseño, un proceso en el cual se planifican ideas con el propósito de abordar las problemáticas o necesidades previamente identificadas en el proyecto. Esta etapa se ha subdividido en tres secciones clave: Desarrollo de Software, Hardware correspondiente e Integración de la Arquitectura. Cada una de estas áreas abordará detalladamente los aspectos tanto del software como del hardware, culminando con la explicación de cómo estos elementos se unirán para constituir la estructura final.

## 7.1. Diseño de software

Basándonos en los requisitos, se ha definido que el sistema debe incorporar cuatro modos de funcionamiento: Manual Baja Velocidad, Manual Alta Velocidad, Automático y Remoto. Además, se ha establecido que el sistema deberá incluir un Estado de Seguridad. Por otra parte, se han identificado la necesidad de realizar cambios en la programación del PLC, el cual tiene la responsabilidad de controlar los motores y, por ende, los movimientos del robot. Estas modificaciones en la programación implicarán la implementación de una máquina de estados para determinar en qué modo se encuentra el robot y diversas funciones de parada que podrían ejecutarse cuando sea necesario.

### 7.1.1. Modo Manual baja velocidad

Este modo facilita la interacción del operario con el robot a través del Teach Pendant, un dispositivo de control que le permite al usuario definir las acciones o movimientos a una velocidad inferior a  $255 \frac{mm}{s}$ . De acuerdo con la normativa ANSI/RIA R15.06, este modo debe cumplir con los siguientes requisitos:

- Velocidad máxima de funcionamiento es de  $255 \frac{mm}{s}$
- Si sobrepasa la velocidad máxima, se debe detener el movimiento
- Permite el control con intervención humana
- Requiere el uso de un dispositivo Hold-To-Run
- Si el Hold-To-Run se suelta y se vuelve a presionar, el robot no debe moverse
- Cuando el pendant es encendido se inicia en el modo Manual Low

- Se debe ejecutar una parada de protección al cambiar a modo Manual High
- Para cambiar de modo se debe soltar el Hold-To-Run y volver a presionarlo

El diseño del modo Manual Low se realizó utilizando un diagrama de flujo, el cual se muestra en el anexo [A](#) en la figura [A.1](#). Al ingresar a este modo, se hacen dos confirmaciones: primero, se evalúa si el robot se encuentra encendido, y en ese caso, se evalúa si el robot ejecutó previamente la rutina de homing. Cumplidas ambas condiciones, el usuario tiene la opción de cambiar de modo o mover el robot, lo puede hacer por medio de botones individuales por eje(X,Y,Z) o ingresando las posiciones en cada uno de ellos. Estas dos opciones para el control de movimiento se ejecutan a la misma velocidad, esta puede ser variada, pero no puede sobrepasar la velocidad máxima de  $255 \frac{mm}{s}$ . Antes de iniciar el movimiento, se verifica que la posición sea alcanzable y que el Hold-To-Run este presionado, si se cumplen estas dos condiciones se inicia el movimiento. Mientras se ejecuta el movimiento, el usuario puede hacer una parada de emergencia o controlada si lo desea.

### 7.1.2. Modo Manual alta velocidad

Este modo permite la interacción del operario con el robot por medio del Teach Pendant, el cual es un dispositivo de control, que le permite al usuario definir las acciones que debe llevar a cabo o movimientos a una velocidad mayor a  $255 \frac{mm}{s}$ . Según la normativa ANSI/RIA R15.06, este modo debe cumplir con los siguientes requisitos:

- Indicar la velocidad de movimiento
- Velocidad predeterminada es de  $255 \frac{mm}{s}$ ,
- Permite el control con intervención humana
- Requiere el uso de un dispositivo Hold-To-Run
- Si el Hold-To-Run se suelta y se vuelve a presionar, el robot no debe moverse
- Se debe ejecutar una parada de protección al cambiar a modo Manual Low

El diseño del modo Manual High se realizó utilizando un diagrama de flujo, el cual se muestra en el anexo [A](#) en la figura [A.2](#). Al ingresar a este modo, se verifican las dos condiciones iniciales que se evalúan en el modo Manual Low. Una vez que estas dos condiciones se han cumplido, si el usuario desea cambiar de modo, puede hacerlo; si el usuario desea mover el robot, lo puede hacer por medio de botones individuales por eje(X, Y, Z) o introduciendo las posiciones en cada uno de estos ejes. Estas dos opciones para el control de movimiento se ejecutan a la misma velocidad, esta puede ser variada, pero no puede sobrepasar la velocidad máxima del sistema que es  $3000 \frac{mm}{s}$ , ni ser menor a  $255 \frac{mm}{s}$ , debido a que estos movimientos se pueden ejecutar en el modo Manual Low. Antes de iniciar el movimiento, se verifica que la posición sea alcanzable y que el Hold-To-Run este presionado, si se cumplen estas dos condiciones se inicia el movimiento. Mientras se ejecuta el movimiento, el usuario puede hacer una parada de emergencia o controlada si lo desea.

### 7.1.3. Modo Automático

Este modo permite la interacción del operario con el robot por medio del Teach Pendant, el cual es un dispositivo de control, que le permite al usuario iniciar una rutina previamente enseñada al robot para que sea ejecutada de forma repetitiva. Conforme a la normativa ANSI/RIA R15.06, se especifica que este modo cumpla con los siguientes requisitos:

- Funciones de seguridad deben funcionar
- No se debe mover si se ejecuta una función de parada
- Cuando se cambia a otro modo, el robot debe detenerse
- Solo se puede utilizar este modo cuando no se encuentren usuarios dentro de la celda robótica.

El diseño del modo Automático se llevó a cabo mediante un diagrama de flujo, el cual se muestra en el anexo [A](#) en la figura [A.3](#). Al ingresar a este modo, se verifican las dos condiciones iniciales que se evalúan en el modo Manual Low. Una vez cumplidas estas dos condiciones, se evalúa si la guarda se encuentra cerrada, garantizando que no se encuentre una persona en su interior durante la realización de la rutina programada. En caso de que el usuario desee cambiar de modo durante la ejecución de los movimientos, tiene la libertad de hacerlo. Antes de iniciar el movimiento, se verifica que la posición sea alcanzable, se realiza el movimiento, finalmente se carga la siguiente posición y repite el proceso. Durante la ejecución del movimiento, el usuario puede hacer una parada de emergencia o controlada si lo desea.

### 7.1.4. Modo Remoto

Este modo permite la interacción del operario con el robot por medio de un computador, este le permite al usuario controlar el robot de forma remota, pero dentro de la red. Conforme a los lineamientos de la normativa ANSI/RIA R15.06, se establece que este modo debe satisfacer los siguientes requisitos:

- Solo debe recibir instrucciones desde un solo dispositivo.
- Su modo de control es el mismo que en el modo Manual Low.

El diseño del modo Manual Remoto se elaboró mediante un diagrama de flujo, el cual se presenta en el anexo [A](#) en la figura [A.4](#). Al ingresar a este modo, se evalúan las dos condiciones iniciales que se evalúan en el modo Manual Low. Una vez se han verificado estas dos condiciones, el usuario tendrá las mismas opciones de control que en el modo de baja velocidad, lo cual le permitirá realizar movimientos por ejes individuales o escribiendo la posición final de cada uno. Para evitar que el sistema reciba ordenes desde el Teach pendant y el computador donde se ejecuta la aplicación de control remoto, este modo solo podrá ser seleccionado y cambiado desde el pendant. Esto significa que el usuario podrá cambiar al modo remoto y esto inhabilitará el pendant para ejecutar movimientos, otorgando el control a la aplicación de computadora. Por otro lado, si el usuario desea salir de este modo, debe hacerlo desde el pendant, habilitando de nuevo su control e inhabilitando la capacidad de la aplicación de dar instrucciones al robot.

### 7.1.5. Estado de seguridad

Este estado permite que el robot se mantenga detenido si no se cumple alguna condición de seguridad, una parada de emergencia o si se requiere acceder al robot directamente. Dado que se fundamenta en la normativa ANSI/RIA R15.06, se establece que el estado debe cumplir con los siguientes requisitos:

- El robot debe estar completamente detenido en este estado.

El diseño del Estado de seguridad se realizó utilizando un diagrama de flujo, el cual se muestra en el anexo [A](#) en la figura [A.5](#). Se bloquea cualquier opción de comando de movimiento una vez que se activa este modo, y esta restricción persiste hasta que el usuario cambie de modo. Para salir de este estado, el operario debe garantizar que el robot no presente ningún error en su funcionamiento, esto incluye errores en los ejes o eventos que pongan en riesgo la integridad de la persona. Por ejemplo, que la puerta se encuentre abierta o que los sensores de límite se encuentren activos.

### 7.1.6. Programación del PLC

EL PLC debe ser programado para que ejecute las funciones que el código principal correrá, pues este ejecuta funciones básicas, como mover los motores, para lo cual necesita la velocidad y posición a la que se debe mover cada motor. Sin embargo, los cálculos matemáticos necesarios para determinar estas dos variables demandan una alta capacidad computacional, la cual el PLC no posee. En consciencia, los cálculos se deben ejecutar en un computador con la capacidad adecuada. Además de lo anterior, en la programación del PLC se implementará una máquina de estados, la cual se encargara de regular el flujo de los modos. Esto logrará que el sistema solo se encuentre en un solo modo a la vez, garantizando así el cumplimiento de sus requerimientos. Su diagrama de flujo se puede ver en el anexo [A](#) en la figura [A.6](#).

## 7.2. Diseño de hardware

De acuerdo al análisis de riesgos hecho al sistema robótico y a la lista de requerimientos expuestos por la norma ANSI/RIA R15.06, se necesitaron hacer cambios a la estructura física del mismo. Entre estos se proyectó el encierro de la zona de peligro que permita supervisar la presencia del operario en zona segura, la implementación de otra parada de emergencia y la integración de un pendant que permita el manejo de modos y visualización de datos en tiempo real por parte del operario.

### 7.2.1. Guarda de encierro

Las barreras de seguridad, permiten garantizar que el operario se encuentre fuera de zona de peligro para la activación de ciertos modos, cuya operación se prohíbe por seguridad, cuando el operario se encuentra dentro de esta zona. Para lograr este objetivo, se estudiaron diferentes alternativas, considerando aspectos como la altura, funcionalidades, categoría de seguridad con la que cada opción cumple y su precio, entre la cortina de luz de seguridad F3SG-RA, de la compañía

OMROM y el uso de barreras físicas de policarbonato junto a un mecanismo de apertura, que permita al operario la entrada a la zona de peligro. Dentro de esta comparación, destaca la necesidad de tener en cuenta una distancia de seguridad mayor en los sistemas no físicos, como los son las barreras de luz y barreras laser, teniendo que considerar factores como el tiempo de respuesta de la máquina y velocidad de reacción del ser humano, comprendidas en los estandares ISO 13854 y ISO 13855, para extremidades superiores e inferiores. [25] - [26]

De acuerdo a los puntos mencionados anteriormente, y la estructura del chasis del robot, la cual facilita el uso de las barreras físicas de policarbonato, se decide por esta opción junto al mecanismo de apertura, el cual se requiere implementar junto al interruptor de puerta de seguridad con bloqueo de protección D4NL de la compañía OMROM, que permite monitorear y bloquear la puerta de acceso a la zona, conectado al PLC principal, por medio de un relé de seguridad G9SE-401, de la misma compañía. Sus conexiones se ilustran en la sección llamada *Interlock*, del diagrama eléctrico del sistema. Su diseño se visualiza en el anexo [B.1], [B.2], [B.3] y [B.4] de la parte frontal, izquierda y completa, mientras que el diseño de la puerta corrediza para la interacción directa del operador con el robot se visualiza en la figura [B.5].

La guarda de encierro, marca el límite entre la zona segura y la zona de peligro. Para su implementación, se requirieron 9 láminas de policarbonato, que deben ser instaladas sobre los laterales de la estructura metálica del robot, compuesta por una altura de 205 cm, ancho de 159 cm y 159 cm de fondo. Las láminas fueron fijadas perforando el chasis que sostiene el sistema, dejando espacio para la puerta de apertura en su lado frontal, encargada de permitir el ingreso del operario a la zona de peligro. Asimismo, se debió instalar el dispositivo Interlock, encargado de bloquear y reportar si la puerta se encuentra cerrada, necesario para la activación de los modos de funcionamiento a alta velocidad, dejando solo en funcionamiento al modo manual bajo (*Manual Low*), cuando el operario reporte la necesidad de apertura desde el botón ubicado en el panel eléctrico, desactivando el interruptor de bloqueo por solenoide.

### 7.2.2. Teach Pendant

A fin de cumplir las especificaciones anteriores, se procede a listar los dispositivos necesarios con el que debe contar el integrado:

- Computadora
- Pantalla táctil
- Botón de parada de emergencia
- Switch de tres posiciones
- Estructura

Posteriormente, se procedió a buscar alternativas para cada dispositivo, comparando distintas opciones que ofrece el mercado y que se ilustran en el anexo B en los cuadros B.11, B.12, B.13, optando por utilizar las siguientes componentes:

Dispositivo	Especificación
Computadora	Raspberry Pi 4 - 4GB RAM
Pantalla	Display LCD capacitivo 7"
Switch de tres posiciones	A4E-C211VA OMROM
Botón de emergencia	Seta de emergencia NA 22mm

Cuadro 7.7: Tabla de especificaciones pendant

Teniendo los dispositivos necesarios, se procede a diseñar la conexión de estos, mostrando las actualizaciones en el diagrama eléctrico, en la sección de *Pendant*, donde el botón de emergencia y el switch de 3 posiciones *A4E-C211VA* se conecta al PLC, y la pantalla táctil se conecta a la computadora *Raspberry pi 4* por medio del cable HDMI. La computadora, a su vez, tiene una conexión inalámbrica al PLC, por medio de la red propia del sistema proporcionada por el Router. Para integración, y buscando un diseño ergonómico para el usuario operador, se diseña una carcasa conformada por una caja y su respectiva caja, que permite al usuario la sujeción del dispositivo con la mano izquierda, al mismo tiempo que este puede sostener el switch de 3 posiciones para la habilitación de las funciones del mismo. el diseño de la caja y de la tapa se ilustran en las figuras B.7 y B.8 respectivamente, mientras su ensamble se observa en la figura B.6 sin componentes, dando como resultado la visualización de la figura 7.1, al añadir el switch, botón de emergencia y pantalla, siendo estos los componentes visibles del dispositivo.

### 7.2.3. Parada de emergencia

El sistema, contaba con un botón de parada de emergencia en la parte exterior de la estructura, en la zona segura para el operario. Sin embargo, de acuerdo a la norma y los riesgos identificados en la matriz de riesgos, se requiere implementar otro botón dentro de la zona de peligro, considerando que el usuario se encuentre en esta área y el robot se encuentre en movimiento en alguno de los modos de operación. Por lo anterior, se diseña la conexión eléctrica para la instalación de un botón dentro de la barrera de seguridad de policarbonato, al cual se le diseñó una caja para su montaje, proyectándose a poder ser fijada en una cara interna del chasis del robot, mostrándose en la sección B.4 de anexos, en las figuras B.10, B.11, B.12 y B.13. Lo anterior, recordando que también se cuenta con un botón de emergencia adicional en el Teach Pendant, completando así, 3 botones de parada de emergencia, distribuidos en las zonas más críticas y más frecuentes en la operación del usuario con la máquina.

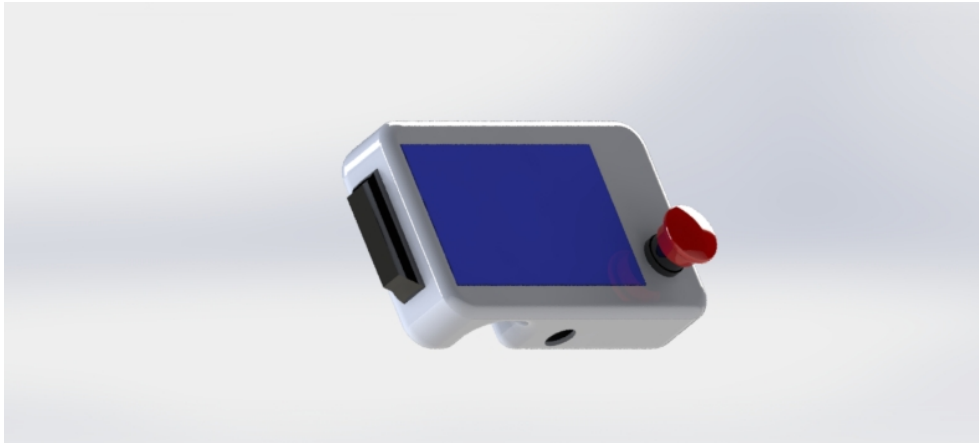


Figura 7.1: Diseño de estructura ensamblada del teach pendant con componentes

### 7.3. Integración de la arquitectura

Una vez definidos los dos diseños, tanto de hardware como software, se debe realizar la integración de ambas partes para darle forma a la arquitectura de control.

Partiendo desde el encendido del robot, la primera acción que realiza el sistema es la validación de que no se encuentre ninguna persona dentro de la celda robótica, gracias a la guarda de encierro de policarbonato se asegura que no se podrá acceder a la celda robotica por sus laterales, solo por la puerta delantera. Se debe asegurar que la puerta se encuentre cerrada y asegurada para iniciar las tareas del robot. Esto se logra por medio del interruptor de puerta de seguridad con bloqueo de protección D4NL que incorpora actuadores para verificar el estado de la puerta (cerrada o abierta) y si está bloqueada o asegurada. Además, gracias al relé de seguridad G9SE-401, el usuario esta obligado a presionar el botón de confirmación ubicado fuera de la zona de peligro. Esto quiere decir que una persona que se encuentre dentro de la celda robótica no podrá presionar el botón, por ende este no iniciara sus funciones. El monitoreo de estas variables es realizado por el PLC, el cual notifica al usuario si se encuentra en condiciones para comenzar con su funcionamiento.

El sistema al estar apagado no asegura la puerta de acceso, lo cual permite el paso a la celda robótica de forma simple, para tareas de manteniendo. En la parte externa de la celda, donde se ubica la puerta, se instalará un selector, el cual puede estar en la posición de *Abierto* o *Cerrado*. El usuario podrá asegurar o desasegurar la puerta dependiendo de las labores a realizar en la celda y un botón el cual debe ser presionado una vez se pase de posición de abierto a cerrado, para confirmar que no hay nadie dentro de la celda y se puede reanudar el funcionamiento del robot. En caso de que el robot sea encendido, el estado del selector notificara al PLC que hay una persona dentro de la celda, por esta razón no debe asegurar la puerta y tampoco permitir que el robot realice movimientos. Para continuar con las funciones del robot, se debe cerrar la puerta, el selector debe estar en la posición de *Cerrado* y finalmente presionar el botón de confirmación.

Si no hay nadie dentro de la celda, el pendant debe encenderse y conectarse a la red para dar inicio al funcionamiento, esto es crucial, pues la Raspberry dentro del pendant ejecuta las funciones del computador donde se realizan los cálculos cinemáticos y dinámicos para el correcto funcionamiento del robot. Inicialmente, el robot entra en modo Manual baja velocidad, donde el usuario puede ejecutar funciones como Homming, Movimientos y configuración del TCP, pero para hacer cualquiera de estas acciones el usuario debe mantener el switch de tres posiciones, localizado en el pendant, presionando en la segunda posición. Si el usuario no mantiene presionado el switch, el robot entrará en el Estado de seguridad, donde no se podrá desempeñar ninguna tarea, hasta que se suelte por completo y luego vuelva a esa posición.

Finalmente, las paradas del robot serán evaluadas y controladas por el software. Si se incumple con una norma de seguridad, se detendrá el robot o no se permitirá la ejecución de la función solicitada por el usuario. Por ejemplo, hacer un movimiento por encima de la velocidad máxima en el modo Manual baja velocidad o hacer un movimiento a una posición inalcanzable en el modo Manual alta velocidad. Además, en caso de requerir una parada de emergencia completa, se podrá llevar a cabo por medio de los botones con enclavamiento localizados fuera y dentro de la celda robótica.

# Implementación del sistema de control

---

## 8.1. Implementación de software

Se llevó a cabo la implementación de los modos diseñados mencionado en la sección anterior. El funcionamiento de cada uno de estos modos se realizó en el lenguaje de programación Python y la programación del PLC se realizó en el lenguaje Texto estructurado con la herramienta Connected Components Workbench (CCW).

Además, utilizando la librería gráfica Tkinter, en Python, se desarrolló la interfaz gráfica con la cual el usuario interactuara con el robot desde el pendant.

### 8.1.1. Interfaz gráfica de usuario (GUI)

Se desarrolló una interfaz gráfica con la que el usuario puede interactuar y le ofrece actuadores para ejecutar acciones e indicadores o testigos para mostrar el estado en que se encuentran diferentes variables de interés, como se puede ver en las figuras a continuación.

En la figura [8.1](#) se puede ver como la interfaz gráfica cuenta con tres secciones. En el segmento izquierdo, se pueden controlar diferentes acciones y monitorear variables del robot, las cuales se describen a continuación.

1. Información de funcionamiento: Este elemento muestra al usuario información del robot, como los son la IPV4 del PLC en la red en la que se encuentra en funcionamiento y el estado de los 3 ejes del robot. Este le indica al usuario el estado de los servomotores individualmente, si están funcionando correctamente se mostrará *OK*, mientras que si se presenta algún error se mostrará *ERROR #* donde # es el código numérico del error que presenta.
2. POWER: Permite el apagado y encendido de los servomotores que accionan el robot. El indicador es color verde cuando están encendidos y color rojo cuando están apagados.
3. RESET: Ordena al PLC que reinicie los Drivers y Servomotres. Esta función se utiliza cuando los motores o drivers se encuentran en conflicto debido a un error interno, por lo que detienen su funcionamiento hasta que estos sean reiniciados.
4. HOMING: Ordena al robot ejecutar la rutina de homing, con la que el robot encuentra su posición inicial. El indicador es color verde cuando la rutina de homing se realizó de forma correcta y color rojo cuando la rutina de homing no ha sido realizada.

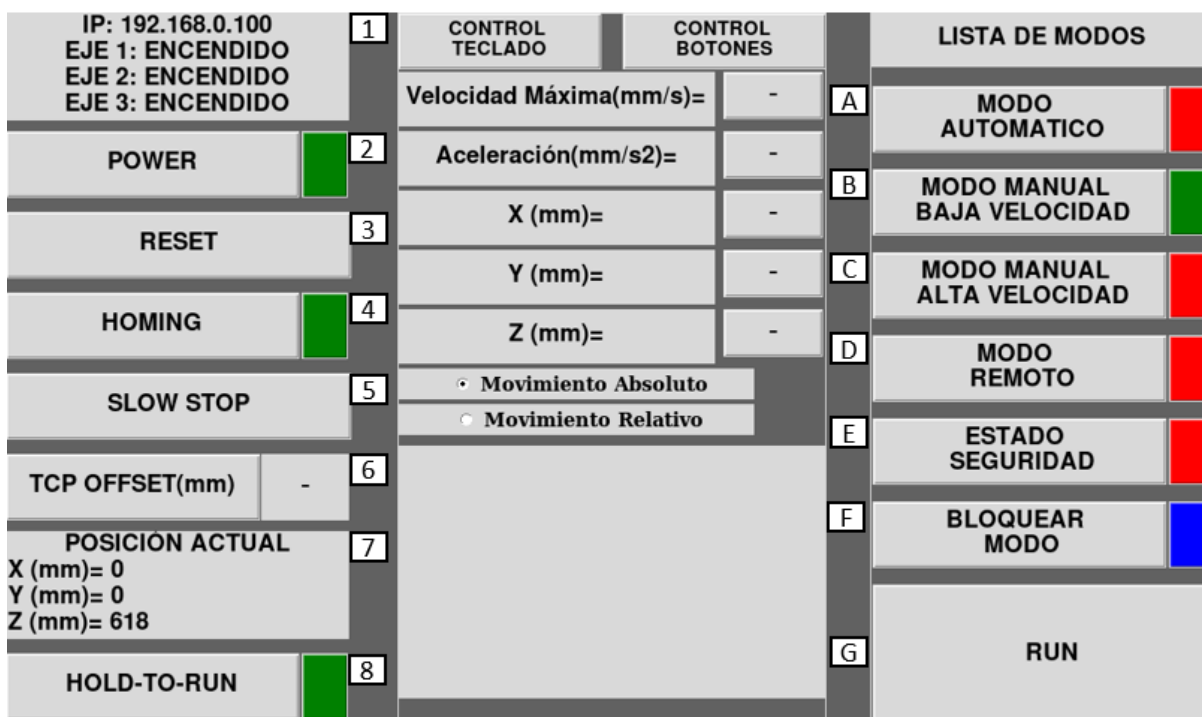


Figura 8.1: Interfaz gráfica de usuario (GUI) sección izquierda

5. SLOW STOP: Se utiliza para que el robot ejecute su función de parada suave, la cual detiene el robot desacelerándolo lentamente para evitar daños o estrés sobre los mecanismos o actuadores del sistema.
6. TCP OFFSET(mm): Despliega un módulo que permite especificar la altura de la herramienta que se ensamblara en la brida del robot para ejecutar la función que desee el usuario. Si este parámetro no se configura correctamente, las operaciones que ejecute la herramienta podrían ser inexactas.
7. POSICIÓN ACTUAL: Muestra las coordenadas(X, Y, Z) donde se encuentra la brida o la punta de la herramienta si se ha configurado previamente.
8. Estado del interruptor de seguridad: Este indicador muestra si el switch de tres posiciones, está en la posición correcta, la cual es la segunda posición. El indicador se torna color verde si el interruptor se encuentra en la segunda posición y rojo si se encuentra en la posición uno o tres.

En el segmento derecho de la figura [8.1](#) se pueden ver botones e indicadores, los cuales hacen referencia a los diferentes modos en los que el robot puede operar. Los botones se utilizan para cambiar entre modos y los testigos muestran en que modo se encuentra el robot actualmente.

A continuación se describe cada uno.

