

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

### **IMPLEMENTACIÓN DE UNA MÁQUINA EXPENDEDORA PARA MASCARILLAS ARTESANALES (ETAPA 2)**

#### **TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN ELECTROMECAÁNICA**

**Byron Fernando Espinosa Ortega**

byron.espinosa@epn.edu.ec

**Steven Fernando Parra Carrillo**

steven.parra@epn.edu.ec

**DIRECTOR: ING. Abraham Ismael Loja Romero, MSc.**

abraham.loja@epn.edu.ec

**CODIRECTOR: ING. Roberto Carlos Rojas Molina, PhD.**

roberto.rojas@epn.edu.ec

**Quito, julio 2021**

# CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por el Sr. Espinosa Ortega Byron Fernando y el Sr. Parra Carrillo Steven Fernando como requerimiento parcial a la obtención del título de TECNÓLOGO EN ELECTROMECAÁNICA, bajo nuestra supervisión:



---

**ING. Abraham Ismael Loja  
Romero, MSC.**  
DIRECTOR DEL  
PROYECTO

---

**ING. Roberto Carlos Rojas Molina,  
PHD.**  
CODIRECTOR DEL PROYECTO

## DECLARACIÓN

Nosotros Espinosa Ortega Byron Fernando con CI: 1720475068 y Parra Carrillo Steven Fernando con CI: 1725650533 declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que hemos consultado las referencias bibliográficas que se incluyen en este documento.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 144 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación – COESC-, somos titulares de la obra en mención y otorgamos una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional.

Entregamos toda la información técnica pertinente, en caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.



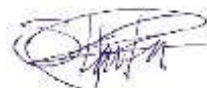
---

**Byron Fernando Espinosa Ortega**

CI: 1720475068

Teléfono: 0992871434

Correo: byron.espinosa@epn.edu.ec



---

**Steven Fernando Parra Carrillo**

CI: 1725650533

Teléfono: 0979283768

Correo: steven.parra@epn.edu.ec

## **DEDICATORIA**

Este proyecto le dedicó especialmente a mi madre Margarita Ortega quien es un gran apoyo en mi vida, pues sin ella no lo habría podido conseguir, a mis hermanos y primas quienes han sabido ser un gran apoyo para que pueda terminar este reto en mi vida, a mis tíos quienes con su experiencia han sabido ser consejeros en mi vida personal y estudiantil y a mi abuelita Casilda Ortega quien ha sabido motivarme y aconsejarme para seguir adelante y poder conseguir los objetivos que me proponga.

Aunque fue muy largo el camino, ustedes muy bien saben el camino que tuve que recorrer para llegar aquí, pero también es muy gratificante haber podido culminar, ya que me llena de satisfacción todo lo que he podido lograr, por lo que este nuevo logro en mi vida se los dedico a ustedes ya que también les pertenece.

Byron Espinosa Ortega

# AGRADECIMIENTO

Le agradezco primeramente a Dios por darme la fuerza y motivación para seguir adelante y poder culminar de la mejor manera esta etapa de mi vida, también a mi madre Margarita Ortega por todo su comprensión, paciencia y cariño que ella me entrega cada día, por darme el mejor ejemplo que una madre puede dar hacia un hijo y por ser la madre que me enseñó a no rendirme sí se presenta algún obstáculo que se me puede presentar en la vida.

A toda mi familia, por brindarme todo su cariño y sus valores que los he sabido acoger de la mejor ya que con humildad, sencillez y responsabilidad se puede conseguir grandes cosas.

Al Ing. Abraham Loja por haber sido una persona comprensible, generosa y sobre todo que haya confiado en nosotros para la elaboración del proyecto, también agradecer al Ing. David Loja por todo su apoyo, sobre todo la predisposición y enseñanza que mantuvo durante la elaboración del proyecto, también a PNUD por la confianza puesto sobre nosotros para la realización del proyecto.

Por último, a la Escuela Politécnica Nacional, que fue mi segundo hogar, donde supe crecer como persona, a romper mis miedos y enfrentarme a las adversidades con dedicación y esfuerzo, gracias por todo EPN.

Byron Espinosa Ortega

# ÍNDICE DE CONTENIDOS

1	Introducción.....	12
1.1	Objetivo general.....	13
1.2	Objetivos específicos.....	13
1.3	Fundamentos.....	13
	Raspberry.....	13
	Python.....	14
	Protocolo de comunicación MDB.....	15
2	Metodología.....	17
2.1	Descripción de la metodología usada.....	17
3.	Resultados y Discusión.....	19
3.1	Requerimientos eléctricos.....	19
	Fuente de alimentación atx 850 (W).....	19
	Fuente de poder 24(V)/5(A) 120(W).....	24
	Fuente de alimentación Raspberry Pi 4, USB-C, 5.1 (V) 3 (A) .....	27
	Regleta de alimentación.....	24
3.2	Algoritmo de control.....	33
	1. Sistema de selección y registro.....	33
	2. Sistema de cobro.....	35
	3. Sistema de expendio del producto.....	36
3.3	Diseño de las placas electrónicas.....	36
	Simulación del circuito para sensores encoder.....	36
	Creación de placa PCB para sensores encoder.....	37
	Simulación del circuito para activación de motores.....	37
	Creación de placa PCB para motorreductores.....	39
3.4	Ensamblaje de los componentes.....	39
	Sistema de expendio.....	39
	Sistema de selección y registro.....	40
	Sistema de cobro.....	41
3.5	Pruebas y Análisis de Resultados.....	41
	Establecimiento de comunicación con protocolo MDB.....	41
	Comprobación de placas electrónicas PCB.....	42
	Comprobación de establecimiento de comunicación serial.....	43
	Calibración de tiempos de expendio del producto.....	45

Comprobación de configuración de precios.....	46
Comprobación de control de ventas.....	47
Comprobación de recarga de monedas.....	47
3.6 Manual de usuario y Mantenimiento.....	48
4 Conclusiones y Recomendaciones.....	49
4.1 Conclusiones.....	49
4.2 Recomendaciones.....	50
5 Referencias Bibliográficas.....	51
ANEXOS.....	54
Anexo 1: Certificado de Funcionamiento.....	55
Anexo 2: Esquema de conexiones.....	56
Anexo 3: Simbología de conexiones norma ANSI-ISA 50.1-1982 (R1992).....	57
Anexo 4: Arquitectura maestro-esclavo de la máquina expendedora.....	58
Anexo 5: Diagrama de flujo de programación en Python.....	59
Anexo 6: Código de programación en IDE de Arduino.....	61

## ÍNDICE DE FIGURAS

<b>Figura 1.1</b> Partes que conforman la Raspberry Pi 4 modelo B.....	13
<b>Figura 1.2</b> Ejemplo de aplicación de librería Tkinter.....	14
<b>Figura 1.3</b> Interfaz MDB.....	15
<b>Figura 1.4</b> Interfaz de conexión MDB .....	16
<b>Figura 3.1</b> Motorreductor.....	20
<b>Figura 3.2</b> Placas de módulos relés.....	20
<b>Figura 3.3</b> Placa para motorreductores.....	21
<b>Figura 3.4</b> Sensor encoder FC-03.....	22
<b>Figura 3.5</b> Placa para sensores encoder.....	23
<b>Figura 3.6</b> Fuente de alimentación atx 850 (W).....	24
<b>Figura 3.7</b> Placa MDB.....	25
<b>Figura 3.8</b> Validador de monedas.....	25
<b>Figura 3.9</b> Lector de billetes.....	26
<b>Figura 3.10</b> Fuente de 24 (V <sub>DC</sub> ).....	27
<b>Figura 3.11</b> Fuente para Raspberry.....	27
<b>Figura 3.12</b> Raspberry Pi 4 modelo B.....	28
<b>Figura 3.13</b> Pantalla táctil para Raspberry.....	28
<b>Figura 3.14</b> Arduino Mega 2560.....	29
<b>Figura 3.15</b> Regleta de alimentación.....	29
<b>Figura 3.16</b> Tira Led.....	30
<b>Figura 3.17</b> Tablet para visualización de publicidad.....	30
<b>Figura 3.18</b> Interruptor termomagnético.....	32
<b>Figura 3.19</b> Portafusible.....	32
<b>Figura 3.20</b> Menú 1.....	33
<b>Figura 3.21</b> Menú 2.....	34
<b>Figura 3.22</b> Submenú 1.....	34
<b>Figura 3.23</b> Submenú 2.....	35
<b>Figura 3.24</b> Submenú 3.....	35
<b>Figura 3.25</b> Simulación en Proteus de circuito para sensores encoder.....	36
<b>Figura 3.26</b> Distribución de elementos en el programa Ares.....	37
<b>Figura 3.27</b> Visualización 3D de la placa de los sensores encoder.....	37
<b>Figura 3.28</b> Simulación en Proteus de circuito para activación de motorreductores.....	38
<b>Figura 3.29</b> Distribución de elementos en el programa Ares.....	39
<b>Figura 3.30</b> Visualización 3D de la placa de motorreductores.....	39



<b>Figura 3.31</b>	Ensamblaje de motorreductores.....	40
<b>Figura 3.32</b>	Ensamblaje de placas electrónicas y fuentes de poder.....	40
<b>Figura 3.33</b>	Ensamblaje de la pantalla táctil.....	40
<b>Figura 3.34</b>	Ensamblaje del sistema de cobro.....	41
<b>Figura 3.35</b>	Comprobación de funcionamiento de placas PCB .....	43
<b>Figura 3.36</b>	Submenú de selección de puertos.....	44
<b>Figura 3.37</b>	Bandeja con producto.....	45
<b>Figura 3.38</b>	Submenú de configuración del producto .....	46
<b>Figura 3.39</b>	Submenú de registro de ventas.....	47
<b>Figura 3.40</b>	Submenú de recarga de monedas.....	48
<b>Figura 3.41</b>	Código QR – Manual de usuario y mantenimiento.....	49

## ÍNDICE DE TABLAS

<b>Tabla 1.31</b>	Comandos para la comunicación con el validador de monedas.....	16
<b>Tabla 1.32</b>	Comandos para la comunicación con el lector de billetes.....	16
<b>Tabla 1.33</b>	Especificaciones técnicas de la interfaz MDB.....	17
<b>Tabla 3.1</b>	Especificaciones eléctricas del motorreductor.....	20
<b>Tabla 3.2</b>	Especificaciones eléctricas de los relés.....	21
<b>Tabla 3.3</b>	Especificaciones eléctricas de la placa para motorreductores .....	21
<b>Tabla 3.4</b>	Especificaciones eléctricas de la placa de sensores.....	22
<b>Tabla 3.5</b>	Especificaciones eléctricas de la placa para los sensores encoder.....	23
<b>Tabla 3.6</b>	Especificaciones eléctricas de la interfaz MDB.....	25
<b>Tabla 3.7</b>	Especificaciones eléctricas del validador de monedas MEI CF-7000.....	25
<b>Tabla 3.8</b>	Especificaciones eléctricas del lector de billetes .....	26
<b>Tabla 3.9</b>	Especificaciones eléctricas de la Raspberry Pi4 modelo B.....	28
<b>Tabla 3.10</b>	Especificaciones técnicas de la pantalla táctil para la Raspberry.....	29
<b>Tabla 3.11</b>	Elementos de consumo eléctrico.....	31
<b>Tabla 3.12</b>	Fuentes de alimentación ( $V_{DC}$ ).....	32
<b>Tabla 3.13</b>	Tabla de verdad para activación de los motorreductores .....	37
<b>Tabla 3.14</b>	Denominaciones para validación de monedas.....	41
<b>Tabla 3.15</b>	Denominaciones para validación de billetes.....	42
<b>Tabla 3.16</b>	Registro de pruebas de placas electrónicas.....	43
<b>Tabla 3.17</b>	Registro de pruebas de comunicación serial. ....	44
<b>Tabla 3.18</b>	Tabla de calibración de los motores. ....	45
<b>Tabla 3.19</b>	Registro de pruebas de configuración de precio.....	46
<b>Tabla 3.20</b>	Registro de pruebas de control de ventas.....	47
<b>Tabla 3.21</b>	Registro de pruebas de recarga de monedas.....	48

## RESUMEN

El presente documento explica el proceso de implementación del sistema eléctrico y electrónico de una máquina expendedora de mascarillas, mediante la utilización de una Raspberry Pi4 modelo B y un Arduino Mega 2560. Estos microcontroladores permiten el control de un monedero, billetero, una pantalla táctil y 20 motorreductores, los cuales se encargan de regular de manera autónoma el sistema de cobro y expendio del producto.

Por lo tanto, se expone los requerimientos eléctricos para la alimentación de los componentes electrónicos con las protecciones respectivas en caso de fallas eléctricas. Los códigos de programación desarrollados en Python y Arduino para la automatización del funcionamiento. El proceso de ensamblaje de cada uno de los componentes en la estructura. Las pruebas de funcionamiento de los sistemas de cobro, selección, expendio y registro de facturación. El manual de usuario y mantenimiento donde se muestra el funcionamiento, el proceso de calibración y cuidado de la máquina expendedora.

De igual manera se presenta en base a los resultados obtenidos; un análisis, resaltando los aspectos más importantes y dificultades que se presentaron en el desarrollo del proyecto. Por ende, también se dan unas sugerencias para facilitar su fabricación. Finalmente, se adjunta como anexos; el certificado de funcionamiento, el esquema de conexiones y su simbología, la arquitectura maestro-esclavo, el diagrama de flujo de la programación en Python y el código en IDE de Arduino.

El desarrollo e implementación del proyecto se realizó para el Programa de las Naciones Unidas para el Desarrollo (PNUD), el cual definirá la ubicación de la máquina para su funcionamiento con el fin de proporcionar otra fuente de comercio y garantizar la seguridad de los comerciantes.

**PALABRAS CLAVE:** Raspberry Pi4 modelo B, Arduino Mega 2560, monedero, billetero, Python.

## ABSTRACT

This document explains the process of implementation of the electrical and electronic system of a mask vending machine, using a Raspberry Pi4 model B and an Arduino Mega 2560. These microcontrollers allow the control of a coin charger, a bill acceptor, a touch screen and 20 gearmotors, which are in charge of the pay and expense system in a regular and autonomous way.

