



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE DEPARTAMENTO DE ENERGÍA Y MECÁNICA

CARRERA INGENIERÍA MECATRÓNICA

Trabajo de titulación, previo a la obtención del título de Ingeniero Mecatrónico

Lockers automáticos para la recepción de paquetes utilizando RPA (robotic process automation) permitirá agilizar el proceso de envío de encomiendas a WOLF

Autores:

Inca Cujilema, Galo Fabricio

Teneda Espín, Alvaro José

Ing. Mendoza Chipantasi, Darío José, *Director*



Antecedentes RPA y CSC

Dentro de RPA se puede referenciar dos tipologías claras:

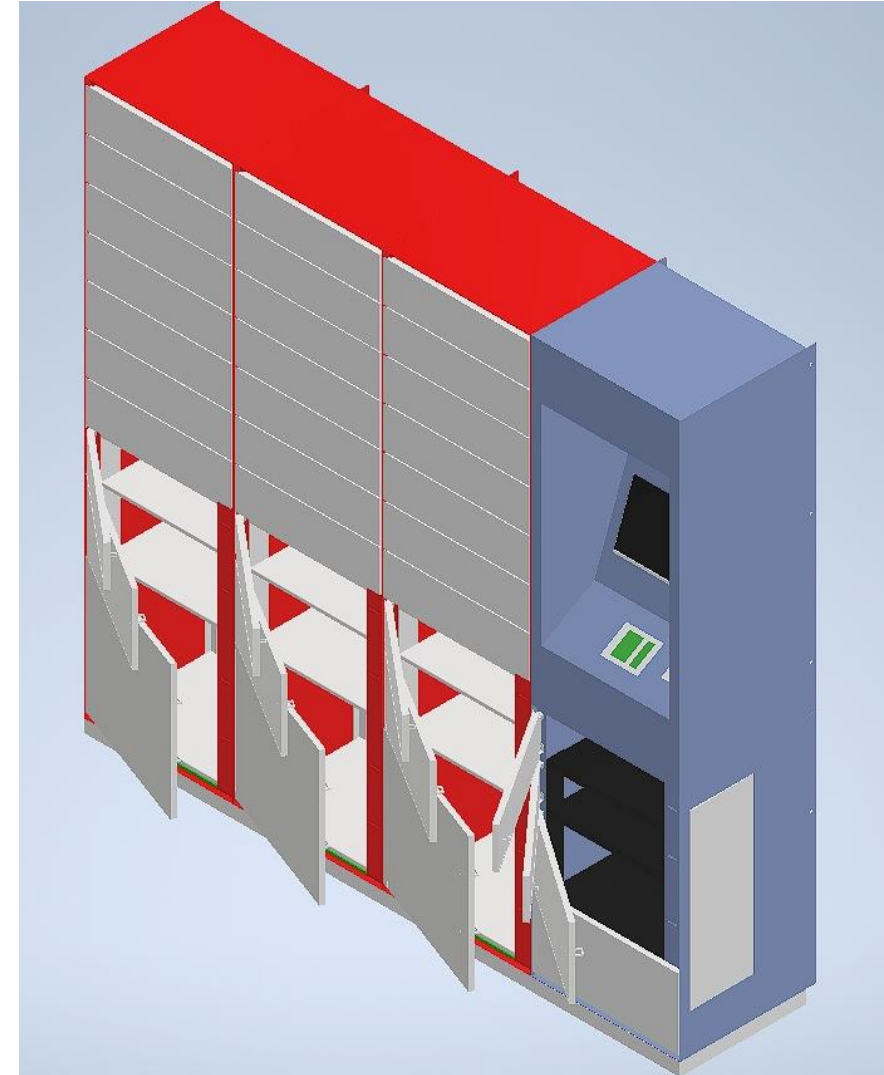
- Aquellos robots permiten gestionar procesos no definidos, mediante un aprendizaje probabilístico y tratando datos no estructurados.
- Procesos deterministas, empleando datos estructurados y procesos basados en normativa.

Estos últimos son los más empleados para generar centros de servicios compartidos ya que ejecutan procesos transaccionales que suelen ser repetitivos y basados en una serie de reglas.



Justificación

- ✓ Sistema que permita la recepción de paquetes basándose en la combinación de las diferentes tecnologías
- ✓ Uso de RPA y CSC
- ✓ Elaboración de casilleros modulares automáticos
- ✓ Facilidad en el pago para el servicio de encomiendas.



Objetivo general

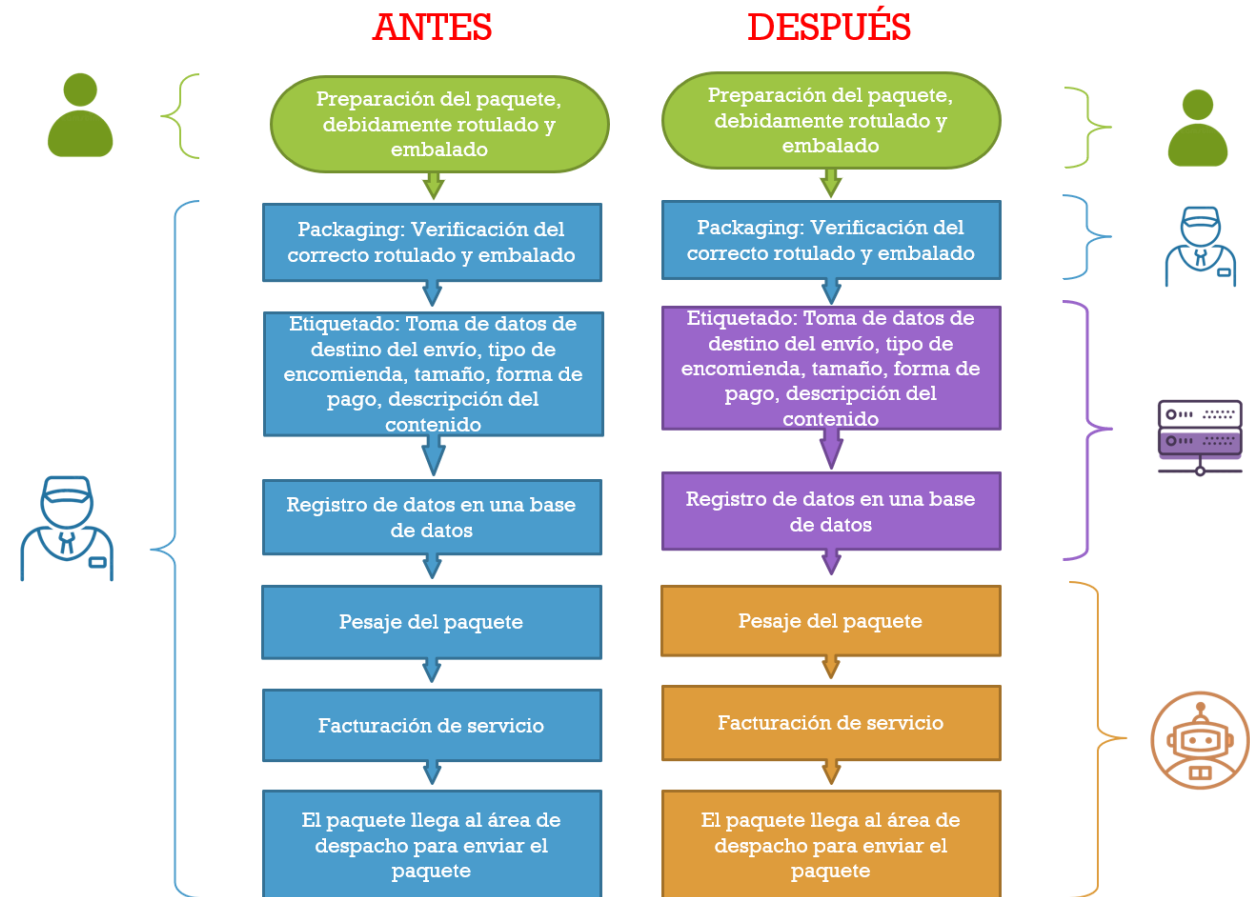
Diseñar y construir un prototipo de casilleros modulares automatizados para la recepción de paquetes.

Objetivos específicos

- Diseñar y construir los casilleros modulares basándose en las necesidades de eficiencia y las diferentes normativas para la fabricación del prototipo.
- Evaluar que el sistema mecánico planteado para que los lockers funcionen correctamente, brinden seguridad y cumplan los requerimientos de la Empresa.
- Diseñar los circuitos de control con sus respectivos diagramas circuitales para PCB que satisfagan las necesidades del usuario y den seguridad al momento de depositar los paquetes.
- Elaborar la placa PCB para los circuitos diseñados y evaluar el correcto funcionamiento de manera física para cada uno de los casilleros al momento de recibir el paquete
- Diseñar la capa de comunicación que permita la recepción de datos entre el hardware y un servidor remoto.
- Evaluar el funcionamiento del diseño modular que facilite la expansión del servicio mediante la automatización del mismo.
- Facilitar el servicio de pago para envío de encomiendas mediante la implementación de una interfaz amigable.
- Realizar pruebas de funcionamiento del sistema completo y de su modularidad.

Criterio de automatización para RPA

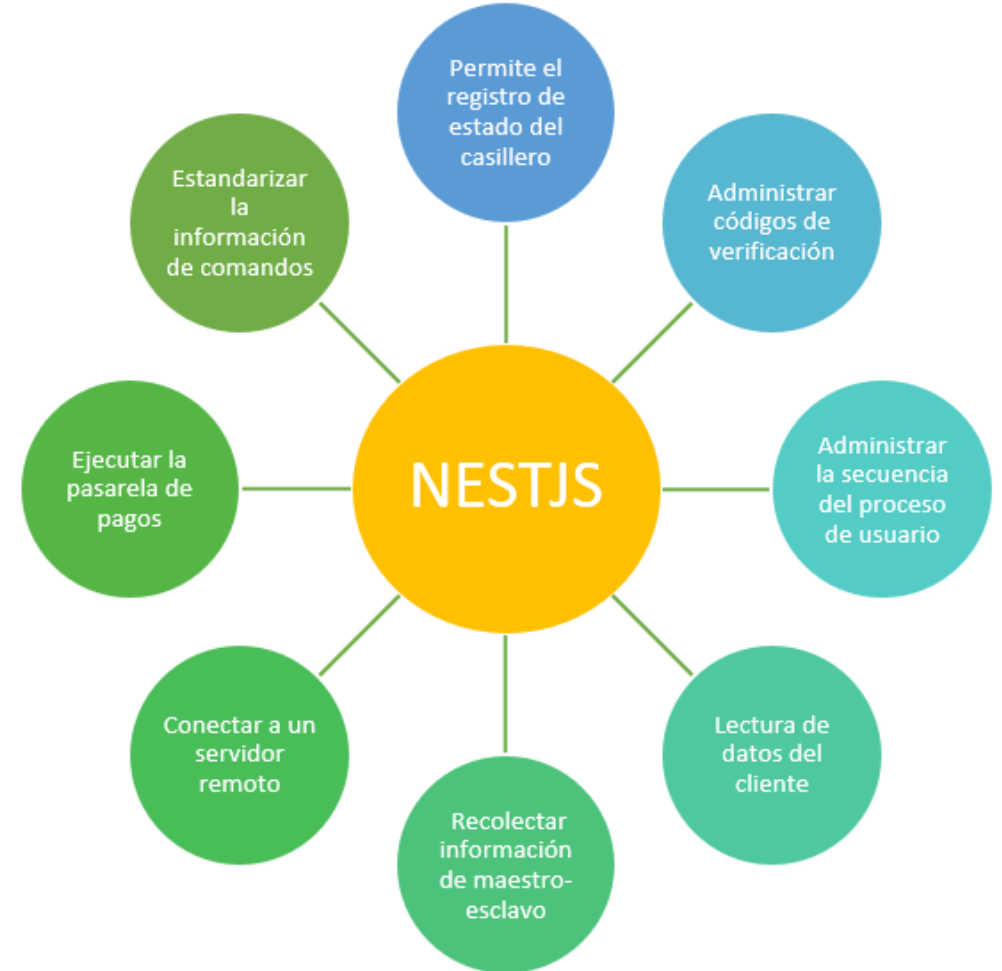
Criterio	Características	Valoración
Múltiples sistemas utilizados	El robot CSC puede acceder al subsistema de pesaje, al subsistema esclavo-maestro, a la pasarela de pagos y a la secuencia de HMI.	Si
Transacciones de alto volumen / valor	Un sistema de paquetería tradicional requiere de una o dos personas tomando en cuenta que se pueden generar errores en la facturación.	Si
Propenso a errores y retrabajo	El proceso manual tiene una frecuencia y complejidad media, sin embargo, resulta en un gran número de errores.	Si
Alta predictibilidad	El proceso está definido de forma secuencial que es visible para el cliente mediante un HMI y lleva un registro de estado de los casilleros.	Si
Excepciones limitadas	El proceso se centra en automatizar la toma de datos del paquete y asegurarlo de forma inmediata.	Si
Carga de trabajo manual significativa	El proceso tradicional es manual en la toma de datos del cliente, en su registro en una base de datos y en adquirir los datos del paquete. Para que el proceso se agilice se necesita aumentar la mano de obra.	Si