- A. Modo automático: Cambia el modo de funcionamiento del robot a modo automático.
- B. Modo manual baja velocidad: Cambia el modo de funcionamiento del robot a modo manual baja velocidad.
- C. Modo manual alta velocidad: Cambia el modo de funcionamiento del robot a modo manual alta velocidad.
- D. Modo remoto: Cambia el modo de funcionamiento del robot a modo remoto.
- E. Estado de seguridad: Se cambia al estado de seguridad cuando los servomotores entran en fallo o no se cumplen con las condiciones de seguridad para el funcionamiento del robot.
- F. Bloqueo de modo: Este botón se utiliza como confirmación para pasar de un modo a otro. Este debe presionarse antes de cambiar a otro modo, se utiliza como una doble confirmación para el cambio y así mitigar el cambio por error, a su lado se encuentra un indicador que se torna de color azul cuando el bloqueo de modo está activo, lo que quiere decir que no se puede realizar el cambio del modo actual a otro modo. Una vez se presiona el botón, el indicador se torna rojo, lo cual indica que el cambio de modo es posible en ese momento.
- G. RUN: Este botón se utiliza en los modos Manual baja velocidad y Manual alta velocidad, donde utilizando el modo de control por teclado, ingresando las coordenadas X, Y, Z manualmente, al presionar el botón *RUN* se ejecuta el movimiento configurado.

Finalmente, se encuentra la sección central, la cual dependiendo del modo cambia sus actuadores para brindarle al usuario herramientas que pueden ser diferentes entre los modos. Esta sección se describirá en cada modo individualmente.

### 8.1.2. Modo Manual baja velocidad

Se implementó este modo, según el diagrama de estados de la figura [A.1](#), donde por medio de condicionales el sistema identifica que hacer, siguiendo las indicaciones del usuario, como por el ejemplo el modo de control, donde el usuario podría indicarle al sistema que el robot se mueva ingresando las coordenadas(X, Y, Z) de destino o moviendo individualmente cada eje hasta llegar a la posición deseada. Este sistema se ejecuta en una Raspberry Pi 4, la cual le envía al PLC qué variables deben cambiar y a que valor, para ejecutar las funciones que el usuario desea.

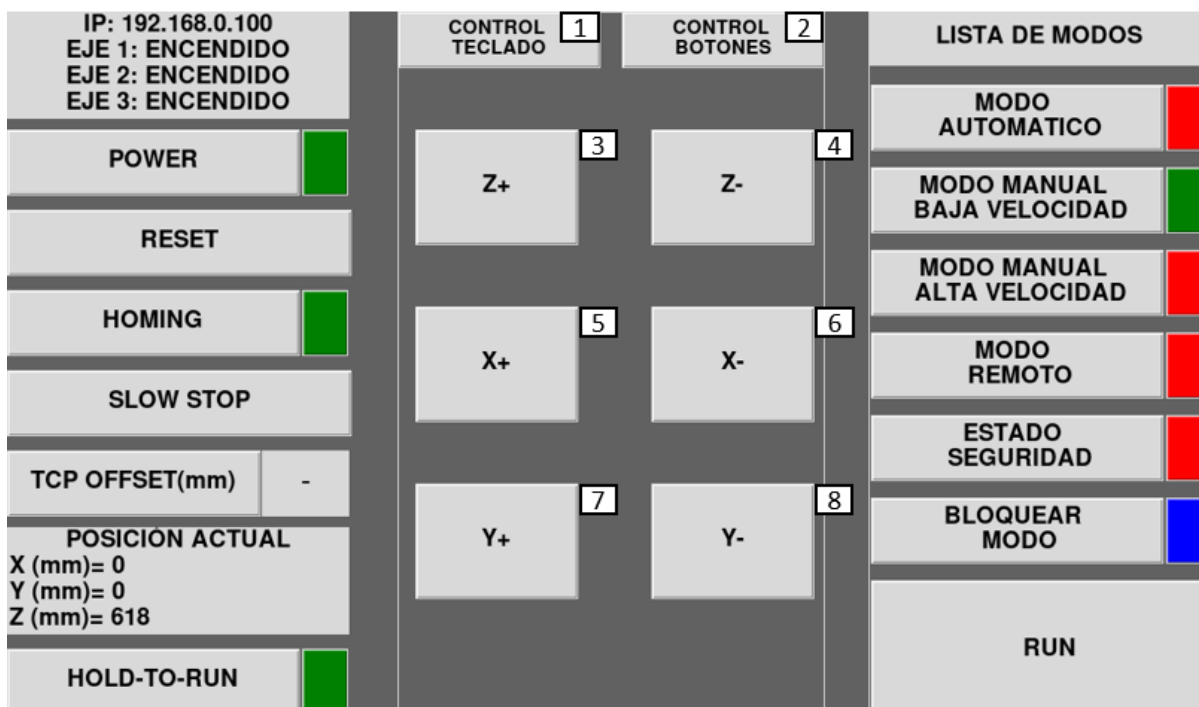


Figura 8.2: Interfaz gráfica modo Manual Baja velocidad (Control ejes individuales)

En la figura [8.2](#) se pueden ver en la sección central diferentes actuadores, estos se utilizan para especificar a donde debe moverse el robot. A continuación se describen sus actuadores, los cuales se habilitan con el modo de control por botones, donde se mueve cada eje individualmente.

1. CONTROL TECLADO: Se utiliza para que el usuario seleccione si desea que el control se realice ingresando manualmente las coordenadas por teclado.
2. CONTROL BOTONES: Se utiliza para que el usuario seleccione si desea que el control se realice moviendo individualmente cada eje (X,Y,Z).
3. Z+: Se mueve la brida del robot en el sentido Z positivo. Viendo el robot desde el frente, es un movimiento hacia arriba.
4. Z-: Se mueve la brida del robot en el sentido Z negativo. Viendo el robot desde el frente, es un movimiento hacia abajo.
5. X+: Se mueve la brida del robot en el sentido X positivo. Viendo el robot desde el frente, es un movimiento hacia la izquierda.
6. X-: Se mueve la brida del robot en el sentido X negativo. Viendo el robot desde el frente, es un movimiento hacia la derecha.
7. Y+: Se mueve la brida del robot en el sentido Y positivo. Viendo el robot desde el frente, es un movimiento hacia atrás.
8. Y-: Se mueve la brida del robot en el sentido Y negativo. Viendo el robot desde el frente, es un movimiento hacia adelante.

Para la realización de este proyecto se utilizo el sistema de coordenadas con el cual el robot ya funcionaba, este sistema no cumple con la regla de la mano derecha".

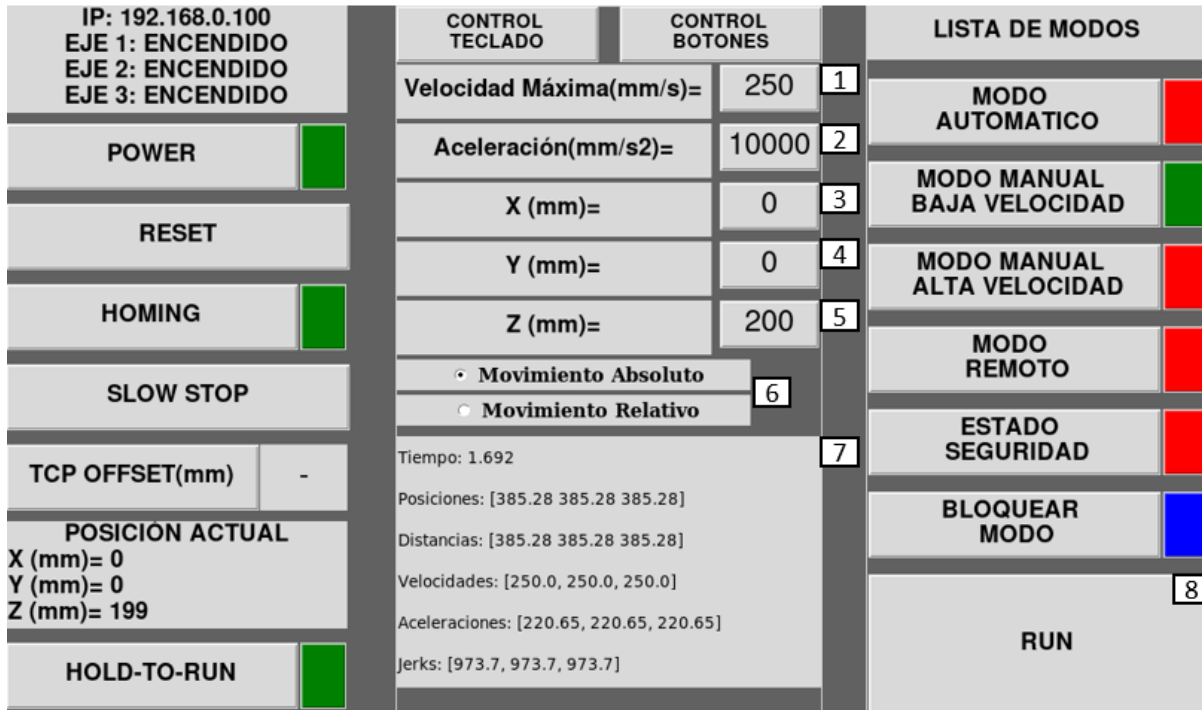


Figura 8.3: Interfaz gráfica modo Manual Baja velocidad (Control por teclado)

En la figura 8.3 se pueden ver, en la sección central, botones diferentes a los de la figura 8.2, la cual se describió anteriormente. A continuación se describirán estos botones, los cuales se habilitan con el modo de control por teclado, donde se definen las posiciones manualmente.

1. Velocidad Máxima(mm/s): Se establece la velocidad con la que se ejecutará el movimiento, esta velocidad es como máximo 250(mm/s) en este modo. Se le notificará al usuario si la velocidad deseada es mayor a la permitida.
2. Aceleración( mm/s<sup>2</sup>): Se selecciona la aceleración con la que se ejecutará el movimiento, esta aceleración es como máximo 10.000( mm/s<sup>2</sup>).
3. X(mm): Se establece la coordenada en el eje X donde se ubicará el robot al ejecutar el movimiento. Se le notificará al usuario si esta ubicación está fuera del área de trabajo.
4. Y(mm): Se define la coordenada en el eje Y donde se ubicará el robot al ejecutar el movimiento. Se le notificará al usuario si esta ubicación está fuera del área de trabajo.
5. Z(mm): Se determina la coordenada en el eje Z donde se ubicará el robot al ejecutar el movimiento. Se le notificará al usuario si esta ubicación está fuera del área de trabajo.
6. Selector de tipo de movimiento: El usuario puede seleccionar si desea que el robot ejecute un movimiento Absoluto o un movimiento Relativo.

7. Información del movimiento: Esta zona muestra diferentes datos sobre el movimiento ejecutado, como Tiempo de ejecución, Distancia recorrida por cada eje, entre otras.
8. RUN: Una vez se terminan de configurar todas las variables del movimiento, se presiona este botón para que el movimiento sea ejecutado.

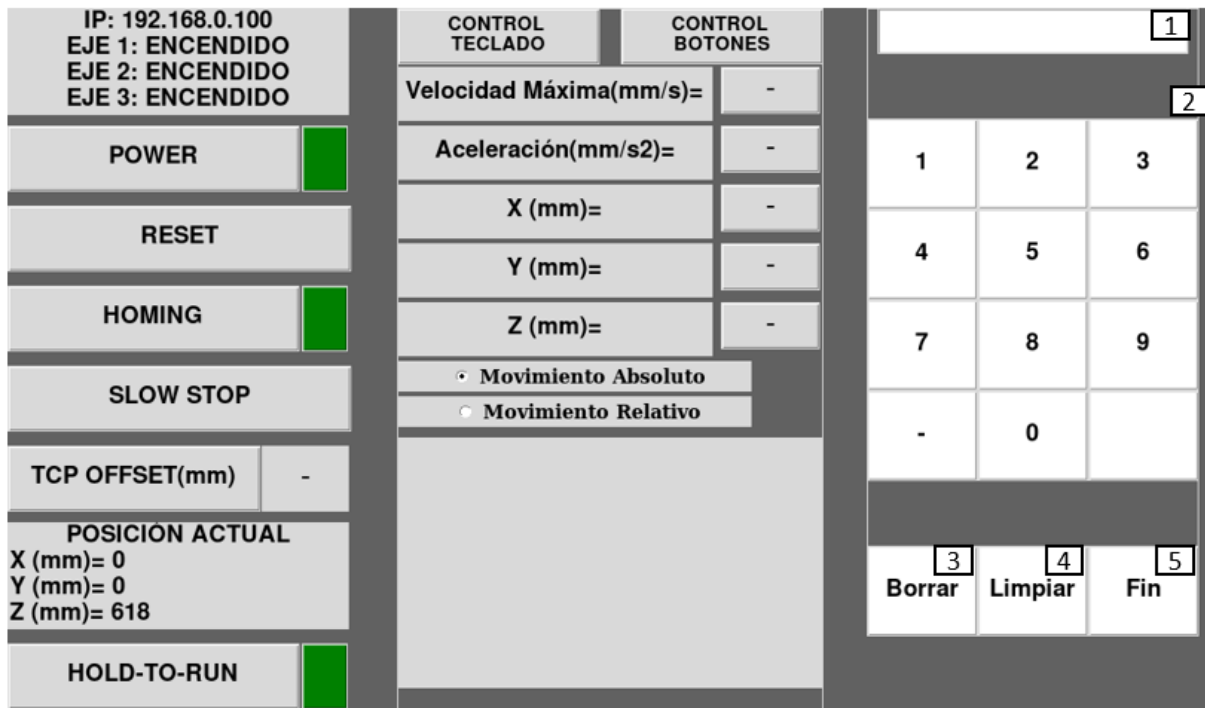


Figura 8.4: Teclado utilizado para el control

Al momento de seleccionar una de las variables Velocidad, Aceleración, X, Y, Z se despliega, en la sección derecha de la pantalla, un teclado en la pantalla. Este se puede ver en la figura 8.4 y sus funciones se explicarán a continuación.

1. Cuadro de texto: Se muestran los dígitos escritos por el usuario.
2. Teclado: Son los botones con los que el usuario podrá escribir el número que desea.
3. Borrar: Este botón borra el último dígito escrito.
4. Limpiar: Este botón borra todo el número escrito.
5. Fin: Se utiliza para finalizar la redacción del número y este sea guardado en la variable seleccionada.

### 8.1.3. Modo Manual alta velocidad

Se implementó este modo, según el diagrama de estados de la figura [A.2](#). Su funcionamiento es similar al modo Manual baja velocidad, pero a diferencia de este, permite que el robot ejecute movimiento a velocidades más altas, en un rango de  $255 \frac{mm}{s}$  hasta  $3000 \frac{mm}{s}$ . Este sistema se ejecuta en una Raspberry Pi 4, la cual le envía al PLC las variables que deben cambiar y su valor, para así ejecutar las funciones que el usuario desea.

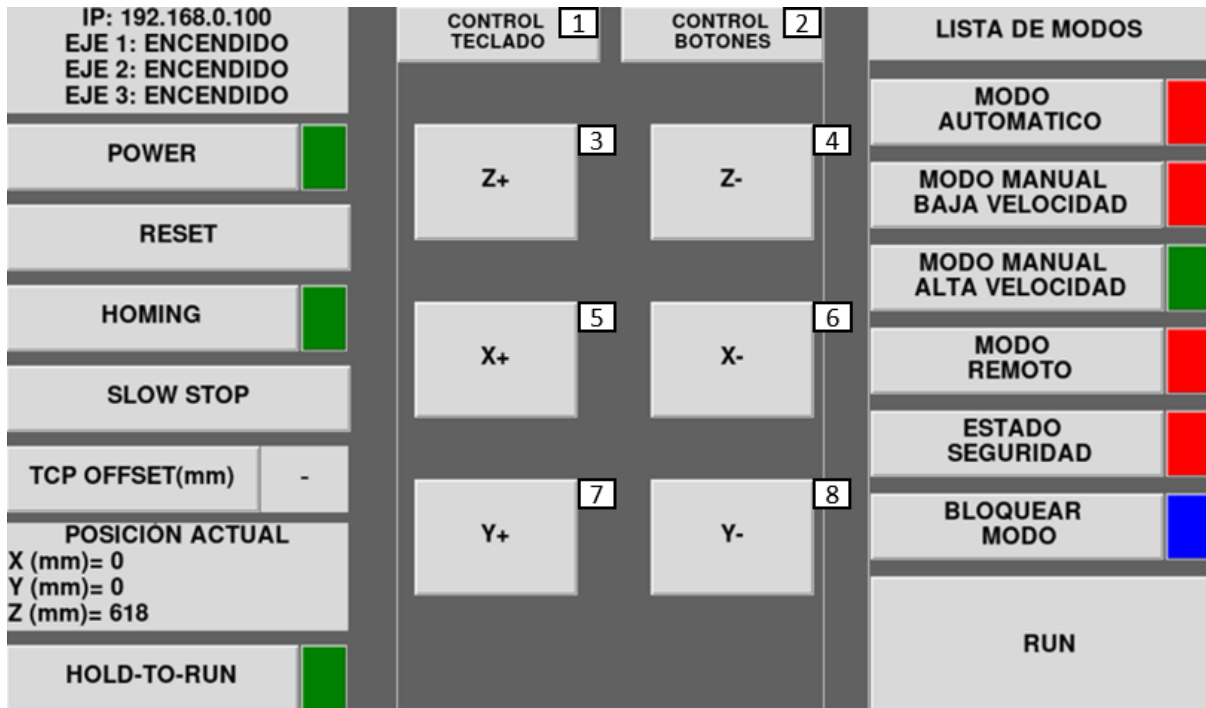


Figura 8.5: Interfaz gráfica modo Manual alta velocidad (Control ejes individuales)

En la figura [8.5](#) se pueden ver en la sección central diferentes actuadores, estos se utilizan para especificar a donde debe moverse el robot. A continuación se describe sus funciones.

1. CONTROL TECLADO: Se utiliza para que el usuario seleccione si desea que el control se realice ingresando manualmente las coordenadas por teclado.
2. CONTROL BOTONES: Se utiliza para que el usuario seleccione si desea que el control se realice moviendo individualmente cada eje (X, Y, Z).
3. Z+: Se mueve la brida del robot en el sentido Z positivo. Viendo el robot desde el frente, es un movimiento hacia arriba.
4. Z-: Se mueve la brida del robot en el sentido Z negativo. Viendo el robot desde el frente, es un movimiento hacia abajo.

5. X+: Se mueve la brida del robot en el sentido X positivo. Viendo el robot desde el frente, es un movimiento hacia la izquierda.
6. X-: Se mueve la brida del robot en el sentido X negativo. Viendo el robot desde el frente, es un movimiento hacia la derecha.
7. Y+: Se mueve la brida del robot en el sentido Y positivo. Viendo el robot desde el frente, es un movimiento hacia atrás.
8. Y-: Se mueve la brida del robot en el sentido Y negativo. Viendo el robot desde el frente, es un movimiento hacia adelante.

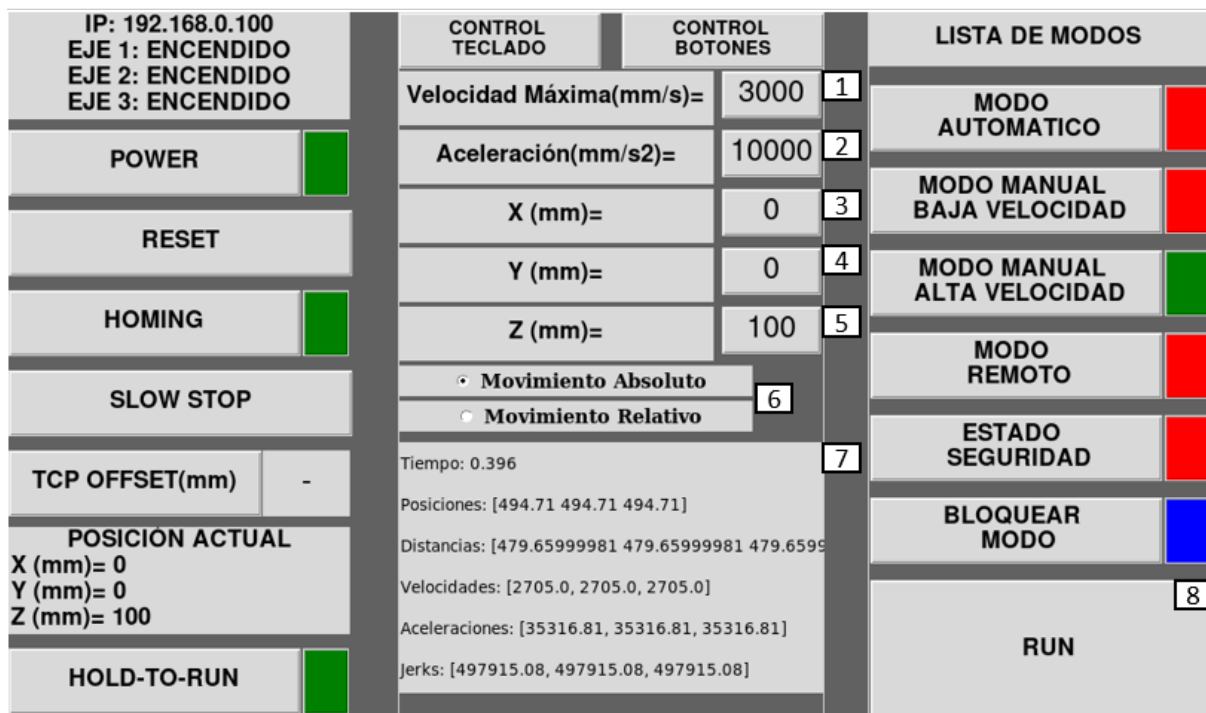


Figura 8.6: Interfaz gráfica modo Manual alta velocidad (Control por posición)

En la figura [8.6](#) se pueden ver, en la sección central, actuadores diferentes a los de la figura [8.5](#), la cual se describió anteriormente. A continuación se describirán estos actuadores, los cuales se habilitan con el modo de control por posición.

1. Velocidad Máxima(mm/s): Se establece la velocidad con la que se ejecutará el movimiento, esta velocidad es como máximo 3000(mm/s) en este modo. Se le notificará al usuario si la velocidad deseada es mayor a la permitida.

2. Aceleración( $\text{mm/s}^2$ ): Se selecciona la aceleración con la que se ejecutará el movimiento, esta aceleración es como máximo  $10.000(\text{mm/s}^2)$ .
3. X(mm): Se establece la coordenada en el eje X donde se ubicará el robot al ejecutar el movimiento. Se le notificará al usuario si esta ubicación está fuera del área de trabajo.
4. Y(mm): Se define la coordenada en el eje Y donde se ubicará el robot al ejecutar el movimiento. Se le notificará al usuario si esta ubicación está fuera del área de trabajo.
5. Z(mm): Se determina la coordenada en el eje Z donde se ubicará el robot al ejecutar el movimiento. Se le notificará al usuario si esta ubicación está fuera del área de trabajo.
6. Selector de tipo de movimiento: El usuario puede seleccionar si desea que el robot ejecute un movimiento Absoluto o un movimiento Relativo.
7. Información del movimiento: Esta zona muestra diferentes datos sobre el movimiento ejecutado, como Tiempo de ejecución, Distancia recorrida por cada eje, entre otras.
8. RUN: Una vez se terminan de configurar todas las variables del movimiento, se presiona este botón para que el movimiento sea ejecutado.

## 8.1.4. Modo Automático

Se implementó este modo, según el diagrama de estados de la figura [A.3](#), donde se muestra el flujo de funcionamiento del mismo. Este sistema se ejecuta en una Raspberry Pi 4, la cual le envía al PLC qué variables deben cambiar y a qué valor, para ejecutar las funciones que el usuario desea.

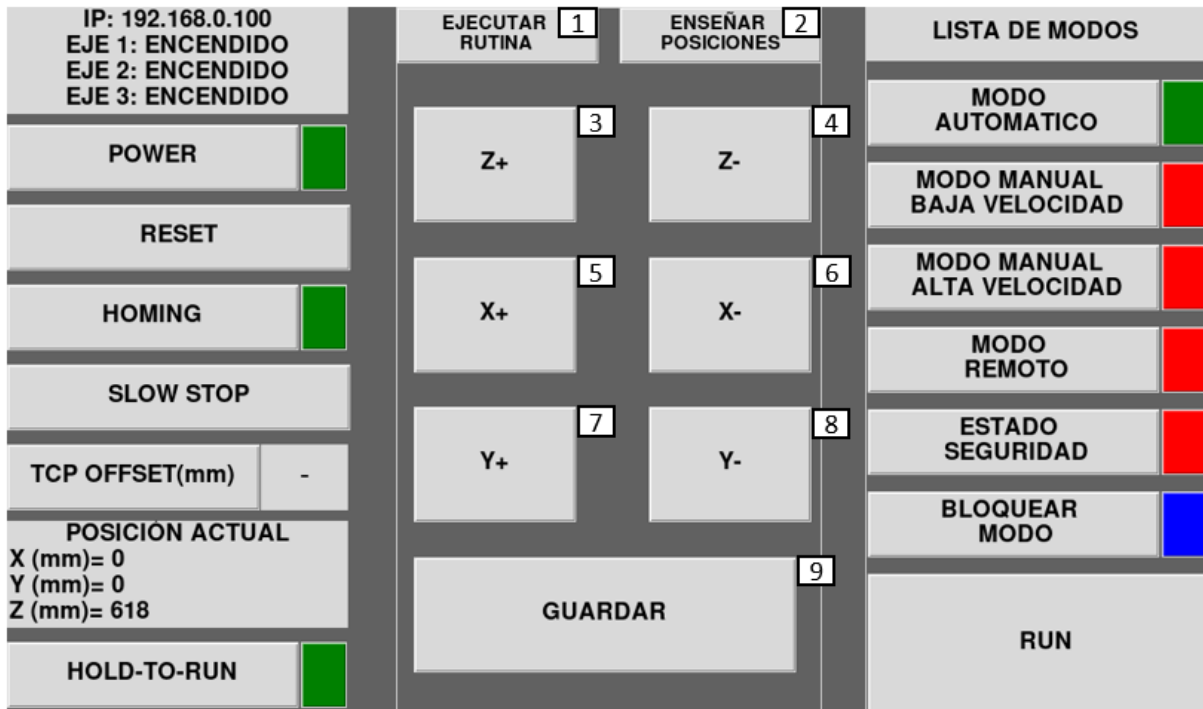


Figura 8.7: Interfaz gráfica modo Automático (ENSEÑAR POSICIONES)

En la figura [8.7](#) se pueden ver en la sección central diferentes actuadores, estos se utilizan para especificar a donde debe moverse el robot. A continuación se describe sus funciones.

1. EJECUTAR RUTINA: Le permite al usuario dar inicio a la rutina que previamente se le fue enseñada al sistema robotico.
2. ENSEÑAR POSICIONES: Se utiliza para entrar en el módulo de control, donde le permite al usuario hacer movimientos de forma manual y guardar estas posiciones en una rutina.
3. Z+: Se mueve la brida del robot en el sentido Z positivo. Viendo el robot desde el frente, es un movimiento hacia arriba.
4. Z-: Se mueve la brida del robot en el sentido Z negativo. Viendo el robot desde el frente, es un movimiento hacia abajo.
5. X+: Se mueve la brida del robot en el sentido X positivo. Viendo el robot desde el frente, es un movimiento hacia la izquierda.

