

# UACM

Universidad Autónoma  
de la Ciudad de México

NADA HUMANO ME ES AJENO

COLEGIO DE CIENCIA Y TECNOLOGÍA

LICENCIATURA EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS INDUSTRIALES

Diseño conceptual de un sistema de almacenamiento automático

TESIS QUE PARA OBTENER EL TÍTULO DE  
LICENCIADO EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS INDUSTRIALES

PRESENTA

**Carlos Josue Prado Sanchez**

Director de Tesis

**Dr. Mario Villafuerte Bante**

Codirector

**M.I. Francisco Javier Rosas Paredes**

Ciudad de México, abril 2026.

## **SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN**



### **UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO COORDINACIÓN ACADÉMICA**

#### **RESTRICCIONES DE USO PARA LAS TESIS DIGITALES**

#### **DERECHOS RESERVADOS<sup>©</sup>**

La presente obra y cada uno de sus elementos está protegido por la Ley Federal del Derecho de Autor; por la Ley de la Universidad Autónoma de la Ciudad de México, así como lo dispuesto por el Estatuto General Orgánico de la Universidad Autónoma de la Ciudad de México; del mismo modo por lo establecido en el Acuerdo por el cual se aprueba la Norma mediante la que se Modifican, Adicionan y Derogan Diversas Disposiciones del Estatuto Orgánico de la Universidad de la Ciudad de México, aprobado por el Consejo de Gobierno el 29 de enero de 2002, con el objeto de definir las atribuciones de las diferentes unidades que forman la estructura de la Universidad Autónoma de la Ciudad de México como organismo público autónomo y lo establecido en el Reglamento de Titulación de la Universidad Autónoma de la Ciudad de México.

Por lo que el uso de su contenido, así como cada una de las partes que lo integran y que están bajo la tutela de la Ley Federal de Derecho de Autor, obliga a quien haga uso de la presente obra a considerar que solo lo realizará si es para fines educativos, académicos, de investigación o informativos y se compromete a citar esta fuente, así como a su autor ó autores. Por lo tanto, queda prohibida su reproducción total o parcial y cualquier uso diferente a los ya mencionados, los cuales serán reclamados por el titular de los derechos y sancionados conforme a la legislación aplicable.

## **AGRADECIMIENTOS**

Primeramente, agradezco a Dios, mi creador, mi guía y el autor de todo lo que hoy soy. Sin Él nada de esto hubiese sido posible. Ha sido Él quien me ha bendecido abundantemente con conocimientos, habilidades y aptitudes que me han permitido crecer, aprender y superar cada desafío presentado en este camino. Cada capacidad desarrollada, cada aprendizaje adquirido y cada meta alcanzada han sido fruto de su gracia en mi vida. En los días de cansancio, angustia e incertidumbre, fue quien sostuvo mis pasos y renovó mis fuerzas para no rendirme. Le agradezco no solo por las bendiciones y los logros, sino también por cada dificultad, porque a través de ellas formó mi carácter, fortaleció mi fe y me enseñó a confiar plenamente en su propósito.

A mis padres, mi mayor ejemplo de amor y sacrificio. Gracias por acompañarme en cada etapa de mi vida estudiantil, por su apoyo incondicional y por creer en mí incluso cuando yo dudaba. Gracias por cada oración antes de salir de casa y por cada esfuerzo económico realizado para que pudiera continuar mis estudios. Gracias por enseñarme el valor de la responsabilidad, la disciplina y el esfuerzo. Este logro también es suyo. Han hecho un gran trabajo con su hijo, y me siento orgulloso de ser el reflejo de su dedicación. Eternamente agradecido con mis padres por ayudar a formarme no solo como Ingeniero, sino como persona.

Al Dr. Mario Villafuerte Bante le agradezco el apoyo brindado en mi carrera universitaria así como cada consejo y guía que ayudaron a fortalecer mis conocimientos.

Al M.I. Francisco Javier Rosas Paredes le agradezco el acompañamiento, tiempo, disposición y conocimiento brindado a lo largo de este trabajo de tesis, trabajar hombro a hombro con él fue una etapa enriquecedora en muchas áreas de mi vida.

“Porque Jehová da la sabiduría, Y de su boca viene el conocimiento y la inteligencia”

Proverbios 2:6 RVR1960

“Mira que te mando que te esfuerces y seas valiente; no temas ni desmayes, porque Jehová tu Dios estará contigo en dondequiera que vayas”

Josué 1:9 RVR1960

## Índice

|                                                                   |    |
|-------------------------------------------------------------------|----|
| Capítulo 1. ....                                                  | 5  |
| Introducción.....                                                 | 5  |
| 1.1 Planteamiento del problema. ....                              | 6  |
| 1.2 Justificación. ....                                           | 7  |
| 1.3 Objetivos. ....                                               | 9  |
| 1.3.1 Objetivo general. ....                                      | 9  |
| 1.3.2 Objetivos específicos.....                                  | 9  |
| 1.4 Metodología de solución del problema. ....                    | 10 |
| 1.5 Estructura del trabajo.....                                   | 10 |
| Capítulo 2. ....                                                  | 12 |
| Antecedentes. ....                                                | 12 |
| 2.1 Conceptos generales. ....                                     | 12 |
| 2.2 En qué consiste un proceso de almacenamiento.....             | 13 |
| 2.3 Sistematización del proceso. ....                             | 14 |
| 2.4 Automatización del proceso.....                               | 14 |
| 2.5 Herramienta de programación a implementar. ....               | 15 |
| 2.5.1 PLCs.....                                                   | 15 |
| 2.6 Propuesta para los dispositivos electrónicos del diseño. .... | 20 |
| 2.6.1 Robot cartesiano. ....                                      | 20 |
| 2.6.2 Motores pasó a paso. ....                                   | 21 |
| 2.6.3 Driver A4988. ....                                          | 23 |
| 2.6.4 Cilindro pistón. ....                                       | 23 |
| 2.6.5 Electroválvula.....                                         | 24 |
| 2.6.6 Sensores infrarrojos.....                                   | 25 |
| 2.6.7 Sensores Light Dependent Resistor (Fotorresistor). ....     | 26 |
| Capítulo 3. ....                                                  | 28 |
| Diseño del sistema.....                                           | 28 |
| 3.1 Sistematización de la estructura del almacén.....             | 28 |
| 3.2. Necesidad de automatización. ....                            | 29 |
| 3.3 Propuesta de solución a las necesidades. ....                 | 30 |

|                                                                                           |     |
|-------------------------------------------------------------------------------------------|-----|
| Capítulo 4. ....                                                                          | 37  |
| Especificaciones de la propuesta de diseño.....                                           | 37  |
| 4.1 Especificaciones de los elementos propuestos. ....                                    | 37  |
| 4.1.1 Características de los PLC's. ....                                                  | 37  |
| 4.1.2 Características del módulo de sensor infrarrojo.....                                | 38  |
| 4.1.3 Características del módulo de sensor LDR.....                                       | 38  |
| 4.1.4 Características del motor paso a paso. ....                                         | 39  |
| 4.1.5 Características del Driver A4988. ....                                              | 40  |
| 4.1.6 Características del cilindro pistón.....                                            | 41  |
| 4.2 Caracterización de las señales de entrada y salida de los distintos componentes. .... | 42  |
| 4.3 Elementos de interfaz y acondicionamiento de las señales entre etapas.....            | 43  |
| 4.3.1 Módulo de relevador. ....                                                           | 43  |
| 4.3.2 Electroválvula.....                                                                 | 45  |
| 4.4 Mapa de entradas y salidas (I/O).....                                                 | 46  |
| 4.5 Diagrama de potencia y control de los motores a pasos .....                           | 47  |
| Capítulo 5. ....                                                                          | 48  |
| Algoritmos y programas. ....                                                              | 48  |
| 5.1 Algoritmo para la lógica del sistema (Parte I). ....                                  | 53  |
| 5.2 Algoritmo para la lógica del sistema (Parte II). ....                                 | 62  |
| 5.3 Algoritmo movimiento de motores. ....                                                 | 68  |
| 5.4 Programa en PLC para la lógica del sistema.....                                       | 78  |
| 5.5 programa en PLC para el movimiento de motores. ....                                   | 89  |
| Capítulo 6. ....                                                                          | 99  |
| 6.1 Conclusiones.....                                                                     | 99  |
| Referencias. ....                                                                         | 103 |

# Capítulo 1.

## Introducción.

Desde hace algunos años los sistemas de almacenamiento automático han generado un gran cambio en industrias como las de logística y manufactura. Esto debido principalmente a que estos sistemas son ideales para mejorar el manejo de inventarios, reducir los tiempos de operación y disminuir los errores dentro del área de almacenes. Para ello usualmente se hace uso de grúas apiladoras, sistemas de elevación vertical y robots autónomos, que suelen trabajar de manera conjunta con softwares de gestión de almacenes para lograr el objetivo de mejorar los procesos de almacenaje.

Este tipo de sistemas actualmente se ocupan en una gran variedad de industrias como la automotriz, las de alimentos y bebidas, la farmacéutica y principalmente en las industrias del sector de comercio electrónico, las cuales han visto un crecimiento importante y por ello han demandado soluciones automatizadas para satisfacer el incremento en los volúmenes de pedidos y la complejidad que también ha aumentado en sus procesos logísticos debido al alto volumen de estas empresas [4]. Ejemplo de lo anterior se puede ver en empresas como Amazon, DHL y Walmart, que han implementado este tipo de sistemas en sus centros de distribución, dando lugar a que empresas como ABB Robotics, Daifuku y Schaefer Systems International lideren el sector [5].

Un factor que también ha contribuido al incremento en la implementación en este tipo de sistemas es el desarrollo del internet de las cosas y la mejora en tecnologías como la visión artificial. Tecnologías que han ayudado a mejorar una implementación más profunda de estos sistemas, dando lugar incluso a sistemas como AutoStore y Attabotics [6].

El cambio en las necesidades de consumo y el amplio desarrollo del comercio electrónico, aunado a la demanda de menores tiempos de entrega, han hecho de la automatización de sistemas de almacenamiento una herramienta importante para que las empresas puedan seguir siendo competitivas. Haciendo que un proyecto como el aquí planteado sea de gran importancia como punto de partida para introducirse en un campo de gran relevancia y en amplia expansión.

### **1.1 Planteamiento del problema.**

El proceso de almacenaje es indispensable para el funcionamiento de cualquier tipo de industria y de cualquier nivel. Ya sea en empresas que se dedican directamente al proceso de venta de productos al menudeo o mayoreo, en donde resulta indispensable contar con un inventario de productos adecuado para poder satisfacer la demanda, o en industrias manufactureras donde es necesario contar con un stock de productos para su posterior distribución y de las materias primas base para producir.

Por esta razón es importante para el buen funcionamiento de cualquier industria, un buen manejo del sistema de almacenamiento y si bien en muchos sistemas de almacenamiento se siguen empleando recursos humanos para su operación, la automatización de estos sistemas es una herramienta cada vez más presente en distintas industrias, sobre todo en aquella dedicadas a la logística y distribución a gran escala como Amazon, mercado libre y diversas empresas de este giro, ya que un sistema de almacenamiento automatizado les permite reducir los errores en el proceso, adicionalmente la automatización de este tipo de procesos, favorece la disminución de riesgos de trabajo por parte de operarios de esta área, ya que en el traslado de mercancía es frecuente que se presenten lesiones en el personal operativo.

De este modo la automatización de sistemas almacenamiento no solo se presenta como una herramienta para mejorar el proceso, sino también como un elemento para la mejora de la seguridad industrial.

En el caso del presente trabajo se busca generar un diseño a nivel conceptual, que sea ilustrativo de cómo se pueden resolver este tipo de situaciones industriales y que permita en trabajos futuros, llegar a un punto de implementación considerando las limitaciones presupuestales para entes académicas como nuestra universidad. Para la automatización de este tipo de sistemas se establecieron ciertas actividades indispensables para el proceso de automatización, número uno se debe de detectar cuando ha llegado un paquete para su almacenamiento, posteriormente se debe detectar qué espacios se encuentran disponibles para poder almacenar el paquete, después de ello se debe de decidir en qué espacio se colocará el paquete y finalmente se debe proceder a trasladar el paquete al espacio seleccionado. Estas actividades están asociadas únicamente al almacenamiento del paquete, para el proceso de suministro de paquetes se debe de guardar en una memoria la ubicación del paquete y posteriormente cuando dicho paquete sea solicitado, se debe trasladar el paquete del espacio en donde se ubica a la posición de recepción de paquetes.

## **1.2 Justificación.**

El presente proyecto se presenta como un acercamiento a una problemática industrial real, esto debido a que se ha visto el gran impacto que tienen los sistemas de almacenamiento automático en la reducción de costos de producción para diversas empresas, destacando entre ellas las que tienen un enfoque de comercio electrónico, esta reducción de costos podría creerse que se puede ver opacada por la inversión que implica la implementación de estos sistemas, sin embargo los posibles beneficios y el retorno de inversión promedio en estas implementaciones compensan la posible inversión en estos sistemas, como se muestra enseguida.

Si bien la inversión inicial en un sistema de almacenamiento automático varía en función del grado de automatización, alcanzando cifras de varios millones de dólares cuando las empresas buscan una automatización completa, haciendo uso de robots autónomos avanzados, sistemas de gestión de inventarios en tiempo real, así como sistemas de embalaje y recolección de productos [2]. Este proceso de automatización se puede implementar de manera gradual o bien con un nivel de sofisticación menor, reduciendo de este modo sus costos iniciales, en áreas de ser más accesibles para empresas de menor tamaño o con un menor presupuesto como se muestra en [3].

En el aspecto de retorno de inversión de este tipo de sistemas, se ha observado que este se puede alcanzar en un plazo promedio de entre 2 a 5 años, dependiendo del grado de automatización y de la sofisticación de los sistemas implementados, así como del tamaño de la producción de la empresa. La justificación de dicho retorno de inversión se origina en los beneficios observados que produce, como la reducción de costos operativos en cuanto a mano de obra, la reducción de errores de inventario, el mayor aprovechamiento del espacio de almacén y la reducción de los tiempos de procesamiento. Empresas como Amazon y Ocado han conseguido reducir de manera significativa sus tiempos de entrega debido a la implementación de este tipo de sistemas, logrando no solo una mayor eficiencia operativa sino además una mayor satisfacción del cliente [1], [2]. Es importante destacar que el retorno de inversión en este tipo de sistemas se ve acelerado sobre todo en industrias donde el costo de almacenaje representa un porcentaje alto de los costos operativos, sobre todo en empresas dedicadas al comercio electrónico y comercio minoritario, donde un manejo de almacén eficiente y una rápida velocidad de entrega son de gran importancia. Es en especial en este tipo de empresas donde la automatización de almacén presenta un mayor impacto debido al alto porcentaje de gastos asociado al sistema de almacenaje.

Si bien el resultado de este trabajo no busca establecerse como una solución alterna a los elementos existentes actualmente en la industria, si consigue establecer las bases para generar una plataforma que funcione como elemento de acercamiento para los estudiantes de nuestra institución a estructuras y problemas de índole industrial. Este trabajo, por ende, aunque no constituye un producto final que aporte una innovación a la industria, si establece bases para un posible producto de uso académico y muestra como puede ser estructurado un proyecto de tipo industrial.

### **1.3 Objetivos.**

#### **1.3.1 Objetivo general.**

Realizar un diseño conceptual de un sistema de almacenamiento automático, que sea capaz de detectar cuando llega un paquete para ser almacenado, ubicar los espacios disponibles para almacenar, seleccionar el espacio para almacenar el paquete y trasladar el paquete a la ubicación seleccionada.

#### **1.3.2 Objetivos específicos.**

- Definir los requisitos y/o necesidades que debe cubrir un sistema de almacenamiento automático.
- Realizar una propuesta para cubrir los requisitos definidos.
- Realizar una propuesta conceptual de un robot prismático que permita llevar a cabo la traslación de los paquetes a su espacio de almacenamiento.
- Realizar una propuesta conceptual del sistema de control, censado y actuación para la automatización del sistema completo a escala.

## **1.4 Metodología de solución del problema.**

Para dar solución al desarrollo del sistema de almacenamiento automático, se procederá a identificar cuáles son las acciones específicas que realizan este tipo de sistemas y se seleccionan cuáles son las acciones que se trabajarán en el diseño, posteriormente se revisará qué alternativas existen para realizar dichas acciones de manera específica sin contemplar el sistema en su totalidad, para posteriormente identificar aquellas que se pueden utilizar de manera conjunta para conseguir un mejor resultado, finalmente se revisarán las necesidades específicas de implementación que surjan y se buscará dar una solución a las mismas por medio de la elección de los mejores elementos mecánicos, eléctricos y electrónicos a nuestro alcance.

## **1.5 Estructura del trabajo.**

A continuación, se describe brevemente el contenido que tendrá este trabajo de investigación.

- Capítulo 1: Se plantea la necesidad existente en el sistema propuesto, fijando objetivos a cumplir así como la estructura del trabajo. Definiendo el proyecto.
- Capítulo 2: Se establecen los antecedentes sobre los robots prismáticos, almacenes, sistemas automatizados y PLCs
- Capítulo 3: Se presenta la estructura general del sistema, tanto en la ubicación de los sensores necesarios como la estructura tanto de la banda transportadora y el robot prismático, que funcionarán como elementos actuadores, de igual forma se presenta la estructura del almacén y como trabajan estos elementos de manera conjunta.
- Capítulo 4: Se especifican los elementos seleccionados que funcionaran como sensores y actuadores, así como los elementos encargados de realizar la parte lógica y los elementos de interfaz entre sensores, actuadores y la parte lógica.
- Capítulo 5: Se presenta la lógica de programación estructurada en los PLCs, así como los elementos necesarios para llevar a cabo la implementación.

- Capítulo 6: Se presentan las conclusiones obtenidas en el desarrollo del trabajo.

## Capítulo 2.

### Antecedentes.

En el presente proyecto de investigación se pretende diseñar de manera conceptual un sistema de almacén automatizado destinado para lugares como industrias, centros logísticos y empresas de paquetería y mensajería. Con el fin de optimizar espacios, realizar el almacén de manera rápida y eficiente, tener un mayor control de inventario y fácil rastreo de productos.

En este apartado, se detallan conceptos fundamentales como en qué consiste un proceso de almacenamiento, la sistematización del proceso y su automatización, así como desarrollar una propuesta de solución a través de herramientas de programación y dispositivos electrónicos. Abordando cada herramienta y dispositivo propuesto, mencionando su funcionamiento, y el por qué fue seleccionado para cubrir cada necesidad.

### 2.1 Conceptos generales.

**Almacenamiento:** acción de guardar artículos u objetos en un específico lugar seccionado con la finalidad de evitar desorden y tener una fácil localización de lo almacenado para su posterior uso [7].

**Proceso:** es la forma en la que se realiza una cierta tarea o se desarrolla un cierto producto. Dependiendo de la complejidad del proceso puede estar compuesto de distintas etapas e incluso requerir que estas estén retroalimentadas con el fin de eliminar errores al final del proceso [8].

**Sistematizar:** estructuración de un proceso descomponiendo todo el proceso en una serie de pasos simples y secuenciales aportándole limitaciones, restricciones y orden [9].

**Automatización:** tomar un proceso enteramente sistematizado con la finalidad de realizar dichas tareas de manera autónoma utilizando componentes como sensores, actuadores y herramientas de programación, quitando en parte la intervención humana.

Teniendo eficiencia en el desarrollo de las tareas, hechas de manera más rápida y con un mejor control [8].

**Packing:** proceso logístico de ubicación, recolección y entrega de un paquete.

**Picking:** proceso logístico de almacenamiento seguro y ordenado de un paquete.

## 2.2 En qué consiste un proceso de almacenamiento.

Un proceso de almacén es una gestión de ciertas actividades que nos permiten tener un espacio destinado y delimitado de manera óptima para conservar productos o materiales por un lapso de tiempo determinado, con la finalidad de tener orden y la fácil ubicación de estos [10]. Cabe mencionar que un proceso de almacenamiento es repetitivo, es decir, cada una de las etapas se llevan a cabo constantemente por lo que es necesario tener un control preciso de cada etapa [11].

Tal gestión de actividades en un proceso de almacén se puede apreciar en distintas áreas, por ejemplo en la vida cotidiana cuando realizas actividades de compra en el supermercado la línea a seguir es:

- Selección y compra de productos.
- Guardar y ordenar los productos en su espacio correspondiente (Almacén).
- Ubicación de cierto producto para su consumo o uso.

En este proyecto nuestras etapas de almacén vienen dadas de la siguiente manera:

- Recolección de productos [12].
- Ordenar en espacios disponibles [10].
- Almacenar la dirección de donde fue ordenado el producto [12].
- Ir a la dirección de almacenamiento según el producto a preparar para su expedición [11].

## **2.3 Sistematización del proceso.**

La sistematización conlleva procesos bien estructurados que trabajen de manera óptima en cada etapa del sistema. Sistematizar un proceso nos permite organizar cada etapa que interviene en nuestro sistema y poder dar un seguimiento secuencial e iterativo, considerando los diferentes eventos que pueden ocurrir para así contar con un sistema robusto y completo, minimizando errores y brindando fiabilidad al sistema [13].

## **2.4 Automatización del proceso.**

El objetivo general de la automatización de un proceso es que la intervención humana sea mínima o en su totalidad nula, todo esto con el objetivo de realizar actividades pesadas, complejas e iterativas de manera autónoma, fácil, rápida, eficiente y controlada eliminando los errores que pudieran existir en los procesos del sistema causados por la intervención humana.

Para llevar a cabo la automatización de un proceso requerimos de maquinarias y dispositivos tecnológicos como lo es el software y el hardware, los cuales trabajando de manera conjunta son capaces de realizar todo tipo de procesos de producción, selección y control [14].

Como se menciona en [14] la estructura de operación se compone de tres elementos necesarios como lo son:

- El sistema: en este se encuentra el software de control y programación, donde se generan las instrucciones desde una computadora.
- La máquina: este es el autómatas encargado de ejecutar las instrucciones provenientes del sistema.
- El producto: aquí se tiene el resultado final de lo llevado a cabo en el proceso.

Es decir, la estructura de la automatización se compone básicamente de tres niveles de funcionamiento:

- Nivel de supervisión: se compone de ordenadores, computadoras industriales o panel de control según la necesidad de la empresa, los cuales utilizan un sistema operativo estándar en donde se trabaja con un software en especial con el objetivo de observar y parametrizar el proceso a ejecutar. Usando como red de comunicación el protocolo Ethernet [14].
- Nivel de control: se ejecutan los programas los cuales contienen las instrucciones a realizar en el proceso de automatización, para ello se utilizan generalmente PLC (controladores lógicos programables), donde se puede utilizar más de un PLC interconectándolos por medio de algún protocolo de comunicación industrial [15].
- Nivel de campo: en esta parte se encuentran los dispositivos de censado y actuadores los cuales son conectados a un PLC por medio de un Bus de campo. Por otro lado, la conexión entre los dispositivos de nivel de campo y el PLC es normalmente de tipo punto a punto, es decir, están conectados directamente sin la intervención de cualquier otro dispositivo [14].

Sistematizar el proceso nos permite dar pauta para implementar la automatización en el proceso, donde se definieron las actividades a realizar así como el orden y control que deben llevar considerando los distintos eventos existentes en el sistema. Posteriormente se consideró las herramientas útiles y capaces que puedan realizar las actividades planteadas.

## **2.5 Herramienta de programación a implementar.**

### **2.5.1 PLCs.**

Un Controlador Lógico Programable o por sus siglas en inglés PLC “Programmable Logic Controller”, es considerado una computadora de grado industrial el cual es capaz de

realizar funciones de control y automatización, así como también es un reemplazo al control eléctrico el cual utilizaba relés mecánicos [16],[17].

Como se menciona en [18], los componentes de Hardware de un PLC son los siguientes:

- Fuente de alimentación.
  - Interna de 120 VAC.
  - Externa de (12VDC, 24VDC o 48VDC)
- Entradas y salidas.
  - Discreto (Digital, Binario).
  - Analógica.
- Módulos especiales.

En las entradas del PLC se conectan los sensores los cuales nos envían señales en forma de voltaje y en las salidas del PLC se conectan los actuadores, es decir, los elementos que transformaran las señales en movimientos, cabe mencionar que un PLC puede tener salidas de tipo relay, transistor o triac y dependiendo de la aplicación se opta por un tipo en específico [17].

Los componentes de Software de un PLC son los siguientes:

- Unidad Central de Procesamiento (CPU). Este es el cerebro del PLC el cual recibe las señales de entrada, ejecuta el programa y envía señales a las salidas de manera controlada [16].
- Memorias.
  - RAM. Aquí se almacena el programa de ejecución y datos temporales como las señales de entrada y salida
  - ROM. Esta contiene el sistema operativo y funciones básicas.
  - Flash. Permite reprogramar, tiene la capacidad de mantener el programa almacenado aun estando apagado y guarda configuraciones y parámetros [18].

El lenguaje de programación de un PLC se destaca principalmente 2 como lo son:

1. Diagramas de escalera (Ladder - LD). Como se menciona en [19] funciona básicamente a través de la lógica de contactos de relevadores y bobinas lo

cual lo hace un lenguaje fácil de entender visto desde el lado eléctrico. Para entender la forma de programación de este lenguaje se puede basar en el uso de las compuertas lógicas como AND, OR y NOT y sus derivadas como NAND, NOR, XOR y XNOR, así como la relación que tienen estas con los contactos de relevadores. Cada una de estas compuertas cuenta con una tabla de verdad las cuales muestran las diferentes combinaciones de entrada y sus respectivas salidas, basándose en números booleanos (1,0) que traducidos a la lógica sería (verdadero, falso) o en la parte eléctrica sería traducido a que hay presencia de voltaje o no, es decir activado o desvariado. Dichas tablas de verdad se muestran a continuación, donde también se muestra su relación con los contactos de relevadores.

- Compuerta AND-(Y)

Esta compuerta lógica la cual significa (Y) hace alusión a que se debe cumplir una condición (Y) también otra condición en sus entradas para que así su salida pueda ser activada de lo contrario no, es decir, se deben cumplir ambas condiciones de entrada en valor booleano 1 para que en la salida tengamos una señal de activación. En la siguiente Tabla 2.1 se muestra su tabla de verdad de dicha compuerta así como su equivalencia en contactos de relevadores y bobinas en diagrama escalera en la Figura 2.1:

**Tabla 2.1. Tabla de verdad de la compuerta AND.**

| Entrada A | Entrada B | Salida |
|-----------|-----------|--------|
| 0         | 0         | 0      |
| 0         | 1         | 0      |
| 1         | 0         | 0      |
| 1         | 1         | 1      |



**Figura 2.1. Equivalente de la compuerta AND en diagrama escalera.**

- Compuerta OR-(O)

Esta compuerta lógica la cual significa (O) hace alusión a que se debe cumplir una condición (O) otra condición en sus entradas para que así su salida pueda ser activada de lo contrario no, es decir, basta con que se cumpla una sola condición de entrada en valor booleano 1 para que en la salida tengamos una señal de activación. En la siguiente Tabla 2.2 se muestra su tabla de verdad de dicha compuerta así como su equivalencia en contactos de relevadores y bobinas en diagrama escalera en la Figura 2.2:

**Tabla 2.2. Tabla de verdad de la compuerta OR.**

| Entrada A | Entrada B | Salida |
|-----------|-----------|--------|
| 0         | 0         | 0      |
| 0         | 1         | 1      |
| 1         | 0         | 1      |
| 1         | 1         | 1      |



**Figura 2.2. Equivalente de la compuerta OR en diagrama escalera.**

- Compuerta NOT- (NO)

Esta compuerta lógica la cual significa (NO) invierte la señal de entrada que recibe, es decir, al ingresar un valor booleano 1 a su entrada obtendremos un 0 en su salida y viceversa. En la siguiente Tabla 2.3 se muestra su tabla de verdad de dicha compuerta así como su equivalencia en contactos de relevadores y bobinas en diagrama escalera en la Figura 2.3:

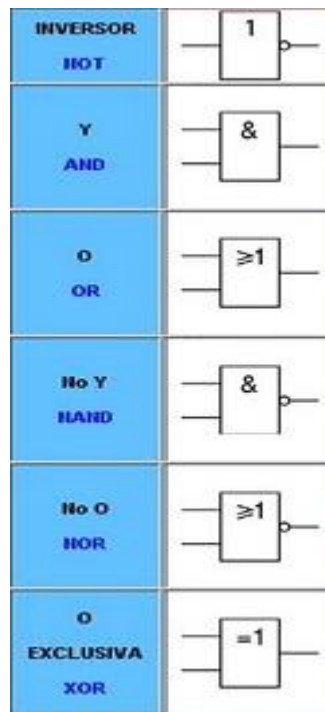
**Tabla 2.3. Tabla de verdad de la compuerta NOT.**

| Entrada | Salida |
|---------|--------|
| 0       | 1      |
| 1       | 0      |



**Figura 2.3. Equivalente de la compuerta NOT en diagrama escalera.**

2. Diagramas a bloques (Function Block Diagram - FBD). Se basa en gráficos los cuales utilizan bloques funcionales como AND, OR, NOT y sus derivadas como se muestran en la Figura 2.4 así como también temporizadores, contadores, etc. Donde se conectan entre sí por medio de sus entradas o salidas para generar instrucciones lógicas.