Therefore, the electrical requirements for the power supply of the electronic components with the respective protections in case of electrical failures are exposed. The programming codes developed in Python and Arduino for the automation of the operation. The assembly process of each of the components in the structure. The performance tests of the billing collection, selection, dispensing and registration systems. The user and maintenance manual showing the operation, the calibration process and the care of the vending machine.

In the same way, it is presented based on the results obtained; an analysis, highlighting the most important aspects and difficulties that arose in the development of the project. Therefore, some suggestions are also given to facilitate their manufacture.

Finally, it is attached as annexes; the certificate of operation, the connection diagram and its symbology, the master-slave architecture, the Python programming flow diagram and the Arduino IDE code.

The development and implementation of the project was carried out for the United Nations Development Program (UNDP), which will define the location of the machine for its operation in order to provide another source of commerce and guarantee the safety of traders.

**KEYWORDS:** Raspberry Pi4 model B, Arduino Mega 2560, coins lector, bills lector, Python.

# 1 INTRODUCCIÓN

La aparición de una nueva cepa de coronavirus denominada Covid-19 y los niveles alarmantes de propagación de contagios entre varios países, hizo declarar a la Organización Mundial de la Salud el 11 de marzo del 2020 como pandemia (OMS, 2020). Consecuentemente, Ecuador a través del Comité de Operaciones de Emergencia (COE) Nacional, establecido el 6 de abril del 2020, dispuso una resolución que regula protocolos de bioseguridad, principalmente en el uso obligatorio de mascarilla (Servicio Nacional de Gestión de Riesgos, 2020)

Adicionalmente, se produjo la reducción de fuentes de empleo y la disminución de ingresos en las familias ecuatorianas. El trabajo informal aumento y dentro de estas actividades, apareció la comercialización de productos de bioseguridad, entre estos, las mascarillas elaboradas artesanalmente. El “Programa de las Naciones Unidas para el Desarrollo” (PNUD) interviene con la asistencia técnica a los emprendedores mediante el financiamiento en el desarrollo e innovación de métodos de comercialización para estos productos (PNUD , 2021)

Existe la necesidad de buscar alternativas en la comercialización de mascarillas, por lo que se plantea construir una máquina expendedora que permita la compra y venta de estos productos de manera autónoma. Para esto es necesario el uso y control de un validador de monedas y un lector de billetes que conforman el sistema de cobro. En la automatización se utiliza el protocolo MDB que quiere decir “Multi Drop Bus” el cual codifica y comunica las lectoras con el controlador Raspberry Pi4, el cual luego de recibir la selección del producto por el usuario a través de la pantalla táctil, se comunica y envía la selección al Arduino Mega mediante su puerto serial. Este se encarga del control del sistema de expendio del producto, que se encuentra conformado por los motorreductores y espirales correspondientes.

El desarrollo de la máquina expendedora consiste en analizar los requerimientos eléctricos y electrónicos. Posteriormente se implementa la arquitectura maestro-esclavo entre la Raspberry y el Arduino. Se realiza los códigos en Python e IDE de Arduino para el control de los sistemas de selección, cobro y expendio. Luego se procede al ensamblaje, el cual consiste en la ubicación, anclaje y cableado de los componentes para posteriormente realizar las pruebas correspondientes a cada uno de los sistemas. Finalmente, se expone el manual de usuario y mantenimiento de la máquina expendedora.

## 1.1 Objetivo general

Implementar una máquina expendedora para mascarillas artesanales (etapa 2).

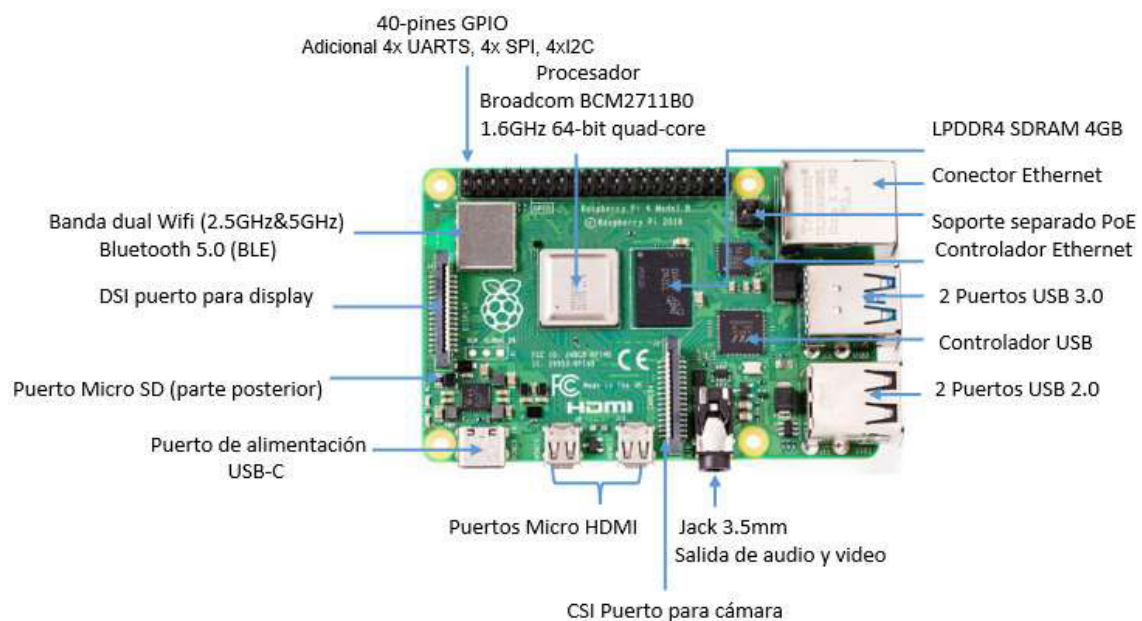
## 1.2 Objetivos específicos

- Requerimientos eléctricos.
- Algoritmo de control.
- Diseño de las placas electrónicas.
- Ensamblaje de los componentes.
- Pruebas y análisis de resultados.
- Manual de usuario y mantenimiento.

## 1.3 Fundamentos

### Raspberry

Es un computador de tamaño reducido cuyo propósito es ser una plataforma que enseña las bases de hardware, los sistemas operativos, la programación y la computación en general. Debido a su versatilidad, simplicidad y sobre todo su bajo costo, además de ser una plataforma de educación para estudiantes, se ha convertido en una plataforma para desarrolladores en todo ámbito. Con estos dispositivos es posible conectar periféricos y cargar un sistema operativo para funcionar como un computador normal o también programarlo para desempeñar funciones específicas a través de componentes electrónicos. Las partes que conforman la Raspberry Pi4 modelo B se pueden apreciar en la Figura 1.1 (Raspberry, 2019).



**Figura 1.1** Partes que conforman la Raspberry Pi 4 modelo B (Raspberry, 2019).

Entre los lenguajes de programación que admite Raspberry están los siguientes: Python, C++, Scratch, Java, etc. De igual manera admite varios sistemas operativos entre los cuales se tiene: Windows 10 IoT Core, Ubuntu Core, Raspbian, Arch Linux ARM, etc. (Pastor, 2019).

## Python

Un lenguaje que se utiliza en varios entornos, ya que posee una licencia de código abierto. La sintaxis de programación es sencilla, legible y elegante que atiende a un conjunto de reglas que hacen muy corta su curva de aprendizaje. En el desarrollo; la automatización de procesos y desempeño de tareas tanto en entorno cliente, como servidor lo hace ideal para trabajar con grandes volúmenes de datos.

Acorde a la aplicación y los componentes que maneja el controlador Raspberry, el código de programación utiliza librerías para el control de los mismos, por lo tanto, a continuación, se expone las librerías utilizadas:

### Librería Tkinter

Tkinter es una biblioteca gráfica Tcl/Tk para el lenguaje de programación Python, como se muestra el ejemplo en la Figura 1.2. Esta librería también se encuentra disponible para otros lenguajes de programación, en donde no es más que una capa de esta librería para el lenguaje Python. Es posible usar Tk en otro lenguaje sin inconvenientes.

```
import tkinter as tk

class Application(tk.Frame):
    def __init__(self, master=None):
        super().__init__(master)
        self.master = master
        self.pack()
        self.create_widgets()

    def create_widgets(self):
        self.hi_there = tk.Button(self)
        self.hi_there["text"] = "Hello World\n(click me)"
        self.hi_there["command"] = self.say_hi
        self.hi_there.pack(side="top")

        self.quit = tk.Button(self, text="QUIT", fg="red",
                               command=self.master.destroy)
        self.quit.pack(side="bottom")

    def say_hi(self):
        print("hi there, everyone!")

root = tk.Tk()
app = Application(master=root)
app.mainloop()
```

**Figura 1.2** Ejemplo de aplicación de librería Tkinter (Python, 2021).

Es una interfaz sencilla que proporciona una perspectiva de lo que se trata el desarrollo de una aplicación, Tkinter permite interactuar con el usuario cuando se requiere el ingreso de datos, capturando la pulsación de teclas, movimientos del mouse, entre otras opciones (Python, 2021).

### **Librería sqlite3**

SQLite es una biblioteca programada en lenguaje C que proporciona una base de datos liviana basada en disco que no requiere un proceso de servidor separado. Permite acceder a la base de datos utilizando una variante no estándar del lenguaje de consulta SQL. Algunas aplicaciones pueden usar SQLite para el almacenamiento interno de datos. También es posible crear un prototipo de una aplicación usando SQLite y luego transferir el código a una base de datos más grande como PostgreSQL u Oracle (Python, 2021).

### **Protocolo de comunicación MDB**

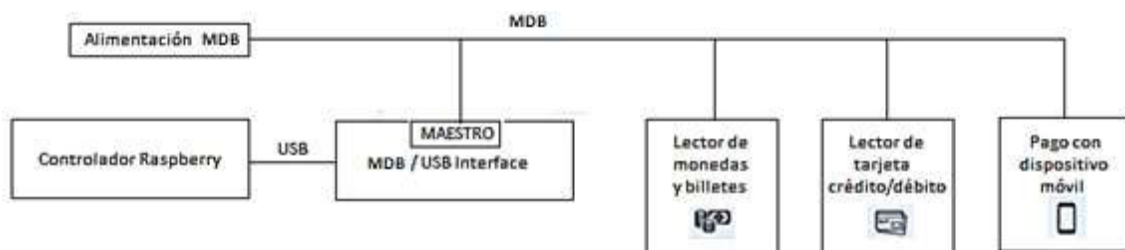
Es una interfaz eléctrica y un protocolo serial y lógico de comunicación para periféricos como validadores de monedas, lectores de billetes, lectores RFID, etc. Este protocolo utiliza 9 bits para su comunicación, el cual no soportan los controladores, por ende, una interfaz MBD como se puede apreciar en la Figura 1.3 es de gran ayuda para conectar y direccionar los periféricos a través de un puerto USB tipo B.



**Figura 1.3** Interfaz MBD.

El protocolo MDB transmite las señales por un bus serie que trabaja con una configuración maestro-esclavo, donde todos los periféricos son esclavos de un controlador maestro. El esquema de comunicación se muestra en la Figura 1.4 a continuación:





**Figura 1.4** Interfaz de conexión MDB (Qibixx, 2017).

Cada periférico tiene una dirección única y un conjunto de comandos, que permiten el acceso al control y comunicación de los mismos, los cuales se detallan a continuación en la Tabla 1.31 (Qibixx, 2017).

**Tabla 1.31.** Comandos para la comunicación con el validador de monedas.

Comando	Descripción
Init	Inicializa el monedero y la secuencia responde si se ha establecido correctamente la comunicación con el dispositivo.
Setup	Obtiene información importante acerca del dispositivo.
Identification	Información acerca de número de serie, modelo, etc.
Tubes	Estado del cartucho de tubos de las monedas.
Coin accepted	Habilita o deshabilita la aceptación de monedas.
Poll	Estado actual del dispositivo, con este comando se tiene que realizar cada 1 a 4 segundos.

En cambio, para la comunicación del lector de billetes los comandos se muestran en la Tabla 1.32 a continuación:

**Tabla 1.32.** Comandos para la comunicación con el lector de billetes.

Comando	Descripción
Init	Inicializa el dispositivo y la secuencia responde si se ha establecido correctamente la comunicación con el dispositivo.
Setup	Obtiene información importante acerca del dispositivo.
Identification	Información acerca de número de serie, modelo, etc.
Tubes	Estado del cartucho de tubos de las monedas.
Bill accepted	Habilita o deshabilita la aceptación de billetes.
Security	Verifica la validez del billete.
Poll	Estado actual de dispositivo, con este comando se tiene que realizar cada 1 a 4 segundos.

Por lo tanto, dichos comandos permiten el control y regulación del funcionamiento de las lectoras acorde a las necesidades del sistema de cobro y dependiendo el tipo de moneda que maneje el lugar donde estas funcionan.

Finalmente, la interfaz MDB posee las siguientes especificaciones técnicas, las cuales se muestran en la Tabla 1.33 (Maya Vending, 2014).

**Tabla 1.33.** Especificaciones técnicas de la interfaz MDB.

Especificación	Descripción
Voltaje de entrada	24 (V)
Voltaje de salida	24 – 36 (V)
Conexiones	RS232
	USB
Leds	3 indicadores de funcionamiento

## 2 METODOLOGÍA

### 2.1 Descripción de la metodología usada

Bajo las necesidades del proyecto, éste se fundamenta en una investigación de tipo explicativa, basada en la dificultad social para desempeñar actividades cotidianas causada por la presencia del virus Covid-19. Por ende, a través de la innovación tecnológica y aplicación de conocimientos relacionados directamente con la carrera de Tecnología en Electromecánica, permite el diseño, selección de componentes, conexión eléctrico-electrónico y programación de los sistemas de selección, cobro y expendio que conforman la máquina expendedora de mascarillas.

Por lo que inicialmente se determinó los voltajes de funcionamiento de los componentes para poder agruparlos y seleccionar las fuentes de alimentación que proporcionen el acondicionamiento y el consumo respectivo. Se realizaron grupos de acuerdo a dos tipos: una que proporciona 5 y 12 (V) con una potencia de 850 (W) para las placas electrónicas, sensores y motores; y otra fuente que proporciona 24 (V) y 5 (A) para el validador de monedas, lector de billetes y la interfaz MDB. De igual manera se analizó las protecciones para proteger el sistema eléctrico, desde la alimentación principal hasta la máquina mediante un interruptor termomagnético de 20 (A), y dentro de la máquina a través de un fusible de 2.5 (A).