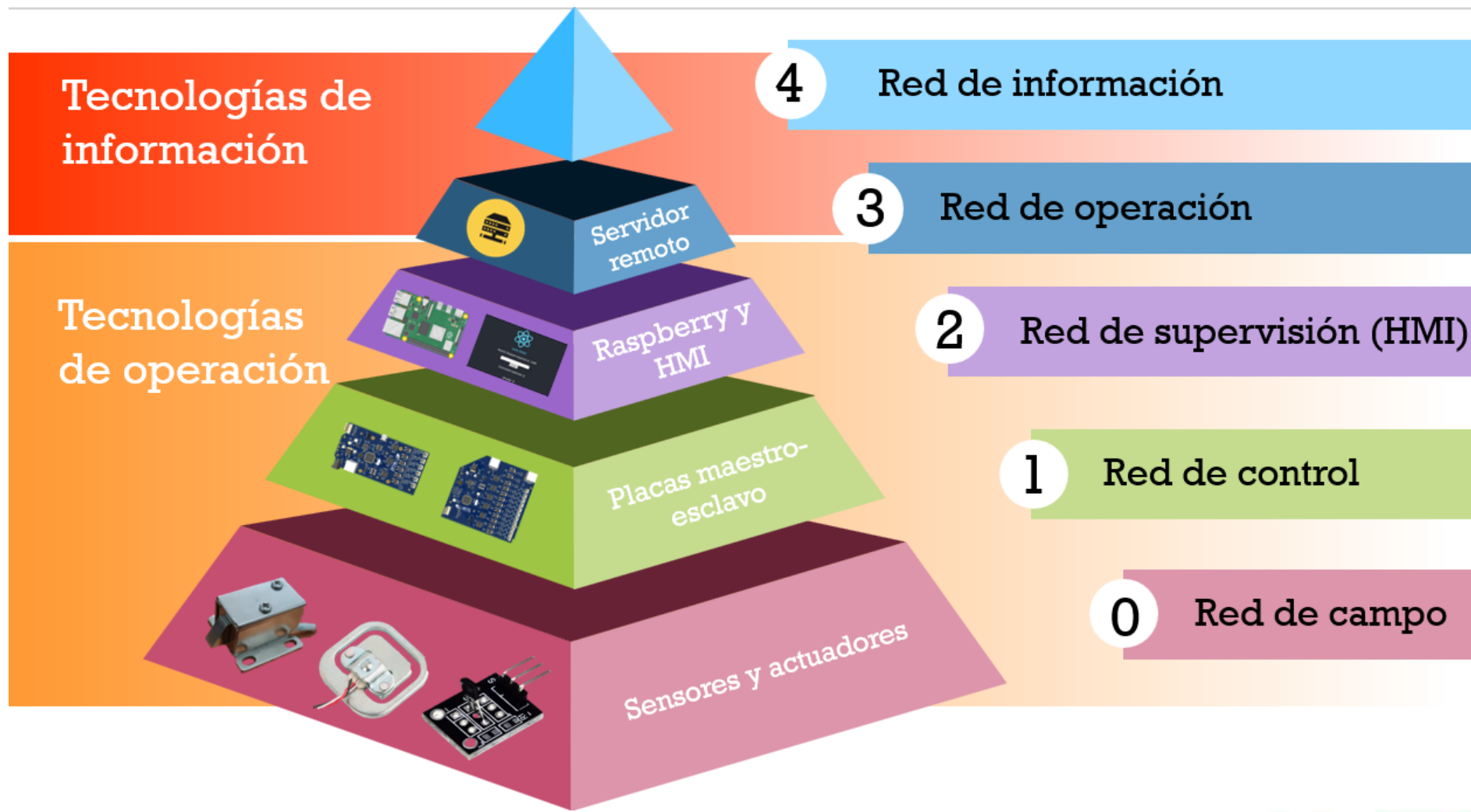
Automatización del proceso mediante RPA

Proceso	Antes	Después	Variable a controlar
Sistema de pesaje	Se requiere de una persona para la toma de datos del paquete	El sistema sensa automáticamente el paquete del usuario.	Peso
Pasarela de pagos	Una persona registra el paquete en una base de datos y realiza el cobro del envío.	El sistema muestra el valor de envío y debita directamente de la tarjeta.	Tarifa
Sistema de identificación del usuario	El personal llena los datos del usuario y del paquete en un formulario cada vez que ocupa el servicio	El servicio requiere validar los datos de envío mediante un código único.	Validación de datos
Seguridad	Se almacenan los paquetes en la zona de despacho.	El locker asegura los paquetes en su estructura de forma individual.	Sensor Hall y estado de la cerradura
Interacción	Se requiere de personal capacitado para brindar el servicio de paquetería	Se reduce el error humano al usar un sistema de envío a base de una interfaz intuitiva.	Tiempo de uso del HMI

Robot CSC



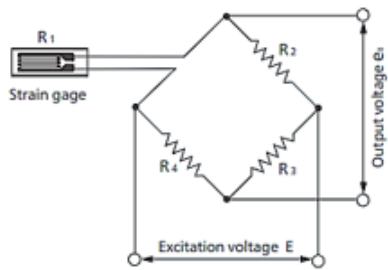
Pirámide de automatización



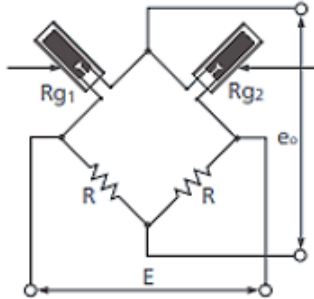
Criterios de selección para galgas

Tipo de galga extensiométrica	Peso	Precableadas	Precisión	Calidad
Rango de medición	30%	50 kg	5 kg	30 kg
Precisión	20%	Media	Alta	Alta
Costo	10%	Baja	Alta	Alta
Facilidad de uso	20%	Medio	Alto	Alto
Robustez	20%	Alta	Medio	Alta
Total	100%	80%	20%	70%

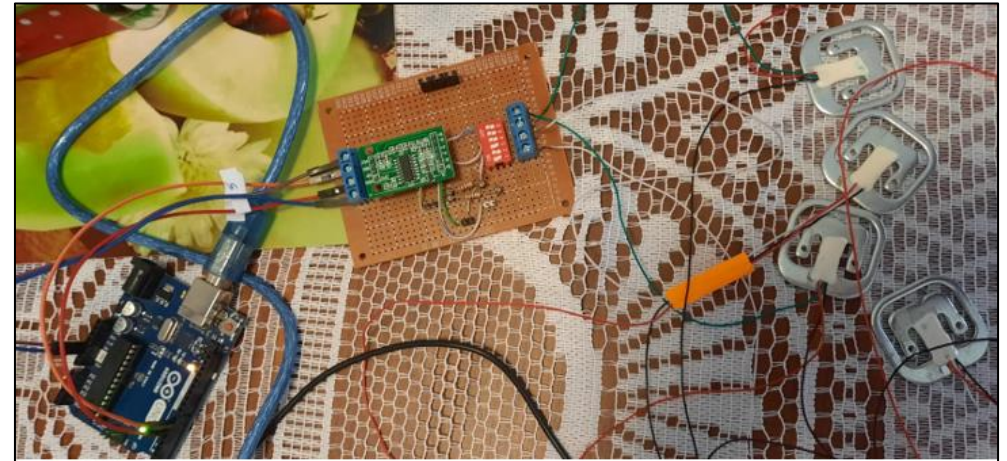
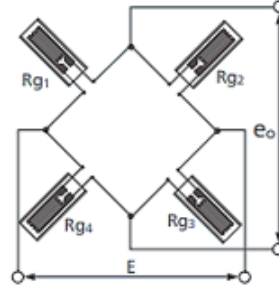
Cuarto de puente



Medio puente



Puente completo



COM4

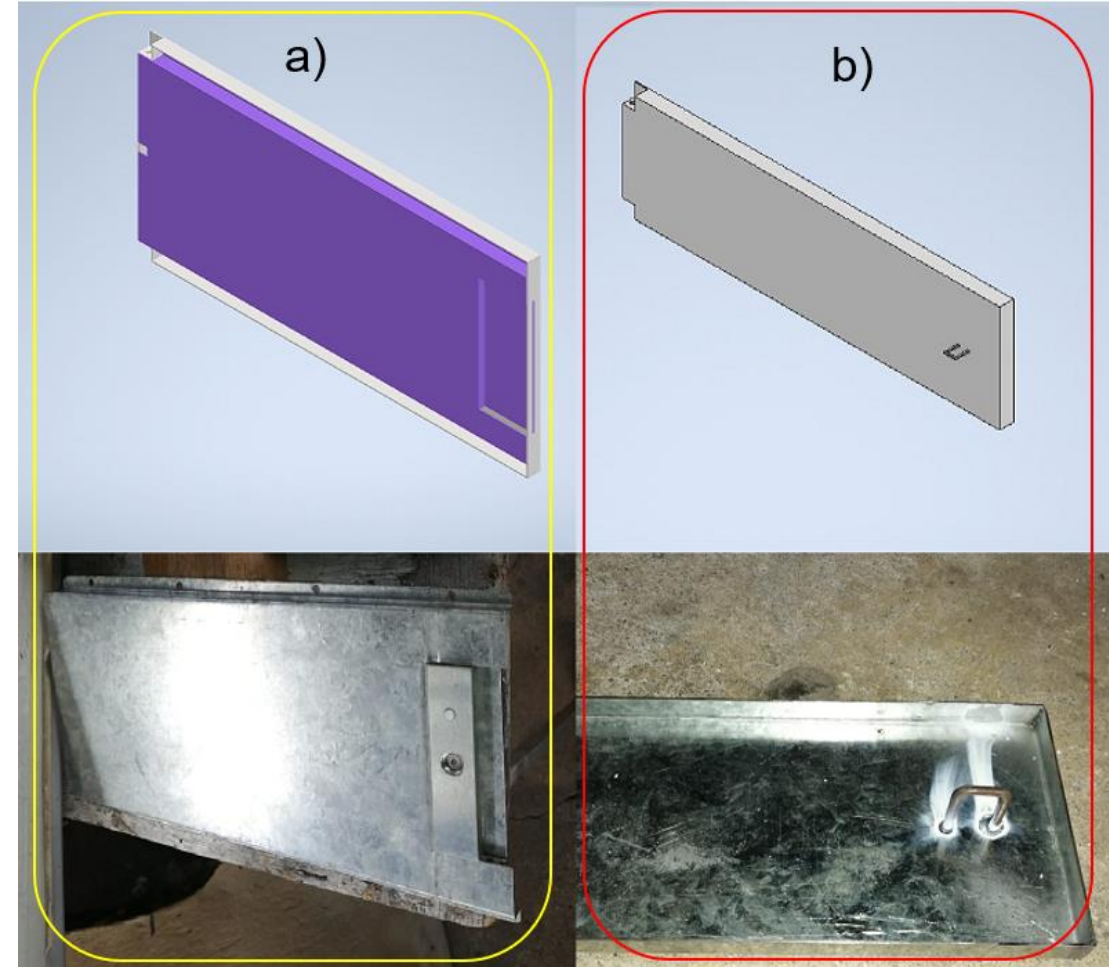
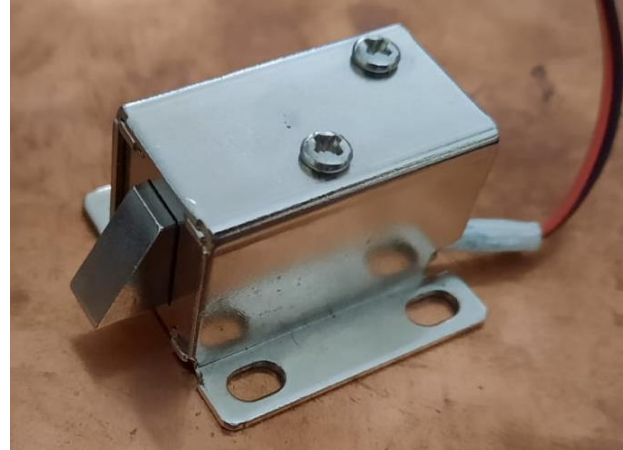
```

Load_cell output val: 89.35
Load_cell output val: 89.98
Load_cell output val: 90.65
Load_cell output val: 90.90
Load_cell output val: 90.67
Load_cell output val: 90.24
Load_cell output val: 89.83
Load_cell output val: 89.62
Load_cell output val: 89.62
Load_cell output val: 89.60
Load_cell output val: 89.76
    
```



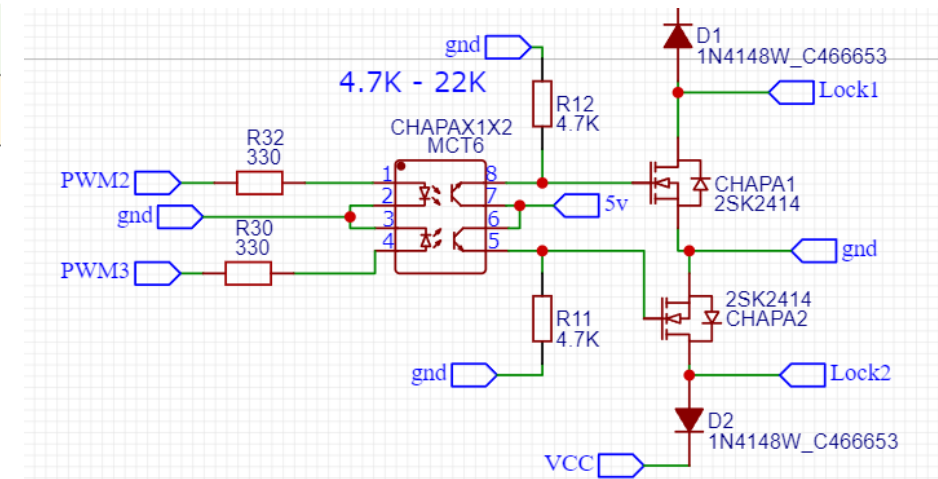
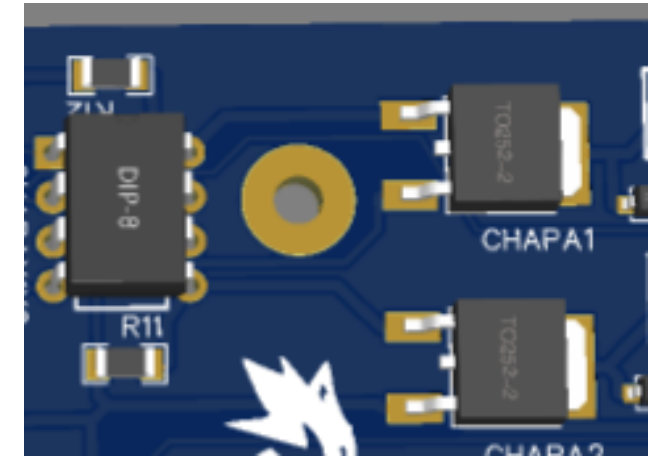
Criterios de selección de cerraduras

Características	Peso	Tipo de cerraduras magnéticas		
		Pequeña (Solenoides)	mediana	grande
Fuerza de cierre	20%	20 kg	60 kg	180 kg
Facilidad de uso	20%	Alto	Alto	Alto
Robustez	20%	Medio	Alta	Alta
Retorno mecánico	5%	si	no	no
Consumo (A) a 12V	15%	0.3	0.4	0.8
Posición inicial	20%	Normalmente cerrado	Normalmente abierto	Normalmente abierto
Total	100%	75%	65%	65%



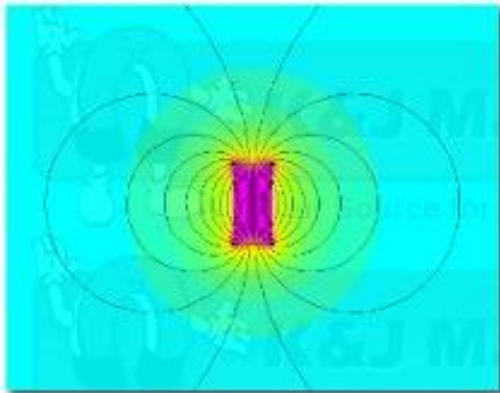
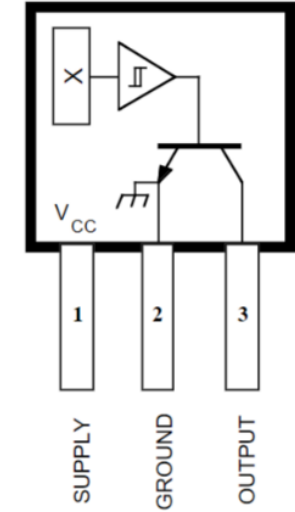
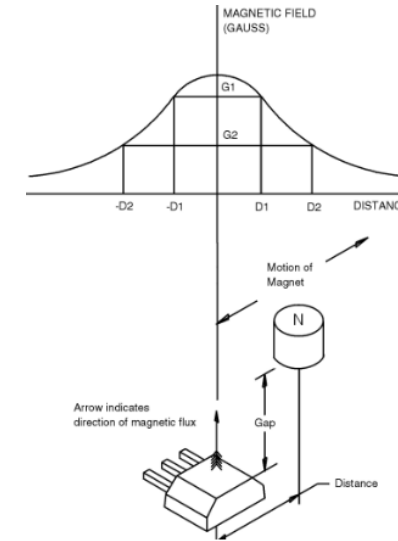
Criterios de selección para elemento de control