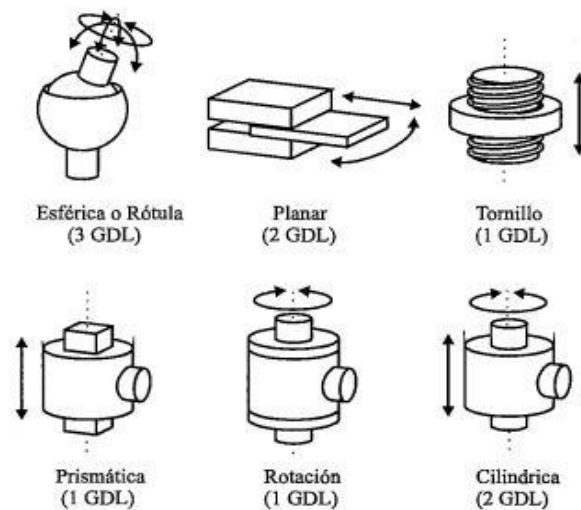


**Figura 2.4. Equivalencia de las compuertas lógicas en diagrama a bloques.**

## 2.6 Propuesta para los dispositivos electrónicos del diseño.

### 2.6.1 Robot cartesiano.

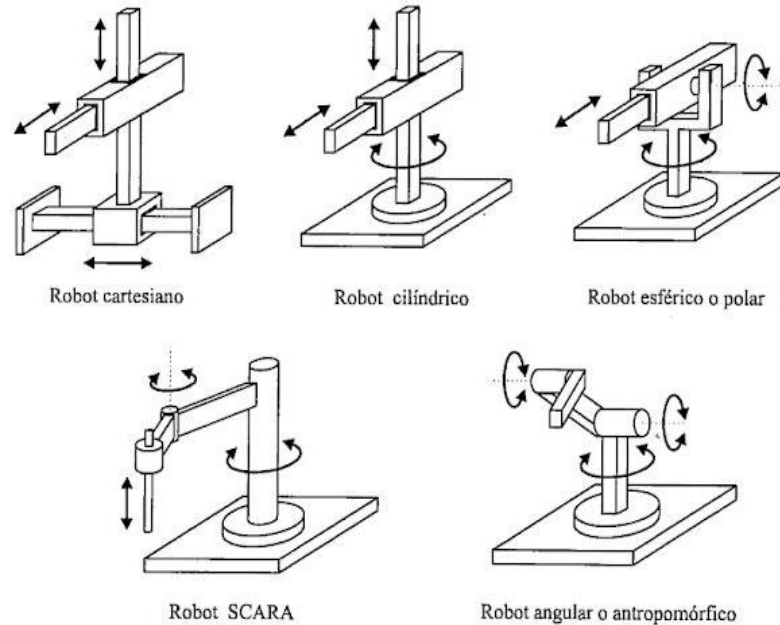
Como se menciona en [20] todo robot industrial mecánicamente está formado por una serie de elementos o eslabones conectados mediante articulaciones permitiendo que exista un movimiento con respecto a una referencia. El movimiento de la articulación puede ser de tipo rotativo, de traslación o una combinación de estos, en la Figura 2.5 se muestra los seis tipos de articulaciones para formar un robot.



**Figura 2.5. Tipos de articulaciones para robots.**

El (grado de libertad) GDL hace referencia a los movimientos independientes que son generados en una articulación.

La combinación de los distintos tipos de articulaciones o el uso de una sola articulación permiten la construcción de un robot, esto da pauta a la existencia de distintas configuraciones de robots como las que se muestran en la Figura 2.6.



**Figura 2.6. Combinaciones de articulaciones más comunes robots industriales.**

El robot cartesiano está formado solo por articulaciones de tipo prismático el cual puede desplazarse en los ejes X (movimiento horizontal), Y (movimiento vertical) y Z (profundidad, es decir hacia adelante o atrás). Los grados de libertad en un robot se obtienen al sumar los grados de libertad de las articulaciones que lo conforman. En base a esto el robot cartesiano cuenta con 3 grados de libertad ya que cuenta con 3 articulaciones y cada una con un movimiento independiente.

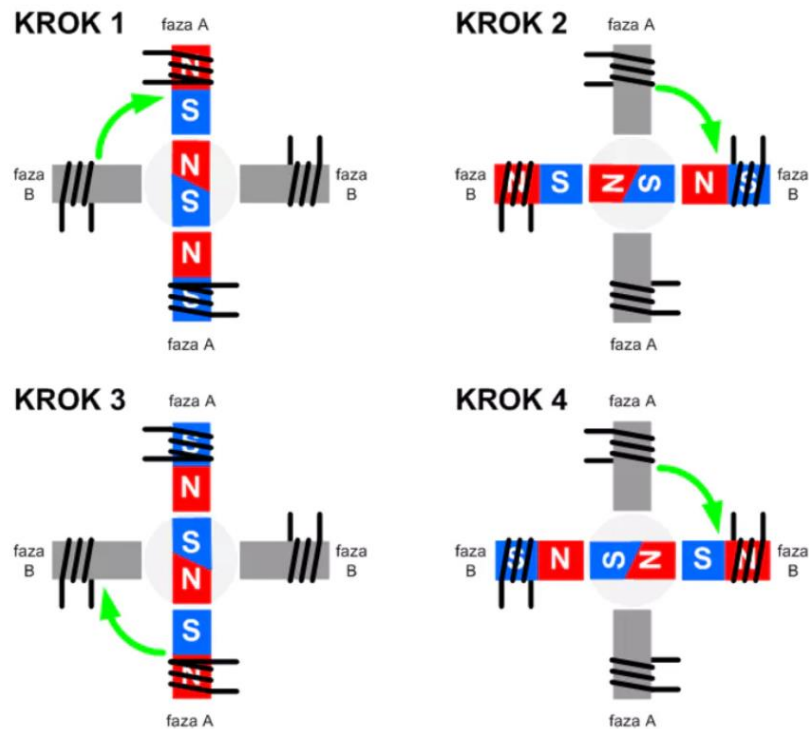
### 2.6.2 Motores pasó a paso.

Como se menciona en [21] este tipo de motor al igual que otros está compuesto por un estator que es la parte estática y un rotor el cual es la parte móvil donde por medio de un eje puede ser transmitido dicho movimiento rotatorio. Este tipo de motor viene estructurado de la siguiente manera:

- El estator normalmente está hecho de acero u otro tipo de metal el cual forma la carcasa del motor, donde vienen montadas internamente bobinas en lugares específicos que rodean el rotor.

- En el rotor vienen montados imanes los cuales ayudan a que exista una interacción entre el campo magnético que genera la bobina al ser energizada y el mismo que proporciona el imán, generando así un movimiento rotativo.

Cabe mencionar que tanto las bobinas como los imanes son de suma importancia en un motor ya que de ambos depende que se genere un movimiento rotativo gracias a sus características magnéticas y su respuesta ante una corriente eléctrica proporcionada.



**Figura 2.7. Operación de las bobinas de un motor a pasos.**

Como se ilustra en la Figura 2.7 el funcionamiento de cómo opera el motor a pasos se genera de la siguiente manera:

- Al energizar un par de bobinas se genera un campo magnético a su alrededor esto dado por la Ley de Ampere, generando así un electroimán.
- Dicho electroimán atrae al imán montado sobre el rotor para alinearse con su campo, generando así un cierto ángulo de desplazamiento, es decir, se comienza

a mover el eje del motor sólo una fracción de giro de los 360, llamado paso para alinear los campos.

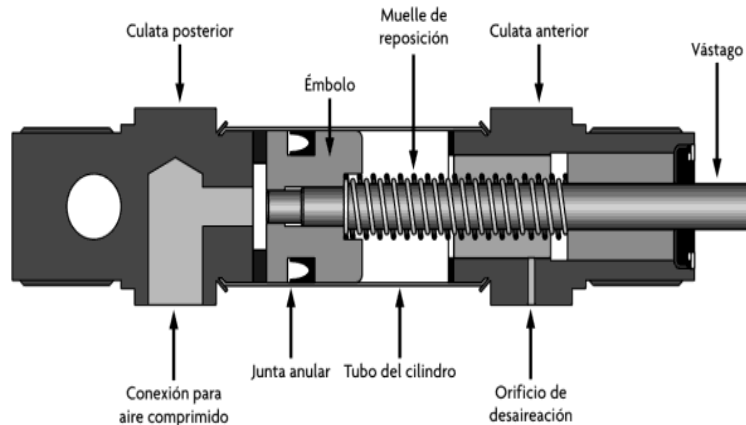
- La energización de las bobinas nos va permitir generar ciertos pasos en secuencia según nuestra necesidad, es decir, se generara una secuencia de energización de las bobinas donde conforme la secuencia cambie se generan ciertos pasos de rotación.
- El orden de la secuencia de activación de las bobinas, se genera a partir de un controlador especial para este tipo de motores, el cual únicamente recibe una señal pulsante para ir alternando la secuencia de energización de la bobinas para lograr girar el número de pasos de acuerdo al número de pulsos enviados, adicionalmente se puede enviar una señal al controlador que permite cambiar el sentido de giro del motor.

### **2.6.3 Driver A4988.**

Para poder controlar parámetros de nuestro motor a pasos como la posición y dirección, así como alimentarlo de manera adecuada es de suma importancia implementar un driver el cual nos ayudará a adaptar las señales que provienen del PLC a las que requiere el motor para su correcto funcionamiento [21].

### **2.6.4 Cilindro pistón.**

Un cilindro neumático tiene como estructura un cilindro cerrado, donde en su interior contiene un pistón el cual se desliza y transmite el movimiento hacia afuera por medio de un vástago. En la Figura 2.8 se pueden apreciar cada una de estas partes.



**Figura 2.8. Partes principales de un cilindro neumático de simple efecto.**

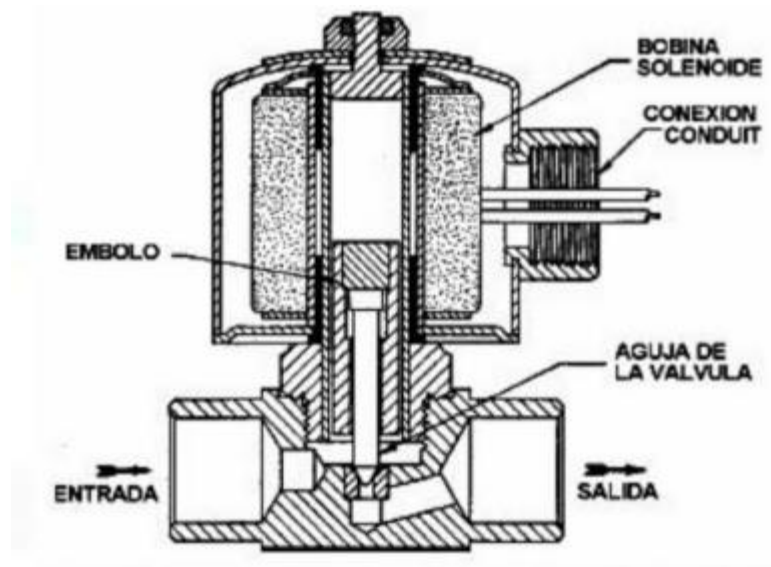
Dicho dispositivo utiliza como fuente de alimentación un compresor de aire, los cuales convierten la energía del aire comprimido en trabajo mecánico produciendo un movimiento lineal [22].

### 2.6.5 Electroválvula.

Como se menciona en [23] una electroválvula es un dispositivo electromecánico el cual controla un fluido dentro de un conducto, es decir, tiene la función de un interruptor la cual es activada de manera eléctrica y cuenta solo con posiciones en abierto o cerrado, es decir, no se regula el flujo del aire solo existe o no la presencia del flujo a la salida.

Funciona a través de una bobina solenoide que al ser alimentada con una corriente eléctrica esta genera un campo magnético donde el embolo es atraído abriendo el paso al fluido, al cortar la corriente eléctrica el embolo regresa a su estado normal impidiéndole el paso al fluido. Esto considerando una electroválvula de tipo normalmente cerrada, la de tipo normalmente abierta opera de manera contraria al ser alimentada o no.

En la Figura 2.9 se muestra la composición de una electroválvula.



**Figura 2.9 Componentes de una electroválvula.**

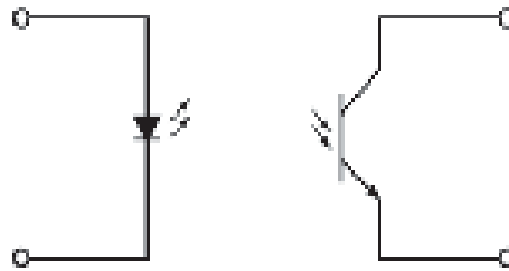
Este elemento de válvula permite controlar el cilindro pistón mediante una señal eléctrica proveniente del PLC.

### **2.6.6 Sensores infrarrojos.**

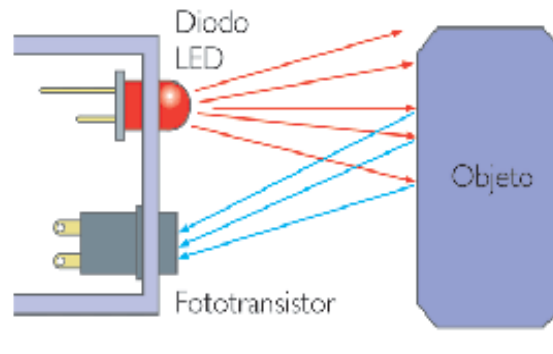
Este sensor se basa en el uso de luz infrarroja la cual es invisible al ojo humano ya que dentro del espectro electromagnético esta tiene como característica una longitud de onda mayor y frecuencia menor a la de la luz roja visible, es por ello que la palabra infrarrojo deriva del término infra-debajo y rojo [24].

Así como se aprovecha la propiedad de un cuerpo con temperatura el cual emite radiación infrarroja, lo cual se utiliza en múltiples tecnologías [25].

Como se muestra en la Figura 2.10 este tipo de sensor utiliza como componentes electrónicos un diodo emisor de luz infrarroja y un transistor con una sensibilidad capaz de detectar dicha emisión, es decir, el diodo emite luz infrarroja y al chocar en un objeto esta se refleja siendo receptada por el transistor como se muestra en la Figura 2.11.



**Figura 2.10. Esquema eléctrico de un sensor infrarrojo.**



**Figura 2.11. Modo de operación de un sensor infrarrojo reflexivo.**

### **2.6.7 Sensores Light Dependent Resistor (Fotorresistor).**

Como se menciona en [26] “Un fotorresistor es un resistor dependiente de la luz. Este dispositivo de hecho de material semiconductor de alta resistencia el cual en oscuridad tiene unos cuantos electrones libres, sin embargo, cuando incide luz en la superficie a una cierta longitud de onda se le proporciona a los electrones la energía suficiente para que pasen a la banda de conducción. Así, el flujo de electrones se incrementa bajando la resistencia.

El Módulo Sensor Fotorresistor LDR es un dispositivo electrónico que permite detectar la intensidad de la luz ambiental mediante una fotorresistencia (LDR, Light Dependent Resistor). Su principio de funcionamiento se basa en la variación de la resistencia eléctrica del sensor según la cantidad de luz que incide sobre él: cuando hay mucha luz,

la resistencia disminuye, y cuando hay poca luz, la resistencia aumenta. Este cambio se traduce en una señal eléctrica digital que puede ser leída por microcontroladores.

## Capítulo 3.

### Diseño del sistema.

#### 3.1 Sistematización de la estructura del almacén.

El proceso de almacén estará dividido en tres tareas principales:

- Tarea de Picking.
- Tarea de Almacenamiento.
- Tarea de Packing.

De manera general se deben de detectar acciones de Picking y Packing. Siendo prioridad la acción de Packing, es decir, si hay una orden de expedición se debe realizar primero que cualquier acción de colección.

El orden de ejecución de nuestras tareas establecidas es el siguiente

Tarea de packing:

1. Se debe de revisar si hay una orden de packing y si no existe una orden de picking o packing en curso.
2. Se debe activar la tarea de almacén para ubicar la posición del paquete a entregar
3. Se recolecta el paquete de la dirección de almacenamiento.
4. Se deposita el paquete en la zona de carga y descarga.
5. Se avanza el paquete para darle salida.

Tarea de picking:

1. Detectar paquete al llegar a zona de picking.
2. Detener banda.
3. Revisar si existen órdenes de packing en cuyo caso se realizarán dichas tareas previamente, con la finalidad de liberar espacios en la zona de almacén.
4. Activar tarea de almacén, con la finalidad de detectar espacios disponibles para almacenar paquete y reservar un espacio para la tarea de picking en curso.
5. Avanzar a la zona de carga y descarga.
6. Trasladar el paquete al espacio previamente reservado.

7. Notificar a la tarea de almacenamiento el producto almacenado en el espacio reservado.

Tarea de almacenamiento:

1. Guardar direcciones de almacenamiento de productos para la tarea de packing.
2. Reservar posiciones disponibles en paquetes para la tarea de picking.

En la zona de Picking se hará la detección de los productos que llegan para ser almacenados, el traslado del producto a la zona de Almacén en los espacios disponibles se debe llevar a cabo con un dispositivo móvil, así como también el dispositivo móvil trasladará el producto de la zona de Almacén a la zona de Packing.

Para tener un control de los productos almacenados y espacios disponibles existirá una memoria la cual guardará la dirección donde fue almacenado el producto y en donde existen espacios disponibles.

### **3.2. Necesidad de automatización.**

En base a la sistematización del proceso se dará pie a la automatización del mismo en donde se hará uso de herramientas, las cuales al operar de manera conjunta realizarán las acciones y tareas planteadas.

Automatización de la tarea de packing:

- Para la revisión si existe una orden de Packing se implementará un panel de nueve botones donde cada uno estará relacionado con cada espacio del almacén.
- Para la zona de carga y descarga se requiere de una banda transportadora donde serán llevados los paquetes para ser almacenados o entregados.
- La recolección de algún paquete llamado a zona de carga y descarga se hará por medio de un dispositivo que genere un deslizamiento planar paralelo a la estructura del almacén, además de un dispositivo que permite depositar y recoger los paquetes del almacén.
- Por medio de un sensor de presencia se detectará cuando un paquete esté listo para ser entregado.

Automatización de la tarea de picking:

- Existirá adicionalmente a la zona de carga y descarga una zona de picking la cual estará conformada por otra banda transportadora. La detección de la llegada de un paquete a la zona de picking se hará con un sensor de presencia, el cual enviará una señal de voltaje y en respuesta a ella la banda se detenga.
- La detección de espacios disponibles dentro del almacén se hará por medio de un sensor de presencia, el cual enviará una señal de voltaje a un dispositivo lógico para así poder reservar espacios disponibles para la tarea de picking.
- Para mover el paquete proveniente de la zona de picking a la zona de carga y descarga se utilizará un dispositivo que permita realizar la traslación entre las dos bandas.
- El traslado del paquete será por medio del dispositivo implementado en la zona de Packing, para la recolección de paquetes.

Automatización para la tarea de almacenamiento:

- La dirección de almacenamiento del producto se guardará en un dispositivo lógico, así como también este controlara los espacios disponibles y generará la programación del sistema.

### **3.3 Propuesta de solución a las necesidades.**

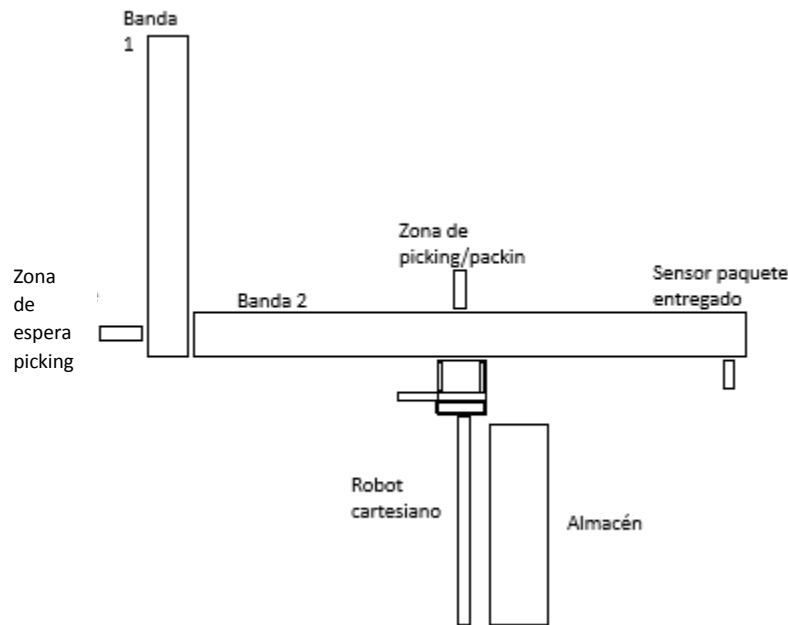
En la Figura 3.3.1 se muestra la estructura y las partes que conforman el sistema de almacén, en donde cada parte tiene su respectiva función como se describe a continuación:

- Banda 1: en esta banda se da el ingreso a paquetes al sistema de almacén, donde son transportados a la zona referenciada como zona de espera packing (parte final de la banda 1).
- Zona de espera picking: en dicha zona se tiene un sensor de tipo infrarrojo el cual detecta la presencia del paquete arribado permaneciendo indefinidamente detenido para ser trasladado a la banda 2 y posteriormente ser almacenado por medio del robot cartesiano si así el sistema ya lo

permite. Dicho traslado de banda 1 a la banda 2 se lleva a cabo por medio de un cilindro pistón el cual empuja el paquete deslizándolo en unos rodillos ubicados entre dichas bandas para así facilitar el traslado.

- Banda 2: en dicha banda se llevan a cabo las siguientes operaciones:
  - Al inicio de esta banda se tienen los paquetes trasladados de la banda 1, donde son llevados a la zona central de la banda referenciada como zona de picking y packing para ser almacenados.
  - En dicha zona también se posicionan los paquetes para ser trasladados al final de la banda y por medio de un sensor de tipo infrarrojo se nos indicara que el paquete ha sido entregado.
- Zona de picking y packing: en dicha zona se ejecutan las órdenes de empaque o desempaqué, donde los paquetes son manipulados por el robot cartesiano, es decir, en esta zona se posiciona los paquetes para ser almacenados o entregados. (sensor) que es cuando se detecta que un paquete llevo a dicha zona
- Robot cartesiano: el robot de tipo cartesiano es el encargado de ejecutar las órdenes de empaque y desempaqué. El cual recolecta los paquetes de la zona de picking/packing para almacenarlos en espacios específicos de manera lógica y también volver al paquete a la zona de picking/packing para ser entregado. Para trasladar de la zona de picking/packing específicamente a la bandeja del robot, de igual manera se tiene un cilindro pistón para el deslice del paquete a través de un sistema de rodillos entre la banda 2 y el robot cartesiano los cuales faciliten el traslado de los paquetes.
- Almacén: rack de nueve espacios en el cual se almacenan los paquetes de manera indefinida. En cada espacio la parte inferior cuenta con sensores de tipo LDR y la parte superior cuenta con lámparas con la función de indicar si un espacio se encuentra disponible o no.

- Sensor de paquete entregado: este sensor está ubicado al final de la banda 2, con la única tarea de indicar cuando un paquete ya ha sido entregado.



**Figura 3.3.1. Estructura y distribución del almacén.**

Los dispositivos electrónicos a implementar serán los siguientes:

- La acción de detección en la zona de Picking se hará con un módulo de sensor infrarrojo, ya que estos dispositivos cuentan con un emisor de señal infrarroja que al chocar con un paquete se refleja hacia el fototransistor (receptor), donde se generará una señal de voltaje con la cual se trabajara para realizar una acción.
- La acción de detección cuando un espacio en el almacén ya está ocupado o disponible se hará con un módulo de sensor LDR, acompañado de una lámpara la cual emitirá luz y por medio de ello el sensor la detectará en su superficie haciendo que su resistencia baje, enviando una señal de voltaje con la cual se asociará que este espacio está disponible. Caso contrario será cuando el sensor no detecte una incidencia de luz dado que el producto puesto en el almacén será una barrera entre el sensor y la lámpara, haciendo que su resistencia suba y de

igual manera este enviará una señal de voltaje con la cual se asociará que el espacio está ocupado.

- El traslado del producto de una zona a otra se hará con un robot tipo cartesiano el cual estará compuesto por una bandeja, que permitirá posicionar los productos al deslizarse horizontal y verticalmente sobre un par de rieles, dichos movimientos se realizarán a través de un mecanismo tornillo sin fin activados cada uno por un motor a pasos, para generar movimientos en un plano paralelo al estante del almacén.
- Para depositar el producto dentro del almacén se realizará con un mecanismo piñón cremallera, es decir este se encargará de realizar el un movimiento de profundidad, acompañado de unas bandejas en las cuales estará cargado el producto.
- Cuando un producto esté en la banda transportadora de zona de picking y pase de dicha a zona a la banda transportadora de zona de carga y descarga, se trasladará con un cilindro pistón el cual se accione por medio de una señal emitida por un sensor infrarrojo, empujando al producto de una banda transportadora a otra.
- La parte lógica de todo el sistema, es decir, el que ejecutará todas las tareas y acciones serán hechas con un controlador lógico programable (PLC), el cual será programado en lenguaje escalera.
- Las bandas transportadoras estarán conformadas por un motor de CD y un fleje de polipropileno en donde correrán los productos.

### **3.4 Estructura de prototipo para el robot cartesiano.**

En base a las necesidades presentadas en la Sección 3.3, las cuales de manera general es realizar tareas de empaque y desempaques dentro de un almacén de nueve espacios de manera precisa y automática. Se realizó la propuesta de implementar un robot de tipo cartesiano como se muestra en la Figura 3.4.1, el cual en base a sus articulaciones de tipo prismáticas permite operar con movimientos únicamente lineales a lo largo de los

ejes (X- desplazamiento horizontal), (Y- desplazamiento vertical) y (Z- desplazamiento de profundidad) .

El desplazamiento de cada uno de los ejes se logra por medio de motores a pasos los cuales generan movimientos rotatorios de manera controlada, estos movimientos rotatorios son convertidos a movimientos lineales por medio de un mecanismo piñón cremallera, posicionando así de manera precisa los ejes en puntos específicos.