Posteriormente para el control de estos componentes, se desarrolló el algoritmo mediante el lenguaje de programación Python, compatible con el controlador Raspberry Pi4 modelo B para el sistema de cobro, selección del producto y registro de facturación. Sin embargo, para el sistema de expendio del producto, el cual consiste en el control de los motorreductores y sensores, se utilizó el microcontrolador Arduino Mega. Por lo

tanto, teniendo en cuenta las posibles situaciones que se pueden presentar en el funcionamiento de la máquina, se estableció el siguiente orden para el desempeño del algoritmo:

- Para acceder al uso de la máquina expendedora se lo hace a través de la pantalla táctil, se realiza una configuración previa a su desempeño autónomo; la cual consiste en la asignación de puertos en la Raspberry, configurar los precios de los productos, llenar el inventario según la mercadería almacenada en los espirales, y cargar de monedas en el sistema de cobro para que pueda dar vuelto. Y así la máquina quede operativa.
- El funcionamiento consiste en que el usuario escoja previamente el producto, proceda al ingreso del dinero según el producto seleccionado, y la máquina realice el expendio del producto y entregue el cambio correspondiente si es necesario.

El funcionamiento del validador de monedas MEI CF-7000 consiste en contabilizar el dinero disponible en su cartucho de tubos para las monedas, el cual debe ser abastecido previamente para poder entregar el vuelto (en caso de que el dinero ingresado por el usuario sobrepase el precio del producto). Por el contrario, en el lector de billetes MEI 2000 se configura para aceptar los billetes según las cantidades establecidas previamente en el código de programación, las cuales están sincronizados entre sí para procesar el cobro.

Luego de que el pago haya sido receptado y aceptado se procede al control del sistema de expendio del producto. El cual consistió en comunicar la Raspberry Pi4 modelo B con el Arduino Mega para el control de los 20 motorreductores; el cual, a través del puerto serial la Raspberry selecciona directamente el puerto al que está conectado el motorreductor en el Arduino. Y mediante la placa de relés (uno por cada motor), aísla la etapa de control de la etapa de potencia, permitiendo girar al motorreductor durante el tiempo programado previamente hasta que el producto caiga y pueda ser retirado por el cliente.

Para el ensamblaje de los componentes en la estructura de la máquina expendedora, se realizó una planificación previa con la etapa mecánica para determinar la ubicación y dimensionamiento de los componentes electrónicos. Además, se definió el diseño del acople para el motorreductor y el espiral el cual permite el expendio del producto. Luego de montar los componentes en sus lugares respectivos se procedió a la realización de las conexiones, el cual consistió en cablear, ordenar e identificar las señales de control y alimentación, en donde se añadió conectores molex para que el desmontaje y montaje sea más cómodo en caso de que se requiera cambiar o revisar algún componente.

Posteriormente para la etapa de pruebas de funcionamiento de los sistemas de cobro, expendio y auxiliares en general, se realizaron por etapas mientras se realizó el montaje, para finalmente realizar la prueba con todos los sistemas conectados y realizar el cobro de todos los productos. Se programó el tiempo de encendido de cada motor para que el sistema detecte cuando ya se terminó el inventario de producto, y de esta manera la máquina funcione correctamente por largos periodos de tiempo.

Finalmente, se elaboró un video explicativo con el funcionamiento de la máquina y su respectivo mantenimiento, el cual consistió en la calibración de los motores, el retiro y abastecimiento de monedas y billetes, conjuntamente con los lineamientos para realizar las actividades de mantenimiento preventivo y correctivo usando el formato de registro correspondiente.

### **3. RESULTADOS Y DISCUSIÓN**

#### **3.1 Requerimientos eléctricos**

La alimentación de la máquina expendedora y según la ubicación aleatoria que puede tener, se la realiza según la tensión estándar en Ecuador provista por la norma NTE INEN 3098 para voltajes normalizados, la cual especifica un voltaje monofásico de 110 (V<sub>AC</sub>) a 60 (Hz) presente en los tomacorrientes a nivel nacional. Sin embargo, el resto de los componentes trabajan a tensiones menores, por lo que fue necesario adquirir fuentes que regulen estos voltajes para que funcionen correctamente y así evitar averías. En el Anexo 2 se muestra el esquema de conexiones de la máquina expendedora.

La simbología de las conexiones se la realizó bajo la norma ANSI, es decir, del Instituto Nacional Estadounidense de Estándares. Referente a señales electrónicas ISA 50.1-1982 (R1992) la cual se adjunta en el Anexo 3 (Ansi, 1992).

#### **Fuente de alimentación atx 850 (W):**

Para la selección de la fuente se consideraron las potencias de los elementos conectados a éste, entre los cuales se tiene:

- **Motores reductores:** Se encuentran distribuidos 5 por bandeja, sumando un total de 20 unidades, el modelo del motorreductor se puede apreciar en la Figura 3.1. Las especificaciones eléctricas (Nbleison, 2021), se muestran en la Tabla 3.1 a continuación:



**Figura 3.1** Motorreductor.

**Tabla 3.1.** Especificaciones eléctricas del motorreductor.

Especificación	Descripción
Torque	19 (kgf.cm)
Velocidad	30 (RPM)
Corriente	280 (mA)
Voltaje directo	12 (V)
Potencia	3.5 (W)

Por lo tanto, debido a que la maquina cuenta con 20 motorreductores distribuidos en 4 bandejas de 5 cada uno, se calculó la potencia total de todos los motorreductores como se muestra a continuación:

$$Pt1 = Pm \times Tm$$

**Ecuación 3.1** Potencia de consumo de los 20 motorreductores.

Donde:

- Pm : 3.5 (W) Potencia del motorreductor
- Tm : 20 Cantidad de motorreductores
- Pt1 : Potencia total motorreductores

Usando la **¡Error! No se encuentra el origen de la referencia.** se obtiene:

$$Pt1 = 70 (W)$$

- **Placas de módulos relés:** Permite la activación de los motorreductores a 12 (V<sub>DC</sub>), según la señal de encendido recibida por el Arduino. Están distribuidos con un módulo de 16 y otro de 4 relés como se observa en la Figura 3.2, sus especificaciones eléctricas (Okystar, 2019), se muestran en la Tabla 3.2 a continuación:



**Figura 3.2** Placas de módulos relés.

**Tabla 3.2.** Especificaciones eléctricas de los relés.

Especificación	Descripción
Voltaje directo	12 (V)
Corriente	60 (mA)
Potencia	0.72 (W)

De igual manera, se calculó la potencia de los 20 relés que corresponden a cada motorreductor, como se muestra a continuación:

$$Pt2 = Pr \times Tr$$

**Ecuación 3.2** Potencia de consumo de los 20 relés.

Donde:

- Pr : 0.72 (W) Potencia
- Tr : 20 Cantidad de relés
- Pt2 : Potencia total relés

Por lo tanto, entre los 20 se tiene una potencia total de:

$$Pt2 = 14.4 \text{ (W)}$$

- **Placa demultiplexora para activación de motorreductores:** Permite el control de los 20 motorreductores a través de 5 bits, es decir, el controlador envía un número binario de 4 bits y 1 enable para ser decodificados por la placa (ver Figura 3.3) y activar el motorreductor seleccionado, con el fin de activar uno a la vez. Las especificaciones eléctricas de los integrados utilizados (Fairchild, 2012) se muestra en la Tabla 3.3 a continuación y su diseño respectivo en la sección correspondiente.



**Figura 3.3** Placa para motorreductores.

**Tabla 3.3.** Especificaciones eléctricas de la placa para motorreductores.

Especificación	Descripción
Voltaje directo	5.2 (V <sub>DC</sub> )
Corriente	1.57 (A)

Por ende, se procedió al cálculo de la potencia de consumo de la placa como se muestra a continuación:

$$Pt3 = Vdc \times I$$

**Ecuación 3.3** Potencia de consumo de la placa de motorreductores.

Donde:

- Vdc : 5.2 (V) Voltaje directo
- I : 1.57 (A) Corriente
- Pt3 : Potencia total placa motorreductores

Por lo tanto, usando la ecuación 3.3 se obtiene:

$$Pt3 = 8.2 \text{ (W)}$$

La placa de los motorreductores solo activará una salida a la vez, por lo que la potencia especificada es la total que consume.

- **Sensores encoder FC-03:** Permite detectar y contabilizar los giros de los motorreductores para que el Arduino pueda controlar el giro de estos. El sensor se observa en la Figura 3.4 y sus especificaciones eléctricas (Junye, 2006) en la Tabla 3.4 a continuación:



**Figura 3.4** Sensor encoder FC-03.

**Tabla 3.4.** Especificaciones eléctricas de la placa de sensores.

Especificación	Descripción
Voltaje directo	5 (V)
Corriente	50 (mA)
Potencia	0.25 (W)

Por cada motorreductor se tiene un sensor encoder, por lo tanto, la potencia total que consume se calcula a continuación:

$$Pt4 = Ps \times Ts$$

**Ecuación 3.4** Potencia total de consumo de los sensores encoder.

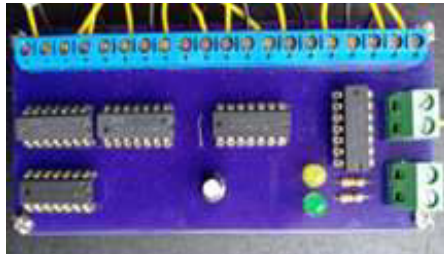
Donde:

- P<sub>s</sub> : 0.25 (w) Potencia del sensor
- T<sub>s</sub> : 20 cantidad de sensores encoder
- P<sub>t4</sub> : Potencia total sensores encoder

La potencia total de consumo de los sensores encoder es:

$$P_{t4} = 5 \text{ (W)}$$

- **Placa de recepción de señales de sensores encoder:** Permite recibir y acondicionar las señales digitales de todos los sensores, los cuales activan una salida que permite el control del giro del motor seleccionado. La placa se aprecia en la Figura 3.5, las especificaciones eléctricas (Texas Instrument, 2009) de los integrados utilizados se muestran en la Tabla 3.5 a continuación; y su diseño respectivo en la sección correspondiente.



**Figura 3.5** Placa para sensores encoder.

**Tabla 3.5.** Especificaciones eléctricas de la placa para los sensores encoder.

Especificación	Descripción
Voltaje directo	5.5 (V)
Corriente	15 (mA)

El cálculo de la potencia de consumo de la placa se muestra a continuación:

$$P_{t5} = V_{dc} \times I$$

**Ecuación 3.5** Potencia de consumo de la placa de los sensores encoder.

Donde:

- V<sub>dc</sub> : 5.5 (V) Voltaje directo
- I : 15 (mA) Corriente
- P<sub>t5</sub> : Potencia total placa sensores encoder

Por lo tanto, usando la ecuación 3.5 se obtiene:

$$P_{t5} = 82.5 \text{ (mW)}$$



La placa de los sensores solo activará una salida a la vez, por lo que la potencia especificada es la total que consume.

Una vez obtenida todas las potencias de consumo se realiza la sumatoria correspondiente:

$$P_{totalE} = P_{t1} + P_{t2} + P_{t3} + P_{t4} + P_{t5}$$

**Ecuación 3.6** Potencia total de consumo del sistema de expendio.

Donde:

- Pt1 : 70 (W) Potencia total motorreductores
- Pt2 : 14.4 (W) Potencia total relés
- Pt3 : 8.2 (W) Potencia total placa motorreductores
- Pt4 : 5 (W) Potencia total sensores encoder
- Pt5 : 82.5 (mW) Potencia total placa sensores encoder
- PtotalE : (W) Potencia total de consumo del sistema de expendio

Según la ecuación 3.6 la potencia total de consumo del sistema de expendio es:

$$P_{totalE} = 97.68 \text{ (W)}$$

Por lo tanto, se requiere una fuente mínima de 100 (W), para lo cual, se seleccionó la fuente comercial atx de 850 (W) (ver Figura 3.6.) debido a que proporciona 5 y 12 (V<sub>DC</sub>) al mismo tiempo y también por la disponibilidad en el mercado, el precio y las protecciones que incluye frente a cortocircuitos, sobrecargas o variaciones de voltaje. De igual manera, se consideró para posibles ampliaciones en mejoras de la máquina expendedora.



**Figura 3.6** Fuente de alimentación atx 850 (W).

### **Fuente de poder 24(V)/5(A) 120 (W)**

Para la selección de la fuente del sistema de cobro se consideró las especificaciones eléctricas de cada uno de los componentes, los cuales se describen a continuación:

- **Interfaz MDB:** Permite la comunicación entre las lectoras y el controlador Raspberry, además se encarga de regular el voltaje para el monedero y billetero, (ver Figura 3.7). Las especificaciones eléctricas (Qibixx, 2017) se muestran en la Tabla 3.6 a continuación:



**Figura 3.7** Placa MDB.

**Tabla 3.6.** Especificaciones eléctricas de la interfaz MDB.

Especificación	Descripción
Voltaje de entrada	24 (V)
Voltaje de salida	24/36 (V)
Corriente	2 (A)
Potencia	48/68 (W)

Al ser un modelo universal también permite conectar otros medios pago, como cobro con tarjeta o recicladores de billetes.

- **Validador de monedas MEI CF-7000:** Recpta las monedas para procesar el costo del producto a través de la Raspberry Pi4 y devolver el cambio si es necesario, (ver Figura 3.8). Las especificaciones eléctricas (MEI, 2006) se muestran en la Tabla 3.7 a continuación:



**Figura 3.8** Validador de monedas.

**Tabla 3.7.** Especificaciones eléctricas del validador de monedas MEI CF-7000.

Especificación	Descripción
Voltaje de entrada	24 (V)
Corriente	2.7 (A)
Potencia del monedero (Pm)	65 (W)

Al ser un gestor de cobro, este posee distintas validaciones para varios tipos de monedas de otros países, el cual se configura mediante programación.