6. X-: Se mueve la brida del robot en el sentido X negativo. Viendo el robot desde el frente, es un movimiento hacia la derecha.
7. Y+: Se mueve la brida del robot en el sentido Y positivo. Viendo el robot desde el frente, es un movimiento hacia atrás.
8. Y-: Se mueve la brida del robot en el sentido Y negativo. Viendo el robot desde el frente, es un movimiento hacia adelante.
9. GUARDAR: Le permite al usuario guardar la posición actual del robot, que se guardará en una rutina que se ejecutará en EJECUTAR RUTINA.

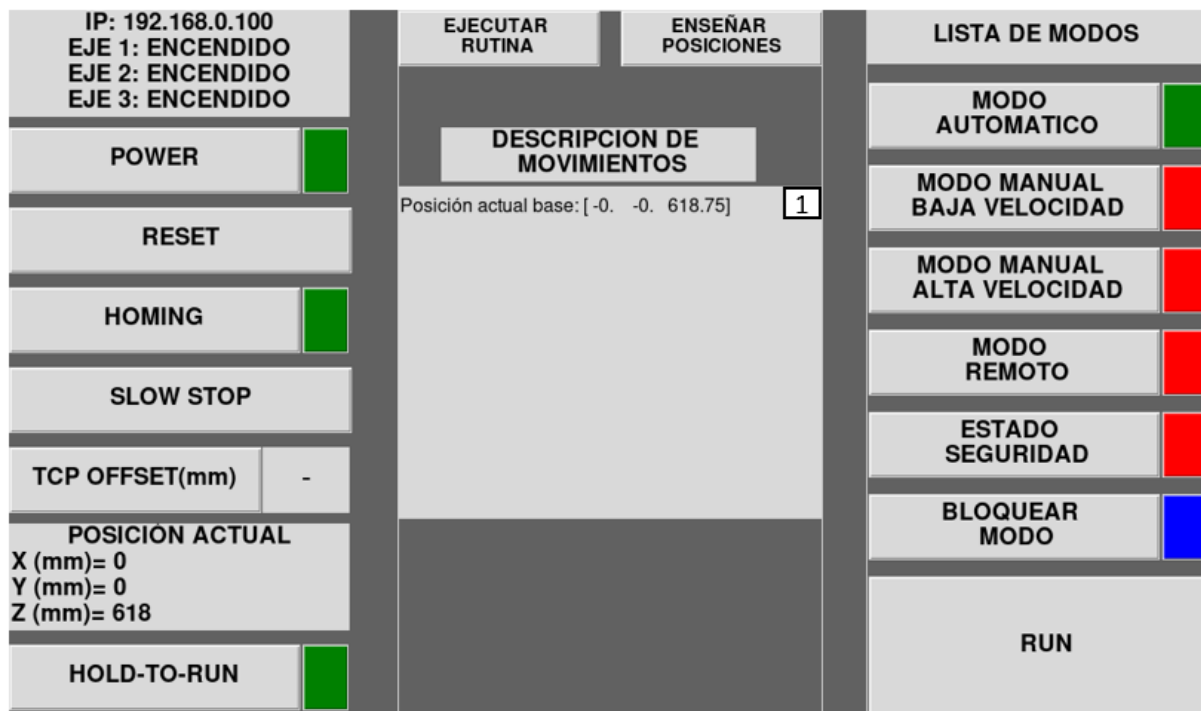


Figura 8.8: Interfaz gráfica modo Automático (EJECUTAR RUTINA)

En la figura [8.8](#) se pueden ver, en la sección central, actuadores diferentes a los de la figura [8.7](#), la cual se describió anteriormente. A continuación se describirán estos actuadores, los cuales se habilitan con el modo de control por posición.

1. DESCRIPCIÓN DE MOVIMIENTOS: En esta zona rectangular se mostrará información sobre el movimiento que se está ejecutando en la rutina.

### 8.1.5. Modo Remoto

Se implementó este modo, según el diagrama de estados de la figura [A.4](#), donde se muestra el flujo de funcionamiento del mismo. Este modo, a diferencia de los demás, se ejecuta en su aplicación en un computador. Esta aplicación está elaborada en el lenguaje de programación Python y cuenta con una GUI para su uso. Para utilizar la función de modo remoto, el computador donde se ejecutará debe conectarse a la red local donde están conectados los otros dispositivos, como lo son el pendant y el PLC. Luego, se abre la aplicación, que tiene dos funcionalidades, las cuales son visualización y control.

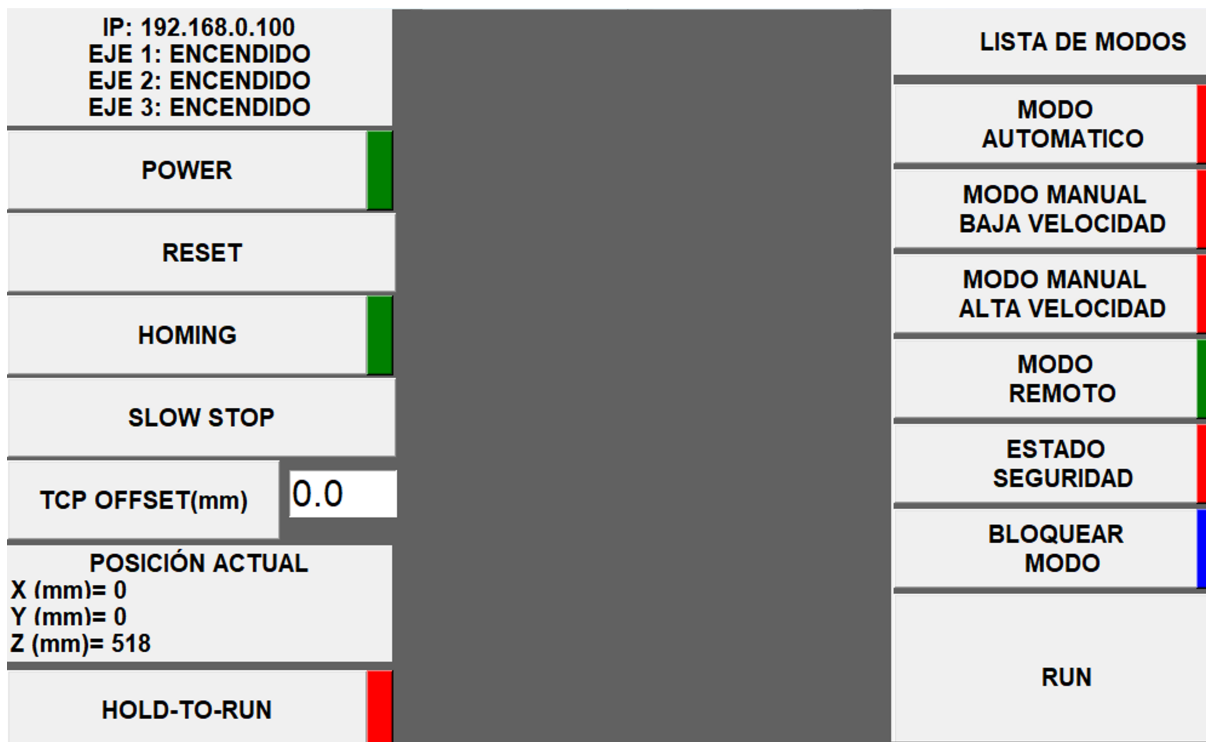


Figura 8.9: Interfaz gráfica modo Remoto (Visualización)

En la figura [8.9](#) se logra ver la función de visualización, donde se le muestran al usuario diferentes variables en tiempo real del sistema robótico, como lo son el modo en el que se encuentra, en que posición, el estado del Hold-To-Run, estado del homing y estado de los motores, si se encuentran encendidos o con un error. La visualización se mantiene mientras el robot se encuentre en un modo diferente al *Remoto*, lo que quiere decir que el operario está ejerciendo control desde el teach pendant. Por esta razón, no se podrá ejecutar ninguna acción desde la aplicación de control remoto, acciones como cambiar de modo, encender o apagar los motores, realizar la rutina de homing o realizar un movimiento solo son posibles desde el teach pendant mientras se encuentre en otro modo.

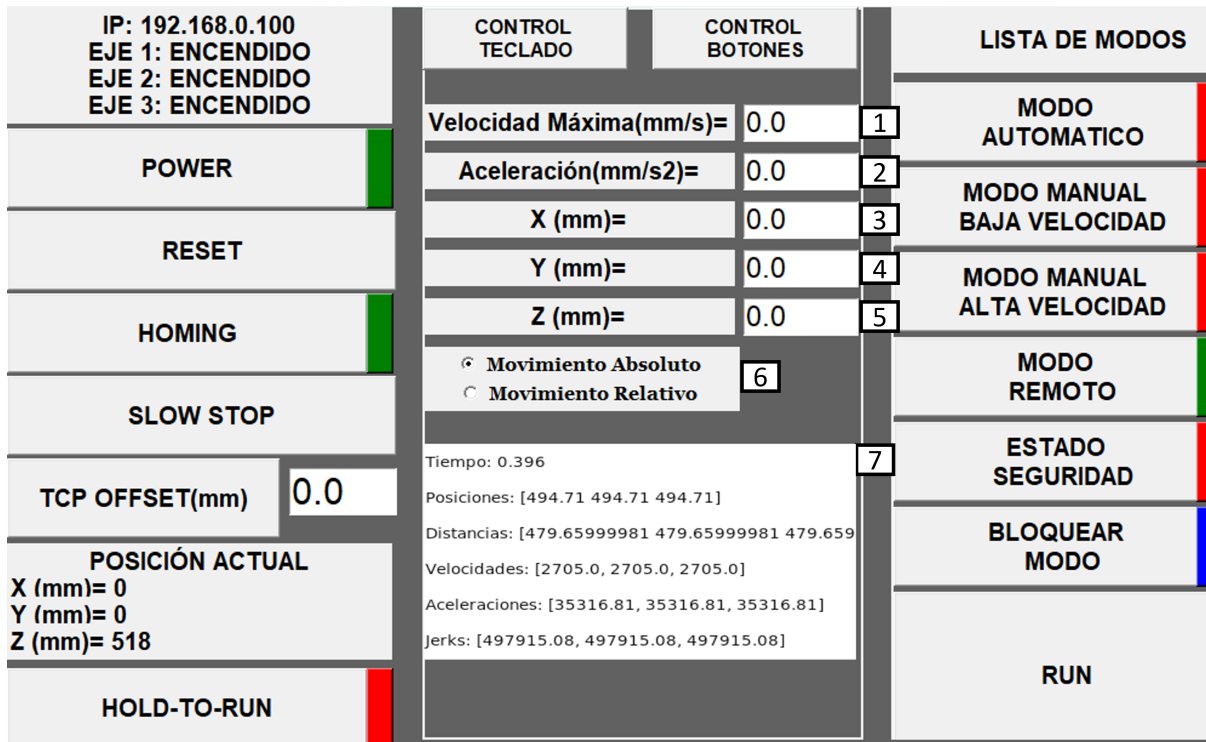


Figura 8.10: Interfaz gráfica modo Remoto (Control por posición)

Por otro lado, la función de control le permite al usuario realizar movimientos desde la aplicación de forma remota. Se activa manualmente desde el teach pendant, donde el operario debe cambiar a esta configuración, deshabilitando así algunas funciones en el dispositivo, como lo son el control de movimientos y los botones del panel izquierdo, que solo solo se podrán ejecutar desde la aplicación. Cabe resaltar que el cambio de modo no se altera, si el usuario desea hacer esa transición, debe hacerlo manualmente desde el pendant. En la figura 8.10 se muestra la interfaz de control, que en la sección central tiene diferentes entradas de texto, las cuales reciben dígitos desde el teclado del dispositivo. Estas se utilizan para establecer las características del movimiento que el operario desea que el robot ejecute. A continuación se describen sus funciones.

1. Velocidad Máxima(mm/s): Se establece la velocidad con la que se ejecutará el movimiento, esta velocidad es como máximo 250(mm/s) en este modo. Se le notificará al usuario si la velocidad deseada es mayor a permitida.
2. Aceleración( mm/s<sup>2</sup>): Se selecciona la aceleración con la que se ejecutará el movimiento, esta aceleración es como máximo 10.000( mm/s<sup>2</sup>).
3. X(mm): Se establece la coordenada en el eje X donde se ubicará el robot al ejecutar el movimiento. Se le notificará al usuario si esta ubicación está fuera del área de trabajo.

4. Y(mm): Se define la coordenada en el eje Y donde se ubicará el robot al ejecutar el movimiento. Se le notificará al usuario si esta ubicación está fuera del área de trabajo.
5. Z(mm): Se determina la coordenada en el eje Z donde se ubicará el robot al ejecutar el movimiento. Se le notificará al usuario si esta ubicación está fuera del área de trabajo.
6. Selector de tipo de movimiento: El usuario puede seleccionar si desea que el robot ejecute un movimiento Absoluto o un movimiento Relativo.
7. Información del movimiento: Esta zona muestra diferentes datos sobre el movimiento ejecutado, como Tiempo de ejecución, Distancia recorrida por cada eje, entre otras.

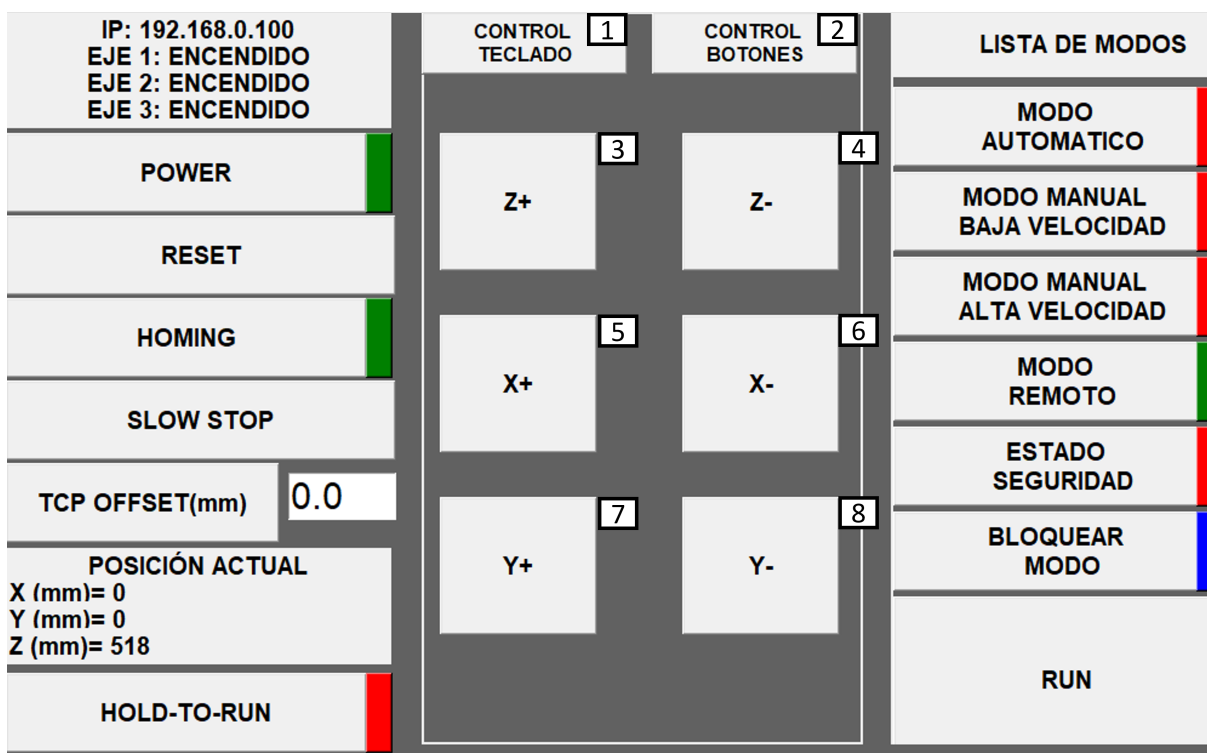


Figura 8.11: Interfaz gráfica modo Manual alta velocidad (por eje)

En la figura 8.8 se pueden ver, en la sección central, actuadores diferentes a los de la figura 8.7, la cual se describió anteriormente. A continuación se describirán estos actuadores, los cuales se habilitan con el modo de control por posición.

1. CONTROL TECLADO: Se utiliza para que el usuario seleccione si desea que el control se realice ingresando manualmente las coordenadas por teclado.
2. CONTROL BOTONES: Se utiliza para que el usuario seleccione si desea que el control se realice moviendo individualmente cada eje (X,Y,Z).

3. Z+: Se mueve la brida del robot en el sentido Z positivo. Viendo el robot desde el frente, es un movimiento hacia arriba.
4. Z-: Se mueve la brida del robot en el sentido Z negativo. Viendo el robot desde el frente, es un movimiento hacia abajo.
5. X+: Se mueve la brida del robot en el sentido X positivo. Viendo el robot desde el frente, es un movimiento hacia la izquierda.
6. X-: Se mueve la brida del robot en el sentido X negativo. Viendo el robot desde el frente, es un movimiento hacia la derecha.
7. Y+: Se mueve la brida del robot en el sentido Y positivo. Viendo el robot desde el frente, es un movimiento hacia atrás.
8. Y-: Se mueve la brida del robot en el sentido Y negativo. Viendo el robot desde el frente, es un movimiento hacia adelante.

#### 8.1.6. Estado de seguridad

Se implementó este estado, según el diagrama de estados de la figura [A.5](#), que muestra el flujo de funcionamiento del mismo. Este estado se diseñó para detener el sistema en caso de que el robot presente una falla en su funcionamiento o no cumpla con las condiciones de seguridad necesarias para garantizar la integridad del operario o del mismo. Por esta razón, no se considera como un modo, pues no le da al usuario herramientas para llevar a cabo una tarea, por el contrario, detiene la actividad del robot enseguida.

En la figura [8.7](#) se pueden ver que el sistema arroja notificaciones en pantalla para alertar al usuario e informarle cuál es el error. A continuación se describen sus funciones.

1. Indicador de HOLD-TO-RUN: Este indicador muestra si el switch de tres posiciones, ubicado en el pendant, se encuentra en la posición correcta, esta es la segunda posición. Si el dispositivo no está presionado o está muy presionado, el robot entrar en el estado de seguridad y detendrá su funcionamiento. El indicador se tornara color rojo, si se encuentra en la posición uno o tres, y se tornara verde, si se encuentra en la posición dos.
2. Indicador de Estado de seguridad: Se activa el estado de seguridad cuando no se cumplen con las condiciones de seguridad para el funcionamiento del robot.
3. Notificaciones: Se desplegará una notificación en pantalla describiendo la razón por la que el estado de seguridad se activó. Esta razón puede ser debido al HOLD-TO-RUN, Error en los servomotores o condiciones externas como iniciar la rutina para abrir la puerta de la celda robótica.

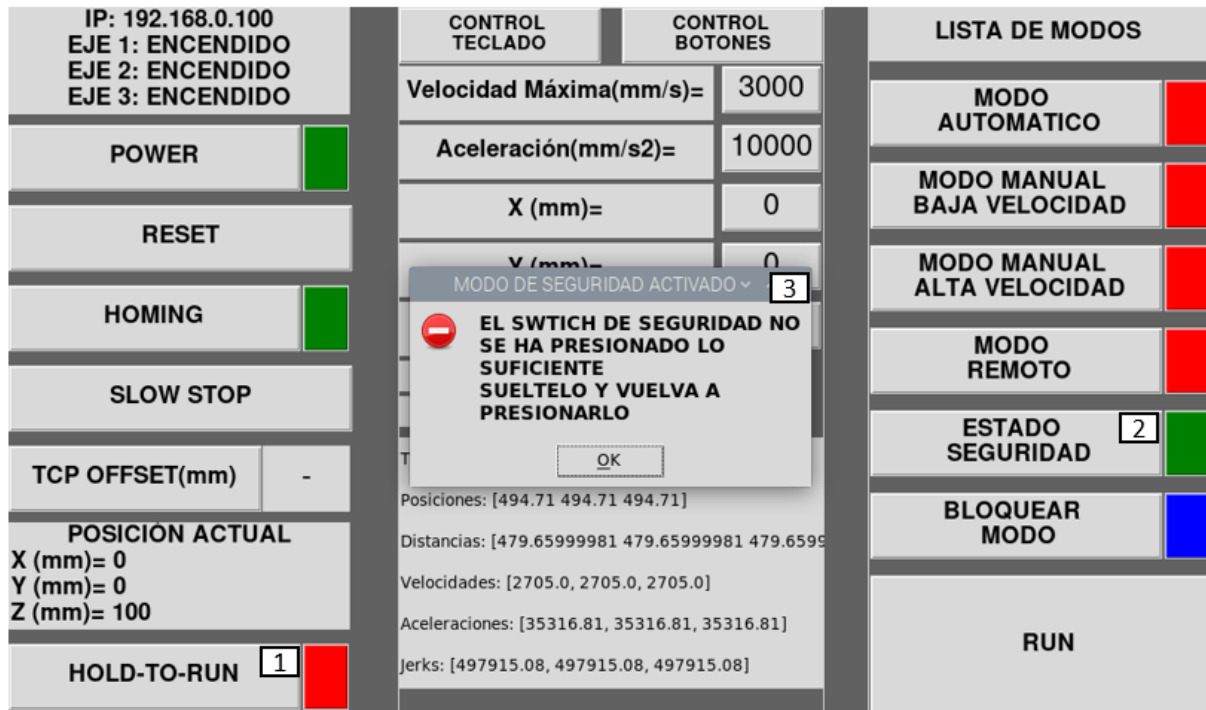


Figura 8.12: Interfaz gráfica Estado de seguridad

### 8.1.7. Programación del PLC

Debido a que el PLC(Programmable Logic Controller) tiene la función controlar los drivers y motores para que estos realicen los movimientos solicitados, además de monitorear el estado de diferentes variables y actuadores conectados a sus entradas, se realizó la programación adecuada para que el sistema logre realizar todas las tareas solicitadas.

Se elaboró un programa, el cual es ejecutado como tarea principal en el PLC, dentro de este se describe el funcionamiento de la máquina de estados, la cual controla el paso de un modo a otro y se asegura que solo se encuentre en un estado a la vez. Además, se elaboraron bloques de funciones para ejecutar diferente acciones como la detención por emergencia, parada suave o el monitoreo de actuadores como el switch de seguridad. A continuación se detallará cada función elaborada para la ejecución de cada acción.

1. DO\_HOMING: Esta función se ejecuta para realizar la rutina de homing y de esta forma encontrar la posición inicial el robot.

Se puede ver en la figura [8.13](#) el código elaborado para llevar a cabo la tarea de esta función. Se utiliza la herramienta de Motion control para comunicarle a los drivers qué acción ejecutar. Por medio de la función `MC_Home_x()`, de Motion control, la cual le ordena al eje que debe realizar la secuencia de búsqueda de la posición inicial, para esto se deben definir mínimo cuatro parámetros de entrada:

```

1  (*
2  TIPO DE HOMING
3  TIPO=0: Proceso buscando el interruptor de límite
4  TIPO=4: Proceso estático. Se define la posición actual como la ingresada (En este caso 0.0).
5  *)
6
7  TIPO := 0;
8  MC_Home_1(EjeA1, IN, VALOR_HOME, TIPO,);
9  MC_Home_2(EjeA2, IN, VALOR_HOME, TIPO,);
10 MC_Home_3(EjeA3, IN, VALOR_HOME, TIPO,);
11
12 AXIS_HOMED := EjeA1.AxisHomed AND EjeA2.AxisHomed AND EjeA3.AxisHomed;
13 ERROR:= MC_Home_1.Error OR MC_Home_2.Error OR MC_Home_3.Error;
14 DONE:= MC_Home_1.Done AND MC_Home_2.Done AND MC_Home_3.Done;
15

```

Figura 8.13: Código de la función DO\_HOMING

- a) *AxisIn*: Define en que eje se realizará la secuencia, cada eje es un driver diferente. En este caso, se utilizaron las variables globales *EjeA1*, *EjeA2*, *EjeA3* para definir cada uno de los ejes. Cada una de estas variables es de tipo *AXIS\_REF*.
- b) *Execute*: Este parámetro define si la secuencia de homing debe realizarse o no, si el parámetro se define como *TRUE*, se ejecutará y si se define como *FALSE*, no se ejecutara. En este caso, se utilizó la variable de entrada *IN* para definir cuando realizar la rutina, esta variable es de tipo *BOOL*.
- c) *Position*: Define la dirección en la que se moverá el eje para encontrar la posición de "home". En este caso, se utilizó la variable global *VALOR\_HOME*, esta variable es de tipo *REAL*.
- d) *HomingMode*: Define que tipo de homing se realizará, donde el tipo 0 busca el interruptor de límite físico y el tipo 4 establece la posición actual de los motores como posición inicial. En este caso, se utilizó la variable *TIPO*, esta variable es de tipo *SINT* y su valor se estableció en 0.

Finalmente, se utilizan la variable global *AXIS\_HOMED* y variables de salida *ERROR* y *DONE* para guardar el estado de diferentes eventos.