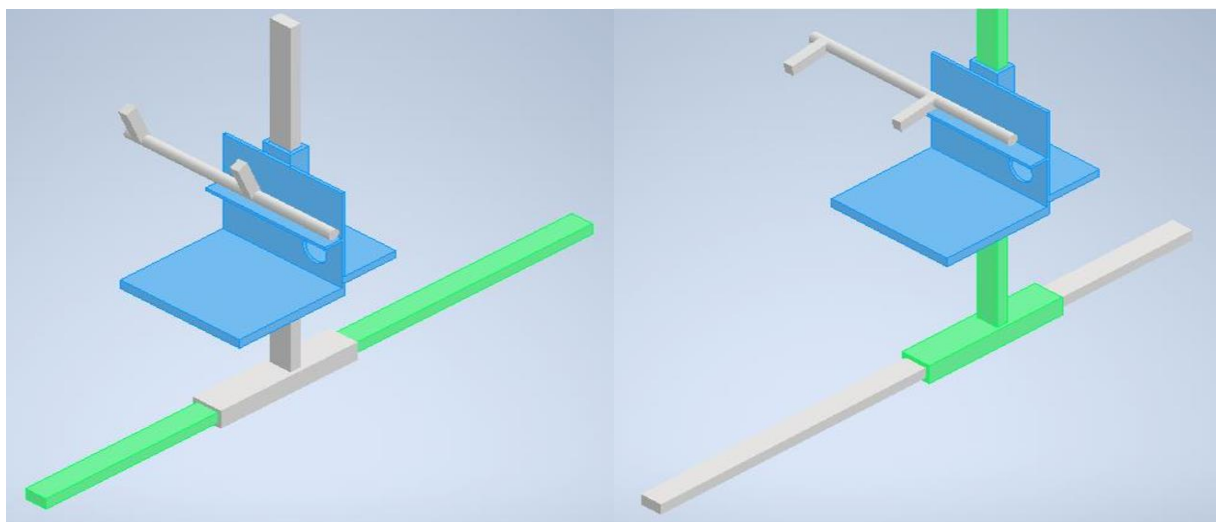
Utilizar motores a pasos tiene como ventajas tener un control más preciso de la posición y dirección sin la necesidad de implementar sistemas de retroalimentación, lo cual es crucial para las tareas que desarrollara. Por medio de un PLC y un driver se llevara a cabo el control de los motores a pasos.

La manipulación de los paquetes se realiza separando su movimiento en dos en donde el primer mecanismo permite posicionar los paquetes desde la zona de picking/packing hasta una posición paralela al espacio específico de almacenamiento, el segundo mecanismo permite introducir o retirar los paquetes del espacio específico de almacenamiento.

- El mecanismo de la bandeja estará montado sobre el eje que realiza los movimientos verticales el cual finalmente se encargara de posicionar el paquete de manera paralela al almacén en un espacio específico.
- El mecanismo de sujeción presenta una estructura simple que permite jalar o empujar los paquetes dentro del espacio de almacenamiento, el principio de giro permite sujetar o liberar el paquete mientras que el desplazamiento lineal permite introducir o sacar el paquete. Esta configuración se contempló junto con otras alternativas de sujeción principalmente de tipo pinza pero se optó por esta configuración al contar con una estructura más esbelta que permitía operar en un espacio más pequeño de almacenamiento sin obstaculizar su avance con el propio espacio de almacenamiento.

La justificación del por qué se optó por un robot de tipo cartesiano son las siguientes:

- Sus movimientos son meramente rectilíneos, lo cual facilita la programación y el cálculo de sus trayectorias.
- Tiene una mayor exactitud en cuanto al trayecto a las posiciones establecidas, así como tener repetibilidad lo cual permite volver a una posición varias veces sin errores.
- Los robots cartesianos cuentan con una menor complejidad mecánica, ya que el uso del piñón cremallera es un mecanismo sencillo y de fácil implementación, así como un bajo costo de fabricación.
- En comparación a robots de tipo SCARA:
  - El de tipo cartesiano cuenta con menos grados de libertad, articulaciones lo cual reduce su mantenimiento mecánico
  - Utilizar un robot de tipo cartesiano puede reducir espacios debido que el trabajo de estos son de manera paralela al almacén con un volumen de trabajo menor, que a diferencia si se implementara uno de tipo SCARA estos operan con movimientos rotacionales generando áreas de barrido circular, complicando la optimización de los espacios ya que ocupan volúmenes de trabajo más amplias.
  - La cinemática de los robots cartesianos son menos complejas y no demandan cálculos avanzados.
  - Al no necesitar de movimientos con ángulos, implementar un robot de tipo cartesiano resulta justo ya que implementar uno de tipo SCARA estaría sobrado e innecesariamente utilizado.



***Figura 3.4.1. Prototipo del robot cartesiano.***

## **Capítulo 4.**

### **Especificaciones de la propuesta de diseño.**

#### **4.1 Especificaciones de los elementos propuestos.**

##### **4.1.1 Características de los PLC's.**

Se consideraron dos marcas de PLC's, como lo es PLC Delta y el PLC LOGO!8 de Siemens ya que con ambos se tiene una familiaridad en su manera de operación y programación, además de haber considerado el precio de cada uno de ellos y sus características las cuales se muestran a continuación:

PLC LOGO!8 de Siemens [27].

- Número de entradas: 8.
- Número de salidas: 4 de tipo Relay.
- Voltaje de entrada (suministro): 120V/AC.
- Corriente continua: 10A con carga resistiva y 3A con carga inductiva.
- Cable de programación: Ethernet.
- Memoria máxima de programa: 400 bloques.
- Software de programación: LOGO! Soft Comfort Versión 8.

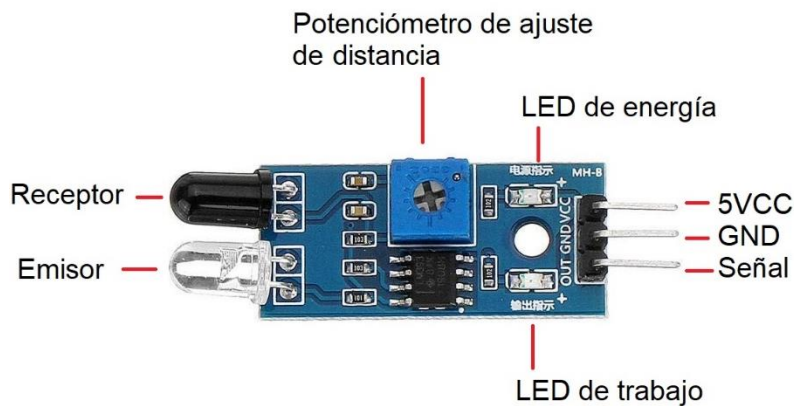
PLC Delta DVP-SX2 [28].

- Número y tipo de entradas: digitales (8) y analógicas (4).
- Número y tipo de salidas: analógico (2) y estado sólido (6).
- Voltaje de la fuente: 24 VCC.
- Registro de datos: 10,000 palabras.
- Capacidad del programa: 16,000 pasos.
- Cable de programación: mini USB, RS232 y RS485.
- Software de programación: ISPSoft.

### 4.1.2 Características del módulo de sensor infrarrojo.

En la selección de este sensor se consideró la familiarización que se tiene con él, el costo de este así como las características las cuales son adecuadas para implementarlo en el proyecto, las cuales son [29]:

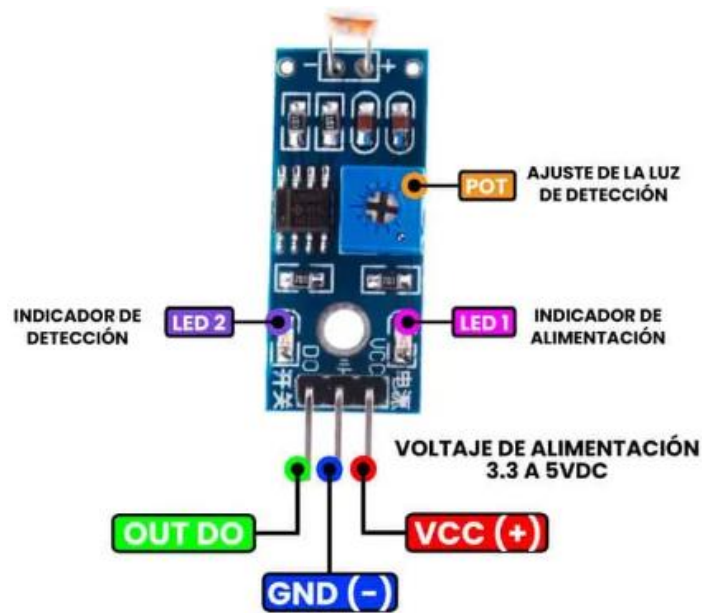
- Rango de detección: 2 cm a 30 cm.
- Voltaje de trabajo: 3V a 5V.
- Consumo de corriente: 23 mA (3.3 V) y 43 mA (5 V).
- Ángulo de cobertura: 35°.
- Sensibilidad: ajustable por medio del potenciómetro.
- Circuito de detección: basado en el LM393 y tecnología IR.
- Tipo de salida: digital (0V o 5V).
- Precio: ronda en los \$16.



**Figura 4.1** Módulo de sensor infrarrojo.

### 4.1.3 Características del módulo de sensor LDR.

En la selección de este sensor de la misma manera se consideró la familiarización que se tiene con él, el costo de este así como las características las cuales son adecuadas para implementarlo en el proyecto, las cuales son [30]:



*Figura 4.2 Modulo de sensor LDR.*

- Voltaje de trabajo: 3.3V a 5V.
- Tipo de salida: digital.
- Circuito de detección: basado en el LM393.
- Consumo de corriente: 15mA.
- Precio: ronda en los \$15.

#### **4.1.4 Características del motor paso a paso.**

Para la selección del motor paso a paso se consideraron dos motores como lo fue el nema 23 2Nm y el nema 17 0.59Nm, donde cada uno ofrece distintas prestaciones de uso, sus características principales son:

Nema 23 2Nm [31]:

- Par de sujeción: 2Nm
- Corriente por fase: 3.0A
- Resistencia por fase: 1 ohm.
- Ángulo por paso: 1.8°
- Inercia del rotor: 480 gfc<sup>2</sup>

- Peso: 1.15 kg
- Largo eje: 76 mm
- Diámetro eje: 8mm

Nema 17 0.59Nm [32]:

- Voltaje nominal: 3.1 V
- Par de sujeción: 0.59Nm
- Ángulo de paso: 1.8°
- Corriente por fase: 1,7<sup>a</sup>
- Largo eje: 20 mm
- Diámetro eje: 5mm
- Resistencia por fase: 1.8 ohm.
- Precisión en el paso: 5%.

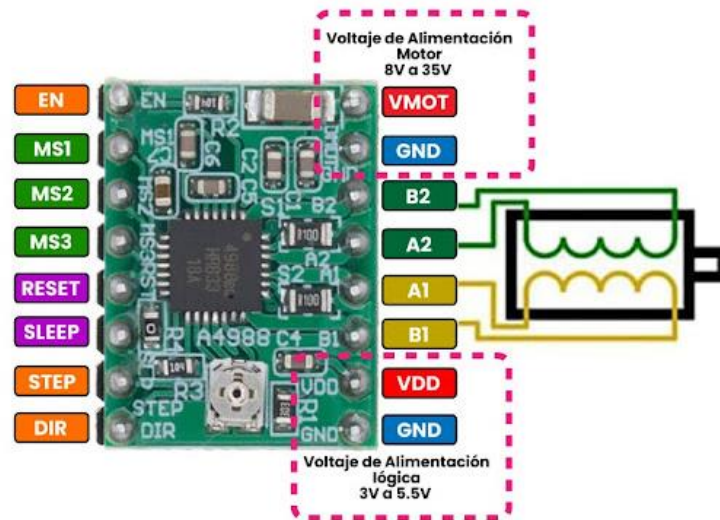
#### **4.1.5 Características del Driver A4988.**

Como se menciona en [33] el Driver A4988 está destinado para controlar solo motores a pasos, con este dispositivo podremos tener un control preciso para los movimientos de nuestro robot tipo cartesiano, es decir, este dará las señales acondicionadas a nuestro motor provenientes del PLC sabiendo cuántos pasos debe recorrer en horizontal y vertical, así como saber el sentido de giro según la tarea a realizar.

Las características eléctricas de dicho Driver son las siguientes:

- Voltaje de operación: 8V-35V.
- Voltaje de control lógico: 3.3V-5V DC.
- Corriente máxima por bobina: 2A.
- Resolución de pasos:  $\frac{1}{2}$ ,  $\frac{1}{4}$ ,  $\frac{1}{8}$  y  $\frac{1}{16}$  según su conexión.

En la figura 4.3 se muestra el Pinout del driver:



**Figura 4.3. Módulo de driver A4988**

Donde cada Pin significa lo siguiente:

- Vmot: (VIN) Pin de voltaje de entrada para la alimentación externa del motor a pasos.
- GND: Pines de tierra las cuales son compartidas.
- VDD: Pin de voltaje de alimentación del driver desde 3V a 5.5V.
- 1A, 1B y 2A, 2B: Pines de salida de corriente para el motor. Controlan la dirección y movimiento.
- MS1, MS2 y MS3: Pines de configuración para la resolución del microstepping.
- Reset: Pin para restablecer el estado del driver.
- Slepp: Pin para poner el driver en modo de bajo consumo cuando no esté en uso.
- Enable: Pin para habilitar o deshabilitar la salida del driver.
- Dir: Pin indicador de la dirección del movimiento del motor.
- Steep: Pin generador de pulsos para controlar el movimiento del motor.

#### **4.1.6 Características del cilindro pistón.**

Se seleccionó este dispositivo neumático ya que se buscaba un movimiento lineal el cual realizará el empuje del producto de una zona a otra, tomando en cuenta la fuerza

necesaria sin llegar a romper o fracturar el mismo. Así como se consideró por la sencillez de operación y las características que nos brinda como lo son las siguientes [34]:

- Fluido: aire comprimido filtrado
- Presión máxima de funcionamiento: 1000 kPa (10 bar)
- Diámetro del embolo: 20 mm
- Diámetro del vástago: 8 mm
- Carrera máxima 50 mm
- Fuerza de avance con 600 kPa (6 bar): 139 N
- Fuerza mínima del muelle de reposición del muelle: 13.6 N
- Conexión: QS-G1/8-4 para tubo flexible PUN 4x0.75

## **4.2 Caracterización de las señales de entrada y salida de los distintos componentes.**

- El sensor infrarrojo en su salida tiene un voltaje de 5V DC, cuya señal si bien constituye una salida para el sensor también es una señal que será una entrada para el PLC.
- Para el sensor LDR se utilizara un alimentación de voltaje de 5V DC dentro de un cierto circuito resistivo, con lo que se espera que el voltaje a la salida de dicho circuito varíe en un rango de 0V a 5V, esta señal de salida también constituye una salida de entrada para el PLC.
- El motor a pasos permitirá obtener una salida mecánica para lo cual requiere de dos señales de entrada, la primera será una señal de potencia de DC de 20V y la segunda será una señal de control tipo pulsante de voltaje máximo de 24V.
- El cilindro pistón requiere una alimentación de aire comprimido de 5 bar adicionalmente para su operación requiere trabajar con una válvula de alimentación de accionamiento eléctrico con una alimentación de 24V DC.
- Por otra parte el PLC a sus entradas digitales requiere de un voltaje de 24V DC.

## 4.3 Elementos de interfaz y acondicionamiento de las señales entre etapas.

### 4.3.1 Módulo de relevador.

Como se mencionó anteriormente el PLC requiere de señales de 24V DC a sus entradas, la problemática surge por parte de los sensores infrarrojos y LDR puesto que ofrecen a su salida señales de 5V DC y es por ello que se tiene la necesidad de implementar un elemento de interfaz que permitan ser controlados con una pequeña señal de voltaje y esta a su vez controle acciones que requieren de un mayor consumo de voltaje.

Para solución de dicha problemática se implementará un módulo de relevador de un canal por cada sensor como el que se muestra en la Figura 4.4.



**Figura 4.4. Módulo relevador de 1 canal.**

El fundamento de este módulo es un relevador el cual es un dispositivo electromagnético que sirve como interruptor eléctrico, es decir, permite el paso de la corriente por un circuito o no y a diferencia de los interruptores mecánicos accionados de manera manual, los relevadores se activan mediante una señal eléctrica enviada a la bobina del relevador. Cuando la bobina es activada por una corriente necesaria de operación se produce un campo electromagnético haciendo que el contacto del relevador normalmente abierto se

cierre y permita el paso de la corriente por un circuito, caso opuesto cuando dejamos de activar la bobina el campo electromagnético se pierde y el contacto del relevador se abre cortando el paso de la corriente [35].

En este caso la señal de control para el relevador será la señal proporcionada por los sensores infrarrojos los cuales darán la alimentación necesaria para que la bobina del relevador sea activada o desactivada.

Sus características son las siguientes [36]:

- 1 canal accionado por corriente con circuito de control por transistor.
- Relevador (Relay) de 1 polo 2 tiros
- El voltaje de la bobina del relevador es de 5 VDC
- Led indicador para cada canal (enciende cuando la bobina del relevador está activa)
- Activado mediante corriente: el circuito de control debe proveer una corriente de 15 a 20 mA
- Puede ser controlado directamente por circuito lógicos
- Terminales de conexión de tornillo
- Terminales de entrada de señal lógica con cabezales macho de 0.1"

Así como sus especificaciones de operación son las siguientes:

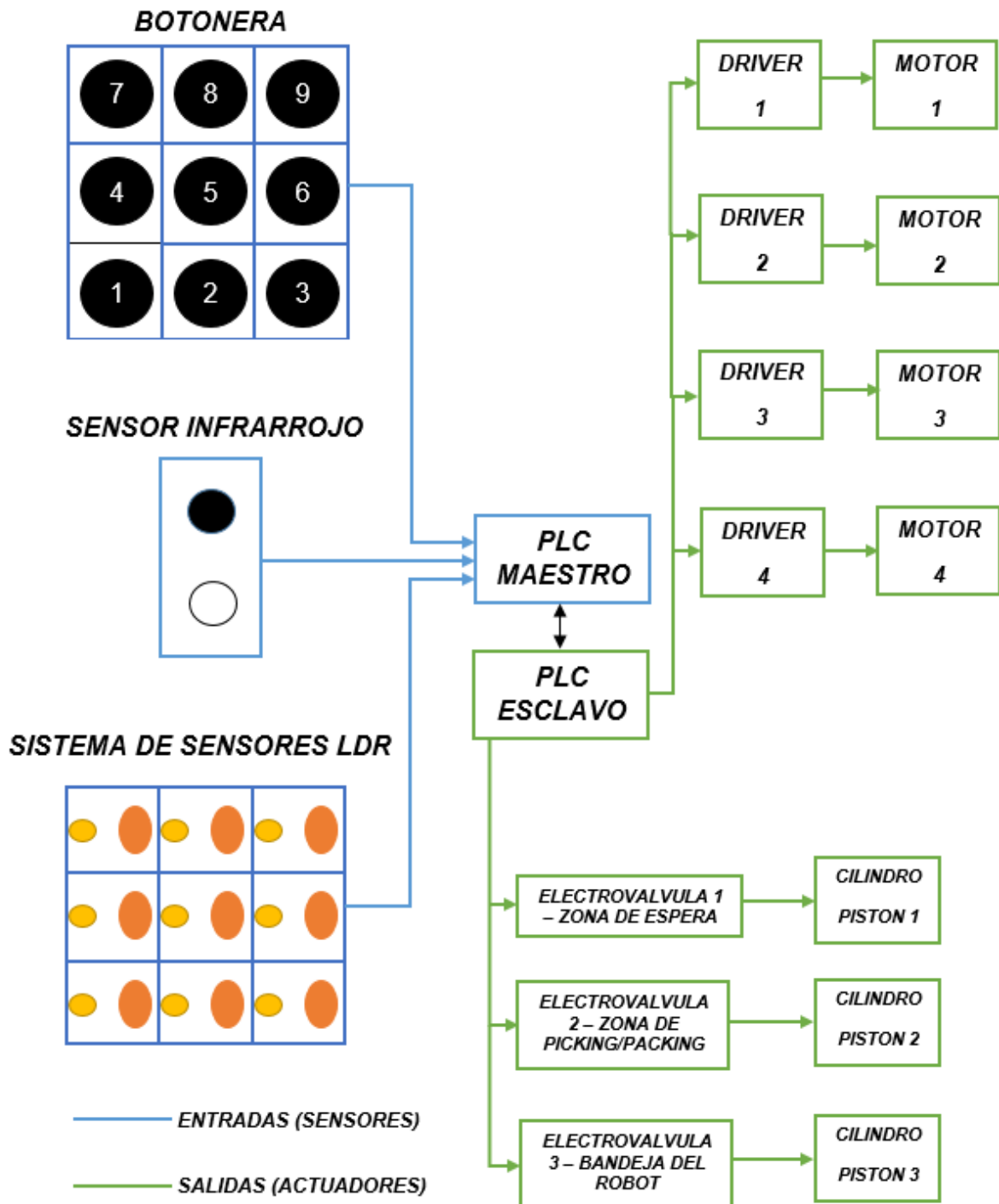
- Carga máxima: AC 250V / 10A, DC 30V / 10A
- Corriente de activación: 5 mA
- Voltaje de funcionamiento: 5V
- DC +: fuente de alimentación positiva (VCC)
- DC -: fuente de alimentación negativa (GND)
- IN: puede ser relé de control de nivel alto o bajo
- NO: interfaz de relé normalmente abierta
- COM: Relés de interfaz común
- NC: interfaz de relé normalmente cerrada

### 4.3.2 Electroválvula.

Los actuadores neumáticos tipo cilindro pistón, los cuales son conectados a las salidas del PLC requieren al igual un acondicionamiento de señal la cual se obtiene mediante una electroválvula, la cual permite activar o desactivar el accionamiento de los cilindros mediante el cambio de posición de la válvula, este cambio se consigue mediante una señal eléctrica proveniente del PLC, para el caso específico de este proyecto se plantea el uso de la válvula 3/2 de Festo con las siguientes características [37]:

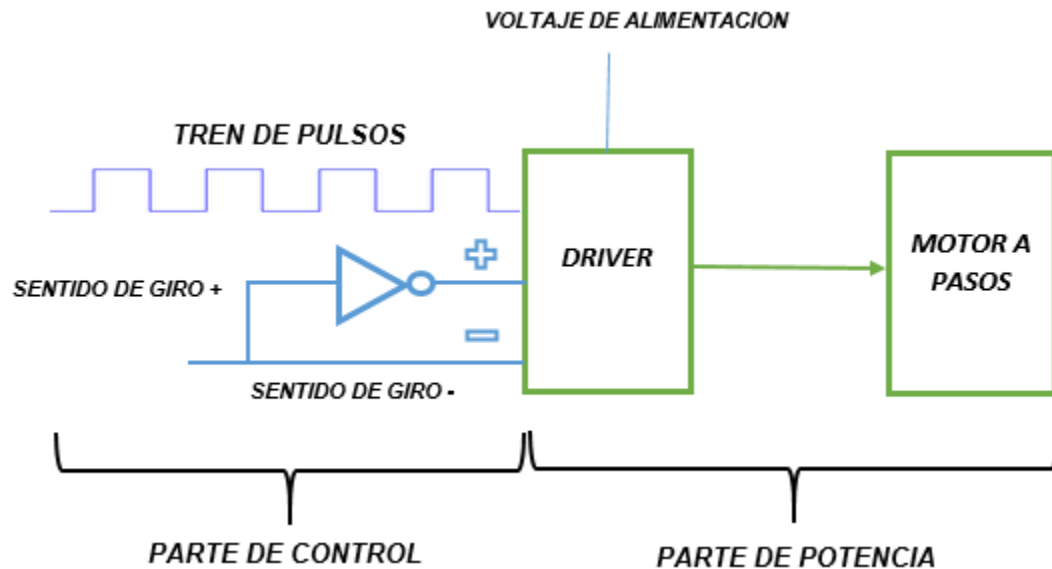
- Voltaje de funcionamiento: 24 VDC
- Conexión: bornes para conectores de seguridad de 4 mm
- Margen de temperatura ambiente: -5 a 60°C
- Fluido: aire comprimido filtrado
- Caudal nominal: 150l/min
- Margen de presión: 150 kPa – 800 kPa (1.5 bar – 8 bar)
- Tiempo de respuesta con 600 kPa (6 bar):
  - Conexión: 6 ms
  - Desconexión: 16 ms

#### 4.4 Mapa de entradas y salidas (I/O).



Para el acondicionamiento de las señales de entrada al PLC por parte de los sensores, así como la adecuación de los voltajes de salida a los drivers y electroválvulas es necesaria la implementación del modulo de relevador de la Sección 4.3.1.

#### 4.5 Diagrama de potencia y control de los motores a pasos.



## Capítulo 5.

### Algoritmos y programas.

Para el programa en PLC se utilizó un software en línea llamado PLC Fiddle, el cual es programado solo en lenguaje escalera ya que es de los más comunes y más sencillos de entender al momento de programar y leer ya que este se basa en dispositivos eléctricos como contactos de relevadores y bobinas. A continuación en la siguiente Tabla 5.1 se muestran las herramientas que se utilizaron dentro del programa, así como sus símbolos y funciones.

**Tabla 5.1. Herramientas utilizadas en PLC Fiddle.**

| Herramienta.                  | Función.                                                                                                                                                                                                                                                                                                                                                                                                | Símbolo en PLC Fiddle.                                                                | Tipo de elemento. |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|-------------------|
| Contacto normalmente abierto. | Este no permite el paso de la corriente en su estado normal, a menos que una señal enviada le active cerrando así el circuito.                                                                                                                                                                                                                                                                          |    | Entrada           |
| Contacto normalmente cerrado. | Este permite el paso de la corriente en su estado normal, a menos que una señal enviada le desactive abriendo así el circuito.                                                                                                                                                                                                                                                                          |   | Entrada           |
| Bobina.                       | Está representa una salida del PLC y pueden ser dispositivos como lámparas, motores, válvulas neumáticas etc. Con esta herramienta también es posible crear memorias almacenando temporalmente señales de entrada o salida según sea el propósito del programa.                                                                                                                                         |  | Salida            |
| Temporizador on-delay         | Este retarda la activación de una salida en un cierto tiempo deseado. Activándose por medio de una señal de entrada. Se puede configurar por medio de su PRE-Preset (valor preestablecido), es aquí donde se coloca el valor deseado al cual llegará el conteo del temporizador, así como también tiene un ACC-Acumulated (valor acumulado), este registra el conteo que va realizando el temporizador. |  | Salida            |

Dentro de este se tienen 3 señales con las cuales se pueden trabajar según el propósito del programa, las cuales son :

- EN (Enable/Habilitación): esta señal indica que el temporizador está activado, comenzando así el conteo.
- TT (Timer Timing/Contando tiempo): esta señal está activada mientras el temporizador esté contando.
- Q (Output/Salida): esta señal es activada cuando el temporizador ha llegado a su valor deseado.

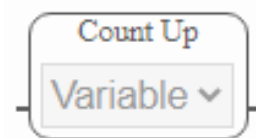
Este contiene un acumulador el cual va incrementando cada vez que recibe una señal de entrada.

Se puede configurar por medio de su PRE-Preset (valor preestablecido), es aquí donde se coloca el valor deseado al cual llegará el conteo, así como también tiene un ACC-Acumulated (valor acumulado), este registra el conteo que se va realizando.

Contador up

Dentro de este se tienen 2 señales con las cuales se pueden trabajar según el propósito del programa, las cuales son :

- EN (Enable/Habilitación): esta señal indica que el contador se está activando, incrementando así el ACC.
- Q (Output/Salida): esta señal es activada cuando el contador ha llegado a su valor deseado.



Salida

Reset

Este reinicia los valores acumulados dentro de un temporizador o un contador



Salida

El programa para el sistema de almacenamiento se diseñó en dos programas en PLC, los cuales estarán interconectados. Donde uno cumple la función de la lógica del sistema,

es decir, enviará las instrucciones al otro programa para realizar tareas de empaque y desempaque, donde entonces el otro programa de ejecución recibiría dichas instrucciones generando así los pulsos para los motores a pasos según sea la instrucción.

Dicho de otra forma el programa de lógica estará relacionado con los sensores que son nuestras entradas del sistema y el programa de ejecución estará relacionado con los motores que son las salidas del sistema.

Se desarrollaron tres algoritmos con los cuales se trabajaron para poder desarrollar el programa en el lenguaje escalera para el PLC. Cada algoritmo realiza distintas tareas, a continuación se muestran y se explican el funcionamiento de cada uno de ellos.

Se realizaron tablas de verdad para visualizar de manera lógica las posibles combinaciones, analizando el comportamiento del sistema para tareas de desempaque.