- **Lector de billetes MEI 2000:** Complementa al sistema de cobro, el cual recepta el dinero en billetes según la aceptación de las denominaciones configurada previamente. Funciona sincronizado con el monedero en el procesamiento del cobro y costo del producto a expender, sus especificaciones eléctricas se muestran en la Tabla 3.8 y su imagen en la Figura 3.9 a continuación:



**Figura 3.9** Lector de billetes.

**Tabla 3.8.** Especificaciones eléctricas del lector de billetes.

Especificación	Descripción
Voltaje de entrada	24 (V)
Corriente	625 (mA)
Potencia del billetero (Pb)	15 (W)

Una vez obtenida todas las potencias de consumo de los componentes del sistema de cobro se realizó la sumatoria correspondiente:

$$P_{totalC} = P_m + P_b$$

**Ecuación 3.7** Potencia total de consumo del sistema de cobro.

Donde:

- P<sub>m</sub> : 65 (W) Potencia del monedero
- P<sub>b</sub> : 15 (W) Potencia del billeteero
- P<sub>totalC</sub> : Potencia total de consumo del sistema de cobro

Usando la ecuación 3.7, la potencia de consumo del sistema de cobro es:

$$P_{totalC} = 80 \text{ (W)}$$

Por lo tanto, se requiere una fuente que disponga de una potencia mínima de 80 (W), para lo cual se seleccionó la fuente comercial de 24 (V<sub>DC</sub>) / 5 (A) a 120 (W), debido a la disponibilidad comercial en el mercado, (ver Figura 3.10).



**Figura 3.10** Fuente de 24 (V<sub>DC</sub>).

#### **Fuente de alimentación Raspberry Pi 4, USB-C, 5.1 (V) 3 (A)**

Es la fuente de alimentación independiente, específicamente para el funcionamiento de la Raspberry Pi4 modelo B, como se muestra en la Figura 3.11.



**Figura 3.11** Fuente para Raspberry (Raspberry, 2019).

Esta proporciona 5 (V<sub>DC</sub>) y posee una conexión USB tipo C compatible con la raspberry la cual se describe a continuación:

- **Raspberry Pi 4 modelo B:** Es el controlador maestro, (ver Figura 3.12) que se encarga de recibir y enviar los comandos a los periféricos conectados a éste, es decir permite la comunicación y control entre los sistemas de selección, cobro y expendio del producto, las especificaciones eléctricas (Raspberry, 2019) se muestran en la Tabla 3.9 a continuación:



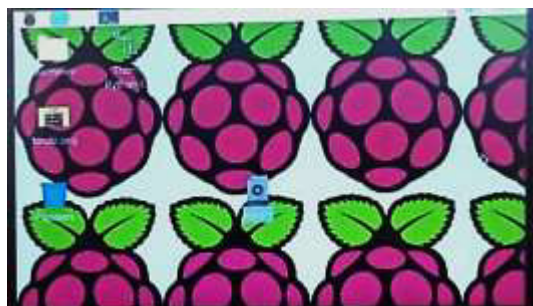
**Figura 3.12** Raspberry Pi 4 modelo B.

**Tabla 3.9.** Especificaciones eléctricas de la Raspberry Pi4 modelo B.

Especificación	Descripción
Voltaje de alimentación	5 (V)
Voltaje de salida	3.3 (V)
Corriente máxima por pin GPIO	16 (mA)
Corriente máxima	2.4 (A)

Para la conexión de componentes electrónicos en los pines GPIO se debe tener en cuenta las especificaciones mencionadas anteriormente, ya que no posee protecciones internas, si sobrepasa dichos valores, puede generar un daño permanente.

- **Pantalla táctil de 5" para Raspberry:** Es el periférico de entrada que permite el control de la Raspberry, una vez iniciado el código de programación, la pantalla despliega los submenús para la configuración previa a la puesta en funcionamiento de la máquina expendedora. La pantalla táctil se aprecia en la Figura 3.13 y las especificaciones técnicas (Dongfeng Liangdian, 2019) se muestran en la Tabla 3.10 a continuación:



**Figura 3.13** Pantalla táctil para Raspberry.

**Tabla 3.10.** Especificaciones técnicas de la pantalla táctil para la Raspberry.

Especificación	Descripción
Voltaje de alimentación	5 (V)
Potencia	1.3 (W)
Resolución	800xRGBx480
Conexión	USB HD MI interface
Dimensiones	109.6x66.8 (mm)

La pantalla táctil va montada junto a la Raspberry, por lo que es necesario añadir un ventilador para ayudar a la refrigeración de ambos componentes.

- **Arduino Mega 2560:** Es el microcontrolador que funciona a 5 (V<sub>DC</sub>) asignado como esclavo, el cual recibe la información de selección desde la Raspberry para la activación del motorreductor, el giro de la espiral y por ende el despacho del producto. Este modelo se caracteriza por la cantidad de pines disponibles (ver Figura 3.14), los cuales permiten conectar los 20 motorreductores.



**Figura 3.14** Arduino Mega 2560.

### Regleta de alimentación

Se encarga de suministrar los 110 (V<sub>AC</sub>), (ver Figura 3.15) y energizar todas las fuentes y componentes extra de la máquina, los cuales se detallan a continuación:



**Figura 3.15** Regleta de alimentación.

- **Tira Led:** Proporciona iluminación blanca a los productos en la parte frontal de la máquina expendedora, el cual funciona a 12 (V<sub>DC</sub>) gracias al transformador que trae incorporado, el cual rectifica el voltaje proporcionado por la regleta, (ver Figura 3.16).



**Figura 3.16** Tira Led.

- **Tablet de 7”:** Funciona a 5 (V<sub>DC</sub>) y es utilizada como pantalla para mostrar publicidad, (ver Figura 3.17).



**Figura 3.17** Tablet para visualización de publicidad.

- **Interruptor termomagnético C20:** Protege la alimentación de la máquina frente a descargas o fallas en el sistema eléctrico del sitio donde se encuentra conectada, para lo cual es necesario obtener el consumo eléctrico de los elementos, como se muestra en la Tabla 3.11.

**Tabla 3.11** Elementos de consumo eléctrico.

Elemento	Voltaje (V <sub>DC</sub> )	Corriente (A)	Potencia (W)
Fuente de alimentación placa MDB	24	3	120
Fuente de alimentación atx 850 (W)	12-5-3,3	5	850
Fuente de Raspberry	5	3	15
Cinta led	12	2	25
Fuente de la tablet	5	2	10
<b>Total</b>	-	<b>15</b>	<b>1020</b>

Por ende, una vez obtenido el consumo eléctrico de la máquina expendedora se realizó el cálculo de la corriente, la cual viene a ser la corriente de cortocircuito:

$$I = P/V$$

**Ecuación 3.8** Corriente de cortocircuito. (Capella, 2003)

Donde:

- P : 1020 (W) Potencia total
- V : 110 (V) Voltaje en alterna
- I : (A) Corriente de cortocircuito

Usando la Ecuación 3.8 se obtiene:

$$I = 9,27 \text{ (A)}$$

Con la corriente de cortocircuito se calculó la corriente del interruptor termomagnético como se muestra a continuación:

$$I_{tm} = I * 1,25$$

**Ecuación 3.9** Corriente del interruptor termomagnético. (Capella, 2003)

Donde:

- I : 9,27 (A) Corriente de cortocircuito
- 1,25 : factor de dimensionamiento
- I<sub>tm</sub> : Corriente del interruptor termomagnético

En base al valor calculado empleando las ecuaciones Ecuación 3.8 y Ecuación 3.9, se obtiene la corriente del sistema, es decir del interruptor termomagnético:

$$I_{tm} = 11,59 \text{ (A)}$$

Por lo tanto, al obtener la corriente máxima de consumo se procede a seleccionar el interruptor termomagnético de 20 (A), como se muestra en la Figura 3.18, de acuerdo con la disponibilidad en el mercado ecuatoriano.





**Figura 3.18** Interruptor termomagnético.

- **Portafusible:** Es la protección interna para las fuentes de alimentación de los componentes electrónicos (ver Figura 3.19), los cuales se detallan en la Tabla 3.12.

**Tabla 3.12** Fuentes de alimentación ( $V_{DC}$ ).

Elemento	Voltaje ( $V_{DC}$ )	Corriente (A)	Potencia consumo (W)
Fuente de poder MBD	24	5	120
Fuente de alimentación atx 850 (W)	12/5/3,3	5	97.68
Fuente de Raspberry	5	3	15
<b>Total</b>	-	-	<b>232.68</b>

De igual manera, se procede a calcular la corriente de cortocircuito como se muestra a continuación:

$$I = P/V$$

**Ecuación 3.10** Corriente de cortocircuito.

Donde:

- P : 232.68 (W) Potencia total
- V : 110 (V) Voltaje en alterna
- I : (A) Corriente de cortocircuito

Usando la Ecuación 3.8 se obtiene:

$$I = 2.11 \text{ (A)}$$

Por lo tanto, de acuerdo con los valores comerciales se seleccionó un fusible de 2.5 (A).



**Figura 3.19** Portafusible.

## 3.2 Algoritmo de control

Entre los componentes electrónicos se cuenta con un controlador Raspberry Pi4 modelo B y un microcontrolador Arduino Mega 2560, los cuales gobiernan el funcionamiento de la máquina expendedora a través de un protocolo de comunicación serial, es decir la transmisión de datos se realiza enviando bytes (conjunto de 8 bits) mediante sus puertos USB. Con el fin de implementar la arquitectura maestro-esclavo (ver Anexo 3) entre la Raspberry y el Arduino, respectivamente; y así distribuir las áreas de control los cuales se detallan a continuación:

### 1. Sistema de selección y registro

Es el área de control diseñada para interactuar con el usuario y el cliente de la máquina expendedora, por tal motivo se diferencian los menús de configuración y registro, de los de selección y expendio respectivamente. A continuación, se detallan los menús de interacción:

#### Menú 1: Selección de puertos para periféricos

El Arduino y la interfaz MDB son los periféricos gobernados por la Raspberry, por esta razón al momento de encender la máquina se despliega el menú 1, como se observa en la Figura 3.20, el cual se encarga de seleccionar el puerto para la conexión de correspondiente a cada uno, con el fin de evitar conflictos en la ejecución del programa.

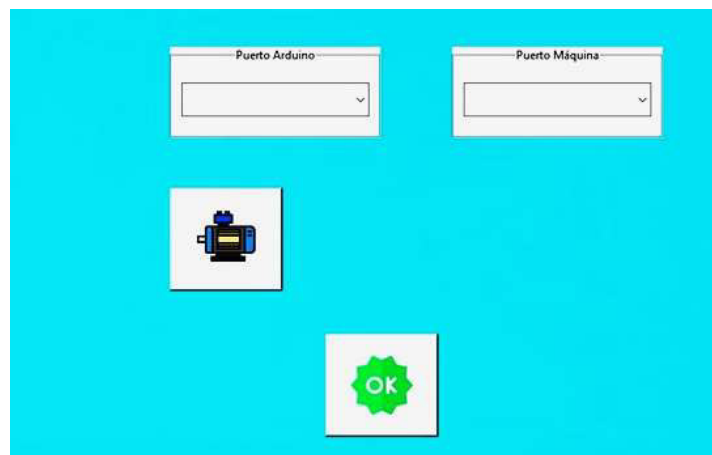
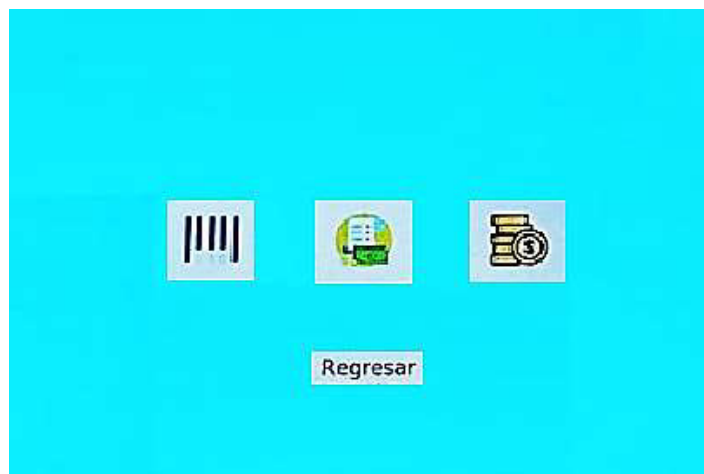


Figura 3.20 Menú 1.

#### Menú 2: Control y registro

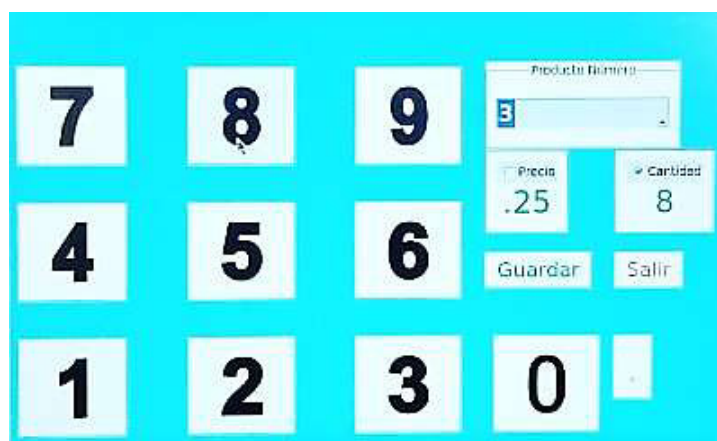
Posteriormente se despliega el menú 2 de control y registro como se muestra en la Figura 3.21, que permite desplazarse entre los submenús según la necesidad de la máquina y del usuario los cuales se muestran a continuación:



**Figura 3.21** Menú 2.

### **Submenú 1: Registro de precios y control de inventario**

Para configurar los precios de los productos y el inventario con el que la máquina funciona, se lo hace a través del submenú 1 como se muestra en la Figura 3.22. En caso de que no se ingrese la cantidad de inventario el programa no permitirá la venta del producto, ya que la máquina asume que no posee producto.



**Figura 3.22** Submenú 1.

### **Submenú 2: Facturación**

El submenú de facturación, (ver Figura 3.23) controla la cantidad en dinero y producto de las ventas que se ha producido en el periodo de tiempo de funcionamiento, con el fin de analizar la rentabilidad de los productos y la ubicación de la máquina.