- a) *AXIS\_HOMED*: Su valor es *TRUE* cuando todos los ejes terminaron de realizar la rutina de homing y es *FALSE* cuando al menos uno de los ejes no terminó de realizar la rutina.
  - b) *ERROR*: Su valor es *TRUE* cuando al menos uno de los ejes presentó un error al realizar la rutina y es *FALSE* cuando todos los ejes terminaron de realizar la rutina correctamente.
  - c) *DONE*: Su valor es *TRUE* cuando todos los ejes terminaron de realizar la rutina de homing y es *FALSE* cuando al menos uno de los ejes aún está realizando la rutina.
2. *DO\_MOVE*: Esta función se utiliza para que el robot realice un movimiento. Para esto, se le indica a los drivers que cada motor debe moverse una determinada distancia, en una determinada orientación y a una determinada velocidad.

```

1  (*Función para ejecutar movimientos a posiciones absolutas en los tres ejes*)
2  (* MC_Move(Eje,Activacion,Distancia,Velocidad,Aceleracion,Desaceleracion,Jerk,,); *)
3  MC_MoveAbsolute_1(EjeA1,IN,D_EJE1,V_EJE1,A_EJE1,A_EJE1,J_EJE1,,);
4  RTO_1(MC_MoveAbsolute_1.Busy,RESET_TIME,T#10s);
5  TIEMPO_EJE1:= RTO_1.ET;
6  MC_MoveAbsolute_2(EjeA2,IN,D_EJE2,V_EJE2,A_EJE2,A_EJE2,J_EJE2,,);
7  RTO_2(MC_MoveAbsolute_2.Busy,RESET_TIME,T#10s);
8  TIEMPO_EJE2:= RTO_2.ET;
9  MC_MoveAbsolute_3(EjeA3,IN,D_EJE3,V_EJE3,A_EJE3,A_EJE3,J_EJE3,,);
10 RTO_3(MC_MoveAbsolute_3.Busy,RESET_TIME,T#10s);
11 TIEMPO_EJE3:= RTO_3.ET;
12 ERROR:= MC_MoveAbsolute_1.Error OR MC_MoveAbsolute_2.Error OR MC_MoveAbsolute_3.Error;
13 DONE:= MC_MoveAbsolute_1.Done AND MC_MoveAbsolute_2.Done AND MC_MoveAbsolute_3.Done;
14
15

```

Figura 8.14: Código de la función DO\_MOVE

Se puede ver en la figura [8.14](#) el código desarrollado que logra que el robot ejecute un desplazamiento. Por medio de la función *MC\_MoveAbsolute\_x()*, de Motion control, la cual le ordena realizar un movimiento controlado a una posición absoluta especificada, para esto se deben definir mínimo siete parámetros de entrada:

- a) *AxisIn*: Define en que eje se realizará la secuencia, cada eje es un driver diferente. En este caso, se utilizaron las variables globales *EjeA1*, *EjeA2*, *EjeA3* para definir cada uno de los ejes. Cada una de estas variables es de tipo *AXIS\_REF*.
- b) *Execute*: Este parámetro define si el movimiento debe realizarse o no, donde si el parámetro se define como *TRUE*, se ejecutara y si se define como *FALSE*, no se ejecutara. En este caso, se utilizó la variable de entrada *IN* para definir cuando realizar la rutina, esta variable es de tipo *BOOL*.
- c) *Position*: Define la posición absoluta a la que el eje debe moverse, este valor se encuentra en la misma unidad de medida que el encoder. En este caso, se utilizó la variable global *D\_EJEx*, esta variable es de tipo *REAL*.
- d) *Velocity*: Define la velocidad a la que se realizara el movimiento. En este caso, se utilizó la variable *V\_EJEx*, esta variable es de tipo *REAL*.
- e) *Acceleration*: Este parámetro establece la tasa de cambio de la velocidad durante la etapa de aceleración del movimiento. En este caso, se utilizó la variable *A\_EJEx*, esta variable es de tipo *REAL*.
- f) *Deceleration*: Este parámetro establece la tasa de cambio de la velocidad durante la etapa de desaceleración del movimiento. En este caso, se utilizó la variable *A\_EJEx*, esto quiere decir que en la etapa de aceleración y desaceleración se tendrá la misma tasa de cambio, esta variable es de tipo *REAL*.
- g) *Jerk*: Establece la tasa de cambio de la aceleración, lo cual en el movimiento se traduce como que tan suavemente se inicia y detiene el movimiento. En este caso, se utilizó la variable *J\_EJEx*, esta variable es de tipo *REAL*.

Por otro lado, se utiliza un temporizador RTO y se utiliza para medir cuanto tiempo se tarda cada eje en realizar el desplazamiento solicitado de forma individual. Este valor se guarda en las variables globales *TIEMPO\_EJE1*, *TIEMPO\_EJE2*, *TIEMPO\_EJE3*.

Finalmente, se utilizan las variables de salida *ERROR* y *DONE* para guardar el estado de diferentes eventos.

- a) *ERROR*: Su valor es *TRUE* cuando al menos uno de los ejes presentó un error al realizar el movimiento y es *FALSE* cuando todos los ejes realizaron el desplazamiento correctamente.
  - b) *DONE*: Su valor es *TRUE* cuando todos los ejes completarán el desplazamiento y es *FALSE* cuando al menos uno de los ejes aún se encuentra en movimiento.
3. *DO\_RESET*: Esta función se corre para reiniciar o restablecer los tres ejes, en caso de un error. De esta forma, se logra que el sistema retorne su funcionamiento normal.

```

1  (*Función para reestablecer en caso de error*)
2  MC_Reset_1(EjeA1, IN);
3  MC_Reset_2(EjeA2, IN);
4  MC_Reset_3(EjeA3, IN);
5  ERROR:= MC_Reset_1.Error OR MC_Reset_2.Error OR MC_Reset_3.Error;
6  DONE:= MC_Reset_1.Done AND MC_Reset_2.Done AND MC_Reset_3.Done;
7

```

Figura 8.15: Código de la función *DO\_RESET*

Se puede ver en la figura [8.15](#) el código que logra llevar a cabo el reinicio de los ejes. Se utiliza la función *MC\_Reset\_x()*, de Motion control, la cual restablece todos los errores internos relacionados con un eje específico, se deben definir mínimo dos parámetros de entrada:

- a) *AxisIn*: Define que eje se reiniciará, cada eje es un driver diferente. En este caso, se utilizaron las variables globales *EjeA1*, *EjeA2*, *EjeA3* para definir cada uno de los ejes. Cada una de estas variables es de tipo *AXIS\_REF*.
- b) *Execute*: Este parámetro define si el reinicio debe ejecutarse o no, si se define como *TRUE*, se ejecutará y si se define como *FALSE*, no se ejecutará. En este caso, se utilizó la variable de entrada *IN* para definir cuando realizar la rutina, esta variable es de tipo *BOOL*.

Finalmente, se utilizan las variables de salida *ERROR* y *DONE* para guardar el estado de diferentes eventos.

- a) *ERROR*: Su valor es *TRUE* cuando al menos uno de los ejes presentó un error al reiniciarse y es *FALSE* cuando todos los ejes se reiniciaron correctamente.

- b) DONE: Su valor es *TRUE* cuando todos los ejes se reiniciaron y es *FALSE* cuando al menos uno de los ejes aún no se ha reiniciado.
4. DO\_POWER: Esta función se utiliza para encender o apagar los drivers y motores. Esto logra que el usuario tenga la posibilidad de controlar la alimentación de los motores y cortarla si es necesario de forma rápida.

```

1  (*Función para encender y apagar los ejes*)
2  MC_Power_1(EjeA1, IN, TRUE, TRUE);
3  MC_Power_2(EjeA2, IN, TRUE, TRUE);
4  MC_Power_3(EjeA3, IN, TRUE, TRUE);
5  _IO_EM_DO_13:= IN;
6  _IO_EM_DO_14:= IN;
7  _IO_EM_DO_15:= IN;
8  ERROR:= MC_Power_1.Error OR MC_Power_2.Error OR MC_Power_3.Error;
9  ENCENDIDO:= MC_Power_1.Status AND MC_Power_2.Status AND MC_Power_3.Status;
10

```

Figura 8.16: Código de la función DO\_POWER

Se puede ver en la figura [8.16](#) el código que logra llevar a cabo esta función. Se utiliza la función *MC\_Power\_x()*, de Motion control, la cual enciende o paga un eje, se deben definir mínimo cuatro parámetros de entrada:

- a) AxisIn: Define que eje se reiniciara, cada eje es un driver diferente. En este caso, se utilizaron las variables globales *EjeA1*, *EjeA2*, *EjeA3* para definir cada uno de los ejes. Cada una de estas variables es de tipo *AXIS\_REF*.
- b) Enable: Este parámetro define si el eje debe apagarse o no, donde si el parámetro se define como *TRUE*, se encenderá y si se define como *FALSE*, se apagará. En este caso, se utilizó la variable de entrada *IN* para definir cuando realizar la rutina, esta variable es de tipo *BOOL*.
- c) Enable\_positive: Este parámetro está relacionado con los movimientos en sentido positivo del eje, donde si el parámetro se define como *TRUE*, se habilitara al eje a hacer movimientos en sentido positivo y si se define como *FALSE*, no se habilitara. En este caso, se asignó el valor contante *TRUE*.
- d) Enable\_negative: Este parámetro está relacionado con los movimientos en sentido negativo del eje, donde si el parámetro se define como *TRUE*, se habilitara al eje a hacer movimientos en sentido negativo y si se define como *FALSE*, no se habilitara. En este caso, se asignó el valor contante *TRUE*.

Por otro lado, se utilizan las salidas del PLC número 13, 14 y 15 para cortar la alimentación de los servomotores. La salida del PLC controla el estado de un relé de estado sólido que a su vez controla un contactor que controla la alimentación de un servomotor, esta arquitectura se

repite en cada una de las tres salidas, lo cual proporciona un control sobre la alimentación de los tres servomotores.

Finalmente, se utilizan las variables de salida *ERROR* y *ENCENDIDO* para guardar el estado de diferentes eventos.

- a) *ERROR*: Su valor es *TRUE* cuando al menos uno de los ejes presentó un error al encenderse o apagarse y es *FALSE* cuando todos los ejes se encendieron o apagaron correctamente.
  - b) *ENCENDIDO*: Su valor es *TRUE* cuando todos los ejes se encuentran encendidos y es *FALSE* todos los ejes se encuentran apagados.
5. *DO\_SLOW\_SLOP*: Esta función se utiliza para ordenar al robot que ejecute la rutina de parada suave. Esto le da al usuario la posibilidad que realizar una parada que no va a ser abrupta, lo cual protege al robot de daños en sus mecanismos.

```

1  (*Función para realizar parada desacelerada*)
2  DECELERATION:= 1000.0;
3  JERK:= 100000.0;
4  MC_Halt_1(EjeA1, IN, DECELERATION, JERK, );
5  MC_Halt_2(EjeA2, IN, DECELERATION, JERK, );
6  MC_Halt_3(EjeA3, IN, DECELERATION, JERK, );
7  ERROR:= MC_Halt_1.Error OR MC_Halt_2.Error OR MC_Halt_3.Error;
8  DONE:= MC_Halt_1.Done AND MC_Halt_2.Done AND MC_Halt_3.Done;
9

```

Figura 8.17: Código de la función *DO\_SLOW\_STOP*

Se puede ver en la figura [8.17](#) el código que logra llevar a cabo esta tarea. Se utiliza la función *MC\_Halt\_x()*, de Motion control, la cual ejecuta una acción de parada controlada, se deben definir mínimo cuatro parámetros de entrada:

- a) *AxisIn*: Define que eje se reiniciará, cada eje es un driver diferente. En este caso, se utilizaron las variables globales *EjeA1*, *EjeA2*, *EjeA3* para definir cada uno de los ejes. Cada una de estas variables es de tipo *AXIS\_REF*.
- b) *Execute*: Este parámetro define si la parada controlada debe realizarse o no, si el parámetro se define como *TRUE*, se ejecutará y si se define como *FALSE*, no se ejecutará. En este caso, se utilizó la variable de entrada *IN* para definir cuando realizar la rutina, esta variable es de tipo *BOOL*.
- c) *Deceleration*: Este parámetro establece la tasa de cambio de la velocidad durante la etapa de desaceleración del movimiento. En este caso, se utilizó la variable *DECELERATION*, de tipo *REAL* y se le asignó un valor de *1000.0*.

d) Jerk: Establece la tasa de cambio de la aceleración, lo cual en el movimiento se traduce como que tan suavemente se inicia y detiene el movimiento. En este caso, se utilizó la variable *JERK*, de tipo *REAL* y se le asignó un valor de *100000.0*.

Finalmente, se utilizan las variables de salida *ERROR* y *DONE* para guardar el estado de diferentes eventos.

- a) *ERROR*: Su valor es *TRUE* cuando al menos uno de los ejes presentó un error al realizar la parada y es *FALSE* cuando todos los ejes lograron realizar la parada correctamente.
- b) *DONE*: Su valor es *TRUE* cuando todos los ejes finalizaron con la acción de parada controlada y es *FALSE* cuando al menos uno de los ejes aún no ha finalizado con la acción.

6. *MONITOREO\_SWITCH*: Esta función se ejecuta constantemente, no depende de la acción del operario y su tarea es la de monitorear en que estado se encuentra el switch de seguridad, localizado en el teach pedant.

```

1  (*SIN PRESIONAR*)
2  (*ELSIF IN AND NOT _IO_EM_DI_24 AND (NOT _IO_EM_DI_25) AND OK_SWITCH THEN*)
3  IF IN AND _IO_EM_DI_25 AND _IO_EM_DI_26 AND NOT _IO_EM_DI_24 THEN
4      MANUAL_ERROR := 1;
5      //EMERGENCY_STOP := TRUE;
6      //OK_SWITCH := FALSE;
7
8  (*SWITCH PRESIONADO POSICION ENABLE*)
9  ELSIF IN AND _IO_EM_DI_24 AND _IO_EM_DI_26 AND NOT _IO_EM_DI_25 THEN
10     MANUAL_ERROR := 0;
11     OK_SWITCH := TRUE;
12
13
14  (*SWITCH PRESIONADO POSICION GRIP*)
15  ELSIF IN AND (NOT _IO_EM_DI_24) AND (NOT _IO_EM_DI_25) AND (NOT _IO_EM_DI_26) THEN
16     //EMERGENCY_STOP := TRUE;
17     MANUAL_ERROR := 3;
18  END_IF;
19
20  IF IN AND _IO_EM_DI_25 AND _IO_EM_DI_26 AND NOT _IO_EM_DI_24 AND OK_SWITCH THEN
21     EMERGENCY_STOP := TRUE;
22     OK_SWITCH := FALSE;
23  ELSIF IN AND (NOT _IO_EM_DI_24) AND (NOT _IO_EM_DI_25) AND (NOT _IO_EM_DI_26) AND OK_SWITCH THEN
24     EMERGENCY_STOP := TRUE;
25     OK_SWITCH := FALSE;
26  END_IF;

```

Figura 8.18: Código de la función *MONITOREO\_SWITCH*

Se puede ver en la figura [8.18](#) el código que logra llevar a cabo este monitoreo. Se divide en dos sentencias de *IF*, la primera que inicia en la línea 3 y finaliza en la línea 18, la segunda inicia en la fila 20 y finaliza en la fila 26.

La primera sentencia de *IF* posee tres condicionales, los cuales permiten diferenciar entre los tres estados del switch de seguridad, estos son *SUELTO*, *HABILITADO*, *TOTALMENTE*

*PRESIONADO*. Se utilizan las entradas al PLC I24, I25 e I26, entradas a las que se encuentran conectados los contactos del switch de seguridad. El primer condicional, el cual está ubicado en la línea 3 del código, representa el estado *SUELTO*, donde se debe cumplir que  $I24 = FALSE$ ,  $I25 = TRUE$ ,  $I26 = TRUE$ . Esto asignará el valor de "1.<sup>a</sup> la variable global *MANUAL\_ERROR*. El segundo condicional, el cual está ubicado en la línea 9 del código, representa el estado *HABILITADO*, donde se debe cumplir que  $I24 = TRUE$ ,  $I25 = FALSE$  y  $I26 = TRUE$ . Esto asignará el valor de 0 la variable global *MANUAL\_ERROR* y asigna el valor de *TRUE* la variable global *OK\_SWITCH*. El tercer condicional, el cual está ubicado en la línea 15 del código, representa el estado *TOTALMENTE PRESIONADO*, donde se debe cumplir que  $I24 = FALSE$ ,  $I25 = FALSE$  y  $I26 = FALSE$ . Esto asignará el valor de 3 la variable global *MANUAL\_ERROR*.

La segunda sentencia de IF posee dos condicionales, estos diferencian cuando el sistema debe detenerse y pasar al estado de seguridad. El primer condicional, el cual está ubicado en la línea 20 del código, donde se debe cumplir que  $I24 = FALSE$ ,  $I25 = TRUE$ ,  $I26 = TRUE$  y  $OK\_SWITCH = TRUE$ . Esta combinación de variables representa le cambio del estado *HABILITADO* a *SUELTO*, por lo tanto, el sistema debe detenerse y pasar al estado de seguridad. Esto asignará el valor de *FALSE* la variable global *OK\_SWITCH* y el valor *TRUE* a la variable global *EMERGENCY\_STOP*. El segundo condicional, el cual está ubicado en la línea 23 del código, donde se debe cumplir que  $I24 = FALSE$ ,  $I25 = FALSE$  y  $I26 = FALSE$  y  $OK\_SWITCH = TRUE$ . Esta combinación de variables representa le cambio del estado *HABILITADO* a *TOTALMENTE PRESIONADO*, por lo tanto, el sistema debe detenerse y pasar al estado de seguridad. Esto asignará el valor de *FALSE* la variable global *OK\_SWITCH* y el valor *TRUE* a la variable global *EMERGENCY\_STOP*.

7. *MONITOREO\_PUERTA*: Esta función se ejecuta constantemente, no depende de la acción del operario y su tarea es la de monitorear en que estado se encuentran los actuadores relacionados con el funcionamiento de la puerta de acceso a la celda robótica. De esta forma se garantiza que el robot no esté en movimiento cuando la puerta esté abierta, lo cual representa que hay un operario dentro.

Se puede ver en la figura [8.19](#) el código que logra llevar a cabo este monitoreo. Se divide en tres sentencias de IF, la primera que inicia en la línea 1 y finaliza en la línea 8, la segunda inicia en la fila 10 y finaliza en la fila 17, la tercera inicia en la fila 20 y finaliza en la fila 34. Se puede ver que se utilizan variables globales y entradas al PLC. Se utilizan las entradas I03, la cual es la salida del relé de seguridad; la I11, la cual está conectado al selector que controla si la puerta está asegurada e I13, la cual está conectada al contacto del interlock que representa si la puerta está cerrada.

La primera sentencia de IF posee dos condicionales. El primer condicional, el cual está ubicado en la línea 1 del código, el cual representa que el relé de seguridad apago su salida porque entro en estado de seguridad, esto se debe a que la puerta se abrió o se desaseguro. Se debe cumplir que  $I03 = FALSE$ , la variable de entrada  $IN = TRUE$  y la variable global  $BAN- DERA\_SEG\_PUERTA = TRUE$ . Esto asignará el valor de *TRUE* la variable global  $A\_SG$ ,

```
1  IF IN AND NOT _IO_EM_DI_03 AND BANDERA_SEG_PUERTA THEN
2      A_SG:= TRUE;
3      BANDERA_SEG_PUERTA:= FALSE;
4
5  ELSIF IN AND A_SG AND NOT BANDERA_SEG_PUERTA THEN
6      A_SG:= FALSE;
7
8  END_IF;
9
10 IF IN AND NOT _IO_EM_DI_11 THEN
11     DO_SLOW_STOP_1(SLOW_STOP);
12     SLOW_STOP:= FALSE;
13     //A_SG:= TRUE;
14     _IO_EM_DO_16 := FALSE;
15 ELSE
16     _IO_EM_DO_16 := TRUE;
17 END_IF;
18
19 (*CUANDO LA PUERTA ESTÁ ABIERTA*)
20 IF IN AND _IO_EM_DI_13 THEN
21     INTERLOCK_ERROR := 4;
22
23 (*CUANDO EL SELECTOR NO ESTÁ EN LA POSICIÓN DE CERRAR*)
24 ELSIF IN AND NOT _IO_EM_DI_11 THEN
25     INTERLOCK_ERROR := 5;
26
27 (*CUANDO NO SE HA PRESIONADO EL BOTON PARA REANUDAR LA ACTIVIDAD*)
28 ELSIF IN AND NOT _IO_EM_DI_03 THEN
29     INTERLOCK_ERROR := 6;
30 ELSE
31     INTERLOCK_ERROR := 0;
32     BANDERA_SEG_PUERTA:= TRUE;
33
34 END_IF;
```

Figura 8.19: Código de la función MONITOREO\_PUERTA

la cual es la bandera que pasa el sistema al estado de seguridad y el valor de *FALSE* a la variable global *BANDERA\_SEG\_PUERTA = FALSE*. El segundo condicional, el cual está ubicado en la línea 5 del código, representa que el sistema ya se encuentra en el estado de seguridad, que debe cumplir que *A\_SG = TRUE*, *BANDERA\_SEG\_PUERTA = FALSE* y *IN = TRUE*. Esto asignará el valor de *FALSE* la variable global *A\_SG*.

La segunda sentencia de IF posee un condicional, este monitorea el selector que asegura o desasegura la puerta, conectado a la entrada I11. El condicional, el cual está ubicado en la línea 10 del código, debe cumplir que *IN = TRUE* y *I11 = FALSE*. Esta combinación de variables representa el cambio de asegurado a desasegurado. Esto ejecuta la función *SLOW\_STOP* y asigna el valor de *FALSE* la salida del PLC *O16*, lo cual desasegura la puerta, permitiéndole al usuario abrir sin ningún peligro. En caso de que el condicional no se cumpla, se le asigna el valor de *TRUE* la salida del PLC *O16*, lo cual asegura la puerta, permitiéndole al robot funcionar con normalidad.

La tercera sentencia de IF posee tres condicionales, estos distinguen entre los estados de la puerta y le asigna un identificador de error para notificarle al usuario cuál es el estado de la puerta. El primer condicional, el cual está ubicado en la línea 20 del código, debe cumplir que *IN = TRUE* y *I13 = TRUE*. Esta combinación de variables representa que la puerta se encuentra abierta. Esto asigna el valor de 4 a la variable global *INTERLOCK\_ERROR*. El segundo condicional, el cual está ubicado en la línea 24 del código, debe cumplir que *IN = TRUE* y *I11 = FALSE*. Esta combinación de variables representa que la puerta se encuentra desasegurada. Esto asigna el valor de 5 a la variable global *INTERLOCK\_ERROR*. El tercer condicional, el cual está ubicado en la línea 28 del código, debe cumplir que *IN = TRUE* y *I03 = FALSE*. Esta combinación de variables representa que aún no se ha presionado el botón de confirmación para reanudar el funcionamiento del robot. Esto asigna el valor de 6 a la variable global *INTERLOCK\_ERROR*. En caso de que ninguno de los condicionales se cumpla, se asigna el valor de 0 a la variable global *INTERLOCK\_ERROR*, lo cual representa que el sistema está funcionando normalmente y se asigna el valor de *TRUE* a la variable global *BANDERA\_SEG\_PUERTA*, lo cual indica que el robot está en condiciones de funcionamiento normal.

8. **BALIZA:** Esta función se ejecuta constantemente, no depende de la acción del operario y su tarea es la de controlar la baliza de luz, que indica diferentes estados del sistema. Esta función recibe un parámetro de entrada que define luces se encienden. Se puede ver en la figura [8.20](#) el código que logra llevar a cabo el control de la baliza.

```

1  IF IN = 1 THEN
2      _IO_EM_DO_18:= TRUE; _IO_EM_DO_19:= FALSE; _IO_EM_DO_11:= FALSE;
3      _IO_EM_DO_10:= FALSE; _IO_EM_DO_12:= FALSE; _IO_EM_DO_17:= TRUE;
4
5  ELSIF IN = 2 THEN
6      _IO_EM_DO_18:= FALSE; _IO_EM_DO_19:= TRUE; _IO_EM_DO_11:= FALSE;
7      _IO_EM_DO_10:= FALSE; _IO_EM_DO_12:= FALSE; _IO_EM_DO_17:= FALSE;
8
9  ELSIF IN = 3 THEN
10     _IO_EM_DO_18:= FALSE; _IO_EM_DO_19:= FALSE; _IO_EM_DO_11:= TRUE;
11     _IO_EM_DO_10:= FALSE; _IO_EM_DO_12:= FALSE; _IO_EM_DO_17:= FALSE;
12
13    ELSIF IN = 4 THEN
14        _IO_EM_DO_18:= FALSE; _IO_EM_DO_19:= FALSE; _IO_EM_DO_11:= FALSE;
15        _IO_EM_DO_10:= TRUE; _IO_EM_DO_12:= FALSE; _IO_EM_DO_17:= FALSE;
16
17    ELSIF IN = 5 THEN
18        _IO_EM_DO_18:= FALSE; _IO_EM_DO_19:= FALSE; _IO_EM_DO_11:= FALSE;
19        _IO_EM_DO_10:= FALSE; _IO_EM_DO_12:= TRUE; _IO_EM_DO_17:= FALSE;
20
21    END_IF;

```

Figura 8.20: Código de la función BALIZA

Contiene una sentencias de IF, la cual que inicia en la línea 1 y finaliza en la línea 41. Se puede ver que se utilizan seis salidas del PLC, cada una controla uno de los cinco colores y la restante la bocina de alerta. Se utilizan la variable de entrada *IN* para cada condicional. El funcionamiento se describe a continuación en el cuadro [8.8](#).

Valor de IN	Baliza
1	Enciende el color rojo y activa la bocina de alarma.
2	Enciende el color amarillo.
3	Enciende el color verde.
4	Enciende el color azul.
5	Enciende el color blanco.

Cuadro 8.8: Colores de la baliza dependiendo de la entrada IN

En el código principal donde se usa esta función, la variable global que cambia su valor para definir que color se enciende es la variable *BALIZA\_MODO*.