En la Tabla 5.2 se muestran las combinaciones posibles para activar una tarea de desempaque específica. Donde Botón N representa a algún botón de los nueve botones asociados a una orden de tarea de desempaque, Tarea en espera N representa una orden almacenada en una memoria en espera para ser ejecutada, Desempaque N las nueve posibles tareas de desempaque a realizarse y Desempaque es una tarea general que indica que existe una tarea en ejecución, la cual se activa a su vez con cualquiera de los nueve desempakes (como se muestra en la Tabla 5.3) y por lo tanto no permite sobrescribir una tarea de desempaque adicional. En este caso, una tarea de Desempaque N se activara siempre y cuando exista la orden proveniente del Botón N o una orden en espera (Tarea en espera N) guardada en la primer memoria y que no exista algún desempaque en curso.

**Tabla 5.2. Tabla de verdad para activar tarea de Desempaque N.**

| Botón                | Desempaque | Desempaque N |
|----------------------|------------|--------------|
| N/Tarea<br>en espera |            |              |
| N                    |            |              |
| 1                    | 1          | 0            |
| 1                    | 0          | 1            |
| 0                    | 1          | 0            |
| 0                    | 0          | 0            |

En la Tabla 5.3 se muestra cómo la tarea general de desempaque se activa con la ejecución de cualquier tarea de desempaque específica y se desactiva al término de la misma.

**Tabla 5.3 Tabla de verdad para activar tarea general de Desempaque.**

| Desempaque N | Desempaque |
|--------------|------------|
| 1            | 1          |
| 0            | 0          |

En la Tabla 5.4 se muestran las combinaciones posibles para activar una Tarea en espera específica, es decir, los requisitos para escribir una orden de espera específica en la primera memoria. Tarea en espera es una instrucción que nos indica la disponibilidad de dicha memoria para almacenar una tarea en espera N, generando una cola de seguimiento en caso de que se desee ejecutar una tarea de desempaque específica al terminar una que este en curso de ejecución, Tarea en espera AN será una tarea de desempaque específica que se desea almacenar en la primer memoria al liberarse la misma. En este caso, una Tarea en espera N se activará siempre y cuando exista la orden proveniente del Botón N o una tarea en espera de almacenarse en la

primera memoria (Tarea en espera AN), exista algún desempaque en curso y no exista ya una Tarea en espera guardada para ser ejecutada.

**Tabla 5.4. Tabla de verdad para activar Tarea en espera N.**

| Botón N/<br>Tarea en<br>espera AN | Desempaque | Tarea en espera | Tarea en espera N |
|-----------------------------------|------------|-----------------|-------------------|
| 1                                 | 1          | 1               | 0                 |
| 1                                 | 1          | 0               | 1                 |
| 0                                 | 1          | 1               | 0                 |
| 0                                 | 1          | 0               | 0                 |

En la Tabla 5.5 se muestra cómo el escribir una tarea en espera N, llena la primera memoria y esta se libera solo cuando dicha orden se desactiva al pasar a un proceso de ejecución.

**Tabla 5.5 Tabla de verdad para activar tarea general de Tarea en espera.**

| Tarea en<br>espera N | Tarea en<br>espera |
|----------------------|--------------------|
| 1                    | 1                  |
| 0                    | 0                  |

En la Tabla 5.6 se muestran las combinaciones posibles para activar una Tarea en espera AN (específica). Donde Tarea en espera A es una instrucción que nos indica la disponibilidad en una segunda memoria, esta segunda memoria se formara en la cola de la primer memoria para poder ser ejecutada una vez se haya realizado el desempaque en curso y la tarea en espera N que se había escrito en la primer memoria. En este caso, una Tarea en espera A-N se activara siempre y cuando exista la orden proveniente del Botón N, que exista algún desempaque en curso, que exista una Tarea en espera y que no exista ya una Tarea en espera A guardada para ser ejecutada.

**Tabla 5.6. Tabla de verdad para activar Tarea en espera A-N.**

| Botón N | Desempaque | Tarea en<br>espera | Tarea en espera<br>A | Tarea en espera<br>A-N |
|---------|------------|--------------------|----------------------|------------------------|
| 1       | 1          | 1                  | 1                    | 0                      |
| 1       | 1          | 1                  | 0                    | 1                      |
| 0       | 1          | 1                  | 1                    | 0                      |
| 0       | 1          | 1                  | 0                    | 0                      |

En la Tabla 5.7 se muestra cómo el escribir una tarea en espera AN, llena la segunda memoria y esta se libera solo cuando dicha orden se desactiva al pasar a la primera memoria.

**Tabla 5.7 Tabla de verdad para activar tarea general de Tarea en espera A.**

| Tarea en<br>espera A-N | Tarea en espera<br>A |
|------------------------|----------------------|
| 1                      | 1                    |
| 0                      | 0                    |

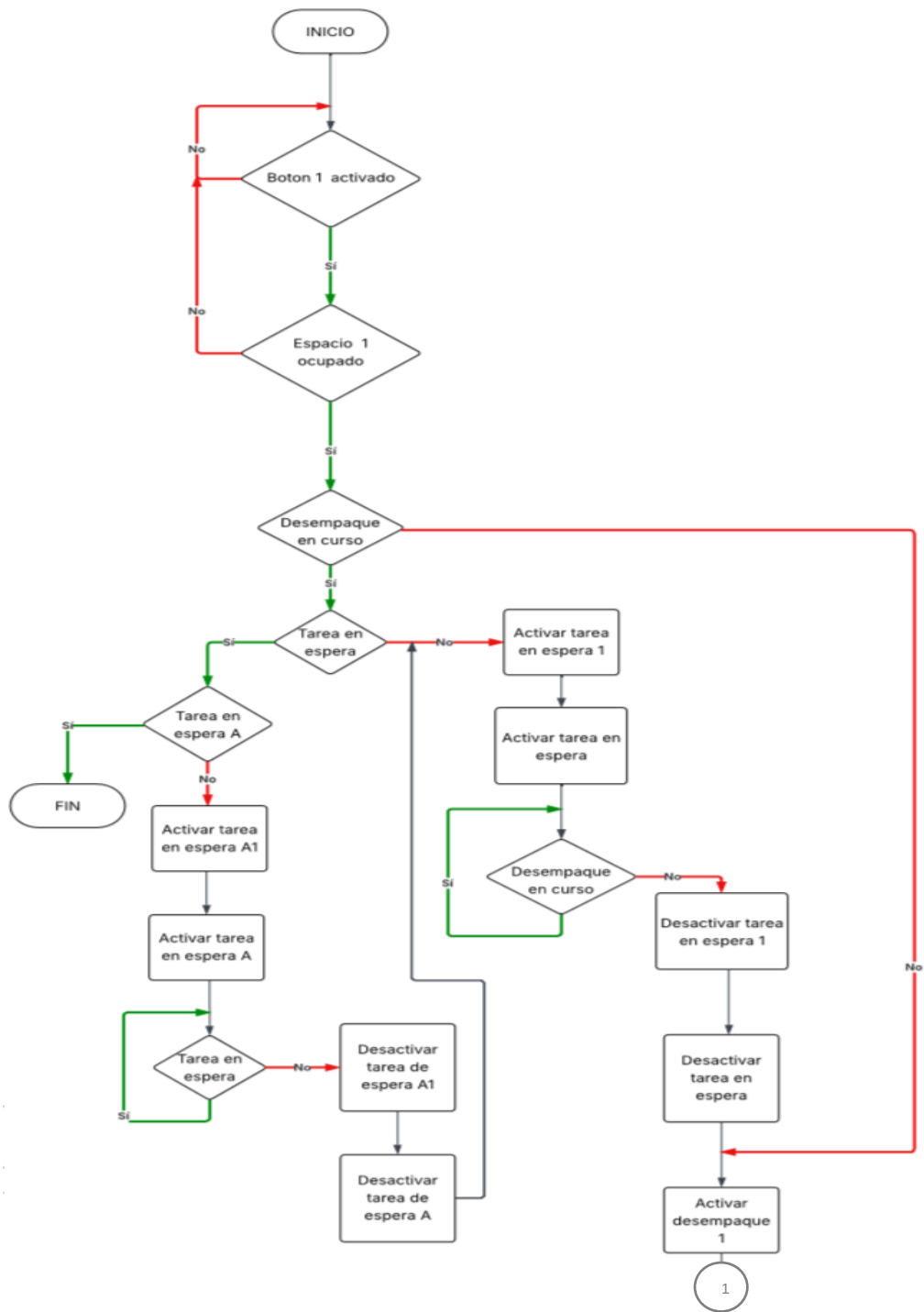
## 5.1 Algoritmo para la lógica del sistema (Parte I).

- Para la parte lógica del programa se establecieron condiciones de prioridad y de memoria, la prioridad la tienen las tareas de desempaque.
- Existen para las acciones de desempaque un conjunto de memorias, denominadas tarea en espera y tarea en espera A, en cuanto a las tareas de empaque no existen memorias, pues los paquetes se atienden según su llegada y la disponibilidad de espacios en almacén, es por ello que las tareas de desempaque tienen prioridad, con el objetivo de liberar espacios, adicionalmente las tareas de empaque tienen la restricción de esperar la liberación de espacios para poder activarse.

- Para la realización de las tareas de desempaque, estas se llevarán a cabo bajo el llamado del paquete solicitado, mediante un conjunto de botones, asociados a la posición donde se ubican el paquete de interés, siempre y cuando exista un paquete en la ubicación solicitada (la información sobre la existencia de paquetes en las posiciones está determinada por la señal de un sensor tipo LDR). En caso de que exista un paquete en la posición solicitada y siempre y cuando no exista un desempaque en curso, se activará una tarea de desempaque específica, indicada como desempaque  $n$ , con  $n$  igual al número de botón activado, esta acción es clave porque se asocia con una rutina de movimiento específica para el robot, como se indica en el algoritmo de las Figuras 5.1, 5.2 y 5.3. Adicionalmente se activará una tarea general denominada desempaque en curso, la cual permitirá indicar que hay una acción de desempaque realizándose en ese momento y bloqueando de este modo la posibilidad de activar algún otro empaque, hasta que esta tarea se concluya, mediante una señal del algoritmo asociado a el movimiento de motores que activará la tarea de desempaque especificada por la botonera, así como la tarea general de desempaque en curso, para permitir el inicio de otra posible rutina.
- Cuando una tarea de desempaque no pueda ser ejecutada, porque exista una tarea de desempaque en curso, se procederá a almacenar la instrucción en una memoria denominada tarea en espera  $n$  (con  $n$  alusiva a el botón de activación), siempre que no exista una tarea en espera almacenada, en este caso se procederá a guardar la memoria y a activar la tarea general denominada tarea en espera, posteriormente se esperará a que la tarea de desempaque en curso termine de ejecutarse, para activar la tarea de desempaque guardada en la memoria y liberar la memoria desactivando la tarea en espera  $n$  específica y la tarea en espera general, regresando al ciclo de ejecución descrito en el párrafo anterior.
- Adicionalmente en caso de que si exista una tarea en espera almacenada en la primer memoria, se procederá a almacenar en una segunda memoria la tarea de

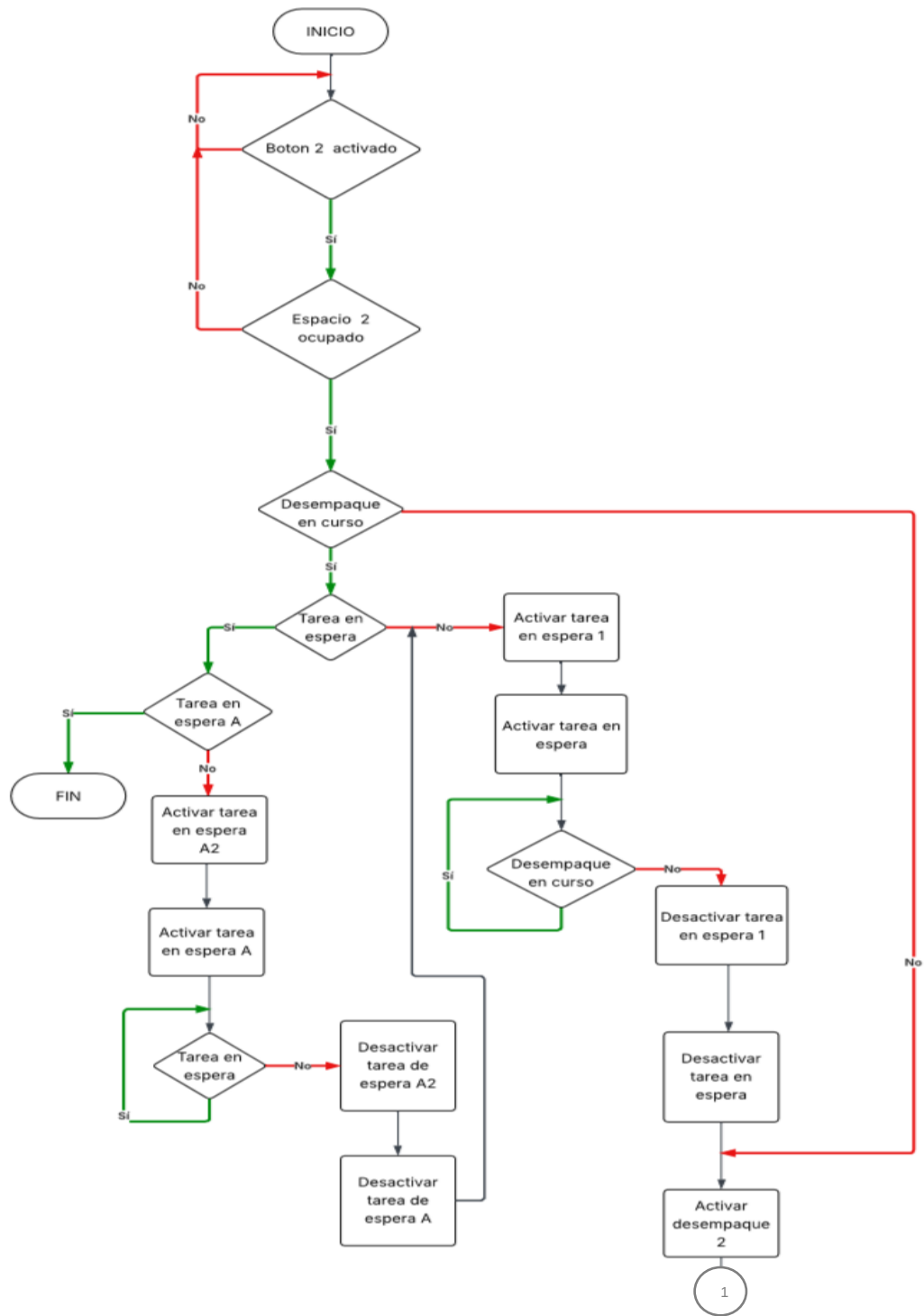
desempaquetar especificada por la botonera, esta memoria denominada tarea en espera A n, funcionara de forma análoga al caso de la primer memoria, activando la tarea general tarea en espera A y esperando la desactivación de la tarea en espera, para avanzar a la primer memoria, procediendo al avanzar a la primer memoria a liberarse la segunda memoria, repitiendo el proceso descrito anteriormente.

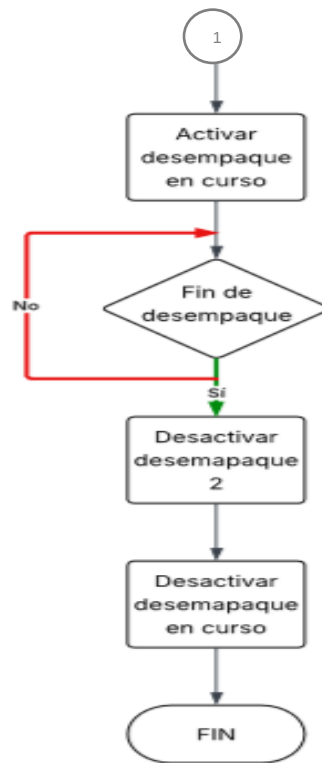
- En caso de que exista una tarea en espera A almacenada en la segunda memoria, no será posible almacenar más instrucciones para este programa específico, pero se podría replicar este proceso si se desea generar una cola más grande de memorias.



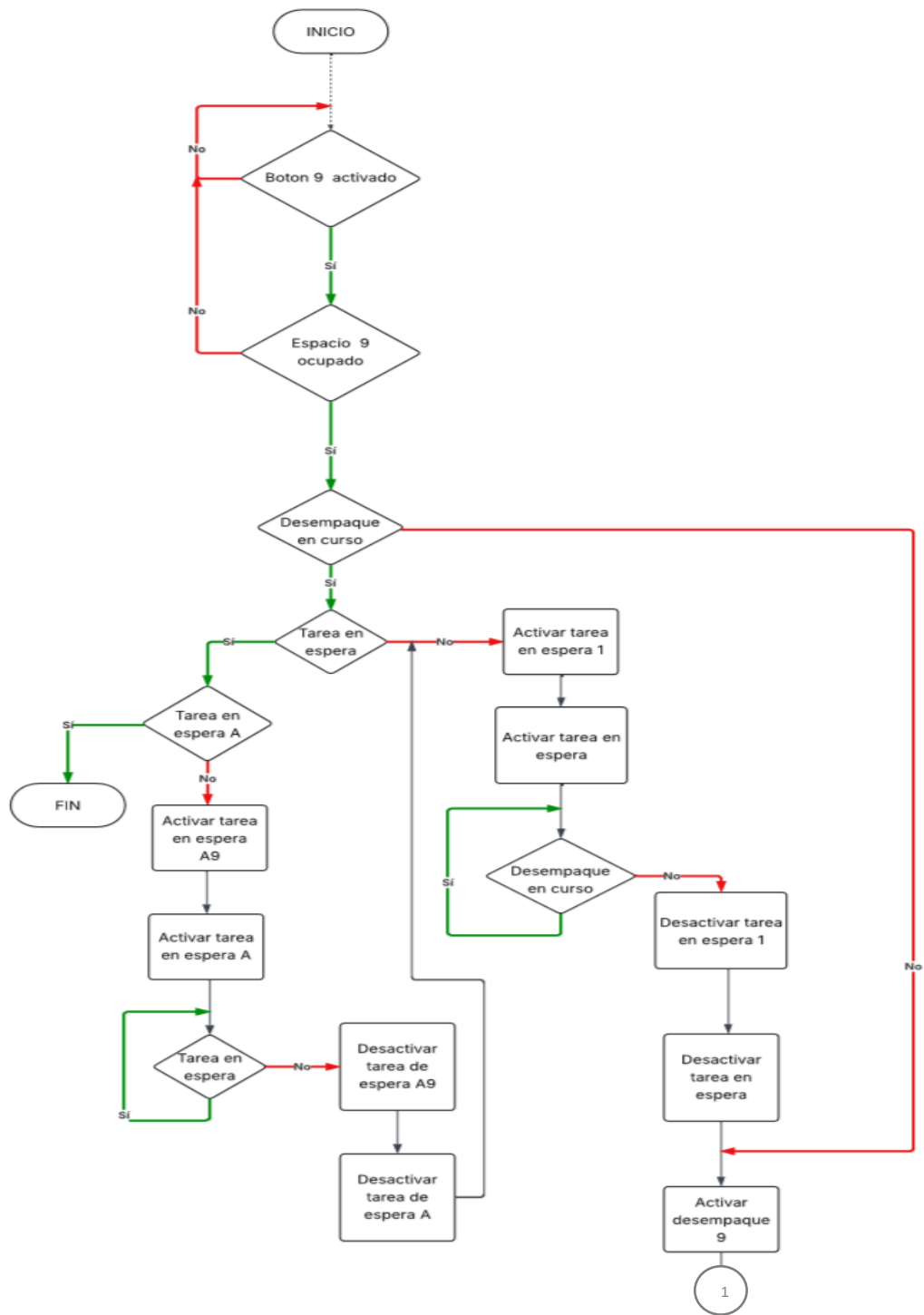


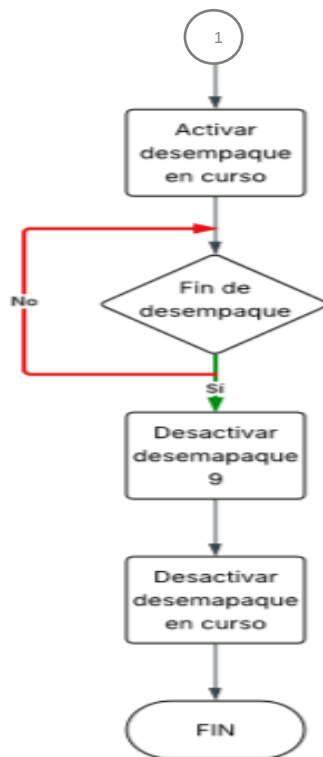
**Figura 5.1. Algoritmo de lógica parte I – Botón 1.**





**Figura 5.2. Algoritmo de lógica parte I – Botón 2.**





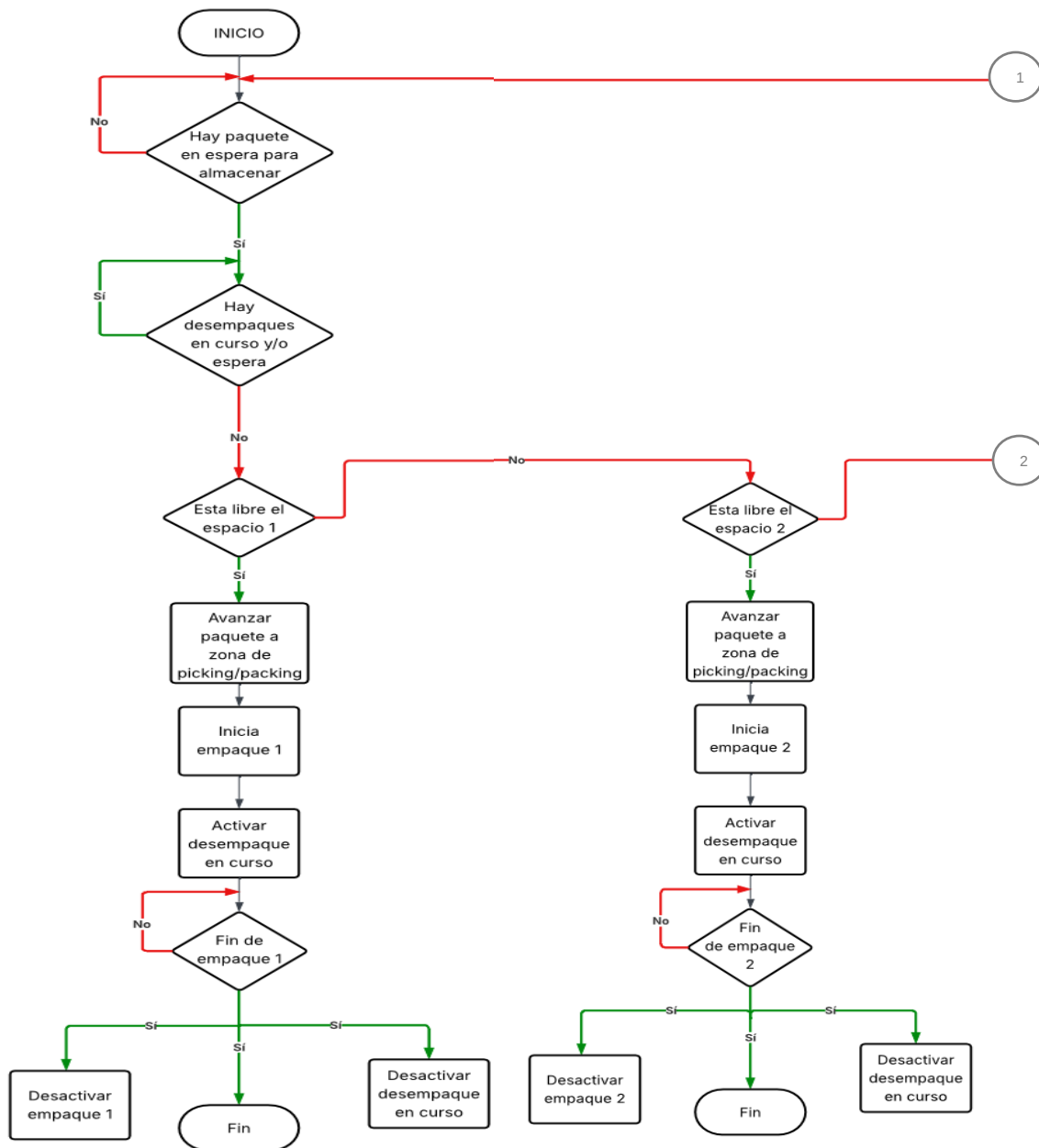
**Figura 5.3. Algoritmo de lógica parte I – Botón 9.**

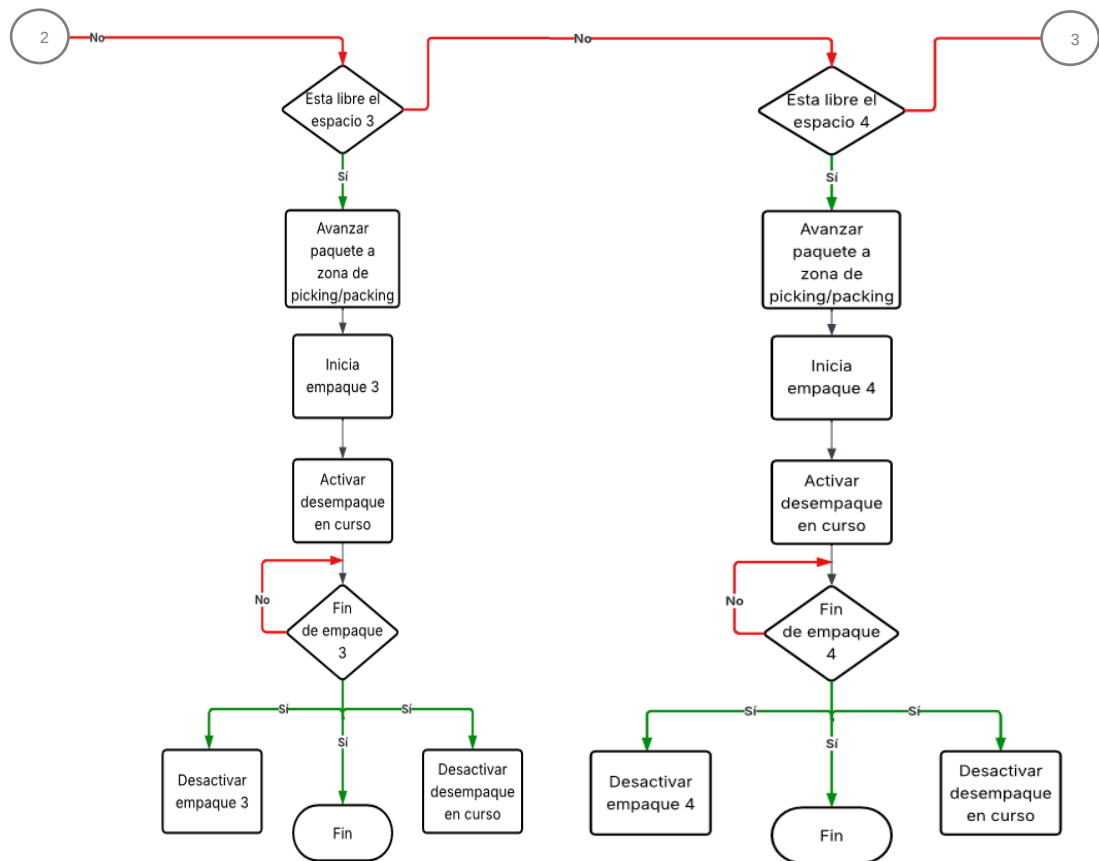
## **5.2 Algoritmo para la lógica del sistema (Parte II).**

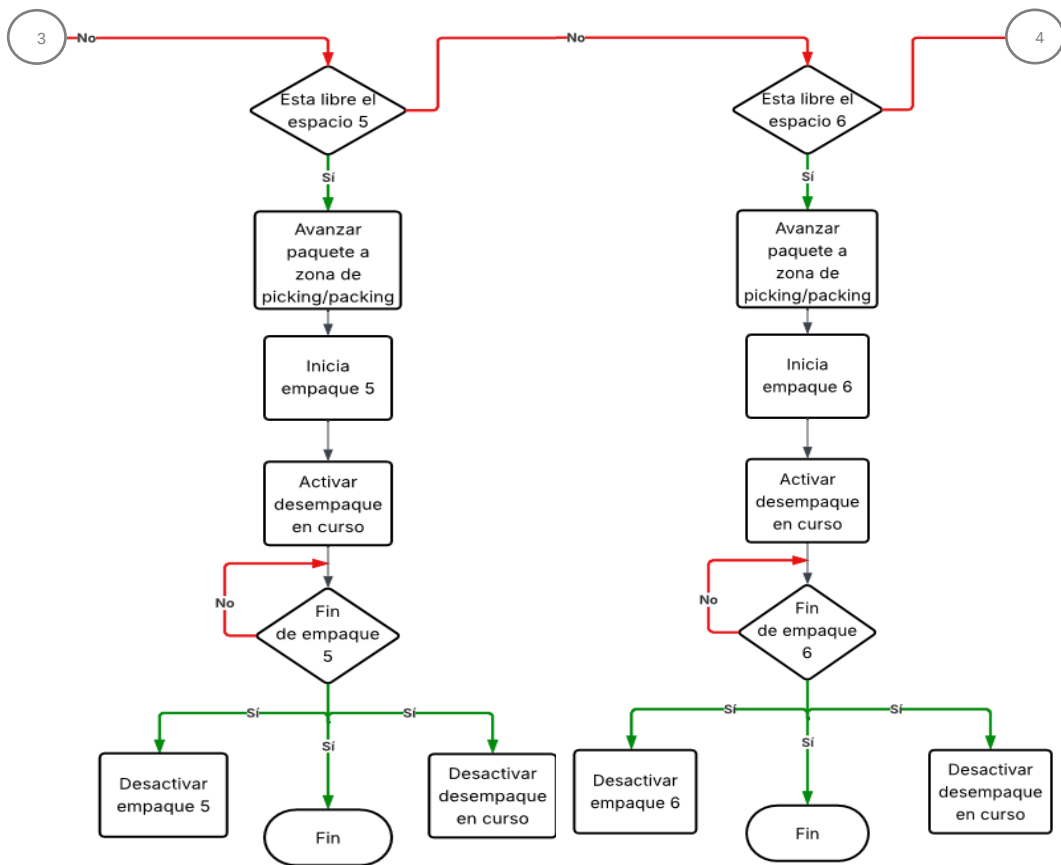
Como se menciona en el anterior algoritmo para la lógica del sistema parte I, existe la necesidad de realizar adicionalmente a las tareas de desempaque tareas de empaque. Por consiguiente, para que se lleve a cabo una tarea de empaque, primero se debe de tener un paquete en la zona de espera para ser almacenado. Después verificar si no existe una tarea de desempaque en curso y/o en espera, si existiese, se procederá a terminar su rutina de ejecución.

