

# **SISTEMA DE CERRADURAS COMANDADAS POR APLICACIÓN MÓVIL CON SERVIDOR CENTRALIZADO**

*LOCK SYSTEM COMMANDED BY MOBILE APPLICATION WITH CENTRALIZED SERVER*

**Aldonso Becerra Sánchez**

Universidad Autónoma de Zacatecas, México  
*a7donso@uaz.edu.mx*

**Jonathan Alejandro Gutiérrez Hernández**

Universidad Autónoma de Zacatecas, México  
*jonathan.a.gtz@gmail.com*

**Gustavo Zepeda Valles**

Universidad Autónoma de Zacatecas, México  
*gzepeda\_@hotmail.com*

**Santiago Esparza Guerrero**

Universidad Autónoma de Zacatecas, México  
*chago@uaz.edu.mx*

**Nancy Delgado Salazar**

Universidad Autónoma de Zacatecas, México  
*nancydesal@uaz.edu.mx*

**Recepción:** 7/noviembre/2021

**Aceptación:** 20/diciembre/2021

## **Resumen**

El acceso a espacios físicos se da tradicionalmente por inserción de llave física, lo que en ciertos casos puede ser tardado o tener inconvenientes, y aún más cuando se espera un acceso rápido. Asimismo, en los métodos convencionales no se pueden difundir los accesos a diversos usuarios de manera instantánea, remota y sin costo. Este trabajo plantea el desarrollo de una aplicación móvil Android, integrada a un sistema domótico, para que sirva de apoyo a diversas personas como un método rápido y seguro de apertura y cierre de puertas de acceso a espacios físicos. El proyecto se realizó por medio del uso de la metodología Prototyping, la cual guía en la generación del sistema domótico que emplea una Raspberry PI como base. El prototipo además implementa un acceso remoto a cámaras IP mediante la aplicación móvil. Las pruebas exhaustivas realizadas mostraron resultados prometedores en resistencia, eficacia y funcionalidad.

**Palabras Clave:** Aplicación móvil Android, cerraduras electrónicas, Mosquitto broker, Raspberry Pi, sistema domótico.

## **Abstract**

*Access to physical spaces is traditionally given by inserting a physical key, which in certain cases can be time consuming or have problems, and even more so when fast access is expected. Likewise, conventional methods cannot grant access to different users instantly, remotely and without cost. This work proposes the development of an Android mobile application, integrated into a home automation system, to serve as support for different people as a fast and safe method of opening and closing access doors to physical spaces. The project was carried out through the use of the Prototyping methodology, which guides the generation of the home automation system that uses a Raspberry Pi as a base. The prototype also implements remote access to IP cameras through the mobile application. Extensive tests carried out showed promising results in strength, efficacy and functionality.*

**Keywords:** *Android mobile app, electronic locks, home automation system, Mosquitto broker, Raspberry Pi.*

## **1. Introducción**

En los últimos años, tras el surgimiento del Internet de las Cosas (IoT, por sus siglas en inglés) y de las tecnologías asociadas a este, se ha logrado crear un tipo de autonomía de ciertos dispositivos para fomentar los edificios inteligentes. Es entonces que nació el término IoT en el Instituto de Tecnología de Massachusetts (MIT) [Salazar, 2015], aunque fue utilizado por primera vez por Kevin Ashton en 1999 cuando estaba trabajando en el campo de la tecnología RFID (Radio Frequency Identification). Sin embargo, IoT, tal como lo conocemos ahora, nació entre 2008 y 2009, relacionándose a lo que hoy se denomina domótica.

En ese sentido, la domótica se puede entender como la incorporación al equipamiento de nuestras viviendas y edificios de una sencilla tecnología que permita gestionar de forma energéticamente eficaz, segura y confortable para al usuario los distintos aparatos e instalaciones domesticas tradicionales [Valdés,

2016]. Cuando se habla de domótica generalmente se hace uso de un controlador que sea capaz de tener cobertura de los dispositivos más utilizados en el hogar. Entre estos elementos se incluyen, desde lámparas interiores o exteriores, pasando por aquellos que brindan entretenimiento como televisiones y equipos de sonido, hasta lograr automatización de actividades cotidianas como la aspersión del jardín y el control del ambiente interno. En la actualidad existen edificios que son totalmente inteligentes, por lo que este trabajo se centra en los accesos a las casas habitaciones, oficinas y edificios comerciales de cualquier clase.

Diversos trabajos relacionados a este tema se han ido desarrollando a lo largo del tiempo, en los cuales tareas variadas han sido abordadas, incluyendo en ello las limitaciones de estos procesos manuales. Por ejemplo, en [Hernández, 2018] diseñaron un prototipo de una aplicación en Android conectada a una placa Arduino, que a través de un relevador, se conecta a una cerradura electromagnética. La aplicación se accede por el lector de huella dactilar del teléfono, permitiendo activar o desactivar la cerradura. El prototipo se desarrolló utilizando un módulo Bluetooth, un relevador, un teléfono Android y una cerradura electromagnética. El tipo de seguridad proporcionado es útil en hogares o sitios en donde se necesite tener un acceso controlado, como lo pueden ser puertas, cajones, gabinetes, entre otros [Hernández, 2018]. De igual forma, Chang diseñó un prototipo de cerradura de puertas enfocado a personas con discapacidad. Las tecnologías usadas implicaban un Arduino, un pulsador, resistencias, cables, una cerradura y un transformador. El trabajo resultante se probó haciendo funcionar la cerradura mediante un código secreto por medio de golpes sonoros [Chang, 2019].