Finalmente, estas funciones se ejecutan en el programa principal del PLC, el cual se muestra a continuación.

```

1  (* Este programa permite ejecutar el movimiento de los tres ejes,
2  es decir, realiza el control de estos al tiempo *)
3
4  (* FUNCIÓN BALIZA*)
5  BALIZA_1(BALIZA_MODO);
6
7  (* FUNCIÓN DE HOMING *)
8  DO_HOMING_1(HOMING);
9  HOMING:= FALSE;
10
11 (* FUNCIÓN DE ENCENDIDO *)
12 DO_POWER_1(POWER);
13
14 (* FUNCIÓN DE REINICIO *)
15 DO_RESET_1(RESET);
16 RESET:= FALSE;
17
18 (* FUNCIÓN DE MOVIMIENTO *)
19 DO_MOVE_1(RUN);
20 RUN:= FALSE;
21
22 (* FUNCIÓN DE MONITOREO SWITCH PENDANT *)
23 MONITOREO_SWITCH_1(MANUAL_HIGH OR ESTADO_SEGURIDAD);
24 MONITOREO_SWITCH_2(MANUAL_LOW OR ESTADO_SEGURIDAD);
25
26 (* FUNCIÓN DE MONITOREO PUERTA*)
27 DOOR_SATE := NOT _IO_EM_DI_03;
28 MONITOREO_PUERTA_1(DOOR_SATE);
29
30 (* PARADAS*)
31 DO_SLOW_STOP_1(SLOW_STOP);
32 SLOW_STOP:= FALSE;
33
34
35
36
37
38 (* ASIGNACIÓN DE LAS VARIABLES DE ERROR *)
39 ERROR_EJE1:= EjeA1.ErrorID;
40 ERROR_EJE2:= EjeA2.ErrorID;
41 ERROR_EJE3:= EjeA3.ErrorID;
42
43 (* ASIGNACIÓN DE LAS VARIABLES DE LOS ESTADOS *)
44 ESTADO_EJE1:= EjeA1.AxisState;
45 ESTADO_EJE2:= EjeA2.AxisState;
46 ESTADO_EJE3:= EjeA3.AxisState;
47
48 (* LECTURA DE POSICIONES *)
49 MC_ReadParameter_1(EjeA1,TRUE,1);
50 MC_ReadParameter_2(EjeA2,TRUE,1);
51 MC_ReadParameter_3(EjeA3,TRUE,1);
52 D1_ACTUAL:= MC_ReadParameter_1.Value;
53 D2_ACTUAL:= MC_ReadParameter_2.Value;
54 D3_ACTUAL:= MC_ReadParameter_3.Value;
55
56 (* LECTURA DE VALORES MÁXIMOS *)
57 MC_ReadParameter_4(EjeA1,TRUE,8);
58 VEL_MAX:=MC_ReadParameter_4.Value;
59 MC_ReadParameter_5(EjeA1,TRUE,13);
60 ACEL_MAX:=MC_ReadParameter_5.Value;
61 MC_ReadParameter_6(EjeA1,TRUE,16);
62 JERK_MAX:=MC_ReadParameter_6.Value;
63
64 IF AUTOMATICO AND A_RM THEN
65     AUTOMATICO := FALSE;
66     REMOTO := TRUE;

```

Figura 8.21: Código principal del PLC

En la figura [8.21](#) se puede ver como se llaman las diferentes funciones enumeradas anteriormente. Algunas de estas tiene como parámetro de entrada una variable global, la cual es modificada por el código de Python que es ejecutado en la Raspberry Pi, se describe a continuación en el cuadro [8.9](#)

Variable	Función
BALIZA_MODO	Su valor define que colores se encenderán en la baliza. Este color depende del estado del robot.
HOMING	Cuando su valor es <i>TRUE</i> , la rutina de homing se inicia.
POWER	Cuando su valor es <i>TRUE</i> , los servomotres se encienden. Cuando su valor es <i>FALSE</i> , los servomotres se apagan.
RESET	Cuando su valor es <i>TRUE</i> , los ejes se reinician, eliminando errores.
RUN	Cuando su valor es <i>TRUE</i> , se ejecuta el movimiento solicitado por el usuario.
SLOW_STOP	Cuando su valor es <i>TRUE</i> , los ejes ejecutan una parada controlada.

Cuadro 8.9: Descripción de variables principales

Por otro lado, las funciones de monitoreo, como lo son *MONITOREO\_SWITCH\_x* y *MONITOREO\_PUERTA\_x*, no tienen una variable que active su funcionamiento. Estas tienen variables que no se controlan desde el código de Python, las variables *MANUAL\_HIGH*, *MANUAL\_LOW*, *ESTADO\_SEGURIDAD* indican si los modos de Manual baja velocidad, Manual alta velocidad o el Estado de seguridad se encuentran activos o no. La variable *DOOR\_STATE* indica el valor negado de la entrada I03 de PLC, esta entrada recibe una señal de salida del Relé de seguridad.

## 8.2. Implementación de hardware

Como complemento del sistema de control por software, realizados entre la Raspberry PI 4 y el PLC, se diseñaron una serie de sistemas físicos, que de la mano con la aplicación programada, garantiza que se cumplan los requerimientos, entre estos se incluyó el sistema de encierro de la zona de peligro, el diseño del pendant para el manejo del robot por parte del usuario, y el sistema de paradas de emergencia, en distintos puntos específicos del sistema robótico:

### 8.2.1. Guarda de encierro

Para la implementación de la guarda de encierro, se inició con la instalación de las láminas de policarbonato en todas las caras del robot y en la fabricación de los soportes y guías para la puerta corrediza, en la figura [8.22](#) se observa el proceso de instalación de las láminas, antes de fabricar la puerta corrediza, en la que se utilizó soportería y guías en aluminio que permiten su movimiento lateral, permitiendo un espacio para la entrada del operario a la zona de peligro. El resultado final de la misma se muestra en las figuras [C.1](#), [C.2](#) y [C.3](#).



Figura 8.22: Proceso de instalación de guardas de seguridad

Para el funcionamiento de la guarda de encierro, el usuario, por medio del Switch mostrado en la figura 8.23, ubicado en el panel eléctrico del robot conectado al PLC, notifica que se requiere abrir la puerta de acceso a la zona de peligro, por medio del selector, donde se verifica a través de la lógica programada, si es seguro el acceso, de lo contrario detiene el robot, para posteriormente desenergizar el solenoide del interruptor D4NL, que permite al usuario abrir la puerta.

Para la reanudación del movimiento seguro del robot, el usuario, al salir de la zona de peligro, debe cerrar la puerta y devolver el selector a la posición de cerrado, para posteriormente presionar 2 veces el botón de confirmación, ubicado al lado izquierdo del selector, con el que confirma querer reanudar todas las modalidades de funcionamiento del robot.



Figura 8.23: Elementos de interacción en el panel eléctrico

### 8.2.2. Teach Pendant

Para complementar la implementación por software del sistema de control, se almacenó la Raspberry PI 4, dentro del teach pendant, el cual cumple la función de controlador de las funciones del sistema de seguridad, como también el cálculo de las dinámicas del robot. Este dispositivo también es la interfaz de comunicación directa con el usuario y como se muestra en la figura 8.25, este cuenta con la Raspberry PI 4 anteriormente mencionada directamente conectada a energía eléctrica, por medio de su adaptador con Salida de 5.0V / 3.0A DC, una pantalla táctil de 7", con la que el usuario puede manejar la Raspberry, y por consiguiente el robot, un switch OMROM de 3 posiciones de la referencia A4E-C211VA, quien actúa como un botón físico que el operador debe activar para habilitar el sistema de control del robot, asegurando que el robot permanezca en un estado seguro hasta que se habilite intencionalmente. Por último, cuenta también en su carcasa con botón de paro de emergencia por software, conectado a la entrada # 20 del PLC, que al presionarlo, envía el robot ha estado de seguridad.

Para su integración se diseñó una carcasa mostrada en la figura B.6, dando como resultado la estructura mostrada en la figura C.4, compuesta por su respectiva caja y tapa, mostradas respectivamente en C.5 y C.6. Teniendo esta estructura, se procedió con el ensamblaje y conexiones respectivas, entre estas la alimentación de la Raspberry, el botón de emergencia, el switch de tres posiciones y su pantalla, mostrado en la figura C.7 antes de atornillar la tapa a la caja, y mostrando el resultado final en la figura 8.24, en la cual se observa la pantalla del dispositivo dando imagen de la raspberry PI 4 interna.

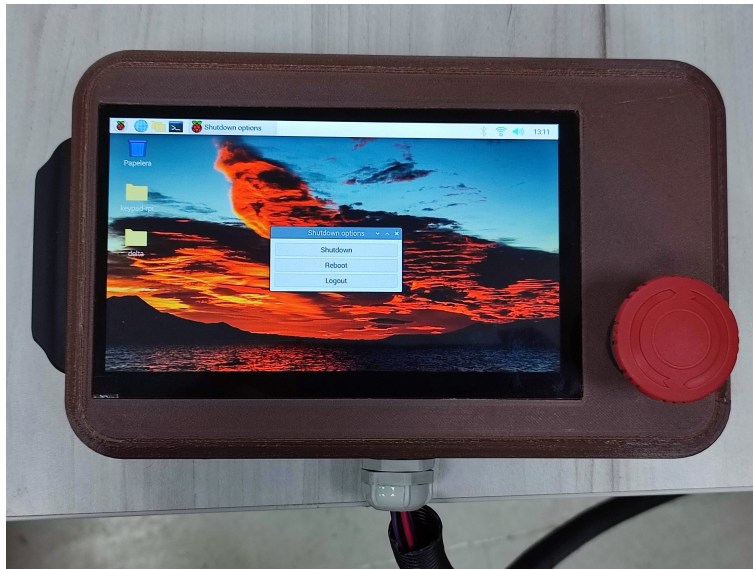


Figura 8.24: Teach pendant completo

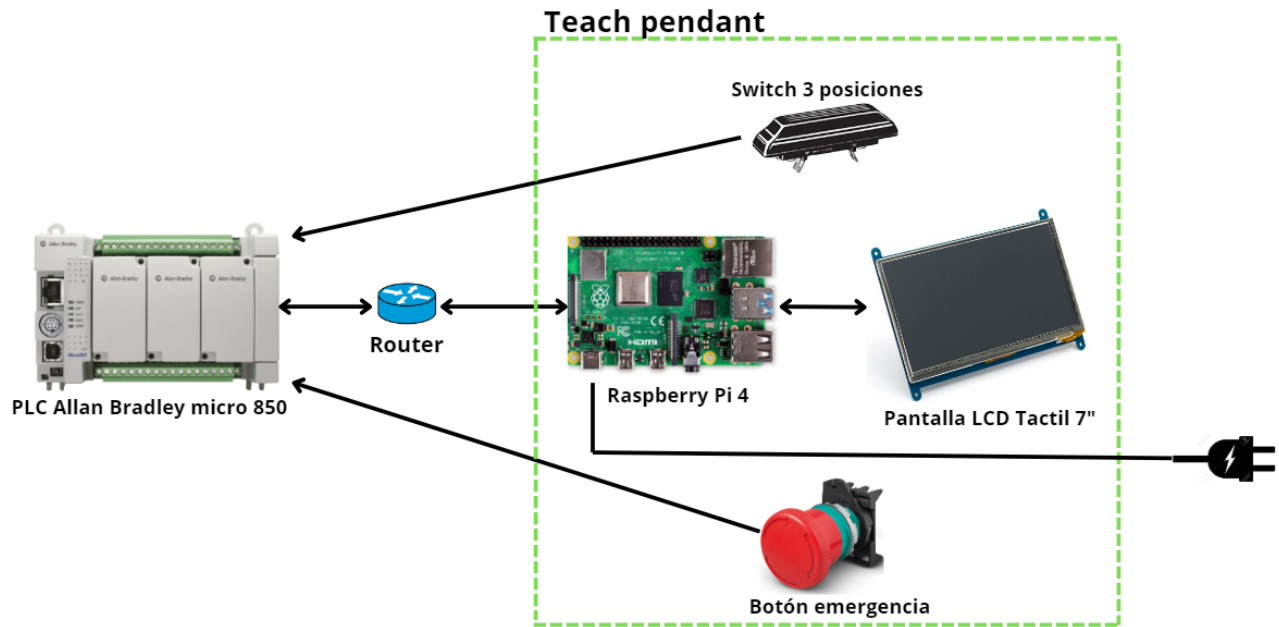


Figura 8.25: Diagrama de conexiones pendant

### 8.2.3. Parada de emergencia

Como se planteó en el diseño, se implementaron 3 botones de emergencia, situados en los puntos más críticos y frecuentes por el usuario, situando el primer botón en la parte exterior del encierro en láminas de policarbonato, en la misma cara por donde tiene la puerta de acceso, el segundo en la zona de peligro, siendo la misma zona interna del encierro, y el tercero en el pendant, diseñado para que este se encuentre todo el tiempo sujetado por el usuario, excepto en modo automático, diseñado para seguir rutinas de movimiento repetitivas, programadas previamente por el usuario. Se visualizan los 3 botones en las imágenes a continuación:



Figura 8.26: Botones de emergencia instalados en parte exterior, teach pendant y dentro de zona de peligro, respectivamente.

### 8.3. Implementación de la arquitectura

La arquitectura diseñada se llevó a cabo integrando el hardware y software desarrollados para que el sistema de seguridad funcione correctamente y cumpla con los requisitos definidos. El establecer una conexión eficiente entre las dos partes, le dará al sistema un mejor desempeño, logrando aprovechar al máximo las herramientas que brinda el hardware y procesando las diferentes señales en el software para que tome las decisiones pertinentes.

En la figura [8.27](#) se muestra el diagrama donde se visualizan los dispositivos de hardware implementados y sus conexiones.

El sistema parte desde el dispositivo que realiza el procesamiento computacional, el cual es la Raspberry Pi 4, esta tiene dos características importantes. Primero, su procesador le da posibilidad de correr con lenguajes de programación de alto nivel, como Python. Segundo, tiene la capacidad de conectarse a una red wifi, lo cual le da la posibilidad de comunicarse con otros dispositivos en una red local. Este dispositivo ejecuta un script de Python donde se encuentra la lógica de control del sistema robótico, en este se realizan los cálculos necesarios para realizar los movimientos del robot, se define la lógica de los modos de funcionamiento, contiene la lógica de la interfaz gráfica utilizada en el pendant; finalmente, contiene la lógica necesaria para que el dispositivo se conecte al PLC y logre leer y escribir sus variables globales. Esta comunicación se logra a través de un router al cual se conecta la Raspberry por medio de wifi y el PLC por medio de un cable Ethernet.

Por otro lado, el PLC Allen-Bradley Micro 850 realiza el control de los diferentes actuadores como lo son el Interlock D4NL-1AFG-B, tres drivers SD 300-30, los cuales a su vez controlan cada uno un Servomotor 80ST-MO243OLB(X), donde gracias a la herramienta de Motion Control, se logra controlar la posición, velocidad y/o par de los motores. A su vez, posee un sistema de detección de fallos, que muestra errores con un número de identificación para que el usuario logre determinar que generó la falla y como solucionarlo, estos errores y su solución se muestran la pantalla del teach pendant.

El PLC Allen-Bradley Micro 850, recibe señales de entrada de dispositivos como el Relé de seguridad G9SE-401, Selectores, Pulsadores, Sensores de proximidad y Switch de tres posiciones.

- **Selector:** La celda robótica utiliza un selector de dos posiciones, con dos contactos Normalmente Cerrados(NC), para abrir y cerrar la puerta de acceso al robot. El robot se mantiene en funcionamiento mientras el selector se encuentre en la posición de *Cerrado*, si el usuario desea acceder al robot, por ejemplo para realizar tareas de mantenimiento, basta con mover el selector a la posición de *Abierto*. Esta acción abre los dos contactos, donde el primero está conectado al PLC y el segundo está conectado al Relé de seguridad G9SE-401. Haciendo uso de esta señal, el PLC detecta que el usuario dese abrir la puerta, por ende ejecuta la rutina de parada suave para detener el robot. Una vez este se detiene, se abre el seguro del Interlock D4NL-1AFG-B lo cual permite abrir la puerta. Por otro lado, el Relé de seguridad G9SE-401, al recibir la señal de Abrir, entra en modo de parada y pone su salida en OFF para indicarle

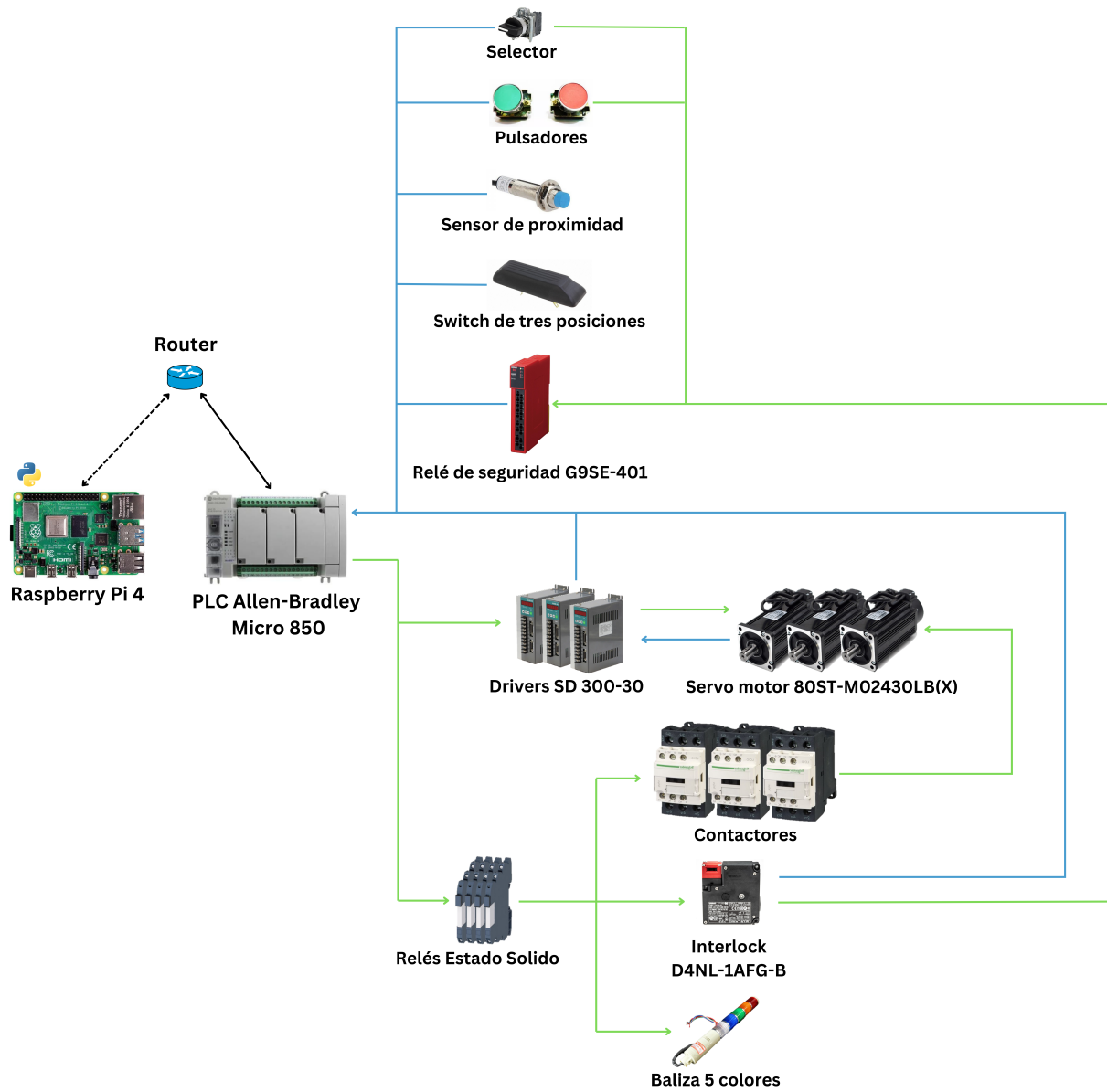


Figura 8.27: Diagrama de integración de la arquitectura

al PLC que el sistema debe mantenerse detenido hasta que las condiciones de seguridad se garanticen para el usuario.

- **Botones:** La celda robótica cuenta con dos tipos de botones. Primero, los pulsadores, los cuales vuelven a su posición inicial después de ser presionados. Segundo, los botones de emergencia, los cuales utilizan un sistema mecánico de enclavamiento para mantenerse presionados hasta que se desbloquean manualmente. Se utilizan en total tres pulsadores y cuatro botones de emergencia.
  1. Pulsador de encendido: Se ubica en el panel eléctrico y se utiliza para encender el robot. Utiliza un contacto Normalmente Abierto(NO) para esta función.
  2. Pulsador de apagado: Se ubica en el panel eléctrico y se utiliza para apagar el robot. Se emplea un contacto Normalmente Cerrado(NC) para esta tarea.
  3. Botones de Emergencia: Están ubicados en el panel eléctrico, lado frontal de la celda robótica, dentro de la celda robótica y en el pendant de control. Se utilizan para detener el funcionamiento del robot en caso de una emergencia, el botón ubicado en el teach pendant está conectado directamente al PLC, donde al ser presionado la señal obliga al PLC a ejecutar la rutina de parada de emergencia, la cual apaga los servomotores. Por otro lado, los tres botones en la estructura apagan todo el robot, debido a que están en serie con la línea de potencia.
  4. Pulsador para confirmación de cierre: Se ubica en el panel eléctrico, al lado del selector, para abrir y cerrar la puerta. El botón contiene un contacto normalmente abierto(NO), para enviar una señal al Relé de seguridad G9SE-401. Para que el Relé de Seguridad salga del modo de parada debe recibir una señal que pase de OFF a ON y de nuevo a OFF, de esta forma restablecen sus funciones y su salida le dice al PLC que el sistema se encuentra con las condiciones de seguridad necesarias para que el operario pueda utilizar el robot.
- **Sensor de proximidad:** Se emplean un total de nueve sensores de proximidad inductivos, dos ubicados en el límite superior de la guía del deslizador y uno ubicado en el límite inferior; los tres ejes tienen este arreglo de sensores. Se utiliza uno en el inicio y otro en el fin, como finales de carrera, para determinar cuando el deslizador llega a su límite superior e inferior y así evitar daños en la estructura. Además, se utiliza un segundo sensor en la parte superior para la rutina de homing. Estos sensores están conectados al PLC como entradas.
- **Switch de tres posiciones:** Este dispositivo está ubicado en el teach pendant, donde el usuario debe mantenerlo en la posición media para que el robot ejecute sus funciones. Los cuatro contactos de este elemento están conectados directamente al PLC como entradas y se utilizan para monitorear en qué posición se encuentra el switch.
- **Interlock D4NL-1AFG-B:** Este dispositivo consta de dos partes. La primera parte es el seguro, donde se encuentran los diferentes contactos de monitoreo y también el mecanismo de

aseguramiento. La segunda parte es la llave de operación, la cual se incierta en al seguro y puede ser retenida o liberada dependiendo de si el seguro está activo o no. El seguro se ubica en la estructura de la celda robótica, al lado de la puerta de acceso, mientras la llave de operación se ubica en la puerta de la celda robótica. El PLC tiene control sobre el mecanismo de aseguramiento de este dispositivo por medio de un Relé de estado sólido, cuando este es encendido cierra el seguro, reteniendo la llave de operación, lo cual impide que la puerta pueda abrirse. Por otro lado, cuando el Relé es apagado, se abre el seguro, liberando la llave de operación y permitiendo que la puerta pueda ser abierta.

El Interlock posee tres contactos para el monitoreo de su estado, los contactos que muestran si la llave de operación está insertada y si el seguro está activo están conectadas como entradas al PLC, para que esta información pueda ser enviada a la Raspberry y mostrada en pantalla al usuario por medio del teach pendant. Por otro lado, el contacto restante, también monitorea estas dos variables, pero de forma simultánea. Este contacto está conectado a una de las entradas del Relé de seguridad para garantizar que la puerta se cerró y aseguró.

- **Relé de seguridad G9SE-401:** Este dispositivo se encuentra ubicado en el panel eléctrico y su función es la de monitorear que la puerta se mantenga cerrada mientras que el usuario no le indique al sistema que desea acceder a la celda robótica.

Cumple con esta tarea, con dos contactos que deben estar siempre cerrados, indicando que el sistema está bien. Cuando uno de los dos se abre, automáticamente la salida de seguridad del Relé se apaga, indicándole al PLC que debe detenerse porque el sistema no es seguro. El primer contacto es del selector, donde su contacto está cerrado cuando se encuentra en la posición de puerta cerrada y se abre cuando se encuentra en la posición de puerta abierta. El segundo contacto es del Interlock D4NL-1AFG-B, el cual está cerrado cuando la puerta está cerrada y asegurada, pero se abre cuando no se cumple alguna de estas dos condiciones.

Si alguno de los dos contactos se abre y el Relé entra en modo parada, para restablecer sus funciones normales se debe asegurar que los dos contactos se encuentren cerrados, lo cual quiere decir que la puerta debe estar cerrada, asegurada y el selector debe estar en la posición de cerrado. Luego, se debe presionar el pulsador para confirmación de cierre, lo cual restablece las funciones del Relé de seguridad.

En la pantalla del tech pendant se le notificará al usuario cuando alguna de las condiciones no se cumpla y qué acción debe realizar para llevar el sistema a su funcionamiento normal.

- **Drivers SD 300-30:** Estos dispositivos son los encargados de alimentar, controlar y regular los movimientos de los Servomotores 80ST-M02430LB(X). Gracias a los drivers, los motores rotan una distancia determinada a una velocidad determinada para hacer posible el desplazamiento de la brida del robot a donde el usuario lo desea.

Adicionalmente, estos drivers están equipados con características de protección y diagnóstico, lo cual les permite detectar problemas en los motores como sobrecorriente, sobretension

o sobrecarga en el motor y detener su funcionamiento preventivamente. Estas alarmas son enviadas al PLC para que este ejecute las acciones pertinentes para garantizar que el usuario se mantenga seguro y que la celda robótica no resulte con daños.

- **Contactores:** Su función es la de entregar o cortar el suministro de potencia a los Servomotres, cada motor está conectado a un contactor individualmente, esto permite prender y apagar los motores sin apagar el sistema completo. Su accionamiento se realiza por medio de Relés de estado sólido, lo cuales son controlados por el PLC. El usuario, desde el teach pendant, tiene la capacidad de poner en marcha y apagar los motores gracias a esto.
- **Baliza 5 colores:** Es un dispositivo utilizado para proporcionar información visual de manera rápida, de esta forma los operarios en el área pueden conocer la condición del robot y tomar acción de ser necesario. Dentro de la información que puede proveer este dispositivo está la indicación del estado de la máquina o proceso, advertencia de peligro utilizando colores brillantes, señalización de emergencias para indicar la necesidad de revisión, control de acceso a zonas de peligro para el usuario, confirmación de operaciones indicando que una rutina se completó exitosamente o se presentó alguna falla, entre otras.

Los colores de la baliza o indicadores lumínicos deben ser definidos y utilizados siguiendo su respectivo significado para evitar un reconocimiento erróneo por parte de los operarios. Por esta razón, un código de colores que puede ser aplicado a cualquier tipo de maquinaria se especifica en el estándar ICE 60204-1 [27].

Siguiendo lo establecido en este estándar, se deben utilizar cuatro colores, lo cuales son Rojo, Amarillo, Azul, Verde y Blanco. El color Rojo, está reservado para errores críticos, lo cual no hace parte del funcionamiento normal del mismo [28]. El color Amarillo, es utilizado para advertir al usuario u operarios de un estado anormal o inminente error crítico [28]. El color Azul, se enciende cuando el robot requiere que el usuario realice una acción obligatoria para realizar una tarea [28]. El color Verde, representa que la máquina se encuentra en su estado normal y es capaz de realizar sus tareas de forma autónoma [28]. Finalmente, la baliza posee el color blanco, que no tiene un significado específico según la norma, por lo que es utilizada de forma libre. Siguiendo la descripción de cada color, según el estándar, se asignaron a los siguientes eventos:

1. Rojo:

- Errores previos o durante la realización de un movimiento o errores los drivers.
- Botón de emergencia en el teach pendant presionado.
- Switch de seguridad se soltó.
- Switch de seguridad no se ha presionado lo suficiente.
- Switch de seguridad se presionó más de lo debido.

2. Amarillo:

- Rutina de homing en ejecución.
- Robot realizando un movimiento.

