

**División de Ciencias Básicas e Ingeniería  
Ingeniería en Computación y Telecomunicaciones**

+

## **Casillero automatizado**

---

Alumno: López Ortega Fabián Andrés, matrícula:2173072585

Director: Dr. Jacobo Sandoval Gutiérrez

Codirector: Dra. Pamela Patricia Vera Tizatl

Alumno: Martínez Flores Erick, matrícula: 2173072905

Director: Dr. Jacobo Sandoval Gutiérrez

Codirector: Dr. Celso Márquez Sánchez

27 de Mayo de 2024



**UNIVERSIDAD AUTÓNOMA METROPOLITANA  
Unidad Lerma**

# Índice

0.1. Casilleros automatizados . . . . .	6
<b>1. Objetivos</b>	<b>7</b>
1.1. Objetivo general . . . . .	7
1.2. Objetivos específicos . . . . .	7
<b>2. Marco teórico</b>	<b>7</b>
2.1. Casilleros inteligentes o los más actuales . . . . .	7
2.2. Raspberry Pi . . . . .	8
2.3. Descripción de código QR y su funcionamiento . . . . .	9
2.4. Cámara v2.1 Raspbberry Pi . . . . .	10
2.5. Funcionamiento de la chapa eléctrica . . . . .	11
2.6. Sensor Infrarrojo detector de obstaculos AD 32 ir-08h ky 032 . . . . .	12
2.7. Descripción de la herramienta XAMPP . . . . .	13
2.8. Base de datos . . . . .	13
<b>3. Costos</b>	<b>15</b>
<b>4. Desarrollo del proyecto</b>	<b>15</b>
4.1. Instalación del sistema operativo . . . . .	15
4.2. Equipo necesario . . . . .	16
4.2.1. Equipo de hardware . . . . .	16
4.2.2. Equipo de Software . . . . .	16
4.2.3. Instalación de Bibliotecas e Interfaces . . . . .	17
4.3. Diseño . . . . .	18
4.4. Base de datos . . . . .	19
4.5. Control de casilleros . . . . .	20
4.6. Interfaz de usuario . . . . .	21
4.6.1. Desarrollo del backend . . . . .	21
4.6.2. Desarrollo de Frontend . . . . .	21
4.6.3. Uso de XAMP . . . . .	22
4.7. Seguridad . . . . .	23
<b>5. Pruebas</b>	<b>24</b>
<b>6. Prototipo</b>	<b>24</b>
6.1. Prototipo versión sin sistema . . . . .	24
6.2. Prototipo versión con sistema . . . . .	24
<b>7. Experimentación</b>	<b>26</b>
7.1. Etapa inicial . . . . .	26
7.1.1. Etapa de control . . . . .	26
7.1.2. Construcción de casillero a escala . . . . .	28

7.2.	Lector de códigos QR con cámara HuskyLens . . . . .	29
7.3.	Etapa final . . . . .	31
7.3.1.	Conexión y testeo Arduino UNO con Grove mosfet . . . . .	31
7.3.2.	Conexión y testeo de Raspberry Pi con Grove mosfet . . . . .	32
7.3.3.	Sensor Infrarrojo detector de obstáculos AD 32 ir-08h ky 032 . . . . .	34
7.3.4.	Lector de códigos QR con el módulo de cámara V2.1 de Raspberry Pi . . . . .	35
<b>8.</b>	<b>Conclusiones</b>	<b>36</b>
<b>9.</b>	<b>Trabajo futuro</b>	<b>37</b>
<b>Referencias</b>		<b>37</b>

## **Resumen**

Los casilleros son muebles ubicados en espacios públicos para que los asistentes puedan dejar bajo resguardo sus pertenencias. La configuración del casillero depende de los tipos de usuarios, por ejemplo, en un tienda comercial los clientes deberán agregar una moneda para poder usarlos o en los vestidores de una Universidad cada usuario deberá llevar su candado. Ambas opciones, moneda o candando podrían no ser las mas convenientes para los usuarios. Por ello, el presente trabajo propone realizar una alternativa para el acceso de los casilleros. El trabajo consiste en investigar y estudiar el estado del arte de los tipos de casilleros recientes, sus tecnologías y proponer un sistema complementario a través del uso de un código digital. La metodología será adaptar una operación manual del casillero hacia un sistema digital de apertura, validación y registro de las operaciones del mismo. La validación del prototipo será en un negocio o en la misma Universidad.

Así mismo se presenta un estudio de los tipos de casilleros, la adquisición de uno con 4 niveles, se instaló una cerradura digital, se configuró una tarjeta digital Raspberry Pi para el accionamiento de la cerradura y se conectó una cámara para decodificar el código QR. Se montó un sensor de movimiento para la activación del lector, se realizó un servicio de una página web para poder crear la interfase con los usuarios. Se construyó la base de datos con el registro de las operaciones (almacenamiento de datos, generación de código QR y comparación), se integraron todos los módulos y se pondrá a dar servicio en la ubicación permitida.

**Palabras Clave:** Casillero, Reconocimiento por visión, Bases de datos PostgreSQL, Interfaz web.

## **Declaratoria**

Nosotros, Lopez Ortega Fabian Andrés y Martínez Flores Erick, declaramos que este es un trabajo original resultado de nuestras propias reflexiones y aprendizajes. De igual manera, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana Unidad Lerma para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Lerma.

---

Yo, Jacobo Sandoval Gutiérrez, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Lerma.

---

Yo, Pamela Patricia Vera Tizatl, adscrito a la Universidad Autónoma Metropolitana unidad Lerma, declaro que he leído en la totalidad este trabajo y apruebo su contenido técnico-científico y originalidad. Además, garantizo y avalo la viabilidad técnica, científica y económica de la propuesta de proyecto terminal, aquí planteada. Así mismo, otorgo mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, unidad Lerma, para publicar el presente documento en la Biblioteca Digital y en el Repositorio Institucional.

---

1 Los casilleros tienen sus raíces en las antiguas civilizaciones tales como la egipcia, griega y romana; desde tiempos antiguos se ha buscado resguardar o almacenar objetos valiosos. En la cultura egipcia se resguardaban los restos de sus faraones o princesas. Por otro lado los romanos y griegos resguardaban tesoros o antiguos pergaminos (véase la figura 1).

Durante la edad media, el uso más común de estos espacios era para proteger objetos religiosos o manuscritos. Hoy en día es más fácil verlos en escuelas, deportivos, espacios públicos y estaciones de transporte. Sin embargo a partir de la crisis mundial que se vivió en el 2020, el uso de estos espacios creció exponencialmente, ya que servían como intermediario entre distintas personas que realizaban algún tipo de comercio o transacción, además con la integración de la industria 4.0 surgieron los casilleros inteligentes, ganando gran popularidad y aceptación entre la gente por la facilidad de uso y adaptabilidad a cualquier tipo de entorno.



Figura 1: Cofre romano (Muñoz, S.F.).

## 0.1. Casilleros automatizados

Actualmente existen diversos usos para los lockers más modernos, este modelo tuvo un crecimiento durante la crisis mundial que se vivió a finales 2019, ya que surgió la tendencia por el cero contacto. A continuación se muestra un listado de algunos beneficios de un sistema de este tipo.

- Control de acceso: Los lockers automatizados utilizan métodos de autenticación, como códigos QR, tarjetas RFID o reconocimiento facial, para garantizar que solo las personas autorizadas puedan acceder a los compartimentos.
- Seguridad: Estos lockers suelen contar con sistemas de seguridad avanzados, como cerraduras electrónicas y sistemas de monitoreo en tiempo real, para proteger los objetos almacenados contra robos y daños.
- Automatización: Los lockers pueden funcionar de manera autónoma y realizar tareas como la apertura y el cierre automatizados, el seguimiento del inventario y la generación de notificaciones cuando se reciben o entregan objetos.
- Entrega y recolección eficiente: Los casilleros automatizados son utilizados en diferentes industrias, como el comercio electrónico y la logística, para facilitar la entrega y la recolección de productos de manera eficiente. Los usuarios pueden recoger sus pedidos en ubicaciones convenientes y en horarios flexibles.
- Experiencia del cliente: Los lockers automatizados ofrecen comodidad y flexibilidad a los usuarios, ya que pueden acceder a sus objetos en cualquier momento, evitando la necesidad de esperar en casa para recibir paquetes o depender de horarios de entrega específicos.
- Monitoreo y análisis: Los sistemas de gestión de lockers recopilan datos sobre el uso, la ocupación y otros parámetros relevantes. Esto permite obtener información útil para mejorar la eficiencia operativa, la planificación de inventario y la toma de decisiones estratégicas.

## **1. Objetivos**

### **1.1. Objetivo general**

Diseñar, implementar y poner en funcionamiento un sistema de casilleros automatizados para el alumnado de la UAM-Lerma utilizando herramientas que permitan la asignación y apertura de casilleros mediante códigos QR.

### **1.2. Objetivos específicos**

A continuación se enlistan los objetivos particulares a alcanzar.