Bajo esta idea, Garrote planteó un sistema domótico para la apertura de puertas por medio de un instrumento pasivo, el cual se localiza en la misma llave que realiza la acción. La funcionalidad incluye un mecanismo electromecánico, así como la posibilidad de la instalación sobre plataformas existentes con diseño orientado a mejoras futuras y de bajo costo [Garrote, 2017]. En un proyecto similar [Kavde, 2017] desarrollaron un prototipo en Java como parte de una casa inteligente, en el que emplearon un microcontrolador AVR Atmega-32 y una base de datos de registros de visitantes. Aquí, el usuario controla la cerradura de la puerta usando un

teléfono Android y Bluetooth, accediendo en ello a una cámara web en tiempo real con el propósito de tener una perspectiva de lo que sucede [Kavde, 2017]. De manera relacionada, en [Adiono, 2019] diseñaron un prototipo seguro de cerradura de puertas con GPS empleando Android, en donde la idea era no requerir una entrada manual de usuario por razones de conveniencia. El sistema toma el control del entorno usando un microcontrolador STM32L100 como núcleo, al que se le integran componentes como un transistor TIP102 y un módulo Xbee [Adiono, 2019]. Con propósitos semejantes, en [Muthumari, 2018] propusieron un sistema de desbloqueo remoto de cerraduras que funciona basándose en un servomotor controlado por Arduino operado bajo Bluetooth. El proyecto utiliza la red inalámbrica y una aplicación en un teléfono inteligente con el fin de acceder a una cámara P2P Wi-Fi, por medio de la cual se observa la persona que desea ingresar al espacio físico [Muthumari, 2018].

La relevancia del proyecto planteado radica en la aportación a la automatización por medio de la red (usando cámara IP para ver el espacio físico), de la apertura y cierre de puertas de acceso a edificios, aulas, laboratorios, oficinas, entre otros; en donde el acceso se da de manera cómoda, remota, con seguridad y de bajo costo. Este trabajo es comparable con los encontrados en el estado del arte, aunque con mayores capacidades de expansión y escalabilidad. Para la realización del proyecto se modificó la cerradura de la puerta, para que mediante software se lleve a cabo un prototipo que satisfaga la parte funcional planteada. Por lo que se propuso desarrollar una aplicación móvil en Android integrada a un sistema domótico controlado por una Raspberry PI. Esta propuesta ofrece un apoyo a las personas como un método eficiente de apertura y cierre de puertas de acceso a espacios físicos, definiendo políticas de control automatizadas en el proceso.

## **2. Métodos**

Para el desarrollo de este proyecto se empleó la metodología ágil Prototyping. Este paradigma permite obtener retroalimentación de los interesados en el sistema en etapas tempranas, indicando qué funcionalidades del sistema sean descartadas, mientras que nuevas funcionalidades y necesidades sean agregadas conforme van

siendo requeridas [Yiek, 2020]. El prototipo es una versión preliminar intencionalmente incompleta o reducida de un sistema [Weitzenfeld, 2007]. El uso de prototipos es una herramienta útil para aplicarse en casi todas las actividades de diseño y creación de hardware y software.

Para el caso de las cuestiones tácticas del presente trabajo, el prototipo generado se basa en las especificaciones bajo los supuestos de la definición de los objetivos del prototipo, la validación de los requerimientos a través de la funcionalidad del producto, el diseño y desarrollo del prototipo, así como la evaluación del funcionamiento de este. El enfoque Prototyping sigue un proceso que se repite en cada iteración, ver figura 1 [Sommerville, 2012], [Becerra, 2018]. Durante el establecimiento de los objetivos del prototipo, el fin es el seleccionar una de las funcionalidades a desarrollar, aunado a definir hasta donde llegaría el prototipo en etapa de evaluación. Esta tarea exige una selección cuidadosa de características que requieran ser evaluadas por el usuario final para su aprobación o rechazo. Es así que cada paso permite mejorar el sistema antes de que esté finalizado, de tal manera que en cada iteración se van consiguiendo nuevas adiciones o mejoras que habían pasado desapercibidas o no se realizaban correctamente.

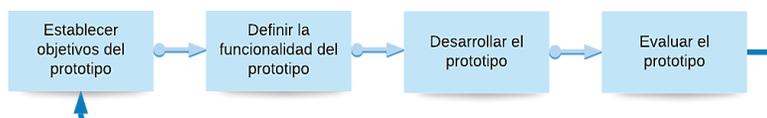


Figura 1 Ciclo de vida de un prototipo.