3. Azul:

- Reinicio de Motion Control utilizando el botón *Reset* en el teach pendant.
- Rutina de homing finalizada.
- Cambio de modo.

4. Verde:

- Puerta de la celda robótica abierta.
- Selector de la puerta no está en posición de cerrado.
- No se ha confirmado la reanudación de actividades.

5. Blanco:

- Motores apagados.

# Resultados y Discusión

---

Con el propósito de efectuar un análisis exhaustivo de los resultados y asegurar el cumplimiento de la normativa de seguridad ANSI/RIA R15.06 y los estándares a los cuales hace referencia, se llevaron a cabo dos tipos de validaciones iniciales. Estas se realizaron con el objetivo de validar el cumplimiento de funcionamiento, secuencia lógica y requisitos contemplados en la norma. Cada aspecto fue sometido a una validación individual para asegurar su correcta aplicación y cumplimiento en el robot. Este proceso se llevó a cabo mediante la implementación tanto a nivel de software como de hardware, respaldando así un manejo seguro por parte de los usuarios y asegurando el logro del objetivo general de este trabajo.

## 9.1. Evaluación del Cumplimiento de Requisitos y Medidas de Protección

La norma de seguridad ANSI/RIA R15.06 abarca conceptos, alcances, definiciones y la evaluación de riesgos en la implementación de robots y sistemas robóticos, como se detalló en los capítulos anteriores. Además, el capítulo 5 de la norma incluye un listado detallado de requisitos y medidas de protección aplicables a robots e integración de sistemas robóticos. Este capítulo se desglosa en varias secciones, cada una de las cuales aborda aspectos específicos de seguridad para robots industriales en general. Sin embargo, no todos estos requisitos son aplicables a un sistema en particular, como se evidencia en las tablas [6.2](#), [6.3](#), [6.4](#), [6.5](#), y [6.6](#) presentadas en el capítulo de especificaciones y requisitos del sistema. En consecuencia, se realizó una verificación detallada del cumplimiento de dichos requisitos,

## 9.2. Pruebas


En esta etapa del trabajo, se llevaron a cabo pruebas exhaustivas para evaluar el desempeño y cumplimiento de las funcionalidades del robot, las cuales fueron derivadas de las tablas de requerimientos previamente establecidas. Dichas funcionalidades se organizaron en una lista dividida en dos categorías principales: requerimientos universales, es decir, aquellos de carácter estricto en cualquier modo de operación del sistema; y funciones específicas relacionadas con los modos manuales, automático, remoto y estado de seguridad.

Para la validación de estas pruebas, se optó por la participación de un evaluador externo con experiencia en robótica, pero sin vinculación directa con este proyecto de grado. El ingeniero elec-

trónico Simón Plata, docente en la Universidad Javeriana de Cali con amplios conocimientos en sistemas embebidos, robótica y automatización de procesos, fue seleccionado como evaluador.

Previamente a la ejecución de las pruebas, el Ingeniero Simón se familiarizó con el robot, recibiendo instrucciones detalladas sobre su funcionamiento básico, los distintos modos de operación y los requisitos normativos que debían cumplirse. Las pruebas, realizadas en diciembre de 2023, quedan documentadas en la tabla 9.10 adjunta, la cual refleja el cumplimiento satisfactorio de todas las funcionalidades evaluadas. Además, se incluye la firma del Ingeniero Simón Plata, validando de manera positiva el funcionamiento integral del sistema. Estos resultados respaldan la efectividad de la implementación y confirman la conformidad del robot con los estándares establecidos.

CRITERIO	DESCRIPCIÓN	VALORACION	
		SI	NO
UNIVERSALES	1.1. Al encender el robot, no debe realizar ningún movimiento sin antes haber hecho el homing y ser solicitado dicho movimiento por el usuario.	x	
	1.2. El robot no deberá recibir órdenes de movimiento si no se ha realizado el homing.	x	
	1.3. Parada de emergencia desde el botón situado en la zona fuera de peligro, apaga todo el sistema.	x	
	1.4. Parada de emergencia desde el botón situado dentro de la zona de peligro, apaga todo el sistema.	x	
	1.5. Parada de emergencia desde el botón situado en el teach Pendant detiene cualquier movimiento y lleva el sistema a estado de seguridad.	x	
	1.6. Se corrige el error en los drivers, reiniciando el sistema desde el botón de "reset".	x	
	1.7. Para cambiar de un modo a otro, el usuario debe realizar una acción de confirmación, por ejemplo oprimir un botón.	x	
	1.8. Pendant muestra datos en tiempo real de posición del robot.	x	
	1.9. La baliza representa el estado actual del robot con los colores tal como describe la norma IEC 60204-1.	x	
	1.10. El interlock mantiene la puerta bloqueada hasta que se solicita su apertura por medio del selector.	x	
	1.11. Al solicitar la apertura de la puerta, se detiene el movimiento del robot.	x	
	1.12. Los controles de actuación se encuentran debidamente indicados, describiendo su función.	x	
	1.13. El robot está en la capacidad de ser manejado por una sola persona.	x	
	1.14. Si el robot se está manejando desde el Pendant, no debe recibir órdenes de movimiento desde otras fuentes.	x	
	1.15. Los dispositivos de visualización (Pendant y HMI), muestran claramente el modo de operación en el que se encuentra el robot.	x	
	1.16. Si se viola un límite de software o hardware, se detiene inmediatamente el movimiento que esté ejecutando el robot.	x	
	1.17. La pérdida de comunicación con el pendant detiene los movimientos del robot, excepto cuando está en modo remoto.	x	
	1.18. El operario debe poder mover manualmente el brazo robótico cuando este no se encuentre energizado.	x	
MODOS MANUALES	2.1. Permitir el recibimiento de órdenes por posición y por botones de ejes individuales.	x	
	2.2. Notificar error y negar ejecución de movimiento si queda por fuera de los límites de cualquiera de los 3 ejes.	x	
	2.3. Negar ejecución de movimiento si el switch de 3 posiciones(hold to run o switch deseguridad) no se encuentra en la posición correcta.	x	
	2.4. Negar ejecución de movimiento si el switch de 3 posiciones(hold to run o switch deseguridad) se presionó a su última posición y no se soltó completamente antes de volver a llevarlo a su posición central.	x	
	2.5. Realizar parada suave(Slow Stop), cuando el robot esté ejecutando una orden, apenas sea solicitada por el operario.	x	
	2.6. Al ordenar un movimiento a una velocidad mayor a 250 mm/s en modo manual de velocidad reducida (Modo Manual Low), el robot no la realiza y notifica al usuario que la velocidad ingresada supera el límite.	x	
MODO AUTOMÁTICO	3.1. Denegar la ejecución de cualquier rutina automática si la puerta de acceso a la zona de peligro se encuentra abierta y activar estado de seguridad.	x	
	3.2. Denegar rutina si esta incumple los límites de posición de al menos uno de los 3 ejes.	x	
	3.3. Realizar parada suave, cuando el robot esté ejecutando una rutina, apenas sea solicitada por el operario.	x	
	3.4. Si se está ejecutando una rutina, y se solicita cambio de modo, se debe detener la ejecución de la rutina y cambiar a modo solicitado.	x	
ESTADO DE SEGURIDAD	4.1. Si el sistema se encuentra en estado de seguridad, y se está notificando un error o un fallo de algún eje, prohíbe cambiar a otro modo hasta que se corrija el error.	x	
	4.2. Al entrar en estado de seguridad, requiere doble confirmación del usuario, para llevar el robot de nuevo a otro modo (confirmación en pantalla y hold-to-run presionado)	x	
MODO REMOTO	5.1. Se permite el manejo del robot de forma remota por medio de la comunicación inalámbrica, desde un dispositivo que esté conectado a la red WI-FI y ejecute el archivo de Python.	x	

  
 Ing. Simón Plata

Cuadro 9.10: Resultado de la evaluación realizada al sistema terminado

# Conclusiones

---

En la culminación de este trabajo de grado, se cumplió satisfactoriamente con el objetivo general de desarrollar un sistema de control para un robot delta lineal, alineado rigurosamente con la normativa ANSI/RIA R15.06-2012. Este logro no solo refleja el compromiso con el cumplimiento normativo, sino que también demuestra la eficacia de los métodos de diseño y desarrollo, meticulosamente adaptados a las necesidades específicas del proyecto.

A través de un proceso detallado y estructurado, se especificaron y aplicaron los requisitos de hardware y software pertinentes, asegurando que cada componente del robot no solo cumpliera con los estándares de la industria, sino que también estuviera optimizado para su rendimiento y seguridad. La integridad del sistema se fortaleció mediante la incorporación de medidas de seguridad críticas, tales como la fabricación y programación del pendant, la instalación de barreras físicas de seguridad, el desarrollo de una puerta con su respectivo interlock, y la implementación de paradas de emergencia accesibles desde múltiples puntos.

La supervisión detallada de las pruebas, dirigidas por el experimentado ingeniero electrónico Simón Plata, fue fundamental para confirmar la efectividad y robustez del sistema. Este enfoque riguroso en la seguridad no solo resultó en un sistema confiable y eficiente, sino que también estableció un precedente para el diseño y desarrollo futuro de sistemas robóticos.

Es importante subrayar la adaptabilidad intrínseca del sistema. Aunque actualmente no se ha asignado un uso específico al robot, su conformidad con la normativa y su diseño adaptable abren un abanico de posibilidades para futuras aplicaciones. Se destaca la importancia de realizar un análisis de riesgo adicional en caso de que se asigne un uso específico en el futuro, reafirmando así el compromiso continuo con la seguridad y la excelencia en cada fase del desarrollo.

Este proyecto no solo ha representado un logro en términos de ingeniería y diseño técnico, sino también un viaje significativo de aprendizaje y adaptación. La experiencia acumulada y las habilidades desarrolladas durante este proceso serán invaluable para futuros proyectos y desarrollos en el campo de la robótica. Se destaca la importancia de la implementación de normas y estándares, algo menos común en el entorno académico pero que ha resultado ser una gran experiencia y aprendizaje para los integrantes de este trabajo. Esta rigurosa adhesión a los estándares ha permitido un desarrollo más organizado y óptimo del sistema. En particular, la incorporación de etapas clave como la evaluación de riesgos y la evaluación del cumplimiento de requisitos ha sido fundamental

para garantizar la calidad del trabajo realizado. Este enfoque estructurado ha llevado a una mayor eficiencia y confiabilidad en el desarrollo del proyecto, estableciendo una base sólida para futuras investigaciones y aplicaciones en el ámbito de la robótica.

# Recomendaciones

---

El diseño e implementación del sistema de seguridad para el robot delta lineal se ha realizado utilizando como base la normativa ANSI/RIA R15.06 este podría mejorarse y así garantizar un entorno de trabajo seguro y eficiente con las siguientes recomendaciones.

1. **Actualización Periódica de Normativas:** Se recomienda informarse regularmente sobre las actualizaciones de las normativas y estándares de seguridad, debido a que estas pueden tener cambios con el tiempo. Estas modificaciones podrían traer cambios pertinentes que implementar en el robot y que aseguren que el sistema sea más seguro.
2. **Implementación de Tecnologías:** La implementación de dispositivos más avanzados para reemplazar elementos que actualmente se encuentran en el robot mejoraría la respuesta del sistema, por ejemplo la implementación de guardas láser o un sensor de presencia en la zona de peligro. Esto no se realizó en el proyecto debido a que el costo de estos dispositivos es muy elevado y su tiempo de entrega es de varios meses.
3. **Evaluación de Riesgos Adicionales:** Realizar evaluaciones de riesgo dependiendo de las tareas que ejecutara el robot, los peligros al realizar labores de *pick and place* o labores similares a la impresión 3D son distintos. Estos pueden variar dependiendo del actuador instalado en la brida, tipo de objetos que manipulará o tareas que realizará el robot. Una evaluación detallada facilitaría el análisis de estos y de que forma mitigarlos.
4. **Almacenamiento de datos:** Integrar un sistema de almacenamiento de datos beneficia al sistema debido a que se podría guardar el estado o valores de diferentes variables cruciales para el funcionamiento del robot. De esta forma, en caso de un reinicio eléctrico, el sistema podría retomar su tarea desde el momento en que se detuvo su operación y no desde el inicio. Además, esto podría darle al usuario la posibilidad de guardar rutinas para ejecutar en el modo automático. Esto no se realizó debido a que no hace parte del objetivo del proyecto el cual es entorno a la seguridad, esta recomendación está relacionada al funcionamiento de el modo automático y como podría mejorarse.
5. **Implementación de Sistema de Registro y Reporte:** La integración de un sistema de registro y reporte que almacene los eventos relacionados con la seguridad e informe a los encargados de esta área el estado del robot. Dentro de este registro se podrían almacenar eventos de seguridad y que acciones se tomaron para solucionar el problema, esto facilitaría el análisis de estos incidentes.

Estas recomendaciones podrían mejorar la eficacia y robustez del sistema de seguridad del robot, garantizando la seguridad del usuario en el entorno de trabajo cumpliendo con la normativa necesaria.

# Bibliografía

- [1] D. Marshall and C. Bredin, “Historia de un éxito: una mirada retrospectiva a la aportación de abb a la robótica industrial,” *Revista ABB*, no. 2, pp. 56–62, 2008.
- [2] O. Görnemann, “Seguridad de los robots colaborativos - mitos – realidades – futuro,” Available at [https://www.osalan.euskadi.eus/contenidos/informacion/ponencias\\_jt170720\\_cursoverano/es\\_def/adjuntos/ponencia\\_jt170720\\_otto\\_goernemann.pdf](https://www.osalan.euskadi.eus/contenidos/informacion/ponencias_jt170720_cursoverano/es_def/adjuntos/ponencia_jt170720_otto_goernemann.pdf), 2017.
- [3] S. B. G. P. K. S. A. Z. A. J. M. M. K. Anand Singh Rajawat, Pradeep Bedi, “Reformist framework for improving human security for mobile robots in industry 4.0,” *Hindawi*, vol. 2021, no. 4744220, p. 10, 2021. [Online]. Available: <https://doi.org/10.1155/2021/4744220>
- [4] A. W. Colombo, S. Karnouskos, X. Yu, O. Kaynak, R. C. Luo, Y. Shi, P. Leitao, L. Ribeiro, and J. Haase, “A 70-year industrial electronics society evolution through industrial revolutions: The rise and flourishing of information and communication technologies,” *IEEE Industrial Electronics Magazine*, vol. 15, no. 1, pp. 115–126, 2021.
- [5] S. Robla-Gómez, V. M. Becerra, J. R. Llata, E. González-Sarabia, C. Torre-Ferrero, and J. Pérez-Oria, “Working together: A review on safe human-robot collaboration in industrial environments,” *IEEE Access*, vol. 5, pp. 26 754–26 773, 2017.
- [6] M. D. Correa Castro, “Análisis de estrategias de financiamiento de las pymes del sector comercio en el municipio de duitama boyacá,” 2021.
- [7] J. E. Otero<sup>o</sup>, G. Carbone, M. Ceccarelli, and J. L. M. Sanz<sup>o</sup>, “Criterios para la seguridad en el uso de robots.”
- [8] C. P. de Competitividad, “Informe nacional de competitividad 2020-2021,” Bogotá, Available at [https://compite.com.co/wp-content/uploads/2021/12/CPC\\_INC\\_2021-2022-COMPLETO.pdf](https://compite.com.co/wp-content/uploads/2021/12/CPC_INC_2021-2022-COMPLETO.pdf), 2021.
- [9] A. J. Alvares, E. A. R. Gasca, and C. I. R. Jaimes, “Development of the linear delta robot for additive manufacturing,” in *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2018, pp. 187–192.
- [10] F. A. F. D. A. Santos, “Diseño conceptual y estudio cinemático de un robot delta de tres grados de libertad,” *Universidad Autónoma de Bucaramanga*, Available at [https://repository.unab.edu.co/bitstream/handle/20.500.12749/12203/2011\\_Articulo\\_Daniel\\_Santos\\_Marin.pdf?sequence=2&isAllowed=y](https://repository.unab.edu.co/bitstream/handle/20.500.12749/12203/2011_Articulo_Daniel_Santos_Marin.pdf?sequence=2&isAllowed=y), 2012.
- [11] M. B. L. Kaiser, A. Schlotzhauer, “Safety-related risks and opportunities of key design-aspects for industrial human-robot collaboration: Third international conference,” in *Interactive Collaborative Robotics*, vol. 1, 2018, pp. 95–104.

- [12] P. Papcun, E. Kajáti, and J. Koziorek, “Human machine interface in concept of industry 4.0,” in *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*, 2018, pp. 289–296.
- [13] R. L. Puente, R. J. P. Chimal, C. H. –Mosqueda, and J. U. M. Minjarez, “Control de brazo robótico clasificador mediante hmi y servidor web,” *Revista de Cómputo Aplicado*, vol. 3, no. 11, pp. 9–15, 2019.
- [14] *Seguridad de las máquinas. Principios generales para el diseño. Evaluación del riesgo y reducción del riesgo - ISO 12100*, ISO Std.
- [15] O. Görnemann, “Seguridad de maquinaria con normas internacionales.”
- [16] U. Barbolla Zautua, “Sistemas de seguridad para maquinaria (safety): estudio teórico y aplicación práctica,” 2019.
- [17] *Protecciones perimetrales de máquinas Guía actualizada*, Troax Systems, DIRECTIVA PARA MÁQUINAS 2006/42/CE NORMAS ISO Y ACTUALIZACIONES DE LA NORMA 14120:2015.
- [18] I. O. for Standardization (ISO), “Iso 10218-1:2011(en) robots and robotic devices — safety requirements for industrial robots — part 1: Robots,” Available at <https://www.iso.org/obp/ui/#iso:std:iso:10218:-1:ed-2:v1:en>, 2011.
- [19] R. I. ASSOCIATION, “ANSI/RIA R15.06-1999, for industrial robots and robot systems — safety requirements,” Available at <https://webstore.ansi.org/Standards/RIA/ansiriar15062012>, 1999.
- [20] V. Murashov, F. Hearl, and J. Howard, “Working safely with robot workers: Recommendations for the new workplace,” *Journal of Occupational and Environmental Hygiene*, vol. 13, no. 3, pp. D61–D71, 2016, pMID: 26554511. [Online]. Available: <https://doi.org/10.1080/15459624.2015.1116700>
- [21] J. Contreras, J. Rubio, and A. Martínez, “Plc based control of robots using plcopen motion control specifications,” in *Advances in Automation and Robotics Research*, H. A. Moreno, I. G. Carrera, R. A. Ramírez-Mendoza, J. Baca, and I. A. Banfield, Eds. Cham: Springer International Publishing, 2022, pp. 109–120.
- [22] B. Belzile and D. St-Onge, *Safety First: On the Safe Deployment of Robotic Systems*. Singapore: Springer Nature Singapore, 2022, pp. 415–439. [Online]. Available: [https://doi.org/10.1007/978-981-19-1983-1\\_14](https://doi.org/10.1007/978-981-19-1983-1_14)
- [23] L. Gualtieri, E. Rauch, and R. Vidoni, “Emerging research fields in safety and ergonomics in industrial collaborative robotics: A systematic literature review,” *Robotics and Computer-Integrated Manufacturing*, vol. 67, p. 101998, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S073658452030209X>

- 
- [24] *Instructional Manual For SD300*, 1st ed., TOPCNC Workshop Automation Technology Co.
- [25] I. O. for Standardization (ISO), “Iso 13854:2017 safety of machinery — minimum gaps to avoid crushing of parts of the human body,” 2017.
- [26] —, “Iso 13855:2010 safety of machinery — safety of machinery — positioning of safeguards with respect to the approach speeds of parts of the human body,” 2010.
- [27] T. Saito, T. Hoshi, H. Ikeda, and K. Okabe, “Global harmonization of safety regulations for the use of industrial robots-permission of collaborative operation and a related study by jniosh,” *Industrial health*, vol. 53, no. 6, pp. 498–504, 2015.
- [28] S. Wischniewski, A. Scheidig, and H.-M. Gross, “Designing an expressive head for a help requesting socially assistive robot,” in *Human-Friendly Robotics 2019: 12th International Workshop*, vol. 12. Springer Nature, 2020, p. 96.



CAPÍTULO 12

# Anexos

---



# Diseño del software de la arquitectura de control

## A.1. Modo Manual Baja Velocidad

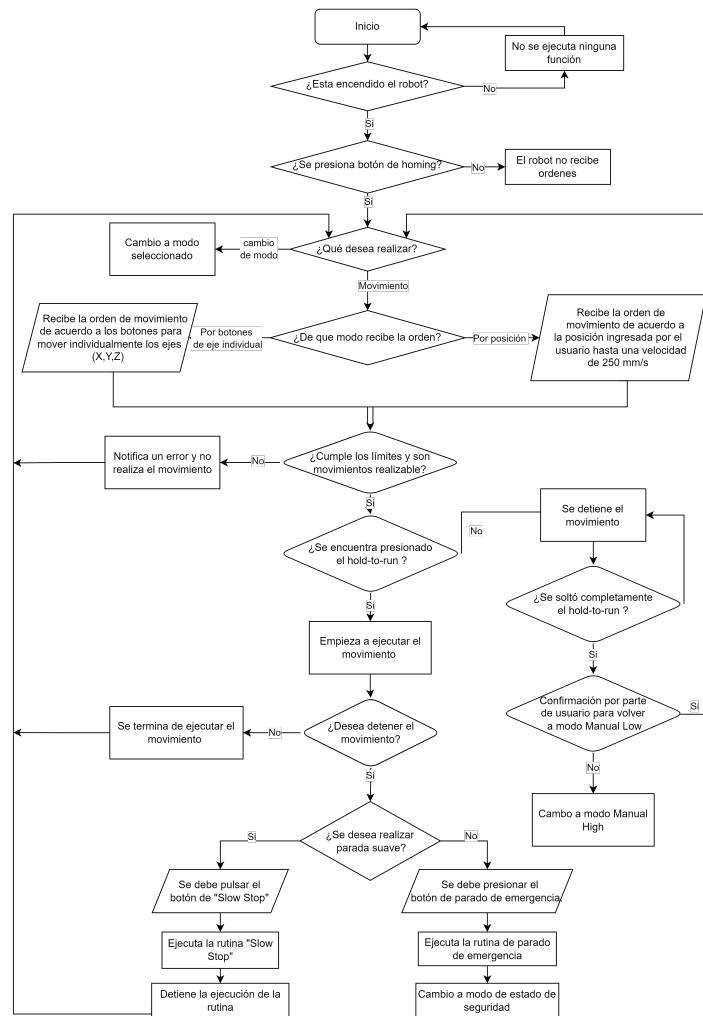
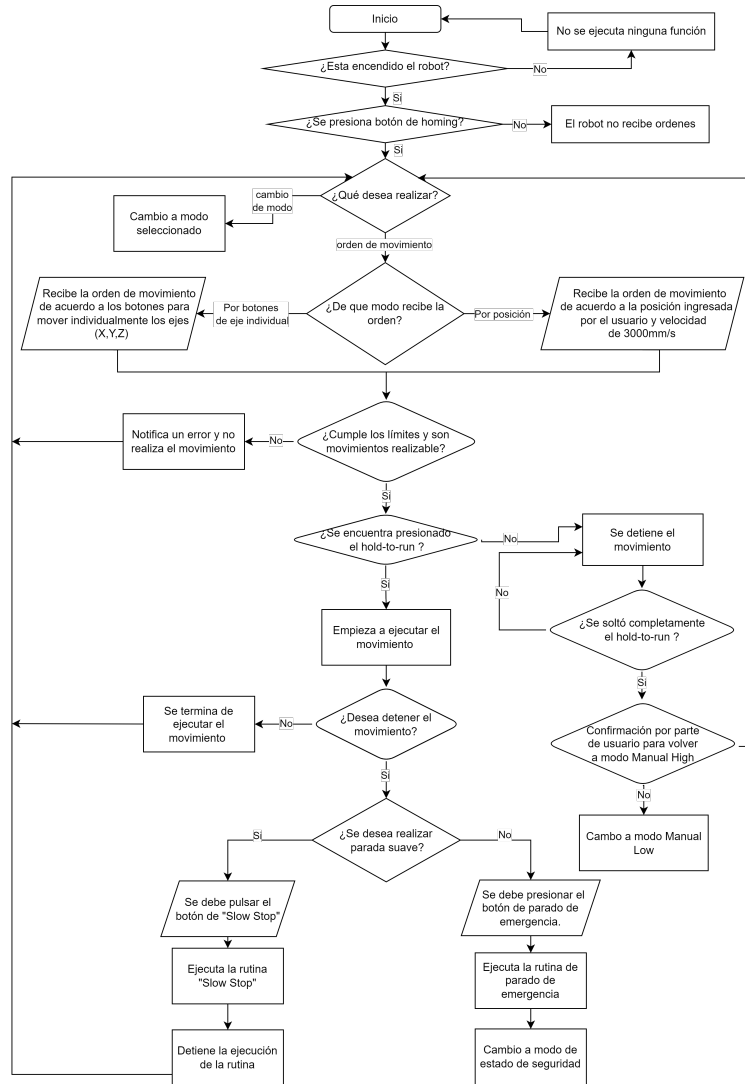


Figura A.1: Diagrama de flujo de modo *Manual Low*

## A.2. Modo Manual Alta Velocidad

Figura A.2: Diagrama de flujo de modo *Manual High*

A.3. Modo Automático

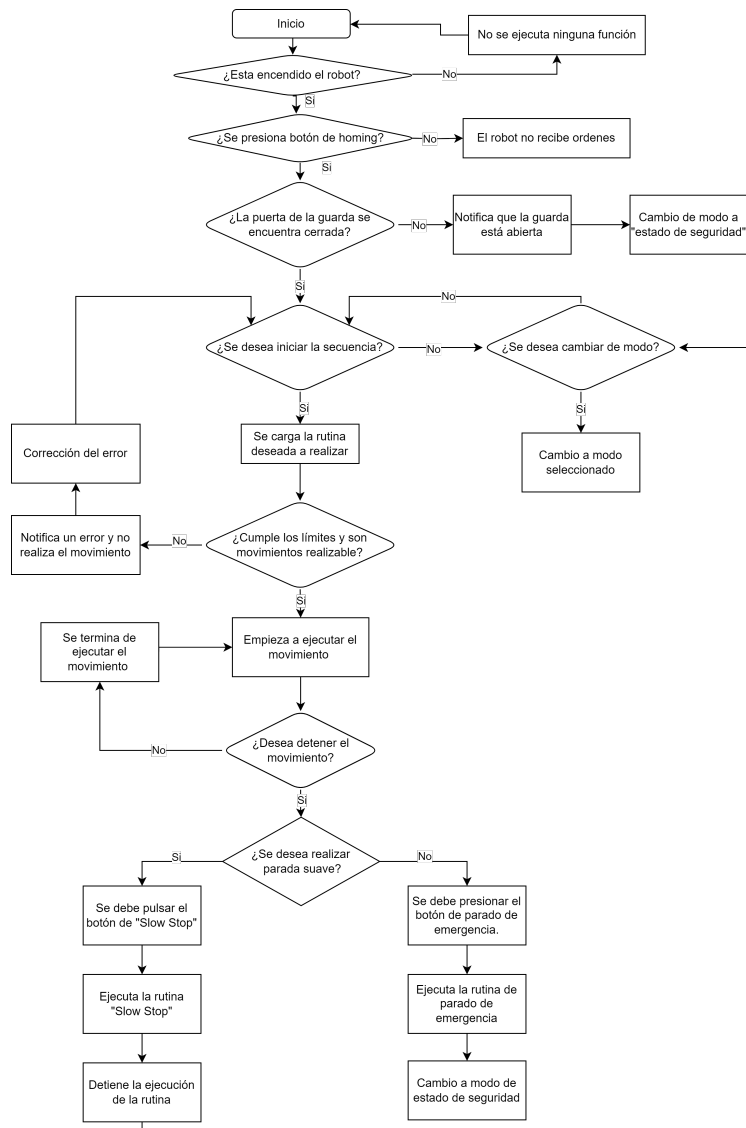
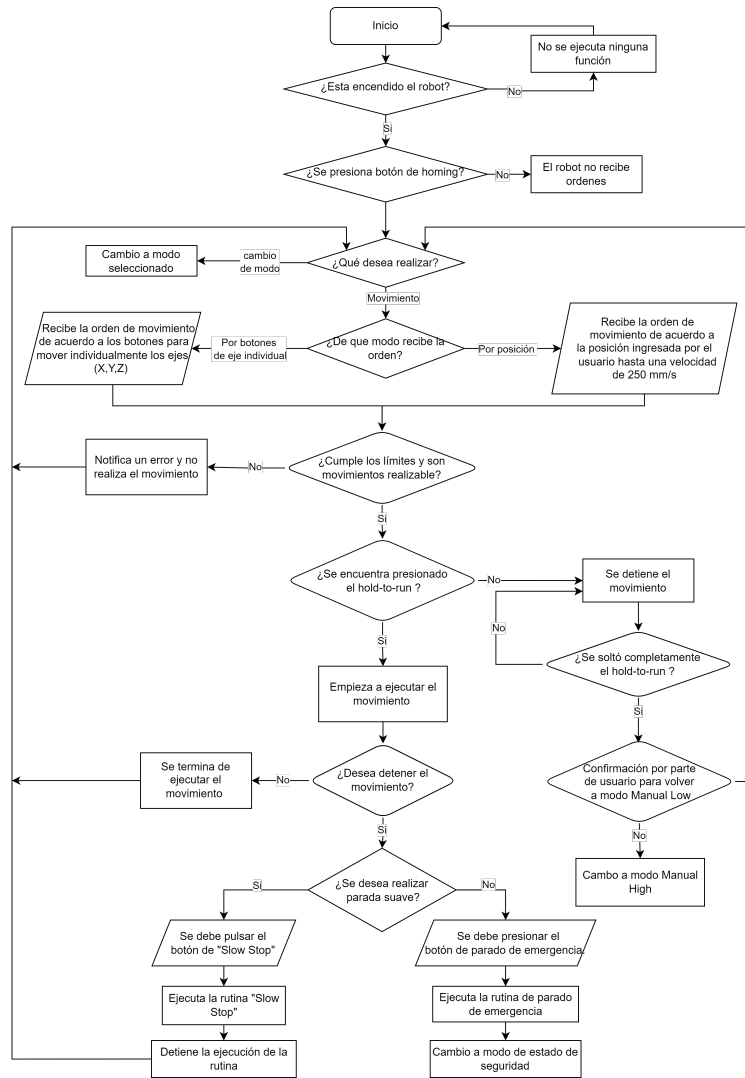


Figura A.3: Diagrama de flujo de modo Automático

## A.4. Modo Remoto

Figura A.4: Diagrama de flujo de modo *Remoto*

## A.5. Estado de Seguridad

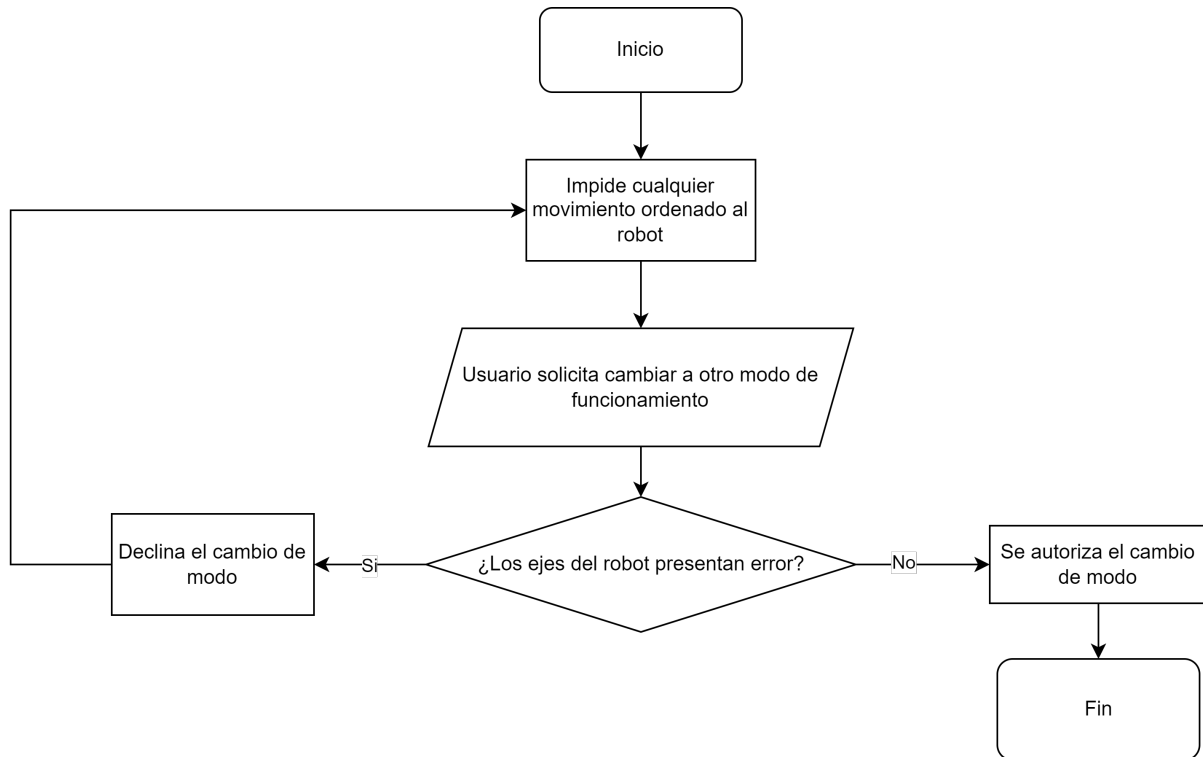


Figura A.5: Diagrama de flujo de estado de seguridad

## A.6. Programación del PLC

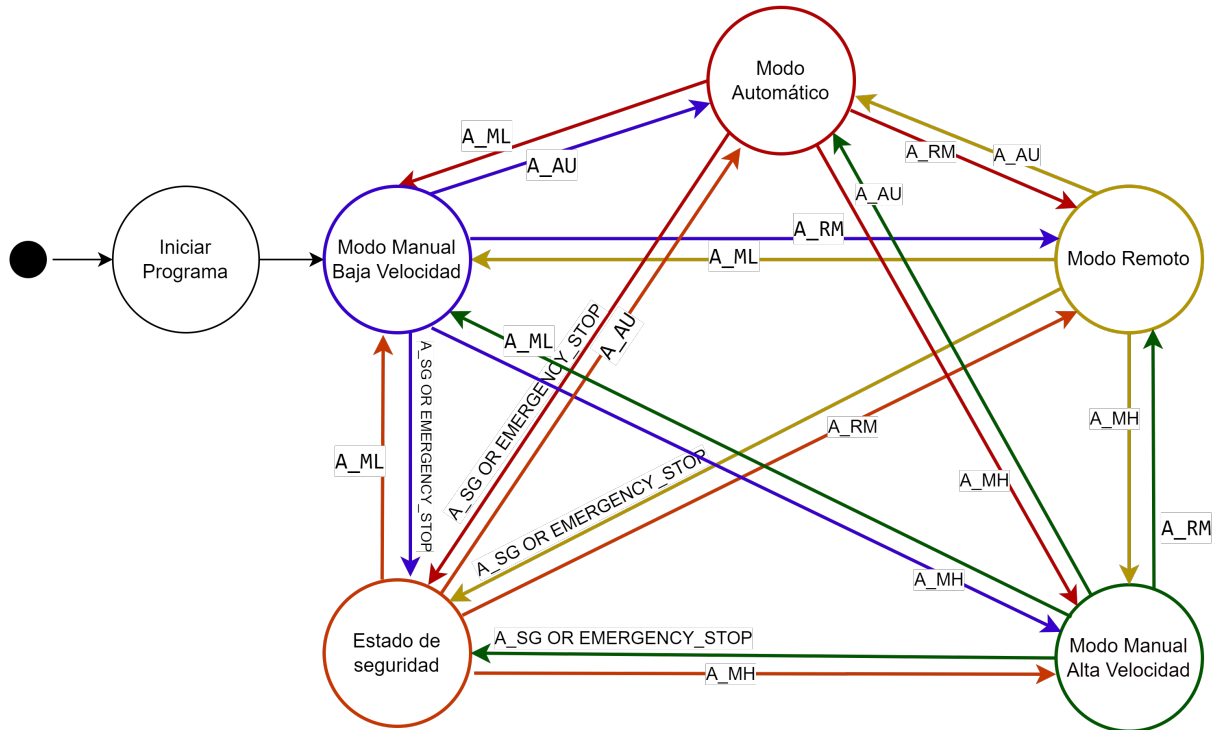


Figura A.6: Diagrama de máquina de estados

# Diseño del hardware de la arquitectura de control

## B.1. Guarda de encierro

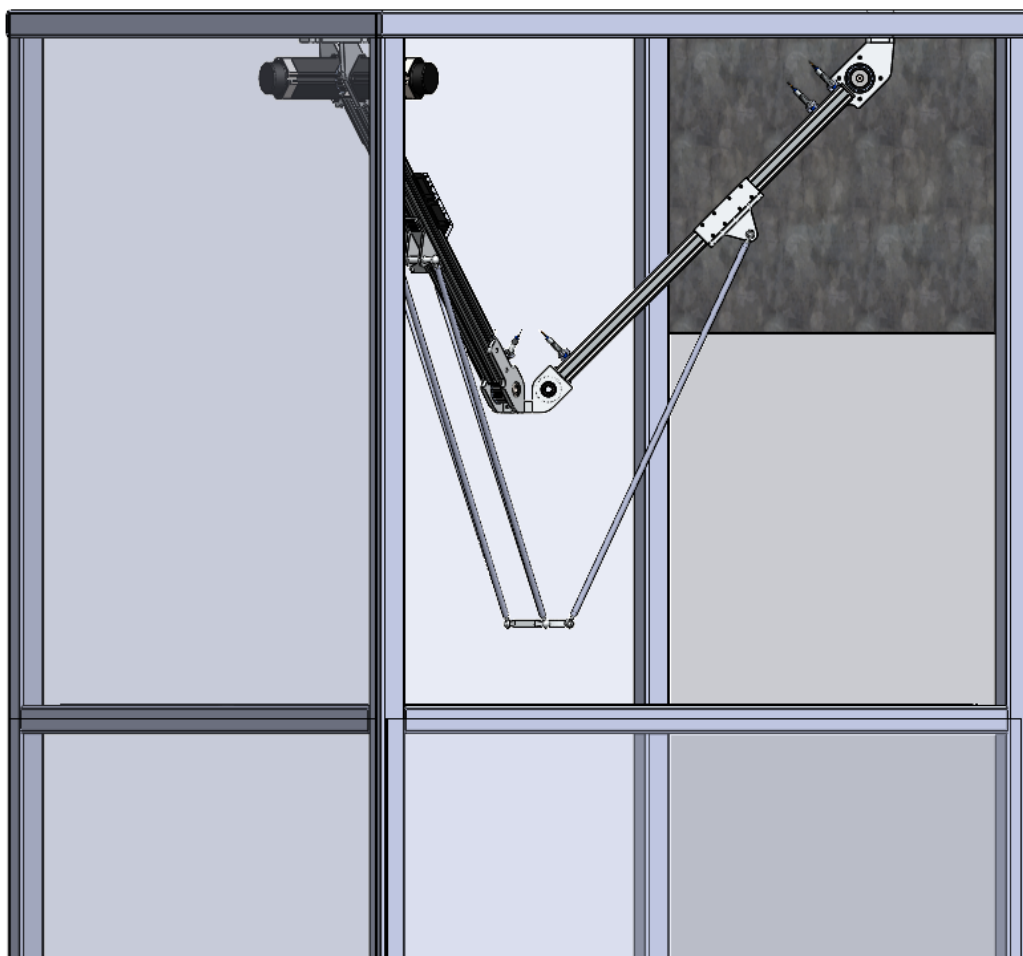


Figura B.1: Diseño de las guardas de seguridad

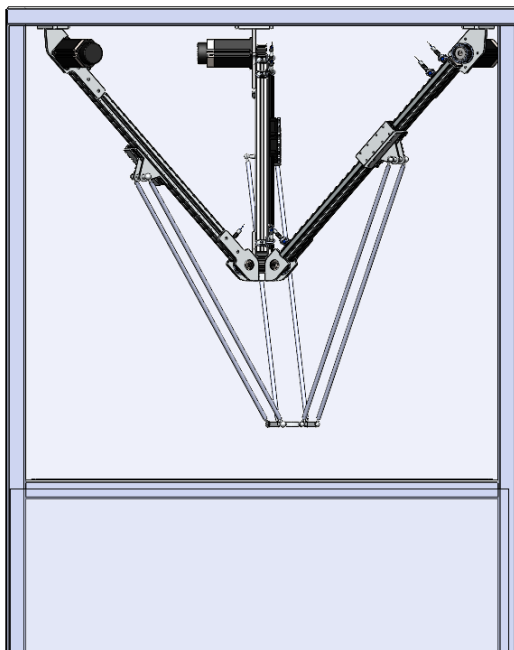


Figura B.2: Diseño de las guardas de seguridad (frontal)

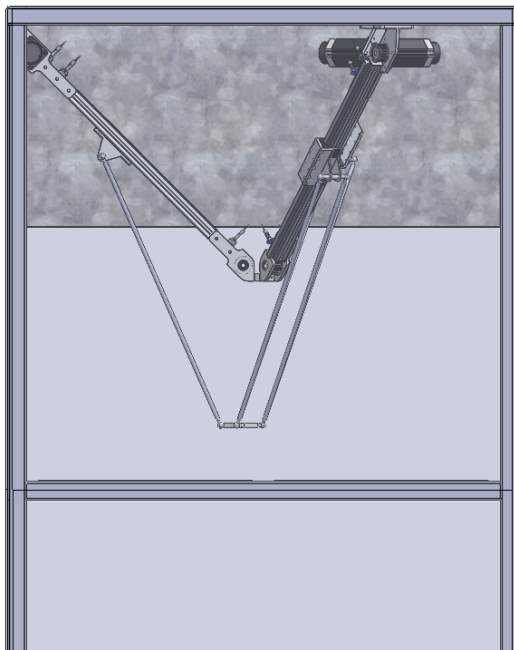


Figura B.3: Diseño de las guardas de seguridad (izquierda)

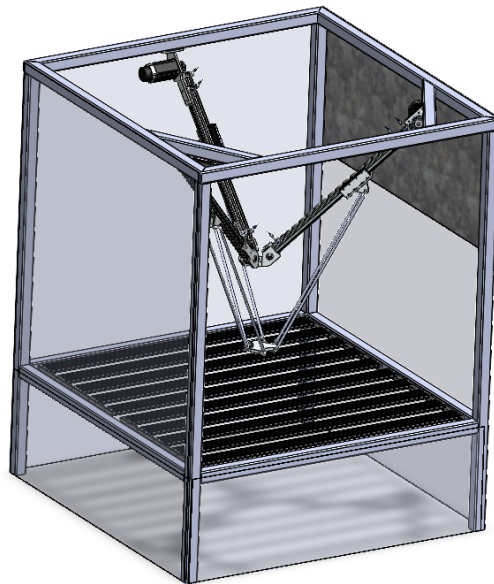


Figura B.4: Diseño de las guardas de seguridad (completa)

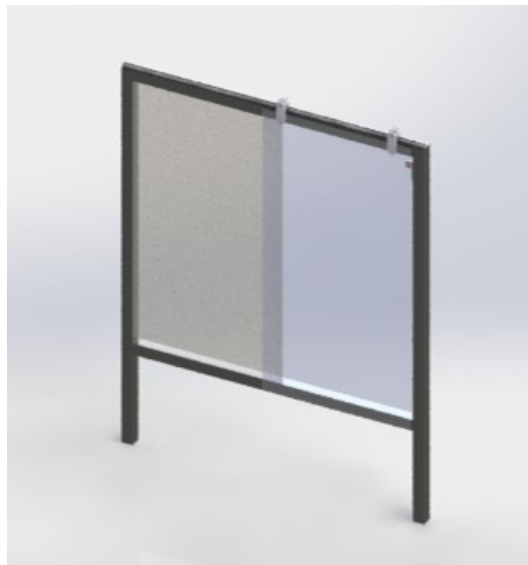


Figura B.5: Diseño de la puerta de acceso a la celda

## B.2. Componentes del Teach Pendant

Criterios	Dispositivo: Computador					
	Raspberry Pi 3 Modelo B+ Tablero (3B+)		Kit Raspberry Pi 4 – 4GB RAM – Starter Pack – Kit Básico		NVIDIA Jetson Nano Kit de desarrollador (945-13450-0000-100)	
	Descripción	Calificación	Descripción	Calificación	Descripción	Calificación
Compatibilidad con python	Si	3	Si	3	Si	2
HDMI Integrado	Si	3	Si	3	Si	3
WiFi Integrada	Si	3	Si	3	No	1
Pines GPIO	Si	3	Si	3	Si	3
Precio	US \$159.99	2	US \$165.99	1	US \$149.00	3
Capacidad de Memoria RAM	1GB	1	4GB	3	4GB	3
Calificación final	15		16		15	

Cuadro B.11: Comparación de dispositivos para selección del Computador

Criterios	Dispositivo: Pantalla					
	Display Backpack HDMI 7" - Con Touchscreen		ROADOM Pantalla Raspberry Pi, monitor de pantalla táctil de 10.1 pulgadas, IPS FHD 1024 x 600, sensible y suave al tacto, entrada HDMI, fácil montaje		Display LCD 7" HDMI/IPS Capacitivo + Audio	
	Descripción	Calificación	Descripción	Calificación	Descripción	Calificación
Tamaño de la pantalla	7 pulgadas	3	10.1 pulgadas	1	7 pulgadas	3
Touch	Si	3	Si	3	Si	3
Compatibilidad HDMI	Si	3	Si	3	Si	3
Precio	\$630.900,00 COP	1	\$522.298,00 COP	2	\$430.000,00 COP	3
Resolucion maxima	800x480 pixeles	1	1024 x 600 pixeles	2	1025 x 600 pixeles	2
Montura	Si	3	Si	3	Si	3
Calificación final	14		14		17	

Cuadro B.12: Comparación de dispositivos para selección de la Pantalla

Criterios	Dispositivo: Switch de habilitacion 3 posiciones					
	A4E-C211VA		HE2B-M211 IDEC Switches, HE2B Series		Enabling switches and Accessories 016P3P33F 3-position enabling switch, 3 N.O.+3 N.C, Faston terminals 2.8x0.5	
	Descripcion	Calificacion	Descripcion	Calificacion	Descripcion	Calificacion
Cubierta plastica	Si	3	No	1	No	1
Numero de contactos	4	3	4	3	4	3
Montura	Si	3	No	1	No	1
Precio	\$487.620,00 COP	2	\$1.340.800,00 COP	1	\$460.000,00 COP	3
Marca	Omron	3	IDEC	3	ELFIN	1
Norma (ANSI RIA 15.06)	Si	3	No	1	No	1
Calificacion final	17		10		10	

Cuadro B.13: Comparación de dispositivos para selección del Switch de seguridad

B.3. Diseño del Teach Pendant

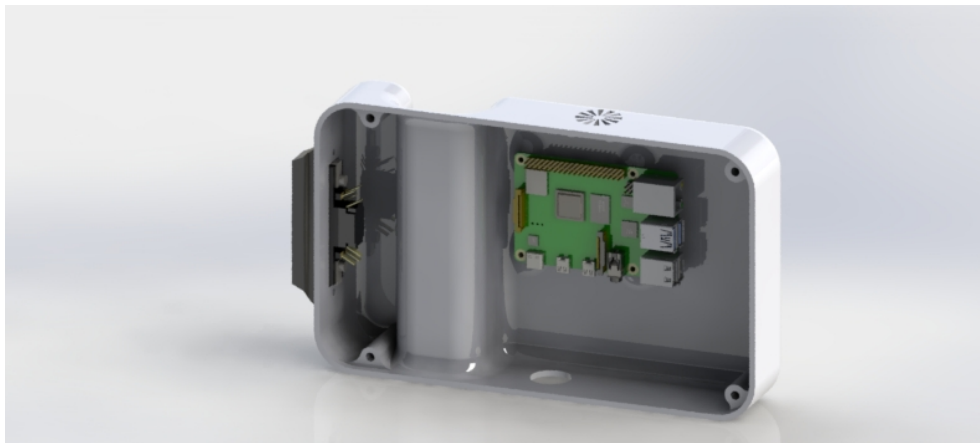


Figura B.7: Diseño de caja de la estructura del teach pendant

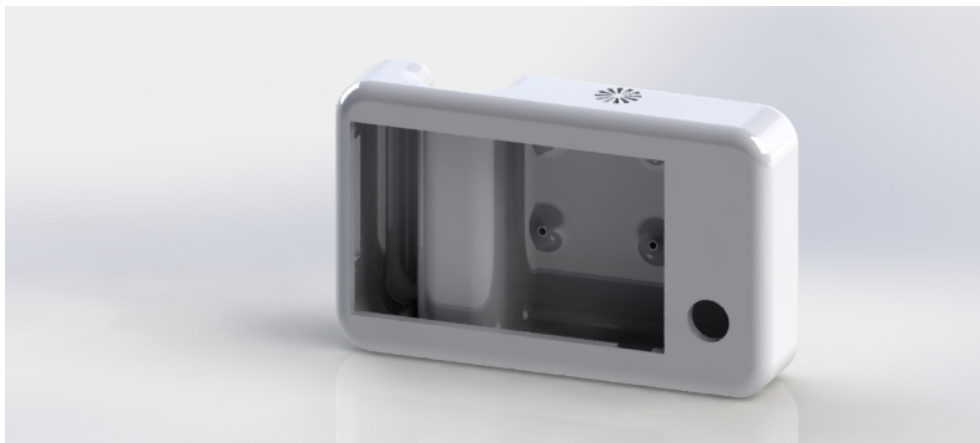


Figura B.6: Diseño de estructura ensamblada del teach pendant

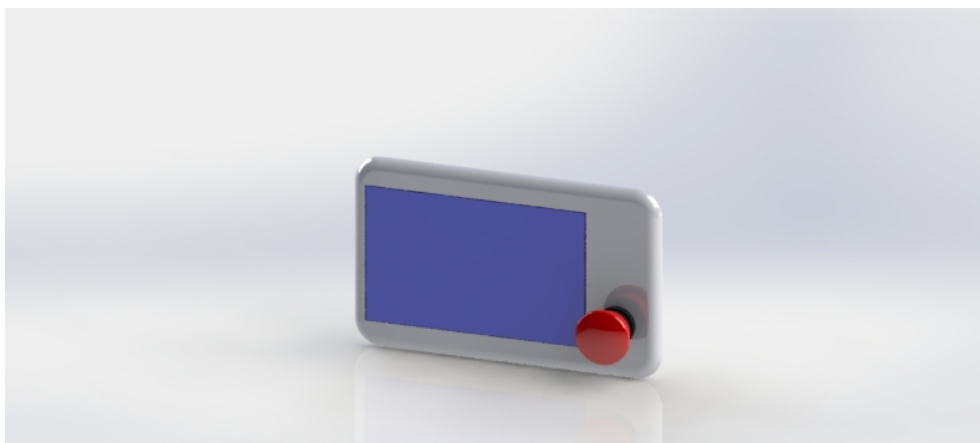


Figura B.8: Diseño de tapa la estructura del teach pendant

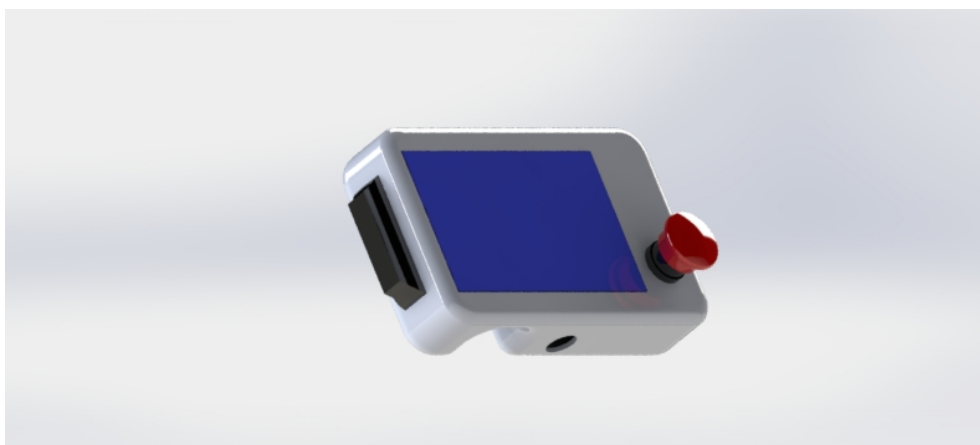


Figura B.9: Diseño de estructura ensamblada del teach pendant con componentes

B.4. Parada de emergencia interna

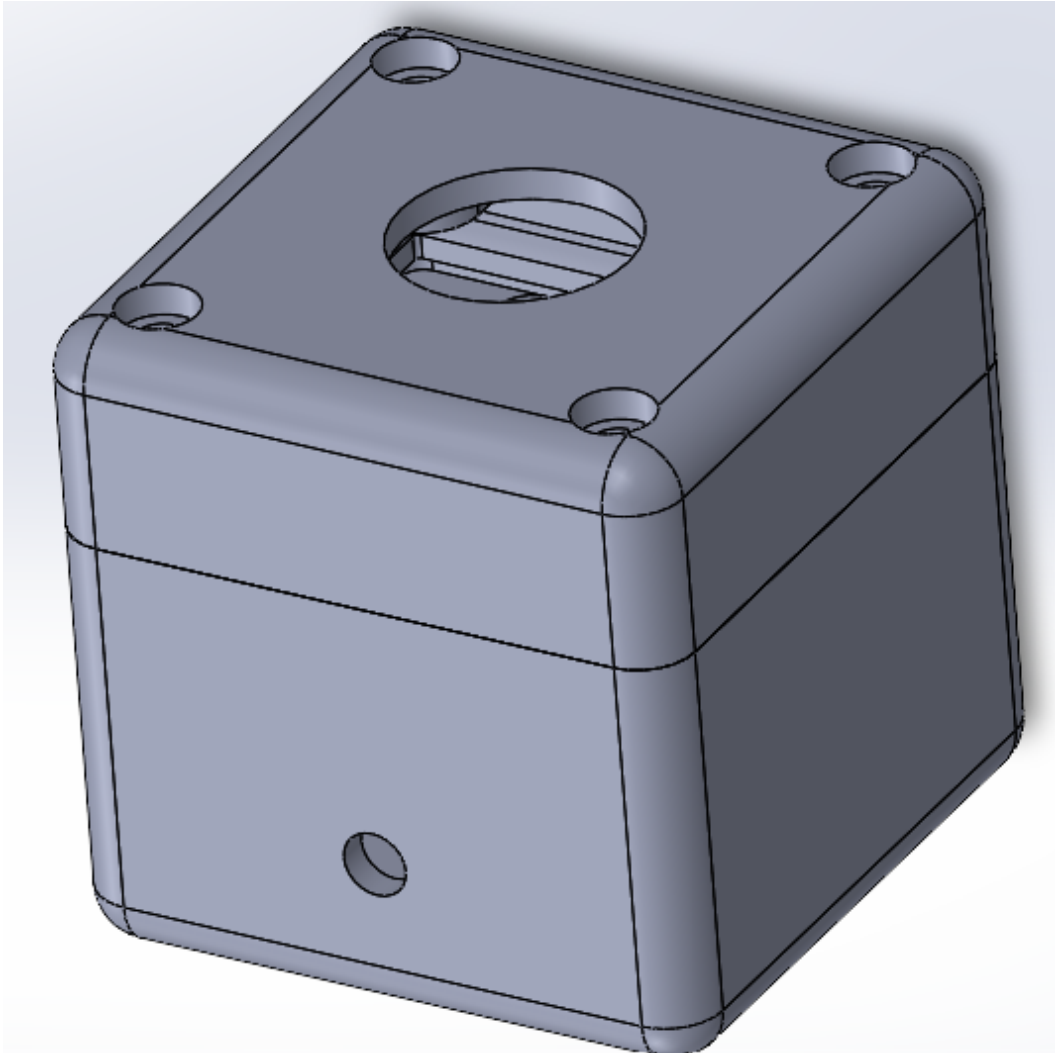


Figura B.10: Diseño de estructura ensamblada para instalación de botón de emergencia interno