Características	Peso	Elemento de control	
		Relé	Optotransistor
Tamaño	15%	grande	pequeño
Principio de funcionamiento	25%	Mecánico	Óptico
Facilidad de uso	25%	Alto	Alto
Robustez	25%	Alta	Media
Señal admisible	10%	Pulso digital	Pulso digital o PWM
Total	100%	65%	75%

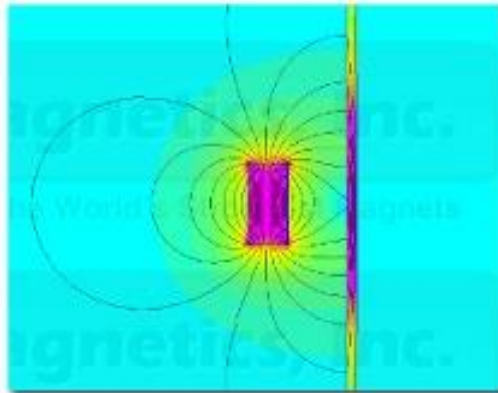


Criterios de selección para final de carrera

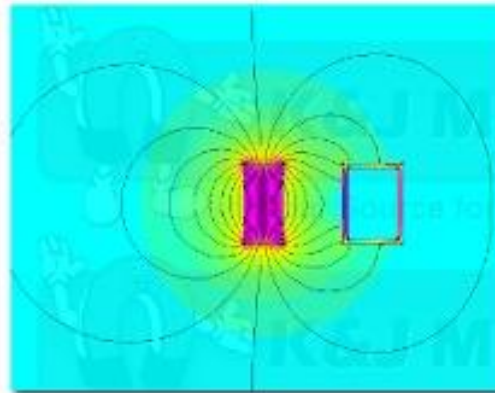
Características	Peso	Sensor	
		Pulsador	Sensor Hall
Tamaño	20%	Pequeño	Pequeño
Principio de funcionamiento	20%	Mecánico	Magnético
Robustez	25%	media	Alta
Visibilidad para el usuario	20%	si	no
Señal de retorno	5%	Digital	Digital
Durabilidad	10%	media	Alta
Total	100%	35%	100%



A magnet in free space



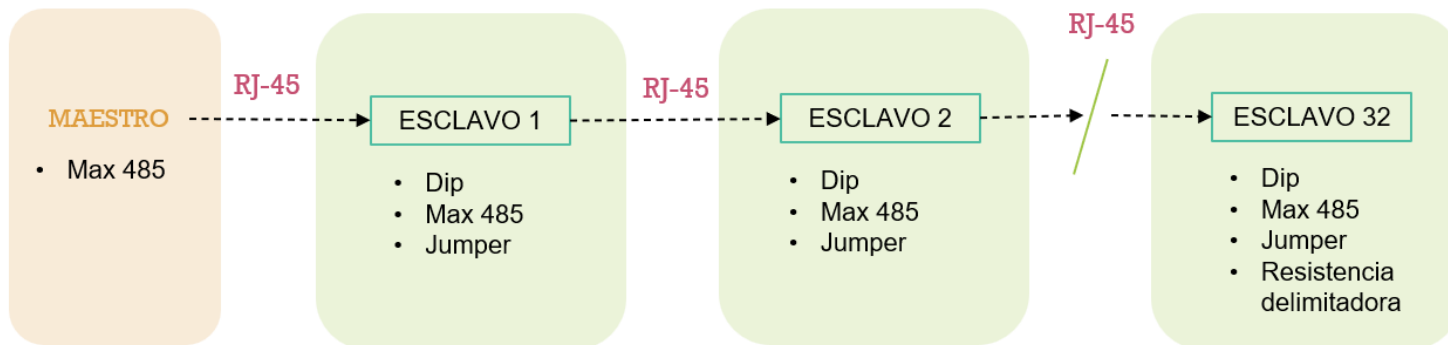
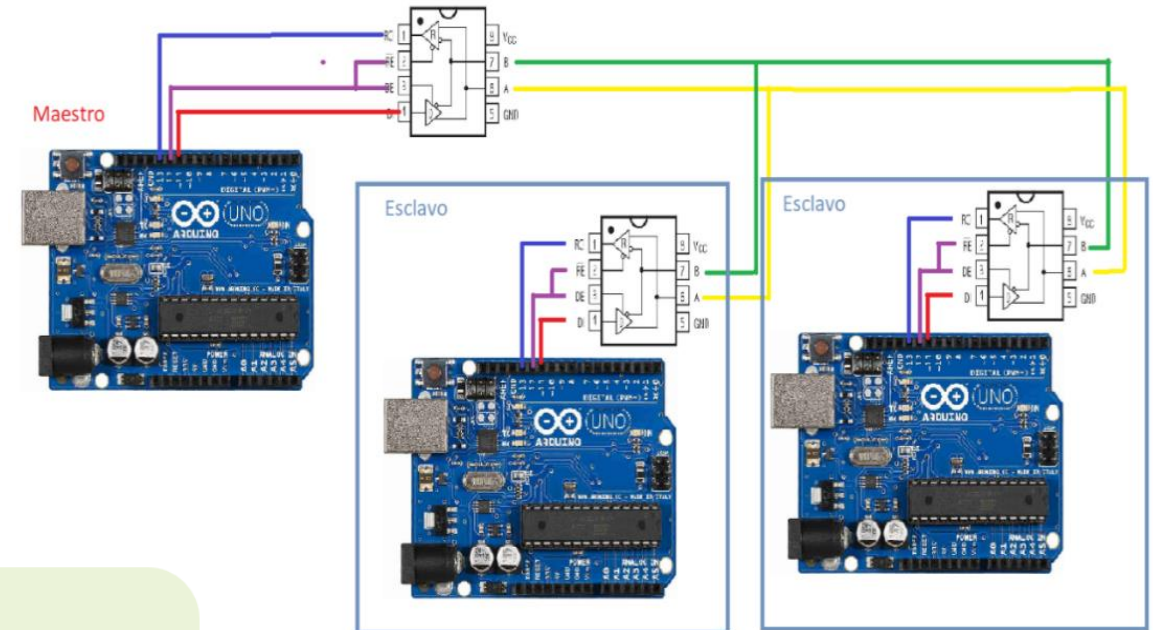
A steel wall



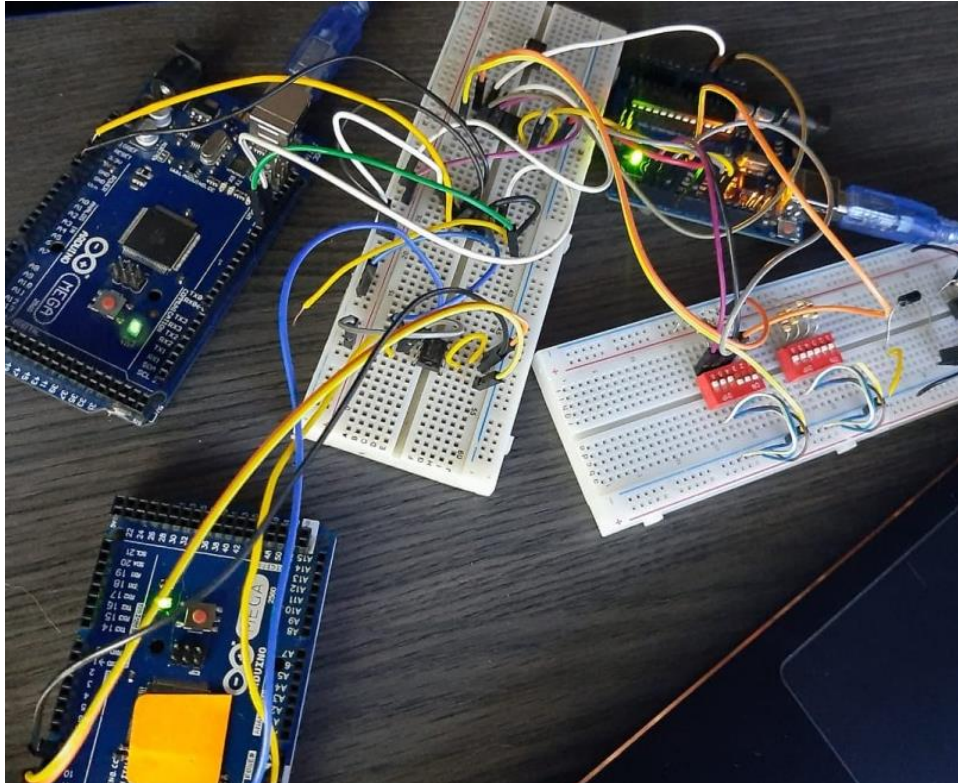
A steel enclosure

Diseño de modularidad

Protocolo	Peso	RS-232	RS-422	RS-485
Modo de trabajo diferencial	20%	No cumple	Si cumple	Si cumple
Número de emisores	10%	1	1	32
Número de receptores	25%	1	Hasta 32	Hasta 32
Topología admisible	25%	Punto a unto	Multipunto y anillo	Bus, multipunto, entre otras
Transmisión	20%	Half dúplex, full dúplex	Full dúplex	Half dúplex
Total	100%	45%	80%	100%



Comunicación 1 maestro con 2 esclavos



```
Maestro Arduino 1.8.15
Archivo Editar Programa Herramientas Ayuda
Maestro
1
2 #include <SoftwareSerial.h>
3 #include <Separador.h>
4 Separador s;
5 const int led=13; //led ON
6 const int entxpin=6; // high: TX, low: RX
7
8
9 SoftwareSerial COM3
10
11 listo
12
13 void setup()
14
15 mys
16 Ser
17 pin
18 pin
19 dig
20 dig
21 }
22
23 void
24 if (Se
25 serialEvent();

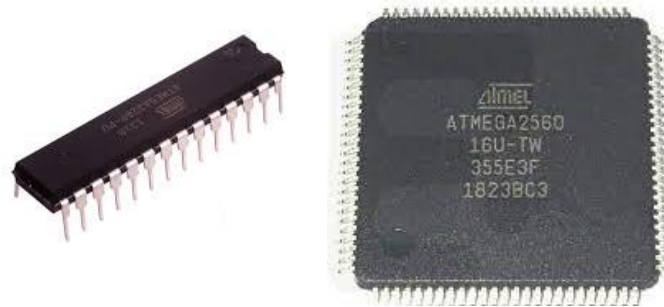
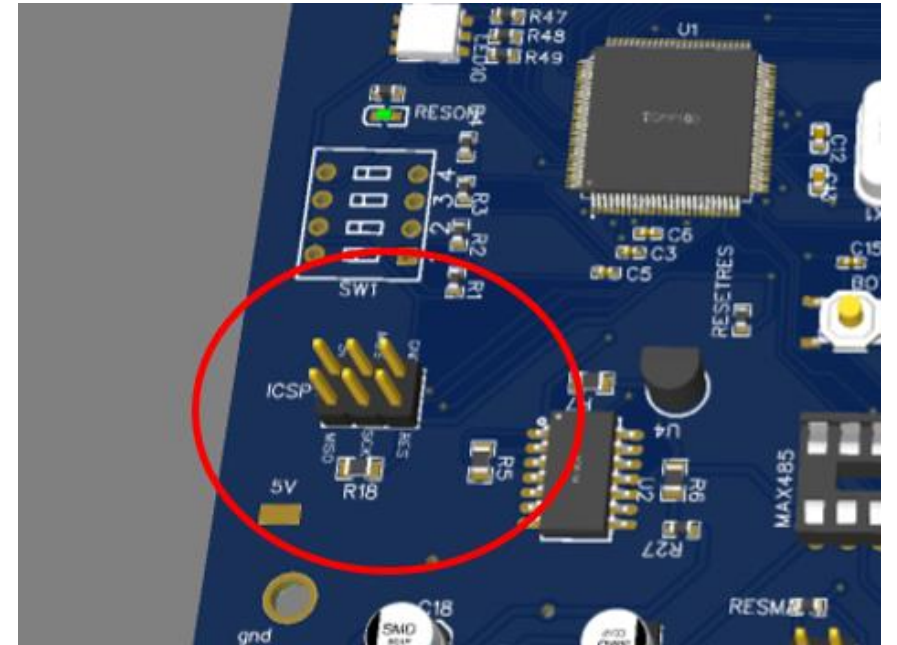
Esclavo Arduino 1.8.15
Archivo Editar Programa H
Esclavo
1
2 #include <S0101
3 #include <S
4 Separador s conectado
5 const int
6 const int c0101
7 const int e
8
9
10 int pin1 = 0101
11 int pin2 =
12 int pin3 =
13 int pin4 =
14
15 int a = 0;
16 int b =
17 int c =
18 int d =
19
20 int aval
21 String v
22 String c
23 char cas
24
25

COM4
Enviar
0101
ESCLAVO 1
0101
0101
0101
0101
Autoscroll Mostrar marca temporal Nueva línea 9600 baudio Limpiar salida

COM5
Enviar
0100
ESCLAVO 2
0100
0100
0100
0100
```

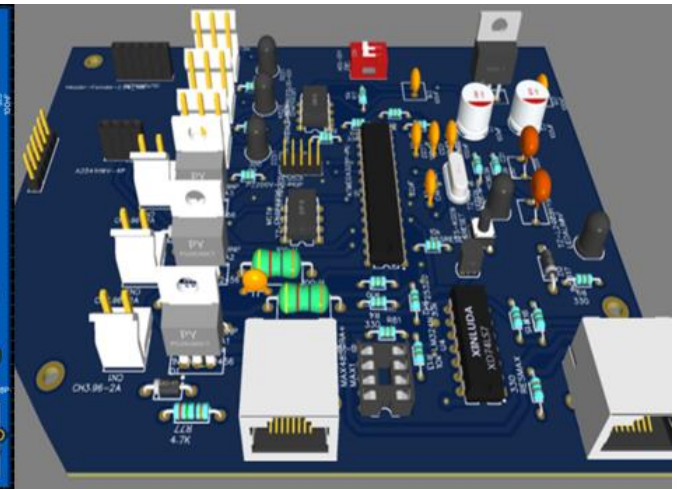
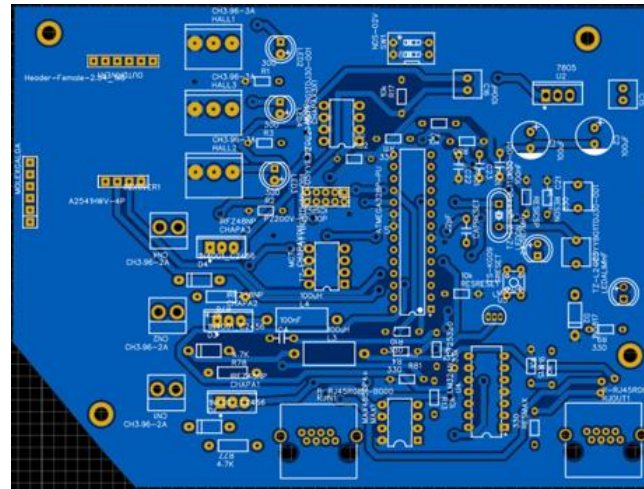
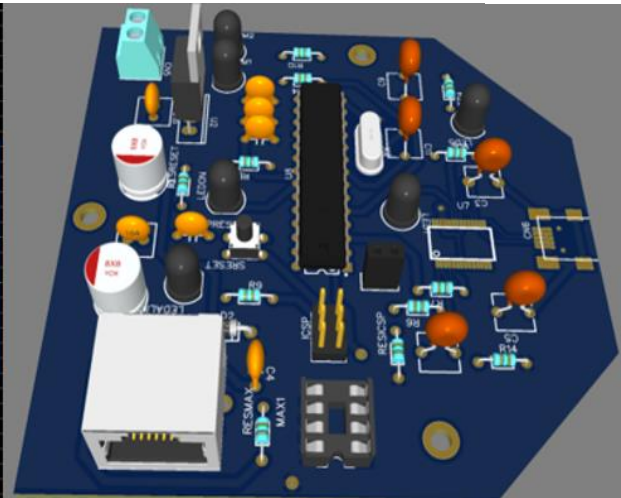
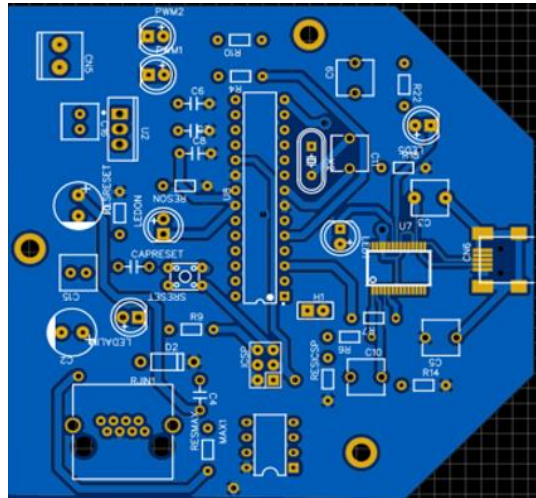