### **Establecimiento de objetivos del prototipo**

El sistema propuesto, Door IO, se basa en las necesidades presentadas por diferentes miembros de la sociedad. Por lo que fue de vital importancia realizar un listado de requerimientos que permitieran cumplir con las expectativas solicitadas. Para ello, inicialmente se dispuso a llevar a cabo los siguientes requerimientos:

- El sistema debe conceder el acceso a espacios físicos a través de una aplicación móvil/web.
- El sistema debe desplegar una lista de espacios físicos.
- El sistema debe conceder agregar, modificar y eliminar espacios físicos.

- El sistema debe conceder agregar, modificar y eliminar usuarios.
- El sistema debe conceder agregar, modificar y eliminar cámaras IP con el fin observar en tiempo real lo acontecido en el acceso al espacio físico.
- Sistema único con la finalidad de evitar problemas futuros al incorporarlo como parte de otros sistemas que actualmente se están desarrollando.
- El sistema debe ser escalable para llevar a cabo más funcionalidades, siendo adaptable a posibles cambios que se proyecten a futuro.
- La aplicación total debe ser amigable para los usuarios finales.

### Definición de la funcionalidad del prototipo

La funcionalidad implica el uso de los servidores y los requerimientos de usuario, de tal manera que se definieron bajo el análisis realizado:

- **Módulo de servidores.** Se debe disponer de un servidor de base de datos para que albergue la información necesaria; en este caso será MYSQL. Se decidió esta plataforma debido a su capacidad, licencia libre, fácil mantenibilidad y control. Además, se deberá contar con un servidor donde se albergue apache y PHP, el cual fungirá en Ubuntu.
- **SRS (Software Requirements Specification):** requerimientos desde el punto de vista del usuario. Los casos de uso describen el proceso funcional de software, como el de administración, apertura, cierre y el login de usuarios del sistema DoorIO. En la tabla 1 se detallan los requerimientos funcionales que contiene el sistema para el control de acceso a espacios físicos.

Tabla 1 Requisitos funcionales del sistema DoorIO.

Requerimiento	Prioridad	Actores	Descripción
Registrar usuarios en el sistema	Alta	Administrador	El administrador añade información de usuarios a la base de datos.
Gestionar usuarios	Alta	Administrador	Un usuario administrador modifica la información de los usuarios registrados.
Registrar espacios físicos en el sistema	Alta	Administrador	El administrador añade información de espacios físicos a la base de datos.
Gestionar espacios físicos	Media	Administrador	Un usuario administrador modifica la información de los espacios registrados.
Registrar cámaras IP	Alta	Administrador	El administrador añade información de cámaras IP a la base de datos.
Gestionar cámaras IP	Media	Administrador	Un usuario administrador modifica la información de la cámara IP.
Iniciar sesión	Alta	Administrador / Usuario	El usuario o administrador debe registrar su asistencia.

En la figura 2 se muestra el diagrama de los casos de uso y las interacciones que tienen los usuarios con el sistema DoorIO. Estos casos de uso están acorde a los requisitos funcionales definidos en la tabla 1.

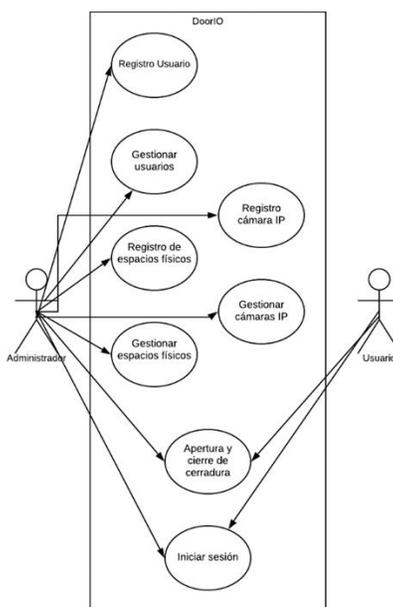


Figura 2 Diagrama de casos de uso del sistema DoorIO.

**Requerimientos no funcionales.** Además de los requerimientos funcionales, no se deben olvidar atributos importantes del sistema, tales como:

- **Confiabilidad.** El sistema deberá de guardar y recuperar los datos al menos un 99% de las veces.
- **Disponibilidad.** El servicio estará en escucha al menos un 99% del tiempo. Cuando el sistema se encuentre ese 1% del tiempo fuera de escucha, el módulo deberá permitir acceder al espacio físico por un medio convencional; en este caso una llave física.
- **Integridad.** El sistema podrá fallar cuando se recuperen o se envíen datos a la base de datos 1 vez en mil de las veces. Después de la falla, el sistema no tendrá pérdida de información, además de que permitirá reenviar la solicitud de los datos a petición del usuario.
- **Usabilidad.** El sistema deberá tener interfaces donde se guíe al usuario final. Cualquier tarea debe hacerse en un máximo de 6 clics.

- **Seguridad.** El sistema deberá denegar el acceso a la aplicación y a la base de datos a personas no autorizadas al menos un 99.99%.

### Desarrollo del prototipo

El sistema domótico está compuesto por cuatro bloques principales. El primero implica la *aplicación*, que se ejecuta en un smartphone con sistema operativo Android, sirviendo como interfaz entre el sistema y el usuario.

