

---

# Restauración de la instalación eléctrica y el sistema de control de un prototipo de transelevador trilateral para el acomodamiento de tarimas de carga en una bodega

---

Herbert Rodrigo Barrios Polanco



UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Restauración de la instalación eléctrica y el sistema de control  
de un prototipo de transelevador trilateral para el  
acomodamiento de tarimas de carga en una bodega**

Trabajo de graduación presentado por Herbert Rodrigo Barrios Polanco  
para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2024



UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería




**Restauración de la instalación eléctrica y el sistema de control  
de un prototipo de transelevador trilateral para el  
acomodamiento de tarimas de carga en una bodega**

Trabajo de graduación presentado por Herbert Rodrigo Barrios Polanco  
para optar al grado académico de Licenciado en Ingeniería Mecatrónica

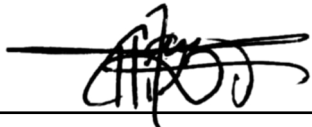
Guatemala,


2024


Vo.Bo.:

(f)   
M. Sc. Carlos Esquit

Tribunal Examinador:

(f)   
M.Sc. Carlos Esquit

(f)   
M. Sc. Miguel Enrique Zea Arenales

(f)   
Ing. Kurt Emmanuel Kellner

Fecha de aprobación: Guatemala, 13 de febrero de 2025.

El presente trabajo de graduación es el resultado de meses de dedicación, esfuerzo, aprendizaje y colaboración. La motivación para llevar a cabo este proyecto surge de la necesidad de restaurar y optimizar un prototipo existente que, si bien no había sido diseñado con fines demostrativos o educativos en un principio, a mi parecer tenía un potencial increíble para poder serlo. Este proyecto no solo refleja los conocimientos y las herramientas adquiridas a lo largo de mi carrera, sino también el esfuerzo por querer aplicarlos de manera práctica y funcional para resolver problemáticas reales e inspirar a nuevas generaciones.

Durante este proceso, enfrenté desafíos continuos que me permitieron fortalecer mis habilidades técnicas, mi capacidad de resolución de problemas y también crecer a nivel personal, académico y profesional. Con este trabajo de graduación concluyo una etapa académica llena de aprendizajes y experiencias significativas que quedaran grabadas en mí para siempre.

Primero que nada, quiero agradecerle a Dios y a la vida por darme la oportunidad de poder vivir y concluir con éxito esta grata experiencia que duró cinco inolvidables años.

Quiero expresar mi más profundo agradecimiento a mis padres, Alejandra Polanco y Herbert Barrios, quienes me han apoyado y amado incondicionalmente toda la vida, por darme la oportunidad de tener una formación profesional superior, siempre empujándome a ser la mejor versión de mí mismo y motivándome a perseguir todos mis sueños. Por inculcar sus valores, enseñarme a ser un caballero y a respetar a los demás. Por brindarme siempre su atención, cariño y tiempo, incluso en las circunstancias más difíciles. Por enseñarme que con una buena actitud, propósitos y metas claras, inteligencia y buena compañía, se puede superar cualquier obstáculo, sobresalir y ser feliz. Por todas sus enseñanzas, consejos y sus continuos recordatorios para ser alguien agradecido con lo que tengo, ser humilde y valorar a los demás. Por ser mi fuente de inspiración y uno de mis motivos para superarme cada día.

Quiero agradecerle a mi hermano, Andrés Barrios, por siempre mostrarme su apoyo y su amor incondicional. Por ser mi motivación a ser una mejor persona cada día. Por siempre sacarme una sonrisa y hacer mi vida mejor. Quiero agradecerles a mis abuelos por criarme, educarme, amarme y apoyarme incondicionalmente. A mi familia, por estar pendiente de mí, por quererme y apoyarme siempre.

Agradezco a mis amigos: Javier Monzón, Cristina Morales, Eduardo Rubin, Luis Pedro Garrido, Carolina Paz, Cristian Catú, Juan Emilio Reyes, Moises Ortiz y Frank Martínez. Por acompañarme a lo largo de estos cinco años de aprendizajes, aventuras, risas, buenos y malos momentos, victorias y fracasos. Por apoyarme, motivarme, inspirarme y convertirse en mi segunda familia y mi equipo de trabajo durante cinco años.

Agradezco a mis asesores de tesis: Pedro Joaquín Castillo y Kurt Emmanuel Kellner, por su invaluable guía, orientación y apoyo durante todo el trabajo de graduación. Quiero agradecerle a mi compañero de trabajo Marlon Vicente, por su dedicación y compromiso con el proyecto, ya que sin él, su diseño mecánico y sus aportes significativos al proyecto, no se hubiera podido restaurar y renovar el prototipo de transelevador.

Para finalizar me gustaría agradecerle a mi casa de estudios, la Universidad del Valle de Guatemala, a mis catedráticos de la carrera, compañeros de promoción, y a todas las personas que, directa o indirectamente, me apoyaron e hicieron posible este logro.

Este trabajo es una muestra del impacto que tiene la perseverancia y el trabajo arduo, el compromiso, la dedicación y el trabajo en equipo para alcanzar metas significativas. Dedico este trabajo a todas las personas mencionadas anteriormente, a todos los que creyeron en mí y a las futuras generaciones que utilizarán este prototipo.

<b>Prefacio</b>	<b>IV</b>
<b>Lista de figuras</b>	<b>IX</b>
<b>Lista de cuadros</b>	<b>X</b>
<b>Resumen</b>	<b>XI</b>
<b>Abstract</b>	<b>XII</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Antecedentes</b>	<b>3</b>
2.1. Prototipo de un transelevador para el acomodamiento de cargas en bodega de Grupo Roca Fuerte, Guatemala . . . . .	3
2.2. Proveedores de transelevadores a nivel mundial . . . . .	8
2.2.1. Mecalux . . . . .	8
2.2.2. HUBMASTER® Warehouse Automation Systems . . . . .	10
<b>3. Justificación</b>	<b>12</b>
<b>4. Objetivos</b>	<b>14</b>
<b>5. Alcance</b>	<b>15</b>
<b>6. Marco teórico</b>	<b>17</b>
6.1. Transelevadores con microprocesadores . . . . .	17
6.1.1. Modos de operación: . . . . .	18
6.1.2. Ventajas . . . . .	19
6.2. Sistema de Gestión de Almacenes (SGA) . . . . .	20
6.2.1. Capacidades funcionales . . . . .	21
6.2.2. Tipos de infraestructura . . . . .	22
6.2.3. Ventajas . . . . .	23
6.3. Elementos para el desarrollo del prototipo de transelevador trilateral a escala	24

6.3.1. Motor paso a paso	24
6.3.2. Arduino Mega 2560	26
6.3.3. RepRap Arduino Mega Pololu Shield (Ramps 1.4)	27
6.3.4. DRV8825	28
6.3.5. Finales de carrera	30
6.3.6. Raspberry Pi 3	31
<b>7. Desarrollo del algoritmo de control del transelevador</b>	<b>33</b>
7.1. Selección de materiales y componentes	34
7.1.1. Arduino Mega 2560	34
7.1.2. RepRap Arduino Mega Pololu Shield (Ramps 1.4)	35
7.1.3. Driver DRV8825	35
7.1.4. Motores NEMA17	36
7.1.5. Finales de carrera	37
7.1.6. Fuente de poder	38
7.2. Calibración de los drivers DRV8825	39
7.3. Configuración inicial y pruebas básicas	42
<b>8. Algoritmo de control</b>	<b>44</b>
8.1. Máquina de estado finito ( <i>FSM</i> )	45
8.1.1. Esperando comando ( <i>WAITING_FOR_COMMAND</i> )	45
8.1.2. Mover a posición ( <i>MOVE_TO_POSITION</i> )	46
8.1.3. Recoger ( <i>PICK</i> )	46
8.1.4. Dejar ( <i>PLACE</i> )	46
8.1.5. Regresar a la posición de origen ( <i>RETURN_HOME</i> )	47
8.1.6. Acción luego de recoger ( <i>POST_PICK_ACTION</i> )	47
8.1.7. Error ( <i>ERROR</i> )	47
8.2. Integración de finales de carrera	48
8.3. Optimización y corrección de errores	49
8.4. Resultados	50
<b>9. Desarrollo de la interfaz gráfica de usuario</b>	<b>51</b>
9.1. Configuración Raspberry Pi 3	52
9.1.1. Instalación y configuración del sistema operativo	52
9.1.2. Configuración de pantalla táctil	53
9.1.3. Acceso remoto y control	53
9.1.4. Optimización del entorno de trabajo	54
9.2. Implementación de la interfaz gráfica de usuario	55
9.2.1. Diseño	55
9.2.2. Programación	56
9.2.3. Envío de comandos y lógica interna	57
9.2.4. Pruebas locales y validación	58
9.2.5. Programación defensiva	59
<b>10. Resultados</b>	<b>61</b>
<b>11. Conclusiones</b>	<b>63</b>
<b>12. Recomendaciones</b>	<b>64</b>

<b>13. Bibliografía</b>	<b>65</b>
<b>14. Anexos</b>	<b>67</b>
<b>14.1. Video del funcionamiento</b> . . . . .	67

---

## Lista de figuras

---

1.	Sistema de traslación horizontal [1] . . . . .	4
2.	Sistema de traslación vertical [2] . . . . .	5
3.	Efecto final (Horquillas) [3] . . . . .	5
4.	Interfaz gráfica de usuario ejecutada en la consola de un ordenador [4] . . . . .	6
5.	Prototipo de Transelevador Trilateral a escala [3] . . . . .	7
6.	Prototipo de Transelevador Trilateral en funcionamiento [3] . . . . .	8
7.	Transelevador Trilateral - Mecalux [6] . . . . .	9
8.	Bodega Automatizada - Mecalux [6] . . . . .	9
9.	Sistema HUBMASTER® [7] . . . . .	10
10.	Transelevador Trilateral - Hubmaster . . . . .	11
11.	Consola de usuario táctil (HMI) - Hubmaster . . . . .	11
12.	Efecto final de un transelevador con microprocesador - Mecalux [6] . . . . .	18
13.	Sistema de gestión de almacenes Easy WMS (Mecalux) [10] . . . . .	20
14.	Diagrama modular y de procesos WMS (Mecalux) [10] . . . . .	22
15.	Easy WMS con conexión wifi (Mecalux) [10] . . . . .	22
16.	Motor paso a paso híbrido Nema17 [12] . . . . .	24
17.	Arduino Mega 2560 [15] . . . . .	27
18.	RepRap Arduino Mega Pololu Shield (Ramps 1.4) [16] . . . . .	28
19.	Driver DRV8825 [17] . . . . .	29
20.	Final de carrera para impresora 3D [18] . . . . .	31
21.	Raspberry Pi 3 [19] . . . . .	32
22.	Arduino Mega 2560 [15] . . . . .	34
23.	Montaje del módulo Ramps con la placa Arduino Mega 2560 . . . . .	35
24.	Driver DRV8825 [17] . . . . .	36
25.	Motor Nema17 [12] . . . . .	36
26.	Diagrama de movimientos . . . . .	37
27.	Final de carrera [18] . . . . .	38
28.	Fuente de poder 12V - 10A . . . . .	39
29.	Materiales para la calibración de los drivers DRV8825 . . . . .	40
30.	Calibración de los drivers DRV8825 . . . . .	41
31.	Espacios para conectores en el módulo Ramps 1.4 . . . . .	42

32. Componentes utilizados en las pruebas . . . . .	42
33. Conexiones realizadas para las pruebas iniciales . . . . .	43
34. Diagrama de conexiones del prototipo de transelevador trilateral . . . . .	43
35. Diagrama de flujo del algoritmo de control . . . . .	45
36. Pruebas realizadas con los finales de carrera antes de ensamblar el prototipo . . . . .	49
37. Conexiones y configuración de la pantalla táctil con la Raspberry Pi 3 . . . . .	53
38. Acceso remoto a Raspberry Pi 3 por medio de RVNC . . . . .	54
39. Esbozo de la pantalla de inicio . . . . .	55
40. Esbozo del menú de operación . . . . .	56
41. Esbozo del menú de inventario . . . . .	56
42. Pantalla de inicio . . . . .	58
43. Menú de operación . . . . .	59
44. Menú de operación . . . . .	59
45. Pruebas locales de envío de comandos por comunicación serial . . . . .	59
46. Pruebas de validación finales del prototipo de transelevador trilateral . . . . .	62
47. Prototipo de Transelevador Trilateral a escala en funcionamiento . . . . .	62
48. Vista frontal del prototipo de transelevador terminado . . . . .	62
49. Código QR para ver el funcionamiento del prototipo de transelevador trilateral . . . . .	67

---

Lista de cuadros

---

1. Posibles combinaciones de micro pasos - Driver DRV8825 [20] . . . . . 41

El presente trabajo de graduación tuvo como objetivo restaurar un prototipo de transelevador trilateral, ubicado en la Universidad del Valle de Guatemala, el cual fue diseñado para automatizar el almacenamiento y recolección de tarimas en una bodega simulada. El prototipo presentaba deficiencias en su instalación eléctrica, control operacional y usabilidad, limitando su funcionalidad y potencial como un proyecto demostrativo o herramienta educativa.

La restauración incluyó el rediseño e implementación de un algoritmo de control basado en una máquina de estados finitos en una placa Arduino Mega 2560. Este permitió al transelevador ejecutar movimientos precisos en los ejes X, Y y Z, así como giros de  $180^\circ$  para operar en pasillos paralelos. Se incorporaron finales de carrera para recalibración automática, aumentando la precisión y fiabilidad del sistema. Por otra parte, se desarrolló una interfaz gráfica en Python, la cual se ejecuta en una pantalla táctil mediante una Raspberry Pi 3. Esta facilitó la interacción con el prototipo y gestionó un inventario virtual sencillo en tiempo real.

Los sistemas restaurados fueron validados mediante pruebas funcionales que evidenciaron un prototipo preciso, eficiente y completamente operativo, capaz de automatizar tareas de almacenamiento y recolección de tarimas en una bodega simulada. Al ser un proyecto demostrativo y funcional, el cual abarca conocimientos de distintos cursos y herramientas aprendidas a lo largo de la licenciatura en ingeniería mecatrónica, este aporta valor educativo, técnico y demostrativo, como una herramienta de aprendizaje en automatización aplicada.

The objective of this graduation project was to restore a trilateral stacker crane prototype located at Universidad del Valle de Guatemala. The prototype was designed to automate the storage and retrieval of pallets in a simulated warehouse. However, it exhibited deficiencies in its electrical installation, operational control, and usability, limiting its functionality and potential as a demonstrative project or educational tool.

The restoration included the redesign and implementation of a control algorithm based on a finite state machine, implemented on an Arduino Mega 2560 development board. This allowed the stacker crane to execute precise movements along the X, Y, and Z axes, as well as 180° rotations to operate in parallel aisles. Limit switches were incorporated for automatic recalibration, enhancing the system's precision and reliability. Additionally, a graphical user interface (GUI) was developed in Python, running on a touchscreen powered by a Raspberry Pi 3. This interface facilitated interaction with the prototype and managed a simple real-time virtual inventory.

The restored systems were validated through functional testing, which demonstrated a precise, efficient, and fully operational prototype capable of automating pallet storage and retrieval tasks in a simulated warehouse. As a functional and demonstrative project encompassing knowledge from various courses and tools acquired throughout the mechatronics engineering program, it provides educational, technical, and demonstrative value as a learning tool for applied automation.

El avance en la automatización ha transformado significativamente la forma en que las industrias manejan sus operaciones logísticas, como el almacenamiento y la recuperación de mercancías. Entre estas soluciones, los transelevadores han demostrado ser sistemas adecuados para optimizar el uso del espacio, reducir tiempos operativos y mejorar la precisión en la gestión de inventarios.

El presente trabajo de graduación tuvo como objetivo restaurar un prototipo de transelevador trilateral, ubicado en la Universidad del Valle de Guatemala, el cual fue diseñado para automatizar el almacenamiento y la recolección de tarimas en una bodega simulada. No obstante, este presentaba deficiencias significativas en su instalación eléctrica, control operacional y en la interacción con el usuario, lo que limitaba su funcionalidad y potencial como proyecto demostrativo y herramienta educativa.

La restauración incluyó el rediseño de un algoritmo de control basado en una máquina de estados finitos, el cual fue implementado en una placa de desarrollo Arduino Mega 2560. Este algoritmo permitió controlar con precisión los movimientos del transelevador en los ejes X, Y y Z, además de un giro de  $180^\circ$  para poder operar en pasillos paralelos dentro de una bodega simulada. Se utilizaron finales de carrera para que el nuevo prototipo de transelevador trilateral contara con una recalibración automática al finalizar cada ciclo de trabajo, aumentando la redundancia en la operación del sistema. El algoritmo de control también incorporó programación defensiva la cual aseguró la detección de errores y una respuesta adecuada ante posibles comandos inválidos, mejorando la fiabilidad del sistema.

Se desarrolló una interfaz gráfica en Python, la cual se ejecuta en una pantalla táctil de siete pulgadas mediante el ordenador de placa única Raspberry Pi 3. Esta interfaz permitió a los usuarios interactuar de manera sencilla con el prototipo, permitiendo seleccionar distintos comandos para poder operar el transelevador trilateral por medio de botones y menús desplegados. La comunicación serial entre la Raspberry Pi 3 y el Arduino Mega 2560 aseguró una transferencia de datos en tiempo real fiable, lo que facilitó la sincronización y el envío de paquetes de datos entre ambas placas. La interfaz gráfica de usuario también incorporaba un inventario virtual sencillo el cual se actualizaba en tiempo real cada vez

que se enviaba un nuevo comando. De igual manera, se implementó programación defensiva contra errores comunes e interacciones no deseadas las cuales podrían llegar a suceder al momento de manipular el prototipo de transelevador trilateral.

La integración de todos los sistemas mencionados anteriormente fue validada por medio de pruebas funcionales que evidenciaron un sistema preciso, eficiente y completamente operativo. El prototipo de transelevador trilateral fue restaurado por completo, logrando integrar un sistema funcional que combina un algoritmo de control robusto, una interfaz gráfica de usuario funcional y fácil de poder utilizar por cualquier persona, programación defensiva y un inventario virtual simple. El sistema demostró alta precisión y eficiencia al automatizar tareas de almacenamiento y recolección de tarimas en una bodega simulada. De igual manera, se evidenció que es un proyecto demostrativo y funcional, el cual abarca conocimientos de distintos cursos y herramientas aprendidas a lo largo de la licenciatura en ingeniería mecatrónica. El prototipo aporta valor educativo, técnico y demostrativo, como una herramienta de aprendizaje en automatización aplicada.

Este trabajo refleja la importancia de aplicar tecnologías emergentes en contextos demostrativos y educativos, promoviendo el aprendizaje activo, la simulación y análisis de posibles escenarios y el desarrollo de soluciones innovadoras para problemáticas reales del presente. El prototipo de un transelevador trilateral a escala, responde a una de las necesidades actuales de la logística moderna, demostrando la viabilidad de soluciones automatizadas para el manejo de almacenes y contribuyendo al desarrollo de habilidades técnicas y aplicadas en estudiantes.

## 2.1. Prototipo de un transelevador para el acomodamiento de cargas en bodega de Grupo Roca Fuerte, Guatemala

Este proyecto de graduación tuvo como objetivo principal innovar en el ámbito industrial de Guatemala por medio del diseño, desarrollo e implementación de una máquina electromecánica que fuera capaz de acomodar productos y cargas de manera rápida y eficiente dentro de una bodega industrial. Los estudiantes que realizaron el trabajo en cuestión fueron Julio José Lazo Valdez [1], Vincenzo Antonio Musella Mombiola [2], Carlos Daniel Hengstenberg Chocooj [3] y Juan Pablo Gomes Villela [4].