Criterios de selección para microcontrolador

Parámetros	Peso	Microcontrolador	
		Atmega328p	Atmega 2560
Tamaño	10%	mediano	Pequeño
Presentación	10%	Encapsulado	SMD
Robustez	10%	Alta	Alta
ICSP	20%	Si	Si
Complejidad para implementación	15%	Fácil	Media
Pines digitales	15%	14	54
Pines analógicos	15%	12	31
Memoria RAM	5%	2kB	8kB
Total	100%	45%	85%

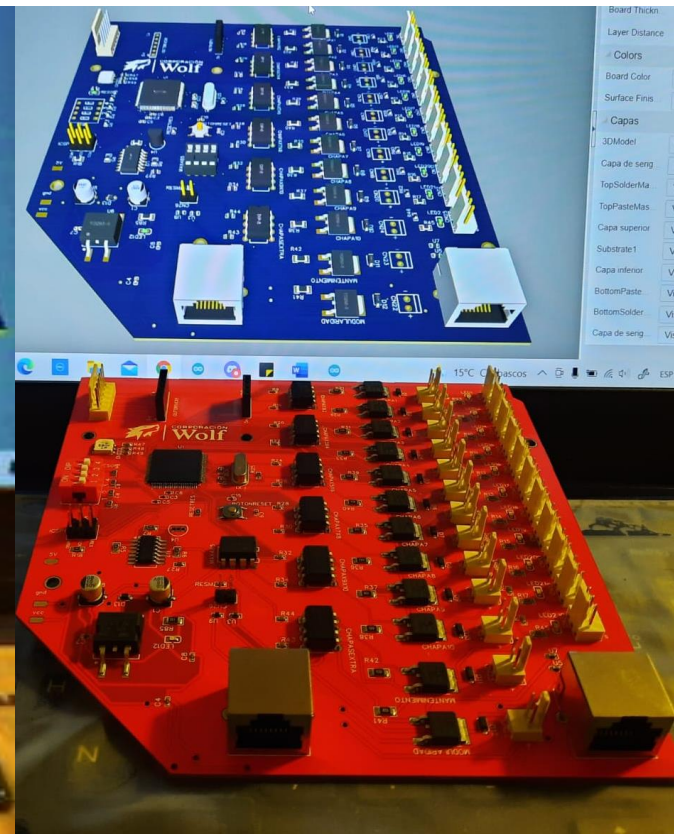
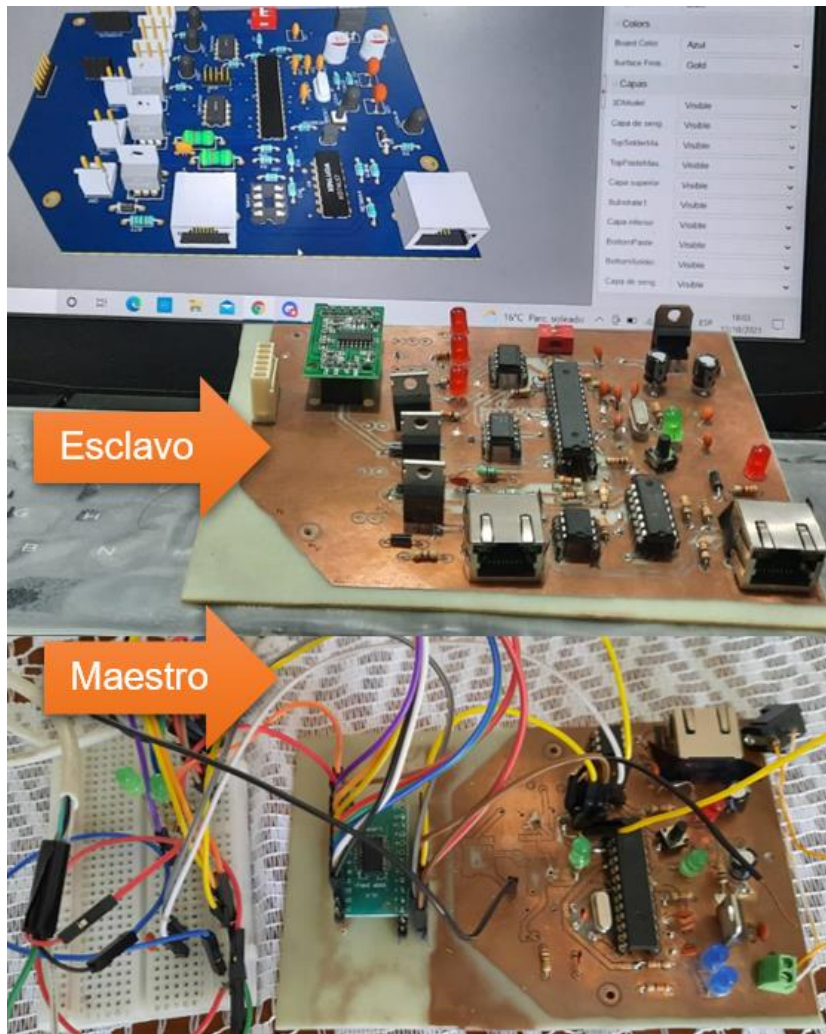


Diseño de pistas maestro y esclavo

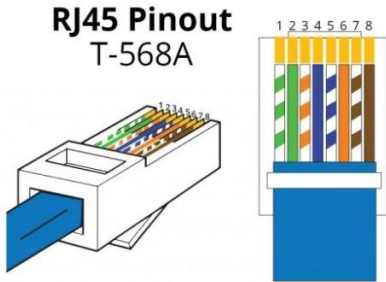
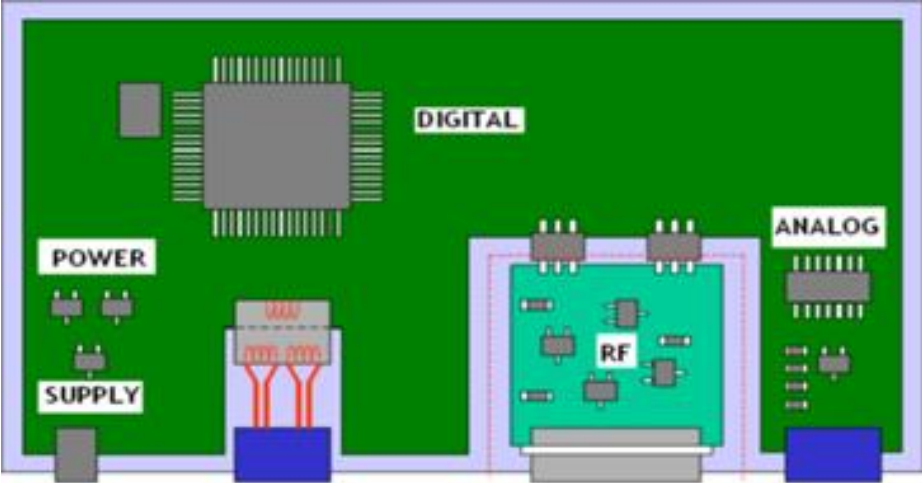
Parámetros	Routeado maestro	Routeado esclavo
Ancho de pista	1 mm	1 mm
Diámetro exterior pad	2mm	2mm
Diámetro de orificio para componentes	0.7 mm	0.7 mm
Capas	2	2
Mallado superior	5v	5v
Mallado inferior	Gnd	Gnd
Espaciado entre pista y mallado	0.3 mm	0.3 mm
Orificios de montaje	3	4
Puerto USB	Si	No



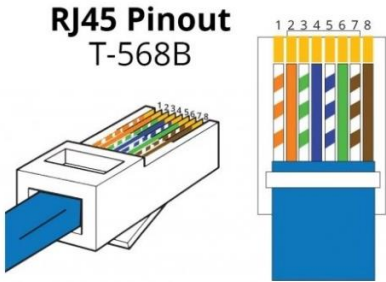
Placas maestro y esclavo



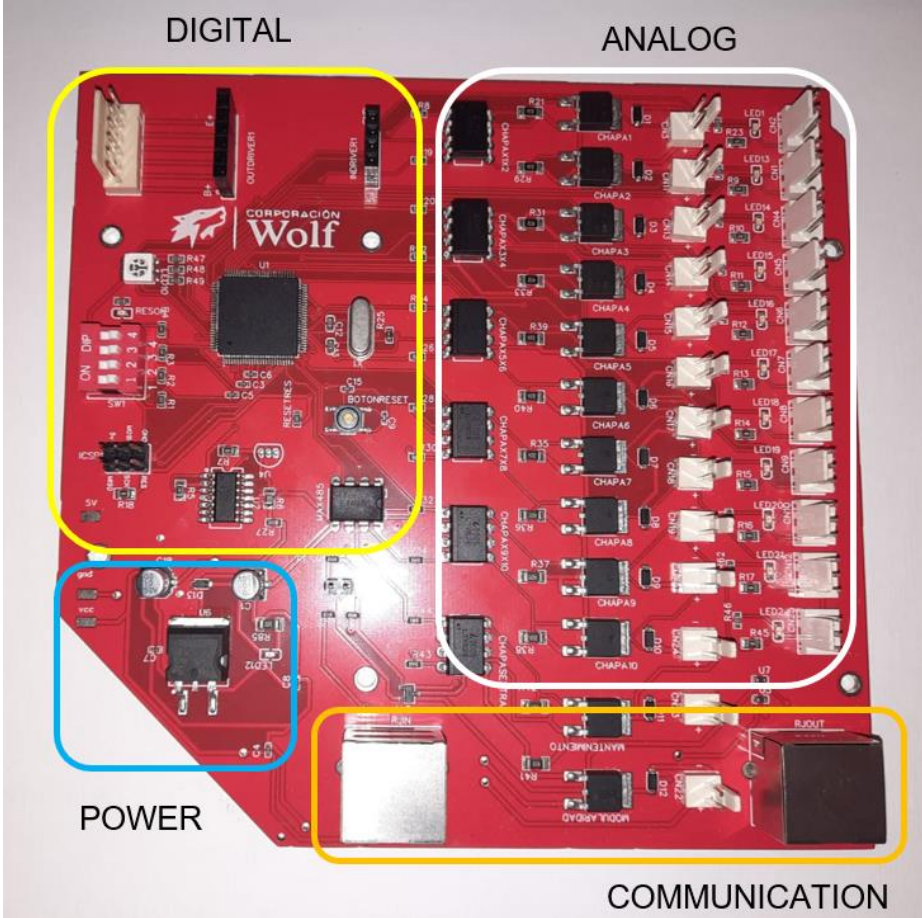
Normativas de diseño en las placas



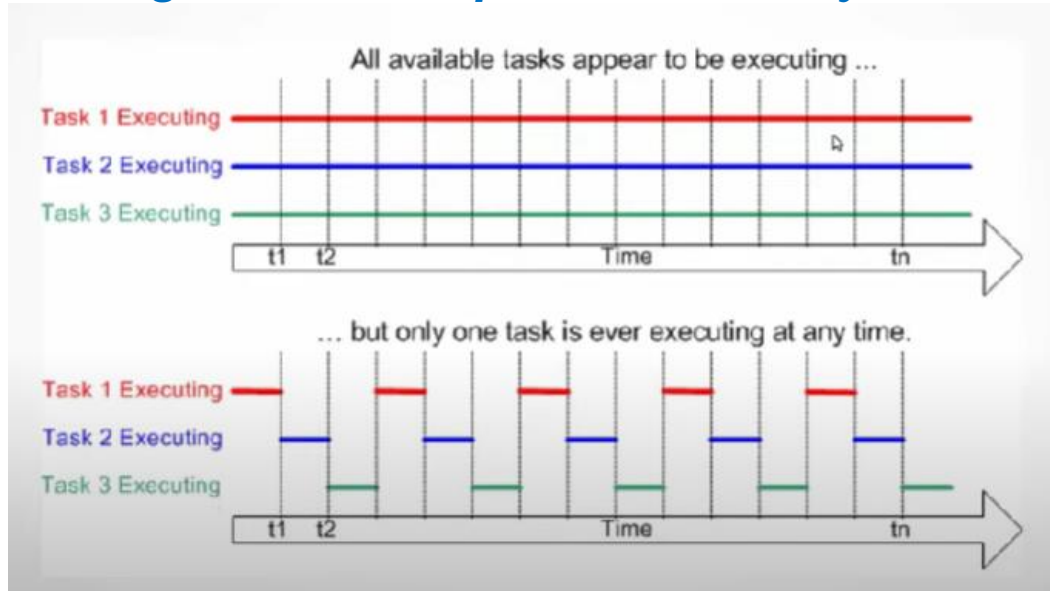
- | | |
|-------------------|------------------|
| 1. Blanco Verde | 5. Blanco Azul |
| 2. Verde | 6. Naranja |
| 3. Blanco Naranja | 7. Blanco Marrón |
| 4. Azul | 8. Marrón |



- | | |
|-------------------|------------------|
| 1. Blanco Naranja | 5. Blanco AZUL |
| 2. Naranja | 6. Verde |
| 3. Blanco Verde | 7. Blanco Marrón |
| 4. Azul | 8. Marrón |