- Realizar una investigación y estudio del estado del arte
- Adaptar un casillero con apertura manual por un actuador digital.
- Diseñar e implementar una base de datos para el control de datos.
- Realizar un interfaz para que el usuario pueda solicitar hacer uso del espacio.
- Realizar una integración entre los módulos de software como de hardware.

## **2. Marco teórico**

### **2.1. Casilleros inteligentes o los más actuales**

En la actualidad, Amazon es el proveedor que mantiene la opción mas popular de casilleros automatizados. Amazon Hub Locker permite a los clientes recoger sus paquetes en espacios seguros en lugar de recibirllos en algún domicilio (véase la figura 2).

El funcionamiento es el siguiente:

En la compra de un artículo se puede elegir la ubicación de algún casillero, una vez que el paquete sea entregado, se recibirá un correo con las instrucciones donde se incluye la dirección y los horarios del punto de entrega. Después de que el paquete se encuentra en el casillero por tres días, si no es recogido, se devuelve y se hace un reembolso.

Para abrir el locker correspondiente es necesario contar con un código de barras o con un código de seguridad para poder teclearlo en la pantalla y así poder sacar el paquete.



Figura 2: Recolección y entrega de paquetes (Amazon, 2020).

## 2.2. Raspberry Pi

Raspberry Pi es un ordenador totalmente funcional en un formato pequeño y de bajo costo, también se le conoce como ordenador de una sola placa del tamaño de un teléfono o una tarjeta de crédito. Para el desarrollo de este proyecto utilizaremos algunas características de este dispositivo, en la figura 3, se muestra el equipo (Upton y Halcree, 2016).



Figura 3: Raspberry pi 3 model B+ (Pi, 2023).

- Conectividad Wi-Fi y Ethernet: Los puertos Ethernet y soporte integrado para Wi-Fi , lo que permite la conexión a una red local o internet para la transmisión de datos.
- GPIO (Entrada/Salida): Los pines GPIO permiten la conexión de sensores, actuadores y otros dispositivos electrónicos. A continuación en la figura 4 se observa el mapeo de los

pines que tiene el dispositivo.

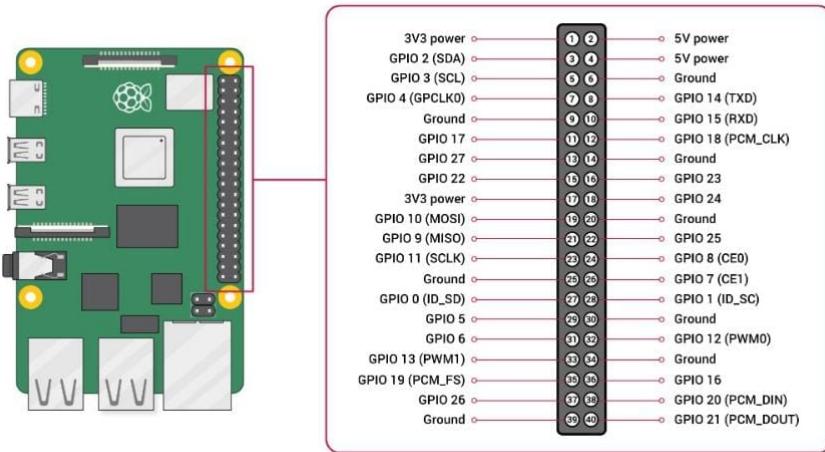


Figura 4: Pines de Raspberry Pi 3 B+ (*Rasp2, s.f.*).

- Puerto de cámara: El puerto de cámara de la Raspberry Pi es un conector que te permite conectar una cámara compatible directamente a la placa de la Raspberry Pi.
- Sistema operativo y software: Existen diferentes sistemas operativos compatibles para cada tipo de Raspberry, tales como: Raspberry Pi OS, Ubuntu, Fedora , entre otros.
- Acceso remoto: Se puede configurar la Raspberry Pi para permitir el acceso de forma remota a través de SSH o una interfaz web.
- Capacidad de personalización: La Raspberry Pi es altamente personalizable y adaptable a cualquier tipo de necesidad , a través de la adición de hardware y software.

### 2.3. Descripción de código QR y su funcionamiento

Los códigos QR, son códigos bidimensionales que se utilizan para almacenar información codificada a través de un fichero o archivo de texto, también conocidos como códigos de respuesta rápida, de allí su abreviatura "Quick Response (QR)". Un código QR consta de tres elementos principales (Mishra, Kumar, Chowdhury, y Bhondekar, s.f.).

- Zona de detección: Es el borde cuadrado que rodea el código y ayuda a los lectores a detectar su posición y tamaño.
- Módulos de datos o información variable: Son los cuadros negros y blancos que codifican la información
- Patrón de alineación: Permite al lector determinar la escala y la orientación de código.

Funcionamiento:

En primer lugar, se selecciona la información que se desea codificar en formato QR, esto puede

incluir texto, números, enlaces URL, entre otras cosas. Es necesario el uso de software para lograr esta transferencia de datos. En la figura 5 se muestran las características de un código QR, así como la correcta orientación para una lectura más eficaz.

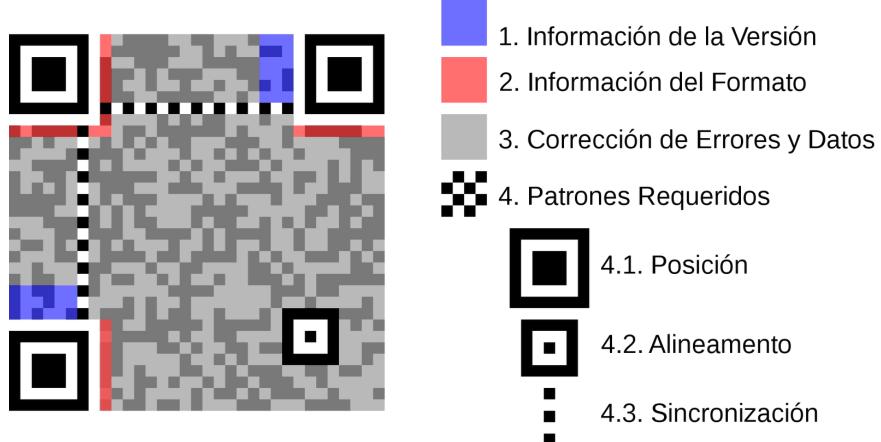


Figura 5: Estructura de código QR (Mishra y cols., s.f.).

## 2.4. Cámara v2.1 Raspberry Pi

Para la lectura del código QR se utilizará el módulo de cámara Raspberry Pi V2 IMX219 de 8MP es una cámara de alta resolución con la capacidad de grabar video en 1080p y capturar imágenes. Puede ser conectada directamente a todos los modelos de Raspberry Pi 3 y 4, y su acceso se realiza a través de las interfaces MMAL y V4L. Es totalmente compatible con el sistema operativo Raspbian.

Este módulo de cámara (Véase la figura 6), diseñado para la Raspberry Pi, es versátil y se puede utilizar tanto para grabar video como para aplicaciones de seguridad y detección de movimientos. Para su conexión, simplemente se debe conectar el cable plano suministrado al puerto CSI (Interfaz Serie de Cámara) de la Raspberry Pi (ELECTRONICS, 2023).

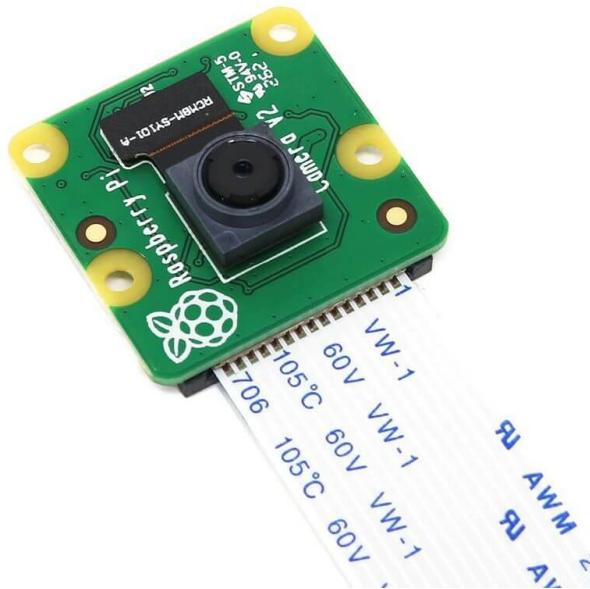


Figura 6: Cámara Raspberry Pi (ELECTRONICS, 2023).