Fecha	Precio	Producto
2021-04-18	3	3
2021-04-18	3	3
2021-04-18	0.05	9
2021-04-18	0.05	9
2021-04-18	1.25	4
2021-04-18	1.25	4
2021-04-18	1.25	4
2021-04-18	1.75	2
2021-04-18	1.75	2
2021-04-18	1.25	4
2021-04-18	5	4
2021-04-18	1.25	7
2021-04-18	5	4
2021-04-18	5	4
2021-04-18	1	2
2021-04-18	1	2

Facturación Mensual

Regresar

\$ 47.66

**Figura 3.23** Submenú 2.

### Submenú 3: Recarga de monedas

Para que la máquina pueda dar el vuelto en caso de que el cliente ingrese una cantidad mayor al precio, el validador de monedas debe contar con dinero en sus contenedores, para lo cual es necesario extraer el validador y hacerlo directamente, por tal motivo se implementó el submenú 3 (ver Figura 3.24) permitiendo recargar las monedas desde el compartimento frontal de la máquina.



**Figura 3.24** Submenú 3.

El sistema de selección y registro se desarrolló en Python, por ende su diagrama de flujo se adjunta en el Anexo 5.

## 2. Sistema de cobro

Luego de realizar la configuración previa de la máquina y el registro de los precios e inventario de los productos, el programa queda listo para que el cliente seleccione el producto y de paso al funcionamiento del sistema de cobro; el cual consiste en ingresar el dinero ya sea en monedas o billetes para realizar el cobro del costo del producto, sin embargo, si el cliente ingresa un monto mayor al precio, el sistema entrega el cambio correspondiente.

Por otro lado, el sistema de cobro permite recargar los compartimentos de monedas desde el compartimento frontal, con el fin de evitar hacerlo directamente en el validador ubicado en el interior de la máquina. Por ende, el sistema de cobro también cuenta el dinero en el caso de que la máquina requiera rellenar los compartimentos de monedas para poder entregar el vuelto.

De igual manera, el sistema de cobro se desarrolló en Python y su diagrama de flujo se adjunta en el Anexo 5.

### 3. Sistema de expendio del producto

Por último, una vez realizado el cobro del producto seleccionado por el cliente, se procede al despacho del mismo. Se lo hace a través del microcontrolador Arduino, el cual recibe el direccionamiento del sistema de selección desde la Raspberry, para que éste realice la activación de la espiral con su motorreductor respectivo, hasta que cumpla un giro de 360° y permita la caída del producto en la bandeja de entrega para el cliente. El código de programación se realizó mediante Arduino IDE y se adjunta en el Anexo 6.

### 3.3 Diseño de las placas electrónicas

#### Simulación del circuito para sensores encoder

El diseño del circuito consta del uso de compuertas AND (ver Figura 3.25), para el acondicionamiento de las señales digitales, es decir un cero o uno lógico proporcionado por los sensores encoder, con el fin de contar los giros de los motorreductores y poder activarlos durante una vuelta completa, por cada compra de manera independiente.

Por lo tanto, la simulación se realizó en el software Proteus, ya que permite comparar los dos estados posibles de las señales enviadas por los sensores, en donde si todos los sensores se encuentran en el mismo estado, se activa una salida.

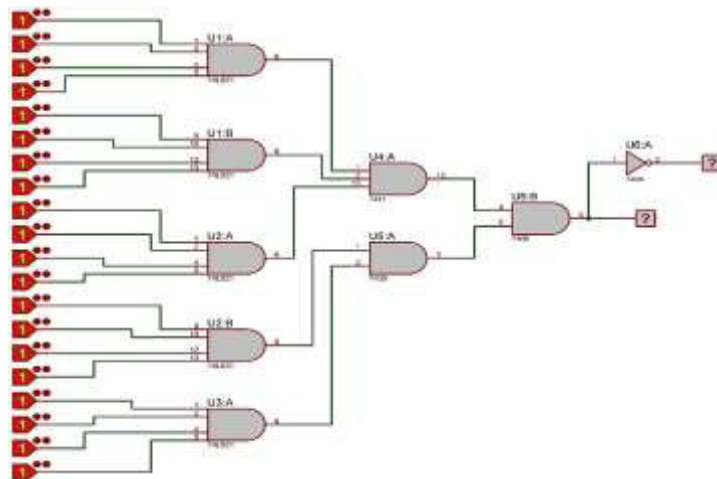
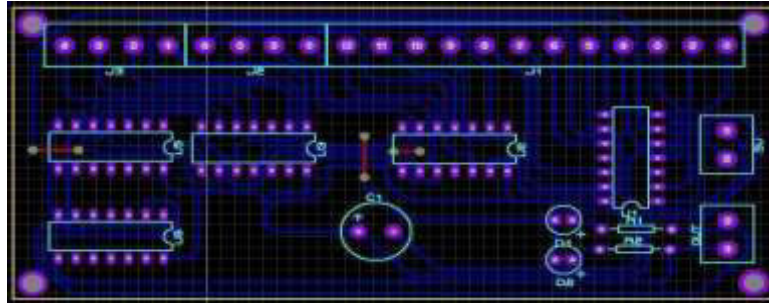


Figura 3.25 Simulación en Proteus de circuito para sensores encoder.

### Creación de placa electrónica PCB para sensores encoder

El diseño se realizó en el software Ares, ya que posee la opción de exportar la simulación para la creación de la placa electrónica de los sensores. Sus dimensiones son de 11x5 (cm) y se puede apreciar en la Figura 3.26.



**Figura 3.26** Distribución de elementos en el programa Ares.

De igual manera se puede visualizar el diseño de la placa terminada en formato 3D como se muestra en la Figura 3.27, el cual es útil para guiarse al momento de implementarla.



**Figura 3.27** Visualización 3D de la placa de los sensores encoder

### Simulación del circuito para activación de motores

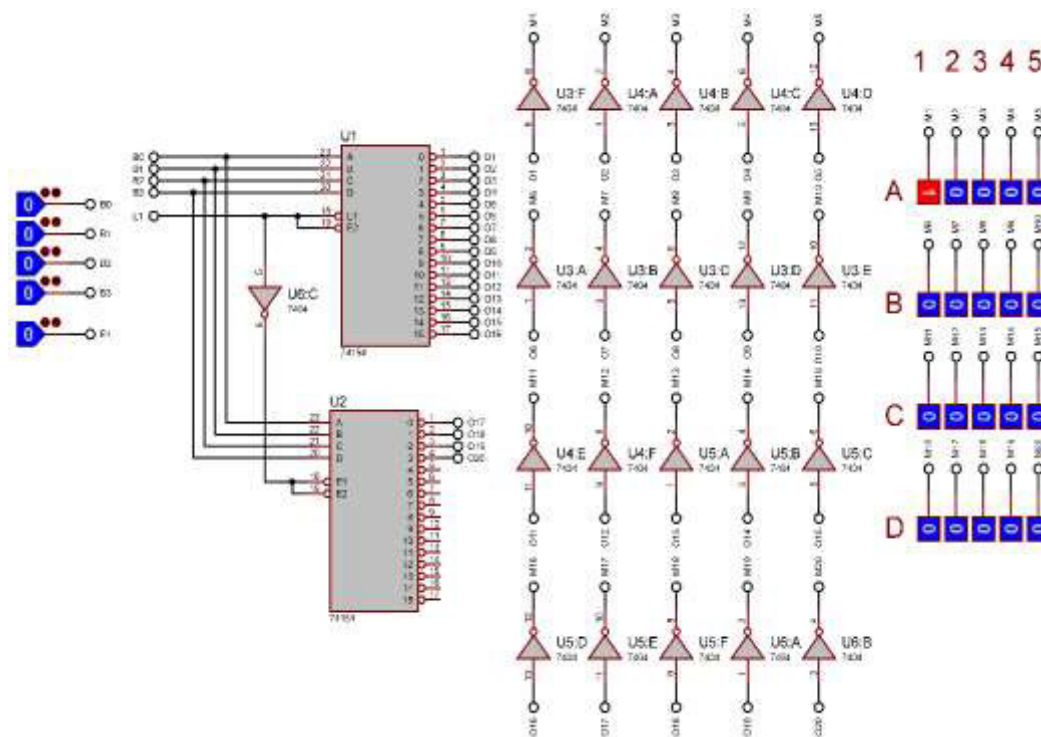
El diseño de la placa para el control de activación de los motorreductores se lo realizó a través de la demultiplexación, el cual consistió en utilizar 5 entradas distribuidas en 4 bits y 1 enable como se muestra en la Tabla 3.13. Permite la activación de 20 motorreductores de manera independiente.

**Tabla 3.13** Tabla de verdad para activación de los motorreductores.

Enable	B3	B2	B1	B0	Motorreductor
0	0	0	0	0	<b>M1</b>
0	0	0	0	1	<b>M2</b>
0	0	0	1	0	<b>M3</b>
0	0	0	1	1	<b>M4</b>
0	0	1	0	0	<b>M5</b>
0	0	1	0	1	<b>M6</b>

0	0	1	1	0	<b>M7</b>
0	0	1	1	1	<b>M8</b>
0	1	0	0	0	<b>M9</b>
0	1	0	0	1	<b>M10</b>
0	1	0	1	0	<b>M11</b>
0	1	0	1	1	<b>M12</b>
0	1	1	0	0	<b>M13</b>
0	1	1	0	1	<b>M14</b>
0	1	1	1	0	<b>M15</b>
0	1	1	1	1	<b>M16</b>
1	0	0	0	0	<b>M17</b>
1	0	0	0	1	<b>M18</b>
1	0	0	1	0	<b>M19</b>
1	0	0	1	1	<b>M20</b>
1	1	1	1	1	<b>Motorreductor en reposo</b>

De igual manera la simulación se realizó en Proteus (ver Figura 3.28), ya que direcciona los 5 bits según la Tabla 1 y así visualizar la activación del motorreductor seleccionado.

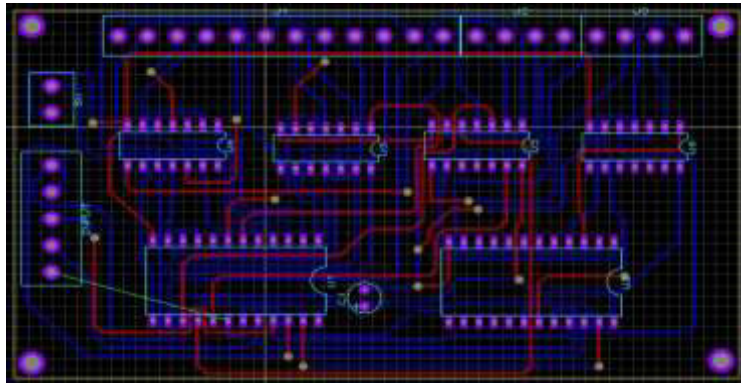


**Figura 3.28** Simulación en Proteus de circuito para activación de motorreductores.

Para el funcionamiento de los motorreductores se emplearon 2 módulos relés, 1 de 16 canales y otro de 4 canales, los cuales se encargan de realizar la activación y desactivación de los motorreductores.

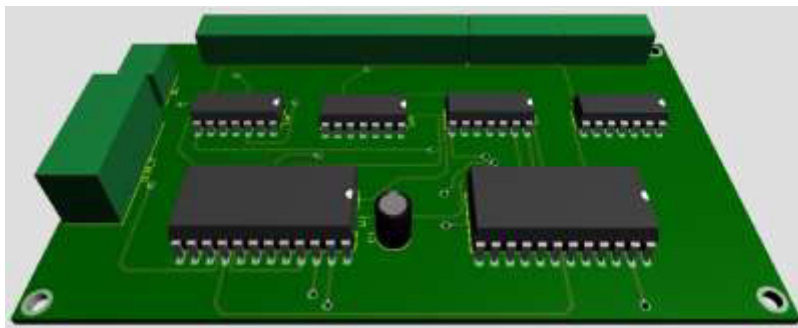
### Creación de placa electrónicas PCB para motorreductores

De la misma manera la placa electrónica se realizó en la herramienta Ares, como se muestra en la Figura 3.29, la cual posee las dimensiones de 12,3x7 (cm).



**Figura 3.29** Distribución de elementos en el programa Ares.

La visualización en 3D, se muestra en la Figura 3.30 y permite comparar la ubicación correcta de los componentes para su implementación.



**Figura 3.30** Visualización 3D de la placa de motorreductores

## 3.4 Ensamblaje de los componentes

Una vez adquirido todos los elementos se procede al ensamblaje según las áreas de control, los cuales se detallan a continuación:

### Sistema de expendio

Consistió en la ubicación, perforaciones y anclajes de los motorreductores según los canales de la bandeja para posteriormente realizar el cableado para la alimentación eléctrica y el control respectivo, (ver Figura 3.31).





**Figura 3.31** Ensamblaje de motorreductores.

La ubicación, anclaje y cableado de placas relés, microcontrolador Arduino y fuentes de poder se muestra en la Figura 3.32, para lo cual se utilizó conectores molex con sus respectivas identificaciones, para facilitar el montaje y desmontaje de los componentes; y además evitar cortocircuitos en sus conexiones.



**Figura 3.32** Ensamblaje de placas electrónicas y fuentes de poder.

### **Sistema de selección y registro**

La pantalla táctil es el componente para interactuar con el usuario, por ende, se realizó la instalación en la parte frontal como se aprecia en la Figura 3.33.



**Figura 3.33** Ensamblaje de la pantalla táctil.

### Sistema de cobro

La ubicación, anclaje y cableado del validador de monedas y lector de billetes se muestra en la Figura 3.34; los cuales requieren que su ubicación se realice de manera vertical y nivelada para la recepción del dinero y devolución del cambio.



**Figura 3.34** Ensamblaje del sistema de cobro.

## 3.5 Pruebas y Análisis de Resultados

### Establecimiento de comunicación con protocolo MDB

La prueba de la comunicación entre la Raspberry Pi4 y las lectoras de dinero a través de la interfaz MBD se realizó con el propósito de verificar el funcionamiento y reconocimiento del dinero para la cantidad ingresada al procesar una venta; sin embargo, en el caso de ingresar una cantidad que sobrepase el costo, se requirió verificar si el vuelto se entregó por completo.

El proceso de comprobación del funcionamiento para el monedero consistió en ingresar el dinero de distintas denominaciones, para verificar las validaciones permitidas las cuales se detallan en la Tabla 3.14; de igual manera se realizó el mismo proceso para el billeteo como se muestra en la Tabla 3.15.

**Tabla 3.14** Denominaciones para validación de monedas.

Monedas		Validación
\$1	Americanas	✓
	Nacionales	✓
0.50 ctvs	Americanas	X
	Nacionales	X
0.25 ctvs	Americanas	✓
	Nacionales	X
0.10 ctvs	Americanas	✓
	Nacionales	X

0.05 ctvs	Americanas	✓
	Nacionales	<b>X</b>

Debido a la configuración predeterminada de fábrica algunas validaciones no son admitidas. Para poder admitir otros tipos de moneda es necesario truncar el validador a través de la configuración de hardware. Las cuales deben tener el mismo tamaño de las monedas admitidas, caso contrario no es posible validar monedas de mayor tamaño.

**Tabla 3.15** Denominaciones para validación de billetes.

Billetes		Validación
\$1	Americanos	✓
\$5	Americanos	✓

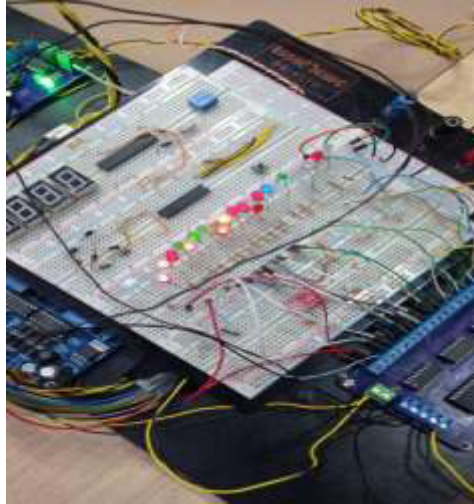
En el caso de entrega de vuelto solo se la puede realizar a través del monedero, por ende, este debe mantenerse con monedas en sus contenedores internos, el cual, se requiere que se lo haga manualmente extrayendo el monedero para poder recargar las monedas, representando un riesgo para las conexiones internas, por lo que se desarrolló a través de la programación un submenú, ver Figura 3.24, que permite cargar los contenedores, ingresando las monedas desde el compartimento frontal.

### **Comprobación de placas electrónicas PCB**

La prueba se realizó con la finalidad de comprobar si las placas electrónicas PCB diseñadas cumplen su funcionamiento al igual que en la simulación de Proteus (ver Figura 3.35). En consecuencia, se probaron de la siguiente manera:

- La placa de los motorreductores se verificó mediante las combinaciones mostradas en la Tabla 3.12 anteriormente.
- Para la placa de los sensores, se calibró la posición de los acoples de los motorreductores en las ranuras de los sensores encoder, para así detectar el número de vueltas.

Durante las pruebas de funcionamiento se obtuvieron los siguientes resultados mostrados en la Tabla 3.16.



**Figura 3.35** Comprobación de funcionamiento de placas electrónicas.

**Tabla 3.16** Registro de pruebas de placas electrónicas.

Prueba	No funciona	Funciona	Comentario
1	X		Impresión de pistas defectuosas.
2	X		Falla de circuitos integrados.
3	X		Variaciones de voltaje.
4	X		Activación múltiple de salidas.
5	X		Alimentación de los relés.
6	X		Referencia común de los circuitos.
7		✓	Activación ordenada de las salidas.
8	X		Variaciones de voltaje.

Por las dificultades presentadas se planteó la alternativa del uso del Arduino Mega 2560 ya que cuenta con la disponibilidad suficiente de puertos, los cuales permiten la activación directa de los motorreductores evitando el acondicionamiento del voltaje. Además, el Arduino descartó el uso de la placa de sensores, ya que el control de giro se realizó mediante temporizaciones independientes a través del código de programación, optimizando las conexiones en el cableado de los componentes del sistema de expendio.

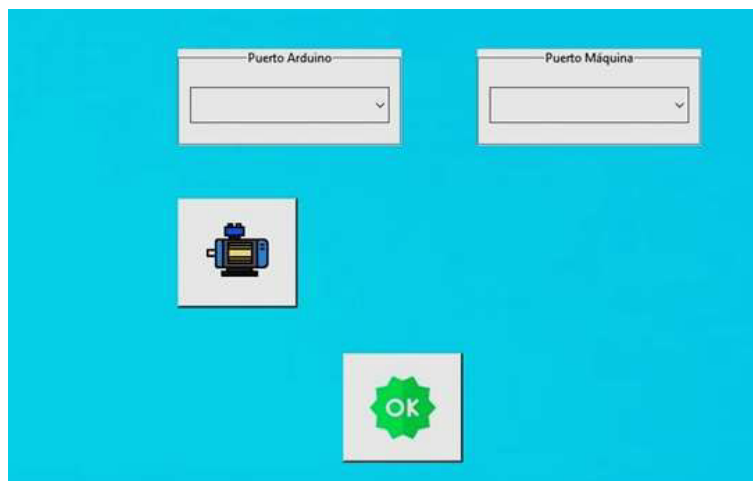
### **Comprobación de establecimiento de comunicación serial**

Dicha prueba se realizó con el fin de comprobar el envío y recepción de información entre la Raspberry, Arduino y la interfaz MDB, para ello se realizó las siguientes pruebas mostradas en la Tabla 3.17 a continuación:

**Tabla 3.17** Registro de pruebas de comunicación serial.

Prueba	No funciona	Funciona	Comentario
1	X		Reconocimiento incorrecto de puertos.
2	X		Conflicto entre puertos.
3	X		No recepción de información Arduino.
4	X		Conexión de puertos USB Raspberry
5		✓	Creación de submenú para selección de puertos.
6		✓	
7		✓	
8		✓	
9		✓	
10		✓	

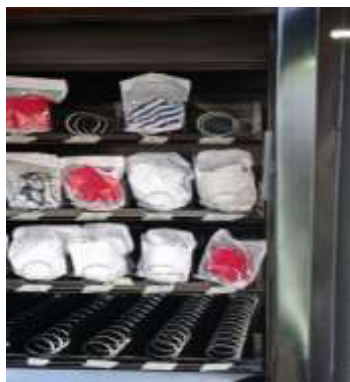
Por lo tanto, para obtener el correcto funcionamiento de la comunicación serial; se creó un submenú en la pantalla táctil que permite asignar los puertos individualmente como se muestra en la Figura 3.36. El submenú cuenta con un botón adicional que permite la activación del motor 1 con el que se podrá confirmar la selección correcta del puerto, que en este caso pertenece al puerto serial del Arduino, caso contrario, si al ser activado dicho botón y el motor 1 no gira, da por entendido que el puerto de conexión pertenece a la interfaz MDB.



**Figura 3.36** Submenú de selección de puertos.

### Calibración de tiempos de expendio del producto

En la siguiente prueba se realizó la configuración del tiempo de activación de cada uno de los motorreductores para el expendio del producto. Esta configuración consistió en colocar el producto en cada uno de los canales como se muestra en la Figura 3.37.



**Figura 3.37** Bandeja con producto.

Se verificó que el motor cumpla con un giro de 360° y se detenga; para ello se tuvo que calibrar individualmente el tiempo de funcionamiento de cada motorreductor, en la Tabla 3.18 se puede evidenciar los tiempos de giro de cada uno de ellos.

**Tabla 3.18** Tabla de calibración de los motores.

Número de bandeja	Número de motorreductor	Tiempo de activación (seg)
<b>Bandeja 1</b>	Motorreductor 1	2.06
	Motorreductor 2	2.02
	Motorreductor 3	1.94
	Motorreductor 4	1.82
	Motorreductor 5	1.93
<b>Bandeja 2</b>	Motorreductor 6	1.94
	Motorreductor 7	1.99
	Motorreductor 8	2.01
	Motorreductor 9	2.02
	Motorreductor 10	2.01
<b>Bandeja 3</b>	Motorreductor 11	2.02
	Motorreductor 12	2.03
	Motorreductor 13	1.97
	Motorreductor 14	1.97
	Motorreductor 15	1.98
<b>Bandeja 4</b>	Motorreductor 16	1.95
	Motorreductor 17	1.97
	Motorreductor 18	2.3
	Motorreductor 19	2.04
	Motorreductor 20	2.08

Es indispensable que en los mantenimientos preventivos se verifique la calibración de los tiempos de giro y de ser necesario corregir la temporización. Y así garantizar el correcto desempeño del sistema de expendio.

### Comprobación de configuración de precios

Para la realización de esta prueba se creó un submenú de interacción con el usuario dentro de la interfaz gráfica, ya que permite realizar la configuración de precios, cantidad de producto y selección del canal a ser cargado, directamente desde la pantalla táctil y no mediante la programación.

La configuración de los precios se ingresa en cifras enteras o decimales como se puede evidenciar en la Figura 3.38. Además, el algoritmo está diseñado para impedir la venta, si el producto no se encuentra en stock.



**Figura 3.38** Submenú de configuración de precios.

Para la comprobación respectiva se configuró todas las bandejas y se evidenció el funcionamiento mediante la venta de los productos, el registro de estos se muestra en la Tabla 3.19

**Tabla 3.19** Registro de pruebas de configuración de precio.

Prueba	Funciona
1	✓
2	✓
3	✓
4	✓
5	✓

La prueba de configuración de precios no presentó inconvenientes en su funcionamiento.

### Comprobación de control de ventas

Para el registro de facturación se diseñó un submenú en el que se visualiza mensualmente las ventas generadas, la prueba consistió en realizar varias ventas en cada uno de los canales, para así evidenciar si se registran de manera correcta los valores de los productos vendidos y el registro de la venta mensual, dicho submenú se visualiza en la Figura 3.39.

Fecha	Precio	Producto
2021-04-18	9	3
2021-04-18	0	3
2021-04-18	0.05	9
2021-04-18	0.05	9
2021-04-18	1.25	4
2021-04-18	1.25	4
2021-04-18	1.25	4
2021-04-18	1.25	4
2021-04-18	1.75	2
2021-04-18	1.25	2
2021-04-18	5	4
2021-04-18	1.25	2
2021-04-18	5	4
2021-04-18	5	4
2021-04-18	1	2
2021-04-18	1	2

Facturación mensual  
**\$ 47.66**

Regresar

**Figura 3.39** Submenú de registro de ventas.

Para la verificación del registro de ventas, se llevó un control del valor total de las ventas generadas y de la cantidad del producto. El registro de pruebas del control de ventas se muestra en la tabla 3.20.

**Tabla 3.20** Registro de pruebas de control de ventas.

Prueba	No funciona	Funciona	Comentario
1	<b>X</b>		Reinicio del registro mensual.
2		✓	
3		✓	
4		✓	
5		✓	

En la prueba se presentó el problema con el reinicio del registro mensual, ya que al pasar al registro del siguiente mes, la facturación mensual no se encero. Por lo que se generó cambios en el programa para que el registro solo se realice mensualmente.



### Comprobación de recarga de monedas

Para esta opción se creó un submenú que facilite la recarga de monedas (ver Figura 3.40), el validador clasifica de manera automática las monedas de \$0,05, \$0,10 \$0,25 ctvs. Ofreciendo la opción de abastecer de monedas al validador desde el compartimento de pago sin necesidad de desmontarlo y así recargar para que pueda generar el vuelto en el caso de que sea necesario.



**Figura 3.40** Submenú de recarga de monedas

Su comprobación respectiva se registró en la Tabla 3.20 a continuación:

**Tabla 3.21** Registro de pruebas de recarga de monedas.

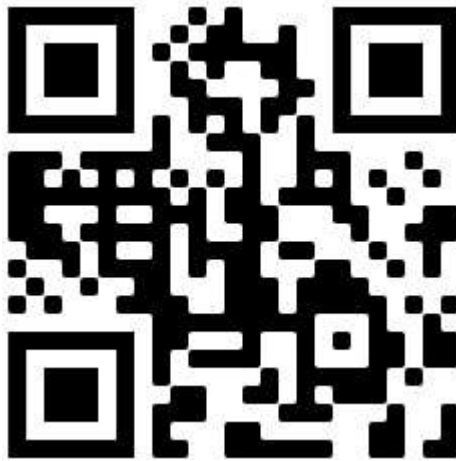
Prueba	Funciona
1	✓
2	✓
3	✓
4	✓
5	✓

La prueba de recarga de monedas no presentó inconvenientes en su funcionamiento.

### 3.6 Manual de usuario y mantenimiento

Para revisar con más detalle el funcionamiento de la máquina dispensadora y el mantenimiento de esta, se recomienda revisar el video explicativo correspondiente. Se

puede acceder al video por medio del código QR que se muestra a continuación en la Figura 3.4.



**Figura 3.41** Código QR – Manual de usuario y mantenimiento.

## **4 CONCLUSIONES Y RECOMENDACIONES**

### **4.1 Conclusiones**

- La interfaz MDB es esencial para el funcionamiento y la comunicación de las lectoras de dinero, ya que el controlador mediante el protocolo de comunicación de 9 bits permitió la configuración de acuerdo con las necesidades del proyecto.
- La ventaja de la implementación de la arquitectura maestro-esclavo es que permite la comunicación entre distintos controladores, independientemente de su lenguaje de programación, el cual permite al controlador maestro aprovechar de los recursos del esclavo.
- El lenguaje de programación Python al poseer una licencia de código abierto dispone de una variedad de librerías gratuitas o proyectos liberados que sirven de base y simplifica el desarrollo del código de programación, de igual manera al poseer una sintaxis simple y una curva de aprendizaje corta, Python brinda el desarrollo de proyectos robustos a corto plazo, como fue el caso de la automatización de la máquina expendedora.

- La librería Tkinter permitió el desarrollo de interfaces gráficas (submenús), mediante un diseño amigable, los cuales facilitan la interacción con el usuario o el cliente en la manipulación de la máquina.
- El uso de la librería sqlite3 permite el manejo de datos a menor escala en el desarrollo de proyectos que manejan un control de datos limitado como el caso del control de la facturación mensual de la máquina expendedora.
- En el diseño de una máquina expendedora es necesario tener en cuenta la versatilidad para ofertar distintos tipos de productos según la demanda del mercado donde esta se encuentre ubicada, por ende, las bandejas fueron diseñadas para contar con el espacio suficiente, en donde para el caso de comercialización de productos refrigerados, la estructura cuenta con un área para añadir un sistema de refrigeración para futuras mejoras.
- Durante la comprobación se presentaron inconvenientes con la placa PCB de los motorreductores, los cuales se relacionaron con los voltajes de salida generando un desorden en la activación, es decir al ingresar la selección, la salida no fue activada de forma correcta. Debido a problemas con la alimentación de los relés, donde además las señales proporcionadas por la Raspberry fueron inferiores al voltaje requerido por la placa.

## 4.2 Recomendaciones

- Para el manejo de varias salidas que propasen la capacidad del controlador, es necesario utilizar un demultiplexor, de igual manera en el caso de que las señales de entrada no dispongan de puertos disponibles es necesario plantear un control a través de compuertas lógicas, siempre y cuando la señal otorgada por el sensor sea digital.
- Para la elaboración de placas electrónicas que poseen varias conexiones o pistas en ambos lados es recomendable imprimirlas a través de una máquina CNC y añadirle una película de antisolder para un mayor cuidado y acabado, ya que al realizarlo de manera artesanal presentan problemas en la continuidad de las pistas o con el tiempo están propensas a la corrosión de las mismas.

- Se recomienda verificar previamente la disponibilidad de los componentes electrónicos en el país, ya que en el caso de requerir una importación se debe tener en cuenta el tiempo y el presupuesto que implica obtener estos elementos, ya que estos generan un retraso en el avance del proyecto.
- Tener en cuenta que los componentes puedan ser desmontables con facilidad, para ello se recomienda el uso de conectores Molex, que permiten la conexión en un solo sentido para evitar cortocircuitos, permita el cambio de algún componente si es necesario o facilite el desmontaje para mayor facilidad en la movilización de la máquina.
- La comunicación entre la Raspberry, el Arduino y la interfaz MDB a través del puerto serial, requiere verificar mediante la programación que estos mantengan una conexión efectiva, para así evitar conflictos de comunicación y por ende problemas en el desarrollo del programa.
- La correcta selección de los elementos de protección como son el interruptor termomagnético y el fusible dependerá de las cargas a proteger, por lo que se recomienda medir las magnitudes eléctricas y comparar con las especificaciones otorgadas por el fabricante, para que el cálculo de estas se realice de manera correcta y así evitar sobredimensionamientos o sub-dimensionamientos en la selección de los mismos.

## 5 REFERENCIAS BIBLIOGRÁFICAS

- Ansi. (13 de 06 de 1992). *webstore.ansi.org*. Obtenido de [https://webstore.ansi.org/preview-pages/ISA/preview\\_S\\_501.pdf](https://webstore.ansi.org/preview-pages/ISA/preview_S_501.pdf)
- Capella, R. (12 de 05 de 2003). *Schneider electric* . Obtenido de [https://frrq.cvg.utn.edu.ar/pluginfile.php/2738/mod\\_resource/content/0/PT071-Protecciones\\_en\\_MT.pdf](https://frrq.cvg.utn.edu.ar/pluginfile.php/2738/mod_resource/content/0/PT071-Protecciones_en_MT.pdf)
- Covantec. (13 de 04 de 2019). Obtenido de <https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion1/holamundo.html>
- Dongfeng Liangdian. (06 de 04 de 2019). Obtenido de <https://app.alibaba.com/dynamiclink?touchId=1600270715332&type=product&s>

- chema=enalibaba%3A%2F%2Fdetail%3Fid%3D1600270715332%26ck%3Dshare\_detail&ck=share\_detail&shareScene=buyer
- Fairchild. (08 de 07 de 2012). Obtenido de <https://pdf1.alldatasheet.com/datasheet-pdf/view/51038/FAIRCHILD/74LS154.html>
  - Junye. (19 de 04 de 2006). Obtenido de <https://datasheetspdf.com/pdf-file/685501/Junye/MOCH22A/1>
  - Maya Vending. (08 de 08 de 2014). Obtenido de <https://mayavending.com/descargas/#1526831692938-05582acf-45e0>
  - MEI. (19 de 04 de 2006). *meigroup.com*. Obtenido de [http://cpitecnico.x10host.com/datasheet/ngc\\_handbook\\_issued\\_version\\_g5\\_es.pdf](http://cpitecnico.x10host.com/datasheet/ngc_handbook_issued_version_g5_es.pdf)
  - MEI. (10 de 11 de 2008). *meigroup.com*. Obtenido de <http://www.unitedchangers.com/MEI%20Series%202000%20%20Operational%20Service%20Guide.pdf>
  - MP. (28 de 02 de 2020). *salud.gob.ec*. Obtenido de [https://www.salud.gob.ec/wp-content/uploads/2020/03/lineamiento-operativo-coronavirus-FINAL\\_02-2020.pdf](https://www.salud.gob.ec/wp-content/uploads/2020/03/lineamiento-operativo-coronavirus-FINAL_02-2020.pdf)
  - Nbleison. (14 de 03 de 2021). Obtenido de <https://www.nbleisonmotor.com/LS-TG38-37MM-Dc-Gear-Motor-pd6666204.html>
  - Okystar. (15 de 06 de 2019). Obtenido de <https://www.okystar.com/product-item/16-channel-12v-coupling-optocoupler-oky3016/>
  - OMS. (11 de 03 de 2020). *Organizacion mundial de la salud*. Obtenido de <https://www.who.int/es/news-room/detail/27-04-2020-who-timeline---covid-19>
  - Pastor, J. (25 de 07 de 2019). *xataka.com*. Obtenido de <https://www.xataka.com/ordenadores/raspberry-pi-4-analisis-caracteristicas-precio-especificaciones>
  - PNUD . (24 de 02 de 2021). *undp.org*. Obtenido de <https://www.ec.undp.org/>
  - Python. (06 de 06 de 2021). *python.org*. Obtenido de <https://docs.python.org/es/3/library/sqlite3.html>
  - Python. (06 de 06 de 2021). *python.org*. Obtenido de <https://docs.python.org/es/3/library/tkinter.html>
  - Qibixx. (04 de 07 de 2017). *qibixx.com*. Obtenido de <https://www.qibixx.com/es/compania/blogs/news/que-es-un-mdb-y-como-se-integra-en-las-maquinas/>

- Raspberry. (15 de 07 de 2019). *raspberrypi.org*. Obtenido de <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/>
- Servicio Nacional de Gestión de Riesgos. (07 de 04 de 2020). Obtenido de <https://www.gestionderiesgos.gob.ec/resoluciones-coe-nacional-07-de-abril-2020/>
- Texas Instrument. (15 de 02 de 2009). Obtenido de [https://www.ti.com/lit/ds/symlink/sn74ls08.pdf?ts=1624495307926&ref\\_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FSN74LS08%253Futm\\_source%253Dsupplyframe%2526utm\\_medium%253DSEP%2526utm\\_campaign%253Dnot\\_alldatasheet%2526DCM%253Dyes%2526dclid%253DCjgKE](https://www.ti.com/lit/ds/symlink/sn74ls08.pdf?ts=1624495307926&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FSN74LS08%253Futm_source%253Dsupplyframe%2526utm_medium%253DSEP%2526utm_campaign%253Dnot_alldatasheet%2526DCM%253Dyes%2526dclid%253DCjgKE)

## **ANEXOS**

## **Anexo 1: Certificado de Funcionamiento**

# **ESCUELA POLITECNICA NACIONAL**

Campus Politécnico "J. Rubén Orellana R

Quito, 01 de Julio de 2021

## **CERTIFICADO DE FUNCIONAMIENTO DE PROYECTO DE TITULACIÓN**

Yo, *Abraham Israel Loja Romero*, docente a tiempo completo de la Escuela Politécnica Nacional y como director de este trabajo de titulación, certifico que he constatado el correcto funcionamiento de la máquina expendedora de mascarillas para El Programa de las Naciones Unidas para el Desarrollo (PNUD) la cual fue implementada por los estudiantes Byron Espinosa y Steven Parra.

El proyecto cumple con los requerimientos de diseño y parámetros necesarios para que los usuarios donde la PNUD disponga la ubicación de la máquina puedan usar las instalaciones con seguridad para los equipos y las personas.



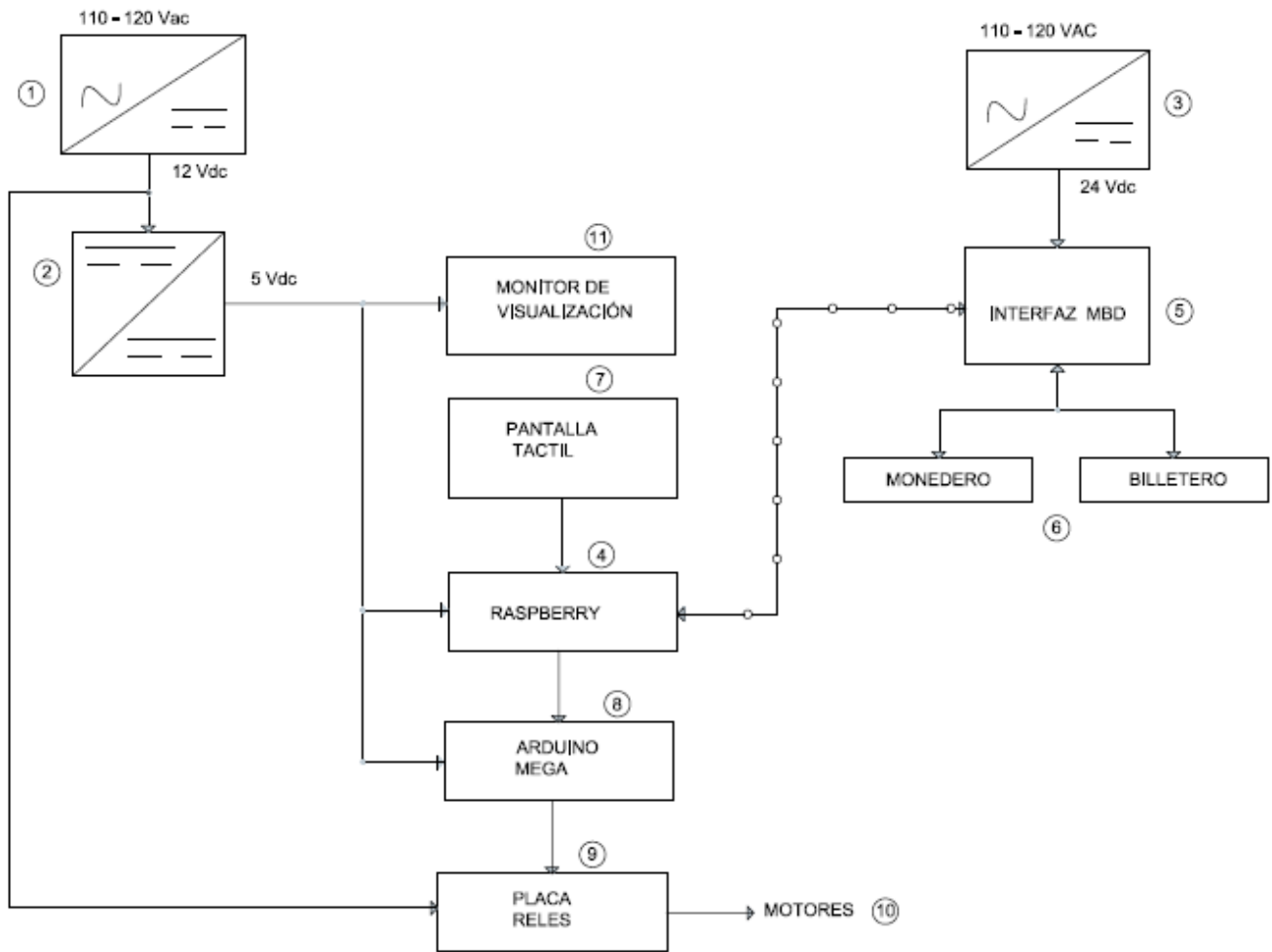
---

**DIRECTOR**




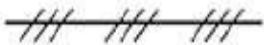





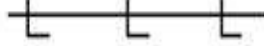
Ing. Abraham Israel Loja Romero, Msc.



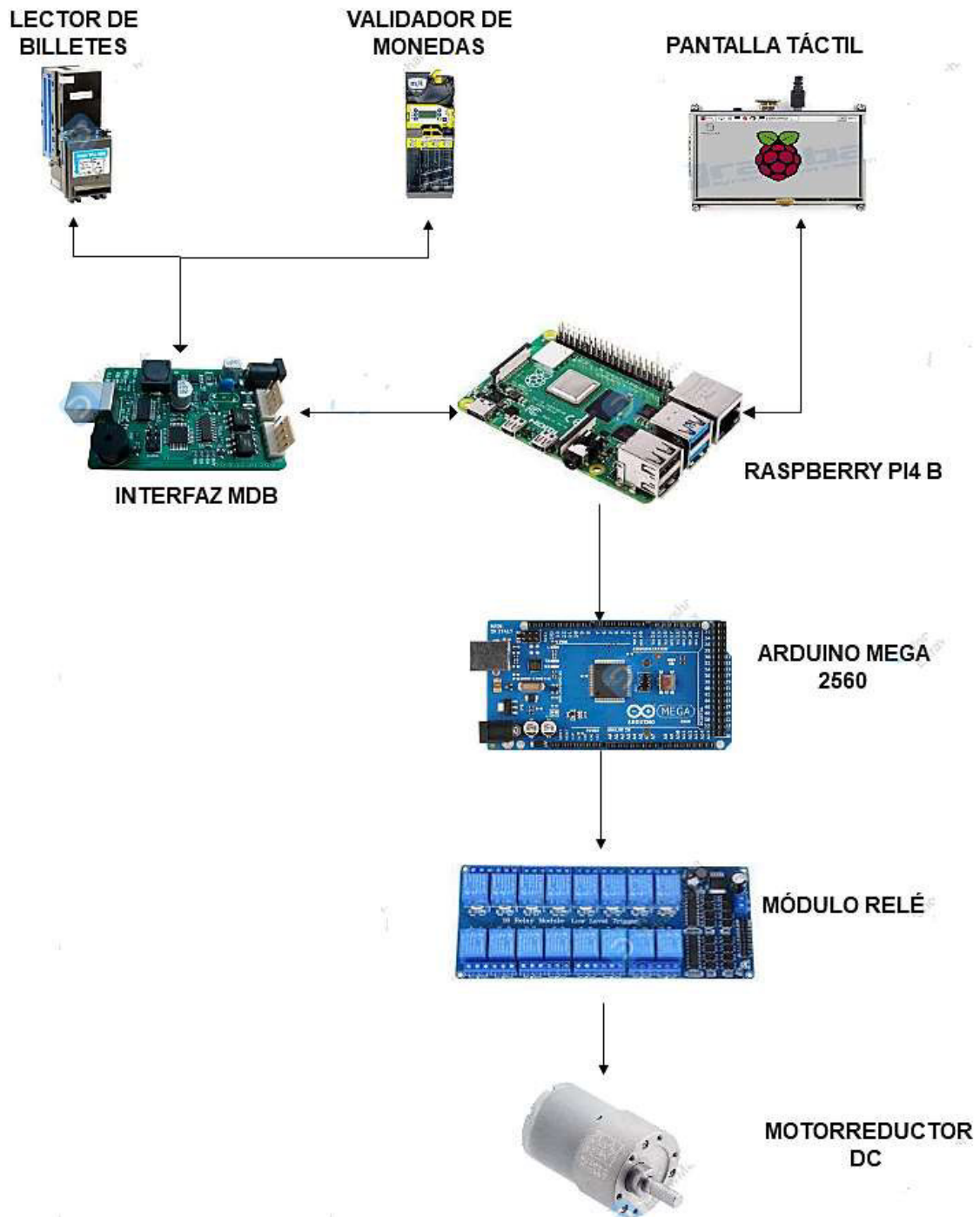
## Anexo 2: Esquema de conexiones



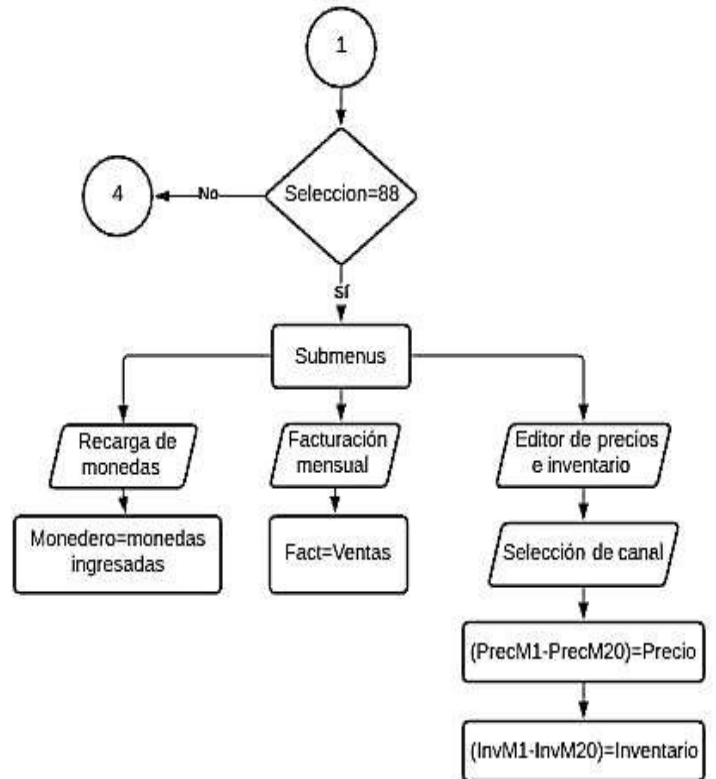
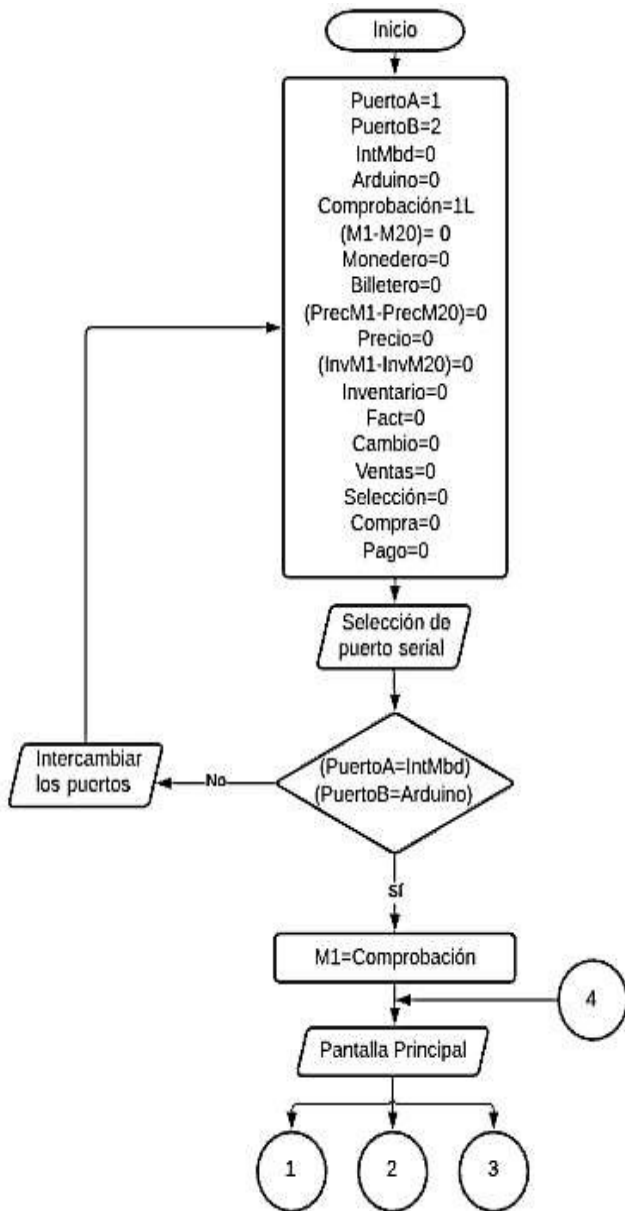
### Anexo 3: Simbología de conexiones norma ANSI-ISA 50.1-1982 (R1992)

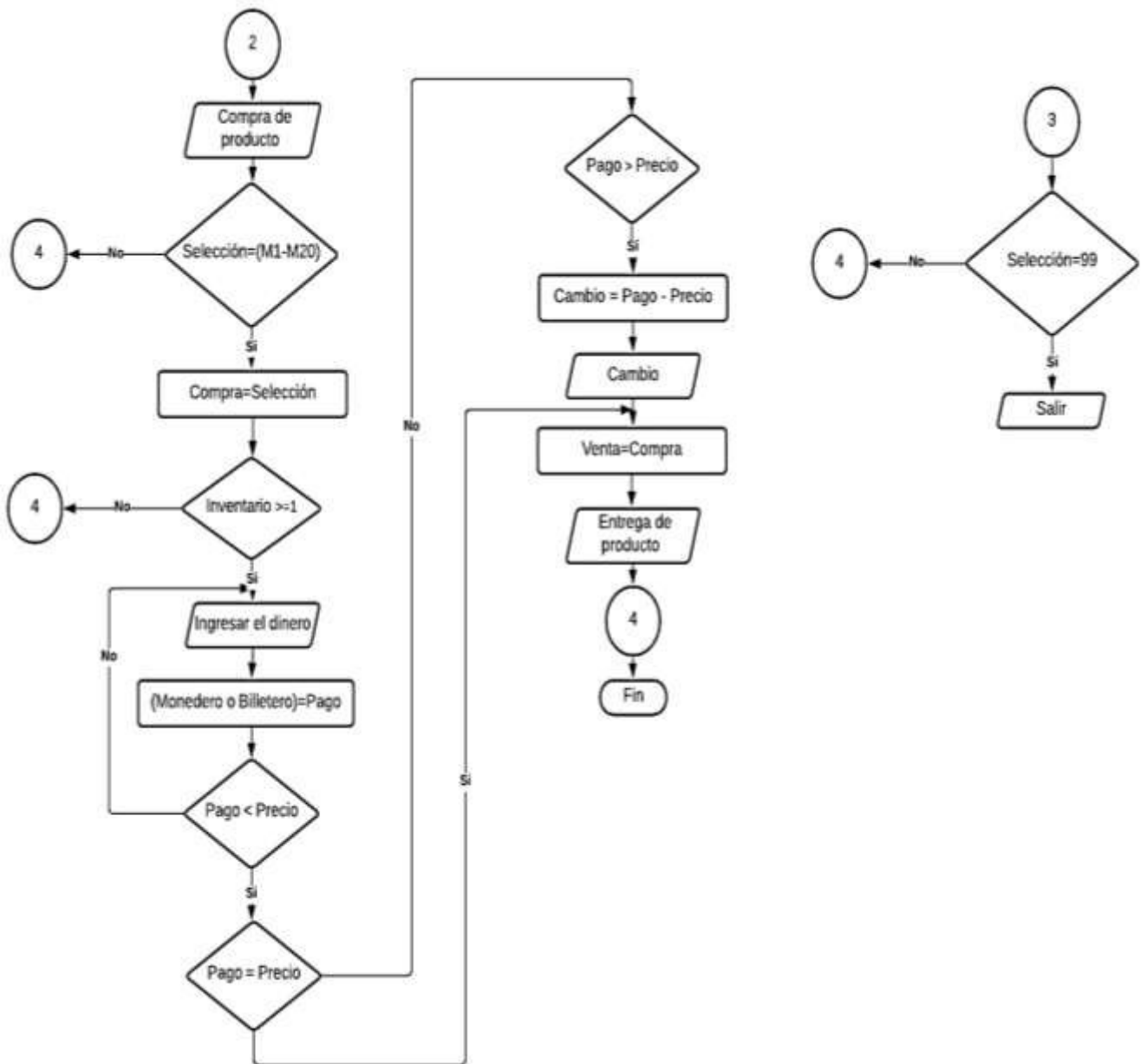
	Conexión a proceso, o enlace mecánico o alimentación de instrumentos.
	Señal neumática
	Señal eléctrica
	Señal eléctrica (alternativa)
	Tubo capilar
	Señal sonora o electromagnética guiada (incluye calor, radio, nuclear, luz)
	Señal sonora o electromagnética no guiada
	Señal de software o datos
	Conexión mecánica
	Señal hidráulica

## Anexo 4: Arquitectura maestro-esclavo de la máquina expendedora



## Anexo 5: Diagrama de flujo de programación en Python





## Anexo 6: Código de programación en IDE de Arduino

```
void setup() {
//declaración de pines a utilizarse para
cada motor
pinMode(22,OUTPUT);
pinMode(23,OUTPUT);
pinMode(24,OUTPUT);
pinMode(25,OUTPUT);
pinMode(26,OUTPUT);
pinMode(27,OUTPUT);
pinMode(28,OUTPUT);
pinMode(29,OUTPUT);
pinMode(30,OUTPUT);
pinMode(31,OUTPUT);
pinMode(32,OUTPUT);
pinMode(33,OUTPUT);
pinMode(34,OUTPUT);
pinMode(35,OUTPUT);
pinMode(36,OUTPUT);
pinMode(37,OUTPUT);
pinMode(38,OUTPUT);
pinMode(39,OUTPUT);
pinMode(40,OUTPUT);
pinMode(41,OUTPUT);

Serial.begin(9600);
for (int i=22; i<=41; i++){ //comparación
de pines
digitalWrite(i,HIGH);
}
}

void loop() {
if (Serial.available()) {
int m = Serial.parseInt();

// configuración de tiempos para cada
motor
if (m!=0){
if(m==18){
motor(orden(m),2.3); //tiempo del
motor 18
}else if(m==1){
motor(orden(m),2.06); //tiempo
del motor 1
} else if(m==2){
motor(orden(m),2.02); //tiempo
del motor 2
}else if(m==3){
motor(orden(m),1.94); //tiempo
del motor 3
}else if(m==4){
motor(orden(m),1.82); //tiempo
del motor 4
}else if(m==5){
motor(orden(m),1.93); //tiempo
del motor 5
}else if(m==6){
motor(orden(m),1.94); //tiempo
del motor 6
}else if(m==7){
motor(orden(m),1.99); //tiempo
del motor 7
}else if(m==8){
motor(orden(m),2.01); //tiempo
del motor 8
}else if(m==9){
motor(orden(m),2.02); //tiempo
del motor 9
}
```

```

    }else if(m==10){
        motor(orden(m),2.01); //tiempo
del motor 10
    }else if(m==11){
        motor(orden(m),2.02); //tiempo del
motor 11
    }else if(m==12){
        motor(orden(m),2.03); //tiempo del
motor 12
    }else if(m==13){
        motor(orden(m),1.97); //tiempo del
motor 13
    }else if(m==14){
        motor(orden(m),1.97); //tiempo del
motor 14
    }else if(m==15){
        motor(orden(m),1.98); //tiempo del
motor 15
    }else if(m==16){
        motor(orden(m),1.95); //tiempo del
motor 16
    }else if(m==17){
        motor(orden(m),1.97); //tiempo del
motor 17
    }else if(m==19){
        motor(orden(m),2.04); //tiempo del
motor 19
    }else {
        motor(orden(m),2.08); //tiempo del
motor 20
    }
}
}
}
}
void motor(int num,float t_){
    t_ = t_*1000;

```

```

Serial.print("Motor ");
Serial.print(num);
Serial.print(" ");
Serial.print(t_/1000);
Serial.println(" s");
digitalWrite(num,LOW);
delay(t_);
digitalWrite(num,HIGH);
}
int orden(int m){
    if(m==1){
        return 22;
    }else if(m==2){
        return 23;
    }else if(m==3){
        return 24;
    }else if(m==4){
        return 25;
    }else if(m==5){
        return 26;
    }else if(m==6){
        return 27;
    }else if(m==7){
        return 28;
    }else if(m==8){
        return 29;
    }else if(m==9){
        return 30;
    }else if(m==10){
        return 31;
    }else if(m==11){
        return 32;
    }else if(m==12){
        return 33;
    }else if(m==13){
        return 34;
    }
}

```

```
}else if(m==14){  
    return 35;  
}else if(m==15){  
    return 36;  
}else if(m==16){  
    return 37;  
}else if(m==17){  
    return 38;
```

```
}else if(m==18){  
    return 39;  
}else if(m==19){  
    return 40;  
}else if(m==20){  
    return 41;  
}  
}
```