Programación de placas maestro y esclavo



```
void setup()
{
  // Now set up two tasks to run independently.
  xTaskCreate(TaskCommunication, "Communication", 128, NULL, 1, NULL);

  xTaskCreate(TaskSecuenciausuario, "Secuenciausuario", 128, NULL, 2, NULL);

  xTaskCreate(TaskSecuenciaAdmin, "SecuenciaAdmin", 128, NULL, 3, NULL);
  // Now the task scheduler, which takes over control of scheduling individual tasks
}

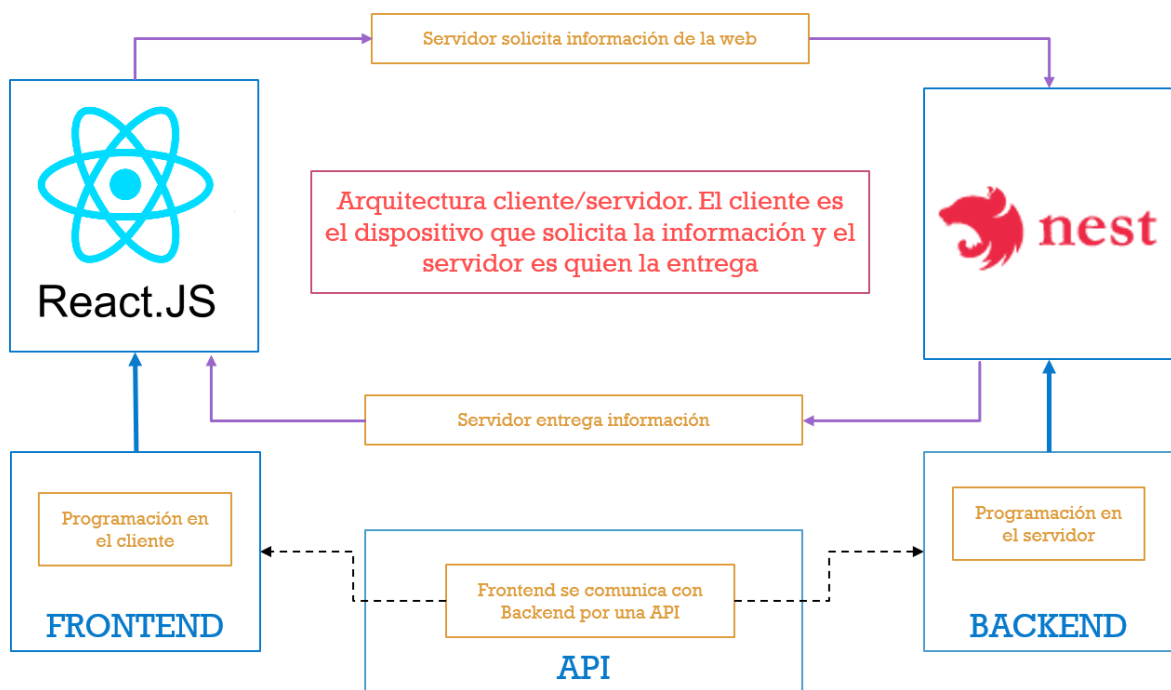
void loop()
{
  // Empty. Things are done in Tasks.
}
```



- ✓ Reducción de tiempo de desarrollo.
- ✓ Refuerza conceptos de modularización
- ✓ Provee de métodos seguros de comunicación entre tareas
- ✓ Asegura la independencia entre tareas
- ✓ Facilita la extensibilidad y verificación del diseño
- ✓ Controla accesos a recursos utilizados por más de una tarea
- ✓ Requiere más memoria RAM del microcontrolador
- ✓ Se reduce de manera significativa el tiempo de respuesta en el sistema



Arquitectura Backend-Frontend

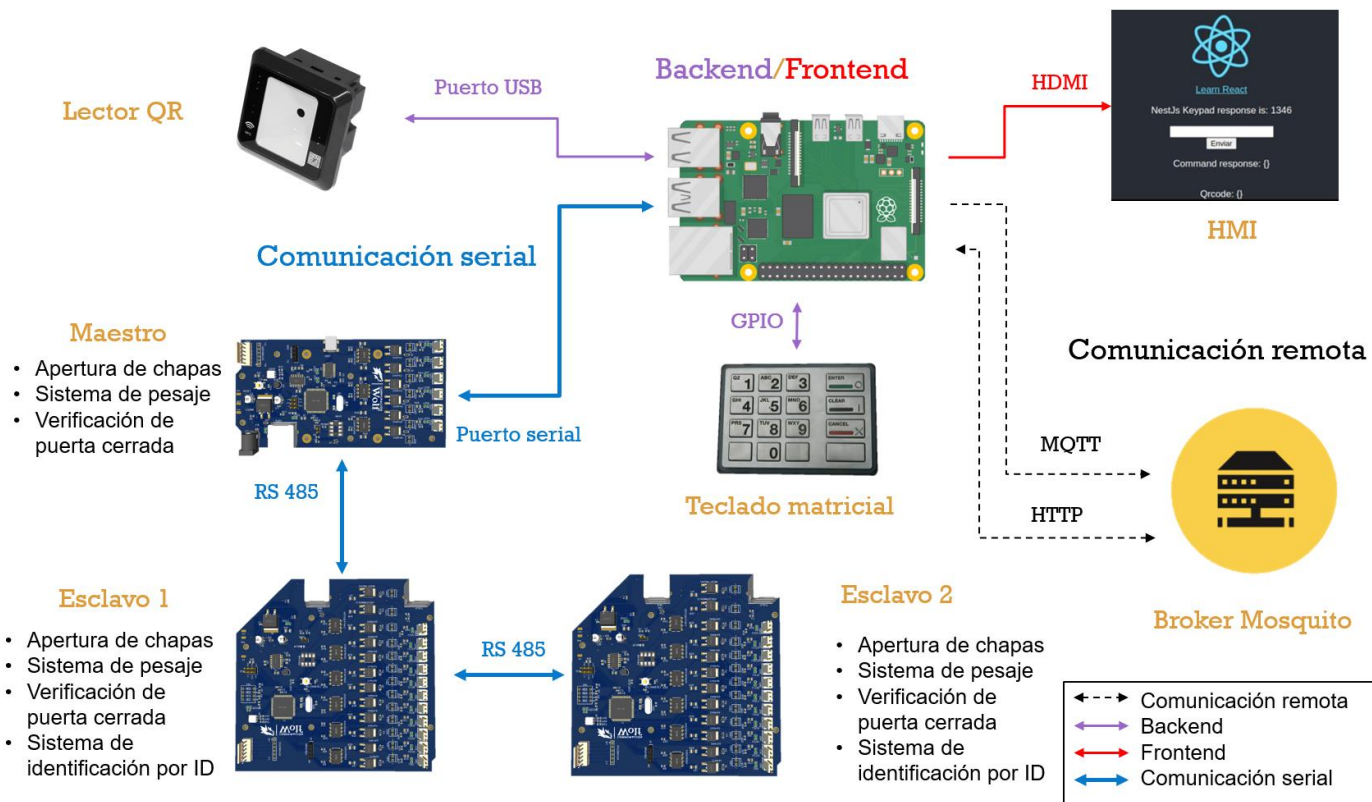


Aplicación	Peso	React	Angular
Desarrollo	10%	Web	Web-Movil
Autosuficiencia	10%	Requiere de librerías adicionales	No requiere de librerías adicionales
Curva de aprendizaje	20%	Estructura simple y tiene constantes actualizaciones	Sintaxis innecesaria y administración compleja
Desempeño	15%	Reduce la carga a los navegadores	Cuantas más vinculaciones tenga, es más lento el proceso
Lenguaje	20%	JavaScript ES6 + JSX script que se asemeja a HTML	Usa JavaScript o TypeScript
Estructura de aplicación	20%	No existe "una única estructura correcta"	La estructura es fija y compleja
Herramientas	5%	Compatible con Visual Studio, Atom, Sublime Text	Compatible con Visual Studio, Atom, Sublime Text
Total	100%	90%	65%

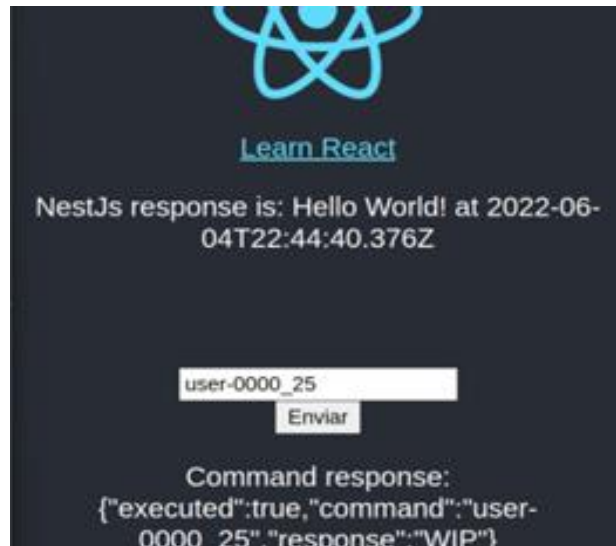
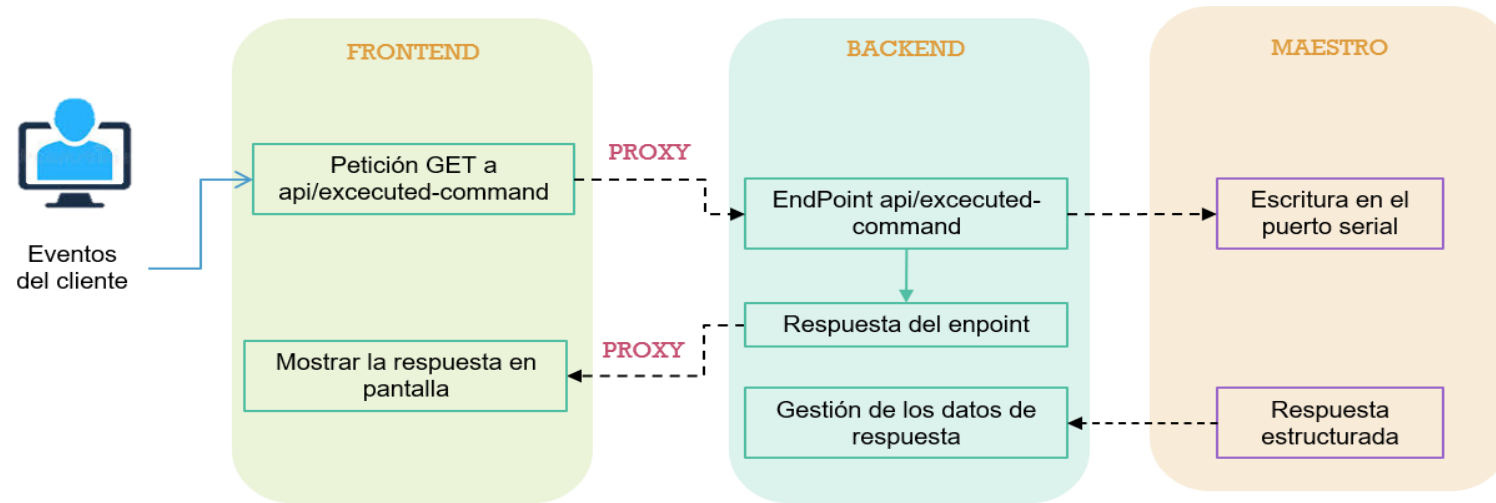


Protocolos de comunicación

Protocolo	Peso	MQTT	CoAP	HTTP
Protocolo Base	10%	TCP	UDP	TCP
Consumo de energía	15%	Mayor que CoAP	Menor que MQTT	Mayor que MQTT
Número de tipos de mensaje usados	10%	16	4	Indefinido (depende del servidor web)
Tamaño de la cabecera	5%	2 bytes	4 bytes	Indefinido
Fiabilidad	20%	3 niveles de calidad de servicio QoS 0: Entrega no garantizada QoS 1: Confirmación de entrega QoS 2: Doble confirmación de entrega Fácil de implementar y complejo para añadir extensiones Puede utilizar TLS/SSL	Mensajes confirmables Mensajes no confirmables ACKs Retransmisiones	Limitado al protocolo TCP
Implementación	15%	Fácil de implementar y complejo para añadir extensiones	Pocas librerías existentes y soporte	Fácil de implementar, no siempre está optimizado
Seguridad	15%	Útil para conexiones en localizaciones remotas	DTLS o IPSec	TLS/SSL
Otro	10%		Baja saturación Baja latencia	Útil para envío de paquetes grandes y capacidad de respuesta
Total	100%	85%	40%	60%