Este módulo cuenta con las siguientes características:

- Sensor de imagen: Sony IMX219
- Resolución de 8 megapixeles
- Captura de imagen hasta 3280 x 2464 pixeles
- Captura de vídeo 1080p30, 720p60 y resoluciones de 640x480p90
- Área de imagen del sensor: 3,68 x 2,76 mm (4,6 mm en diagonal)
- Tamaño de pixel: 1.4  $\mu$ m x 1,4  $\mu$ m pixeles con la tecnología OmniBSI
- Tamaño óptico: 1/4
- Apertura: 2.9
- Ángulo de Visión: 62.2°
- Distancia focal: 3.04mm
- Dimensiones: 25mm x 23mm x 9mm
- Peso: 27 gramos

## 2.5. Funcionamiento de la chapa eléctrica

La apertura del locker se realiza por medio de la Chapa eléctrica XG-01 12V, que es un interruptor automático de modo seguro, esto quiere decir que a la hora de ser energizada se enciende y cuando se apaga se bloquea. De manera bruta se utiliza para realizar la apertura de la puerta del casillero, sin embargo para usos más específicos se integrará un sistema de control que

regule la entrada de energía para activarla o apagarla después de recibir la señal de confirmación por medio de una interfaz. En la figura 7, se muestra la estructura de la cerradura.



Figura 7: Chapa eléctrica XG-01 12V (Club, Noviembre,2021).

## 2.6. Sensor Infrarrojo detector de obstáculos AD 32 ir-08h ky 032

El Sensor de Obstáculos KY-032, también conocido como detector de línea infrarrojo, es un aparato diseñado para identificar la presencia de un objeto mediante la reflexión de luz. Se emplea luz infrarroja (IR) con el propósito de que no sea perceptible por los seres humanos. Su funcionamiento es eficaz a distancias cercanas, generalmente entre 2 y 40 cm. En la figura 8 se muestra el dispositivo así como sus pines.

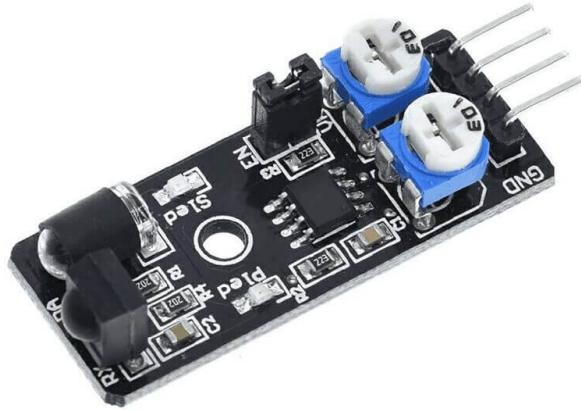


Figura 8: Sensor Ky-032 (Electronics, S.F.).

A continuación se enlistan algunas características del equipo:

- Voltaje de funcionamiento 3.3 a 5 [V].
- Distancia de detección: 2 a 40 cm.
- Corriente de trabajo 20 mA.
- Sensor IR: TCRT5000L.
- Dimensiones 41mm x 18mm x 13mm.
- Ángulo efectivo: 35°.
- Peso 3 [g].

## 2.7. Descripción de la herramienta XAMPP

Este tipo de herramienta nos permite instalar un proyecto local de nuestro PC y realizar distintas pruebas para el aprendizaje de lenguajes tales como PHP o MySQL sin requerir algún tipo de hosting o conexión a internet, ya que desde el mismo ordenador se pueden crear y visualizar en el navegador las páginas del proyecto en el que se este trabajando (HTML) (Bou, 2019).

## 2.8. Base de datos

Utilizaremos la base de datos PostgreSQL principalmente porque nos permite relacionarlo con otras herramientas que estamos utilizando. También porque es de código abierto y es una base de datos relacional, lo que beneficia a este proyecto debido a que permite gestionar datos

estructurados, por lo tanto, es adecuado para el almacenamiento de nombres, identificadores, números de pedido, casillero, etc.

Ahora bien en términos de implementación, primeramente se resalta que PostgreSQL utiliza SQL, un lenguaje manejable y cómodo para la realización de consultas y gestión de datos. Además, este software tiene la capacidad de manejar grandes volúmenes de datos, lo que es esencial para mantener un sistema de entregas rápido y eficiente, incluso ofrece opciones de escalabilidad, esto permitirá adaptar el sistema para manejar el crecimiento de la cantidad de usuarios y de datos. Por último, otras características que tiene PostgreSQL que serán de ayuda para este proyecto son:

- Tiene herramientas como pgAdmin, que ofrece una interfaz gráfica intuitiva que facilitará la administración de datos.
- Cuenta con una documentación completa y detallada que contiene guías y ejemplos, por otro lado, su instalación es sencilla y su configuración es amigable para adaptarlo para satisfacer las necesidades de este proyecto.
- Es utilizado en la industria (lo utilizan empresas como Uber o Apple para servicios internos), lo que permitirá incluir personas con habilidades en PostgreSQL para colaborar en el proyecto.
- Se garantiza la consistencia e integración de datos mediante el soporte de transacciones ACID (Atomicity, Consistency, Isolation and Durability), lo que es fundamental para el sistema de gestión de entregas confiable.
- Cuenta con PostGIS, que agrega herramientas de geolocalización para que en acciones posteriores se pueda integrar la gestión de ubicaciones geográficas de los puntos de entrega.
- Es gratuito, lo que reduce el costo total de este proyecto.

En conclusión, PostgreSQL utiliza el lenguaje SQL para consultas y emplea modelo relacional para organizar los datos, lo que quiere decir que utiliza tablas, filas, columnas, dominios, llaves primarias y relaciones entre tablas.

### **Modelo ACID.**

**Atomicidad.** Cada transacción se considera como una unidad atómica, es decir, se realiza o no se realiza, sin estados intermedios. Si la transacción falla, se realiza un “rollback” para que todas las operaciones sean revertidas.

**Consistencia.** Solo se aplican transacciones que lleven la base de datos de un estado válido a otro estado válido. Esto para asegurar que no se introduzcan datos inconsistentes.

**Aislamiento.** Cada transacción se aplica de manera independiente y aislada de otras transacciones.

**Durabilidad.** Cada que una transacción se ha confirmado, los efectos en la base de datos son permanentes, y persisten incluso en una falla del sistema, en otras palabras, la información no puede perderse.

### **3. Costos**

Para el desarrollo de este proyecto es necesario considerar los gastos en la adquisición del equipo, a continuación en la tabla1, se muestra un costo aproximado de los materiales, el precio puede variar de acuerdo al proveedor.

Tabla 1: Hardware necesario.

Dispositivo	Costo aproximado
Locker de 4 casilleros	\$ 3490 .90
Raspberry Pi 3 B+	\$ 1044.00
Cámara V2.1	\$ 600.00
Grove Mosfet	\$ 75.00
Disipadores Raspberry Pi	\$ 45.0
Chapa eléctrica XG-01	\$ 88.00
Sensor de proximidad	\$ 75.00
Bisagra	\$ 10.00
Tarjeta micro SD 64Gb	\$ 88.00
Total	\$ 5515.90

### **4. Desarrollo del proyecto**

#### **4.1. Instalación del sistema operativo**

Para iniciar a trabajar con la tarjeta de desarrollo Raspberry Pi, es necesario instalar el sistema operativo, en este caso se trabaja con ayuda del instalador para windows y se instalo una imagen iso con el sistema Raspbian. A continuación se muestra la metodología a seguir (Upton y Halcree, 2016).

##### **Descarga de instalador y sistema operativo**

1. Para realizar la instalación es necesario tener todos los paquetes. el siguiente enlace direcciona a la descarga del instalador oficial de Raspberry Pi, de acuerdo a las características de su equipo.

<https://www.raspberrypi.com/software/>.

2. Realice la descarga correspondiente a los parámetros de su equipo.
3. Espere que descargue, abra el archivo y seleccione instalar. Al termino de la instalación aparecerá la venta Teniendo el instalador, debemos descargar el archivo .iso (Sistema operativo), en formato .zip o directamente en imagen.iso, a continuación, el enlace de descarga: <https://www.raspberrypi.com/software/operating-systems/>.

##### **Proceso de instalación**

4. Inserte la memoria microSD a su equipo con el adaptador.

5. Una vez realizado este paso, diríjase a la sección de Opearating System en el instalador Raspberry Pi.
6. Seleccione el archivo .iso y cárguelo en el instalador.
7. Ahora diríjase a la sección de: "Storage", y seleccione el almacenamiento en la memoria externa microSD.
8. Finalmente, seleccione la opción: "WRITE". Espere a que termine la escritura del sistema operativo en la tarjeta

#### **Comprobación del sistema operativo**

9. Una vez que retires la tarjeta del lector, inserta la MicroSD en el compartimiento correspondiente en la Raspberry.
10. Se realiza la conexión con los cables HDMI al monitor, y se enciende el equipo.
11. Al iniciar el equipo te pedirá que configures el idioma y la conexión a internet. Después verificará actualizaciones de ser necesario.

## **4.2. Equipo necesario**

### **4.2.1. Equipo de hardware**

Para iniciar un proyecto de este tipo es necesario contar con el siguiente material de software y de hardware, en la tabla2 se muestran los componentes eléctricos que se utilizaron durante el desarrollo.

Tabla 2: Hardware necesario.