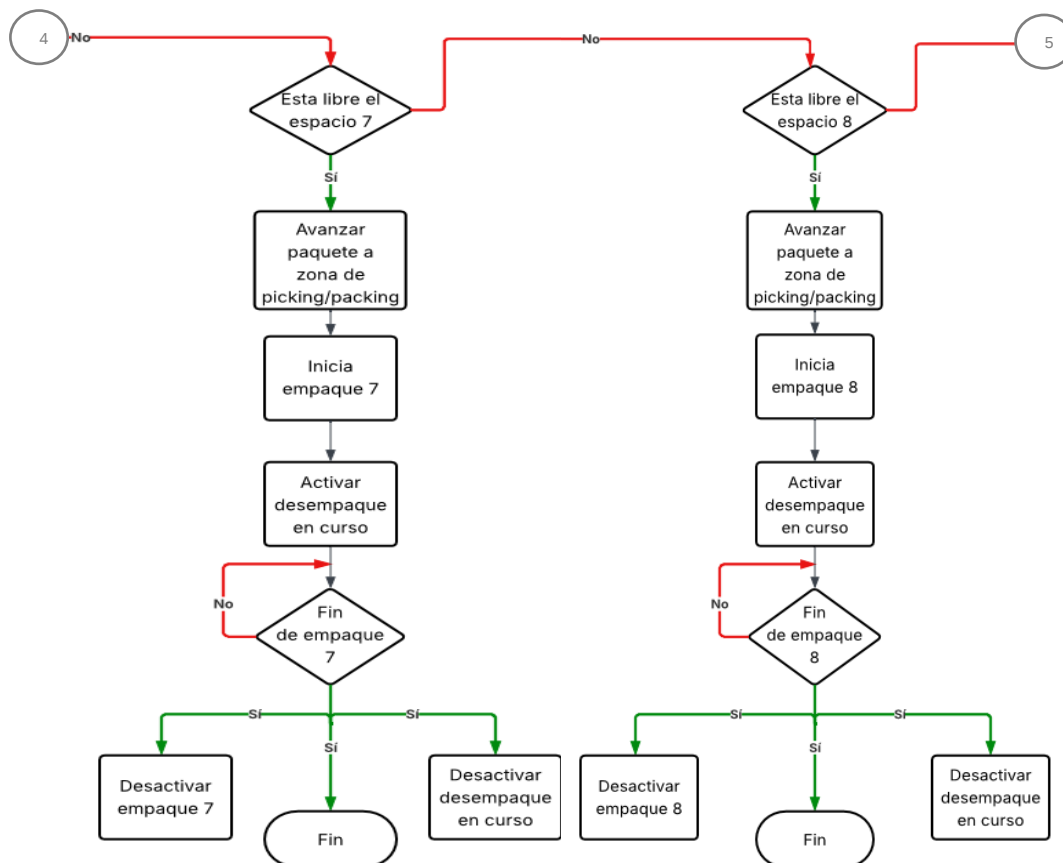
En caso de no existir o una vez finalizadas las tareas de desempaque se procederá a verificar, en una secuencia del espacio 1 al 9 la disponibilidad en el almacén, en donde por medio de las señales proporcionadas por los sensores LDR se podrá determinar qué espacios son los que se encuentran desocupados. Al encontrar un espacio disponible dentro del almacén para el paquete, se avanzará el paquete a la zona de picking/packing iniciando específicamente la rutina de movimiento asociada al espacio disponible realizada por el robot, como se indica también en el algoritmo de la Figura 5.4.

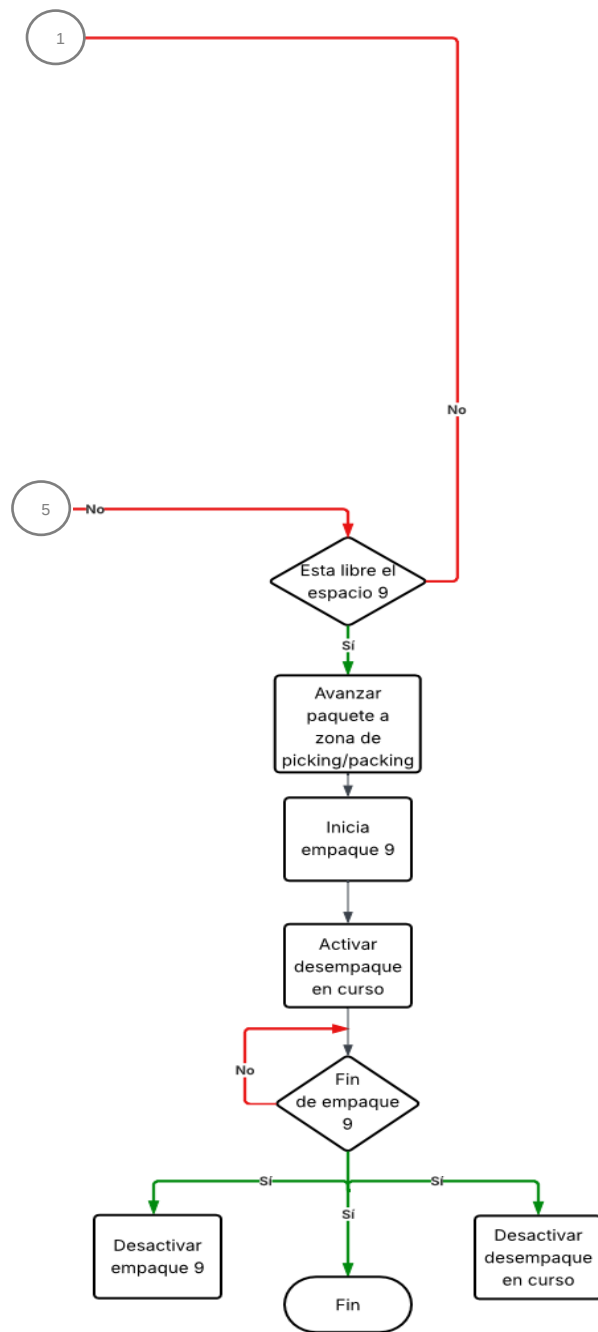
Por otro lado, si al verificar la disponibilidad en los 9 espacios, se encuentra que no existe la posibilidad de almacenar el paquete, este quedará nuevamente en espera, repitiendo así todo el proceso mencionado. Es por ello que antes de iniciar cualquier tarea de empaque se debe ejecutar primero toda tarea de desempaque, para así lograr la mayor disponibilidad de espacios dentro del almacén.











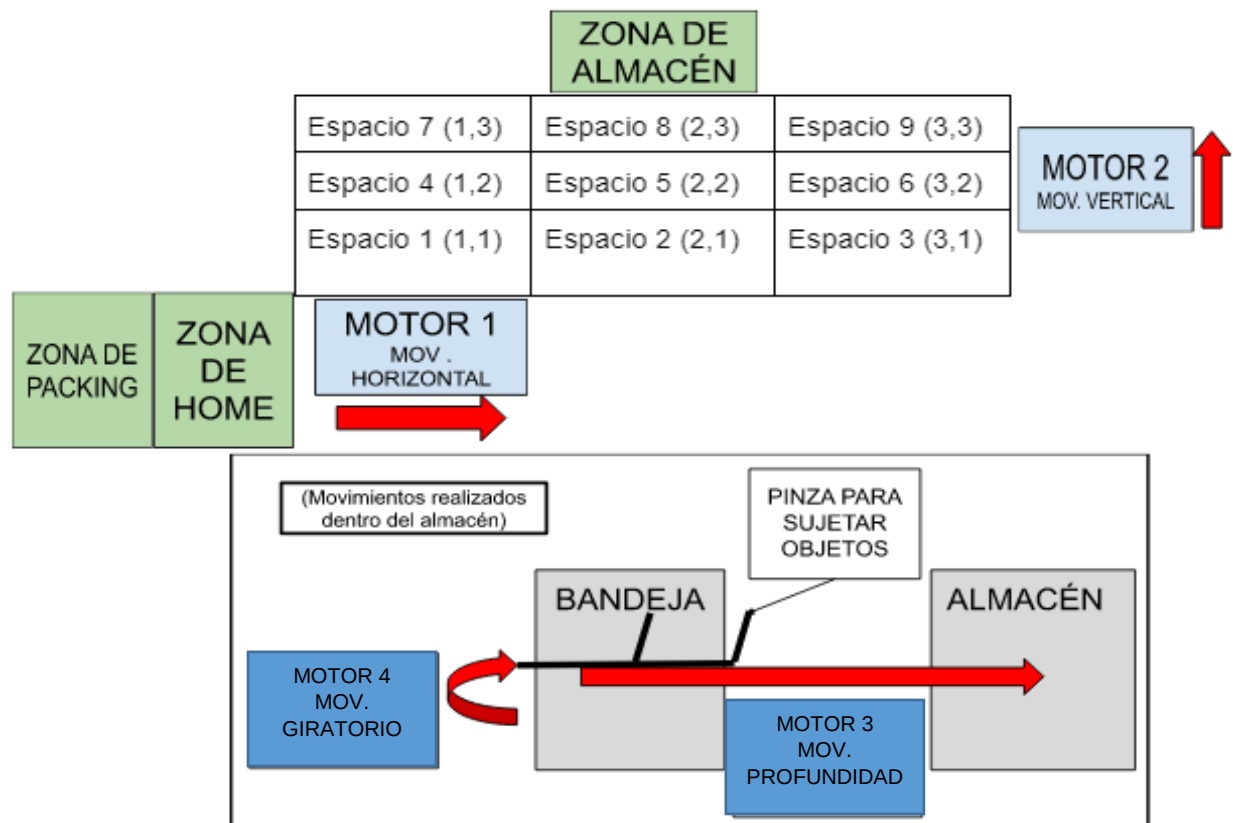
**Figura 5.4. Algoritmo de lógica parte II.**

### 5.3 Algoritmo movimiento de motores.

En el siguiente esquema de la Figura 5.5 se ejemplifica la distribución de los espacios de almacenamiento, mostrando tanto las combinaciones para las posiciones en el almacén como un plano cartesiano. De igual forma se puede apreciar la asignación de los cuatro motores con su respectivo movimiento a realizar, a partir de esta configuración de motores se puede establecer el orden de activación de motores dependiendo de que tarea se realizará:

- Tarea de empaque: motor 1, motor 2, motor 4 y motor 3.
- Tarea de desempaque: motor 1, motor 2, motor 3 y motor 4.

Así como también se muestra la zona de carga, descarga y almacén de los objetos.



**Figura 5.5. Esquema del sistema de almacenamiento.**

En la tabla del esquema se puede observar que los espacios (1,4 y 7), (2,5 y 8) y (3,6 y 9) se encuentran en las mismas columnas, es decir, en el eje X se requieren el mismo

avance para alcanzar esas posiciones horizontales, por otro lado cada espacio está asociado a un renglón distinto, por lo que se requiere un desplazamiento vertical diferente para cada espacio, en este caso los espacios (1,2 y 3), (4,5 y 6) y (7,8 y 9) se encuentran en los mismos renglones por lo que requieren el mismo avance vertical, aunque cada elemento se ubica ahora en una posición horizontal distinta.

Para ejemplificar estos procesos, se explica el arreglo asociado a los espacios 1 y 4 los cuales requieren el mismo avance en horizontal pero el avance en vertical es distinto, por otro lado los espacios 7 y 8 requieren el mismo avance vertical pero un avance horizontal diferente. En conclusión cada espacio tiene una cierta combinación de avances, pero, puede tener una cierta similitud en sus avances con otros espacios ya sea horizontal o verticalmente.

En base al esquema planteado y lo mencionado anteriormente se tiene que con los cuatro motores junto a los mecanismos piñón cremallera (los cuales nos permitirán transformar los movimientos rotativos de los motores en movimientos lineales) se pueden generar los avances necesarios generando una cierta serie de pulsos distintos asociados a las 3 diferentes opciones de desplazamiento tanto horizontalmente como verticalmente y a partir de combinaciones de estos 3 posibles movimientos horizontales y verticales se puede alcanzar cualquier posición asociada a un espacio de almacén, para el caso del motor 4 su movimiento rotativo de salida se conserva ya que se utiliza de forma directa para generar la acción de sujeción de los paquetes al hacer girar la estructura tipo F, en cuanto al motor 3 también hay una conversión de movimiento rotacional a lineal, pero en todos los casos su desplazamiento es igual, con todas estas herramientas es como se forma nuestro robot cartesiano el cual nos permitirá realizar las tareas tanto de empaque como las de desempaque. El algoritmo realizado nos ayudará a entender el cómo se generaron los movimientos en cada motor para realizar las tareas planteadas, es decir, en este algoritmo se generaron los movimientos requeridos para mover

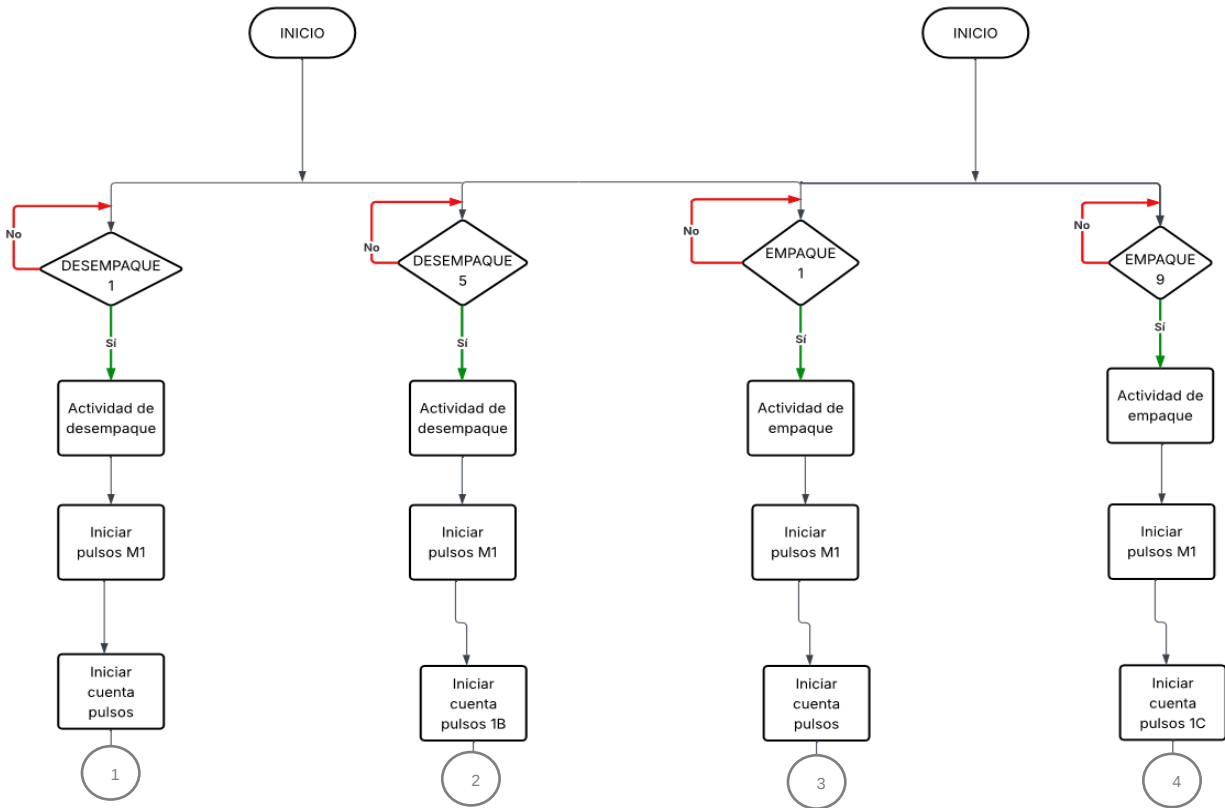
los objetos a las distintas posiciones del almacén. El funcionamiento del algoritmo es el siguiente:

- Cuando se genera la orden de un desempaque específico a través del algoritmo lógico (desempaque), se procederá a recoger el paquete en el espacio especificado mediante la siguiente secuencia de movimientos.
- El motor 1 será el primero en realizar su avance, para lo cual se procederá con una generación de pulsos para accionar el mismo y con ello avanzar a la bandeja del robot a la columna donde se realizará la tarea de desempaque, para lo cual se hará uso de un cuenta pulsos asociado a la posición horizontal deseada, cuando se cumpla la secuencia de pulsos necesaria para posicionar la bandeja en la columna específica, el contador finalizara la generación de pulsos y procederá a activar la siguiente secuencia de pulsos para activar el motor 2, el cual avanzará a la bandeja del robot verticalmente al renglón asociado a la tarea de desempaque, repitiendo el mismo mecanismo para alcanzar el renglón deseado con un cuenta pulsos específico para dicho renglón.
- Al ser una tarea de desempaque se debe de activar primero el motor 3 el cual permitirá avanzar el mecanismo de sujeción dentro del espacio de almacenamiento, este proceso se realizará de forma similar a el caso de los motores 1 y 2, solamente que en este caso el avance siempre será igual para cualquier desempaque, es decir solo se requería un único cuenta pulsos para esta labor, el cual al terminar su conteo activará el motor 4, que permite sujetar el paquete por medio del giro del mecanismo de sujeción F, en esta caso nuevamente se dispone de un único cuenta pulsos para esta tarea como en el caso del motor 3.
- Para terminar el desempaque el robot realizará una secuencia inversa de sus movimientos que le permitirá llevar la bandeja del robot a la zona de home para posteriormente empujar el paquete a la zona de picking/packing, esta secuencia se realizará bajo los mismos principios que el inicio de esta rutina solo que con el siguiente orden Motor 4→Motor 3→Motor 2 →Motor 1 y estos movimientos serán

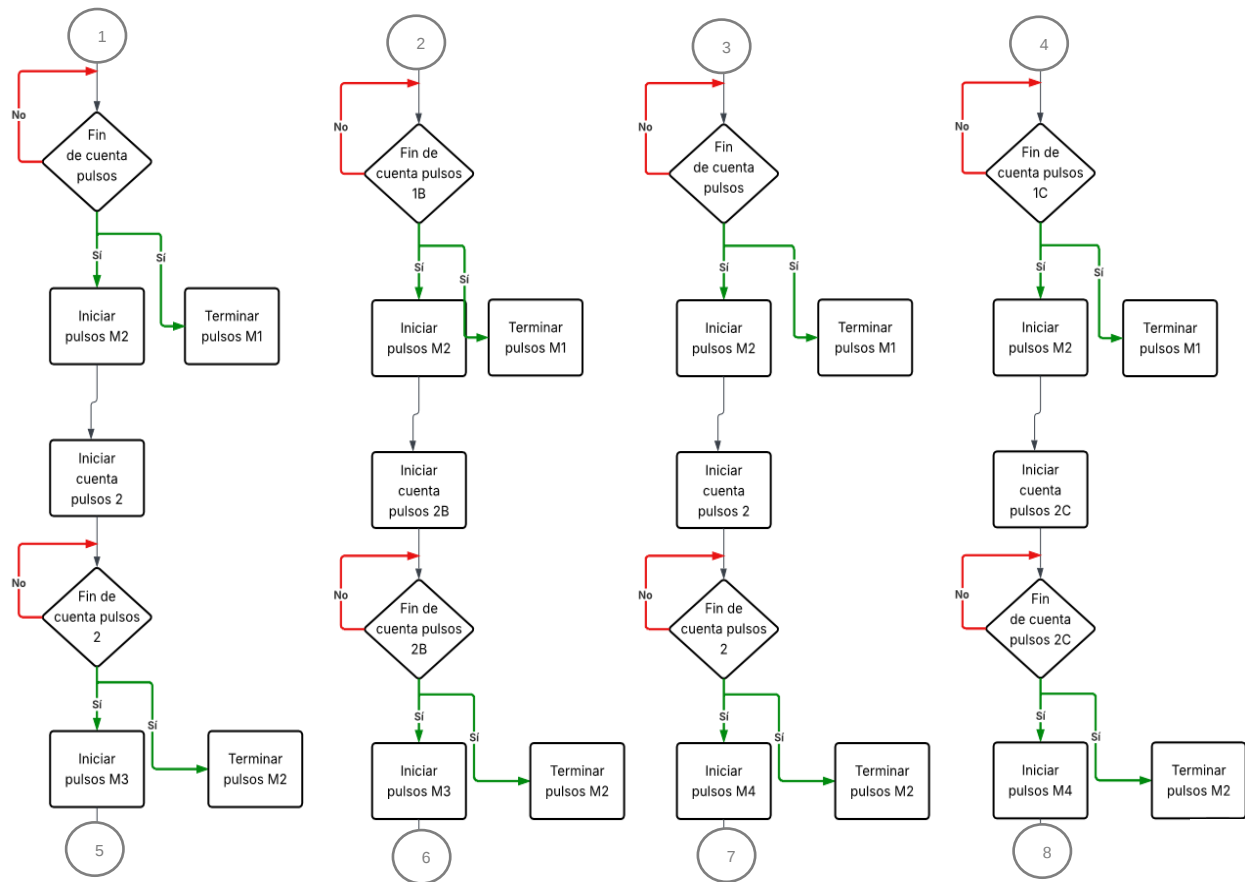
de la misma dimensión que en el caso inicial de cada motor solo que en sentido contrario.

- Ahora si se genera una orden de empaque a través del algoritmo lógico (empaque), se procederá a avanzar el paquete de la zona de packing a la bandeja del robot, realizando una secuencia de movimientos de la siguiente manera.
- Al igual que en una orden de desempaque el motor 1 será el primero en realizar su avance, repitiendo el mismo mecanismo de la generación y conteo de los pulsos necesarios para posicionar la bandeja del robot en la columna donde se realizará el empaque, una vez concluido el avance del motor 1 y de igual manera que en la tarea de desempaque se procederá a realizar el avance del motor 2 el cual posicionará la bandeja del robot en el renglón de interés para realizar el empaque haciéndolo de igual manera con el mecanismo ya mencionado.
- Al ser ahora una tarea de empaque se debe de activar primero el motor 4 permitiendo que el mecanismo de sujeción gire dejando así el paquete listo para ser empujado al espacio de almacenamiento, el proceso nuevamente se realizará de forma similar a el caso de los motores 1 y 2, y como ya se mencionó cuando es una tarea de desempaque solamente se requería de un único cuenta pulsos para ambas tareas, el cual al terminar su conteo activará ahora el motor 3, que permitirá el avance del mecanismo de sujeción y con ello el avance del paquete dentro del almacén, nuevamente se dispone de un único cuenta pulsos para esta tarea como en el caso del motor 4.
- Para terminar por completo la tarea de empaque el robot también realizará una secuencia inversa de sus movimientos que le permitirá llevar la bandeja del robot a la zona de home y estar a la orden de una tarea de empaque y/o desempaque, esta secuencia se realizará bajo los mismos principios que el inicio de esta rutina solo que con el siguiente orden Motor 3→Motor 4→Motor 2 →Motor 1 y estos movimientos serán de la misma dimensión que en el caso inicial de cada motor solo que en sentido contrario.

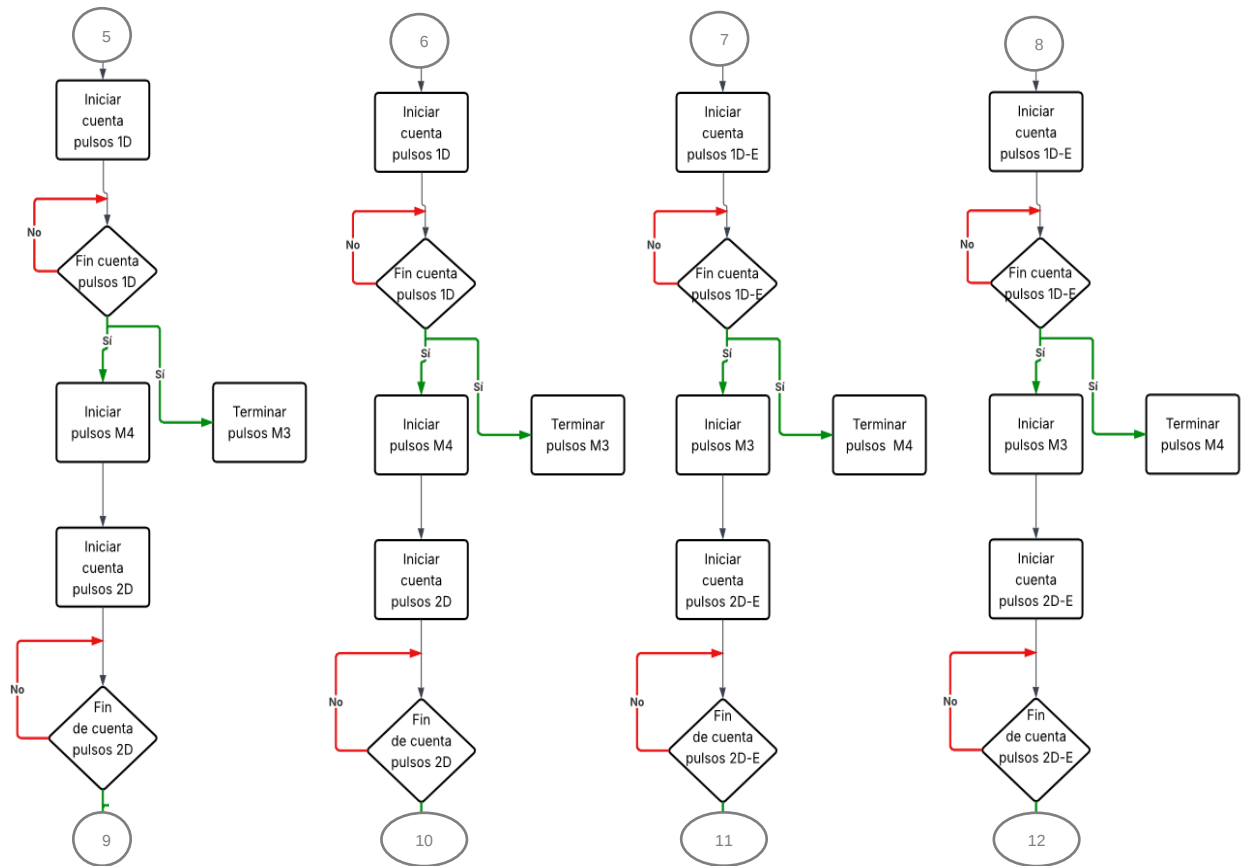
- De esta manera es como el robot lleva a cabo tareas de empaque y desempaques de manera autónoma por medio de los motores a pasos y mecanismos como el piñón cremallera.



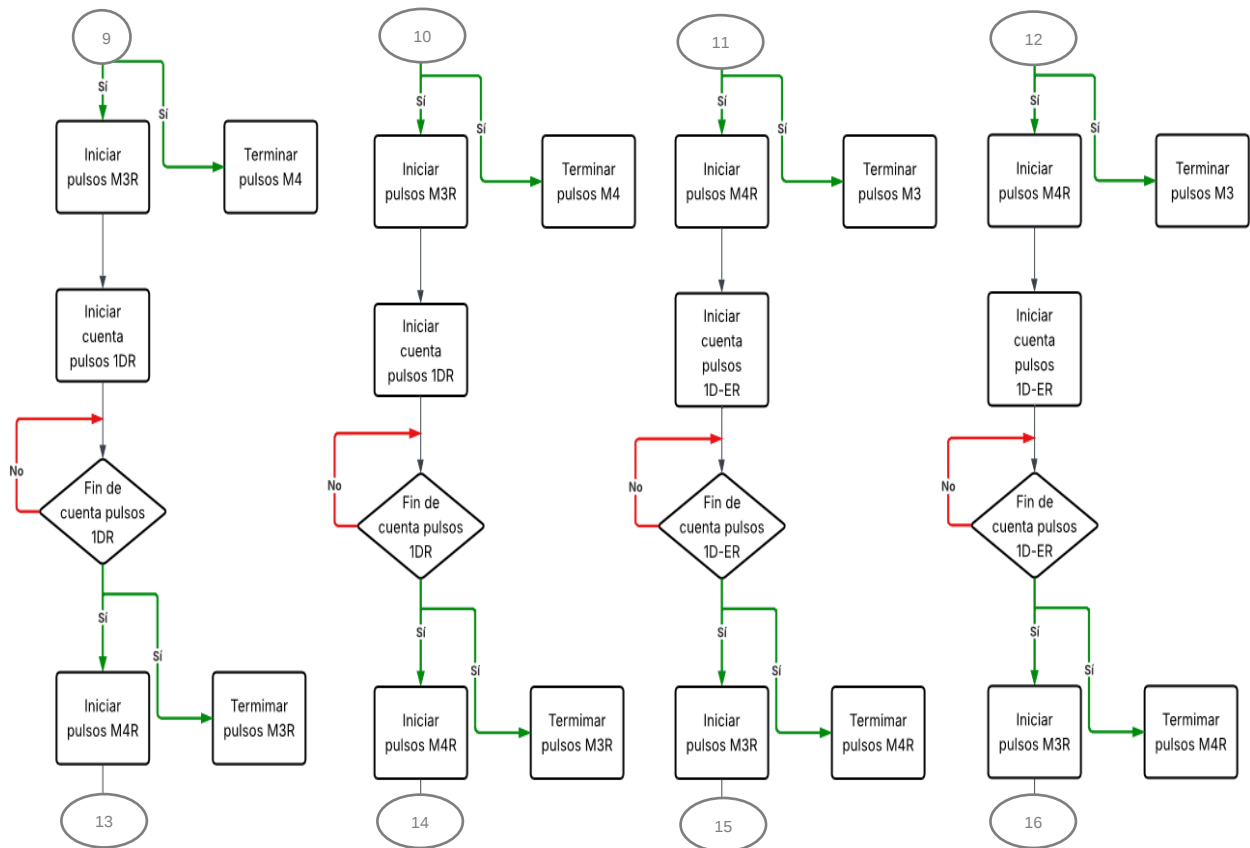
**Figura 5.6. Algoritmo movimiento de motores sección I.**



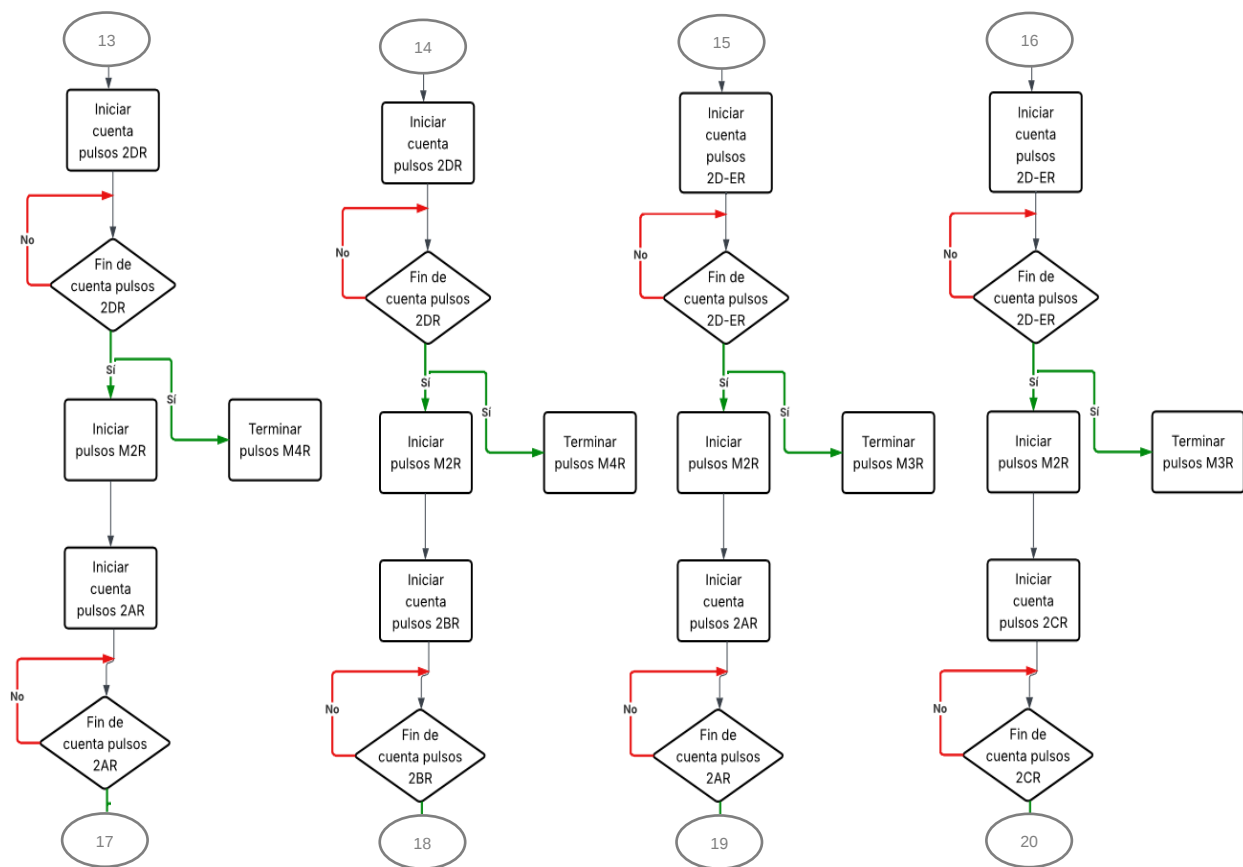
**Figura 5.7. Algoritmo movimiento de motores sección II.**



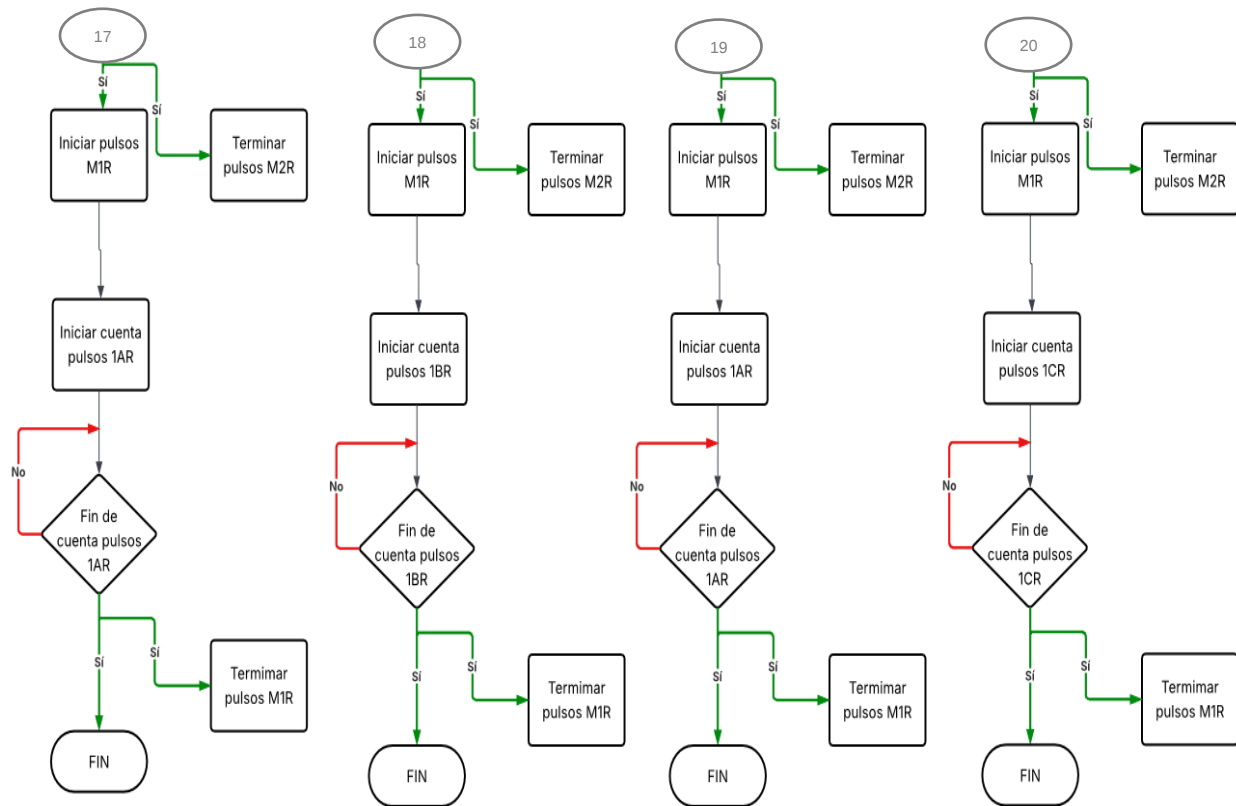
**Figura 5.8. Algoritmo movimiento de motores sección III.**



**Figura 5.9. Algoritmo movimiento de motores sección IV.**



**Figura 5.10. Algoritmo movimiento de motores sección V.**



**Figura 5.11. Algoritmo movimiento de motores sección VI.**

## 5.4 Programa en PLC para la lógica del sistema.

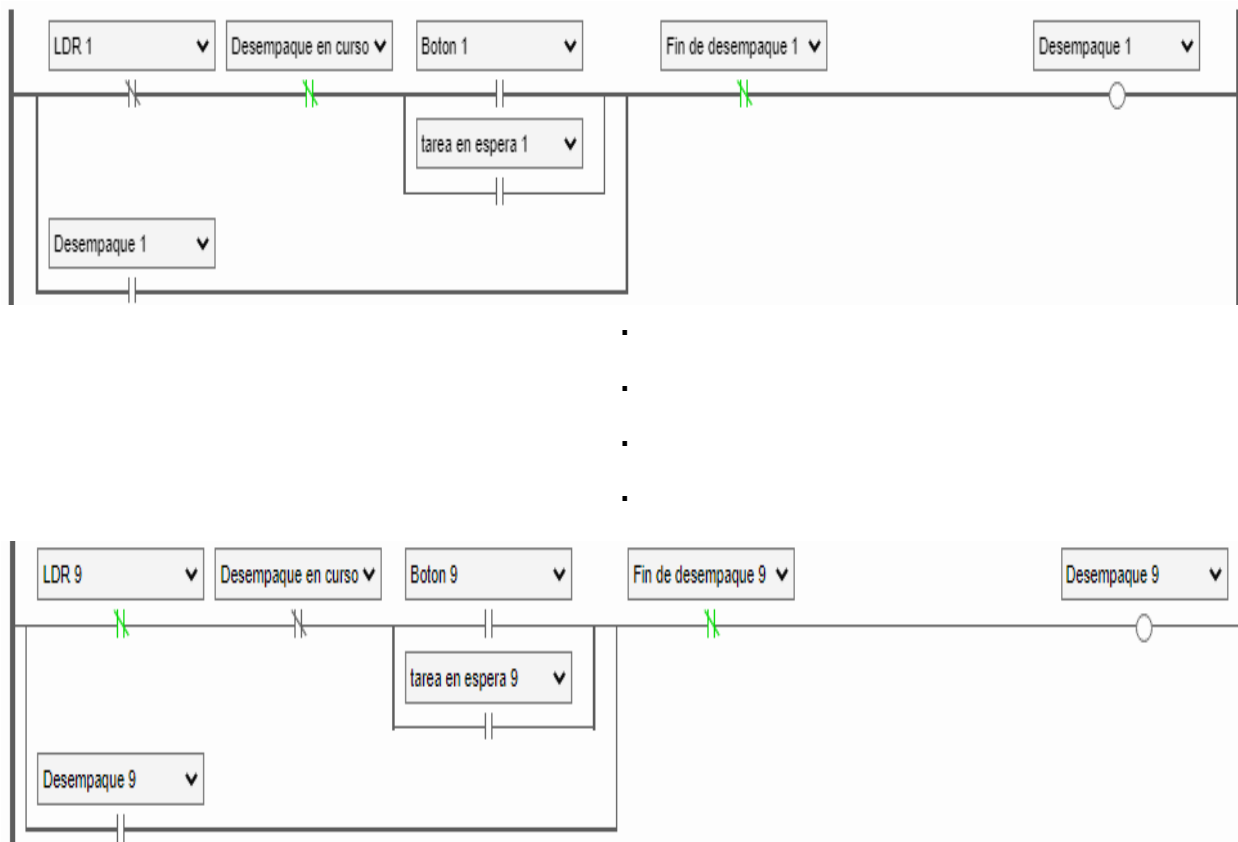
- Las señales para activar una tarea de empaque vienen por medio de un sensor de presencia que en términos de lenguaje escalera opera como un contacto normalmente abierto y para activar una tarea de desempaque o alguna de las colas de almacenamiento de tareas (tarea en espera, tarea en espera A), vienen por medio de una señal del botón que al igual este se representa como un contacto normalmente abierto, la diferencia entre uno y otro es que el contacto para la tarea de empaque se activa de manera automática al detectar el sensor la presencia de un objeto y el contacto para la tarea de desempaque se activa el botón de manera manual este al tener que ser presionado por un operario, tomando en cuenta que son 9 botones uno por cada espacio en el almacén, es decir, en general cada uno de estos botones puede activar este contacto.
- Mediante sensores de tipo LDR y trabajando con la señal proporcionada a la salida, podremos saber cuáles de los nueve espacios del almacén se encuentran disponibles y/o ocupados, esto nos permitirá identificar en qué espacios podremos almacenar objetos así como también nos permitirá realizar una tarea de desempaque si existiese un objeto almacenado en el espacio asociado a la orden solicitada.
- Como se observa en el algoritmo de la parte lógica, tenemos que la tarea de desempaque tiene la prioridad de ejecución ante una tarea de empaque, con el objetivo como se explicó anteriormente, de tener la mayor cantidad de lugares disponibles para poder almacenar objetos, estos se lograra al poner como prioridad de ejecución las tareas de desempaque así como generar una cola de espera de desempaque de máximo dos objetos, una vez terminada la tarea de desempaque en curso y concluidas las que están almacenadas en la cola de espera, se podrá realizar una tarea de empaque si así lo hubiera.

Para la implementación del programa como se establece en el algoritmo, se distinguen dos tipos de tareas, las tareas específicas que sirven para generar una rutina de movimiento específico y las tareas generales que sirven para bloquear

el accionamiento de tareas específicas adicionales y permitir generar memorias de almacenamiento, como por ejemplo en cuanto a tareas específicas tenemos el “desempaque 1”, “empaque 2”, “tarea en espera 3”, “tarea en espera A 4” y las tareas generales son “desempaque en curso”, “empaque en curso”, “tarea en espera” y “tarea en espera A”. El modo de operación de estas tareas se explica enseguida.

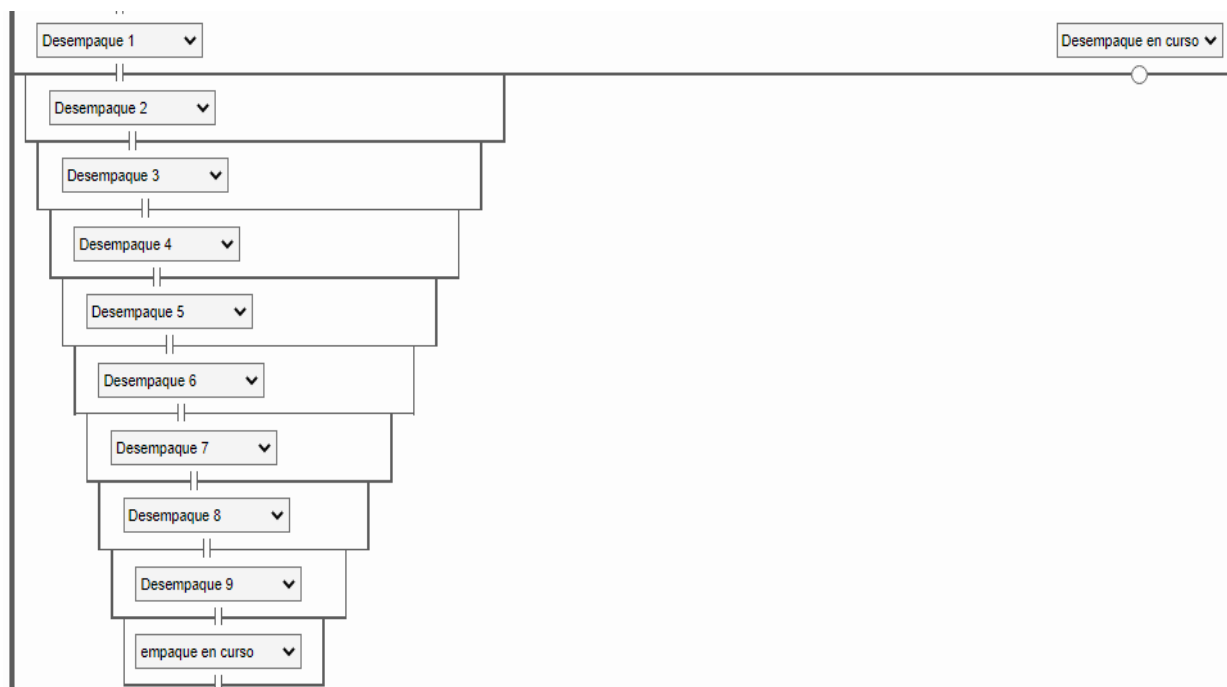
El primer objetivo es poder activar una tarea específica de desempaque, para lo cual se requiere cumplir ciertas condiciones como se indica en el algoritmo, estas condiciones de inicio se representan mediante un conjunto de contactos normalmente abiertos y cerrados conectados en serie, el primer contacto normalmente cerrado corresponde a la detección de existencia de un objeto almacenado a desempacar y se revisa mediante un sensor de tipo LDR que se desactiva cuando existe un objeto almacenado en la posición, la segunda condición a revisar es la inexistencia de una actividad en curso que nuevamente se revisa mediante un contacto normalmente cerrado etiquetado desempaque en curso, la tercer condición a revisar es la solicitud del desempaque de interés, esta condición se revisa mediante dos opciones la primera directamente por la solicitud de la tarea mediante un botón específico o la segunda opción mediante una tarea en espera específica, siempre y cuando se cumplan las tres condiciones se puede activar la tarea específica de desempaque, pero esta debe de permanecer activa aun cuando alguna condición de inicio deje cumplirse, esto se logra mediante el enclave de la tarea de desempaque específico el cual se genera poniendo un contacto normalmente abierto con dicha tarea de desempaque específico.

Finalmente la tarea se desactiva hasta que concluya la rutina de movimientos del programa del robot asociada a la tarea de desempaque específica esto se logra mediante un contacto normalmente cerrado etiquetado fin de desempaque. En la Figura 5.12 se pueden observar dos secciones del programa en PLC con lenguaje escalera con lo mencionado anteriormente.



**Figura 5.12. Sección del programa para tareas específicas de desempaque.**

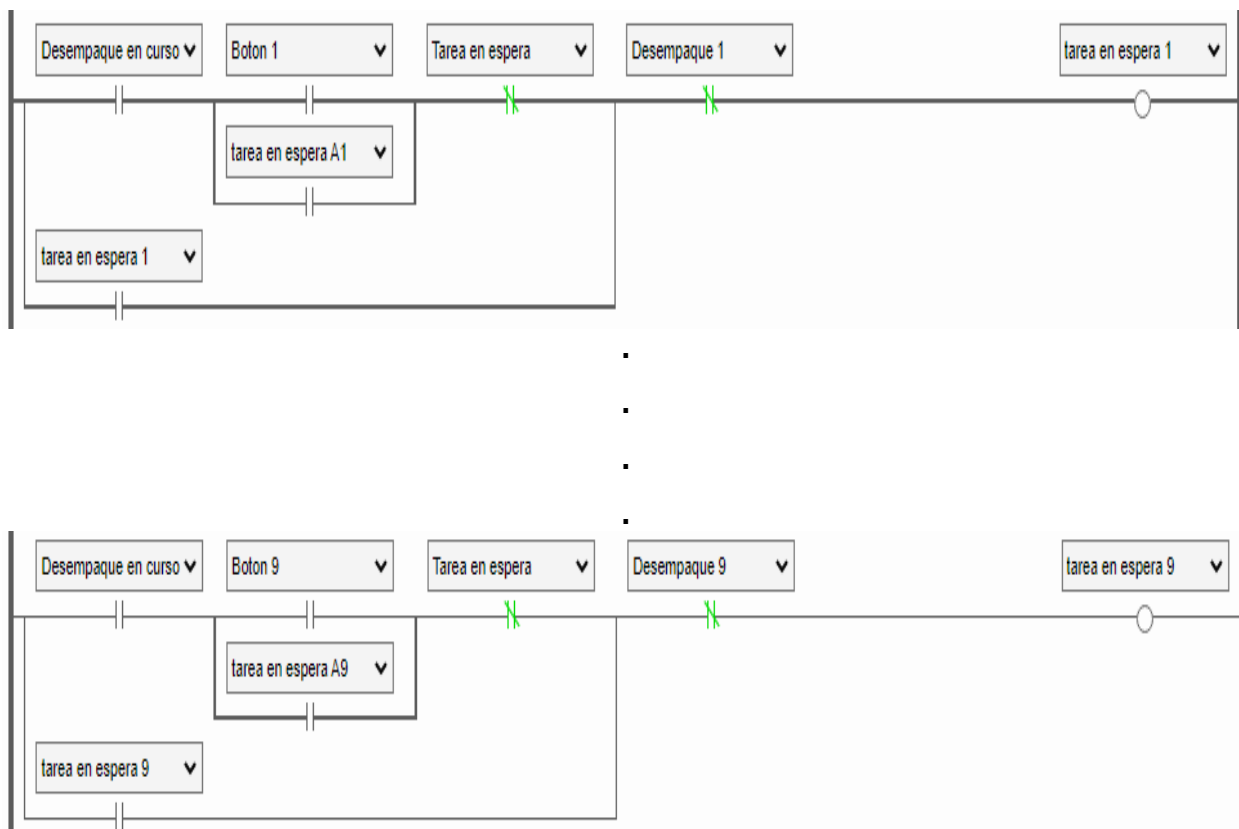
La tarea general desempaque en curso tiene la finalidad como se explicó en el párrafo anterior, de permitir bloquear tareas adicionales y como se explicará más adelante también permitirá activar la primera memoria denominada tarea en espera específica. Esta tarea general se activará siempre que exista alguna tarea específica de desempaque o un empaque en curso, lo cual se consigue con todos los contactos normalmente abiertos asociados a las tareas específicas de desempaque y la tarea de empaque en curso conectados en paralelo como se muestra en la Figura 5.13, y sólo permanecerá activa mientras alguna de estas tareas esté activa.



**Figura 5.13. Sección del programa para tarea general de desempaque en curso.**

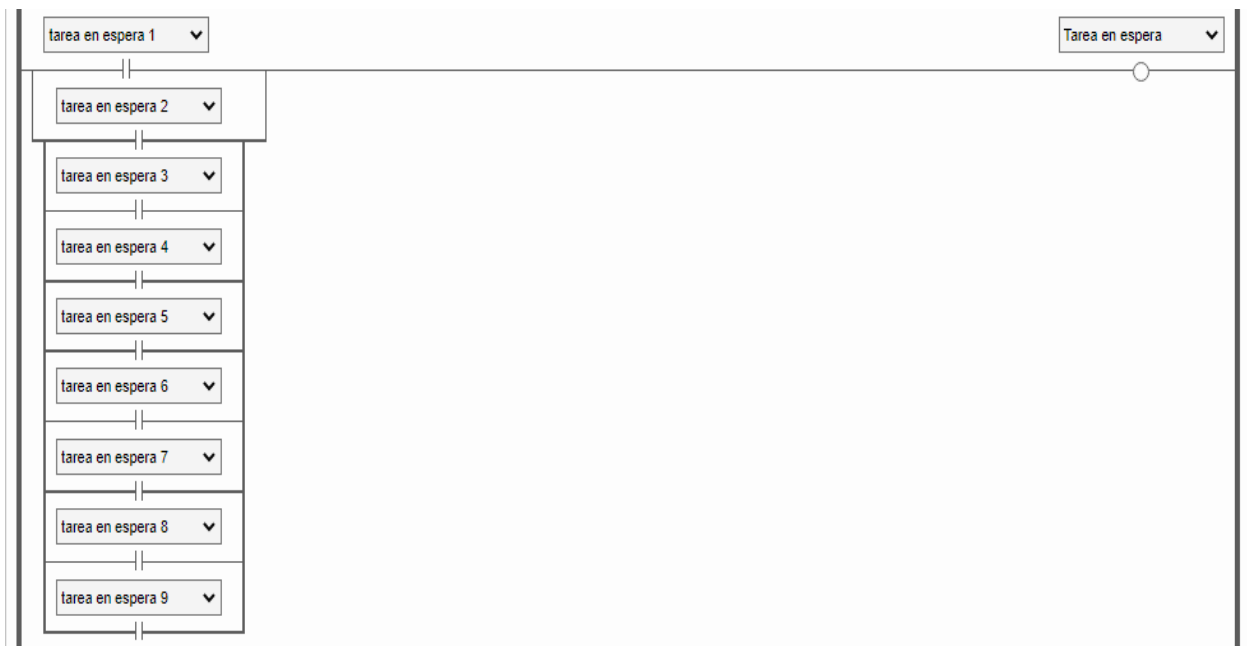
El segundo objetivo de la tarea general desempaque es poder activar una primera memoria de almacenamiento para una tarea específica de desempaque que se desee ejecutar al terminar un desempaque en curso o una tarea de empaque en curso, para lo cual se requiere cumplir ciertas condiciones como se indica en el algoritmo, estas condiciones de inicio se representan mediante un conjunto de contactos normalmente abiertos y cerrados conectados en serie, el primer contacto normalmente abierto corresponde a la existencia de un desempaque en curso etiquetado como desempaque en curso, la segunda condición a revisar es la solicitud del desempaque de interés, esta condición nuevamente mediante dos opciones la primera directamente por la solicitud de la tarea mediante un botón específico o la segunda en este caso mediante una segunda tarea de espera específica, la tercer condición a revisar es la inexistencia de una tarea en espera que nuevamente se revisa mediante un contacto normalmente cerrado etiquetado Tarea en espera. Siempre y cuando se cumplan las tres condiciones se puede activar la tarea específica de tarea en espera, pero esta debe de permanecer

activa aun cuando alguna condición de inicio deje cumplirse, esto se logra mediante el enlace de la tarea de tarea en espera específica el cual se genera poniendo un contacto normalmente abierto con dicha tarea de tarea en espera específica. Finalmente la tarea se desactiva hasta que se concluya la tarea de desempaque específica en curso, esto se logra mediante un contacto normalmente cerrado etiquetado desempaque específico. En la Figura 5.14 se pueden observar dos secciones del programa en PLC con lenguaje escalera con lo mencionado anteriormente.



**Figura 5.14. Sección del programa para tarea específica de tarea en espera.**

La tarea general Tarea en espera tiene la finalidad de permitir bloquear tareas en espera específicas adicionales, así como también permitirá activar la segunda memoria denominada Tarea en espera A específica. Esta tarea general se activará siempre que se agregue una tarea en espera específica en la primera memoria, lo cual se consigue con todos los contactos normalmente abiertos asociados a las tareas en espera específicas conectadas en paralelo como se muestra en la Figura 5.15 y sólo permanecerá activa mientras alguna de estas tareas esté activa.



**Figura 5.15. Sección del programa para tarea general de tarea en espera.**

El otro objetivo de la tarea general Tarea en espera es poder activar una segunda memoria de almacenamiento para una tarea específica de desempaque que se desee ejecutar al terminar una tarea en espera de un desempaque en curso, para lo cual se requiere cumplir ciertas condiciones como se indica en el algoritmo, estas condiciones de inicio se representan mediante un conjunto de contactos normalmente abiertos y cerrados conectados en serie, el primer contacto normalmente abierto corresponde a la existencia de una tarea en espera etiquetado como Tarea en espera, la segunda condición a revisar es la solicitud del desempaque de interés, esta condición ahora solamente se indica mediante

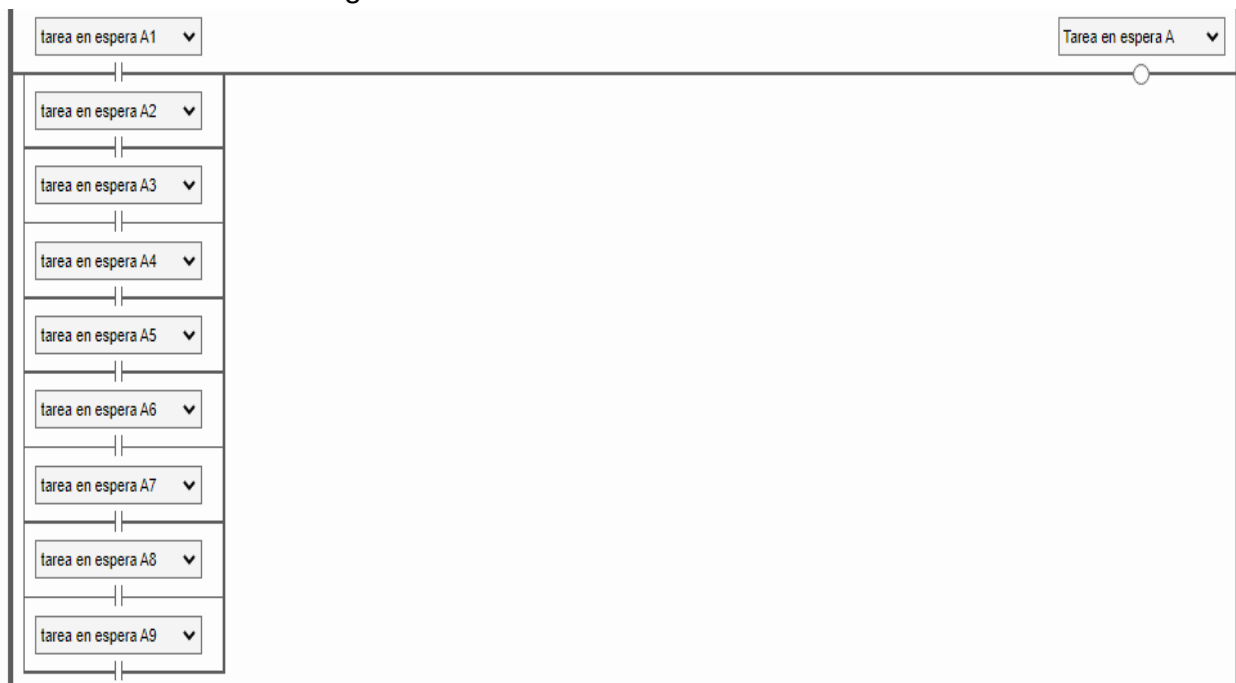
una opción la cual es directamente por la solicitud de la tarea mediante un botón específico, la tercer condición a revisar es la inexistencia de una segunda tarea en espera que nuevamente se revisa mediante un contacto normalmente cerrado etiquetado Tarea en espera A. Siempre y cuando se cumplan las tres condiciones se puede activar la tarea en espera A específica, pero esta debe de permanecer activa aun cuando alguna condición de inicio deje cumplirse, esto se logra mediante el enclave de la tarea de tarea en espera A específica el cual se genera poniendo un contacto normalmente abierto con dicha tarea de tarea en espera A específica en paralelo con el arreglo de las condiciones de inicio. Finalmente la tarea se desactiva hasta que la tarea en espera A específica avance a convertirse en una tarea en espera específica, esto se logra mediante un contacto normalmente cerrado etiquetado tarea en espera específico, lo cual indica que la tarea en espera A específica ha pasado a la primera memoria por lo que se puede eliminar de la segunda memoria. En la Figura 5.16 se pueden observar dos secciones del programa en PLC con lenguaje escalera con lo mencionado anteriormente.





**Figura 5.16. Sección del programa para tarea específica de tarea en espera A.**

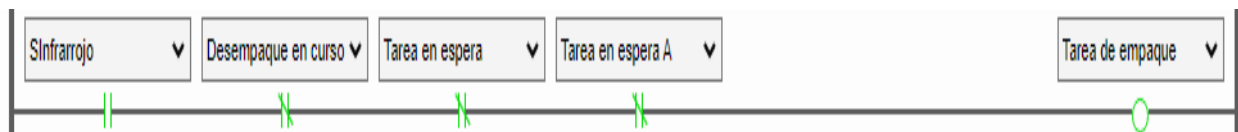
La tarea general Tarea en espera A tiene la finalidad de permitir bloquear el almacenamiento de otras tareas en espera A específicas, evitando de este modo sobrescribir en la segunda memoria más de una cola de almacenamiento a la vez. Su forma de activación es por medio de cualquiera de las tareas en espera A específicas, por lo cual su activación se da por medio de todas las tareas en espera A específicas indicadas por medio de contactos normalmente abiertos conectados en paralelo como se muestra en la Figura 5.17 y sólo permanecerá activa mientras alguna de estas tareas esté activa.



**Figura 5.17. Sección del programa para tarea general de tarea en espera A.**

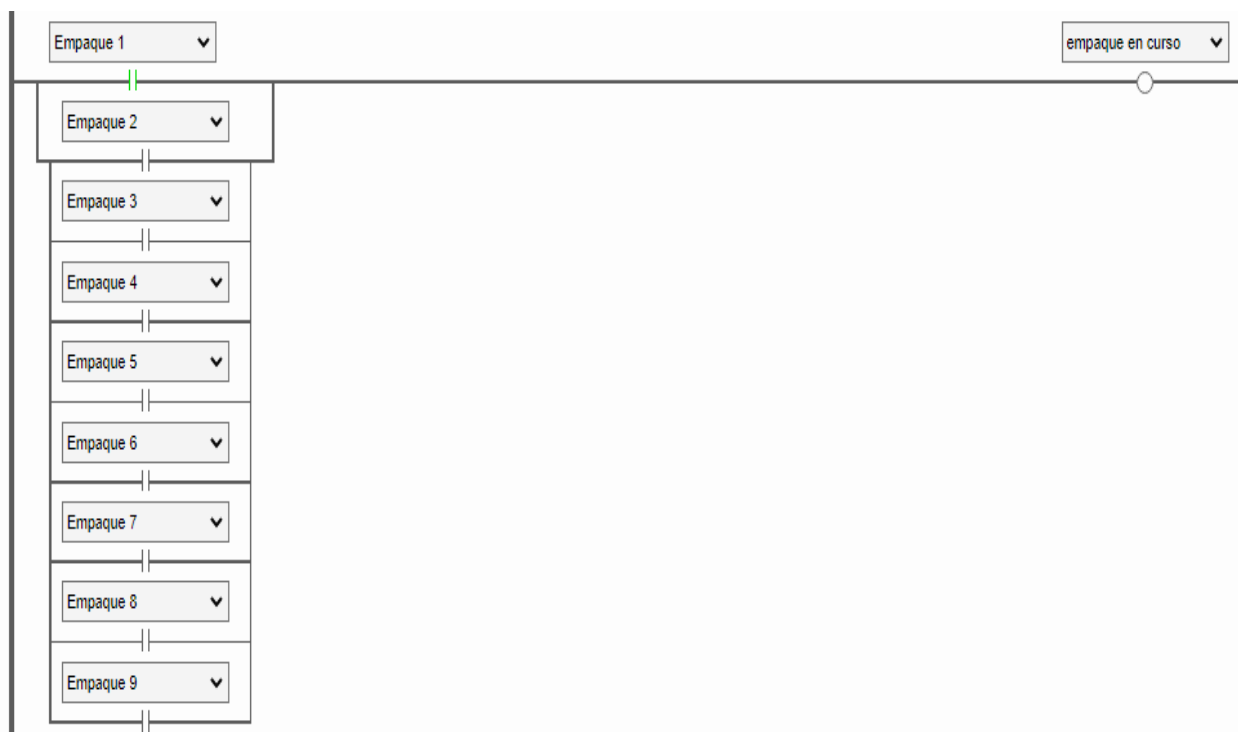
Como se observa en el algoritmo de la sección 5.2, tenemos las tareas de empaque las cuales como ya se mencionó no son prioridad de ejecución, a continuación se mencionara bajo qué condiciones se puede llevar a cabo una tarea de empaque específica.

Se tienen dos tareas generales, la primer tarea general tiene como objetivo activar una Tarea de empaque, para lo cual se requiere cumplir ciertas condiciones, como se indica en el algoritmo, estas condiciones de inicio se representan mediante un conjunto de contactos normalmente abiertos y cerrados conectados en serie como se muestra en la Figura 5.18, el primer contacto normalmente abierto corresponde a la detección de un paquete por medio de un sensor de tipo infrarrojo etiquetado como Sinfrarrojo , la segunda condición a revisar es la inexistencia de una actividad en curso que se revisa mediante un contacto normalmente cerrado etiquetado desempaque en curso, la tercera condición a revisar es la inexistencia de una tarea en espera que nuevamente se revisa mediante un contacto normalmente cerrado etiquetado Tarea en espera y la cuarta condición a revisar es la inexistencia de una tarea en espera A que al igual se revisa mediante un contacto normalmente cerrado etiquetado Tarea en espera A. Siempre y cuando se cumplan estas cuatro condiciones se podrá activar la tarea general de Tarea de empaque.



**Figura 5.18. Sección del programa para tarea general de tarea de empaque.**

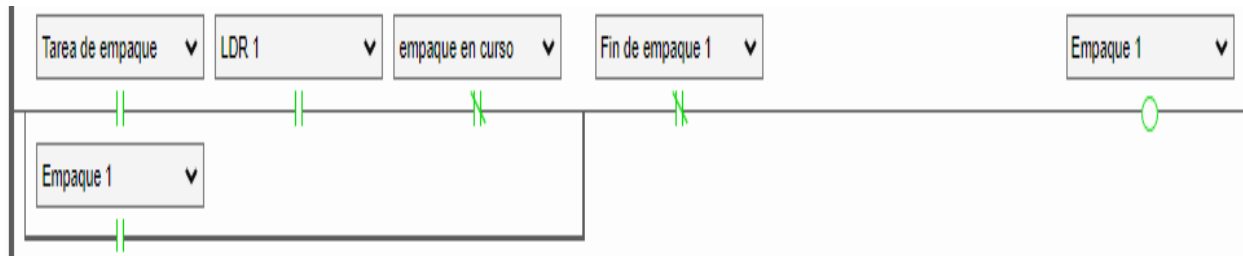
La segunda tarea general tiene como objetivo activar un empaque en curso con la finalidad de permitir bloquear tareas adicionales de empaque, así como también el bloquear tareas adicionales de desempaque. Esta tarea general se activará siempre que exista alguna tarea específica de empaque, lo cual se consigue con todos los contactos normalmente abiertos asociados a las tareas específicas de empaque conectadas en paralelo como se muestra en la Figura 5.19 y sólo permanecerá activa mientras alguna de estas tareas esté activa.



**Figura 5.19. Sección del programa para tarea general de empaque en curso.**

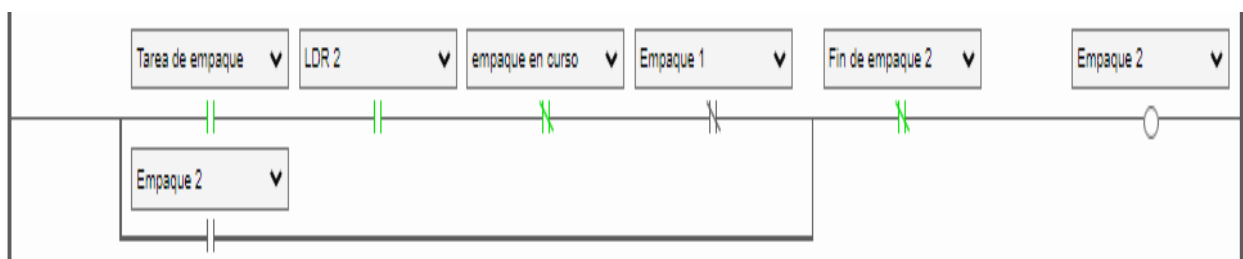
Finalmente para realizar una tarea específica de empaque se tienen que cumplir las siguientes condiciones de inicio, las cuales se representan mediante un conjunto de contactos normalmente abiertos y cerrados conectados en serie como se muestra en la Figura 5.20, el primer contacto normalmente abierto corresponde a la existencia de una tarea de empaque etiquetado como Tarea de empaque, el segundo contacto normalmente abierto corresponde a la inexistencia de un paquete almacenado en un espacio específico por medio del sensor tipo LDR, el tercer condición a revisar es la inexistencia de una actividad en curso que nuevamente se revisa mediante un contacto normalmente cerrado etiquetado empaque en curso, siempre y cuando se cumplan las tres condiciones se puede activar la tarea específica de empaque, pero esta debe de permanecer activa aun cuando alguna condición de inicio deje cumplirse, esto se logra mediante el enclave de la tarea de empaque específico el cual se genera poniendo un contacto normalmente abierto con dicha tarea de empaque específico. Finalmente la tarea

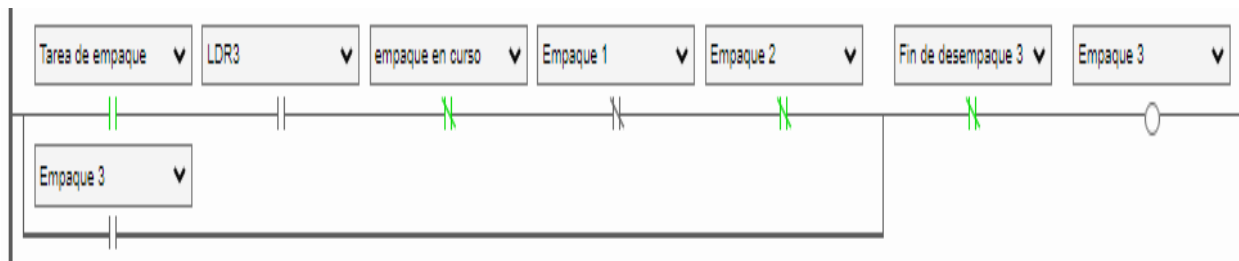
se desactiva hasta que concluya la rutina de movimientos del programa del robot asociada a la tarea de empaque específica esto se logra mediante un contacto normalmente cerrado etiquetado fin de empaque específico.



**Figura 5.20. Sección del programa para tarea específica de empaque.**

Para realizar un empaque en un cierto espacio del almacén se debe de realizar un barrido en orden desde el espacio 1 al 9 para así ubicar los espacios en los cuales se puede realizar un empaque, este proceso es similar al proceso explicado anteriormente, es decir, se deben cumplir forzosamente las tres condiciones de inicio, así como el respectivo enlace específico y finalización de la tarea de empaque específica. Pero para cada espacio específico subsecuente se verifica por medio de contactos normalmente cerrados etiquetados como empaque específico si se tiene la inexistencia de espacios disponibles como se ejemplifica en la Figura 5.21. De esta manera se realizará las tareas de empaque de forma ordenada en lugares los cuales presenten disponibilidad de almacenamiento.

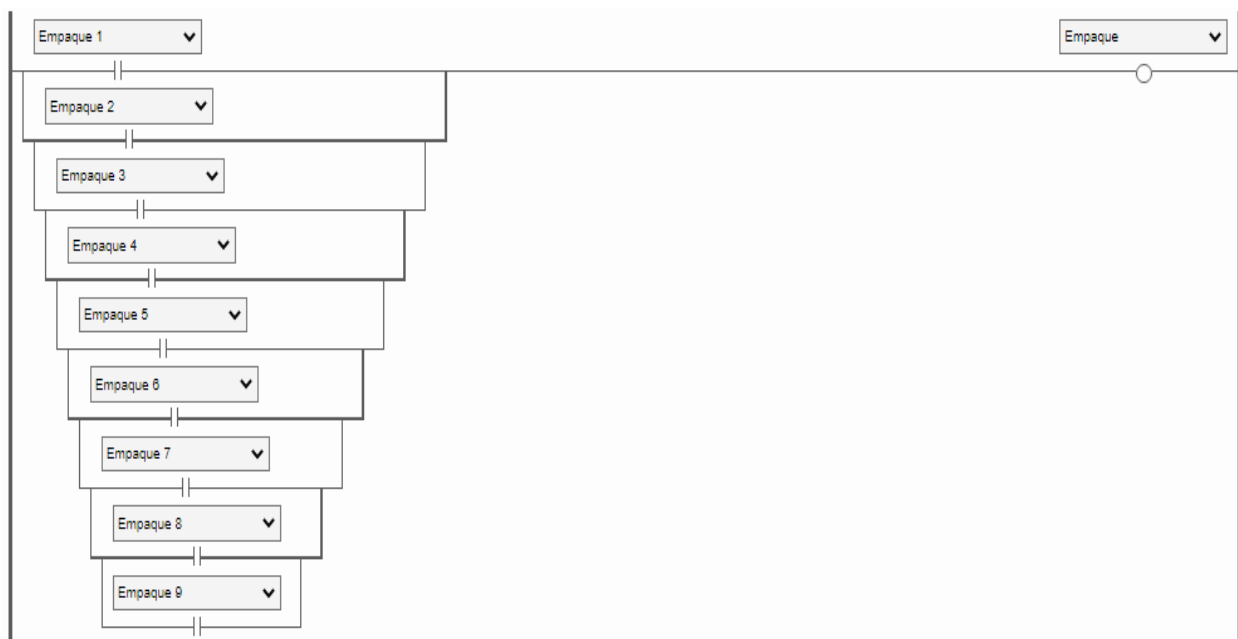




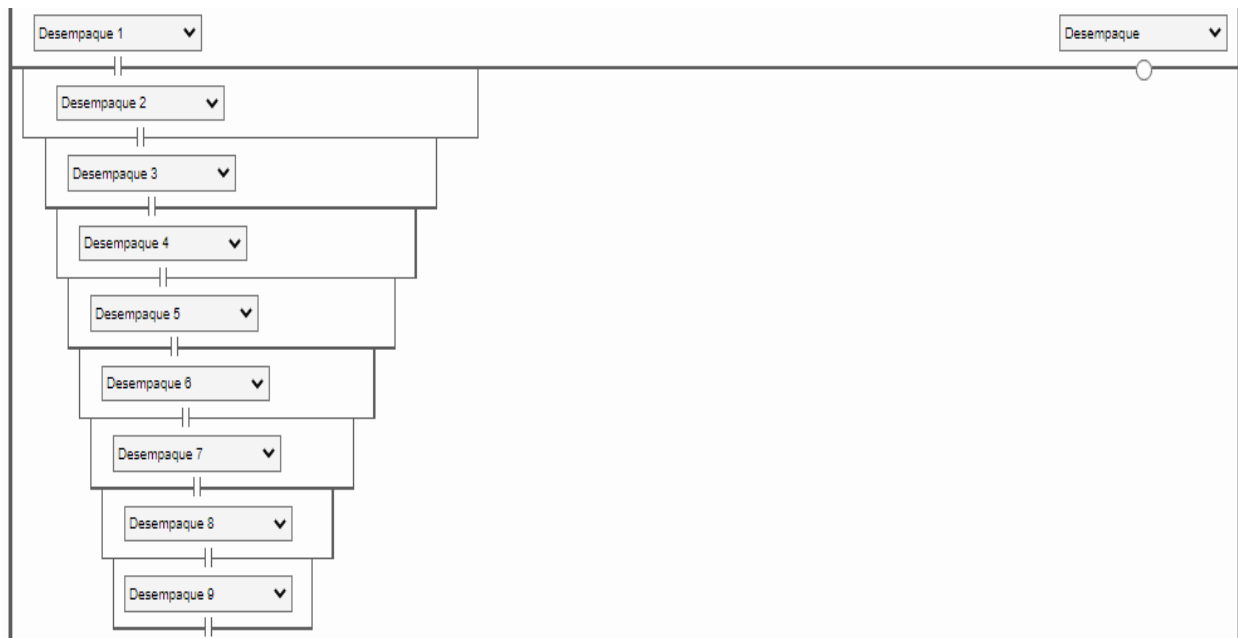
**Figura 5.21. Sección del programa para tareas específicas de empaque.**

## 5.5 programa en PLC para el movimiento de motores.

- Dado que se utilizarán motores a pasos es necesario generar pulsos eléctricos con los cuales trabaja este para rotar el eje del motor y así alcanzar las distintas posiciones de manera precisa, a continuación se explica el funcionamiento del programa en PLC con lenguaje escalera para poder generar dichos pulsos y poder rotar los motores en ciertas posiciones deseadas para así lograr las posiciones de empaque y desempaquetado de objetos.
1. Todas las tareas de empaque tienen las mismas rutinas de movimiento, solo varían en cuanto a las distancias que recorren en las posiciones del almacén, por lo cual, se diseña una única tarea de empaque y posteriormente dependiendo de la posición donde se realizará el empaque se utilizara la tarea específica para limitar el movimiento. Esta misma situación se repite en las actividades de desempaquetado, razón por la cual, se diseña nuevamente una tarea de desempaquetado general. En ambos casos la tarea de empaque y desempaquetado general, se activa a partir de cualquier actividad específica, por lo que en la estructura del programa se seleccionan una configuración en paralelo normalmente abierta de todas las tareas, para la activación de las tareas generales, como se ilustra en la Figuras 5.5.1 y 5.5.2.



**Figura 5.5.1. Sección del programa para tareas generales de empaque.**



**Figura 5.5.2. Sección del programa para tareas generales de desempaque.**

2. Esta sección es para poner en movimiento al motor 1 para las rutinas de empaques y desempaques específicamente (1,4 y 7) los cuales requieren el mismo número de pulsos, al estar ubicados a la misma distancia en el eje X de la zona de Home.

- Para generar los pulsos eléctricos seleccionamos dos timers de tipo On Delay nombrados como On timer y Off timer estos dos se estarán conmutando entre ellos aprovechando su estructura de funcionamiento, en primer lugar el On timer se activará de por medio de un contacto normalmente abierto que responde a cualquier orden de empaque o desempaque, después de un tiempo establecido el On timer mandará una señal de activación etiquetada como On timer.Q que permitirá activar el Off timer, el cual después de un periodo de tiempo establecido activará su propia señal Off timer.Q. la cual apagará momentáneamente el On timer y esté simultáneamente apagará el Off timer, al apagarse la señal de Off timer, el On timer nuevamente podrá encenderse y el proceso se repetirá de forma continua generando una señal pulsante para los motores a pasos.
- El On timer permitirá además activar una bobina llamada Motor 1 por medio de un contacto normalmente abierto llamado On timer.TT y mientras la bobina esté pulsando un contador de tipo ascendente llamado cuenta pulsos llevará el conteo de dichos pulsos parando hasta el valor establecido de conteo y cortando así la conmutación entre los timers por medio de un contacto normalmente cerrado llamado cuenta pulsos.Q. Dichos pulsos son establecidos de acuerdo a la distancia que se necesita recorrer para llegar a la posición deseada.
- Los contactos normalmente abiertos de los empaques y desempaques (1,4 y 7), son colocados en paralelo para que cuando se seleccione cualquiera de las seis tareas el contador cuenta pulsos se active y sea el que permite parar la acción del motor 1, esto debido a que como se mencionó estas posiciones se requiere que el motor 1 que realiza los movimientos en el eje X vaya a la misma posición del almacén.



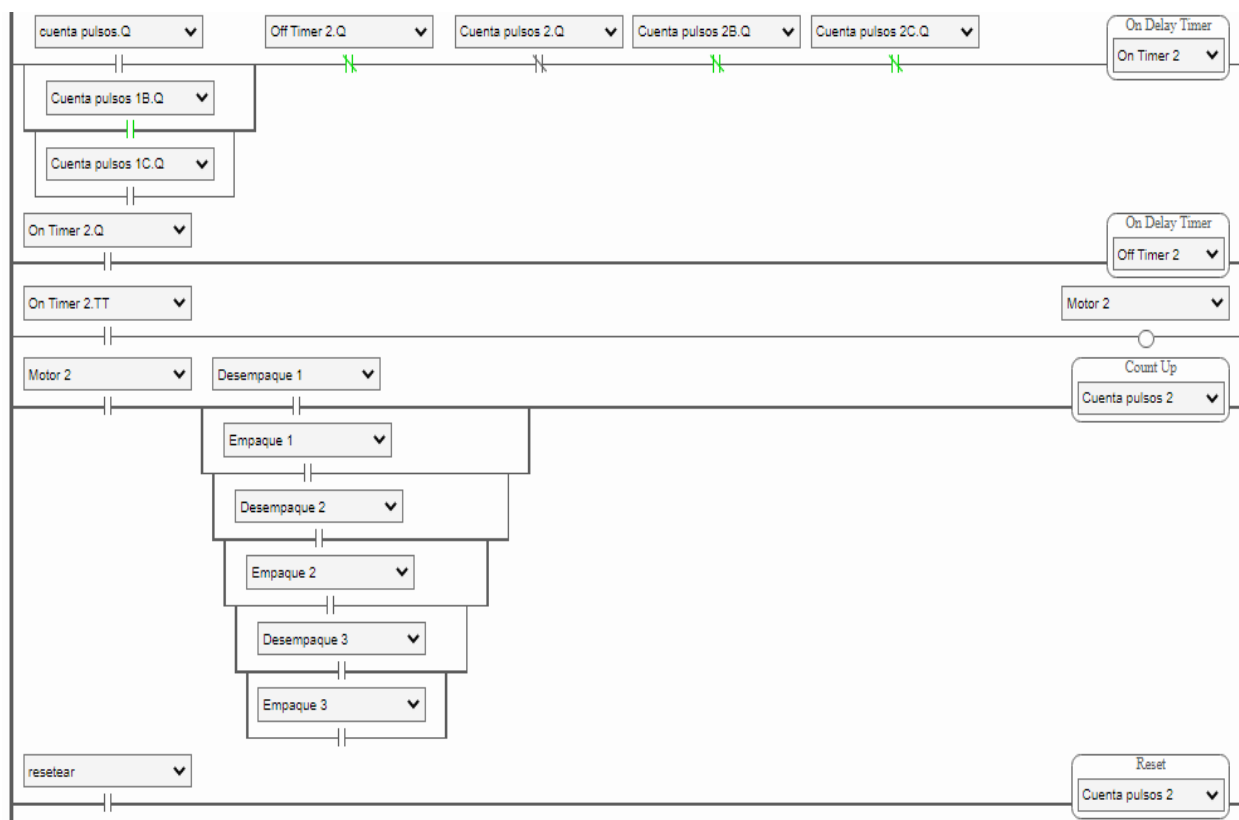
**Figura 5.5.3. Sección del programa para el movimiento del motor 1 de tareas de empaque y desempaque (1,4 y 7).**

- El mecanismo de movimiento para el motor 1 es el mismo cuando se realizan empaques y desempaques los cuales solo pueden alcanzar tres posibles posiciones horizontales asociadas a las tres columnas de almacenamiento, para el caso de la primer columna como se explicó anteriormente se pueden alcanzar las posiciones (1,4 y 7) en el caso de las posiciones (2,5 y 8) y (3,6 y 9) las encontramos en las columnas 2 y 3 respectivamente, de tal forma que se requiere el uso de los mismos dos timers para generar los pulsos que en el caso de la primera columna pero de un cuenta pulsos diferente para cada columna como se muestra en la Figura 5.5.3 , y la secuencia de pulsos meta establecidos en cada cuenta pulsos tiene valores específicos para la posición deseada. Por lo tanto se tendrán tres distintos cuenta pulsos cada uno asociado a las tres posibles distancias que el motor 1 va a recorrer a lo largo del eje X. Los dos timers serán

los mismos para el motor 1, donde los cuenta pulsos serán los encargados de limitar dicha generación de pulsos según sea la posición de interés.

3. Para el movimiento el motor 2 tomando como ejemplo las rutinas de empaques y desempaques como (1,2 y 3) los cuales requieren el mismo número de pulsos, al estar ubicados a la misma distancia del eje Y. El mecanismo es el siguiente:

- Una vez terminada la rutina del motor 1 para cualquiera de estas 6 tareas y por medio del cuenta pulsos que haya estado trabajando, se enviará una señal de inicio para que nuevamente dos timers de tipo on delay distintos a los del motor 1, comiencen a conmutarse repitiendo así el mecanismo ya explicado para el movimiento del motor 1.



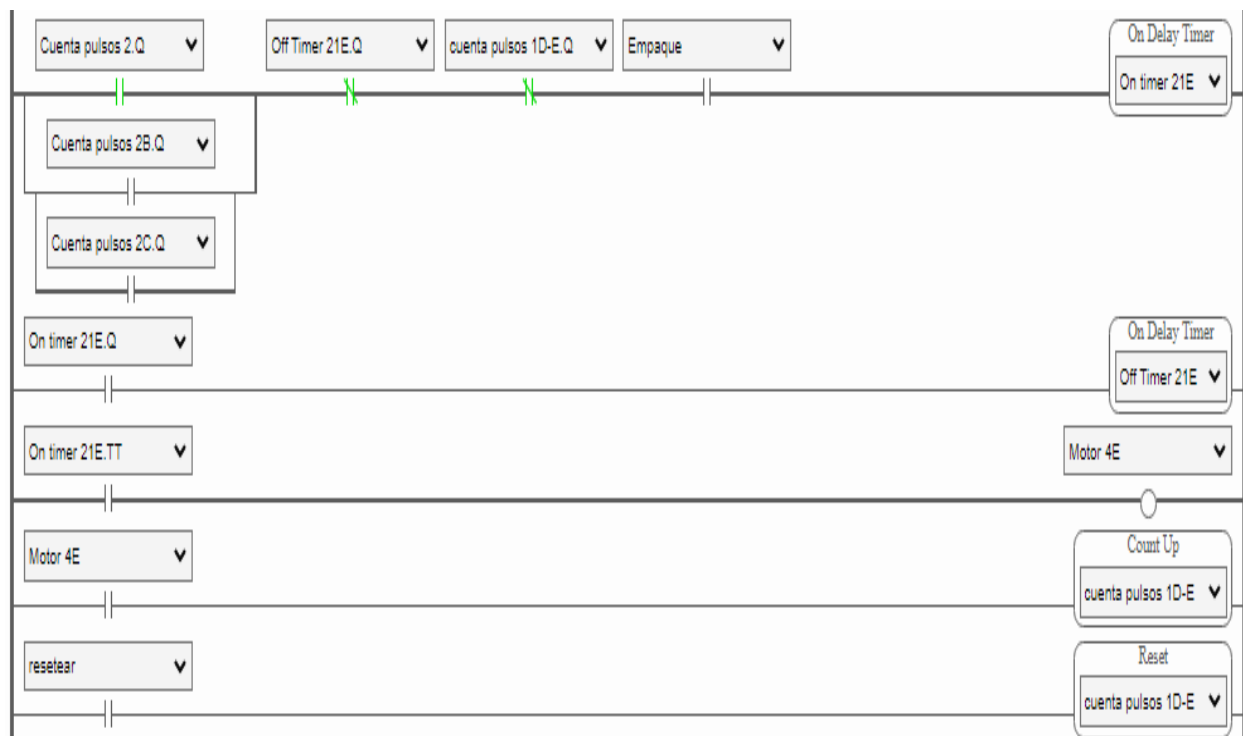
**Figura 5.5.3. Sección del programa para el movimiento del motor 2 de tareas de empaque y desempaque (1,2 y 3).**

- Para las rutinas del motor 2 se requieren de tres distintos cuenta pulsos establecidos a un cierto número de pulsos para lograr alcanzar las tres distintas alturas en el almacén como se observa en la Tabla 5.5.1.

***Tabla 5.5.1. Asignación de cuenta pulsos en los nueve espacios.***

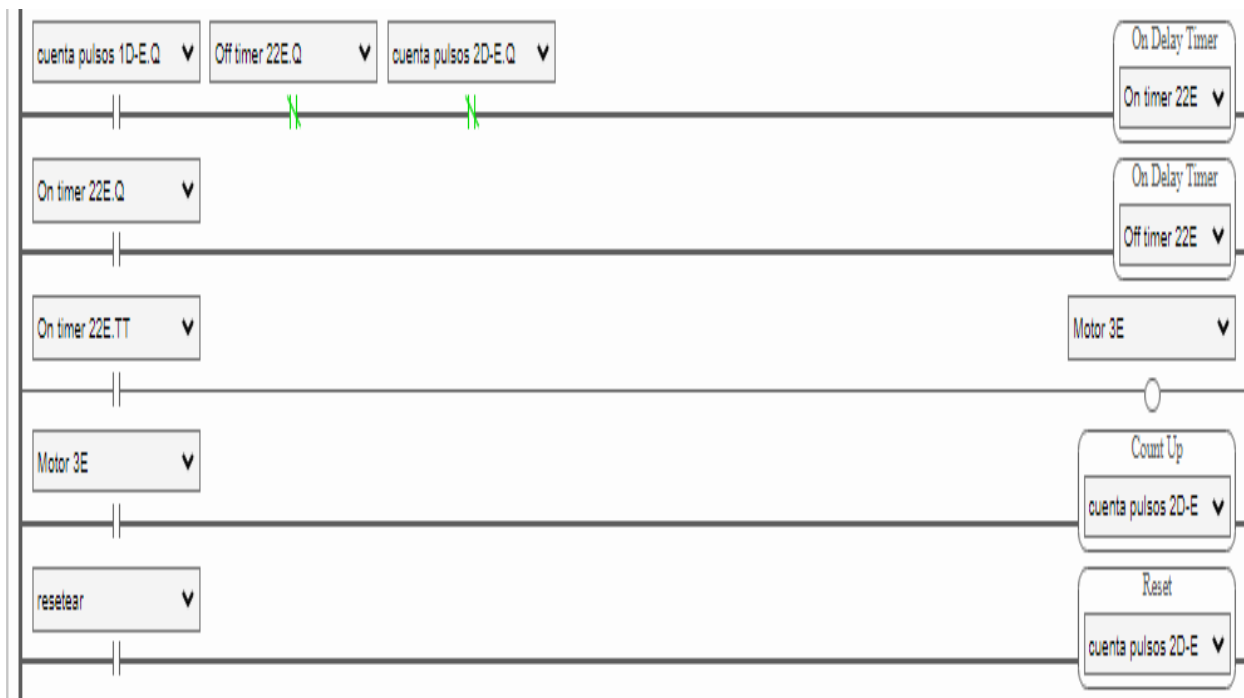
|                                                 |                                                    |                                                    |
|-------------------------------------------------|----------------------------------------------------|----------------------------------------------------|
| Posición 7<br>(Cuenta pulsos, Cuenta pulsos 2C) | Posición 8<br>(Cuenta pulsos 1B, Cuenta pulsos 2C) | Posición 9<br>(Cuenta pulsos 1C, Cuenta pulsos 2C) |
| Posición 4<br>(Cuenta pulsos, Cuenta pulsos 2B) | Posición 5<br>(Cuenta pulsos 1B, Cuenta pulsos 2B) | Posición 6<br>(Cuenta pulsos 1C, Cuenta pulsos 2B) |
| Posición 1<br>(Cuenta pulsos, Cuenta pulsos 2)  | Posición 2<br>(Cuenta pulsos 1B, Cuenta pulsos 2)  | Posición 3<br>(Cuenta pulsos 1C, Cuenta pulsos 2)  |

- Para el movimiento del motor 3 y 4 importa si se ejecutará una tarea de empaque o desempaque, ya que como se puede observar en el algoritmo el orden de activación es distinto.
- Si se realiza una tarea de empaque al terminar el movimiento del motor 2, se iniciará el movimiento del motor 4 con la señal proveniente del cuenta pulsos que estaba en operación con el motor 2 repitiendo el mecanismo de funcionamiento con sus respectivos timers y su cuenta pulsos.



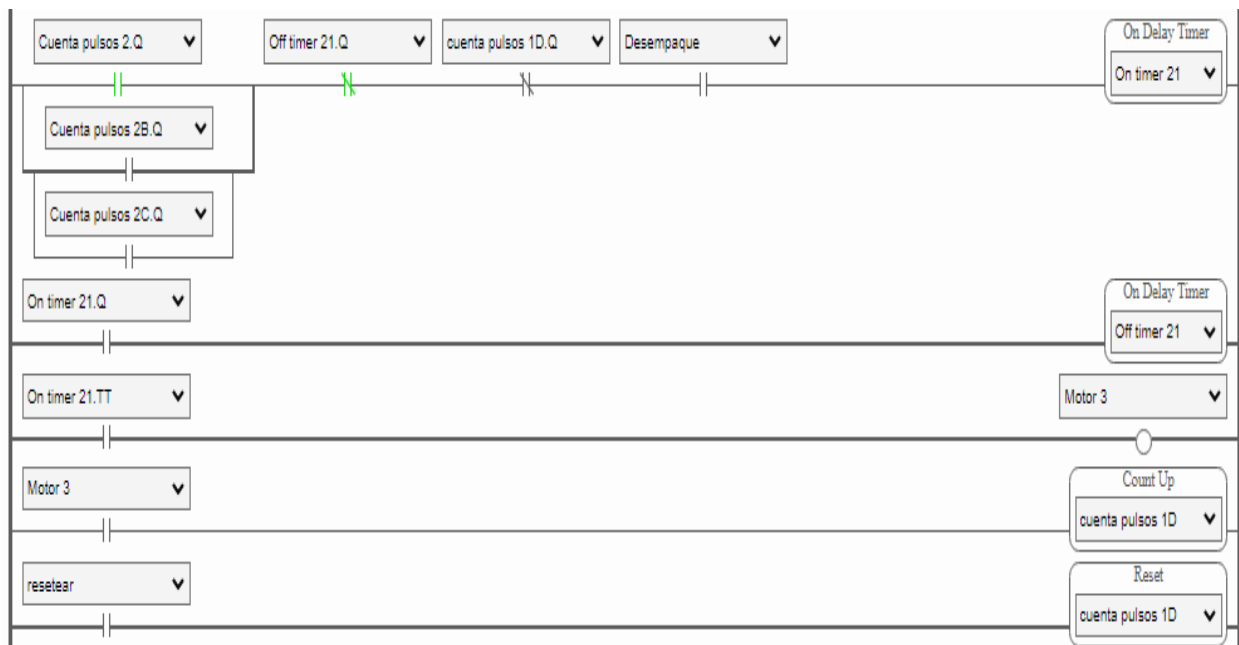
**Figura 5.5.4. Sección del programa para el movimiento del motor 4 de tareas de empaque específicas.**

- Terminado el movimiento del motor 4 se iniciará ahora el movimiento del motor 3 que de igual manera con la señal proveniente del cuenta pulsos del motor 4 dará inicio al mecanismo de funcionamiento al igual con sus respectivos timers y su cuenta pulsos.

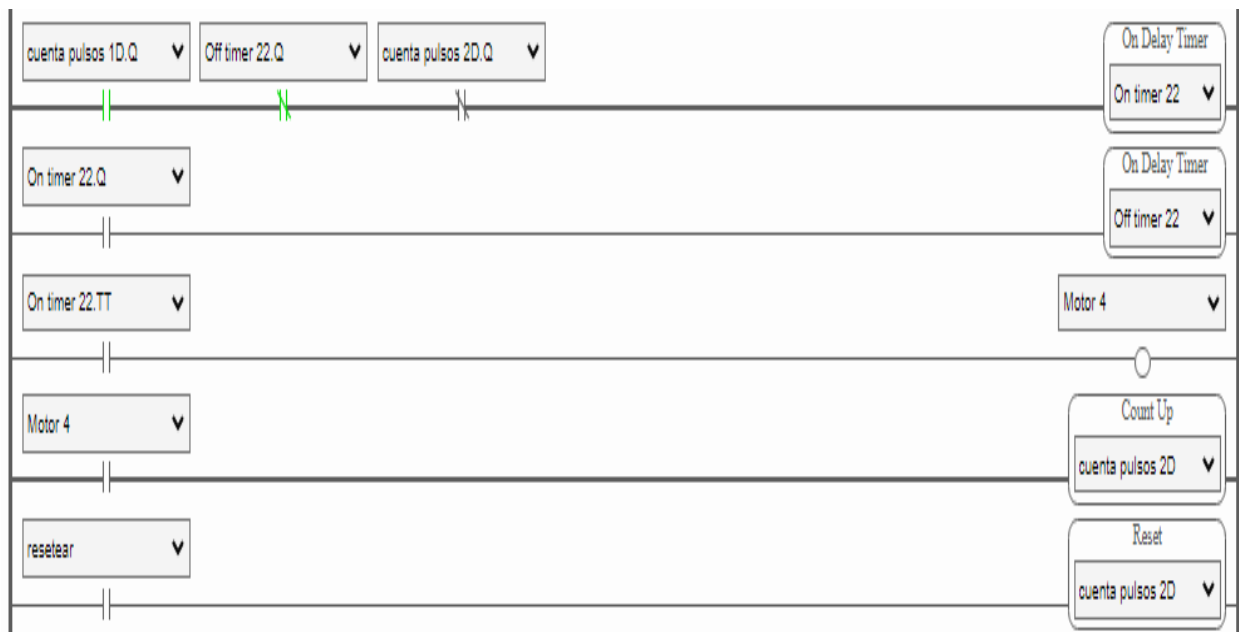


**Figura 5.5.5. Sección del programa para el movimiento del motor 3 de tareas de empaque específicas.**

- Las secuencias de activación del motor 3 y 4 para un desempaque son distintas como se observa en el algoritmo pero el mecanismo de operación es el mismo solo que de igual manera se le asocian distintos timers y su cuenta pulsos a cada motor.



**Figura 5.5.6. Sección del programa para el movimiento del motor 3 de tareas de desempaque específicas.**



**Figura 5.5.7. Sección del programa para el movimiento del motor 4 de tareas de desempaque específicas.**

- Estos pasos solo son para el trayecto de ida, para el trayecto de regreso, el mecanismo de funcionamiento es el mismo solo cambia el orden de la secuencia de activación el cual es inverso al trayecto de ida. Cada motor ahora en retroceso es asociado a un conjunto diferente de timers y cuenta pulsos pero con los mismos parámetros establecidos en cada timer y cuenta pulsos como en los trayectos de ida. Este proceso se muestra de manera completa en el apartado del algoritmo de movimiento de motores de la Sección 5.3.
- Para que él cuenta pulsos una vez finalizado su conteo al valor establecido pueda ser reseteado a cero para un posterior conteo ya sea para una tarea de empaque y/o desempaque, cuenta con una señal de reset el cual se encargará de restablecer el ACC del cuenta pulsos a cero, esta señal de reset es enviada también de manera manual por medio de otro botón único y destinado para realizar esta tarea, en el programa también es representado como un contacto normalmente abierto llamado resetear como se observa en las Figuras de la Sección 5.5.

## Capítulo 6.

### 6.1 Conclusiones.

1. Se consiguió reconocer el proceso requerido para el funcionamiento de un almacén industrial, identificando la serie de pasos que se deben ejecutar para realizar la actividad de almacenar y entregar paquetes de forma estructurada.  
Se logró identificar y caracterizar el proceso necesario para lograr el funcionamiento de un almacén industrial automatizado. Donde se identificaron y estructuraron cada una de las etapas para llevar a cabo las tareas de picking, packing y almacenamiento. Esto dio pauta para lograr la sistematización del proceso y posteriormente plantear una propuesta conceptual de automatización para cada etapa, cubriendo las necesidades mediante herramientas de control.
2. Se identificaron las estructuras físicas con las que debe contar el almacén.  
La propuesta del diseño se basó en criterios del aprovechamiento de espacios, distribuyendo cada componente de forma que se redujeran espacios desaprovechados, mejorando así el uso de espacios tanto como la eficiencia operativa. Se contempló un rack de almacenamiento de nueve espacios para fines de ejemplificación y demostración, cabe mencionar que tanto el diseño mecánico como la cantidad de espacios en el rack son escalables y adaptables conforme a las limitaciones y prestaciones del lugar de implementación.
3. Se identificaron los elementos mecatrónicos necesarios para el funcionamiento automático. Dando lugar a la propuesta de un robot de tipo cartesiano como elemento de manipulación debido a su estructura simple, precisión de movimientos en los tres ejes y sencilla programación y control. El robot propuesto está compuesto por motores a pasos los cuales generan movimientos rotacionales controlados y por medio de mecanismos piñón cremallera se transformaron dichos movimientos a movimientos lineales. Adicionalmente se propusieron bandas transportadoras compuestas por motores de DC para el traslado de los paquetes de una zona a otra, así como el uso de sistemas rodillos los cuales permiten el

traslado de una banda 1 a la banda 2 o de la banda 2 a la bandeja del robot, con la finalidad de facilitar el traslado de los paquetes.

4. Se generaron tanto los algoritmos lógicos para el funcionamiento del sistema de almacenamiento automático, así como los programas adecuados en lenguaje de escalera para los PLCs.

En base a las necesidades presentadas en el proyecto y a las limitaciones de este se desarrollaron dos algoritmos de los cuales el primero permite solucionar la estructura lógica del almacén y el segundo logra controlar los movimientos del robot para alcanzar las posiciones del almacén deseadas tanto para las tareas de empaque como para las de desempaque.

El primer algoritmo permite generar una jerarquía para las tareas a realizar, dando prioridad a las tareas de packing y estableciendo para ellas las restricciones necesarias para su activación, que permitan no realizar tareas innecesarias ni activar más de una tarea a la vez, este algoritmo también permite la activación de un conjunto de dos memorias para permitir establecer una cola de trabajo. En cuanto a la tarea de picking también se establece la lógica para realizar estas tareas en el mismo algoritmo, a estas tareas se les da una menor prioridad para lograr obtener la mayor cantidad de espacios disponibles.

El segundo algoritmo cubre la ejecución de las tareas de picking y packing, es decir, este algoritmo lleva a cabo la actividad que ordene el algoritmo lógico. En base a estos algoritmos se facilitó el desarrollo de los dos programas en lenguaje escalera para los PLCs, cabe mencionar que para fines de implementación el PLC que lleve a cabo la lógica del sistema deberá ser declarado maestro y el PLC que lleve a cabo las ejecuciones de dichas tareas deberá ser declarado esclavo.

5. Es importante recalcar que el presente trabajo constituye únicamente un diseño de concepto para la automatización del proceso de almacenamiento y esta propuesta de diseño se realiza en un entorno académico, por lo cual los elementos considerados se ajustan a limitantes presupuestales, de igual forma al ser un diseño conceptual existen elementos de diseño para los cuales no se desarrollo

una solución completa, tales como la consideración de elementos que permitan garantizar la referencia cero en la posición de home para el robot, este elemento podría resolverse mediante el uso de sensores de fin de carrera para la posición en cuestión, otro punto a considerar para trabajos futuros de implementación es la inclusión de elementos de control para las bandas y elementos neumáticos del sistema dentro del programa esclavo o bien dentro del programa maestro, entro algunos otros elementos visibles en la etapa de implementación.

6. Este proyecto me permitió consolidar aprendizajes desarrollados durante mi formación universitaria, particularmente los obtenidos en las materias como fundamentos y programación de los controladores lógicos programables y automatización de procesos, ambas materias se complementan ya que en automatización de procesos se incentiva a la necesidad de automatizar un proceso de carácter industrial en donde el uso del PLC puede ser crucial para la solución a las necesidades. En la materia de fundamentos y programación de los controladores lógicos programables, se desarrollan habilidades y conocimientos como conocer las partes principales del PLC, sus diversos lenguajes de programación destacando el lenguaje escalera, sus conexiones de entrada, salida y comunicación con otros PLC así como sus aplicaciones y su alcance a nivel industrial.

El desarrollo de este proyecto me presento un reto, ya que no solo requerí aplicar mis conocimientos previamente obtenidos, sino también identificar un problema industrial y dar soluciones desde cero permitiendo así fortalecer mi análisis, ingenio y criterio. Puesto que en las clases en ocasiones se nos presentaban problemas, pero con soluciones definidas y el reto era solo replicar y comprobar la solución con los conocimientos adquiridos. En esta ocasión el proyecto me demando el identificar el problema de manera objetiva y crear la solución sin una guía de por medio para encontrar la respuesta, permitiéndome explorar distintas formas y compararlas entre sí.

En el siguiente link se muestran las pruebas de testeo del programa maestro el cual ejecuta la parte lógica del sistema:

[https://drive.google.com/file/d/1mtp3vIQzHJ\\_SgZJjMlB3iX0fZrcsIlry/view?usp=sharing](https://drive.google.com/file/d/1mtp3vIQzHJ_SgZJjMlB3iX0fZrcsIlry/view?usp=sharing)

En el siguiente link se muestran las pruebas de testeo del programa esclavo el cual ejecuta las tareas solicitadas por el programa maestro:

[https://drive.google.com/file/d/1JBiUSvLTO4488-VQJ5\\_pkcTgTp1XyBi/view?usp=sharing](https://drive.google.com/file/d/1JBiUSvLTO4488-VQJ5_pkcTgTp1XyBi/view?usp=sharing)

## Referencias.

- [1] Wurman, P. R., D'Andrea, R., & Mountz, M. (2008). Coordinating hundreds of cooperative, autonomous vehicles in warehouses. *AI Magazine*, 29(1), 9–20. <https://doi.org/10.1609/aimag.v29i1.2083>
- [2] Gupta, A. (2023). Warehouse automation: Evaluating return on investment and operational efficiency. *Transport Topics*. <https://www.ttnews.com/articles/warehouse-automation>
- [3] MHI – Automated Storage and Retrieval Systems Industry Group. (2020). Add up the savings in three key areas for significant returns on ASRS investment. <https://www.warehouseautomation.org/2020/01/20/add-up-the-savings-in-three-key-areas-for-significant-returns-on-asrs-investment/>.
- [4] Mordor Intelligence. (2024). Automated Storage and Retrieval System Market Size. Disponible en: <https://www.mordorintelligence.com>
- [5] Inven.ai. (2024). Ranking the Top 22 ASRS Companies Globally. Disponible en: <https://www.inven.ai>
- [6] RoboticsTomorrow. (2024). Top Innovations in Warehouse Automation: Automated Storage Systems. Disponible en: <https://www.roboticstomorrow.com>
- [7] Calzado-Girón, D. (2020). La gestión logística de almacenes en el desarrollo de los procesos logísticos. Redalyc.
- [8] Ferreira, B., & Reis, J. (2023). A systematic literature review on the application of automation in logistics. *Logistics*, 7(4), 80.
- [9] Naujokienė, L. (2025). Conditions for increasing the level of automation in logistics processes. *Systems*, 13(7), 608.
- [10] AlMataani, N. A. R., & Ullah, A. (2024). Storage operations in warehouses: A review. *International Journal of Social Sciences and Management Review*, 7(1).
- [11] Julião, B. J. B., Reis, M. S., & Duarte, B. P. M. (2025). A systematic lean-driven framework for warehouse optimization. *Systems*, 13(9), 813. [7] Barrientos, A., Peñin, L.

- [12] Tikwayo, L. N., et al. (2023). Applications of Industry 4.0 technologies in warehouse management. *Logistics*, 7(2), 24.
- [13] Ortiz, C., Quezada, L., & Oddershede, A. (2024). Industrial process management model to improve productivity and reduce waste. *Sustainability*, 16(4), 1606.
- [14] Folgado, F. J., Calderón, D., González, I., & Calderón, A. J. (2024). Review of Industry 4.0 from the perspective of automation and supervision systems: Definitions, architectures and recent trends. *Electronics*, 13(4), 782.
- [15] Burhan, I. B. K. (2024). Automation in industry using programmable logic controllers (PLCs) and SCADA systems. *International Journal of Computer and Communication Network*, 6(1A), 83.
- [16] Bolton, W. (2015). *Programmable logic controllers* (6th ed.). Newnes.
- [17] Petruzella, F. D. (2017). *Programmable logic controllers* (5th ed.). McGraw-Hill Education.
- [18] Siemens AG. (2018). *SIMATIC S7-1200 programmable controller system manual*. Siemens Industry Sector.
- [19] Kuphaldt, T. R. (2021). Relay ladder logic and digital logic functions. En *Lessons in Electric Circuits* (Vol. IV – Digital). Socratic Electronics.
- [20] F., Balaguer, C., & Aracil, R. (1997). *Fundamentos de robótica* (2a ed.). McGraw-Hill.
- [21] Transfer Multisort Elektronik (TME). (2020). *Motor paso a paso – tipos y ejemplos del uso de motores paso a paso*.
- [22] Corona Ramírez, L. G., Abarca Jiménez, G. S., & Mares Carreño, J. (2014). *Sensores y actuadores*. Grupo Editorial Patria
- [23] Angadi, S. V., & Jackson, R. L. (2022). A critical review on the solenoid valve reliability, performance and remaining useful life including its industrial applications. *Engineering Failure Analysis*, 136, 106231.

- [24] Serway, R. A., & Jewett, J. W. (2018). Physics for scientists and engineers (10th ed.). Cengage Learning.
- [25] Hamamatsu Photonics K.K. (2020). Infrared detectors and sensors: Technical information. Hamamatsu Photonics.
- [26] Valdez Salas, B. (2006). Tecnología en la UABC. Miguel Ángel Porrúa.
- [27] Siemens AG. (2024). LOGO! 8 system manual (Manual de sistema). Siemens Industry Online Support
- [28] Delta Electronics, Inc. (2020). DVP-TP70 series touch panel user manual.
- [29] AG Electrónica SAPI de CV. (2024). Sensor de proximidad infrarrojo FC-51 OKY3127 [Archivo PDF].
- [30] UElectronics. (s. f.). Módulo sensor fotoresistor LDR [Página de producto].
- [31] RM Partes. (s. f.). Kit motor paso a paso Nema 23 – 2 Nm + driver. RM Partes.
- [32] Steren. (s. f.). Motor a pasos NEMA 17. Steren.
- [33] UElectronics. (s. f.). A4988 driver para motor a pasos con disipador [Página de producto].
- [34] Festo Didactic GmbH & Co. KG. (2016). Cilindro de simple efecto (152887\_es\_v2.01) [Archivo PDF].
- [35] All About Circuits. (2013). Relay construction and operation. En Electromechanical relays. All About Circuits
- [36] SANDOROBOTICS. (s. f.). Módulo de 1 canal relay – HR0047 [Página de producto].
- [37] Festo Didactic GmbH & Co. KG. (2021). Electroválvula 32 vías (567198\_es\_v2.0) [Archivo PDF].