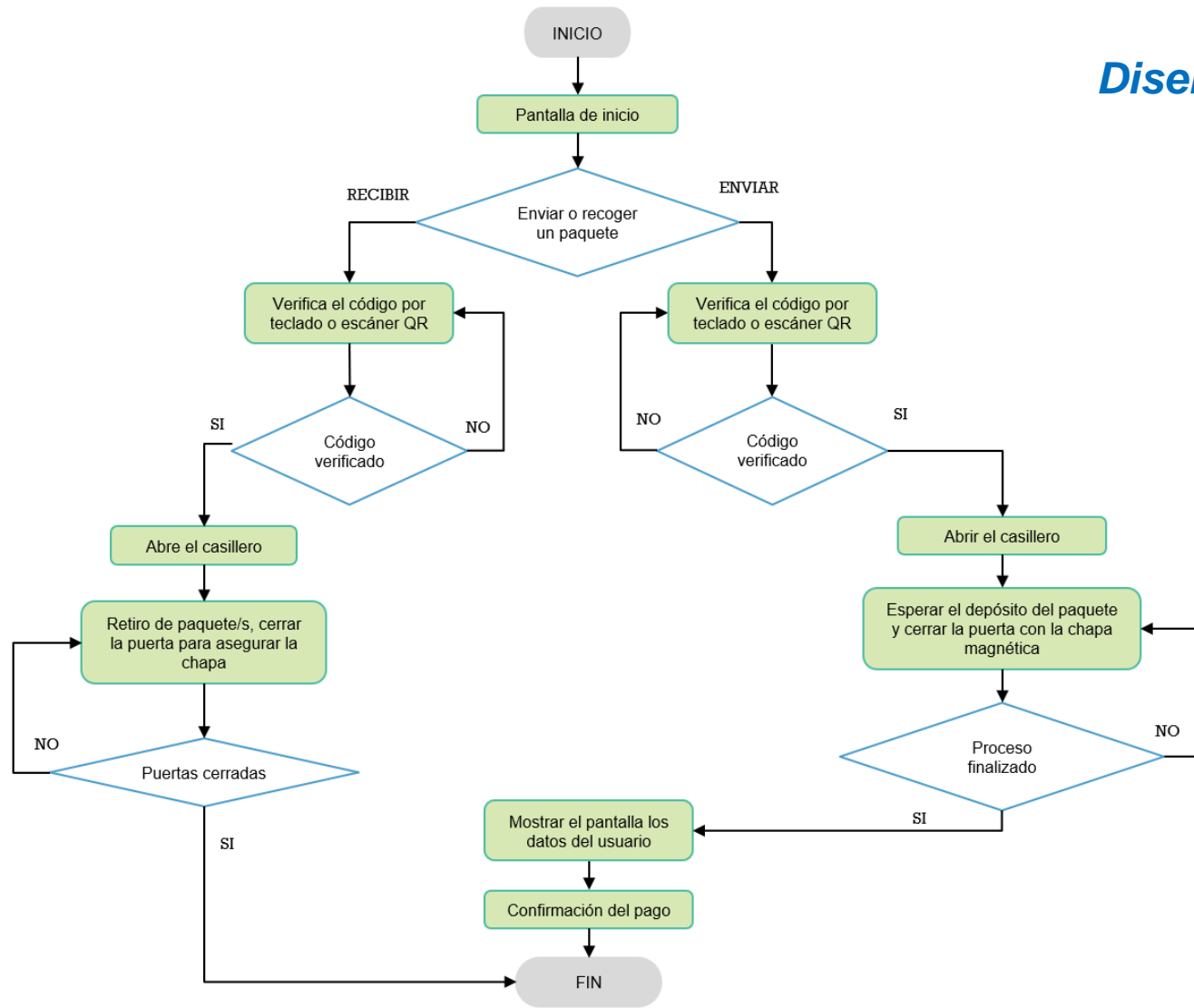


Eventos por trama

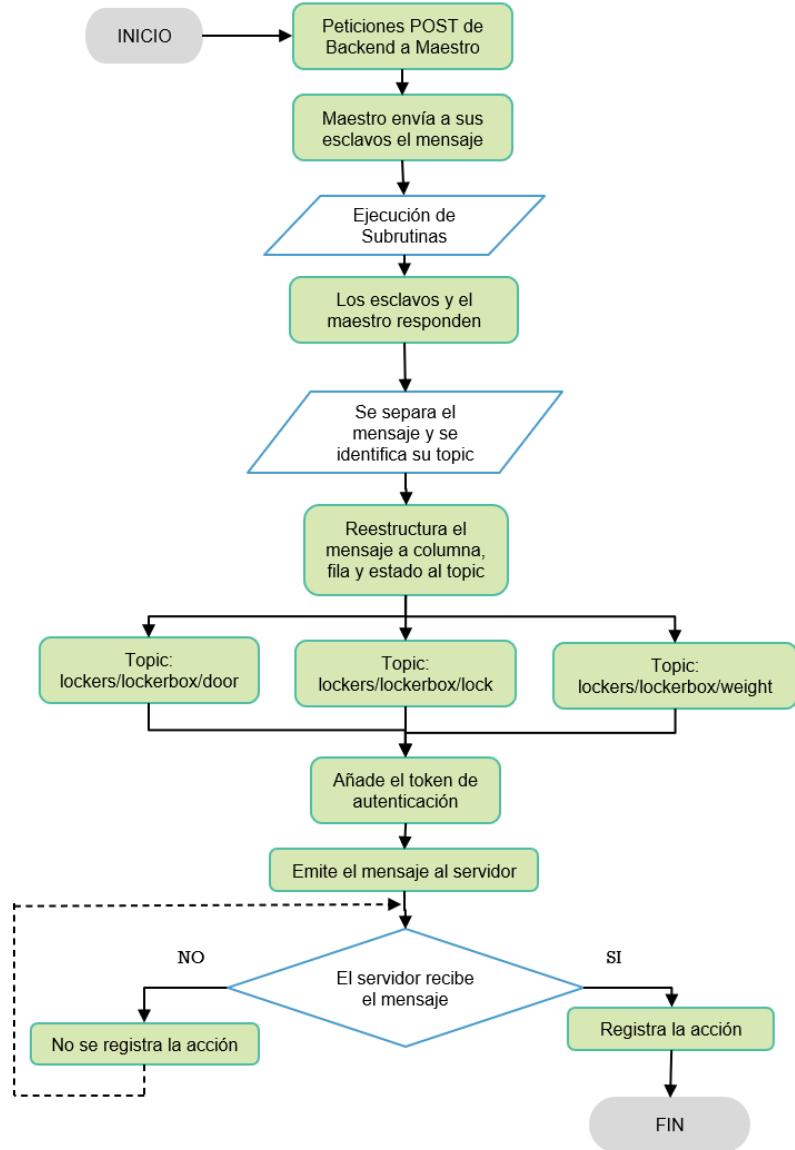


```
pi@raspberrypi: ~/lockers-raspberry-platform
Archivo Editar Pestañas Ayuda
compiled successfully!
You can now view raspberry-frontend in the browser.
  Local:      http://localhost:8000
  On Your Network:  http://192.168.106.25:8000
Note that the development build is not optimized.
To create a production build, use npm run build.
Parser got word from serial: #
parser got word from serial: ID:1212
parser got word from serial: Locker y accion:
parser got word from serial:
parser got word from serial: Usuario incorrecto, ingrese nu
parser got word from serial:
parser got word from serial: ID:alvaro
parser got word from serial: Locker y accion:16_11
parser got word from serial:
AC
```

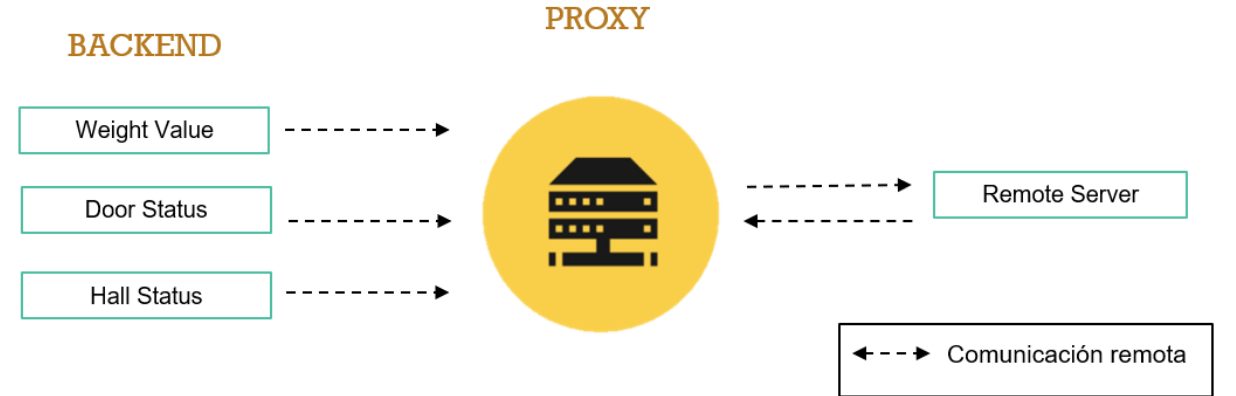
Diseño de HMI



Módulo	Ejecución
Aplicación web administrador	<ul style="list-style-type: none"> • Conectividad al servidor remoto MQTT • Conectividad al servidor remoto HTTP • Monitoreo de casilleros por comandos • Expansión modular
Aplicación web usuario	<ul style="list-style-type: none"> • Validación de códigos de usuario • Visualización del proceso • Monitoreo de casilleros por comandos • Pasarela de pagos
Funcionalidad del prototipo HMI	<ul style="list-style-type: none"> • Lectura de código QR • Lectura del teclado matricial • Conexión al servidor remoto • Proceso: Apertura de casillero • Proceso: Cierre de casillero



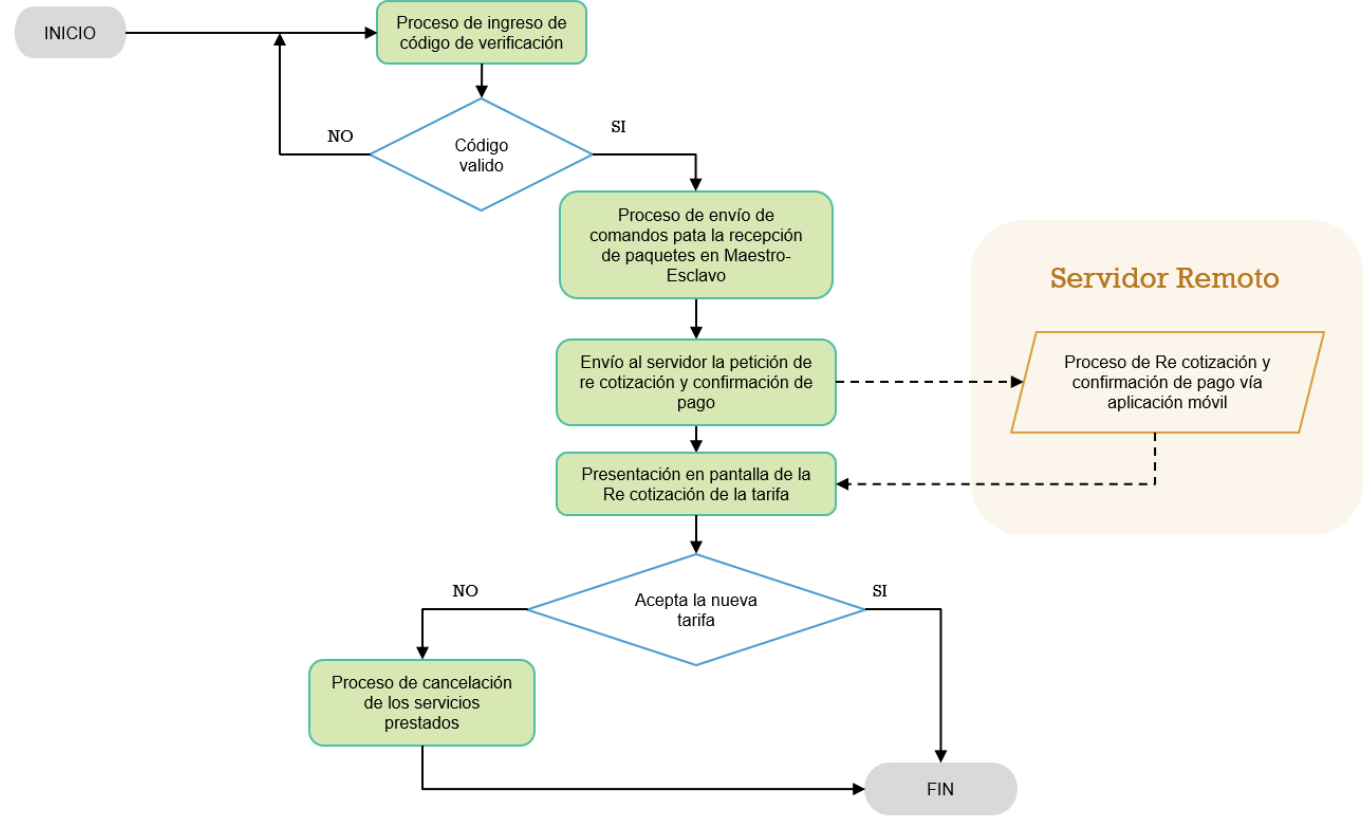
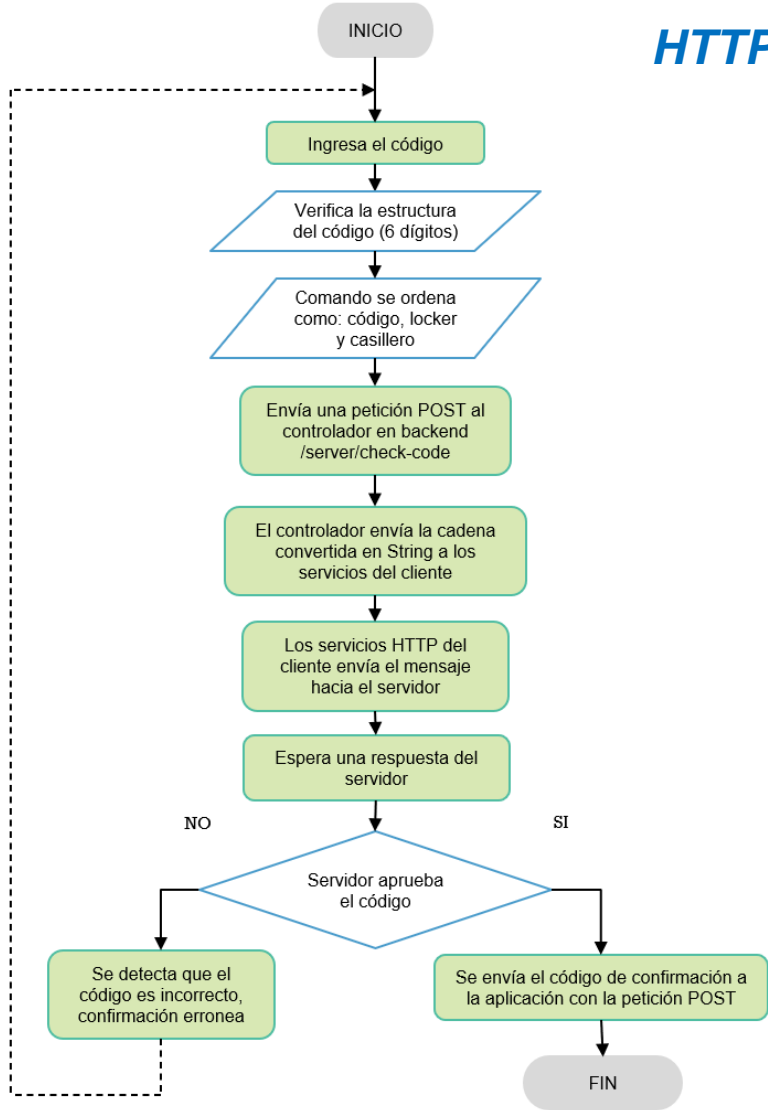
MQTT para estados del locker



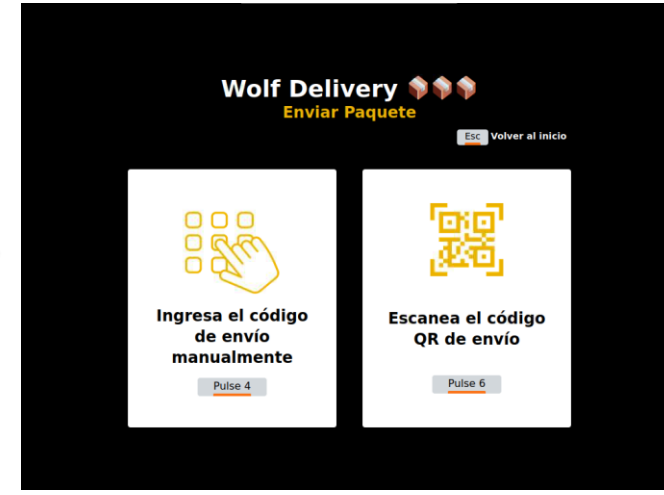
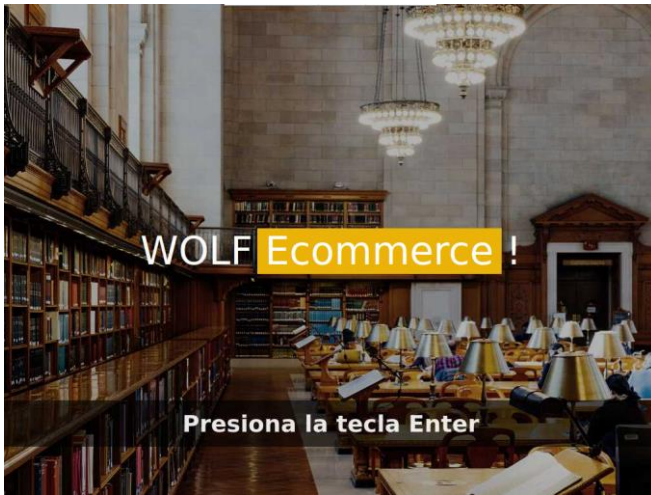
Comandos	Descripción	Canal MQTT
Lock Status	Se usa cuando el seguro está cerrado ("l") o abierto ("u")	sendLockerBoxLockStatus
Door Status	Responde al estado del sensor con los comandos "o" para abierta y "c" cuando está cerrada	sendLockerBoxDoorStatus
Box Weight	Se aparta el valor del peso al final de la trama	sendLockerBoxWeight



HTTP para validar información (código y pasarela de pagos)