El segundo abarca el *módulo del servidor*, el cual contiene los gestores de base de datos y de comunicación necesarios. El tercero (*unidad de control por medio de la Raspberry Pi 3b+*) se encuentra integrado por un microcontrolador encargado de interpretar comandos recibidos, así como activar los circuitos para los servomotores. El cuarto (*actuadores*) lo integran el servomotor y la cerradura, los cuales fungen como el proceso principal de movimiento.

### Esquema general propuesto para el sistema DoorIO

En la figura 3 se observa el esquema general de la propuesta, en donde el componente 1 indica que un administrador puede registrar un usuario, registrar una puerta y registrar una cámara IP.

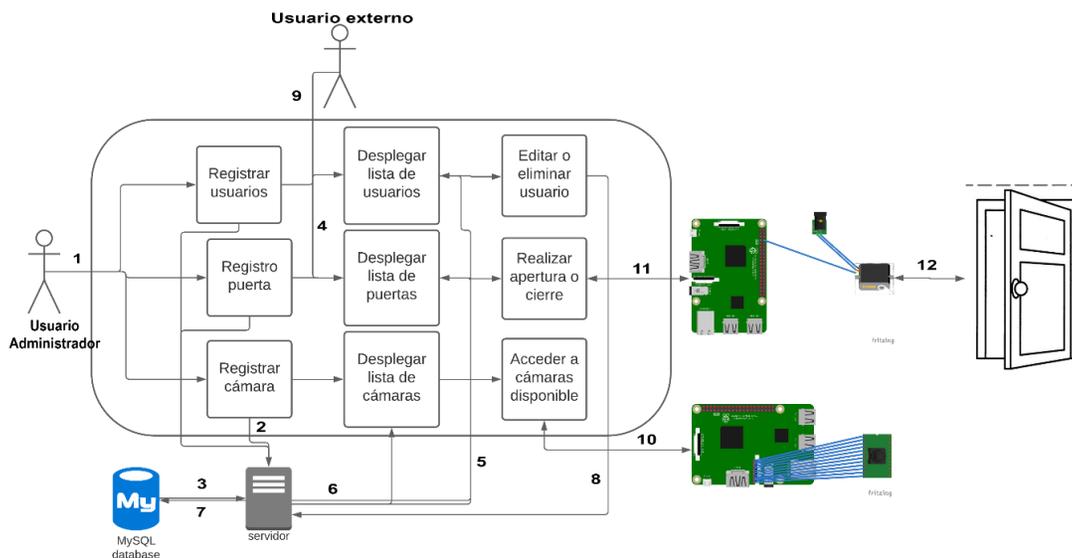


Figura 3 Esquema general propuesto para el sistema DoorIO.

Al realizar cualquiera de estas acciones, el sistema se comunica con el servidor (véase el número 2), donde además se presenta el almacenamiento de los datos (véase el número 3). En el número 4 se encuentran 3 componentes que el sistema despliega en caso de que se regrese un valor verdadero ante algún registro. Lo que se despliega aquí abarca tanto la lista de usuarios como la lista de puertas en donde se encuentran las cámaras. Dado que en el registro de puertas se puede tener una cámara o no, la opción de registro y la posterior lista de cámaras se encuentra vinculada a una puerta.

El módulo propuesto para la cámara IP (véase el número 10) trabaja bajo otra Raspberry Pi; aunque esta opción puede remplazarse según la necesidad del usuario, debido a que este puede colocar una cámara con una mayor resolución, además de poder posicionar esta cámara donde se requiera. La dirección IP y nombre de estas cámaras serán obtenidos del servidor de almacenamiento (véase el número 7); posteriormente se enviarán a través del servidor para ser presentados en el componente de la lista de cámaras (véase el número 6).

Los datos presentados en los módulos de listas (véase el número 5) provienen del servidor de almacenamiento de datos (véase el número 7). Los datos serán enviados del servidor para posteriormente ser proyectados en los competentes de listas indicadas por el número 4 en la ilustración. Para el uso del componente de edición o eliminación de usuarios, se precisa de un usuario administrador para que realice estas actividades. Estos cambios serán enviados a través del servidor (véase el número 8), los cuales se verán reflejados en el servidor de almacenamiento de datos. El administrador puede acceder al listado de puertas para realizar la apertura o cierre de la puerta asignada. Esta acción se llevará a cabo mediante el envío de una cadena al servidor Mosquitto instalado en la Raspberry, donde se comparará la cadena entrante para determinar cuál de las dos acciones realizar. Adicionalmente se tiene el módulo del motor, que realiza mecánicamente el cierre o apertura de la cerradura (véase el número 11). Posteriormente, dependiendo de las dos acciones disponibles que el usuario ejecute, se verá reflejada esa acción (véase el número 12). En cambio, si un usuario externo autorizado ingresa (véase el número 9), solo tendrá acceso a los módulos de listas de puertas (véase el número 4). Este hecho

se da previa autorización por parte del administrador, el cual le asigna qué puertas puede manipular. Por tanto, después de acceder a una puerta mostrada en el listado, podrá utilizar el módulo para realizar una apertura o cierre con fines de acceder al espacio autorizado. Es así que se tiene la interacción únicamente con los módulos especificados con los números 11 y 12.

### Configuración del servidor de datos

Este servidor requiere de una serie de herramientas, las cuales conllevan un proceso de instalación y configuración. Primeramente, se requiere la instalación del sistema operativo Ubuntu, sobre el cual se instaló Apache 2.4.29 y PHP 7.0.33. Una vez hecho esto, se configuró el firewall de Linux con el fin de brindar acceso al servidor web desde la red. Posteriormente se configuró Apache mediante la edición de varios archivos que se encuentran en la ruta `/etc/httpd/`, siendo el principal `httpd.conf`. En este último es donde se editan los permisos que los usuarios podrán tener en el servidor. Finalmente se instaló la versión de MySQL 5.7.33, la cual sirve para organizar y guardar la información por parte de los diversos usuarios. Una vez que la instalación finalizó, se ejecuta la siguiente instrucción para iniciar el demonio: `sudo MySQL`. En consecuencia, se abrirá la consola de MySQL, en donde se creó un usuario con una contraseña, como se muestra a continuación: `CREATE USER 'user'@'localhost' IDENTIFIED BY 'password'`; se presiona enter y después exit.

### Diseño general de la aplicación Android

La plataforma de desarrollo empleada para la aplicación móvil fue Flutter, programada con Dart en Android Studio. El icono de escritorio de la aplicación desarrollada se observa en la figura 4a.



Figura 4 Vista inicial de la aplicación móvil.

Al iniciar sesión mediante credenciales válidas de administrador, se mostrará la pantalla principal de la aplicación móvil; en ella se incluye su nombre, acompañada por un icono para desplegar el menú. En la parte inferior a esta se muestra la lista con los espacios físicos registrados, además de un botón circular de color azul con el signo de adición. En la figura 4b se pueden observar los espacios físicos disponibles en el servidor.

Una vez seleccionado un espacio de la lista, se presenta la vista principal que controla la cerradura del espacio especificado, ver figura 5a. Aquí, la barra superior se encuentra en color rojo dado que no está conectado al dispositivo de dicho espacio. Una vez que se conecta, la barra superior cambia a color verde, además de que se observa el video de la cámara que se proporcionó en el registro. En las figuras 5b y 5c se puede ver el accionar del botón deslizante, el cual se encontraba de color rojo, una vez que este se desliza, cambia a color verde, con la leyenda "Abierto". Ese botón reacciona al momento de que otro usuario autorizado desliza el botón desde su sesión; en donde ambos usuarios pueden observar los cambios en el momento que se realizan.



Figura 5 Estado de desconexión/conexión con cámara observando a una puerta.

### Conexiones eléctricas generales del sistema domótico

El uso de diagramas de conexiones brinda al lector una idea más clara de cómo realizar el cableado correcto entre los dispositivos, así como poder llegar a tener un buen funcionamiento de lo que se desee hacer. La figura 6 muestra una vista de este tipo, en donde se observa cómo conectar los componentes que integran este

sistema domótico. Seguir los diagramas establecidos hace que la conexión sea más fácil de entender; también muestra cómo llevar a cabo dichas conexiones para prevenir y no causar algún corto circuito, así como evitar dañar los componentes implicados. Para llevar a cabo la conexión entre los mecanismos, se utilizaron dos cables USB conectados a dos cargadores de 5 V para poder alimentar la tarjeta y el servomotor, y esta pueda abastecer energía eléctrica a los dispositivos. Además, se muestra cómo queda finalmente ensamblado el circuito del sistema domótico. Una vez realizado el cableado correcto entre el circuito de conexiones del sistema domótico, se procede a su instalación.



Figura 6 Ensamblado de circuito de Raspberry Pi y servomotor.

### **Configuración del servidor Raspberry Pi**

Otra parte esencial del prototipo es la configuración de la placa Raspberry Pi, la cual será la encargada de manejar la lógica para la realización de la apertura y el cierre de la cerradura del espacio físico. El sistema operativo que se eligió para este caso es Raspbian OS, el cual permite controlar los GPIO de manera más sencilla. En este caso se decidió usar un servidor MQTT para la comunicación aplicación-motor, instalando el módulo Mosquitto. Antes de correr el broker, se realizaron algunas configuraciones requeridas; por ejemplo, asignar el puerto 8080 en el archivo de configuración generado por defecto (`/etc/mosquitto.conf`). También se configuró una salida digital de la placa Raspberry Pi en el GPIO 8, en la cual se conectó un servomotor mediante un cable USB a un adaptador de corriente de 5.2 V a 3 A para proteger la tarjeta Raspberry Pi. En la figura 7 se contemplan el cable VCC y GND extraídos para realizar la conexión al cable USB; donde el cable GND se conectó al pin GND de la tarjeta Raspberry.

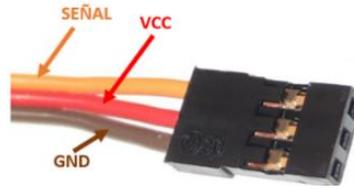


Figura 7 Conexiones del servomotor.

En la figura 8 se presenta el circuito del servomotor, ahí se puede observar que el cable de color amarillo (o de control) del servo es conectado al GPIO 8; de igual forma, el cable de color rojo se conecta a 5 V, y el cable de color café se conecta a la tierra de la tarjeta y al cable de color negro del cable USB.

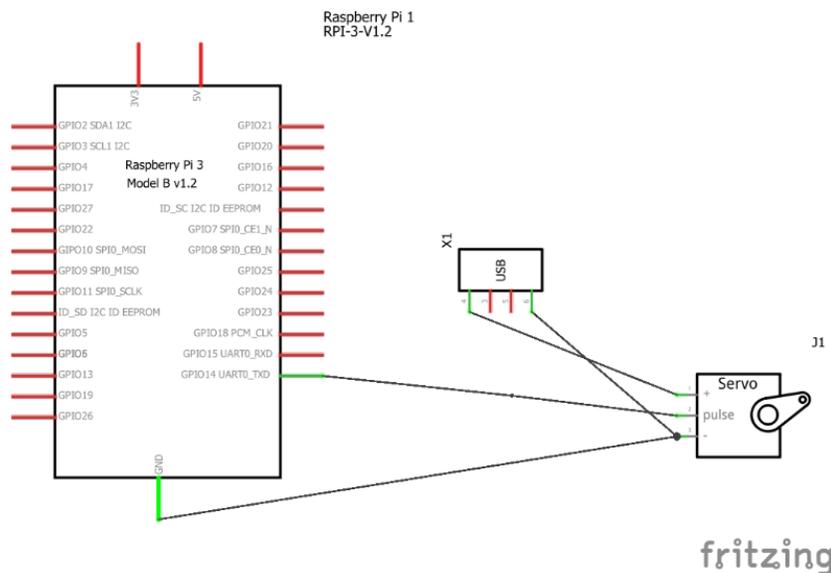


Figura 8 Ensamblado de circuito de Raspberry Pi y servomotor.

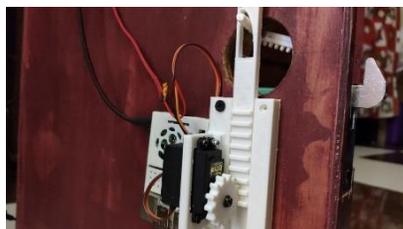
Para manejar el micro servomotor MG995 con la Raspberry, es necesario incluir en el código de programación la librería RPi, permitiendo manipular a la gran mayoría de los servomotores. Se declara entonces RPi.GPIO como GPIO, para posteriormente declarar una variable servo en el pin 8 y poner la Raspberry en modo board; luego se pone el pin del servo como salida. Al final se coloca en modo PWM y se envía una frecuencia de 50 Hz (50 pulsos por segundo), para finalmente iniciar el servo sin ningún pulso, ver figura 9.

```
4 import RPi.GPIO as GPIO #Importación de la librería RPi
5
6 import time #Importación time para poder usar time.sleep
7 import paho.mqtt.client
8
9 servo = 8#Ponemos al servo en el pin 8
10
11 GPIO.setwarnings(False)
12 GPIO.setmode(GPIO.BOARD) #Se pone la Raspberry en modo Board
13 GPIO.setup(servo, GPIO.OUT) #Se pone el pin del servo como salida
14
15 p=GPIO.PWM(servo,55) #Se pone en modo PWM y se envía una frecuencia de 50hz (50 pulsos por segundo)
16
17 p.start(0.75) #Se inicia el servo sin ningún pulso
18 time.sleep(1)
```

Figura 9 Configuración de servomotor en código Python.

## Evaluación del prototipo

En esta fase de evaluación del prototipo se dictaminó qué elementos se pueden considerar aceptables y qué elementos se pueden someter a una nueva iteración de acuerdo con el modelo de Prototyping. Para completar el diseño físico de la chapa, se hicieron impresiones 3D de engranes que acomodaran el motor con el tipo de cerradura corrediza usada en las pruebas, figura 10.



a) Estado de cierre



b) Estado de apertura

Figura 10 Estado de la puerta con el mecanismo implementado.

En el sistema desarrollado se hicieron varias pruebas en los diversos módulos; estas pruebas fueron realizadas durante 10 minutos para cada acción de apertura y cierre, con una diferencia de 4 segundos entre ellas. Se mostrarán algunas imágenes donde se demuestra cómo es el sistema instalado en una puerta de acceso a un espacio físico. Se mostrarán dos estados, los cuales son, estado de espera y de acción, además se comentarán tiempos de ejecución, así mismo se indicarán las fallas o contratiempos en el sistema. En la figura 10a se observa cómo es el sistema en condición de “Cerrado”. Ahí se denota que el motor está en estado de reposo, además de que el pestillo de la cerradura se encuentra visible; entonces el sistema se encuentra a la espera para llevar a cabo la apertura por parte de un

usuario autorizado. En cambio, en la figura 10b se puede observar que el mecanismo se encuentra en estado de “Abierto”, y se observa como el pestillo se encuentra retraído. En ambos casos se puede observar que la placa no tiene color, debido a que la placa Raspberry se encuentra en modo “board”. El indicador led rojo de power se encuentra apagado y solo se cuenta con el led de color verde, aunque este es intermitente y no está todo el tiempo en ejecución, por lo que no se aprecia en las imágenes. La prueba general realizada consistió en dejar al mecanismo funcionar (apertura y cierre) por 10 minutos en dos fases, una con funcionamiento continuo y otro pausado. Los tiempos para realizar las observaciones fueron de dos minutos de diferencia, los resultados se plasmaron en la tabla 2.

Tabla 2 Comparativa entre uso continuo y uso pausado del servomotor.

Tiempo	Ejecución continua	Ejecución pausada
2 minutos	El motor funciona bien, no se notan problemas.	El motor funciona bien, no se notan problemas.
4 minutos	El motor funciona bien, no observan detalles mayores.	El motor funciona bien, no se notan problemas.
6 minutos	El motor comienza a sobrecalentarse y comienza a costar la realización de la apertura.	El motor funciona bien, no se notan problemas.
8 minutos	El motor se siente muy caliente al tacto, además de que no realiza acciones consecutivas y tiende a fallar al momento de realizar la acción de apertura y cierre.	El motor funciona bien, no se notan problemas.
10 minutos	Se decidió detener la prueba, debido a que el motor ya no realizaba ninguna acción de apertura o cierre.	El motor funciona bien, no se notan problemas.

Las pruebas realizadas son satisfactorias en los casos en los que se suponía debería funcionar. Dada la inoperatividad del motor totalmente sellado, después de 10 minutos da la sensación de que no se cumple con estándares de funcionalidad. Sin embargo, cabe recalcar que esto fue debido a que se corrió la prueba de forma continua y sin descanso; donde se ponía a prueba la resistencia del motor ante un caso de falla como sobrecalentamiento. Dicha prueba fue un éxito, puesto que, en un entorno normal, el uso continuo del mecanismo no sería así. Además de que el sistema se puede instalar en cualquier ambiente que se requiera, lo que hace que el sistema se considere rentable, puesto que responde a las necesidades.

La aplicación desarrollada es compatible con dispositivos móviles que cuenten con el sistema operativo Android a partir de la versión 6. Esta cuenta con un sistema de autenticación mediante un usuario y contraseña localizados en la base de datos.

Este prototipo de sistema de automatización maneja en su totalidad las herramientas adecuadas, las cuales fueron planteadas antes de su puesta en marcha, además de la interfaz desarrollada para dispositivos móviles.

Las tecnologías empleadas hoy en día están al alcance de la mayoría de la población, por lo que además de lo dicho en esta investigación, se puede desarrollar domótica por cualquier método para implementarla en espacios del mundo real. Por ello, estos sistemas pretenden que las personas logren una mejor calidad de vida y un ahorro en su economía. Si la red eléctrica o la red de internet llegaran a caer, el sistema podrá seguir funcionando de manera manual con llave física; puesto que el sistema está diseñado para no solo depender de la energía eléctrica, si no para que la cerradura siga siendo autónoma en estos casos.

### **3. Resultados**

El producto se desarrolló siguiendo varias consideraciones, entre ellas se incluyen la realización de una investigación preliminar de los elementos relevantes del tema de caso de estudio. Así se determinaron los tipos de cerradura electrónica que se manejan en el mercado, además de la infraestructura física y de hardware que de soporte a los requerimientos planteados; permitiendo diseñar la arquitectura que defina el esquema de cerradura electrónica. Todo esto permitió ir desarrollando un prototipo de cerradura con la capacidad de apertura o cierre, definiendo la arquitectura y esquema de la aplicación móvil. Al final se logra generar una aplicación que se utilice como llave electrónica de control, además de integrar el acceso remoto a una cámara IP mediante el módulo correspondiente. De esta manera se utilizan los beneficios de la domótica, que incluyen entre otros, software gratuito y hardware asequible; los cuales ayudaron a implementar un prototipo que eventualmente reduce costos y brinda comodidad al personal que se encuentra en esos espacios. En este proyecto se han tomado a consideración los sistemas y aplicaciones existentes en el mercado actual, para por medio de esto poder comparar la utilidad y precio con respecto al sistema usado. Donde la comparación resulta ser ventajosa para el sistema creado mediante esta investigación. Esto se debe a que su utilidad se ve equiparable a diversas cerraduras de fabricantes

comerciales tomadas como referencia (como August, Kwikset, OWSOO, Steren, TASO); en donde en algunos casos, se observa superior en aspectos técnicos que las que se mencionan no poseen.

#### **4. Discusión**

El desarrollo de una aplicación móvil, integrada a un sistema domótico, permite ofrecer una alternativa de apoyo a diversas personas como método seguro y cómodo para la apertura y cierre de puertas de acceso a espacios físicos. Al comparar el proyecto realizado con trabajos relacionados, el prototipo propuesto se puede percibir con mejoras en algunos sentidos desde el punto de vista de practicidad, comodidad y escalabilidad; sin considerar la expansión que puede llegar a tener con la inclusión en varios sistemas adjuntos. Esto con base a cómo se planteó el diseño de dichas piezas y la fuerza propuesta para el servomotor.

Bajo la base de la funcionalidad de los prototipos presentados en esta sección, se debe tomar en cuenta que cualquier sistema es expuesto a consideraciones de cambios futuros, tanto en el diseño como en el propio esquema del software desarrollado. Se sintetizan los cambios expuestos que quedan abiertos en la propuesta actual; en donde las “interfaces” representan los elementos gráficos (con probabilidad de cambio mediana), la “funcionalidad” implica los requisitos del usuario (con probabilidad de cambio alta), los “datos” (con probabilidad de cambio mediana) y “funciones” (con probabilidad de cambio alta) equivalen a los componentes internos que se usan para describir a los “objetos” (correspondientes a las estructuras de datos básicas de la programación, con probabilidad baja de cambio), en tanto que la “información” (con probabilidad de cambio baja) representa el dominio del problema en una aplicación. La arquitectura del sistema debe distinguir entre elementos con mayor y menor probabilidad de cambios. De igual manera, el desarrollo del software debe contemplar un modelo de procesos en el que los componentes de mayor probabilidad de cambio no “arresten” a los más estables. Por ende, el catálogo de funcionamientos y herramientas proporcionadas por el prototipo presentado aquí propician un cambio constante en su diseño para adecuarse a los momentos y lugares físicos donde se implemente.

## **5. Conclusiones**

En este trabajo se desarrolló una aplicación móvil, integrada a un sistema domótico, para que sea considerada como una alternativa en apoyo a diversas personas como un método seguro y cómodo de apertura y cierre de puertas para acceso a espacios físicos. El prototipo, el cual está compuesto por un sistema mecánico y electrónico desarrollado usando la metodología Prototyping, permite que la cerradura de una puerta realice su función de forma automática o manual. Con el uso de una Raspberry PI se logró conseguir la potencia necesaria para realizar las tareas que se requieren en este mecanismo. Asimismo, el producto brinda una conexión a la red mediante su puerto Ethernet o su conexión Wi-Fi. Adicionalmente se cuenta con una conexión de cámara IP para la visualización del espacio físico en el cual se aplicará el mecanismo; esto con fines de acceso remoto a la función y como procedimiento complementario de seguridad.

Como trabajo futuro se espera en primer lugar integrar un mecanismo que pueda ser aplicado para diversos tipos de cerraduras. También se pretende mejorar aún más la seguridad y versatilidad en un sistema de este tipo, integrando para ello esquemas de reconocimiento facial, así como módulos que comanden la cerradura por voz o huella dactilar. En consecuencia, se espera generar módulos que den respuesta sobre el estado en el que se encuentra la cerradura mediante audios. Además, se visualiza la necesidad de la implementación de un sistema de respaldo de energía por si llegara a existir un corte eléctrico en el lugar de instalación. Asimismo, se espera incluir un módulo que registre los datos de localización GPS del lugar en dónde se está realizando la apertura o cierre de una puerta, tal que se brinde certidumbre acerca del manejo de esta información.

## **6. Bibliografía y Referencias**

- [1] Adiono, T., (2019). IoT-Enabled Door Lock System. *International Journal of Advanced Computer Science and Applications*, Vol. 10, No. 5, 445–449.
- [2] Chang, A., (2019). Diseño de prototipo de cerradura de puerta controlada para personas con discapacidad, Tesis de licenciatura, Universidad de Guayaquil. Repositorio institucional.

- [3] Becerra, A., (2018). Learning content management software personalized for a university environment, *Pistas Educativas*, Vol. 40, No. 130, 347-362.
- [4] Garrote, E., (2017). Cerradura electrónica con sistema de alimentación integrado en la llave, Tesis de licenciatura, Universidad Politécnica de Madrid. Repositorio institucional.
- [5] Hernández, A. & Guzmán, D., (2018). Prototipo de cerradura inteligente usando aplicación en Android, Tesis de licenciatura, Universidad Autónoma De Ciudad Juárez. Repositorio Institucional.
- [6] Kavde, S., (2017). Smart Digital Door Lock System Using Bluetooth Technology. ICICES Proceedings.
- [7] Muthumari, M., (2018). Arduino Based Auto Door Unlock Control System by Android Mobile through Bluetooth and Wi-Fi. ICCIC Proceedings.
- [8] Salazar, J. & Silvestre, S., (2020). Internet de las cosas. TechPedia: <http://www.techpedia.eu>.
- [9] Sommerville, I., (2012). *Software Engineering*, 5a ed., vol. 10. Boston: Addison-Wesley.
- [10] Valdés, W., (2016). Domótica un desarrollo sustentable, Tesis de licenciatura, Universidad de La Salle, Bogotá. Repositorio institucional.
- [11] Weitzenfeld, A. & Guardati, S., (2007). Capítulo 12: Ingeniería de software: el proceso para el desarrollo de software. Libro: *Introducción a la Computación*. México D.F: CENGAGE Learning.
- [12] Yiek, W., (2020). Face Reidentification System to Track Factory Visitors using OpenVINO, Tesis de licenciatura, Universiti Tunku Abdul Rahman. Repositorio institucional.