Material	
Raspberry Pi 3 B+	Fuente de alimentación
Teclado con puerto usb	Tarjeta micro SD
Mouse	Módulo de cámara V2 IMX219 Raspberry Pi
Cable micro HDMI	Disipadores para Raspberry Pi
Grove-Mosfet	Bisagra mini
Locker tradicional de 4 casilleros	Chapa eléctrica XG-01 12V
Equipo PC	4 tablas de 10x10cm
Sensor de movimiento	Ventilador para Raspberry Pi

### **4.2.2. Equipo de Software**

En la tabla3 se muestran los editores de texto para la realización y compilación durante las pruebas y el montado del equipo, algunos software pueden cambiar de acuerdo al los requerimientos del equipo de computo o gustos del programador.

Tabla 3: Software necesario.

Editor de textos	
Thonny	Visual Studio Code

Por ultimo, en la tabla4 se muestran las bibliotecas e interfaces que se usaron durante el proyecto.

Tabla 4: Bibliotecas e Interfaces necesarias.

Nombre	
Psycopg2	OpenCV (cv2)
Numpy	RPi.GPIO
PHPMailer	ARGON2
PHPQRCode	

#### 4.2.3. Instalación de Bibliotecas e Interfaces

A continuación se muestra una breve descripción de la funcionalidad de cada herramienta así como el proceso de instalación por medio de comandos directamente desde una terminal de Raspberry Pi.

- cv2: Open Source Computer es una biblioteca de visión por computadora es decir que contiene herramientas para el procesamiento de imágenes, detección y seguimiento, reconocimiento facial, entre otras.

Proceso de instalación:

1. Instalación de dependencias.

```
sudo apt-get install libopencv-dev python3-opencv
```

2. Instalación de herramientas y bibliotecas adicionales.

```
sudo apt-get install cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev
```

- Numpy: Este tipo de biblioteca proporciona un soporte para trabajar con arreglos y matrices multidimensionales, a demás de añadir una colección de funciones matemáticas de alto nivel.

1. Instalación de dependencias.

```
sudo apt-get install libopencv-dev python3-opencv
```

2. Instalación de herramientas y bibliotecas adicionales.

```
sudo apt-get install cmake git libgtk2.0-dev pkg-config libavcodec-dev libavformat-dev libswscale-dev
```

- Psycopg2: Este tipo de módulo facilita la conexión y la intersección con la base de datos "PostgresSQL", sirve en conjunto con el lenguaje de programación Python.

1. Instalación de paquetes.

```
pip install psycopg2
```

2. Conexión a la base de datos (Parámetros de conexión).

```
dbname = "nombre_de_la_base_de_datos"
user = "usuario"
password = "contraseña"
host = "localhost"
```

- RPi.GPIO: Esta biblioteca de python proporciona una interfaz para interactuar con los pines de propósito general(GPIO). Los pines GPIO son pines que pueden ser configurados para realizar diversas funciones, como entrada o salida digital, comunicación serial, control de PWM (modulación de ancho de pulso), entre otras.

1. Instalación de paquete de desarrollo de Python.

```
sudo apt-get install python3-dev
```

2. Instalación de bibliotecas RPi.GPIO.

```
sudo pip3 install RPi.GPIO
```

### 4.3. Diseño

En la figura 9 se muestra el algoritmo del sistema, a grandes rasgos el usuario podrá solicitar el uso del espacio, llenando un formulario el cual podrá acceder desde cualquier navegador, una vez teniendo su registro, se almacena la información en la base de datos y se realiza un procedimiento simultaneo, al generarse un código QR, único y de un solo usuario, ya que cada código estará relacionado a la matrícula del alumno. Este código es almacenado en la base de datos y enviado al correo ingresado. Una vez que el usuario tenga su código QR, puede acceder al locker el cual contara con un sensor de proximidad, que activará el Lector de códigos (Cámara V2.1), este módulo se encarga de escanear y mandar la información a la base de datos para hacer una comparación y saber si el código es correcto, en caso de coincidir, la cerradura se abrirá. Si el usuario ya no quisiera hacer uso del espacio, puede abrir el casillero cuantas veces quiera con su código de acceso.

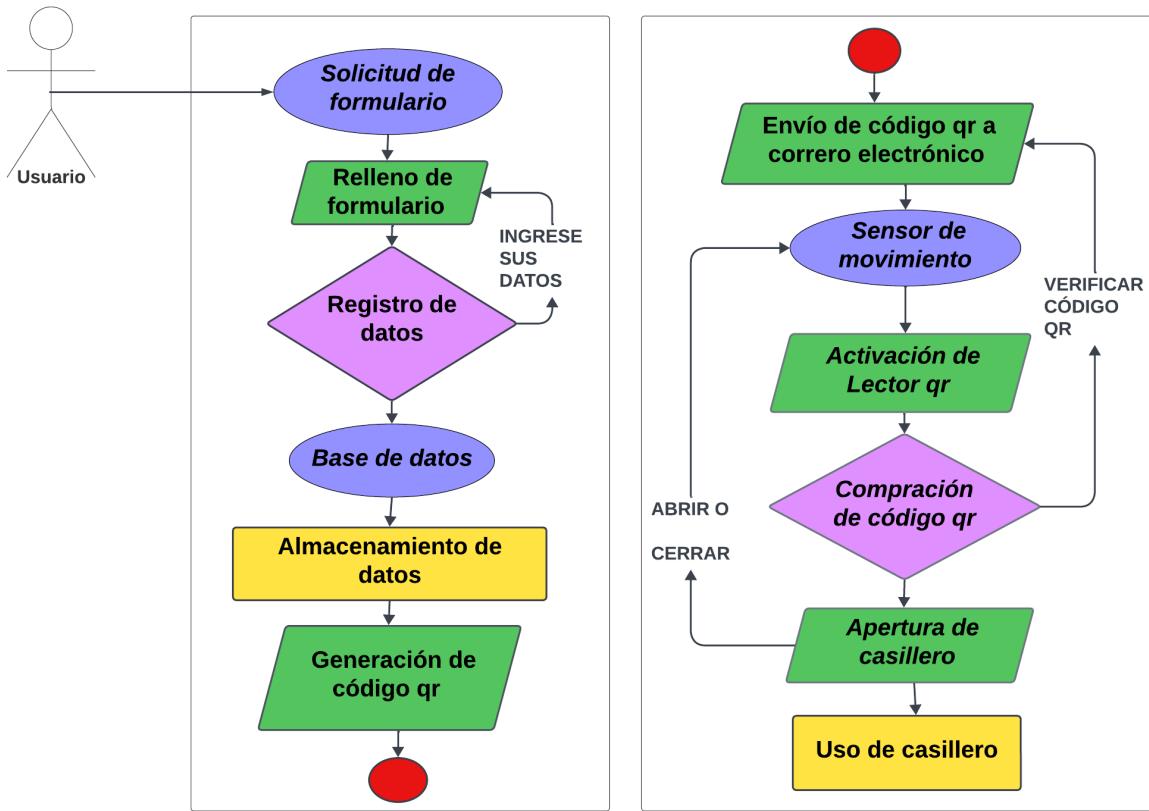


Figura 9: Algoritmo del sistema.

#### 4.4. Base de datos

- Como se había mencionado en el apartado 3 (Marco teórico) sección 3.8 (Base de Datos), la base de datos que se utiliza en este proyecto es PostgreSQL porque es una base de datos relacional confiable y durable en el almacenamiento de datos, y por consecuencia es necesario la utilización de tablas.

Con base en el funcionamiento de la interfaz de registro, la asignación de códigos QR a paquetes y la disponibilidad de lockers, tenemos como resultado el siguiente diagrama UML que describe las clases, atributos y operaciones de la base de datos.

En este diagrama mostramos 3 clases:

- Usuario.
- Locker.
- Paquete.

En cada clase están plasmados los atributos con su tipo de dato y también su llave primaria, además de las operaciones que se pueden hacer en cada una. Sumado a esto, también se

muestran dos asociaciones y se definen cuantos objetos de una clase están relacionados con los objetos de otra clase, en ambos casos la relación es de uno a múltiples (véase la figura 10).

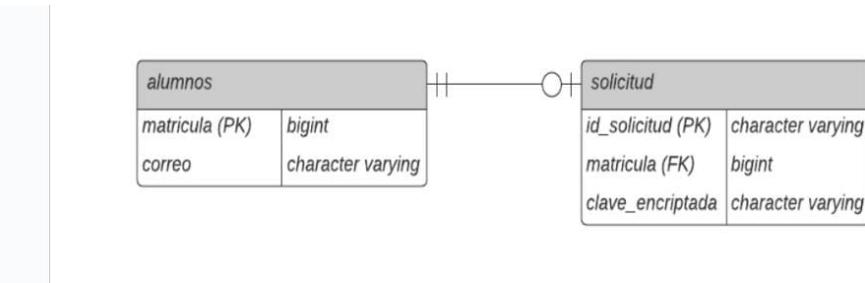


Figura 10: Diagrama entidad relación

#### 4.5. Control de casilleros

El proceso de control de casilleros, la base de datos juega un papel muy importante porque cada casillero contará con un identificador único asociado a un usuario específico, esto se logra almacenando la información del código QR enviado al usuario junto con el identificador del casillero disponible. Cuando el usuario escanea su código QR, el sistema busca en la base de datos el casillero y si lo encuentra asociado pasa a un estado disponible, entonces el cliente puede abrir el casillero y recoger su paquete.

Para garantizar que existan la menor cantidad de errores en el sistema, decidimos utilizar 3 métodos.

El primero es hacer una revisión exhaustiva en cada componente del sistema, desde la generación del sistema, el almacenamiento de información, hasta la apertura del casillero. El segundo método es hacer este sistema muy intuitivo para el usuario y previniendo errores o excepciones en la utilización de este modelo. Por último, destinaremos gran parte del tiempo de la realización de este sistema para tener una validación y verificación de la información que se almacene en la base de datos.

Este último método abarca diferentes formas de validación o verificación de datos. Restricciones que podemos usar en la base de datos:

**Utilizacion de llaves primarias y llaves foraneas** Para garantizar consistencia entre las relaciones de las tablas.

**Restricciones NOT NULL** Para asegurar que algunos campos no queden vacíos.

**Restricciones CHECK** Para validar valores permitidos en un campo.

**Restricciones UNIQUE** Para garantizar que algunos campos tengan valores únicos.

## **4.6. Interfaz de usuario**

### **4.6.1. Desarrollo del backend**

Para el desarrollo invisible de este proyecto contamos con cinco componentes indispensables para el funcionamiento de este sistema. El primero es el proceso de registro de usuarios, aquí los usuarios por medio de una interfaz se registran en el sistema proporcionando información para posteriormente recibir un código QR asociado a su perfil. El siguiente proceso consta de la generación de códigos QR, en esta sección se utilizará la información ingresada por el usuario durante el registro para hacer que cada código QR sea único y que se asocie al paquete y al perfil específico.

Ahora en otro punto del sistema que no está asociado con la generación de códigos QR y a la interfaz de registro podemos encontrar la lectura de códigos QR, en este módulo se decodificará la información que contenga el código mediante el módulo de cámara v2.1, para después hacer una comparación entre la información que hay en la base de datos con la información del código QR y si coinciden, se dará acceso al locker correspondiente. Otro proceso fundamental para este modelo es la base de datos que conecta a todos los puntos del sistema, debido a que además de contener la información de usuarios, contiene la información de los códigos QR que han sido almacenados con anterioridad para después dar acceso del locker al usuario, incluso, si es necesario, de esta base de datos podemos obtener un historial de uso.

### **4.6.2. Desarrollo de Frontend**

Durante esta etapa de desarrollo utilizamos la herramienta "Ninjamook", para realizar un mapeo de una posible visualización en el interfaz de usuario. En la figura 11, se muestra la pagina inicial en donde el usuario ingresara su correo y su matrícula, en la figura 12 se observa el salto de pagina a en donde el registro a sido exitoso y el proceso a iniciado.

Figura 11: Vista preliminar del interfaz.

Figura 12: Solicitud exitosa.

#### 4.6.3. Uso de XAMP

El funcionamiento de este módulo es que a través del formulario, los usuarios puedan ingresar su nombre y un correo al cual se les envíe el código QR, una vez que se envíe la información, se almacenara en la base de datos. Para poder realizar este proceso, es indispensable contar con un nodo que permita interactuar con la base de datos y realizar operaciones de almacenamiento y recuperación de datos, este nodo es Database Node.

Por lo tanto, primero tenemos que crear un formulario en HTML para el registro de los usuarios que capture nombre y correo electrónico para que se envíen primero a Node-Red cuando el usuario de clic en el botón “registrarse”. Después configuramos Node-Red agregando un nodo HTTP input para recibir solicitudes POST y que el flujo de información sea adecuado para transmisión el recibimiento de datos.

Para almacenar los datos correctamente necesitamos procesarlos antes de almacenarlos, esto se hace agregando un nodo llamado: "Function". Por último, podemos enviar una respuesta al usuario para confirmar que su registro ha concluido con éxito.

#### **4.7. Seguridad**

La seguridad es fundamental en cualquier sistema que involucre manejo de información y acceso a recursos, es por eso por lo que en primera instancia nos enfocamos en garantizar que la producción de códigos QR sea única para cada paquete-usuario, y que no se puedan replicar o falsificar.

Para la generación de códigos QR seguros y únicos, nos enfocamos primeramente en que a simple vista no se pueda detectar el locker, el usuario o el paquete correspondiente con solo leer el número del código QR, para esto elegimos utilizar Argon2, que es un algoritmo de derivación de claves y función de hash de contraseña diseñado por expertos en criptografía.

Argon2 fue seleccionado como ganador del concurso "Password Hashing Competition" en 2015 debido a su alta resistencia contra ataques de fuerza bruta y ataque de diccionario, por lo tanto, hace que este método sea un algoritmo con alto nivel de seguridad.

Tomamos en cuenta que Argon2, además de resistir ataques de fuerza bruta y de diccionario, permite configurar el tiempo, la memoria y los parámetros paralelos para adaptarse a diferentes entornos, de esta manera puede existir un equilibrio entre seguridad y rendimiento, incluso utiliza operaciones de memoria intensivas para complicar posibles ataques que intentan parallelizar el proceso.

##### **Ataques de paralelización.**

Son estrategias empleadas por los atacantes para mejorar la eficiencia y velocidad de los ataques al realizar múltiples operaciones simultáneamente utilizando recursos paralelos (múltiples tareas al mismo tiempo utilizando varios recursos como varios núcleos de CPU o computadoras en una red), esto se hace para que los sistemas de seguridad para vulnerar contraseñas, cifrados y otras medidas de protección.

##### **Hash.**

Función matemática que toma una entrada (o mensaje) y produce una cadena de caracteres alfanuméricos de longitud fija, generalmente en formato hexadecimal. Esta cadena de caracteres es única para cada conjunto de entrada, lo que significa que un cambio incluso mínimo en la entrada resultara en un hash completamente diferente.

## 5. Pruebas

## 6. Prototipo

### 6.1. Prototipo versión sin sistema

Para el desarrollo de este proyecto se utilizará un locker tradicional de cuatro casilleros, véase la figura 13. En la tabla 5 se muestran las características del casillero.



Figura 13: Locker tradicional de cuatro casilleros (Gabinetes, 2023).

Tabla 5: Descripción del Casillero.

No. de Lockers	Dimensiones interior: Ancho x Profundidad X Altura (35 X 45 X 55 ) CM	Color
4	Dimensiones exterior: Ancho x Profundidad X Altura (37 X 45 X 180 ) CM	Gris

### 6.2. Prototipo versión con sistema

En la figura 14, se puede apreciar el montaje del sistema sobre una base de aluminio. El componente más prominente es la placa de desarrollo Raspberry Pi 3 B+, ubicada frontalmente junto al lector de códigos QR. En la parte superior, se encuentran montados el sensor de proximidad

y el mosfet, ambos conectados a sus respectivos pines en la placa de trabajo. El módulo "Grove-mosfet" cuenta con dos terminales: uno para la fuente de alimentación y otro para el dispositivo a controlar. Durante el desarrollo, prescindimos de la fuente de alimentación, ya que este dispositivo puede recibir energía de un microcontrolador. Dentro del casillero, se aloja la chapa eléctrica, con el seguro fijado en la puerta del locker y el cuerpo de la chapa alineado justo enfrente para un cierre perfecto.



Figura 14: Sistema implementado.

En la figura 15 se observa la vista frontal del proyecto.



Figura 15: Sistema final.

## 7. Experimentación

### 7.1. Etapa inicial

Toda la información, códigos y vídeos de pruebas pueden ser consultadas en el siguiente repositorio: <https://github.com/DeliverEasyCompany/DeliverEasyCompany.github.io.git>.

#### 7.1.1. Etapa de control

Una de las primeras pruebas fue la etapa de control, fue realizar el encendido y apagado de un selenoide, en el cual se integraron los siguientes componentes:

- Mosfet irf44n.
- Transistor bc547.
- Resistencias: (1k, 10k y 100k) [Ohm].
- Selenoide.
- Fuente de poder.

- Placa de desarrollo Arduino.
- Arduino IDE.

La figura 16 muestra el circuito que se utilizó para esta prueba.

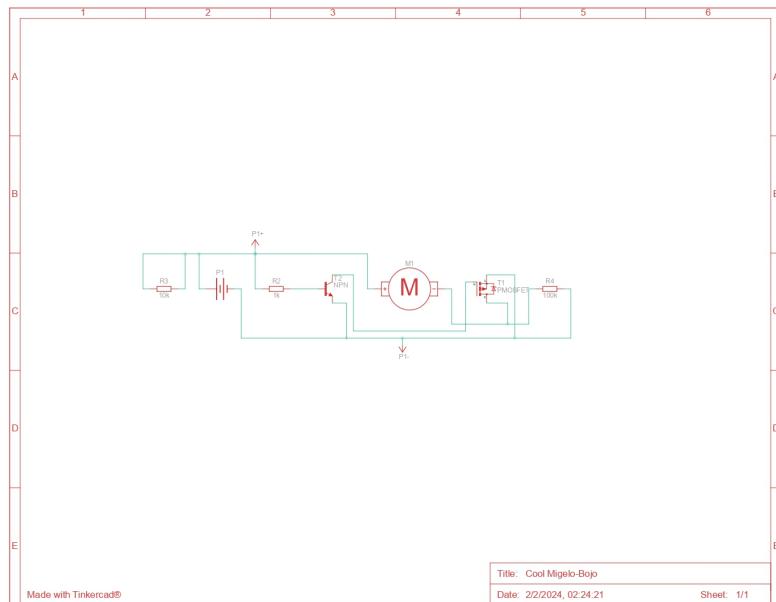


Figura 16: Etapa de potencia.

Durante el experimento se sustituyó la cerradura por el selonoide, se utilizó la placa de trabajo Arduino y se logró tener un control de las activaciones del motor; en la figura 17 se muestra el proceso de la prueba. Más adelante se optó por simplificar el circuito por medio del módulo "Grove mosfet", como se verá en la siguiente prueba.

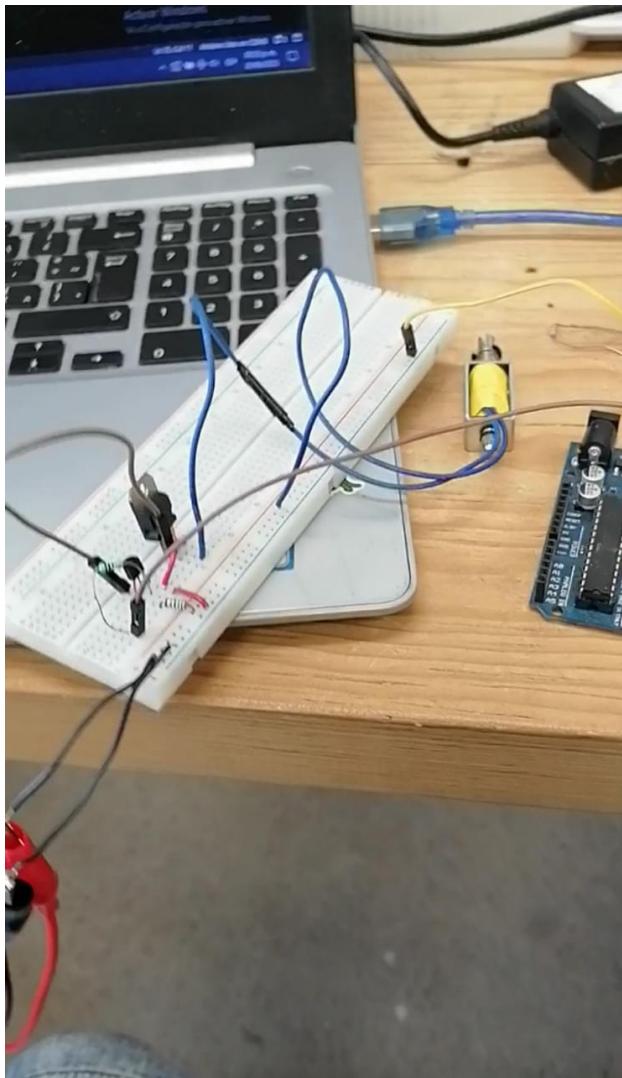


Figura 17: Preuba de Selenoide

### 7.1.2. Construcción de casillero a escala

A continuación se pude observar en la figura 18, el armado del casillero de madera.fue necesario hacer uso de la bisagra, los pedazos de madera y la chapa eléctrica. Al ser energizado el sistema, la chapa bota inmediatamente el seguro, logrando tener una respuesta en bruto, es decir sin algún tipo de control.

Material necesario: Bisagra, cuatro pedazos de madera de 10x10 cm, chapa electrónica, pegamento y Fuente.

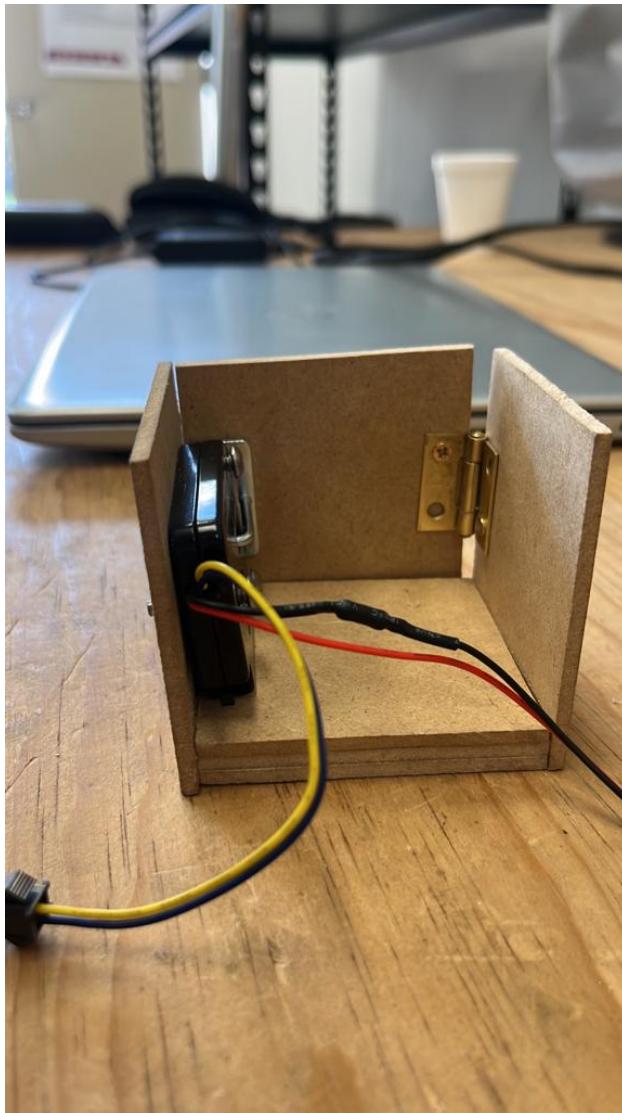


Figura 18: Armado y montura de casillero de madera.

## 7.2. Lector de códigos QR con cámara HuskyLens

La lectura de códigos QR se realiza por medio la cámara HuskyLens que utiliza inteligencia artificial para el procesamiento de imágenes y el reconocimiento de patrones. Este dispositivo utiliza los algoritmos de detección de códigos QR. que consiste en buscar patrones únicos de este código QR, haciendo énfasis en los cuadrados de las esquinas y en los patrones de posicionamiento (Véase la figura 19). Más adelante se optó por cambiar de cámara, ya que existía poca información acerca del dispositivo y por cuestión de costos.

Se realizaron las siguientes pruebas:

1. Detección de rostro.
2. Decodificación de código QR.



Figura 19: Prueba de cámara.

A continuación se muestra el proceso de configuración de la cámara HuskyLens: Con base en las recomendaciones del fabricante (DFRobot) para Windows 10:

1. Descargamos el instalador HuskyLens.
2. Descargamos el controlador (driver) USB a UART y lo instalamos.
3. Una vez que esté instalado, podemos usar el puerto USB de la cámara para conectarla con nuestra computadora y seleccionamos el puerto “Silicon LabsCP210x”.
4. Continuamos descargando el firmwre mas reciente para consultar todas las versiones.
5. Ejecutamos el instalador de HuskyLens, después de cierto tiempo aparecerá la interfaz donde podemos seleccionar el archivo para cargar el firmwre.
6. Le damos clic en: “subir”, y tardara aproximadamente 5 minutos porque el archivo es grande. Una vez que se haya completado este proceso la interfaz se iluminará y estará lista nuestra cámara para empezar a configurar sus funciones.

## Configuración

1. El dispositivo HuskyLens cuenta con un botón que se mueve a la izquierda y a la derecha, con este botón vamos desplazando hasta que en la parte superior de la pantalla aparezca la configuración general y presionamos para ingresar.
2. Ya que estamos dentro de la configuración general habrá parámetros diferentes, en nuestro caso nos movemos hasta encontrar la opción: "Tag Recognition", y presionamos para permanecer en esta configuración de lectura de códigos QR.

### 7.3. Etapa final

#### Integración del módulo Grove-Mosfet

Para esa prueba se simplifico el circuito añadiendo el módulo "Grove-mosfet", aunado a que se utilizo la cerradura XG-01. A continuación se describen brevemente los procesos de testeo, con las placas de trabajo Arduino y Raspberry Pi.

##### 7.3.1. Conexión y testeo Arduino UNO con Grove mosfet

Material necesario: Arduino UNO, Grove- MOSFET y Fuente de alimentación.

En esta fase de pruebas se integro un sistema control, regulado digitalmente por el código editado en el interfaz Arduino IDE, el hardware eléctrico "grove-mosfet", y la alimentación directa de la fuente poder, a partir de los 5 volts el sistema se energizaba y lograba abrir la chapa eléctrica. Cada cierto lapso de tiempo existía una recarga, en la cual se podía volver a montar la cerradura y posteriormente abrir. A continuación se muestran las conexiones requeridas para el desarrollo y en la figura quince se puede observar el armado simplificado del circuito (seeed Studio, 2023).

- En la tabla6 se muestran las conexiones a realizar.

Tabla 6: Conexión Arduino UNO-Mosfet.

Arduino UNO	Grove-Mosfet
5 [v]	VCC
Tierra	GND
No conectado	NC
PIN-6	SIG

- Una vez realizada la conexión, se abrirá el editor de texto Arduino IDE y se cargara un código ejemplo. Este código puede ser obtenido directamente dentro de la sección de ejemplos, en la sección "Blink", respetando la salida de pines y configurando el tiempo de acuerdo a las preferencias de prueba.
- La segunda opción es cargar el código de ejemplo que puede ser consultado en el repositorio del proyecto, compilando y ejecutando dentro de la interfaz Arduino IDE, véase la figura 20.

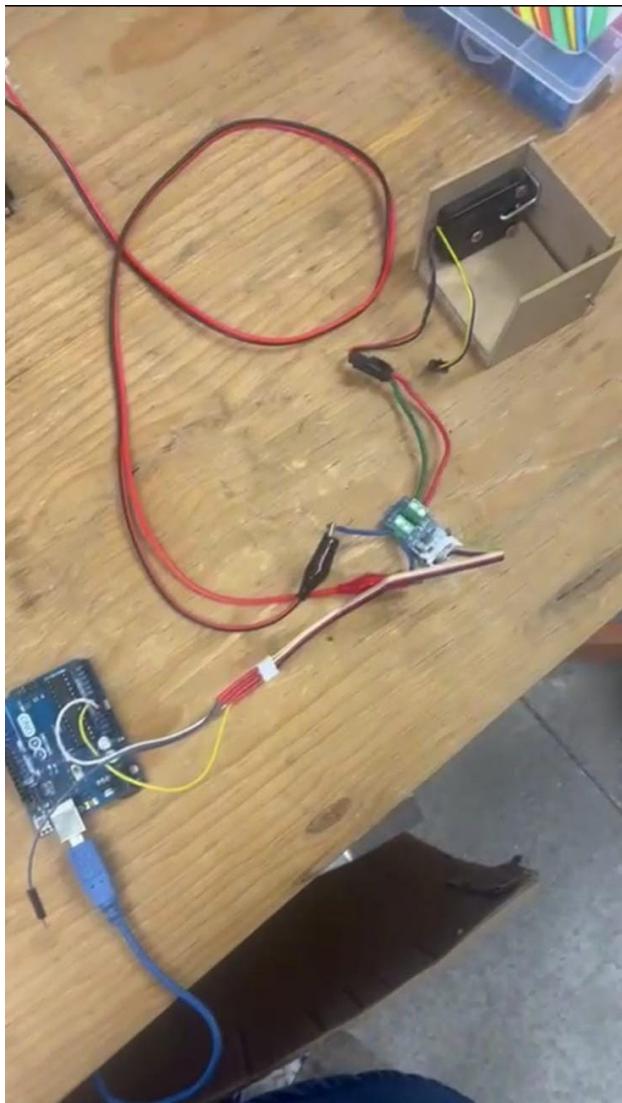


Figura 20: Conexión Arduino, Grove-Mosfet y Fuente de poder.

### 7.3.2. Conexión y testeo de Raspberry Pi con Grove mosfet

Para esta segunda fase de pruebas, se traspaso el circuito a la placa principal del proyecto, en las siguientes tres tablas (7:9), se muestran las conexiones respectivas, y en la figura 21, el datasheet del módulo a conectar.

Material necesario: Raspberry Pi, Grove- MOSFET y Fuente de alimentación.

- Se realizan las siguientes conexiones:

Tabla 7: Conexión Raspberry-Mosfet.

Raspberry Pi	Grove-Mosfet
5 [v]	VCC
Tierra	GND
No conectado	NC
PIN-6	SIG

Tabla 8: Conexión Casillero en escala- Grove Mosfet.

Casillero escala	Grove-Mosfet
V(in)	V(out)
Tierra	GND

Tabla 9: Conexión Fuente de poder-Grove Mosfet.

Fuente Pi	Grove-Mosfet
V(in)	V(out)
Tierra	GND

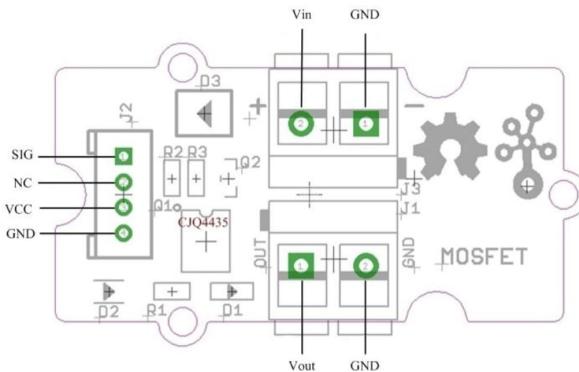


Figura 21: Datasheet Grove-Mosfet (seeed Studio, 2023).

- Es necesario tener todo organizado, así que se creó una carpeta respectiva para el testeo del mosfet, se puede acceder directamente a la ventana de archivos de la Raspberry Pi y crear una carpeta, o mediante una terminal, con la siguiente instrucción: `mkdir /home/"Nombre de la carpeta"`
- Con el comando: `ls`, sabremos donde estamos ubicados
- Crearemos un archivo con el código de prueba, esto se puede agregar directamente desde el editor "Geany" que viene por defecto en la Raspberry Pi para desarrollo de software, o con la siguiente instrucción dentro de una terminal: `nano "nombre del archivo.py"`, es importante mantener la extensión python al final del nombre ".py".

- Por último se carga el código de prueba dentro del editor, se salva y después se procede a ejecutar mediante la siguiente instrucción: *python "nombre del archivo.py"*

### 7.3.3. Sensor Infrarrojo detector de obstáculos AD 32 ir-08h ky 032

En la tabla10 se muestran las conexiones respectivas, en este caso se usara la placa de desarrollo Raspberry Pi.

Tabla 10: Conexión Raspberry-KY-033.

Raspberry Pi	Sensor KY-033
5 [v]	VCC
Tierra	GND
PIN-7	SIG

En la figura 22 se puede observar el la conexión del circuito a la placa de Raspberry Pi, el sensor detecta movimiento y en automatico activa el módulo de cámara v2.1 que sirve como lector.

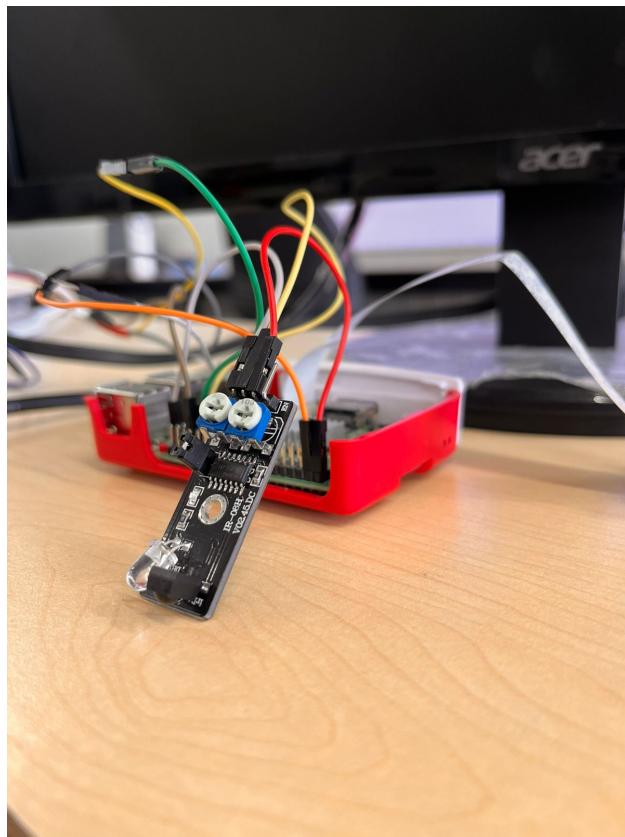


Figura 22: Sensor Ky033

#### 7.3.4. Lector de códigos QR con el módulo de cámara V2.1 de Raspberry Pi

En la figura 23 se observa el módulo de cámara v2.1 Raspberry Pi, se realizó una impresión 3D para resguardar el dispositivo.

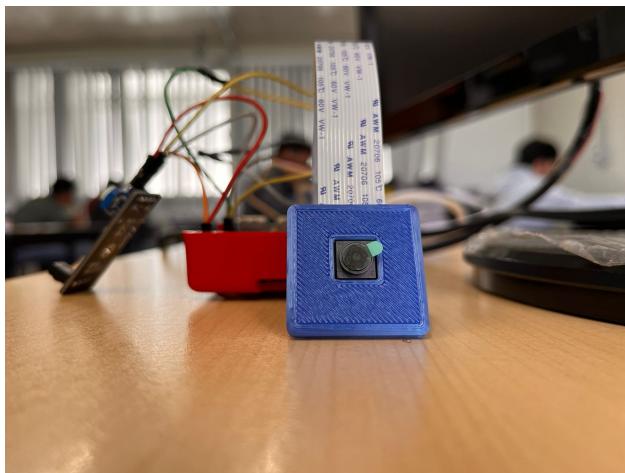


Figura 23: Cámara con su carcasa de protección.

En esta prueba se llevó a cabo la decodificación de un código QR, se integró el sensor KY-032 para poner en marcha la cámara de la Raspberry Pi en cuanto detecta movimiento. Al detectar y comparar el código con la base de datos, si es correcto el mismo, es decir, si coincide con el del usuario, la chapa se abrirá, en caso contrario, la cerradura quedará inamovible (véase la figura 24).



Figura 24: Activación de Lector.

## 8. Conclusiones

En resumen, el presente proyecto representa un hito significativo en la intersección entre la tecnología y nuestra vida cotidiana, demostrando cómo la automatización, la conectividad y la gestión centralizada pueden mejorar tanto la seguridad como la conveniencia. A través de la aplicación de una metodología rigurosa, hemos logrado avanzar en su desarrollo, realizando pruebas exhaustivas para garantizar su eficacia y compatibilidad. Actualmente, el proyecto cumple con su objetivo inicial, pero también deja espacio para futuras mejoras y expansiones. Además, su conclusión no solo refleja el éxito técnico, sino también la integración de los conocimientos adquiridos a lo largo de nuestra formación académica, destacando la aplicación de competencias de investigación y redacción que hemos cultivado durante nuestro recorrido universitario. En última instancia, este proyecto no solo representa un logro técnico, sino también un testimonio de nuestro crecimiento y desarrollo como profesionales en formación.

## 9. Trabajo futuro

Actualmente existe un trabajo futuro, este tipo de sistemas pueden llegar a ser implementados en áreas publicas, así como en espacios cerrados. A continuación se muestran algunas mejoras del sistema.

- Reducir costos.
  - Renovar la estética del producto
  - Añadir nuevos módulos de hardware y/o software.
- 

## Referencias

**Rasp2Rasp2.**(s.f.).

**AmazonAmazon.**Amazon. (2020). *Puntos de entrega: seguros y convenientes para recoger tu paquete en tu horario.* Descargado de [https://www.amazon.com.mx/ulp/view?pd\\_rd\\_w=7xVbbcontent - id = amzn1.sym.56328743 - da16 - 4977 - 872e - 1f88238fd7e1 : amzn1.sym.56328743 - da16 - 4977 - 872e - 1f88238fd7e1pf, d\\_p = 56328743 - da16 - 4977 - 872e - 1f88238fd7e1pf, d\\_r = 9YBK90FKX8G3GKD1840Gpd, d\\_wg = PXFxjpd, d\\_r = 73f0fefe - f535 - 4d47 - 8b61 - 27fea33aedbqid = 1707168090](https://www.amazon.com.mx/ulp/view?pd_rd_w=7xVbbcontent - id = amzn1.sym.56328743 - da16 - 4977 - 872e - 1f88238fd7e1 : amzn1.sym.56328743 - da16 - 4977 - 872e - 1f88238fd7e1pf, d_p = 56328743 - da16 - 4977 - 872e - 1f88238fd7e1pf, d_r = 9YBK90FKX8G3GKD1840Gpd, d_wg = PXFxjpd, d_r = 73f0fefe - f535 - 4d47 - 8b61 - 27fea33aedbqid = 1707168090)

**BouBou.**Bou, R. C. (2019). *Usandoxampp con bootstrap y wordpress.* Mercedes Gómez Alcalá.

**ClubClub.**Club, D. T. (2022). *Husylens : Ai camera with raspberry pi.* Descargado de <http://community.dfrobot.com/makelog312234.html>

**ClubClub.**Club, D. T. (Noviembre,2021). *Testing a cheap ai camera: Gravity husylens.* Descargado de <https://www.youtube.com/wach?v=5PIqpGVp90A=4s>

**David Alejandro Herrera Mencias****David Alejandro Herrera Mencias.**David Alejandro Herrera Mencias, D. C. E., Jairo. (s.f.). Diseño de un sistema automatizado de lockers para la gestión de proyectores.

**Drake y Worsley****Drake y Worsley.**Drake, J. D., y Worsley, J. C. (2002). *Practical postgresql.* .o'Reilly Media, Inc.

**ELECTRONICSELECTRONICS.**ELECTRONICS, U. (2023). *Raspberry pi cámara módulo v2 imx219 8mp.* Descargado de <https://uelectronics.com/producto/raspberry-pi-camara-modulo-v2-imx219-8mp/>

**Electronics****Electronics.**Electronics, U. (S.F). *Sensor de obstáculos módulo ky-033.* Descargado de <https://uelectronics.com/producto/modulo-ky-033-sensor-de-obstaculos/>

**GabinetesGabinetes.**Gabinetes, L. . (2023). *locker metalico de 4 puertas*. Descargado de <https://lockersygabinetes.mx/producto/locker-metalico-grande-4-puertas/>

**Mishra, Kumar, Chowdhury, y BhondekarMishra y cols..**Mishra, P., Kumar, R., Chowdhury, S. R., y Bhondekar, A. P. (s.f.). Region identification and decoding of security markers.

**MuñozMuñoz.**Muñoz, M. J. (S.F). *Trunk boxes*. Descargado de <https://www.pinterest.de/pin/360288038911886408/>

**PiPi.Pi, R.**(2023). *Raspberry pi 3 model b+*. Descargado de <https://datasheets.raspberrypi.com/rpi3/raspberry-pi-3-b-plus-product-brief.pdf>  $g_1 = 1 * 13q2fbg *_g a * NzA2NjU4OTMyLjE3MDUwOTY1MTg. *_g a_2FD70LWDS * MTcwNzI4MjM5MC4xLjEuMTcwNzI4MjQ0My4wLjAuMA..$

**Richardson y WallaceRichardson y Wallace.**Richardson, M., y Wallace, S. (2012). *Getting started with raspberry pi*. .o'Reilly Media, Inc.

## **SEBASTIÁN NICOLÁS CALVACHE ARCOSSEBASTIÁN NICOLÁS CALVACHE ARCOS.**

SEBASTIÁN NICOLÁS CALVACHE ARCOS, D. A. C. P. (s.f.). ImplementaciÓn de un sistema automatizado de lockers para la gestiÓn de proyectos.

**seeed Studioseeed Studio.**seeed Studio. (2023). *Grove - mosfet*. Descargado de <https://wiki.seeedstudio.com/Grove-MOSFET/>

**Singh, Gupta, Singh, y KaurSingh y cols..**Singh, G., Gupta, I., Singh, J., y Kaur, N. (2022). Face recognition using open source computer vision library (opencv) with python. En *2022 10th international conference on reliability, infocom technologies and optimization (trends and future directions)(icrito)* (pp. 1–6).

**S.L.S.L..**S.L., V. (S.F). *Diseño básico de páginas web en html*. Descargado de [https://books.google.es/books?hl=eslr=id=Q4VL8Ti7gcCoi=fndpg=PP2dq=htm\\_lots=Vkc7H-4lwdsig=YwP101\\_sOP9-z2nGpzHRICp93uL\\_8v=onepageq=htmlf=false](https://books.google.es/books?hl=eslr=id=Q4VL8Ti7gcCoi=fndpg=PP2dq=htm_lots=Vkc7H-4lwdsig=YwP101_sOP9-z2nGpzHRICp93uL_8v=onepageq=htmlf=false).

**Upton y HalfacreeUpton y Halfacree.**Upton, E., y Halfacree, G. (2016). *Raspberry pi user guide*. John Wiley & Sons.

**Vivek, Nikhitha, Shaik, Anjali, y SaraswathiVivek y cols..**Vivek, K., Nikhitha, N. R., Shaik, R. S., Anjali, K., y Saraswathi, M. (2022). Developing and reading qr codes. *Journal of Engineering Sciences*, 13(12).

**WetzelsWetzels.**Wetzels, , W. J., A. L. G. M. (S.F). *The password hashing competition and argon2*. Descargado de <https://arxiv.org/ftp/arxiv/papers/1602/1602.03097.pdf>

(Upton y Halfacree, 2016) (Richardson y Wallace, 2012) (Vivek, Nikhitha, Shaik, Anjali, y Saraswathi, 2022) (Drake y Worsley, 2002) (Mishra y cols., s.f.) (David Alejandro Herrera Mencias, s.f.) (SEBASTIÁN NICOLÁS CALVACHE ARCOS, s.f.) (Club, Noviembre,2021) (S.L., S.F) (Club, 2022) (Wetzels, S.F) (Bou, 2019) (Amazon, 2020) (Pi, 2023) (ELECTRONICS, 2023) (Singh, Gupta, Singh, y Kaur, 2022) (Electronics, S.F) (*Rasp2*, s.f.) (Mishra y cols., s.f.) (Upton y Halfacree, 2016) (Gabinetes, 2023) (seeed Studio, 2023)