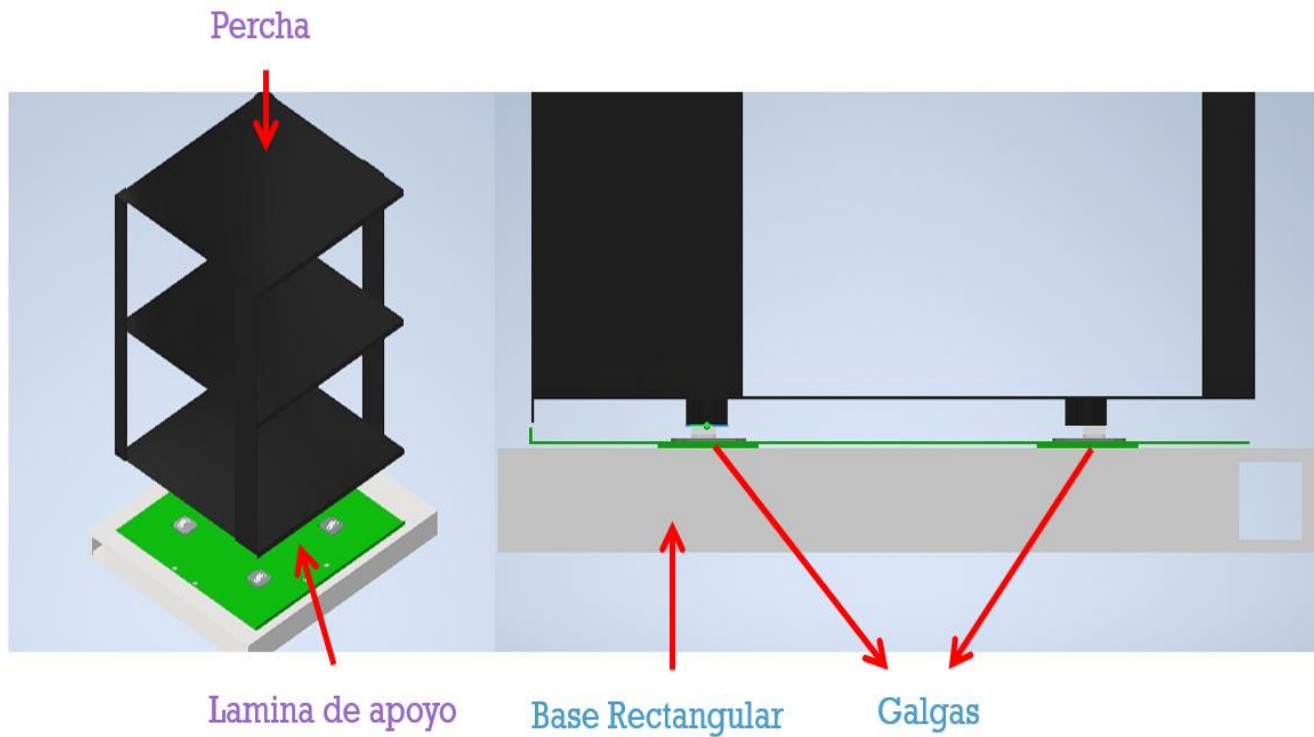


Secuencia de la interfaz

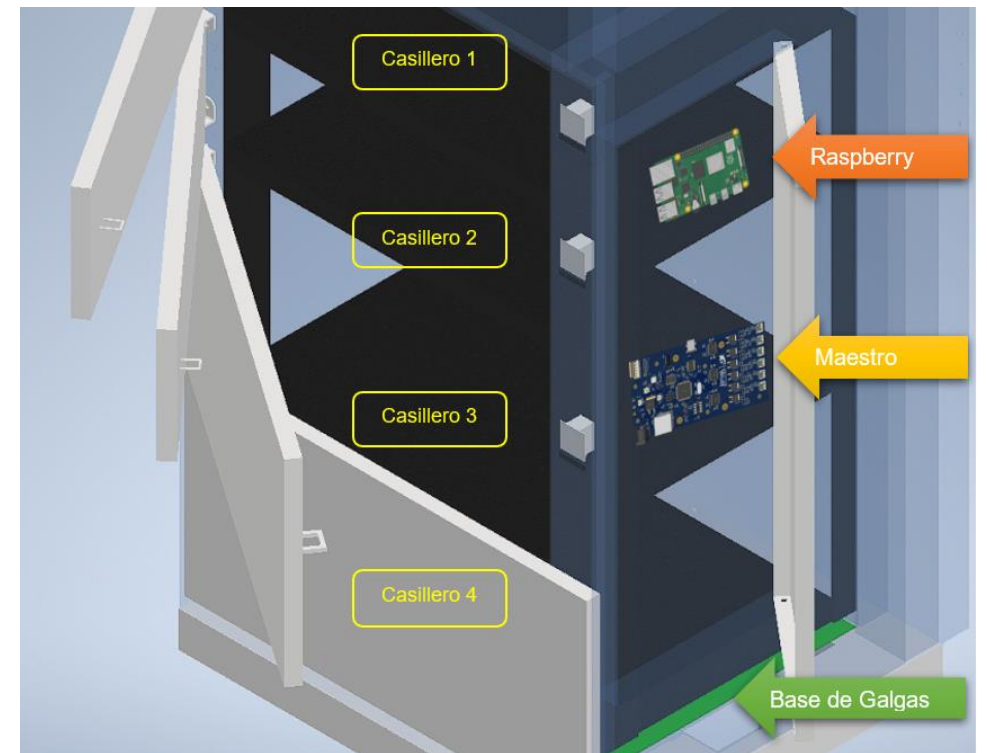


Diseño mecánico

Subsistema de pesaje



Armazón



Subsistema de pesaje

Diseño de la percha interna

Dimensionamiento de las columnas para la carga

$$\# \text{ pisos} = 3$$

$$d_{\text{pisos}} = 220 \text{ mm}$$

$$\text{carga en cada piso } (C_{cp}) = 2.5 \text{ kg}$$

$$\text{Area de la plancha por piso } (A_p) = 358 \times 340 \text{ mm}$$

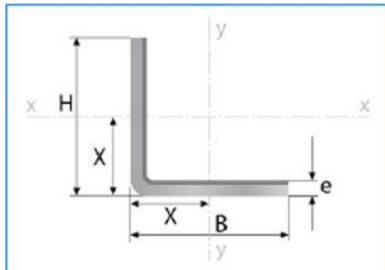
Material empleado ASTM A36 con resistencia a la fluencia (S_y) de:

$$S_y = 2528.9 \frac{\text{kg}}{\text{cm}^2}$$

$$A_g = \frac{187.66 \text{ kg}}{2528.9 \frac{\text{kg}}{\text{cm}^2}}$$

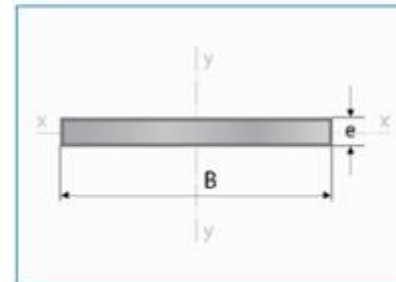
$$A_g = 0.0742 \text{ cm}^2$$

Área recomendada $> 0.0742 \text{ cm}^2$



Largo Normal:
6 metros
Recubrimiento:
Negro o Galvanizado
Calidad de Acero:
ASTM A36 / ASTM A572 Gr 50 /
ASTM A653

Designaciones	Esp.	Peso	Propiedades Estáticas								
			Eje x-x			Eje y-y			Distancia de eje y-y superficial		
			Momento de inercia	Módulo de resistencia	Radio de giro	Momento de inercia	Módulo de resistencia	Radio de giro			
H	B	e	P	P	I	W	i	I	W	i	X=Y
mm	mm	mm	kg/m	kg/6m	cm ⁴	cm ³	cm	cm ⁴	cm ³	cm	cm
20	20	1,50	0,44	2,61	0,21	0,15	0,47	0,21	0,15	0,47	0,54
		2,00	0,58	3,46	0,28	0,20	0,62	0,28	0,20	0,62	0,59
		3,00	0,83	4,96	0,38	0,28	0,60	0,38	0,28	0,60	0,64
25	25	1,50	0,55	3,32	0,42	0,24	0,59	0,42	0,24	0,59	0,67
		2,00	0,73	4,38	0,56	0,32	0,78	0,56	0,32	0,78	0,72
		3,00	1,06	6,36	0,79	0,45	0,76	0,79	0,45	0,76	0,77
30	30	1,50	0,67	4,03	0,75	0,35	0,71	0,75	0,35	0,71	0,79
		2,00	0,89	5,34	1,00	0,46	0,94	1,00	0,46	0,94	0,84
		3,00	1,30	7,78	1,41	0,67	0,92	1,41	0,67	0,92	0,89



Largo normal:
6 metros
Recubrimiento:
Negro
Espesores:
Desde 3,00 a 12,00 mm
Calidad de Acero:
ASTM A36 / ASTM A 572 Gr 50
Norma de Fabricación:
INEN 2222 / ISO 1035 /

Dimensiones	Espesor	Peso
B	e	P
Pulg.	mm	kg/6m
1/2 x 1/8	3,00	1,70
1/2 x 3/16	4,00	2,26
1/2 x 1/4	6,00	3,39
3/4 x 1/8	3,00	2,68
3/4 x 3/16	4,00	3,58
3/4 x 1/4	6,00	5,37
1 x 1/8	3,00	3,53
1 x 3/16	4,00	4,71
1 x 1/4	6,00	7,07
1 1/2 x 1/4	6,00	10,74
1 1/2 x 3/8	8,00	16,11
1 1/2 x 1/2	12,00	21,48
2 x 1/8	3,00	7,07
2 x 3/16	4,00	9,50

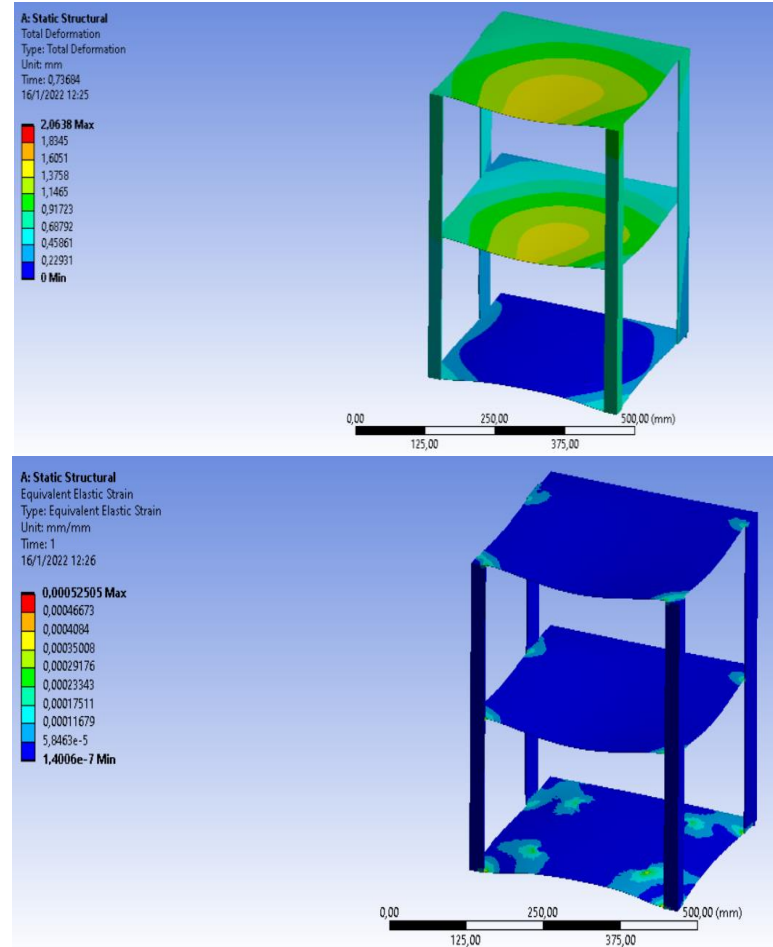


ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Subsistema de pesaje

Análisis de la percha interna

Mediante el análisis estructural de la percha por el método de elementos finitos, se obtuvieron los resultados de equivalente estático y deformación total de la estructura, ratificando el correcto diseño para una percha de tres pisos, teniendo como resultado una deformación aceptable para la recepción de paquetes.



Subsistema de pesaje

Diseño de la lámina de apoyo

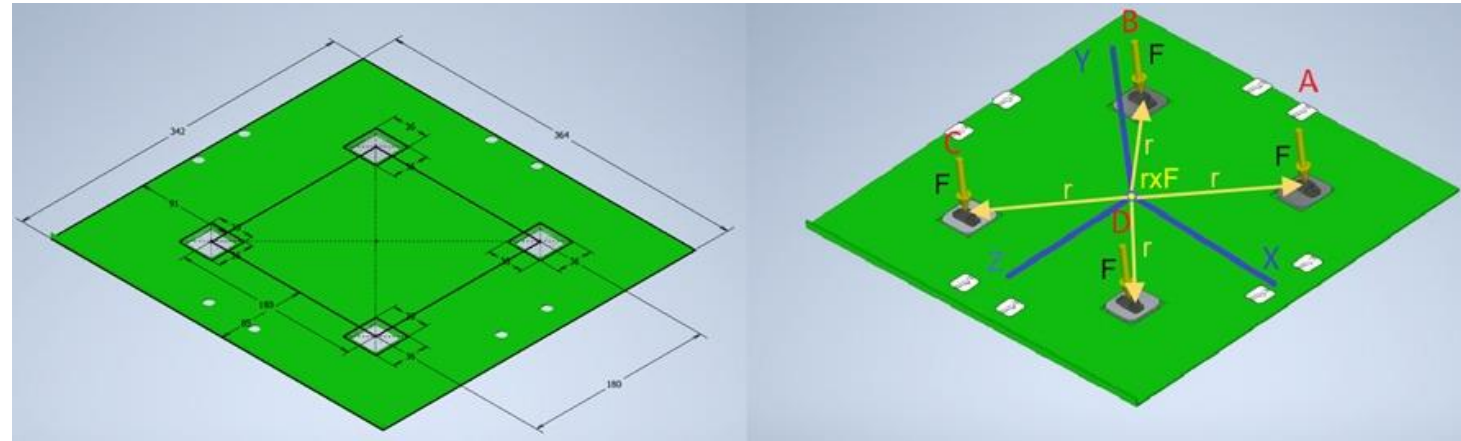
Para el cálculo se analizó el primer prototipo de percha con 3 pisos, en cada uno se estima un paquete de 2 kg dando como resultado final 6 kg, además, se suma el peso de la percha estimado en 2 kg. El peso total de la percha con paquetes es de 8 kg, se divide para los 4 puntos de apoyo obteniendo:

$$\text{Masa en cada pata} = 2\text{kg}$$

Se calcula el peso de cada pata en función de la masa planteada:

$$w = 2\text{kg} * 9.81 \frac{m}{s^2}$$

$$w = 19.6\text{N}$$



F	r [mm]	F [N]	rxF [kg.mm]
A	90i-90k	-19.6j	-1764i-1764k
B	-90i-90k	-19.6j	-1764i+1764k
C	-90i+90k	-19.6j	1764i-1764k
D	90i+90k	-19.6j	1764i+1764k
Total (Σ)		-78.4j	0i+0k