El proyecto consiste en un modelo a escala el cual trata de ejemplificar lo que se podría implementar en la bodega Cementos Roca Fuerte, en Villanueva, Guatemala. Se buscaba poder optimizar al máximo el espacio dentro de la bodega, ya que este era bastante reducido. El carro transportador tenía que ser capaz de desplazarse horizontalmente, verticalmente y, también, tener los grados de movimiento necesarios para poder sacar e ingresar cargas en un espacio reducido. Además, debía de contar con una base de datos la cual se actualizaría en tiempo real de forma que fuera posible saber cuánto espacio libre se tenía en cada una de las estanterías o zonas de la bodega. La idea y el patrocinio venía de Progreso Labs, una rama de Grupo Progreso que se dedica al apartado del desarrollo de proyectos e ideas innovadoras, las cuales aporten valor y tengan gran impacto en la empresa.

Progreso Labs es una iniciativa de innovación la cual busca habilitar la ejecución de proyectos de manera colaborativa. Fue fundada por Grupo Progreso en el año 2019 con la finalidad de buscar soluciones a retos que faciliten los procesos constructivos [5]. Ha trabajado en varias ocasiones con la Universidad del Valle de Guatemala, como fue el caso de los cuatro ex alumnos que desarrollaron el prototipo de transelevador trilateral para el acomodamiento de cargas en bodega de Grupo Roca Fuerte.

Todo el diseño mecánico y de piezas individuales en este proyecto se realizó en el programa CAD de *Autodesk Fusion360*. Se utilizó este programa CAD debido a que éste presenta

la característica de que varios usuarios pueden trabajar en tiempo real en un mismo proyecto. De esta manera todos los integrantes del proyecto de Progreso Labs fueron capaces de aportar, diseñar y modificar al proyecto.

La mayoría de las piezas en este proyecto fueron creadas a partir de impresión 3D con PLA o resina y corte láser en MDF. Es importante mencionar que se realizaron los análisis de esfuerzos de todas las partes y piezas involucradas en este diseño. Dentro del trabajo de graduación, también se encuentran los resultados que se obtuvieron de los prototipos que se realizaron previamente y las de sus iteraciones.

El transelevador se encuentra conformado por cuatro partes fundamentales:

- **Sistema mecánico de movimiento horizontal:** Está compuesto por un mecanismo de faja y poleas, el cual se mueve con un motor de pasos. La faja puede ser adaptada por medio de un sistema tensor, el cual permite que se encuentre más ajustada u holgada. Este sistema se encarga de trasladar el carro del transelevador por un riel, de manera que puede avanzar y retroceder.



Figura 1: Sistema de traslación horizontal 1

- **Sistema mecánico de movimiento vertical:** Se encuentra conformado por un carro el cual se desplaza a lo largo de un perfil de aluminio por medio de un motor de pasos. Dicho desplazamiento se da por medio de un sistema de transmisión de potencia con un tornillo sin fin. Este sistema se encarga de poder subir y bajar las tarima de carga dentro de la bodega.

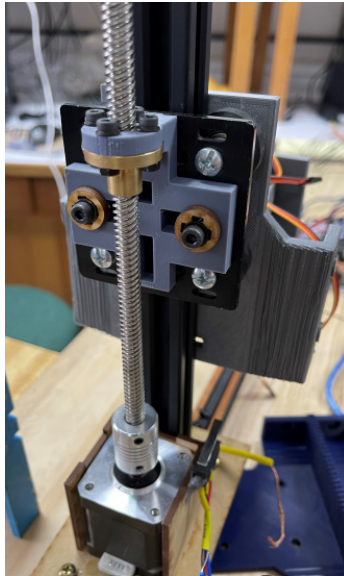


Figura 2: Sistema de traslación vertical **2**

- **Efector final:** Está conformado por un sistema de rotación el cual se mueve con un servo motor y un efector final que tiene dos horquillas. Cuenta con un rango de 180 grados de rotación que permite trabajar de manera paralela en un pasillo de la bodega. El efector final también cuenta con un movimiento lineal el cual permite que este avance o retroceda por medio de un engranaje y una cremallera. Este sistema se encarga de recoger las tarimas de carga y colocarlas de manera cuidadosa y ordenada en los espacios designados de la bodega.

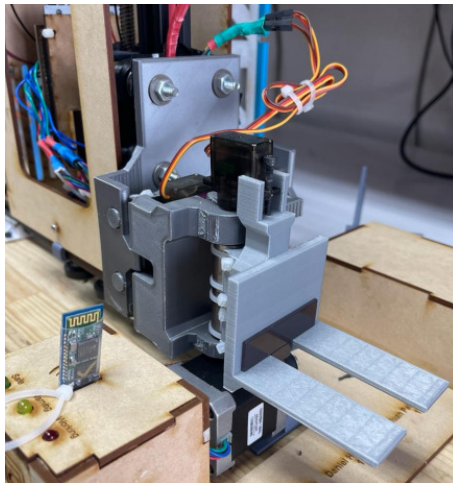


Figura 3: Efector final (Horquillas) **3**

- **Control del transelevador:** Se encarga de integrar y controlar todos los sistemas antes mencionados por medio de un algoritmo de programación basado en un sistema de gestión de almacenes (SGA). Dentro de las capacidades funcionales que este sistema puede entregar se encuentran la reposición, ubicación y expedición de tarimas de carga. Adicionalmente, este sistema también incluye una interfaz gráfica de usuario la cual

puede utilizarse para el registro, consulta, actualización, eliminación y envío de datos.

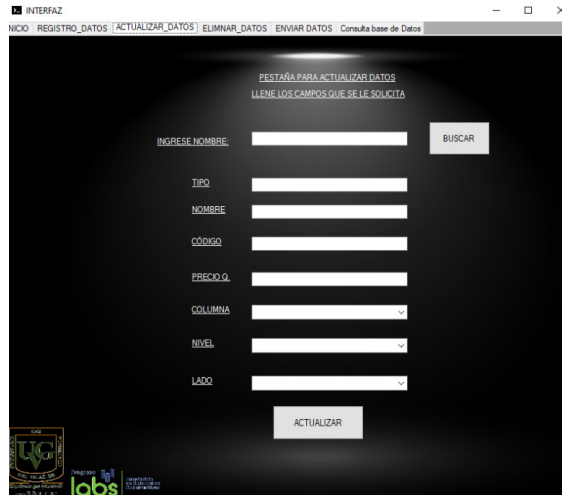


Figura 4: Interfaz gráfica de usuario ejecutada en la consola de un ordenador [4](#)

Cada una de las partes en este proyecto de graduación fue diseñada, construida e implementada por un estudiante distinto. El sistema de traslación horizontal fue trabajado por Julio José Lazo Valdez [1](#), el de sujeción de cargas fue obra de Carlos Daniel Hengstenberg Chocooj [3](#), el de traslación vertical fue elaborado por Vincenzo Antonio Musella Mombiela [2](#) y la integración de la interfaz gráfica y la base de datos fue desarrollado por Juan Pablo Gomes Villela [4](#). Cada uno de los integrantes del proyecto también se encargaba de los sistemas secundarios o que se derivaban de las secciones principales, por ejemplo, los sistemas de alimentación, los actuadores involucrados, entre otros.

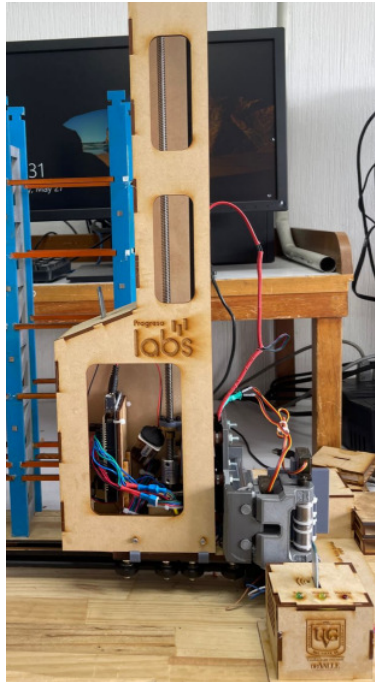


Figura 5: Prototipo de Transelevador Trilateral a escala **3**

Se desarrolló un algoritmo de control en lenguaje *C* y *Python* para poder controlar todos los sistemas y también realizar la comunicación entre la interfaz gráfica y el transelevador. De esta forma, se podía recibir información desde la base de datos, después se procesaba dicha información y por último se mandaba al transelevador de manera que este ejecutara las acciones necesarias para cumplir con la tarea que se le había asignado. Toda la comunicación que existía entre la base de datos y el montacargas automatizado se realizó por medio de dos microcontroladores Arduino y un módulo inalámbrico bluetooth.

El proyecto finalizó con la implementación de una bodega a escala en donde el transelevador fue capaz de realizar todas las acciones y tareas que se esperaba. Se concluyó que la garra era capaz de soportar una carga máxima de 600 gramos, sin embargo, se recomendó que no excediera de los 500 gramos. El sistema era capaz de sujetar y colocar tarimas de carga en estanterías que se encontraban paralelas entre sí, ya que los grados de movimiento del sistema permitían un giro de  $180^\circ$ . El sistema logró cumplir con su objetivo principal, de manera que fue capaz de recolectar y colocar pallets de manera estable, sin comprometer el equilibrio de todo el mecanismo o el funcionamiento de los actuadores.

Por último, los desarrolladores del proyecto recomendaron realizar un prototipo el cual fuera programado en un PLC, de manera que fuera más sencillo implementar la programación a una escala real. Hicieron énfasis en usar motores que incluyeran un encoder rotativo para que la lectura de datos fuera mejor y que de esta forma se tuviera una mejor precisión. También sugirieron implementar sensores infrarrojos, de presencia o de otro tipo, los cuales permitieran tener la información de manera más rápida. Su última recomendación fue agregar un riel o un soporte en la parte superior del transelevador para evitar cualquier tipo de tambaleo o fuerza extra que pudiera llegar a afectar o poner en riesgo el funcionamiento de todo el sistema.

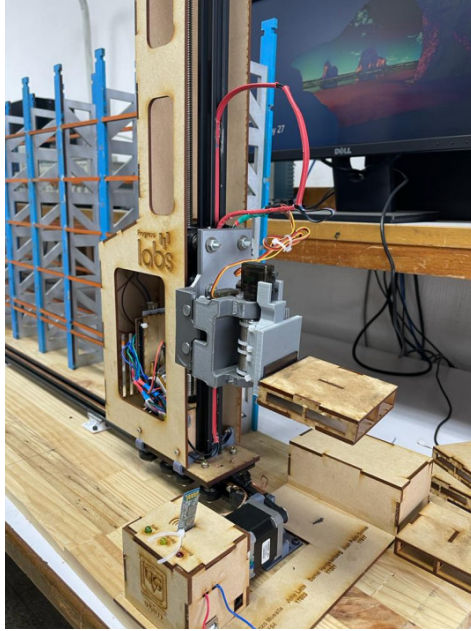


Figura 6: Prototipo de Transelevador Trilateral en funcionamiento [3]

## 2.2. Proveedores de transelevadores a nivel mundial

### 2.2.1. Mecalux

Mecalux es una de las compañías líderes en tecnología intralógica a nivel global, contando con sucursales en México, España, Brasil y otros países, con un alcance total de 70 países en su red comercial. Cuentan con 55 años de experiencia en el mercado y dentro de su catálogo de servicios y productos se puede encontrar la automatización de almacenes, *software* para el control de producto en almacenes, robots móviles autónomos, racks, tarimas y anaqueles.

Dentro de la automatización de almacenes, que es el área de interés, cuentan con transelevadores normales, trilaterales, transportadores, *pallet shuttle*, electrovías y elevadores. El modelo en concreto que se quisiera poder replicar y adaptar sería el transelevador trilateral automatizado. Este presenta dentro de sus características ser una de las soluciones más económicas, tener una implementación sencilla y no tener que modificar las estructuras del almacén. Con esta solución es posible automatizar estanterías convencionales de hasta 15 metros de altura.

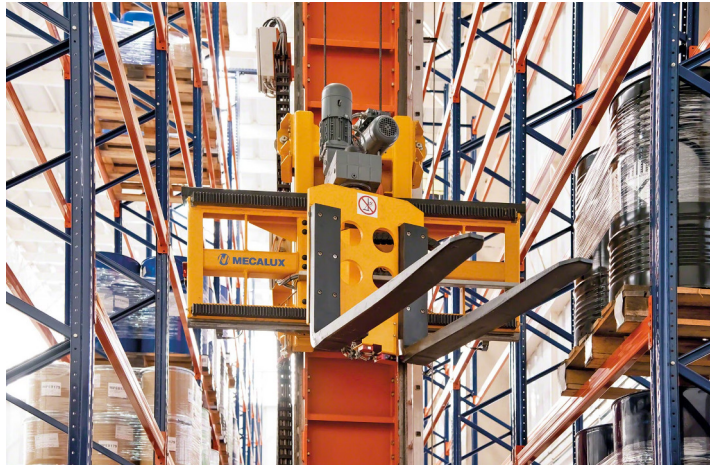


Figura 7: Transelevador Trilateral - Mecalux [6]

Según el fabricante [6], los transelevadores trilaterales automáticos permiten automatizar de forma muy sencilla almacenes de estanterías convencionales que trabajan con montacargas manejados por operarios, tanto para almacenes ya existentes como en instalaciones nuevas. El transelevador mueve las paletas hasta los extremos del pasillo dejando la carga sobre ménsulas o sobre un sistema de transporte automático. Esto es posible gracias a que dispone de un cabezal giratorio que permite recoger y dejar las paletas en tres posiciones: una frontal y otras dos laterales.

El transelevador trilateral automático, guiado por dos carriles, sólo tiene cuatro puntos de apoyo sobre el suelo. No resulta necesario el uso de carriles en la parte superior, lo que facilita que pueda ser instalado en cualquier almacén donde habitualmente se utilicen carretillas o montacargas. Además, se adapta a casi todos los espacios de acceso directo a paletas. Su funcionamiento es totalmente automático y todos los movimientos del transelevador responden a órdenes de almacenaje y de extracción enviadas por un sistema de control que optimiza los movimientos.

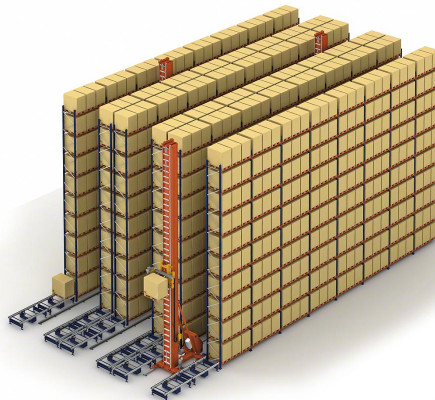


Figura 8: Bodega automatizada - Mecalux [6]

### 2.2.2. HUBMASTER® Warehouse Automation Systems

Es una compañía con más de 20 años de experiencia en el sector intralogístico, la cual ofrece soluciones de automatización para instalaciones industriales en todo el mundo. Su sede principal se encuentra en Barcelona, España, y cuenta con socios estratégicos en Europa, América y Asia.

Cuentan con distintas tecnologías patentadas las cuales se utilizan en una gran variedad de sectores, como lo son; almacenes con distintas especificaciones, distribución, automoción, logística, entre otros. Dentro de sus productos más relevantes se encuentra el sistema *HUBMASTER*®, el cual es un sistema de almacenamiento y recuperación de cargas automatizado que integra tecnología de transelevadores multi-pasillo y sistemas de gestión de almacenes. Su funcionamiento es parecido al de un sistema tradicional de almacenamiento y recuperación automática (ASRS), con la diferencia que este se encuentra configurado para realizar un cambio del transelevador rápido y fácil entre distintos pasillos.

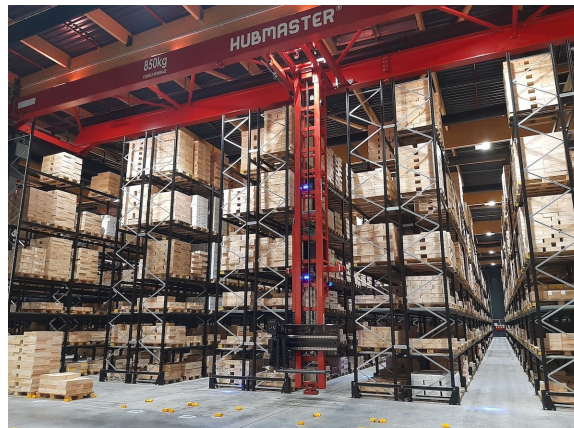


Figura 9: Sistema HUBMASTER® [7]

Este sistema permite manipular una amplia variedad de productos con distintos pesos y dimensiones. Se adecua dependiendo de la aplicación por medio de porta-horquillas adaptables, anchura variable, distintos rangos de movimiento y profundidad, así como otros accesorios que se pueden personalizar para cargas especiales. Dispone de un control volumétrico y de peso abordo del mismo transelevador, lo que permite recoger pallets directamente del suelo y llevarlos inmediatamente a la estantería sin pasar por punto de control. Por otro lado, también es resistente a condiciones extremas, tanto en congelado ( $-30^{\circ}\text{C}$ ) como a alta temperatura ( $45^{\circ}\text{C}$ ) y humedad, o en condiciones de almacenamiento químico (APQ).



Figura 10: Transelevador Trilateral - Hubmaster

7

Para controlar el sistema de gestión, se utiliza el software *HMWare™ WCS* (Por sus siglas en inglés: *Warehouse Control Software*), el cual se integra fácilmente con el software de gestión del cliente (WMS, ERP, MRP). Este dirige los controladores lógicos programables (*PLCs*) del sistema y proporciona funcionalidad para la gestión del almacén e integración bidireccional con el software de gestión del cliente en tiempo real.

El paquete básico proporciona un funcionamiento con entradas directas desde una consola de usuario táctil (HMI). Por lo tanto, ofrece soluciones para almacenes automáticos simples, como almacenes de herramientas que no están conectados directamente con un sistema de gestión de producción. El modo avanzado proporciona soporte para un funcionamiento autónomo las 24 horas del día, los 7 días de la semana, integrado junto con el software de gestión del cliente.

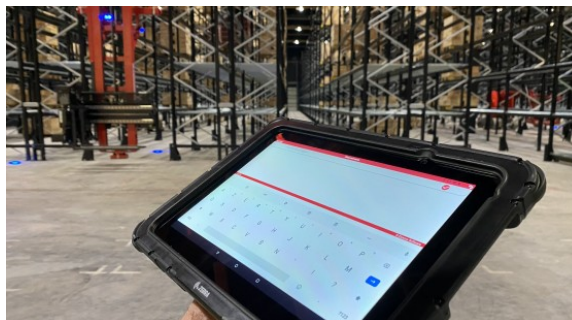


Figura 11: Consola de usuario táctil (HMI) - Hubmaster

7

En los procesos actuales de la industria, los almacenes y bodegas de almacenamiento desempeñan un papel fundamental como espacios decisivos de la gestión logística de las empresas. La eficiencia y optimización de procesos resultan ser pilares fundamentales para la competitividad y el éxito operativo de cualquier negocio. A lo largo de la historia, estos espacios se han considerado simplemente como sitios estáticos de almacenamiento, dándoles poca relevancia en términos de logística, costos y ventas. Por lo tanto, generalmente presentan sistemas de manipulación de cargas que, si bien fueron adecuados en su momento, en la actualidad han presentado limitaciones evidentes caracterizadas por demandas de rapidez, precisión y seguridad sin precedentes.

Dentro de este contexto, la automatización industrial se presenta como una solución estratégica con la capacidad de crear una transformación contundente en términos de la eficiencia y productividad de las operaciones logísticas. El proyecto de automatización de la bodega de Grupo Roca Fuerte [1] - [4] por medio de la implementación de transelevadores de carga, representa un acontecimiento significativo en este proceso progresivo. La adopción de esta tecnología avanzada no solo aborda los desafíos inherentes a los tiempos de entrega y despacho prolongados, sino que también responde de manera general a las demandas actuales de agilidad, precisión y seguridad en la gestión de inventarios y preparación de pedidos.

Una de las principales ventajas de la automatización de bodegas consiste en la optimización del tiempo de despacho. Los transelevadores, con su capacidad para almacenar y recuperar mercancía de manera automatizada y precisa, pueden llegar a reducir significativamente los tiempos de manipulación y desplazamiento, agilizando de esta forma el proceso de preparación y envío de pedidos. Esta mejora operativa resulta en una mayor satisfacción para el cliente final, ya que al minimizar los tiempos de espera y garantizar la puntualidad de las entregas, fortalece las relaciones proveedor-cliente y también la competitividad de la empresa en el mercado.

Además de los beneficios tangibles en términos de eficiencia, reducción del error por factor humano y productividad, la implementación de transelevadores implica una mejora significativa en la seguridad laboral. Esto debido a que se reduce la exposición de empleados

a situaciones de riesgo asociadas con el manejo manual de cargas pesadas y equipos de elevación. De esta forma, se puede promover un entorno de trabajo que sea más seguro y funcional, donde se minimicen los accidentes y se priorice la integridad física y mental del personal.

A través del re diseño y reacondicionamiento del proyecto desarrollado previamente para Progreso Labs [1] - [4], se busca poder mostrar a los estudiantes y a cualquier persona que visite el campus de la Universidad del Valle de Guatemala el potencial de este tipo de proyectos, junto con la importancia de los puntos mencionados previamente.

Este trabajo de graduación también tiene como objetivo poder reflejar la importancia de aplicar tecnologías emergentes en contextos demostrativos y educativos, promoviendo el aprendizaje activo, la simulación y análisis de posibles escenarios y el desarrollo de soluciones innovadoras para problemáticas reales del presente. El prototipo de un transelevador trilateral a escala, responde a una de las necesidades actuales de la logística moderna, demostrando la viabilidad de soluciones automatizadas para el manejo de almacenes y contribuyendo al desarrollo de habilidades técnicas y aplicadas en estudiantes.

### 4.1 Objetivo general

Restaurar la instalación eléctrica y el sistema de control del prototipo de transelevador trilateral para el acomodamiento de tarimas de carga en una bodega, ubicado en la Universidad del Valle de Guatemala.

### 4.2 Objetivos específicos

- Rediseñar el algoritmo de control para el prototipo de transelevador.
- Desarrollar una interfaz por medio de la cual el usuario pueda interactuar con los controles del prototipo de transelevador.
- Renovar y mejorar el sistema eléctrico, conexiones y cableado del prototipo de transelevador.
- Integrar el programa de control y la interfaz de usuario para la operación del prototipo de transelevador.

El presente trabajo de graduación aborda la restauración y renovación del prototipo de un transelevador trilateral a escala, ubicado en la Universidad del Valle de Guatemala. Esto con el propósito de convertirlo en un sistema plenamente funcional y demostrativo el cual permita simular procesos automatizados de almacenamiento y recolección de tarimas en una bodega. El trabajo se centra en la mejora de tres aspectos fundamentales del prototipo: el sistema de control, la interfaz gráfica de usuario y la instalación eléctrica, asegurando la integración eficiente y precisa de estos sistemas para garantizar un desempeño óptimo.

La renovación de la instalación eléctrica del prototipo abarca la revisión, reparación y reemplazo de cableado, conexiones y fuentes de alimentación, asegurando la estabilidad y seguridad de la operación del sistema. Las configuraciones de los motores paso a paso debían ser optimizadas ajustando los valores de corriente para garantizar movimientos precisos y prevenir daños en los componentes electrónicos.

De igual manera, la renovación pretende rediseñar el algoritmo de control del transelevador, implementando una lógica basada en máquinas de estados finitos. Este enfoque permitiría gestionar de manera eficiente los movimientos en los ejes X, Y y Z, así como el giro de  $180^\circ$  necesario para operar en dos pasillos paralelos de la bodega. También incorporaría funcionalidades avanzadas, como la detección de errores, la recalibración automática mediante finales de carrera y la capacidad de manejar comandos inválidos sin comprometer la estabilidad del sistema.

La renovación también abarcaría el desarrollo de una interfaz gráfica de usuario empleando Python, la cual sería desplegada en una pantalla táctil. Esta interfaz permitiría una interacción fácil e intuitiva con el prototipo, facilitando la selección de comandos y brindando información en tiempo real sobre el estado del sistema. De esta manera se aseguraba que cualquier persona, sin importar su rango de edad, pudiera interactuar y operar el transelevador.

Un protocolo de comunicación serial serviría para enviar y recibir comandos, asegurando la transferencia confiable de datos entre los sistemas mencionados anteriormente, junto con

confirmaciones de las operaciones realizadas. Esto garantizaría un sistema fiable, preciso y con una respuesta bastante rápida.

El trabajo de graduación también involucraría la validación integral del sistema mediante pruebas funcionales y ajustes iterativos. Esto permitiría demostrar que el prototipo cumplía con sus objetivos iniciales, teniendo mejoras significativas en términos de precisión, eficiencia y capacidad de respuesta. Su funcionalidad restaurada y mejorada lo posicionaría como un proyecto demostrativo el cual abarca conocimientos de distintos cursos y herramientas de la licenciatura en ingeniería mecatrónica de la Universidad del Valle de Guatemala. El prototipo aportaría valor educativo, técnico y demostrativo, como una herramienta de aprendizaje en automatización aplicada.

El prototipo del transelevador trilateral a escala, pretende responder a una de las necesidades actuales de la logística moderna, demostrando la viabilidad de soluciones automatizadas para el manejo de almacenes y contribuyendo al desarrollo de habilidades técnicas y aplicadas en estudiantes.

### 6.1. Transelevadores con microprocesadores

Un transelevador es una máquina creada para almacenar y extraer de manera automatizada mercancía de un almacén, la cual, principalmente, son tarimas, cajas, contenedores, entre otros. Estas máquinas incorporan un microprocesador, componente electrónico cuyo objetivo consiste en procesar y ejecutar las instrucciones provenientes de un programa de gestión de almacén (SGA) [8].

Los transelevadores recogen los productos desde el extremo de un pasillo y los trasladan e introducen en la ubicación asignada de una estantería. Por otro lado, estos también retiran la carga de la estantería y la conducen hasta el transportador de salida o la zona de carga que se sitúa al final del pasillo. Al conjunto de estas dos operaciones se le denomina en logística como *ciclo combinado*. Básicamente, un ciclo combinado es cuando se deposita una tarima y se retira otra aprovechando el mismo viaje del transelevador.

Estos equipos representan una evolución de los equipos de manutención tradicionales, como los montacargas, con los operarios a bordo dirigiendo y controlando sus movimientos y acciones. En cambio, los transelevadores con microprocesadores operan idealmente con total autonomía. Por lo tanto, se recomienda invertir e instalar estos equipos cuando se busca aumentar el rendimiento de un almacén, mejorar el nivel de servicio y reforzar la seguridad de la mercancía y de los operarios. Su uso se aconseja especialmente cuando se gestiona una gran variedad de mercancía de consumo medio o a gran escala y se desea lograr un número elevado de movimientos de manera más óptima y eficiente (ciclos/hora de entrada, salida o combinados).

Los transelevadores con microprocesador se desplazan sobre carriles lineales e incorporan un sistema de extracción o efector final el cual permite colocar las unidades de carga en las estanterías. Gracias a estos factores pueden instalarse en un pasillo con estanterías en ambos lados, en función de la capacidad de almacenaje que se necesite dentro del almacén.

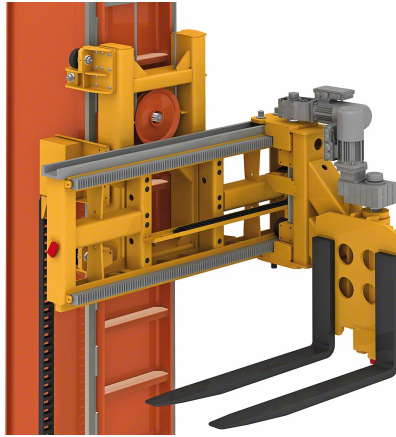


Figura 12: Efecto final de un transelevador con microprocesador - Mecalux 6

### 6.1.1. Modos de operación:

#### **Automático:**

Este es el modo de trabajo más habitual del transelevador, en el que se lleva a cabo la ubicación y extracción de la mercancía de manera automatizada. El transelevador se controla desde un PLC (*Programmable Logic Controller*, es decir, un controlador lógico programable de su traducción del inglés), un ordenador el cual generalmente se utiliza para controlar maquinaria industrial de una planta, realizar operaciones, entre otras acciones. El transelevador recibe las órdenes del sistema de gestión a través del microprocesador integrado programable. El software cuya misión es coordinar los diferentes PLC que operan en un almacén automático se conoce como sistema de control de almacén (SCA).

#### **Semiautomático:**

En este modo se dispone de un ordenador en uno de los extremos del pasillo el cual es utilizado por un operario para poder enviar instrucciones a los transelevadores. El operario introduce el destino, espacio o las coordenadas que necesita el transelevador para que este efectúe cualquier movimiento. Con este modo es posible implementar funciones de apoyo como acceder a ubicaciones específicas o re ubicar tarimas con mercancía.

#### **Manual:**

Este modo se utiliza principalmente para tareas de mantenimiento y reparación. Desde una cabina el operario controla la máquina mediante comandos y mandos manuales. Generalmente, se trabaja a una baja velocidad y con el equipo y las medidas de seguridad necesarias. Cabe destacar que los transelevadores poseen sistemas de seguridad como escaleras de mano, cables de seguridad o plataformas de mantenimiento para proteger a las personas que operan con ellos.

### **6.1.2. Ventajas**

Dentro de los beneficios principales que un transelevador con microprocesador puede ofrecer se encuentran:

#### **Aprovechamiento del espacio:**

Esto se debe a que permite tener una alta densidad de almacenaje, tanto en metros cuadrados como en altura. Otro factor que se debe tomar en cuenta es que los pasillos para operar dentro de los almacenes pueden llegar a ser muy estrechos cuando se busca maximizar los espacios dentro de una bodega. Esto puede traer retos y dificultades al momento de utilizar varios montacargas convencionales al mismo tiempo, mientras que un transelevador puede operar de manera continua sin problema.

#### **Alta accesibilidad de las cargas:**

Los transelevadores tienen un acceso rápido y directo a todos los productos almacenados. Las entradas y salidas de las tarimas se realizan en dos tipos de movimientos: simples, únicamente entradas o salidas de productos, o combinados, es decir, que introducen un producto en el almacén y aprovechan ese movimiento para extraer otro.

#### **Inventario permanente:**

Cuando se habla de un inventario permanente se hace referencia a que no existe la necesidad que una persona realice un inventario o conteo de lo que hay en el almacén, ya que en todo momento se tiene un inventario el cual se actualiza en tiempo real. En una instalación automática operada por un transelevador es indispensable disponer de un sistema de gestión de almacenes (SGA) que lleve un control preciso de las tarimas y la mercadería dentro del almacén. Con un inventario permanente se podrá saber en todo momento dónde se halla cada artículo. De esta manera se pueden llegar a eliminar casi en su totalidad los errores derivados de la gestión manual.

#### **Seguridad total de la carga:**

La manipulación de las cargas se realiza de manera completamente automatizada, sin la intervención de los operarios. Esto previene accidentes o equivocaciones, de manera que también se reducen costos como sueldos, capacitaciones, producto dañado, entre otros.

#### **Total disponibilidad y productividad:**

Estos sistemas automatizados pueden operar sin interrupción jornadas de 24 horas, de manera que lo convierte en una solución idónea para compañías con un alto número de

movimientos y con una preparación de pedidos masiva.

### **Eficiencia energética:**

No todos los transelevadores poseen esta característica, sin embargo, hay algunos transelevadores que incorporan generadores de energía que devuelven a la red eléctrica la energía generada durante las maniobras de descenso y deceleración del eje. Esto se traduce en un ahorro de entre un 15% y un 20% en promedio, sobre el consumo energético total de una máquina. Todo ello con el propósito de promover una logística verde que reduzca el impacto medioambiental que causan las actividades empresariales.

Cabe destacar que, para poder instalar un transelevador con microprocesador, se debe tomar en cuenta que las inversiones iniciales suelen ser bastante elevadas en comparación con los gastos de un almacén operado manualmente. Además, para poder sacar el máximo provecho a un transelevador este debería operar idealmente en un único pasillo el cual se encuentre equipado con otros sistemas de transporte auxiliar.

## **6.2. Sistema de Gestión de Almacenes (SGA)**

También conocido por sus siglas en inglés como WMS (*Warehouse Management System*) [9], es un software que tiene como objetivo principal ayudar a las empresas a gestionar y controlar las operaciones diarias de los almacenes. Este se utiliza desde el momento en que los bienes o materiales entran a un centro de distribución o almacén hasta el momento en que estos salen.

Un SGA ayuda a optimizar todas las fases en la gestión de almacenes, desde los procesos de recepción, almacenamiento, retiro, embalaje y envío, hasta el seguimiento y re abastecimiento del inventario. De igual manera, un buen SGA se caracteriza por poder dirigir y organizar todas actividades desde una única interfaz. Los SGA más robustos incluso logran integrar otras herramientas a este proceso, como podrían ser los Sistemas de Gestión de Transporte (TMS), etiquetado RFID, dispositivos de realidad aumentada (AR), entre otros.

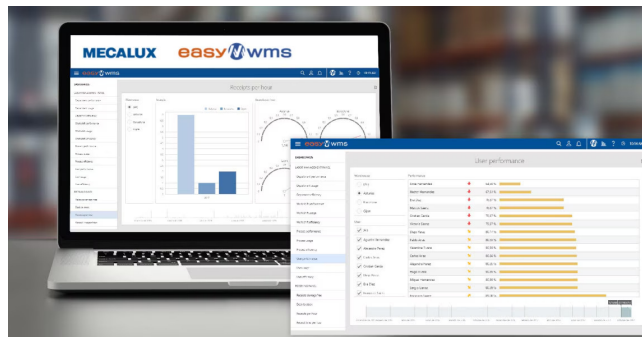


Figura 13: Sistema de gestión de almacenes Easy WMS (Mecalux) [10]

### **6.2.1. Capacidades funcionales**

#### **Recepción:**

En la mayoría de sistemas de gestión de almacenes se realiza una comparación entre las existencias que se tenía previsto recibir y lo que realmente entró al almacén. Otro aspecto que contemplan es en qué parte del proceso logístico de la cadena de suministro se encuentra el producto, ya sea que se encuentre en movimiento, ya llegó al destino, etc. Para lograrlo un operario confirma las recepciones de la mercadería o en algunos casos existen lectores o sensores que pueden hacerlo también. Pueden llegar a existir distintos tipos de recepciones las cuales se podrían clasificar: Devoluciones, trasposos, directas de proveedores, entre otras.

#### **Ubicación:**

Dentro de esta parte del proceso se ubican las existencias disponibles y las nuevas recepciones dentro del almacén en un espacio que se encuentre libre. Esta función puede desempeñarse con base en la toma de decisiones humanas, es decir por parte de los operarios o los coordinadores del almacén, o bien, en los SGA más sofisticados el software del fabricante es capaz de determinar la ubicación más óptima dentro del almacén. Para hacerlo se pueden establecer ciertos parámetros y restricciones en dicho software. En los SGA más avanzados incluso se utilizan herramientas combinadas, como los comandos por voz y la inteligencia artificial para poder agilizar la toma de decisiones y el proceso en general.

#### **Reposición y expedición:**

Consiste en realizar las acciones y operaciones necesarias para mover los productos desde el lugar donde se encuentran almacenados hasta el área de carga u otra zona. En algunos casos especiales también se reponen o recuperan productos para trasladarlos de una área del almacén a otra. Un ejemplo de esto podría ser el caso en que se tengan productos perecederos, indicaciones especiales, almacenes demasiado grandes, entre otros. Los productos perecederos deben de colocarse en espacios con un alto flujo de movimiento o donde el despacho sea lo más rápido posible. En el caso de algunos almacenes que son demasiado grandes, se debe de estar moviendo los productos más antiguos a ubicaciones más cercanas a las zonas de carga, de manera que su despacho sea mucho más rápido. De esta forma será posible seguir abasteciendo el almacén con productos nuevos.

Para ejecutar cualquiera de estas acciones se necesita siempre la autorización de un coordinador o encargado del almacén. Existen casos particulares en los que se desarrollan y programan rutinas de almacenamiento, reposición y expedición para ciertos productos en almacenes los cuales cuentan con procesos bastante optimizados o lineales. Otro caso en el que se aplican estas rutinas es en los que se cuenta con una planta de producción desde la cual sale un producto empaquetado, de manera que es posible almacenarlo directamente en una zona especificada.

Existe una gran cantidad de módulos complementarios y herramientas que pueden ser implementadas durante estos procesos. Dentro de los más comunes se encuentran módulos

facturación, de embarque y muelle, optimización estratégica de distribución de productos en almacenes (también conocido como *Slotting*, de su traducción al inglés), análisis de la cadena de suministro, de alertas para re abastecimiento, entre otros.

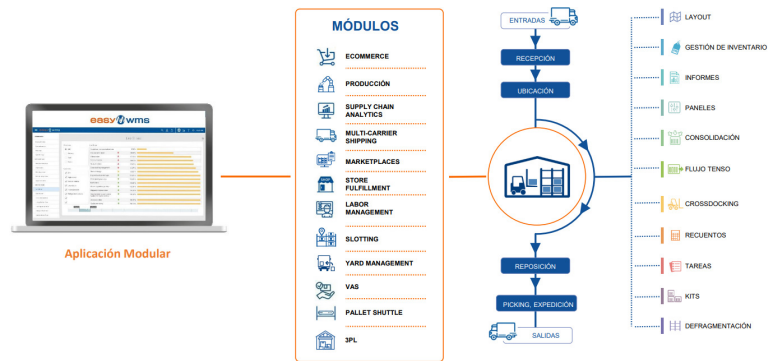


Figura 14: Diagrama modular y de procesos WMS (Mecalux) [10]

### 6.2.2. Tipos de infraestructura

Existen dos tipos de infraestructura dentro de los sistemas de gestión de almacenes:

**Modelo basado en internet y la nube:** Dentro de este apartado, el software que se adquiere se puede ir actualizando de manera remota, siempre y cuando se tenga una conexión segura a internet. Cabe destacar que con esta opción todo debería pasar por una conexión segura y sin tener que invertir en infraestructura informática para alojar el software. De igual manera, es posible vincular aplicaciones para dispositivos móviles y opciones para que un usuario acceda por medio de un navegador a la información de un almacén. En muchos de los casos, los usuarios deberán pagar algún tipo tarifa mensual aparte para utilizar estos servicios. Un ejemplo de este tipo de infraestructura es el sistema SAAS, de *Easy WMS* creado por la compañía Mecalux [10].



Figura 15: Easy WMS con conexión wifi (Mecalux) [10]

**Modelo de despliegue en la infraestructura del cliente:** En este apartado, tanto el software como el hardware necesarios para poder ejecutar el SGA deben de ser instalados y

configurados dentro del almacén donde se utilizará. La comunicación dentro de este modelo es directa y sin capas intermedias de por medio, lo que minimiza la pérdida de datos y aumenta la velocidad de la tasa de refresco de las bases de datos. Este modelo consigue aumentar la estabilidad y seguridad de la información ya que la mantiene a un nivel local.

### **6.2.3. Ventajas**

Dentro de los beneficios principales que un SGA puede ofrecer se encuentran:

#### **Eficiencia operativa mejorada:**

Al automatizar y optimizar los procesos dentro de un almacén, desde que un producto ingresa hasta que este sale, se obtienen procesos más rápidos, operaciones más fluidas y la capacidad de manejar mayores volúmenes. Los SGA mejoran tanto los tiempos de almacenamiento como los de entrega y disminuyen drásticamente los errores, casi eliminándolos por completo. Por otro lado, también disminuyen los costos logísticos, ya que optimiza el recurso humano y los costos de manipulación. Al automatizar el flujo de información y de procesos, se agiliza la toma de decisiones cruciales.

#### **Menos pérdidas:**

En el caso que se tenga dentro del inventario productos perecederos, restringidos por fechas de vencimiento o con características especiales a tomar en cuenta, los SGA pueden mandar advertencias o mensajes a los coordinadores de bodega o a los operarios para que estos lo tengan en cuenta. De esta manera se puede identificar qué productos necesitan un impulso de ventas o deben ser retirados primero. Los SGA más avanzados son capaces de determinar el uso más eficaz dentro de un almacén, de manera que pueden ubicar los productos dependiendo de estas consideraciones para que se encuentren más cerca o lejos del despacho o incluso desarrollar recorridos óptimos por medio de simulaciones avanzadas.

#### **Visibilidad del inventario en tiempo real:**

Es posible tener un control absoluto del inventario de un almacén por medio de la implementación de la codificación de barras, el etiquetado RFID, sensores y otros métodos de seguimiento de ubicación. Gracias a esta visibilidad se puede crear previsiones de demanda precisas, ejecutar estrategias de inventario y tomar decisiones fundamentadas en data real y no estimaciones. Mejorar la trazabilidad de los productos de una empresa puede llegar a tener un impacto significativamente grande en los procesos, costos y decisiones de la cadena de suministro.

## 6.3. Elementos para el desarrollo del prototipo de transelevador trilateral a escala

### 6.3.1. Motor paso a paso

Un motor paso a paso (Comúnmente llamados *Stepper Motors* por su traducción del inglés) es un motor de corriente continua (DC) sincrónico sin escobillas el cual funciona con una corriente pulsada, donde cada pulso hace girar el rotor una fracción de una rotación completa. Está conformado por un rotor central rodeado de un estator el cual contiene un número variable de bobinados según el modelo. Para inducir el movimiento, estos bobinados se energizan, convirtiéndose en electroimanes con una polaridad determinada por la dirección de la corriente [11].

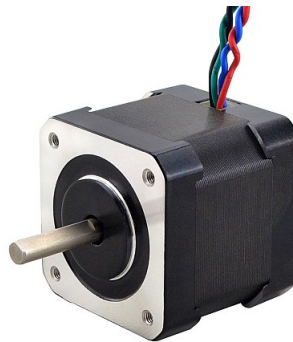


Figura 16: Motor paso a paso híbrido Nema17 [12]

Los motores paso a paso son robustos, económicos y adecuados para proyectos que requieren alta precisión o aplicaciones sin retroalimentación de bucle cerrado. Se utilizan generalmente para discos duros giratorios, impresoras, robótica, máquinas herramienta CNC, impresoras 3D, entre otras aplicaciones. Los tres tipos de motores paso a paso más comunes de encontrar son los siguientes:

- **Motor paso a paso de imán permanente:** Tienen rotores contruidos con imanes permanentes, que interactúan con los electroimanes del estator para crear rotación y torque. Los motores paso a paso de imán permanente suelen tener características de potencia comparativamente bajas con otros modelos y pueden producir más par por unidad de potencia de entrada.
- **Motor paso a paso de reluctancia variable:** Estos motores no están fabricados con imanes permanentes, sino con hierro y se parecen a un engranaje. Tienen protuberancias o dientes alrededor de la circunferencia del rotor. Los dientes dan lugar a motores paso a paso de reluctancia variable que tienen un grado muy alto de resolución angular. Sin embargo, esta alta precisión suele producirse a costo de tener un menor torque.
- **Motor paso a paso síncrono híbrido:** Utilizan las mejores características de los

motores paso a paso mencionados anteriormente. El rotor de este motor tiene un núcleo de imán permanente, mientras que la circunferencia está hecha de hierro y tiene dientes. Por lo tanto, un motor síncrono híbrido tiene una alta resolución angular y buen torque. Los motores Nema17 pertenecen a este apartado, siendo uno de los modelos más utilizados para proyectos de robótica y automatización, debido a su alta precisión, fiabilidad y buen torque.

## Modos de funcionamiento

Los motores paso a paso tienen tres modos de funcionamiento principales. Tomando en consideración un motor básico con un rotor de 200 dientes, los modos de funcionamiento pueden ser:

- **Modo de paso completo:** Por cada rotación de  $360^\circ$  del eje del motor, el rotor avanza a través de 200 pasos, cada uno de exactamente de  $1.8^\circ$ . Durante el funcionamiento de paso completo, dos de las fases del estator siempre están energizadas. Esto proporciona el par máximo, pero la resolución angular y la precisión se encontrarán limitadas por la cantidad de dientes del rotor.
- **Modo de medio paso:** Por cada rotación de  $360^\circ$  del eje del motor, el rotor avanza a través de 400 pasos, cada uno de exactamente  $0.9^\circ$ . Durante el funcionamiento de medio paso, se alterna entre tener una o dos fases activadas en el estator. Esto proporciona el doble de nivel de resolución angular, obteniendo una mayor precisión de posicionamiento, pero se produce menos torque.
- **Modo de micropasos:** Por cada rotación de  $360^\circ$  del eje del motor, el rotor avanza una cantidad delimitada de micro pasos. Las configuraciones más comunes de micro pasos tienen un factor de  $1/8$ ,  $1/16$  y  $1/32$  por cada revolución. Durante el funcionamiento en micropasos, las fases del estator pueden estar activadas, desactivadas o activadas parcialmente. Este modo se utiliza en aplicaciones donde se necesita un posicionamiento de alta precisión, aunque el par nominal podría llegar a reducirse hasta en un 30%. Dependiendo de la resolución de micro pasos que se seleccione, tanto el torque como la precisión de posicionamiento, pueden verse afectadas o favorecidas.

Al elegir un motor paso a paso y su modo de funcionamiento es importante tomar en cuenta las siguientes consideraciones: Cuánto torque o par requerirá la aplicación, qué tan importante es y qué papel juega la precisión de posicionamiento en la aplicación, a qué velocidad o ritmo de trabajo estará trabajando y las dimensiones o el espacio que se tiene disponible para su instalación [13].

## Micropasos

Los motores paso a paso funcionan mediante un pulso eléctrico que hace girar el motor un paso completo. Este pulso se introduce de forma continua, lo que hace que el motor siga girando en incrementos de pasos completos. Los micropasos se producen cuando el motor

gira menos de un paso completo. Estos pueden reducir significativamente el tamaño de un paso, lo más común es entre 16 y 64 micropasos por cada paso completo. Esto se logra introduciendo algoritmos personalizados para ajustar la fuerza del impulso eléctrico.

El valor de los micropasos no radica en una mayor precisión; por el contrario, estos pueden llegar incluso a reducirla en algunos escenarios. A medida que los micropasos se hacen más pequeños, la precisión puede comenzar a disminuir. Esto se debe a que los motores paso a paso tienen un error inherente debido a su construcción.

Otro aspecto a tomar en consideración al usar micropasos es que el par incremental disminuye a medida que aumenta el número de micropasos. Si hay par de carga, se producirá un contragolpe magnético que desplazará el rotor de su posición prevista hasta que se genere el par suficiente.

Dependiendo de la aplicación, la principal ventaja de los micropasos es que estos permiten un movimiento más suave y uniforme. Esto debido a que reducen el ruido mecánico, proporcionan un accionamiento más suave y minimizan los problemas de resonancia [14].

### 6.3.2. Arduino Mega 2560

El Arduino Mega 2560 es una placa de desarrollo basada en el microcontrolador ATmega2560 de Atmel. Es una placa que fue diseñada para proyectos complejos que requieren mayor capacidad de procesamiento, una memoria amplia y un mayor número de entradas y salidas digitales y analógicas. Es ampliamente utilizada en aplicaciones de robótica, automatización y en sistemas que requieren controlar múltiples motores y el manejo de datos, como impresoras 3D. Su diseño modular y flexibilidad lo convierten en una opción ideal tanto para principiantes como para desarrolladores experimentados.

Entre las principales especificaciones técnicas del Arduino Mega 2560 se destacan las siguientes: Utiliza un microcontrolador ATmega2560 y opera con un voltaje estándar de 5 voltios. Su voltaje de entrada recomendado varía entre 7 y 12 voltios, lo que le permite trabajar en una amplia gama de entornos energéticos. Dispone de 54 pines digitales de entrada/salida, de los cuales 15 pueden configurarse como salidas PWM, y 16 entradas analógicas, que permiten trabajar con múltiples sensores. Incorpora 4 puertos UART, lo que facilita la comunicación serial con varios dispositivos de manera simultánea. Su memoria incluye 256 KB de memoria Flash, de los cuales 8 KB están reservados para el bootloader, junto con 8 KB de SRAM y 4 KB de EEPROM, lo que le proporciona una gran capacidad para manejar programas complejos y almacenar datos persistentes. La placa opera a una frecuencia de reloj de 16 MHz, lo cual es estándar en la mayoría de los microcontroladores de la serie Arduino [15].

El Arduino Mega 2560 presenta ventajas significativas en comparación con otras placas, como el Arduino Uno. Su principal ventaja radica en el mayor número de pines de entrada y salida, lo que permite gestionar proyectos que involucren numerosos sensores, actuadores y dispositivos. Por otro lado, la memoria ampliada, tanto en términos de Flash como de SRAM y EEPROM, ofrece la posibilidad de implementar programas más grandes y complejos, mientras que los 4 puertos UART lo convierten en una solución ideal para proyectos que requieren comunicación simultánea con múltiples dispositivos. Es compatible con la mayoría

de los shields diseñados para el Arduino Uno y también los de otros fabricantes, lo que facilita la integración de hardware adicional sin problemas de compatibilidad.

En términos de aplicaciones, el Arduino Mega 2560 se utiliza frecuentemente en proyectos avanzados que requieren robustez y capacidad. Es ideal para controlar impresoras 3D, gracias a su capacidad para manejar múltiples motores paso a paso y sensores. También se emplea en robótica, especialmente en diseños que necesitan controlar numerosos actuadores y procesar señales de múltiples sensores. Además, su versatilidad lo convierte en una excelente herramienta para sistemas de automatización industrial y doméstica.

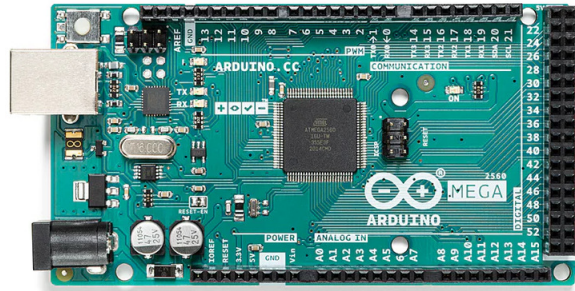


Figura 17: Arduino Mega 2560 15

### 6.3.3. RepRap Arduino Mega Pololu Shield (Ramps 1.4)

El RepRap Arduino Mega Pololu Shield (RAMPS 1.4) es una placa de expansión diseñada para funcionar junto con la placa Arduino Mega 2560, creada específicamente para el control de impresoras 3D y máquinas CNC. Este módulo forma parte de los componentes fundamentales en el ecosistema de proyectos DIY ("Do it yourself" por sus siglas en inglés o "Hazlo tú mismo." en español) relacionados con la impresión 3D y la fabricación digital. Su propósito principal es proporcionar una interfaz eficiente y modular que permita la conexión y gestión de diversos dispositivos esenciales, como motores paso a paso, sensores de temperatura, calentadores, ventiladores y otros elementos típicos de una impresora 3D o sistemas similares.

El diseño de la RAMPS 1.4 destaca por su modularidad y flexibilidad, ya que integra de manera eficiente toda la electrónica necesaria para el control de una impresora 3D en una sola placa. Esto simplifica el proceso de ensamblaje y reduce significativamente el espacio requerido. La placa está equipada para soportar hasta cinco controladores de motores paso a paso, como el A4988 o el DRV8825, lo que la hace ideal para gestionar el movimiento en tres ejes (X, Y, Z) y hasta dos extrusores en sistemas de impresión 3D más avanzados.

Entre sus especificaciones técnicas más relevantes, la RAMPS 1.4 incluye tres salidas de potencia controladas por MOSFETs, destinadas al manejo de calentadores o ventiladores, y tres circuitos dedicados para la lectura de termistores que monitorizan las temperaturas de la cama caliente y los extrusores. Adicionalmente, incorpora fusibles de seguridad: uno de 5 amperios para proteger los componentes electrónicos y otro de 11 amperios destinado exclusivamente a la cama caliente. También dispone de pines de expansión I2C y SPI, que

permiten integrar módulos adicionales como pantallas LCD o lectores de tarjetas SD para ampliar la funcionalidad del sistema. Los indicadores LED en la placa facilitan la supervisión visual de los estados de los calentadores y otros elementos conectados.

Las ventajas de esta placa son múltiples. Por un lado, su diseño modular permite reemplazar y actualizar componentes de manera sencilla, lo que resulta especialmente útil en proyectos que evolucionan con el tiempo. Por otro lado, su compatibilidad con diversos controladores de motores paso a paso y otros accesorios asegura su adaptabilidad a diferentes aplicaciones, desde impresoras 3D básicas hasta máquinas CNC más avanzadas. El módulo RAMPS 1.4 se beneficia de una amplia documentación y soporte de la comunidad entusiasta de los proyectos mencionados anteriormente, lo que facilita su implementación y resolución de problemas durante el desarrollo de proyectos.

En términos de aplicaciones, la RAMPS 1.4 es ampliamente utilizada en sistemas de impresión 3D basados en la plataforma RepRap [16], un movimiento que promueve la fabricación de impresoras 3D construidas por los usuarios y que son de bajo costo. Su diseño eficiente y económico ha hecho que sea una opción popular tanto para aficionados como para desarrolladores experimentados que buscan soluciones personalizables y confiables.

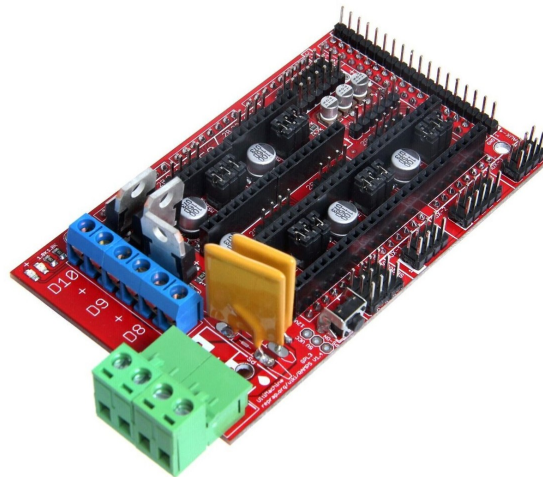


Figura 18: RepRap Arduino Mega Pololu Shield (Ramps 1.4) [16]

#### 6.3.4. DRV8825

El DRV8825 es un controlador de motores paso a paso desarrollado por Texas Instruments [17], diseñado específicamente para gestionar motores bipolares en aplicaciones que requieren movimientos precisos, controlados y eficientes. Este dispositivo es ampliamente utilizado en proyectos de impresión 3D, máquinas CNC y otros sistemas de automatización que demandan un control avanzado de motores. Su funcionalidad incluye la capacidad de utilizar micropasos, regulación de corriente y diversas protecciones integradas, lo que lo convierte en una herramienta versátil para desarrolladores y entusiastas de este tipo de

proyectos.

El DRV8825 opera en un rango de voltaje de 8.2 a 45 voltios, lo que le permite ser utilizado en una amplia variedad de aplicaciones con motores que diferentes características. Su capacidad para manejar una corriente máxima de salida de hasta 2.5 amperios por bobina lo hace apto para motores de mayor potencia. Una de sus características más destacadas es su compatibilidad con resoluciones de micropasos que llegan hasta 1/32 de paso completo, lo que permite movimientos más suaves y precisos, fundamentales en sistemas que demandan alta exactitud y vibraciones mínimas.

Este controlador utiliza una interfaz de control sencilla basada en las señales *STEP* y *DIR*, lo que facilita su integración en sistemas que utilizan microcontroladores como Arduino o plataformas de desarrollo similares. Además, incluye una variedad de protecciones integradas, como mecanismos contra sobrecorriente, cortocircuitos, subtensión y sobrecalentamiento, lo que aumenta la fiabilidad y seguridad del sistema en el que se emplea. También incorpora un modo de reposo de baja corriente, que contribuye a reducir el consumo energético cuando el dispositivo no está en uso.

El DRV8825 presenta ventajas significativas en comparación con otros controladores de motores, como el popular A4988. Su mayor rango de voltaje de operación permite trabajar con motores más grandes y potentes, mientras que su capacidad de micropasos más precisa (1/32 frente a 1/16 en el A4988) ofrece una resolución superior para aplicaciones que requieren movimientos altamente controlados. Además, su capacidad de manejar mayores corrientes de salida lo hace más adecuado para aplicaciones que exigen motores de mayor rendimiento.

En términos de aplicaciones, el DRV8825 es ideal para sistemas que involucran movimientos controlados y repetitivos, como impresoras 3D, fresadoras CNC y robots industriales. Su capacidad de utilizar micropasos y su diseño robusto permiten controlar motores con un alto grado de precisión, facilitando la implementación de sistemas complejos y confiables.

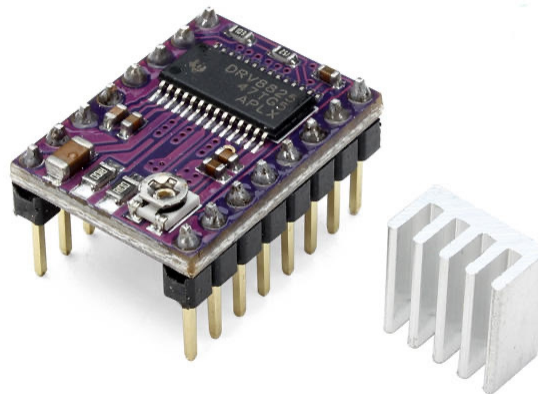


Figura 19: Driver DRV8825 17

### 6.3.5. Finales de carrera

Un final de carrera, también conocido como interruptor de límite, es un dispositivo electromecánico diseñado para detectar la posición o el movimiento de un objeto dentro de un sistema. Su funcionamiento se basa en la interacción física entre el objeto y un actuador que, al ser accionado, altera el estado de los contactos eléctricos internos. Esto permite abrir o cerrar un circuito, enviando señales que pueden ser utilizadas para controlar máquinas, detener movimientos, o iniciar procesos en sistemas automatizados. Los finales de carrera se encuentran comúnmente en aplicaciones industriales y de automatización, donde se requiere precisión y seguridad en la operación de maquinaria.

Estos dispositivos tienen un papel crucial en sistemas de control. Por ejemplo, en una cinta transportadora, un final de carrera puede detectar el final del recorrido de un producto y detener el movimiento para evitar daños. En máquinas herramientas, se usan para delimitar el rango de movimiento, asegurando que no se excedan los límites físicos predefinidos. Además, se emplean como mecanismos de seguridad, actuando como interruptores de emergencia en sistemas críticos.

Desde el punto de vista técnico, los finales de carrera presentan una variedad de especificaciones adaptadas a diferentes necesidades. Los actuadores pueden variar en diseño e incluir configuraciones como palancas, rodillos o émbolos, dependiendo del tipo de movimiento que se desea monitorear. Los contactos eléctricos pueden ser normalmente abiertos (NO), normalmente cerrados (NC) o una combinación de ambos, lo que define cómo responde el circuito en estado de reposo o activación. Estos dispositivos son capaces de manejar corrientes y voltajes específicos, generalmente indicados en sus características técnicas, para garantizar un funcionamiento seguro y eficiente. Además, están diseñados para soportar ciclos de operación mecánica y eléctrica, indicando su durabilidad, y cuentan con especificaciones ambientales como rangos de temperatura y resistencia a factores externos como humedad o polvo.

Las ventajas que ofrecen los finales de carrera son múltiples. Su fiabilidad y precisión los convierten en herramientas esenciales para la detección exacta de posiciones y límites de movimiento. Son dispositivos robustos, capaces de operar en entornos industriales exigentes, como aquellos con altas vibraciones, temperaturas extremas o partículas en suspensión. Su instalación es sencilla y requieren un mantenimiento mínimo, lo que los hace prácticos en sistemas donde el tiempo de inactividad debe minimizarse. Además, su versatilidad los hace adecuados para una amplia gama de aplicaciones, desde sistemas industriales avanzados hasta equipos domésticos automatizados [18].

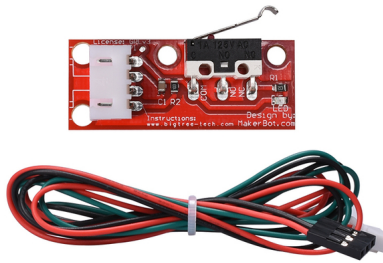


Figura 20: Final de carrera para impresora 3D [18](#)

### 6.3.6. Raspberry Pi 3

La Raspberry Pi 3 es una microcomputadora de placa única desarrollada por la Fundación Raspberry Pi, diseñada para proporcionar acceso económico a la computación y fomentar la enseñanza de la informática. Su versatilidad y potencia la han convertido en una herramienta fundamental en el ámbito educativo, el desarrollo tecnológico y los proyectos de innovación en diversas áreas, como la automatización, la robótica y el Internet de las Cosas. Dentro de la serie Raspberry Pi 3 se incluyen los modelos Model B, Model B+ y Model A+, cada uno de ellos adaptado a necesidades específicas.

La Raspberry Pi 3 Model B está equipada con un procesador Broadcom BCM2837, de cuatro núcleos ARM Cortex-A53 a 1.2 GHz, y una memoria RAM de 1 GB LPDDR2. Su conectividad incluye Wi-Fi 802.11n, Bluetooth 4.1 y Ethernet 10/100, mientras que sus interfaces físicas constan de cuatro puertos USB 2.0, un puerto HDMI, una salida de audio de 3.5 mm, y pines GPIO para conexiones personalizadas. Por su parte, el modelo B+ mejora el rendimiento al incorporar un procesador Broadcom BCM2837B0 a 1.4 GHz y Wi-Fi de doble banda (2.4 GHz y 5 GHz) con soporte para Bluetooth 4.2, además de incluir Ethernet Gigabit (limitado a 300 Mbps) y conservar los mismos 1 GB de RAM. Finalmente, el modelo A+, más compacto y económico, mantiene el procesador del modelo B+ pero con una memoria RAM reducida de 512 MB y un único puerto USB 2.0 [19](#).

Esta serie de dispositivos tiene aplicaciones diversas que abarcan desde la educación hasta la industria. En el ámbito educativo, las Raspberry Pi 3 son ampliamente utilizadas como herramientas para enseñar conceptos básicos y avanzados de programación, permitiendo a estudiantes y docentes experimentar con sistemas reales. En el internet de las cosas, su conectividad inalámbrica integrada facilita el desarrollo de proyectos como estaciones meteorológicas, sistemas de monitoreo remoto o dispositivos inteligentes. Además, son empleadas en la creación de centros multimedia para el hogar, sistemas de automatización doméstica y controladores para robots en el campo de la mecatrónica.

Entre las principales ventajas de la Raspberry Pi 3 destacan su accesibilidad económica, que la hace ideal para estudiantes y desarrolladores con presupuestos limitados, y su versatilidad, derivada de su diseño modular y sus interfaces de expansión. La eficiencia energética

es otra de sus fortalezas, lo que la convierte en una opción adecuada para proyectos de largo plazo o aplicaciones que requieren bajo consumo. Asimismo, la Raspberry Pi 3 cuenta con el respaldo de una activa comunidad global de usuarios, que aporta documentación, recursos y soporte técnico, facilitando su adopción en nuevos proyectos.

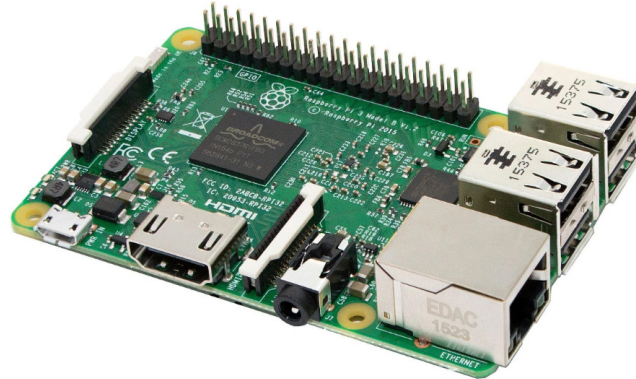


Figura 21: Raspberry Pi 3 19

---

## Desarrollo del algoritmo de control del transelevador

---

Este capítulo se enfoca en el desarrollo y programación del algoritmo de control del prototipo de transelevador trilateral, el cual se basó en las necesidades de un Sistema de Gestión de Almacenes (*WMS*). El objetivo principal de esta etapa del proyecto consistió en diseñar y ejecutar un algoritmo que permitiera el control preciso del transelevador en tres ejes (X, Y, Z) y la rotación de  $180^\circ$  del efector final para operar en dos pasillos paralelos. La implementación se realizó utilizando la plataforma *Arduino Mega 2560* y un módulo *Ramps 1.4*, integrando motores paso a paso, finales de carrera y otros componentes esenciales para garantizar movimientos eficientes y seguros en un entorno controlado.

Para alcanzar este objetivo, se establecieron diferentes pasos en el desarrollo del algoritmo. Primero, se seleccionaron los componentes y se realizaron pruebas iniciales para validar su funcionamiento. Luego, se procedió a diseñar un sistema de control basado en una máquina de estados finita (*FSM*) que gestionaba los comandos del usuario y luego realizaba los movimientos. Por último, se agregó la re-calibración automática del transelevador mediante la implementación de finales de carrera. Este capítulo detalla cada una de estas fases, así como las iteraciones que fueron necesarias para optimizar el sistema y garantizar que los movimientos del transelevador cumplieran con las especificaciones técnicas de un sistema de gestión de almacenes.

A continuación, se presenta una cronología general del proceso llevado a cabo en este capítulo:

- **Selección de componentes:** Se escogieron los motores, controladores, y demás hardware basados en criterios como compatibilidad, precisión y robustez.
- **Configuración inicial y pruebas básicas:** Se validaron las conexiones de los motores y la respuesta del sistema sin finales de carrera.
- **Desarrollo del algoritmo de control:** Se implementó la máquina de estados finitos

(*FSM*) y las funciones básicas para controlar el transelevador.

- **Integración de finales de carrera:** Se añaden estos componentes al diseño del prototipo para asegurar re-calibración, aumentar la redundancia y evitar posibles errores.
- **Optimización y corrección de errores:** Se ajustaron los tiempos de respuesta, movimientos y manejo de errores. También se reforzó el sistema para manejar errores y comandos inválidos sin afectar su funcionamiento.
- **Resultados:** Se valida el funcionamiento de la primera fase del proyecto, exponiendo los resultados obtenidos.

Este enfoque permitió cumplir con los objetivos preestablecidos para esta fase del proyecto, asegurando un sistema robusto y eficiente que respondió adecuadamente a las interacciones del usuario.

## 7.1. Selección de materiales y componentes

### 7.1.1. Arduino Mega 2560

La elección de la placa *Arduino Mega 2560* se fundamenta en la necesidad de un microcontrolador con mayor capacidad de procesamiento y almacenamiento en comparación con la placa *Arduino Uno*. *Arduino Mega 2560* cuenta con un microcontrolador *ATmega2560* que ofrece un total de 54 pines digitales de entrada/salida, 16 pines analógicos y 256 KB de memoria flash para almacenar el código, lo cual resulta ser un factor crítico para proyectos de mayor demanda y precisión como el transelevador trilateral.

Otro de los factores decisivos en la elección de la placa *Arduino Mega 2560* fue la cantidad de pines disponibles, ya que el proyecto requería controlar 4 motores *NEMA17* y 6 finales de carrera, únicamente para el apartado del algoritmo de control. Además, su compatibilidad con el entorno de desarrollo *Arduino IDE* facilitó la integración del módulo *RAMPS 1.4* y los *drivers DRV8825*. Su facilidad de uso y la extensa documentación disponible en la comunidad de desarrolladores también fueron factores clave e influyeron en la decisión.

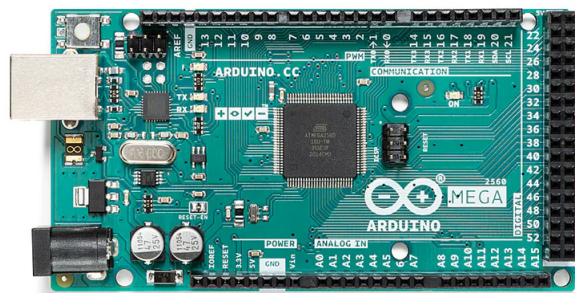


Figura 22: Arduino Mega 2560 15

### 7.1.2. RepRap Arduino Mega Pololu Shield (Ramps 1.4)

El módulo RAMPS 1.4 (*RepRap Arduino Mega Pololu Shield*) fue seleccionado debido a su capacidad para controlar múltiples motores paso a paso de manera eficiente y por su compatibilidad directa con la placa *Arduino Mega 2560*. Este módulo se utiliza comúnmente en impresoras 3D y máquinas *CNC* (Control numérico por ordenador, por sus siglas en inglés), lo que lo hace ideal para proyectos que requieren un control preciso de movimientos en varios ejes.

El RAMPS 1.4 es un módulo diseñado específicamente para ser montado sobre la placa *Arduino Mega 2560*, facilitando la conexión de hasta cinco controladores de motores paso a paso, como los *DRV8825*, así como finales de carrera, ventiladores y otros periféricos. En este proyecto, se utilizaron los conectores de motores para controlar los ejes X, Y y Z (avance de la cremallera), y un cuarto conector para controlar el motor que gira las horquillas 180 grados. Este *shield* permitió gestionar fácilmente las conexiones necesarias para operar los motores y sensores, asegurando un funcionamiento óptimo del transelevador.

Otro factor el cual contribuyó a la selección de este módulo fue su facilidad de instalación y la extensa documentación disponible en la comunidad de desarrolladores. Al utilizar este módulo también se abordó uno de los objetivos específicos del proyecto, el cual consistía en renovar y mejorar el sistema eléctrico. Esto debido a que las conexiones y el cableado de los motores y de los finales de carrera sería directo hacia el módulo, sin tener la necesidad de utilizar una placa de pruebas o placas de inserción.

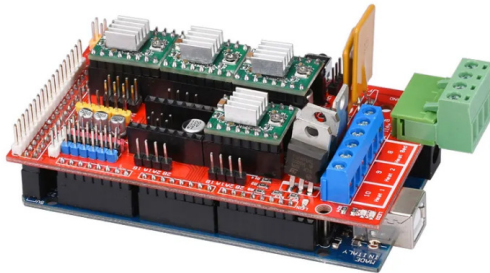


Figura 23: Montaje del módulo Ramps con la placa Arduino Mega 2560

### 7.1.3. Driver DRV8825

El *Driver DRV8825* es un controlador de motores paso a paso conocido por su alta precisión y capacidad para manejar corrientes de hasta 2.2 amperios por fase. Estas especificaciones hicieron que fuera una opción adecuada para controlar los motores *NEMA17* de este proyecto. Cada controlador incluye un potenciómetro ajustable el cual permite calibrar la corriente máxima suministrada a un motor. Esto resulta ser fundamental para evitar el sobrecalentamiento y garantizar un funcionamiento eficiente.

Otro factor que contribuyó a la elección de utilizar estos controladores, fue que estos admiten distintos modos de micro pasos. Esto permite configuraciones que van desde un

paso completo (1/1) hasta 1/32 de micro paso, lo que garantizaría movimientos más suaves y precisos en los ejes X, Y y Z del transelevador. De lo contrario, se podrían haber experimentado problemas como los que se tuvo durante las primeras pruebas de los motores, en donde el uso de pasos demasiado grandes o toscos hacía que los motores vibraran en exceso.

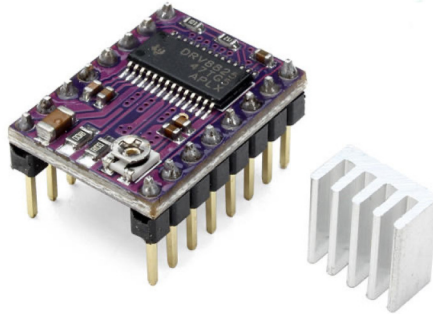


Figura 24: Driver DRV8825 17

#### 7.1.4. Motores NEMA17

Estos motores fueron seleccionados debido a sus dimensiones, buen torque y precisión, los cuales se adecuaban a las necesidades del proyecto. Las especificaciones de los *NEMA17* de 48 milímetros de altura incluyen un torque de retención de hasta 55 Newton-centímetro y una corriente nominal de hasta 2 amperios por fase. Esto los hace ideales para aplicaciones como el transelevador, en donde se requería el torque necesario para mover un sistema de transmisión de potencia por medio de poleas y otro con un tornillo sin fin. Por otro lado, cada motor tiene una resolución de 200 pasos por revolución, lo que proporciona una alta precisión en los movimientos.

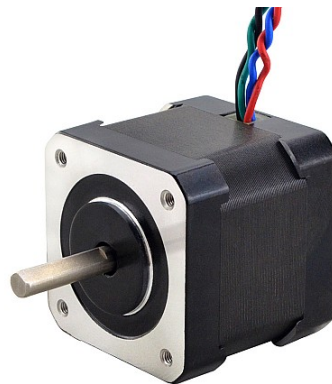


Figura 25: Motor Nema17 12

Los motores de 23 milímetros de altura fueron implementados en las partes del proyecto donde no se requería tanto torque como el de las fases que se mencionaron anteriormente. En su lugar, se necesitaba de un motor con menores dimensiones para que no fuera tan pesado y que ocupara menos espacio en el efector final del transelevador. Cuentan con un torque de 17 Newton-centímetro y una corriente nominal de 1 amperio por fase. Cada uno de estos cuenta con una resolución de 200 pasos por revolución, lo que aporta una alta exactitud en los movimientos.

El proyecto ocupó un total de cuatro motores *NEMA17*, cada uno asignado a una función específica:

- **Motor del eje X:** Responsable del movimiento lateral del transelevador.
- **Motor del eje Y:** Responsable del desplazamiento vertical del transelevador.
- **Motor del eje Z:** Controla el avance y retroceso de la cremallera para manipular las tarimas con las horquillas.
- **Motor de rotación (Eje A):** Realizaba el giro de 180 grados que permite al transelevador operar en pasillos paralelos.

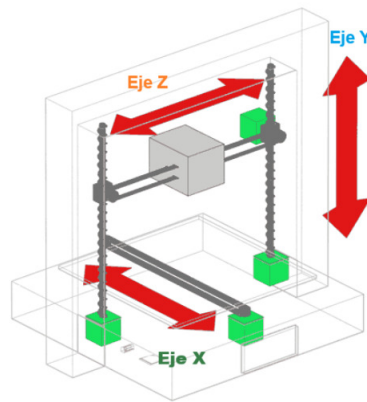


Figura 26: Diagrama de movimientos

### 7.1.5. Finales de carrera

Los finales de carrera fueron componentes fundamentales para garantizar que el sistema siempre retornara a su posición de origen de manera precisa. Fueron implementados en la etapa final de esta fase del proyecto, con el objetivo de tener una mayor redundancia. De esta forma, el funcionamiento del transelevador no dependería exclusivamente de la ejecución del algoritmo de control. Las señales enviadas por la activación de los finales de carrera permitieron validar el correcto funcionamiento del transelevador y también calibrar los motores con cada ciclo cumplido.

En total se instalaron seis finales de carrera, cada uno conectado a un pin específico en el *RAMPS 1.4*. Al tratarse de un módulo desarrollado para la implementación de impresoras

3D y máquinas *CNC*, este contaba con un apartado dedicado únicamente a los finales de carrera, por lo que la conexión fue bastante sencilla e intuitiva. De este modo también se cumpliría con los objetivos específicos del proyecto, mejorando el cableado y facilitando la conexión de los componentes. Esto también resulta bastante útil en caso de que se tenga que renovar o cambiar uno de los componentes.

A continuación, se detalla la asignación de los finales de carrera:

- **Posición inicial en X** (Ramps 1.4 = X Min): Define el origen (*Home*) para el motor del eje X.
- **Posición inicial en Y** (Ramps 1.4 = X Max): Utilizado para establecer el cero del eje Y (Altura 0).
- **Límite de avance en Z** (Ramps 1.4 = Y Min): Controla el límite de avance de la cremallera (motor del eje Z).
- **Límite de retroceso en Z** (Ramps 1.4 = Y Max): Controla el límite de retroceso de la cremallera (motor del eje Z).
- **Posición del efector final sin giro** (Ramps 1.4 = Z Min): Define el ángulo de 0 grados para el motor de rotación (Motor del eje A).
- **Posición del efector final a 180°** (Ramps 1.4 = Z Max): Define el ángulo de 180 grados para el motor de rotación (Motor del eje A).

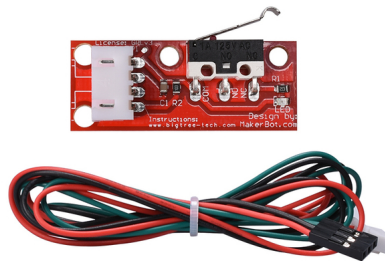


Figura 27: Final de carrera 18

### 7.1.6. Fuente de poder

Para suministrar energía a los motores, placas y drivers, se utilizó una fuente de poder de 12 voltios y 10 amperios. Esta fuente se conecta directamente a un toma corriente y entrega el voltaje y amperaje deseado, siendo el máximo el descrito anteriormente. Para configurar el voltaje se debía de ajustar un potenciómetro el cual tenía integrado la fuente, con un destornillador pequeño.

La elección de la fuente de 12 voltios se fundamentó en el hecho que cumplía con todos los requisitos establecidos para el funcionamiento óptimo del proyecto. Asimismo, se tenía buenas referencias y ejemplos de esta fuente, debido a que es estándar para proyectos similares a este, como podrían ser impresoras 3D y máquinas *CNC*. Tanto el módulo *RAMPS 1.4* como los controladores *DRV8825* y los motores *NEMA17* operan de manera eficiente con estas especificaciones. Los 10 amperios que entrega la fuente también proporcionan un margen seguro para evitar sobrecargas, tomando en consideración que los motores necesitaban aproximadamente 1 - 2 amperios cada uno dependiendo de su carga.

Esta fuente no solo permite un funcionamiento estable, sino que también ayuda a mantener una buena disipación de calor. Esto es importante para evitar el sobrecalentamiento de los componentes electrónicos, especialmente en proyectos de larga duración o con accionamiento continuo, como este.



Figura 28: Fuente de poder 12V - 10A

## 7.2. Calibración de los drivers DRV8825

La calibración correcta de los *Drivers DRV8825* fue clave para asegurar que los motores *NEMA17* recibieran la cantidad adecuada de corriente. De esta manera se previnieron problemas como sobrecalentamiento o un mal funcionamiento de los motores. Este proceso de ajuste se llevó a cabo utilizando la fórmula que se detalla a continuación para calcular el voltaje de referencia ( $V_{REF}$ ) que requería cada motor.

Fórmula para calcular el voltaje de referencia según el desarrollador ([20] - [17]):

$$V_{REF} = \frac{I_{max}}{2} \quad (1)$$

Donde:

- $V_{REF}$ : Es el voltaje de referencia.
- $I_{max}$ : Es la corriente máxima que el motor puede manejar de manera segura para un funcionamiento óptimo.

Como se mencionó anteriormente, el proyecto cuenta con dos tipo de motores *NEMA17*. Para el motor de 42x42x48 mm se utilizó 1.9 amperios como el valor máximo de corriente que se le podía suministrar al motor. Al utilizar la ecuación 1 se obtuvo como resultado que el voltaje de referencia debía ser de 0.95 voltios. Se utilizó un margen de seguridad del 5% por debajo de la corriente de trabajo máxima del motor con la finalidad de evitar cualquier tipo de sobrecalentamiento o mal funcionamiento.

$$V_{REF} = \frac{1.9A}{2} = 0.95V \quad (2)$$

De igual manera, para los motores de 42x42x23 mm se utilizó un margen de seguridad del 5% por debajo de la corriente de trabajo máxima de los motores. Luego de utilizar la ecuación 1 se obtuvo como resultado que el voltaje de referencia debía ser de 475 mili voltios. De esta forma se garantizó un rendimiento más óptimo, evitando el sobrecalentamiento de los motores, con un margen de trabajo más conservador.

$$V_{REF} = \frac{0.95A}{2} = 0.475V \quad (3)$$

El proceso de ajuste del voltaje de referencia requirió el siguiente equipo: Un destornillador pequeño de precisión para ajustar el potenciómetro en el *driver DRV8825*, un multímetro para medir el valor del voltaje y unos cables banana-lagarto para conectar los terminales del multímetro y realizar las mediciones.

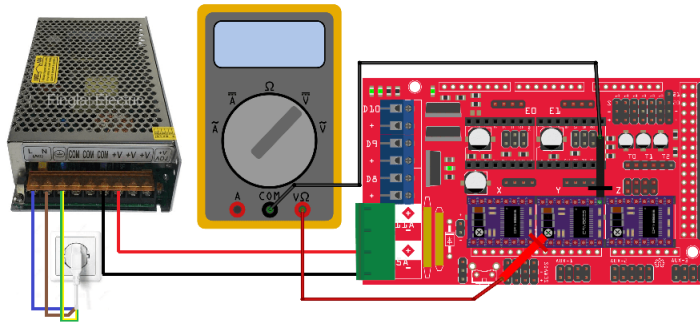


Figura 29: Materiales para la calibración de los drivers DRV8825

El procedimiento consistió en ajustar el potenciómetro que se encontraba en el *driver*, mientras se medía el valor de  $V_{REF}$ . Para realizarlo, se conectó la terminal positiva del multímetro al destornillador y la terminal negativa a la tierra común, es decir, la de la fuente de poder. Se realizaron giros y ajustes muy pequeños en el potenciómetro hasta alcanzar los valores deseados. Estos valores fueron los que se calcularon anteriormente en las ecuaciones 2 - 3.

Esta calibración es crucial porque afecta directamente a la capacidad de los motores de manejar las cargas asignadas y de realizar los movimientos requeridos de manera precisa. Durante las pruebas iniciales los motores vibraban y se calentaban mucho. Es por eso que luego de la calibración se verificó que no hubiera sobrecalentamiento y que los motores funcionaran de manera suave en los ejes X, Y, Z y en el giro de 180 grados.

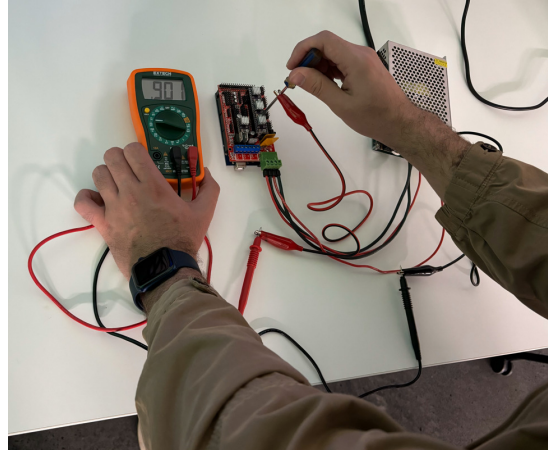


Figura 30: Calibración de los drivers DRV8825

Asimismo, todos los *drivers DRV8825* utilizados en el proyecto fueron configurados con un micro paso de 1/16. Se decidió utilizar este micro paso, debido a que es una configuración intermedia, la cual permite un movimiento más suave y uniforme, sin la necesidad de sacrificar tanto torque y precisión. De esta forma, también se reduciría el ruido mecánico, proporcionando un accionamiento más suave y disminuiría los problemas de resonancia.

Para modificar la resolución de los micro pasos de un *driver DRV8825* se necesita establecer una configuración específica de las entradas digitales  $M0$ ,  $M1$  y  $M2$ . Los estados de estas entradas pueden ser alto (*High* - 1) o bajo (*Low* - 0). A continuación se muestra una tabla con todas las posibles configuraciones de resolución de micro pasos para un *driver DRV8825*:

DRV8825 Microstepping			
M0	M1	M2	Step
Low	Low	Low	Full
High	Low	Low	Half
Low	High	Low	1/4
High	High	Low	1/8
Low	Low	High	1/16
High	Low	High	1/32
Low	High	High	1/32
High	High	High	1/32

Cuadro 1: Posibles combinaciones de micro pasos - Driver DRV8825 [20]

Al utilizar un módulo *Ramps 1.4* es posible modificar la resolución de los micro pasos del *driver DRV8825* por medio de conectores de dos pines. Bajo el espacio donde se acoplan los *drivers DRV8825*, en la *Ramps 1.4*, se encuentran un total de 6 pines para poder modificar la configuración. Para establecer la configuración de 1/16 de paso, se colocó un conector en los pines destinados al modo 2 ( $M2$ ), en todos los *drivers*.

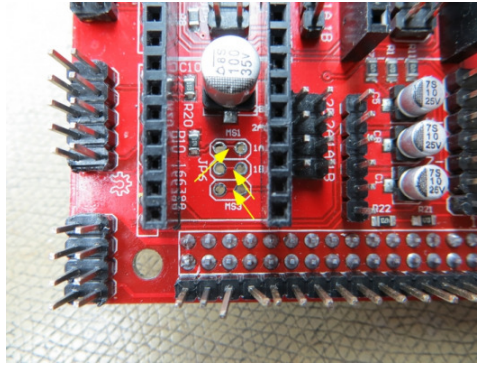


Figura 31: Espacios para conectores en el módulo Ramps 1.4

### 7.3. Configuración inicial y pruebas básicas

Las primeras pruebas se centraron en validar que cada motor *NEMA17*, controlado por los *drivers DRV8825*, funcionara adecuadamente. Para verificar el funcionamiento correcto de los componentes y las conexiones eléctricas se implementó un código básico que hacía girar los motores en ambos sentidos por medio de un ciclo de giros predefinidos. Los motores del eje X, Y, Z (cremallera) y el motor de rotación de 180 grados se movieron secuencialmente hasta completar tres revoluciones en sentido de las agujas del reloj y luego en sentido contrario. De esta manera fue posible garantizar que los motores respondieran correctamente a las señales enviadas desde el *Arduino Mega 2560* a través del módulo *RAMPS 1.4*.

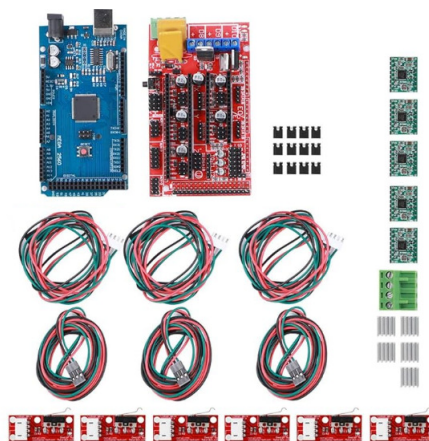


Figura 32: Componentes utilizados en las pruebas

Una vez re-calibrados los *drivers*, se llevaron a cabo pruebas individuales para cada uno de los ejes. Para el motor del eje X, el cual controlaba el movimiento lateral del transelevador, se ejecutaron movimientos hacia adelante y atrás con distintas distancias. Se programaron desplazamientos pequeños variando hasta llegar al recorrido completo a lo largo del pasillo. El código para estas pruebas incluía un ciclo de pasos predefinidos para garantizar que los motores siguieran las instrucciones de manera continua.

El motor del eje Y, que se encargaba del desplazamiento vertical del transelevador, también fue probado en múltiples escenarios. Se comprobó que el transelevador subiera y bajara en diferentes posiciones, las cuales correspondían a una simulación de las filas del almacén. De igual manera, el código utilizado durante estas pruebas era una secuencia de pasos que se ejecutaba en orden hasta completar un ciclo.

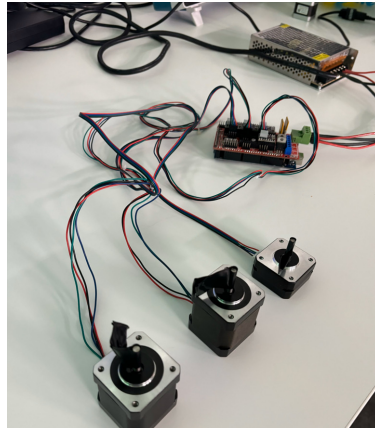


Figura 33: Conexiones realizadas para las pruebas iniciales

Los motores de los ejes Z y A, responsables del avance de la cremallera y de realizar el giro de 180° del efector final, también fueron sometidos a pruebas simples. Las pruebas del eje Z se centraron en asegurar que las horquillas pudieran avanzar y retroceder sin obstrucciones, moviéndose hasta que alcanzaran el número de revoluciones programado. Las pruebas del eje A, consistían en hacer girar las horquillas un total de 180° grados, esperar un par de segundos y luego regresar a la posición del ángulo de 0°.

Durante todas estas pruebas se verificó que el sistema pudiera realizar movimientos suaves y de manera continua. También se buscaba que los motores vibraran lo menos posible, lo que indicaría que la calibración y los micropasos configurados en los *drivers DRV8825* estaban ajustados de manera óptima. Las conexiones y los componentes del prototipo de transelevador trilateral quedarían como se muestra en la figura a continuación:

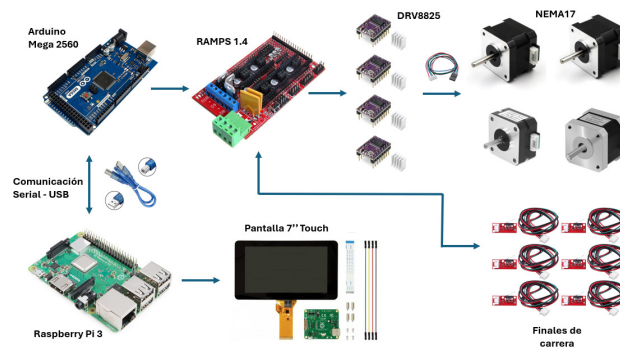


Figura 34: Diagrama de conexiones del prototipo de transelevador trilateral

---

## Algoritmo de control

---

El desarrollo del algoritmo de control del prototipo de transelevador trilateral fue un proceso que involucró iteraciones constantes, pruebas exhaustivas y ajustes continuos. La complejidad del sistema requirió llevar a cabo pruebas continuas para verificar que cada componente operara correctamente y que el flujo de trabajo cumpliera con las especificaciones planteadas desde el inicio del proyecto. Todo esto con el objetivo de corregir problemas y mejorar la eficiencia del sistema a medida que avanzaba el desarrollo.

La decisión de implementar una máquina de estados finitos (*FSM*) fue determinante para gestionar las operaciones del transelevador de manera secuencial y eficiente. A través de la *FSM*, se establecieron diferentes estados que representaban cada fase del proceso de carga y descarga de tarimas en un almacén, desde la recepción de comandos hasta la ejecución de movimientos y la reiteración de ciclos operativos.

Cada estado dentro de la máquina de estados finitos fue programado para cumplir con una tarea específica. De esta manera el sistema sería capaz de gestionar todas las operaciones del transelevador sin encontrarse con interferencias o errores de lógica. El estado se modificaría dependiendo de las entradas o condiciones que se hayan cumplido o no. A continuación, se detallan los estados implementados y un diagrama de flujo que muestra la lógica del algoritmo.



Figura 35: Diagrama de flujo del algoritmo de control

## 8.1. Máquina de estado finito (*FSM*)

### 8.1.1. Esperando comando (*WAITING\_FOR\_COMMAND*)

Este es el estado inicial del sistema, donde el transelevador se mantiene a la espera de un comando válido a través del puerto serial. El formato del comando consta de tres partes:

El primer carácter es una letra del alfabeto, la cual puede ir desde la *A* hasta la *O*. Dicha letra representa la columna en la que debe posicionarse el transelevador dentro del almacén. En la maqueta del prototipo de transelevador hay un total de 15 columnas, 10 de un lado del pasillo y 5 del otro.

El segundo carácter es un número que va del 1 al 4. Este representa la fila del almacén. En la maqueta del prototipo de transelevador hay un total de cuatro posibles alturas, es decir cuatro filas.

El tercer carácter únicamente puede ser la letra *P* o *L*. Este indica la operación que el transelevador debe de realizar. La letra *P* establece que la operación debe de ser recoger (*PICK*) la carga en el espacio indicado, mientras que la letra *L* determina que la operación debe ser dejar (*PLACE*) la carga en el espacio designado.

Cuando se recibe información en el puerto serial, el sistema se asegura de que el comando sea válido antes de pasar al siguiente estado. Estos son ejemplos de comandos que sí serían válidos: *D3L*, *R2P*, *A1L*, *H4O*. En el caso que se reciba un comando inválido, el sistema entra en el estado *ERROR*. Cuando el comando enviado es correcto, el sistema cambia al estado *MOVE\_TO\_POSITION*, en donde se ejecutarán los movimientos solicitados.

### 8.1.2. Mover a posición (*MOVE\_TO\_POSITION*)

En este estado, el transelevador se desplaza a la columna y fila especificadas por el comando. El movimiento se realiza en tres pasos secuenciales:

- **Movimiento en el eje X:** En este paso el transelevador se desplaza lateralmente hasta la columna que se indicó. Los pasos necesarios para accionar el motor y llegar a la columna se calculan utilizando arreglos predefinidos. Estos se encuentran formados por valores con un número de pasos específico para cada columna.
- **Movimiento en el eje Y:** Cuando el transelevador terminó con el paso anterior y ya se encuentra en la columna indicada, el motor del eje Y posiciona al transelevador en la fila que le corresponde de acuerdo al comando recibido. Del mismo modo, los pasos necesarios para que el transelevador llegue a la fila especificada se calculan por medio de una serie de valores predefinidos dentro de un arreglo para las distintas alturas.
- **Rotación de 180 grados:** Si la columna recibida en el comando se encuentra en el rango de *P* a *T* (es decir, en el lado opuesto del pasillo), el sistema rota las horquillas 180 grados antes de ejecutar los movimientos mencionados anteriormente (Movimiento en los ejes X & Y). Esto permite que el transelevador opere en ambos lados del pasillo.

Después de completar todos los movimientos, el sistema cambia al estado *PICK* o *PLACE*, dependiendo según lo indicado en el comando.

### 8.1.3. Recoger (*PICK*)

Este estado se activa cuando el comando recibido contiene la letra *P*, es decir, recoger una tarima. El proceso sigue los siguientes pasos:

El motor del eje Z desplaza las horquillas hacia adelante hasta que alcanzan la tarima. Una vez alcanzada, el motor del eje Y realiza un ligero movimiento hacia arriba para asegurar la tarima. Luego las horquillas retroceden en el eje Z hasta llegar a la posición inicial. Tras completar todas las acciones, el sistema pasa al estado *RETURN\_HOME* para continuar con el proceso.

### 8.1.4. Dejar (*PLACE*)

Este estado se activa cuando el comando recibido tiene la letra *L*, es decir, dejar una tarima. Este comando, ejecuta una acción antes de pasar al estado *MOVE\_TO\_POSITION*. Primero, el motor del eje Z hace avanzar las horquillas hasta alcanzar la tarima que se quiere ir a dejar. Luego el motor del eje Y levanta la tarima. Después el transelevador regresa en el eje Z hasta la posición original, pasando al estado *MOVE\_TO\_POSITION* de manera normal.

El proceso y la lógica después de completar estas acciones preliminares es similar al que se lleva a cabo en el estado *PICK* pero con leves diferencias. En lugar de levantar la tarima, las horquillas la dejan en la ubicación seleccionada de esta forma:

El motor del eje Z hace avanzar a las horquillas hacia adelante hasta alcanzar la distancia de profundidad estándar. Luego el motor del eje Y mueve las horquillas ligeramente hacia abajo para dejar la tarima en su lugar. Por último, las horquillas retroceden en el eje Z hasta alcanzar la posición inicial. Tras completar el comando operación, el sistema pasa al estado *RETURN\_HOME* para continuar con el proceso.

#### 8.1.5. Regresar a la posición de origen (*RETURN\_HOME*)

Este estado es fundamental para volver a calibrar el sistema al finalizar cada ciclo y poder ejecutar otro comando después. El transelevador debe regresar a su posición de origen (*HOME*), para comenzar otro proceso. Para conseguirlo, este estado sigue los siguientes pasos:

Primero el motor del eje Z retrocede las horquillas hasta alcanzar su posición original. Después el motor de rotación ajusta las horquillas para que se sitúen en 0 grados si es necesario. Luego el motor del eje Y baja el transelevador hasta la altura inicial (Fila 0). Por último, el transelevador regresa lateralmente hasta la posición inicial (*HOME*) por medio del accionamiento en reversa del motor del eje X. Una vez completado el retorno a la posición inicial, el sistema vuelve a regresar al estado *WAITING\_FOR\_COMMAND*. De esta manera el sistema se vuelve a calibrar y queda listo para recibir un comando nuevo.

#### 8.1.6. Acción luego de recoger (*POST\_PICK\_ACTION*)

Este estado fue añadido después de varias pruebas, con la finalidad de manipular la tarima que el transelevador fue a recoger luego de ejecutar los comandos *PICK* y *RETURN\_HOME*. Esta etapa no se ejecuta en el caso de que la operación especificada en el comando sea dejar (*L - PLACE*). Su propósito principal es permitir un tiempo de espera y asegurar que la carga recogida se mantenga en una posición segura y estable antes de proceder con nuevos comandos o acciones adicionales. Este tiempo de espera fue programado para ofrecer un margen en el que el usuario o cualquier operario pueda retirar la tarima de la zona de operación del transelevador de manera segura.

#### 8.1.7. Error (*ERROR*)

Este estado se activa cuando se recibe un comando inválido o se detecta una irregularidad en el funcionamiento del transelevador. El sistema detiene todas las operaciones y deshabilita los motores para evitar cualquier tipo de daño o desajuste.

En este estado se activan las siguientes funciones:

- **handleError():** Esta función es responsable de deshabilitar el funcionamiento del transelevador y llamar a las funciones responsables del re-lanzamiento del sistema.
- **resetSystem():** Restablece las variables del sistema y lo prepara para recibir nuevos comandos.

- **disableMotors():** Desactiva los motores, garantizando que no se produzcan movimientos indeseados mientras se resuelve el error.
- **enableMotors():** Reactiva los motores una vez que se ha manejado y corregido el error.

Tras solucionar el error, el sistema retorna al estado *WAITING\_FOR\_COMMAND*. Esto permite que el transelevador continúe operando sin interrupciones.

## 8.2. Integración de finales de carrera

La siguiente fase en el desarrollo del algoritmo de control fue la integración de los finales de carrera (*endstops*). Esta incorporación resultó de la necesidad de tener una mayor redundancia y la re-calibración automática del sistema luego de realizar varias pruebas exhaustivas. Los finales de carrera actuarían como interruptores físicos los cuales detectarían la posición inicial y límite de cada eje. Esto permitiría al sistema retornar siempre a una posición de origen conocida (*HOME*) al finalizar cada ciclo de operación.

La implementación de los finales de carrera permitió que el transelevador se pudiera recalibrar de manera automática al inicio de cada ciclo. En la función *RETURN\_HOME*, el sistema utiliza los finales de carrera para mover cada eje hasta su posición inicial. Comienza por el eje Z (retroceso de las horquillas), después el eje de rotación para asegurar el ángulo correcto, luego baja por completo en el eje Y y por último regresa a la posición de origen en el eje X.

Esta misma rutina, se ejecuta cuando se energiza por primera vez el sistema. Cuando las posiciones de origen son establecidas, es decir que se cumplió con la rutina de los finales de carrera descrita previamente, el sistema entra en el bucle de la máquina de estados finitos. De esta manera se asegura un funcionamiento óptimo y calibración de las posiciones iniciales desde el inicio, antes de ejecutar cualquier operación.

Los finales de carrera no solo recalibran el sistema, sino que también proporcionan redundancia y seguridad operacional. En situaciones en las que el transelevador pueda llegar a perder el control o no complete una operación debido a un error, los finales de carrera permitían detener el avance de los motores de forma segura. Esto previene daños a los componentes, una operación incorrecta del sistema y provee retroalimentación continua al algoritmo de programación sobre lo que está sucediendo en el entorno físico.

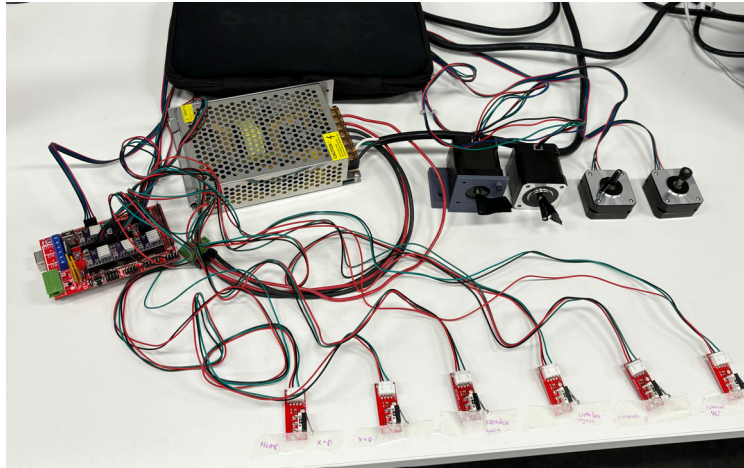


Figura 36: Pruebas realizadas con los finales de carrera antes de ensamblar el prototipo

### 8.3. Optimización y corrección de errores

A lo largo de las diferentes etapas del proyecto, se lograron identificar varios problemas que afectaban el funcionamiento, rendimiento y la precisión del sistema. Algunos de los problemas más destacados incluyeron la falta de sincronización entre los comandos recibidos y los movimientos de los motores, así como errores que provocaban que el sistema dejara de responder después de recibir un comando que era inválido.

Durante las primeras pruebas se detectó que en algunas situaciones el transelevador no completaba los movimientos en el orden esperado. Por ejemplo, después de recibir un comando válido, el sistema a veces no activaba los motores en la secuencia correcta. Este problema se abordó mejorando el manejo de los estados en la máquina de estados finitos (*FSM*). Esto se logró por medio de la implementación de banderas las cuales activaban distintas funciones dependiendo del estado del transelevador. De esta manera se aseguró que cada estado fuera ejecutado en el orden correcto y sin interferencias de por medio.

Otro problema recurrente fue la gestión de comandos inválidos. Al inicio, cuando se enviaba un comando con un formato incorrecto, el sistema entraba en un estado erróneo y dejaba de funcionar. Esto requería de un reinicio manual y volver a inicializar los motores. Este problema fue abordado implementando un sistema de reinicio automático que se activaba cada vez que se detectaba un comando inválido. Se implementó la función *handleError()* que se encargaría de detener todos los motores y reiniciar el sistema, permitiendo que el transelevador estuviera listo para recibir nuevos comandos sin comprometer toda la operación [35](#).

Para hacer el sistema aún más robusto, se implementaron estrategias de programación defensiva. Esto permitió abordar varios errores sin que el sistema se des-calibrara o dejara de funcionar. Las funciones dentro del código realizaban validaciones en cada fase del ciclo operativo. Estas iban verificando que los comandos recibidos fueran correctos, que los motores se activaran en el momento adecuado y que el sistema pudiera recuperarse rápidamente después de cualquier fallo.

Uno de los mayores aciertos durante esta etapa del proyecto fue la implementación de finales de carrera. Esto surgió a partir de la necesidad de minimizar daños a los componentes y proveer retroalimentación continua al algoritmo de programación sobre lo que estaba sucediendo en el entorno físico. De esta forma, se logró re-calibrar el sistema cada vez que terminara un ciclo operativo y cuando se inicializara el transelevador.

## 8.4. Resultados

El desarrollo del algoritmo de control resultó en una implementación funcional y robusta que cumplió con los objetivos planteados al inicio del proyecto. Al concluir con la fase de desarrollo, se logró poner en marcha un prototipo del transelevador trilateral que ejecutaba los movimientos requeridos en tres ejes distintos y era capaz de realizar rotaciones de 180 grados de manera eficiente. Este sistema también integraba mecanismos de seguridad y redundancia los cuales se implementaron para garantizar un funcionamiento confiable en entornos interactivos.

La integración de los finales de carrera proporcionó mayor seguridad y confiabilidad al sistema. Uno de los principales logros alcanzados en esta etapa del proyecto fue la implementación de la re-calibración automática al finalizar cada ciclo operativo y la implementación de programación defensiva. Esto permitió manejar errores y comandos inválidos de manera eficiente, logrando que el sistema fuera capaz de reiniciar sus operaciones tras la detección de cualquier comando erróneo, sin comprometer el funcionamiento general del transelevador. Este resultado se logró alcanzar gracias a la lógica de manejo de errores que se implementó y a las funciones que se fueron agregando a lo largo de las distintas pruebas que se hicieron.

El buen desempeño de esta fase proporcionó una base sólida para la siguiente etapa del proyecto, la cual se enfocó en la implementación de la interfaz gráfica de usuario y la integración del sistema completo.

---

## Desarrollo de la interfaz gráfica de usuario

---

Esta fase del proyecto tuvo como objetivo principal desarrollar una interfaz gráfica de usuario para el prototipo de transelevador trilateral, con el propósito de permitir una interacción amigable, intuitiva y eficiente con el sistema de control descrito en el capítulo anterior. Este sistema cuenta con movimientos en distintos ejes, rotaciones y una amplia serie de comandos que se pueden ejecutar, por lo que requería una plataforma gráfica que facilitara al usuario la operación de estas funciones de manera clara y simple. La interfaz no solo debía cumplir con estos requisitos, sino que también debía ofrecer una experiencia que resultara accesible para usuarios con distintos niveles de familiaridad con tecnologías similares.

Debido a que el transelevador se diseñó con el objetivo de ser un proyecto de demostración para distintas audiencias, fue necesario que la interfaz gráfica permitiera una interacción accesible a cualquier persona. Los usuarios interesados en el proyecto pueden ir desde expertos o personas interesadas en temas de electrónica, mecánica y otros campos, hasta usuarios casuales que quieran probar e interactuar con el proyecto. La accesibilidad y simplicidad en el diseño fueron factores decisivos en la elección de los componentes visuales. Esto aseguró que tanto niños, adolescentes, adultos e incluso personas mayores, pudieran utilizar el sistema sin dificultades. Una breve introducción al tema es suficiente para que los usuarios sepan cómo realizar las operaciones de dejar o recoger tarimas en el almacén simulado.

Para garantizar la fiabilidad del sistema, se implementó programación defensiva que protegiera al sistema de posibles errores humanos o interacciones no deseadas. Esta programación incluyó la validación constante de los comandos introducidos y la notificación de errores cuando se intentaba realizar operaciones inválidas. Estas medidas de seguridad fueron fundamentales para asegurar el buen funcionamiento del transelevador y poder proporcionar una experiencia de usuario sin complicaciones.

La interfaz gráfica fue diseñada para ejecutarse en una pantalla táctil de 7 pulgadas

conectada a una *Raspberry Pi 3*, la cual sirve como la plataforma principal de interacción. Se configuró para mostrar la interfaz en pantalla completa y desde allí poder enviar los comandos necesarios al sistema de control desarrollado en el [Capítulo 7](#). La elección de esta plataforma se fundamentó en la accesibilidad a ella, su gran versatilidad y la compatibilidad que esta tenía con la programación en entornos de *Python*.

Este capítulo abordará cada una de las etapas que conformaron el desarrollo de esta interfaz gráfica. Comenzando con la configuración de la *Raspberry Pi 3*, pasando por el diseño y programación de los elementos gráficos, hasta llegar a la validación final de la interfaz en la pantalla táctil. Del mismo modo, se abordarán las optimizaciones implementadas y los retos superados durante el proceso que garantizaron una experiencia fluida y funcional, adaptada a las necesidades de los usuarios y del sistema de control.

## 9.1. Configuración Raspberry Pi 3

El desarrollo de la interfaz gráfica para el prototipo de transelevador trilateral comenzó con la configuración de la *Raspberry Pi 3*. Esta plataforma fue seleccionada por su disponibilidad en la Universidad del Valle de Guatemala, la capacidad para ejecutar entornos gráficos, su compatibilidad con múltiples interfaces de hardware y su facilidad de programación utilizando *Python*. La opción de poder conectar una pantalla táctil al proyecto, sin la necesidad de tener que conectar más periféricos o de tener que correr comandos en una consola para hacer funcionar el transelevador también fue un factor clave.

### 9.1.1. Instalación y configuración del sistema operativo

El primer paso consistió en instalar el sistema operativo recomendado para este tipo de proyectos: *Raspberry Pi OS* (antes conocido como *Raspbian*). Esta es una distribución de Linux diseñada específicamente para la *Raspberry Pi*. Se instaló la versión con interfaz gráfica debido a la necesidad poder ejecutar un escritorio o interfaz visual que fuera accesible para cualquier usuario, sin la necesidad de tener conocimientos de programación. El sistema operativo fue descargado desde el sitio oficial de la *Raspberry Pi* [\[21\]](#), y posteriormente se grabó en una tarjeta SD utilizando el software *Raspberry Pi Imager* [\[22\]](#). Este instalador fue fundamental para facilitar la instalación del sistema operativo en la *Raspberry Pi* de manera eficiente y sin errores.

Después de instalar la imagen de *Raspberry Pi OS*, se realizó la primera configuración inicial de la *Raspberry Pi*. Durante este proceso se establecieron las configuraciones básicas como la selección del idioma, la zona horaria y la configuración del teclado. Para establecer las configuraciones iniciales se utilizaron dos puertos periféricos de la placa para conectar un teclado y ratón.

### 9.1.2. Configuración de pantalla táctil

Luego de configurar la placa se comenzó con la configuración de la pantalla táctil de 7 pulgadas de la *Raspberry Pi*. Esta pantalla proporcionaba el tamaño adecuado para realizar la interfaz gráfica, ofreciendo una resolución de 800x480 píxeles. Estas especificaciones eran ideales para poder visualizar todos los elementos gráficos de manera clara y sin distorsión. Del mismo modo, al ser una pantalla táctil, esto permitía una interacción directa del usuario con el sistema. Este factor eliminaba la necesidad de tener que utilizar periféricos adicionales como teclados o ratones.

Para garantizar el funcionamiento correcto del sistema, se conectó la pantalla de acuerdo a las instrucciones del fabricante. También se habilitó el soporte de pantalla táctil en las configuraciones de la *Raspberry Pi*. Luego de realizar la conexión exitosa de la pantalla, se realizó una actualización completa del sistema operativo. Esto para asegurar la compatibilidad con los controladores más recientes de la pantalla. Por último, se establecieron las dimensiones de trabajo correctas para la pantalla, ajustando los bordes que fueran necesarios. Estos pasos fueron importantes para mejorar la experiencia del usuario y también asegurarse que la interfaz fuera visualmente atractiva y funcional.

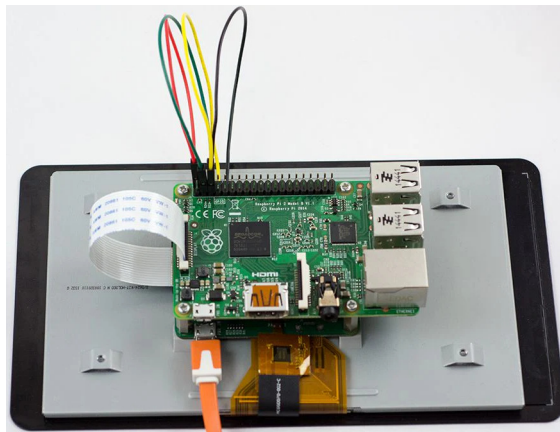


Figura 37: Conexiones y configuración de la pantalla táctil con la Raspberry Pi 3

### 9.1.3. Acceso remoto y control

Después de terminar con la configuración inicial de la *Raspberry Pi*, se optó por implementar un acceso remoto que facilitara el desarrollo y la optimización de la interfaz gráfica desde un ordenador principal. Esta configuración permitió realizar cambios y transferencias de archivos sin necesidad de conectar periféricos directamente a la Raspberry Pi, lo que resultó ser conveniente para esta etapa del proyecto.

Se utilizó *VNC Viewer* para habilitar el acceso remoto a la *Raspberry Pi* desde el ordenador principal. Esta es una herramienta ampliamente utilizada en la comunidad de desarrolladores la cual permite visualizar y controlar el entorno gráfico de la *Raspberry Pi* en tiempo real [23]. Esta herramienta fue fundamental durante la fase de programación de la interfaz gráfica. Para habilitar esta opción, se activó el servidor *VNC* en la *Raspberry Pi* por

medio del comando de configuración *raspi-config*. Esto permitió conectarse a la *Raspberry Pi* desde cualquier red local.

La configuración del acceso remoto permitió aumentar la productividad en la etapa de desarrollo de la interfaz. Esto debido a que permitió realizar cambios de manera rápida y habilitó el intercambio de archivos entre el ordenados y la placa. También ofreció la posibilidad de realizar pruebas en tiempo real sin tener la necesidad de estar físicamente presente frente a la pantalla táctil. Esto aceleró el proceso de desarrollo y solución de problemas con mayor eficiencia.



Figura 38: Acceso remoto a Raspberry Pi 3 por medio de RVNC

#### 9.1.4. Optimización del entorno de trabajo

La configuración de arranque directo de la interfaz gráfica fue uno de los ajustes que permitió mejorar varios aspectos del proyecto. Este ajuste hace que al encender la placa, la interfaz gráfica se inicie automáticamente sin la necesidad de ejecutar comandos adicionales desde la terminal. Para lograrlo, se modificó el archivo *autostart* en el directorio correspondiente. De este modo, la aplicación se ejecutaría en pantalla completa siempre que se inicie el sistema. Este enfoque mejoró tanto la productividad en el desarrollo de la interfaz, como la experiencia de usuario. También garantizó una transición rápida y fluida desde el inicio del sistema hasta la ejecución de la interfaz.

Por último, se desactivaron ciertos elementos que podrían interferir con la experiencia del usuario o con el desarrollo de la interfaz. Las notificaciones del sistema y el protector de pantalla fueron inhabilitados en el apartado de configuraciones. De esta forma, se evitó que la pantalla se apagara automáticamente y que aparecieran notificaciones indeseadas mientras el sistema se encontraba operando.

## 9.2. Implementación de la interfaz gráfica de usuario

El desarrollo de la interfaz gráfica del transelevador trilateral fue una de las etapas más importantes y desafiantes del proyecto. La interfaz debía de ser funcional, pero al mismo tiempo intuitiva y amigable para cualquier usuario. Al ser un proyecto de demostración destinado a la interacción con niños, adolescentes y adultos, fue fundamental crear un entorno que pudiera ser utilizado de manera sencilla por personas con poco contexto o sin conocimientos previos del proyecto.

### 9.2.1. Diseño

La primer etapa en el desarrollo de la interfaz gráfica consistió en crear un esbozo inicial para poder visualizar cómo sería la disposición de los elementos en la pantalla. Primero se definió la estructura básica de la interfaz, la cual tendría un menú principal y dos sub-menús: el menú de operación y el de inventario. Luego se enumeraron todas las acciones y comandos que la interfaz debía de poder hacer. Por último, se creó un bosquejo preliminar de cómo se vería la interfaz gráfica en la pantalla de siete pulgadas

El menú principal fue diseñado con tres botones principales: *Operación*, *Inventario* y *Salir*. Los botones de la interfaz gráfica debían ser lo suficientemente grandes para ser accesibles en la pantalla táctil, sin requerir una precisión extrema por parte del usuario para poder oprimirlo. De igual manera, se utilizaron distintos colores para cada botón, de manera que se pudieran diferenciarse mejor.



Figura 39: Esbozo de la pantalla de inicio

En el menú de operación, el usuario puede seleccionar el espacio de la bodega (columna y fila) donde quiere que el transelevador realice una operación de carga o descarga. Se colocaron dos botones adicionales: *Traer* y *Dejar*. Estos permitirían al usuario especificar si deseaba recoger una carga o dejarla en la ubicación seleccionada. Debajo de estos botones, se encontraba el botón de *Enviar*, que serviría para confirmar el comando que se quería mandar. En la parte inferior de la pantalla se agregó el botón de *Menú* para poder regresar al menú principal.



Figura 40: Esbozo del menú de operación

En el menú de inventario, el usuario puede seleccionar una ubicación del almacén (columna y fila) para verificar su estado actual, es decir, si se encuentra ocupado o si está disponible. También se incluyó el botón de *Modificar* para poder cambiar el estado de la ubicación seleccionada de forma manual. De igual manera, en la parte inferior de la pantalla se agregó el botón de *Menú* para poder regresar al menú principal.



Figura 41: Esbozo del menú de inventario

### 9.2.2. Programación

Esta fase del proyecto se implementó utilizando el lenguaje de programación *Python* y la librería *Tkinter*, la cual permitió la creación de los elementos visuales de la interfaz. Estos componentes gráficos incluyeron botones, menús desplegables (*OptionMenu*), etiquetas y mensajes de error. También se utilizó la librería *PIL* (*Python Imaging Library*), específicamente su módulo *ImageTk*, para poder gestionar las imágenes dentro de la interfaz, como la imagen del transelevador y el logotipo de la institución.

Las dimensiones de la ventana que se programó para la interfaz gráfica eran iguales a las de la resolución de la pantalla táctil de 7 pulgadas (800x480 píxeles). Este aspecto fue fundamental para garantizar que todos los elementos fueran visibles y que el usuario pudiera interactuar con ellos de manera cómoda.

Para desarrollar la interfaz gráfica se implementó una estructura lógica secuencial y un conjunto de funciones diseñadas para interactuar con el sistema de control del transelevador (Capítulo 7). Se comenzó con la definición de una estructura modular la cual permitiera gestionar múltiples menús y operaciones globales de manera eficiente.

El código se organizó programando funciones independientes, cada una destinada a gestionar una parte distinta de la interfaz. Estas podían ejecutarse para realizar cambios en los estados de las variables, limpiarlas, actualizarlas o enviarlas. De igual manera, habían funciones como *seleccionar\_operacion()*, la cual no solo cambiaba el estado visual de los botones, sino que también actualizaba la variable interna que almacenaba la operación seleccionada, lo que permitía enviar el comando correcto más adelante.

Además de este tipo de funciones, habían otras las cuales se enfocaban únicamente en la modificación de los elementos gráficos de la interfaz. Por ejemplo, la función *limpiar\_ventana()*, se encargaba de garantizar que al cambiar entre menús distintos, los componentes del menú anterior fueran removidos correctamente antes de cargar los nuevos. Esto se ejecutaba cada vez que el usuario cambiaba de un menú a otro, evitando la superposición de elementos gráficos.

Cada acción dentro de la interfaz dependía en su totalidad de la interacción del usuario. Debido a este factor, se implementó funciones de eventos que reaccionaban al cambio de selección en los menús desplegados o la activación de botones. Por ejemplo, la función que manejaba la selección de la columna y fila del inventario, siempre tenía que verificar y actualizar el estado de cada ubicación del almacén según los comandos que iba recibiendo.

La interfaz fue diseñada para funcionar de manera secuencial, es decir, que la salida del sistema dependía tanto de los valores de entrada actuales, como de los pasados. Primero el usuario selecciona la ubicación, luego elige la operación y por último envía el comando. Esta cronología de eventos se traducía a un flujo de trabajo en el código de la interfaz gráfica.

### 9.2.3. Envío de comandos y lógica interna

Uno de los principales desafíos en el desarrollo de la interfaz fue garantizar que los comandos se formaran de manera correcta y se enviaran sin errores al algoritmo de control del transelevador. Por esta razón, la lógica de la interfaz gráfica fue diseñada en torno a la función *enviar\_comando()* [36]. Esta se encargaba de generar y enviar los comandos desde la interfaz gráfica hasta un puerto serial de la *Raspberry Pi*.

Los comandos consistían en una combinación de tres caracteres:

- **Columna:** Indica la columna dentro del almacén donde se desea realizar la operación. Es representada por medio de una letra que puede ir desde la *A* hasta la *O*. Se define por medio de un menú desplegable dentro de la interfaz gráfica.
- **Fila:** Indica la fila dentro del almacén donde se desea realizar la operación. Es representada por medio de un número que puede ir desde 1 hasta 4. Se define por medio de un menú desplegable dentro de la interfaz gráfica.

- **Operación:** Indica qué acción se quiere realizar, si dejar o traer una carga. Se representa mediante las letras  $P$  (traer) y  $L$  (dejar). Se define dependiendo del botón que el usuario selecciona en la interfaz gráfica.

Por ejemplo, si el usuario seleccionaba la columna  $D$ , la fila 4 y la operación  $Dejar$ , el comando resultante sería  $D4L$ . La lógica interna del sistema verificaba que este comando fuera válido antes de enviarlo al algoritmo de control desarrollado en el [Capítulo 7](#).

#### 9.2.4. Pruebas locales y validación

Las pruebas iniciales de esta fase del proyecto se realizaron de manera local, utilizando *Visual Studio Code* [\[24\]](#) como entorno de desarrollo. Antes de realizar la interconexión de las dos partes del proyecto y de enviar los comandos a un dispositivo externo, como el Arduino Mega 2560, los comandos se enviaron a la terminal de comandos de un entorno local para verificar su fidelidad.

Estas pruebas fueron esenciales para poder identificar errores en la lógica de la interfaz gráfica, antes de proceder con la integración del hardware. Cada vez que el usuario interactuaba con la interfaz y mandaba una instrucción, el comando generado se mostraba en la terminal. Esto permitió verificar que las combinaciones de los caracteres se formaran de manera correcta. Este proceso de validación garantizó que la interfaz gráfica fuera robusta y confiable, minimizando la posibilidad de errores durante su uso en la fase de pruebas físicas con el transelevador.

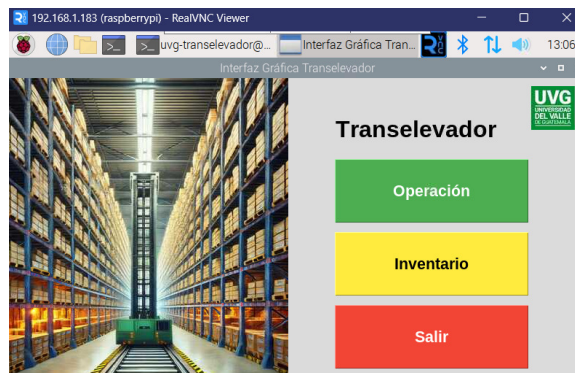


Figura 42: Pantalla de inicio



Figura 43: Menú de operación

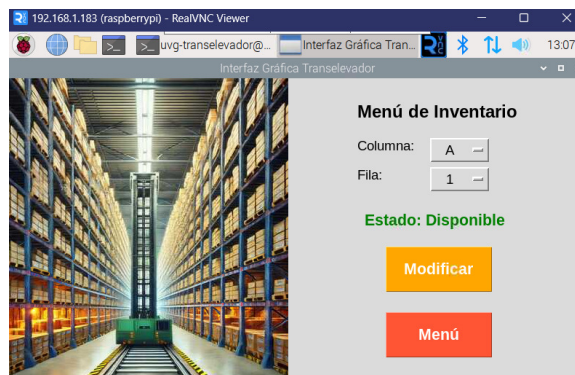


Figura 44: Menú de operación

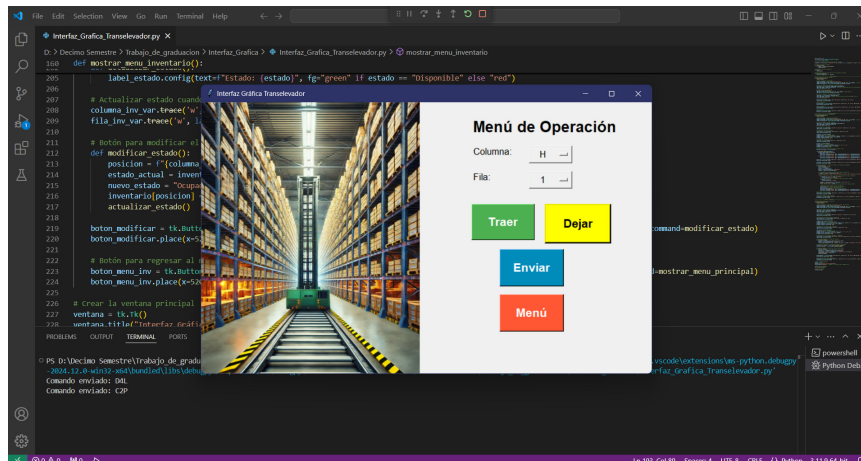


Figura 45: Pruebas locales de envío de comandos por comunicación serial

### 9.2.5. Programación defensiva

Uno de los puntos clave en este sistema fue la implementación de programación defensiva. Dado que la interfaz estaría operando en un entorno dinámico, con distintos tipos de au-

diencia, era fundamental que el sistema pudiera gestionar errores o interacciones incorrectas por parte de los usuarios.

Para impedir cualquier tipo de inconveniente o mal funcionamiento durante las demostraciones del proyecto, se implementaron las siguientes verificaciones:

- No se permite dejar una carga en un espacio ya ocupado.
- No se permite enviar un comando sin seleccionar todas las opciones necesarias (columna, fila, operación).
- Si se intenta realizar una operación que no era válida, se muestra un mensaje de error para que el usuario pudiera corregir la acción.

Este enfoque aseguró que el sistema fuera tolerante a errores y que siempre se guiara al usuario hacia interacciones que fueran válidas y seguras.

El prototipo de transelevador trilateral a escala fue restaurado y renovado por completo, logrando integrar un sistema funcional que combina un algoritmo de control robusto, una interfaz gráfica de usuario práctica y fácil de poder utilizar, programación defensiva y un inventario virtual simple. Se cumplió con las expectativas de precisión, fiabilidad y facilidad de uso que se plantearon en los objetivos del proyecto.

La restauración del sistema eléctrico permitió una operación estable, eliminando los problemas técnicos que limitaban su funcionalidad. Las conexiones seguras, la correcta selección de componentes y el cableado renovado garantizaron un flujo eléctrico adecuado, reduciendo el riesgo de fallos y extendiendo la vida útil del prototipo.

El algoritmo de control rediseñado demostró ser eficiente y preciso, permitiendo que el transelevador ejecutara movimientos en los ejes X, Y y Z, además de rotaciones de 180 grados. Este algoritmo, basado en una máquina de estados finitos, mejoró significativamente la sincronización y exactitud de los desplazamientos, lo que se reflejó en la capacidad del sistema para acomodar y recoger tarimas dentro de los espacios de la bodega simulada.

La interfaz gráfica desarrollada en Python y desplegada en una pantalla táctil de Raspberry Pi permitió una interacción fácil e intuitiva con el prototipo. Los usuarios, sin importar el contexto que tenían acerca del proyecto o su rango de edad, pudieron enviar comandos con facilidad, recibiendo retroalimentación visual en tiempo real sobre el estado del sistema. Este nivel de interacción hizo que el prototipo pudiera servir como una herramienta accesible tanto para fines educativos como para demostraciones técnicas.

La integración entre la interfaz gráfica y el algoritmo de control mediante comunicación serial aseguró una operación sincronizada y sin errores. Este vínculo permitió que el sistema ejecutara comandos con una respuesta rápida y precisa, validando el desempeño del diseño implementado.

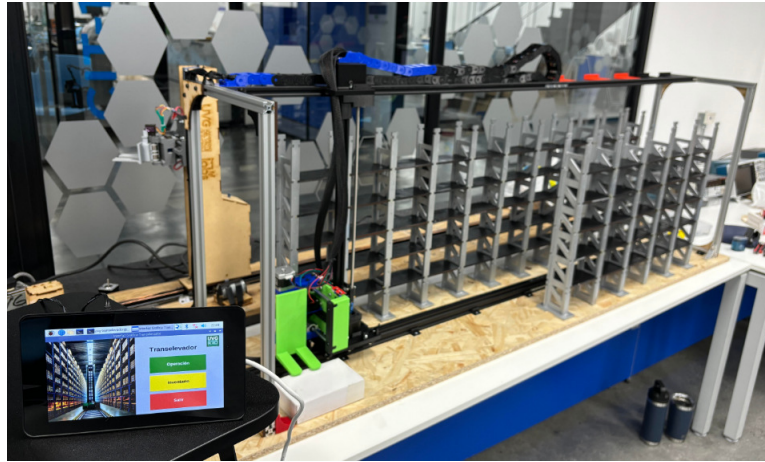


Figura 46: Pruebas de validación finales del prototipo de transelevador trilateral

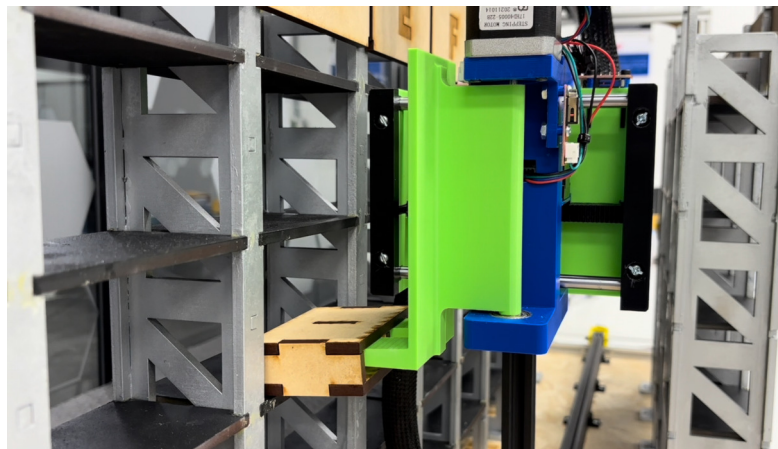


Figura 47: Prototipo de Transelevador Trilateral a escala en funcionamiento



Figura 48: Vista frontal del prototipo de transelevador terminado

- Se rediseñó e implementó un nuevo algoritmo de control basado en una máquina de estados finitos, el cual permitió gestionar de manera eficiente y precisa los movimientos del transelevador en tres ejes distintos, así como giros de 180°.
- Se desarrolló una interfaz gráfica de usuario que permitió a cualquier persona, sin importar su rango de edad o el contexto previo que tenía, interactuar de manera sencilla y directa con el transelevador.
- Los nuevos sistemas implementados garantizaron un manejo confiable de las operaciones, con mecanismos de recalibración automática y manejo de errores, aumentando significativamente la fiabilidad del prototipo.
- Se logró la integración completa entre el sistema de control y la interfaz gráfica, junto con los componentes físicos del prototipo, por medio de un protocolo de comunicación serial, asegurando la sincronización precisa de las acciones y la transferencia de datos.
- Las pruebas funcionales realizadas evidenciaron que el prototipo es capaz de operar de manera precisa, eficiente y segura, simulando escenarios del manejo automatizado para almacenar y recoger tarimas de carga en una bodega.
- El prototipo de transelevador trilateral a escala fue restaurado y renovado por completo, logrando integrar un sistema funcional que combina un algoritmo de control robusto, una interfaz gráfica de usuario práctica y fácil de poder utilizar, programación defensiva y un inventario virtual simple.
- El prototipo de transelevador trilateral a escala demostró ser un proyecto demostrativo y funcional, el cual aporta un valor educativo y técnico como una herramienta de aprendizaje para demostraciones de la automatización aplicada, reflejando la importancia de aplicar tecnologías emergentes, promoviendo el aprendizaje activo, las simulaciones, el análisis de posibles escenarios y el desarrollo de soluciones innovadoras a problemáticas reales del presente.

- Implementar un sistema de control en lazo cerrado, de manera que se aumente la redundancia del prototipo de transelevador trilateral. Esto se podría alcanzar añadiendo más finales de carrera y sensores de proximidad o mediante la incorporación de seguidores de línea, cámaras de visión artificial, entre otros. Esta mejora resultaría en un sistema mucho más seguro, versátil, capaz de gestionar problemas menos convencionales y también que no dependería de una autocalibración por cada ciclo de operación finalizado.
- Desarrollar una plataforma en la nube o una aplicación móvil que permita monitorear y controlar el prototipo de forma remota. Esto resultaría ser muy útil en escenarios donde se requiera supervisión continua o un control centralizado desde diferentes ubicaciones en una bodega real. Como proyecto demostrativo, también sería interesante que los usuarios pudieran interactuar y controlar el prototipo de forma remota.
- Implementar algoritmos de aprendizaje o inteligencia artificial para optimizar el control del transelevador, permitiendo que el sistema aprenda patrones de uso y adapte sus movimientos para reducir tiempos de operación y maximizar la eficiencia.
- Implementar un paro de emergencia el cual deshabilite todos los motores y detenga cualquier operación que se esté realizando en el momento.
- Programar rutinas con operaciones predefinidas para mostrar el funcionamiento completo del transelevador de manera rápida y concisa.
- Programar rutinas de auto calibración parecidas a las que se utilizan en las impresoras 3D, de manera que no se tenga que reiniciar todo el sistema en caso de un error.

- 
- [1] J. Valdez, “Sistema de traslación horizontal de transelevador para acomodamiento de cargas en bodega de Cementos Rocafuerte,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2022.
  - [2] V. Mombiela, “Sistema de traslación vertical para la integración en un transelevador prototipo para el acomodamiento de cargas en bodega de Grupo Roca Fuerte Guatemala,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2022.
  - [3] C. Chocooj, “Diseño y construcción de prototipo de sistema de sujeción de carga de un transelevador para el acomodamiento de tarimas en bodega de Grupo Roca Fuerte Guatemala,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2022.
  - [4] J. Villela, “Integración de interfaz gráfica y base de datos en sistema de transelevador para acomodamiento de cargas en bodega de GRUPO ROCAFUERTE,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2023.
  - [5] Progreso. “Nuestra Historia.” (2022), dirección: <https://progreso.com/nosotros/nuestra-historia/#:~:text=Progreso%20Labs%20habilita%20proyectos%20de,que%20faciliten%20los%20procesos%20constructivos.&text=Progreso%20X%20crea%20C%20dise%C3%B1a%20y,Progreso%20C%20startups%20y%20sus%20ecosistemas.>
  - [6] Mecalux. “Transelevador trilateral automático.” (2024), dirección: <https://www.mecalux.com.mx/almacenes-automatizados-para-tarimas/transelevador-trilateral-automatico>.
  - [7] Hubmaster. “The HUBMASTER® System.” (2020), dirección: <https://hubmaster-system.com/es>
  - [8] Mecalux. “Transelevadores con microprocesador, seguridad para la carga y total autonomía.” (2024), dirección: <https://www.mecalux.com.mx/blog/transelevadores-con-microprocesador>.

- [9] SAP. “¿Qué es un sistema de gestión de almacenes (WMS)? | SAP Insights.” (2019), dirección: <https://www.sap.com/latinamerica/products/scm/extended-warehouse-management/what-is-a-wms.html#:~:text=Un%20WMS%2C%20o%20sistema%20de,e1%20momento%20en%20que%20salen.>
- [10] Mecalux. “Warehouse Management System - Easy Mecalux WMS.” (2024), dirección: <https://www.mecalux.com/software/warehouse-management-system-wms>
- [11] A. A. for Advancing Automation. “Stepper Motor Tutorial: Learning the Basics.” (2017), dirección: <https://www.automate.org/motion-control/blogs/stepper-motor-tutorial-learning-the-basics>
- [12] StepperOnline. “Motor Paso a Paso Híbrido.” (2024), dirección: <https://www.omc-stepperonline.com/es/motor-paso-a-paso-hibrido>
- [13] A. A. for Advancing Automation. “Understanding Stepper Motor Types Operating Modes for Motion Control.” (2019), dirección: <https://www.automate.org/motion-control/blogs/understanding-stepper-motor-types-and-operating-modes-for-motion-control>
- [14] A. A. for Advancing Automation. “Microstepping Tutorial: A Brief Overview.” (2017), dirección: <https://www.automate.org/motion-control/blogs/microstepping-tutorial-a-brief-overview>
- [15] A. O. Store. “Arduino Mega 2560 Rev3.” (2020), dirección: <https://store.arduino.cc/products/arduino-mega-2560-rev3?srs1tid=AfmB0oo92Yi2LZ0dXW1sv2rZ666WK7frqigMJ1a3q>
- [16] RepRap. “RAMPS 1.4.” (2019), dirección: [https://reprap.org/wiki/RAMPS\\_1.4](https://reprap.org/wiki/RAMPS_1.4)
- [17] T. Instruments. “DRV8825.” (2014), dirección: <https://www.ti.com/product/es-mx/DRV8825#params>
- [18] A. T. Joy. “Guía Completa de Finales de Carrera.” (2024), dirección: <https://tameson.es/pages/final-de-carrera?>
- [19] R. Pi. “Raspberry Pi hardware - Raspberry Pi Documentation.” (2024), dirección: <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#raspberry-pi-3-model-b>
- [20] P. -. R. Electronics. “DRV8825 Stepper Motor Driver Carrier.” (2024), dirección: <https://www.pololu.com/product/2133>
- [21] RaspberryPi. “Raspberry Pi Ltd.” (2024), dirección: <https://www.raspberrypi.com/>
- [22] RaspberryPi. “Raspberry Pi OS using Raspberry Pi Imager.” (2024), dirección: <https://www.raspberrypi.com/software/>
- [23] RealVNC®. “VNC Viewer | VNC Connect.” (2024), dirección: <https://www.realvnc.com/es/connect/download/viewer/>
- [24] Microsoft. “Visual Studio Code.” (2021), dirección: <https://code.visualstudio.com/>

### 14.1. Video del funcionamiento



Figura 49: Código QR para ver el funcionamiento del prototipo de transelevador trilateral

<https://youtu.be/HbDdj4FUnsM>