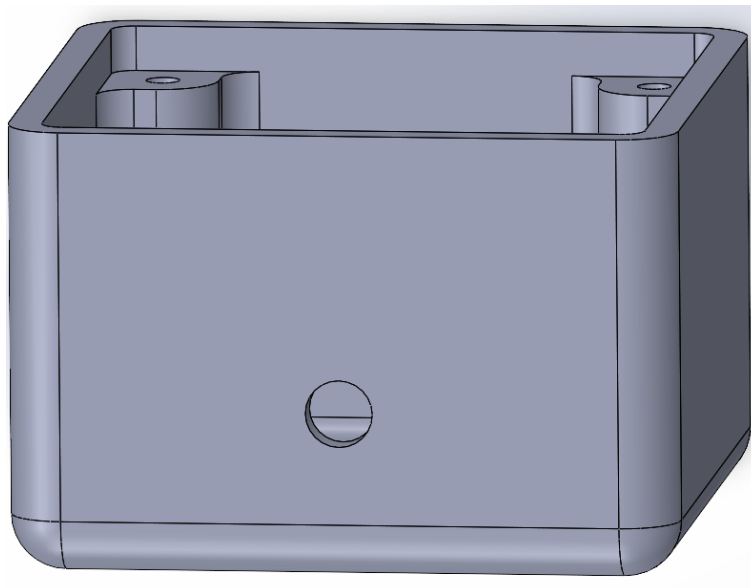


Figura B.11: Diseño de caja para instalación de botón de emergencia interno (vista 1)

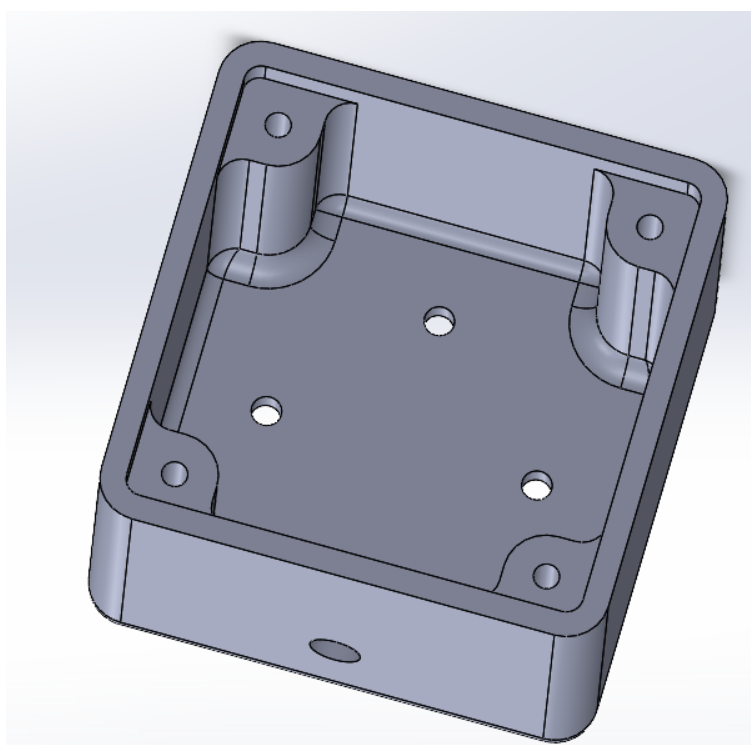


Figura B.12: Diseño de caja para instalación de botón de emergencia interno (vista 2)

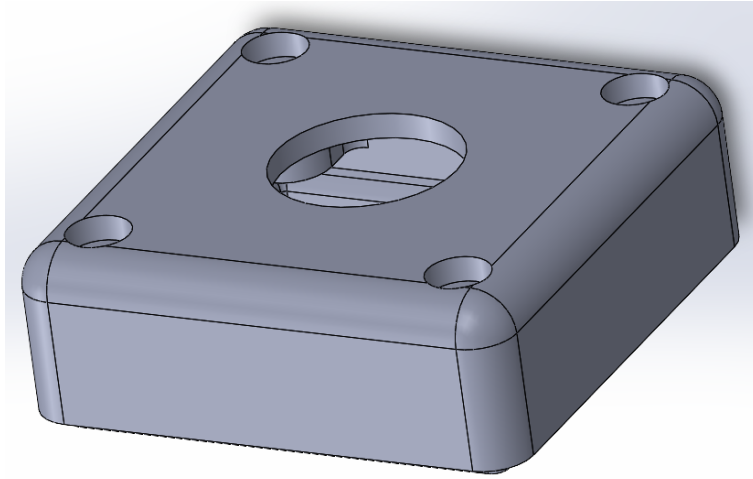


Figura B.13: Diseño de tapa para instalación de botón de emergencia interno (vista 1)

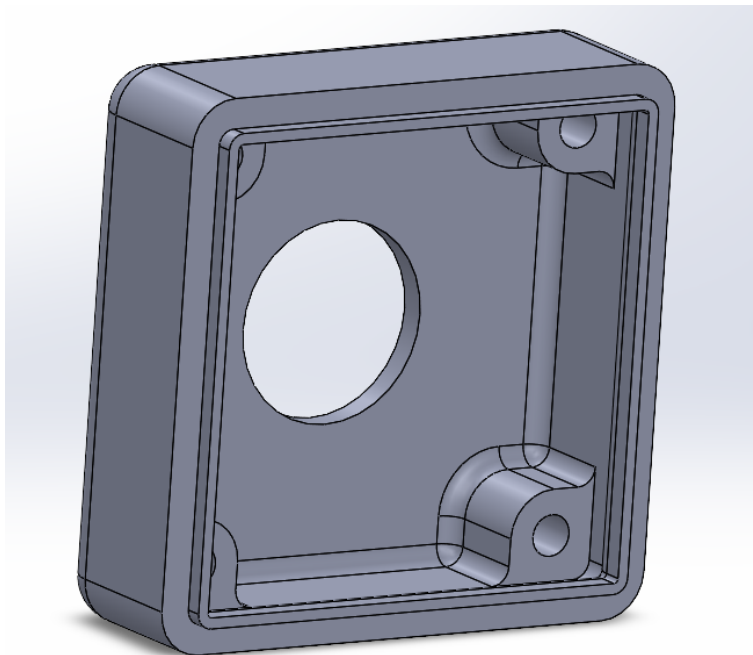


Figura B.14: Diseño de tapa para instalación de botón de emergencia interno (vista 2)



# Implementación del hardware de la arquitectura de control

---

## C.1. Guarda de encierro



Figura C.1: Celda robótica con las guardas y puerta implementadas (vista 1)

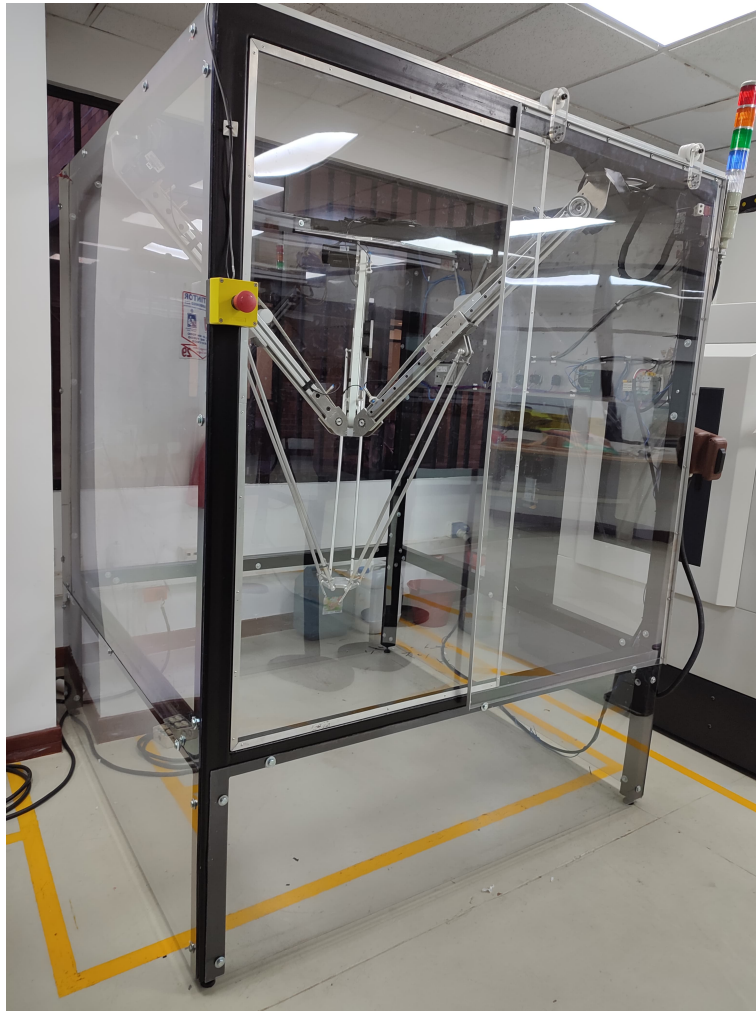


Figura C.2: Celda robótica con las guardas y puerta implementadas (vista 2)

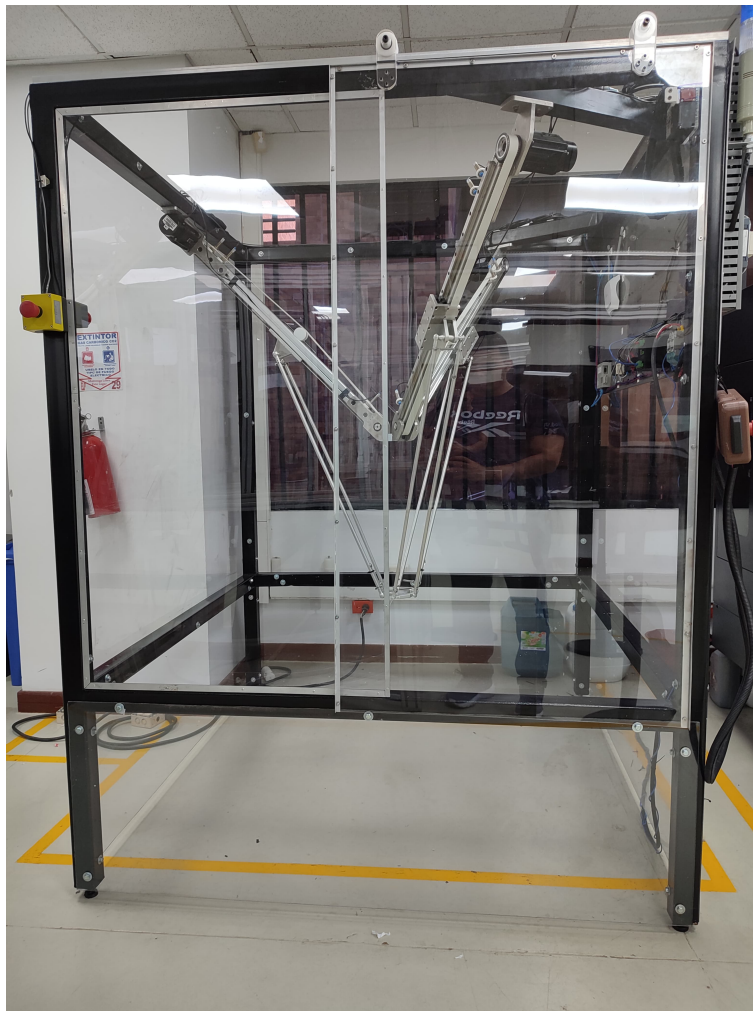


Figura C.3: Celda robótica con las guardas y puerta implementadas (vista 3)

## C.2. Teach Pendant



Figura C.4: Estructura ensamblada del teach pendant



Figura C.5: Caja para la estructura del teach pendant



Figura C.6: Tapa para la estructura del teach pendant



Figura C.7: Estructura con componentes del teach pendant

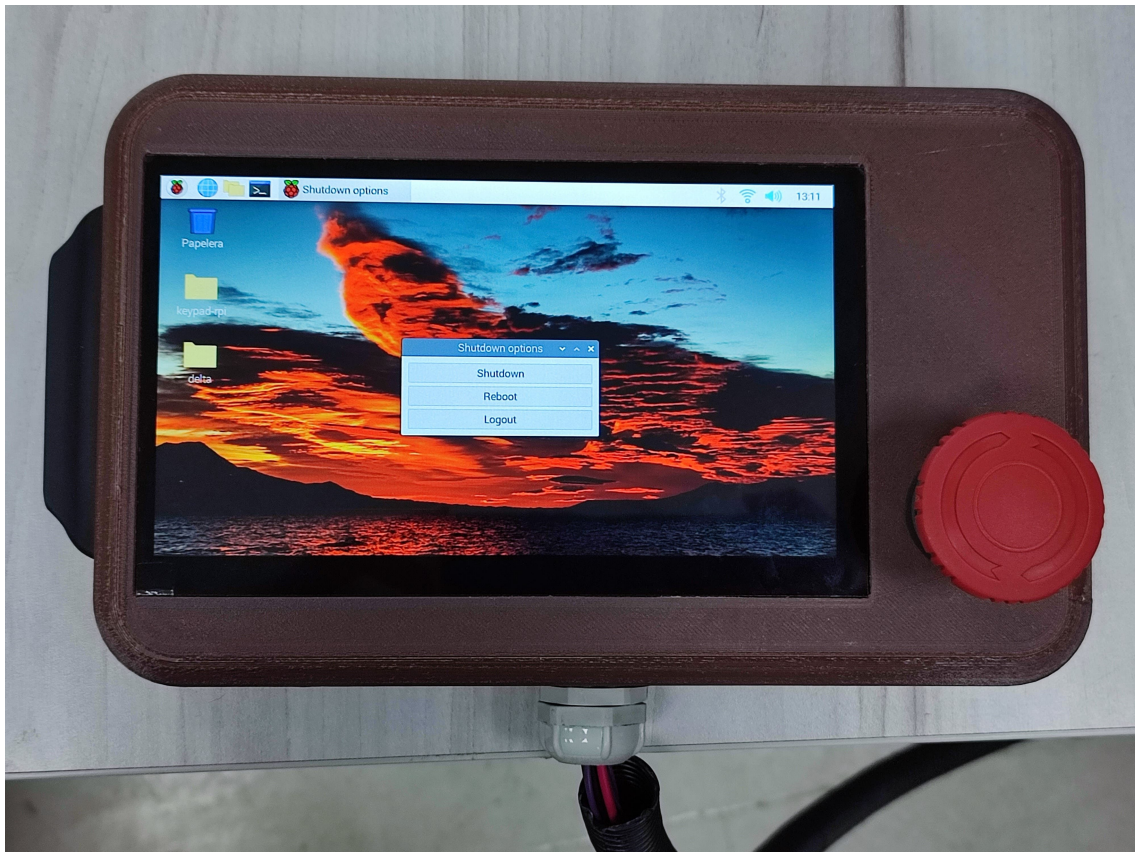


Figura C.8: Teach pendant completo

C.3. Parada de emergencia interna

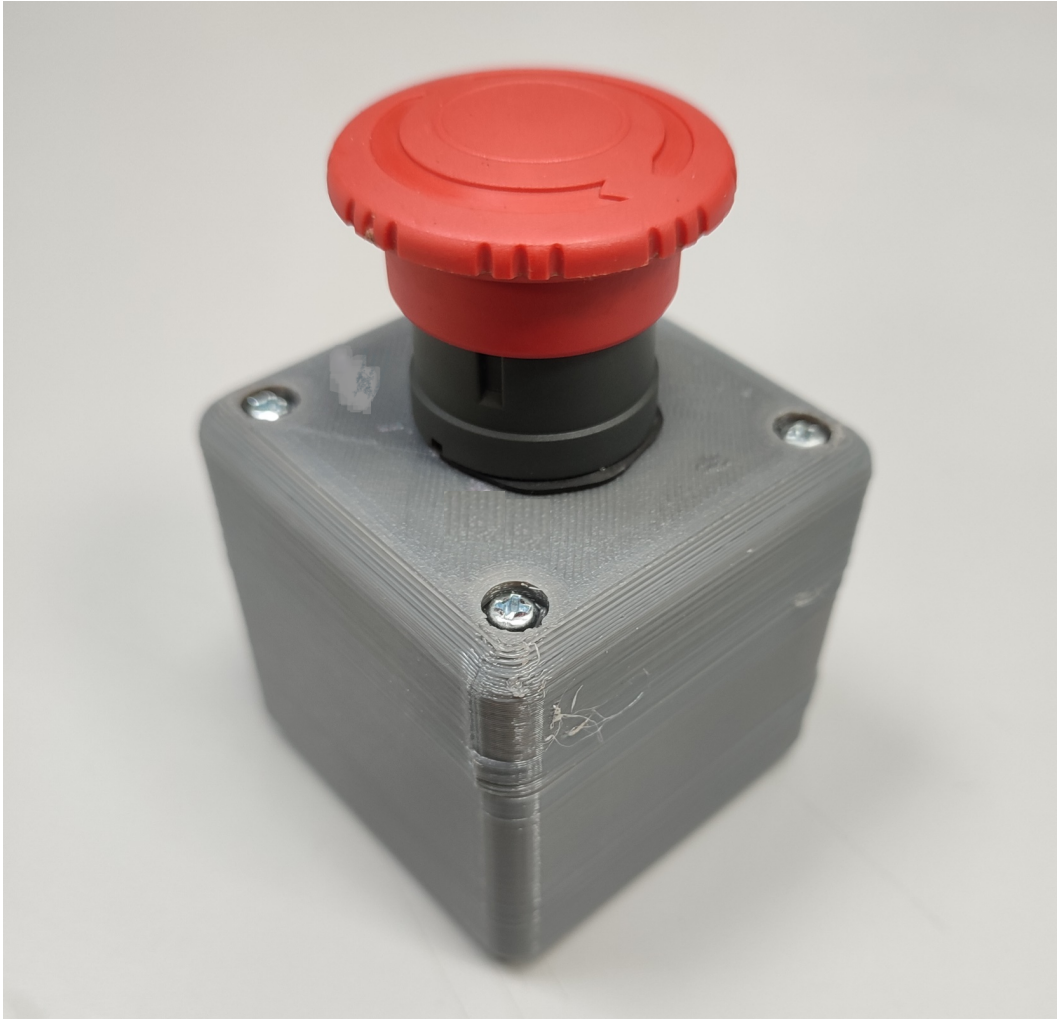


Figura C.9: Estructura de botón de emergencia interno ensamblada

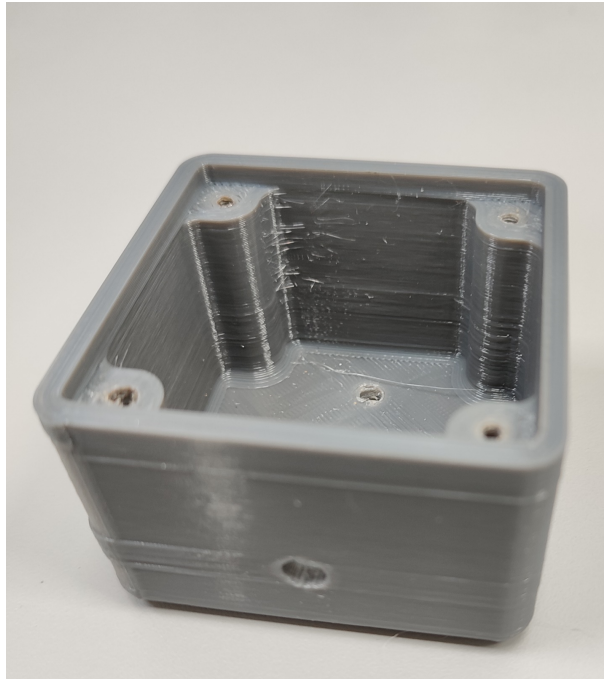


Figura C.10: Caja para la estructura de botón de emergencia interno (vista 1)

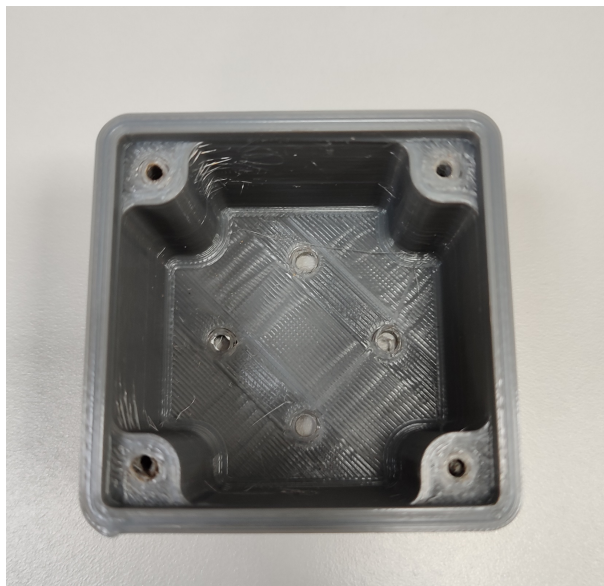


Figura C.11: Caja para la estructura de botón de emergencia interno (vista 2)

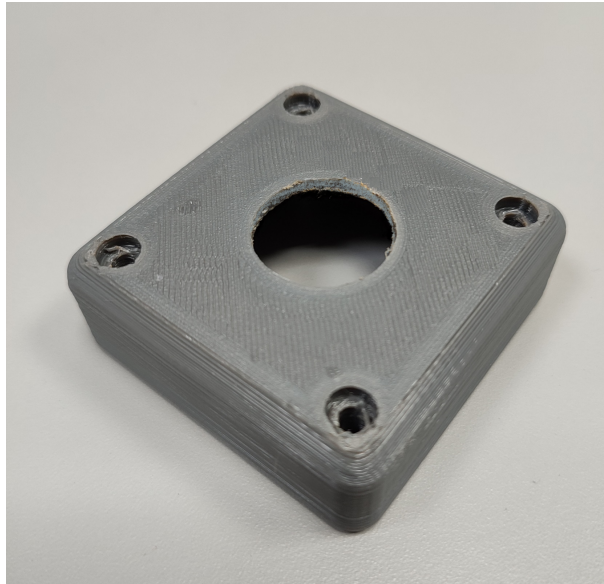


Figura C.12: Tapa para la estructura de botón de emergencia interno (vista 1)

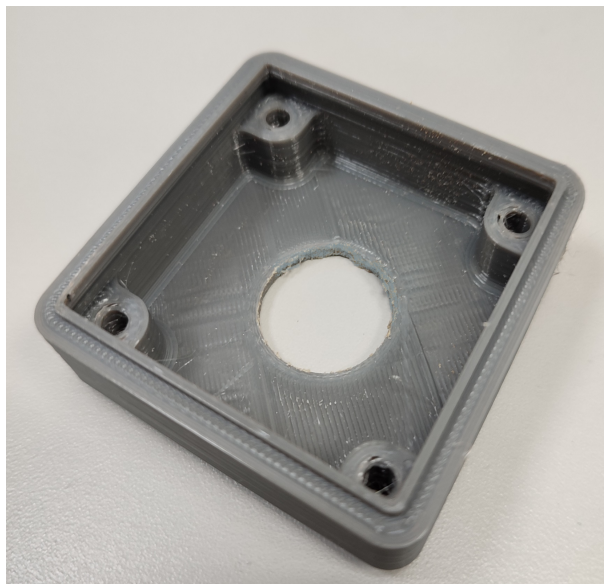


Figura C.13: Tapa para la estructura de botón de emergencia interno (vista 2)