Subsistema de pesaje

Análisis de la lámina de apoyo

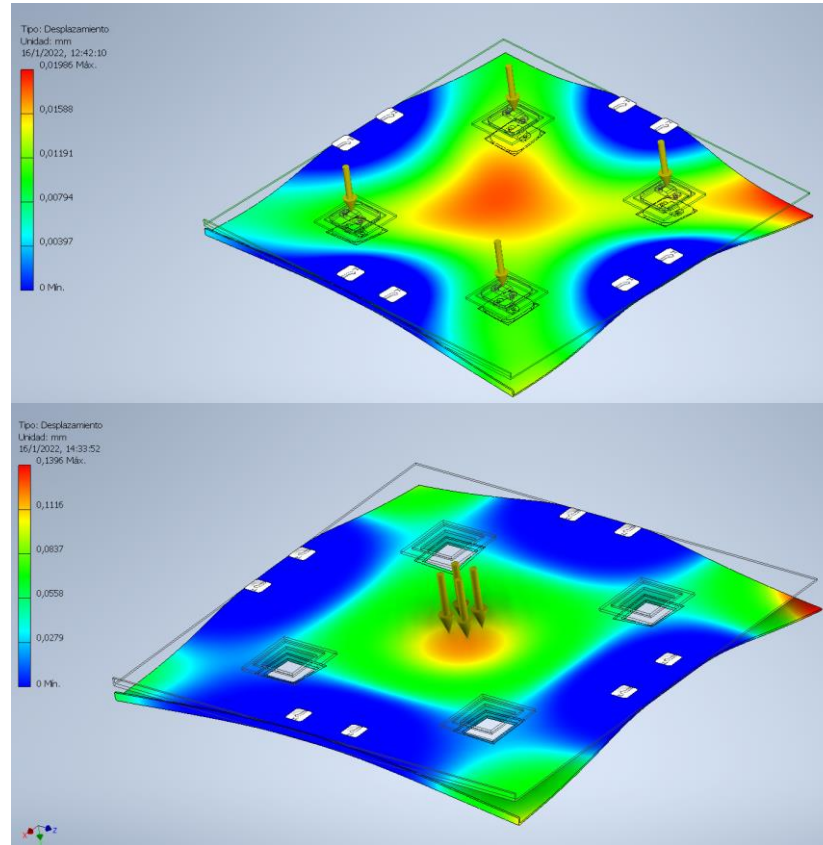
Calculando el esfuerzo máximo en el centro de la lámina de apoyo

$$\sigma_0 = \frac{F_0}{A}$$

$$\sigma_0 = \frac{-78.4N}{0.1245m^2}$$

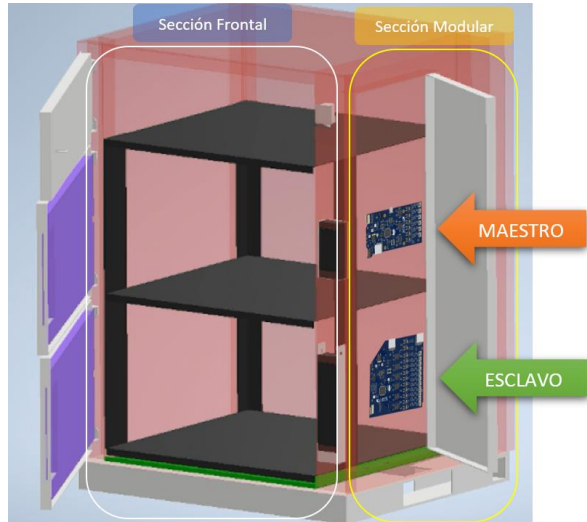
$$\sigma_0 = -632.25Pa$$

El análisis de deformación muestra como resultado que, pese a que la sumatoria de fuerzas en el centro de la placa es nula, se genera un esfuerzo que provoca una deformación máxima de 0.1396 mm, calificándolo como aceptable para soportar la estructura de la percha y los paquetes.

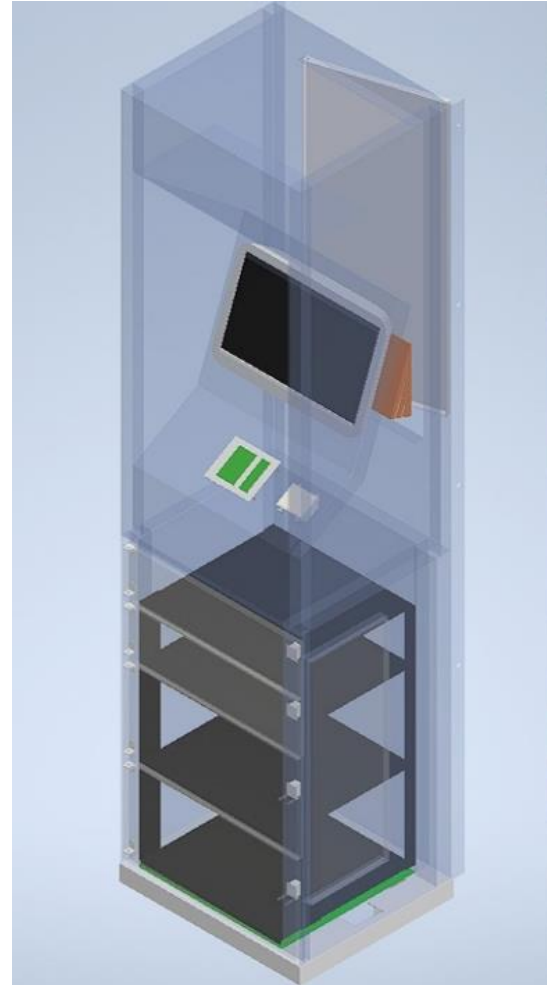


Armazón

Primer prototipo



Locker final



Funcionamiento del sistema

The image displays a system interface for a QR-based payment or identification system. It is divided into three main sections:

- Top Left:** A live video feed from an iVCam showing a hand holding a smartphone with a QR code in front of a scanner. The scanner screen displays a QR code and the text "Coloca el código QR frente del lector".
- Top Center:** A VNC Viewer window titled "celu (raspberrypi): VNC Viewer" showing the same scanner interface.
- Bottom:** A code editor (VS Code) showing a TypeScript file with a `SerialPort` binding and a console output window displaying JSON data.

The console output shows the following JSON data:

```
alvaro-Master_11  
[ 'Master', '1', 'u' ]  
Chapa: [ 'Master', '1', 'u' ]  
[ 'Master', '1', 'o' ]  
Puerta: [ 'Master', '1', 'o' ]  
[ 'Master', '1', 'c' ]  
Puerta: [ 'Master', '1', 'c' ]  
[ 'Master', '1', 'l' ]  
Chapa: [ 'Master', '1', 'l' ]  
[ 'Master', '1', '1740' ]  
Peso: [ 'Master', '1', '1740' ]  
TRAMA Master 1 1740  
PRECIO DEL ID: 34
```

Below this, another JSON object is shown:

```
email: 'cnfucio@gmail.com',  
name: 'Galo Inca',  
origin: 'Av. los chásquis 1745-1651, Ambato',  
destination: 'Auditorio ESPOCH, Ent. 1 ESPOCH,  
Riobamba',  
orderId: 34,  
limitsRates: 150,  
discount: 0,  
promotion: null,  
total: '575',  
pkgs: [  
  {  
    pkgId: '27f59716-0b76-4ed5-bd54-15db5cecfb7  
a',  
    weight: 1740,  
    price: '425',  
    content: 'Pesas 3',  
    boxName: 'CP-2'  
  }  
]
```


Validación de hipótesis

- Hipótesis nula

H_0 = El diseño y construcción de lockers automáticos para la recepción de paquetes utilizando RPA (robotic process automation) no permitirá agilizar el proceso de envío de encomiendas a WOLF

- Hipótesis válida

H_1 = El diseño y construcción de lockers automáticos para la recepción de paquetes utilizando RPA (robotic process automation) permitirá agilizar el proceso de envío de encomiendas a WOLF

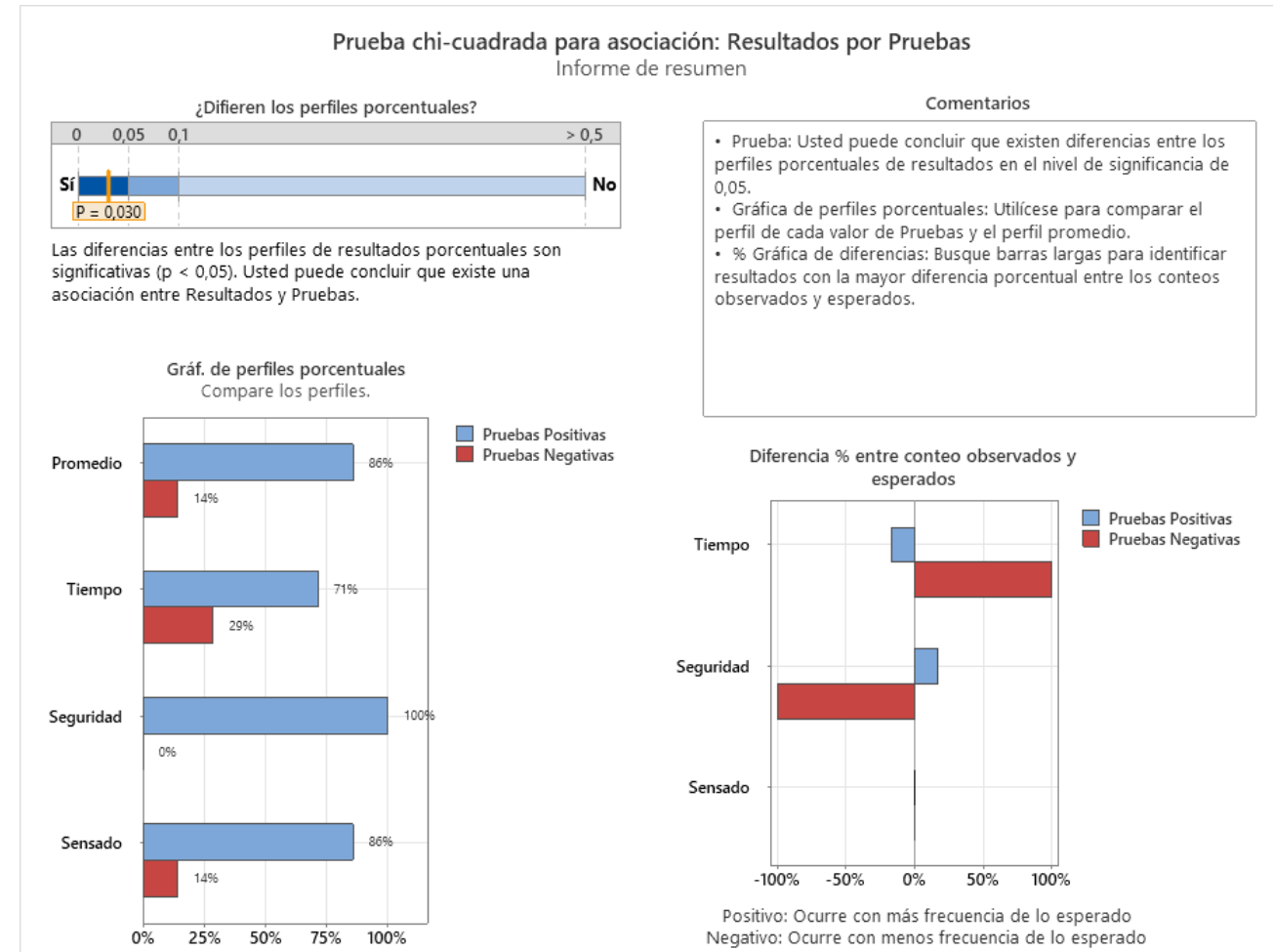
LOCKER	Tiempo	Seguridad	Sensado	Total
Pruebas Positivas	15	21	18	54
Pruebas Negativas	6	0	3	9
Total	21	21	21	63

$$p = 0.03$$

$$p > \alpha = H_0 \text{ es válida}$$

$$p \leq \alpha = H_1 \text{ es válida}$$

$$0.03 < 0.05 = H_1 \text{ es válida}$$



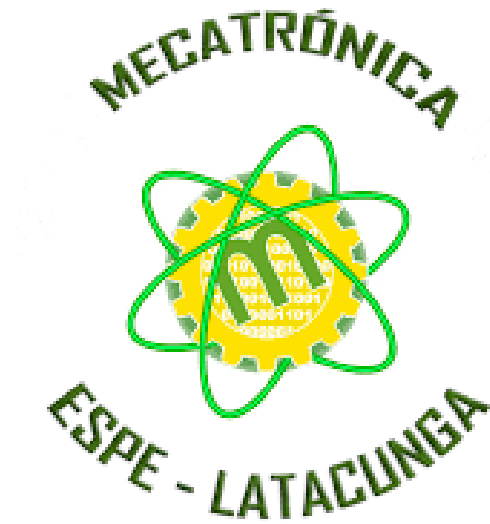
CONCLUSIONES

- La aplicación de un robot CSC en el sistema de paquetería con RPA, es capaz de reducir tiempo de servicio, además de que brinda seguridad ante errores y tiene alta escalabilidad en la pasarela de pagos.
- El sistema de sensado de peso, es independiente para cada torre y se encuentra susceptible a dar información errónea por efecto de vibraciones de la percha al momento de receptar el paquete.
- NestJS como servidor Backend (CSC) permitió construir una arquitectura flexible, proporcionando la capacidad de hacer pruebas y realizar cambios en caliente con el modo de desarrollador.
- El tiempo de uso del servicio por parte del cliente se ve reducido significativamente mientras se va familiarizando con el HMI y con el proceso.
- Se diseñaron dos placas PCB (esclavo-maestro) para controlar los lockers implementando un protocolo de comunicación MAX 485 con la finalidad de poder conectar de manera fácil y rápida a los esclavos con un ID individual de forma secuencial al sistema.
- Mediante el componente max485 y la librería de Software Serial se desarrolló la conexión tipo bus entre las placas Maestro/Esclavo. Se estableció comandos para administradores y usuarios para que la placa maestro gestione el reenvío al esclavo y este responda las peticiones

RECOMENDACIONES

- A nivel de desarrollo se recomienda emplear GitHub para realizar respaldos continuos de los archivos y generar un control de versiones con la potencia de trabajar en equipo y llevar un registro de mejoras para el proyecto.
- Al momento de implementar la comunicación serial se recomienda utilizar una arquitectura jerarquía en el envío de la trama para que no exista una colisión de datos, respetando el orden de pregunta (maestro) – respuesta (esclavo) generando un mejor flujo de datos.
- Se recomienda usar los puertos de comunicación serial propios del microcontrolador Atmega2560, que a diferencia del Atmega328p, dispone de 4 puertos físicos sin la necesidad de crear puertos “ficticios”.
- Cuando se carga el archivo “.hex” por ICSP usando el programa “Progisp.exe”, se debe tener cuidado de no mover los cables de datos durante el proceso
- Los periféricos conectados a la Raspberry pueden ser solo de tipo HID para que el sistema los reconozca sin la necesidad de instalar drivers o librerías externas.
- Si la producción de las placas va a ser masiva, se podría elaborar un solo modelo que funcione como maestro o esclavo; de esta manera se abarataría costos de producción para PCB
- Se recomienda mejorar el enlace entre la secuencia de ingreso de paquete y el HMI mediante la implementación de Socket.io para tener una lectura de los valores de estado del locker y sincronizarlos con la navegación del HMI.

GRACIAS



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA