

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

**DESARROLLO DE UN SISTEMA DE SEGURIDAD INALÁMBRICO
MEDIANTE EL USO DE UNA APLICACIÓN MÓVIL Y MÓDULOS
ESP32**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN ELECTRÓNICA Y REDES DE INFORMACIÓN**

JUAN DAVID TAPIA LÓPEZ

DIRECTOR: M.Sc. PABLO WILIAN HIDALGO LASCANO

Quito, marzo 2022

AVAL

Certifico que el presente trabajo fue desarrollado por Juan David Tapia López, bajo mi supervisión.

M.Sc. Pablo Hidalgo L.
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Juan David Tapia López, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

Juan David Tapia López

DEDICATORIA

Este proyecto va dedicado en primer lugar a mi familia, mis padres que han entregado todo su esfuerzo para llegar a este punto hicieron que cada paso que parecía imposible y cada error cometido sea menos grave. A mis hermanas que me ayudaron y alentaron el todo el trayecto, que conocen cada caída y cada vez que no tenía ganas de continuar.

Dedicado a mis amigos de toda la vida, Carlos, Andrés, Daniel, quienes a pesar de verme fallar siempre estuvieron dando un hombro al cual poder arrimarme, así como también, la mano para levantarme y ayudarme a seguir adelante. A mis amigos de la universidad, porque son los únicos que entienden el sacrificio y la felicidad de haber culminado.

A mi tutor de tesis y profesor dentro de la universidad, ingeniero Pablo Hidalgo, por su gran ayuda, por cada recomendación, por cada consejo y experiencia compartida, ya que ha hecho que el proceso sea más fácil.

AGRADECIMIENTO

En primer lugar, agradezco a Dios, por permitirme culminar una etapa más en mi vida, a pesar de los altos y bajos que se han dado a lo largo del camino y el tiempo que ha tomado llegar a este momento.

Agradezco también a mis padres, por ser las manos que siempre me ayudaron a seguir, por ponerse de escalones, ya que, gracias a esos esfuerzos, es que he podido llegar tan lejos.

Agradezco a mis hermanas, Geovi y Thaly por ayudarme en cada semestre, no dejar que me rinda y poder culminar la etapa de la universidad. El amor que llegamos a tener y la comprensión fueron esenciales para el progreso.

A todos mis amigos, desde los de la infancia como los de la universidad porque con ellos se hicieron todas las locuras de adolescencia.

ÍNDICE DE CONTENIDO

AVAL.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
ÍNDICE DE FIGURAS	I
RESUMEN	I
ABSTRACT	II
1. INTRODUCCIÓN.....	1
1.1 OBJETIVOS	1
1.2 ALCANCE	1
1.3 MARCO TEÓRICO	3
1.3.1 SISTEMAS DE SEGURIDAD.....	3
1.3.2 SISTEMA OPERATIVO ANDROID.....	3
1.3.2.1 Arquitectura de la Plataforma Android.....	4
1.3.2.2 Aspectos básicos de la Aplicación	5
1.3.3 FIREBASE	8
1.3.4 ARDUINO IDE.....	9
1.3.5 MÓDULO ESP32	10
1.3.5.1 Especificaciones Técnicas del Módulo ESP32	11
1.3.5.2 ESP32-CAM.....	12
1.3.6 SENSORES	13
1.3.6.1 Sensor de movimiento PIR HC-SR501.....	13
1.3.6.2 Sensor Biométrico	14
1.3.7 METODOLOGÍAS DE DESARROLLO KANBAN	15
2. METODOLOGÍA.....	17
2.1 DISEÑO DEL SISTEMA.....	17
2.1.1 REQUERIMIENTOS DEL SISTEMA.....	17
2.1.2 CASOS DE USO	23
2.1.3 REQUERIMIENTOS DE HARDWARE.....	25
2.1.4 REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES.....	26

2.1.5	HISTORIA DE USUARIO.....	26
2.1.6	DEFINICIÓN DE TABLERO KANBAN	29
2.1.7	ESTRUCTURA GENERAL DEL SISTEMA INTEGRADO	30
2.1.7.1	Herramientas para el desarrollo del Sistema	30
2.1.8	DIAGRAMA DE FLUJO	31
2.1.9	DIAGRAMAS DE SECUENCIA.....	34
2.1.10	ESQUEMA DE AUTENTICACIÓN.....	37
2.1.11	ESQUEMA DE BASE DE DATOS.....	37
2.1.12	DISEÑO DE ACTIVIDADES Y FRAGMENTOS DE LA APLICACIÓN	37
2.1.13	PROGRAMACIÓN EN SOFTWARE ARDUINO IDE.....	40
2.1.13.1	Dispositivo de apertura de puerta principal.....	40
2.1.13.2	Dispositivo de control de acceso.....	41
2.1.14	SISTEMA DE ALIMENTACIÓN (UPS)	42
2.2	FASE DE IMPLEMENTACIÓN.....	44
2.2.1	IMPLEMENTACIÓN DEL SOFTWARE	44
2.2.1.1	Implementación de la aplicación móvil	44
2.2.1.1.1	<i>Especificación de la interfaz de usuario.....</i>	45
2.2.1.1.2	<i>Uso de adaptadores para listar datos de una vista</i>	48
2.2.1.1.3	<i>Eventos.....</i>	49
2.2.1.1.4	<i>Actividades de la aplicación.....</i>	50
a.	Pantalla de Inicio.....	50
b.	Pantalla de Sesión	51
c.	Interfaz de Consola	52
d.	Fragmento Inicio (Sistema Seguridad)	53
e.	Interfaz de Cámara de Seguridad	55
f.	Interfaz de Control de Acceso a Puerta.....	55
g.	Fragmento Perfil.....	57
h.	Interfaz de administración de Usuarios	57
i.	Pantalla ingreso usuarios.....	59
2.2.1.2	Implementación de Software de módulos ESP Arduino IDE.....	60
2.2.2	FASE IMPLEMENTACIÓN HARDWARE	63
2.2.2.1	Implementación Dispositivos Electrónicos.....	63
2.2.2.1.1	<i>Cámara de Videovigilancia.....</i>	63
2.2.2.1.2	<i>Control Activador Puerta Principal.....</i>	65
2.2.2.1.3	<i>Diseño del dispositivo de control de acceso</i>	67
2.2.3	MONTAJE EN EL LUGAR A SUPERVISAR.....	70
3.	RESULTADOS Y DISCUSIÓN	73

3.1	PRUEBAS EN APLICACIÓN MÓVIL.....	73
3.1.1	REGISTRO Y ACCESO USUARIO	73
3.1.2	NAVEGACIÓN EN PANTALLA PRINCIPAL	77
3.1.3	CREACIÓN DE USUARIOS INVITADOS	77
3.2	PRUEBAS EN DISPOSITIVOS.....	79
3.2.1	CÁMARA DE SEGURIDAD	80
3.2.2	CERRADURA PUERTA PRINCIPAL	86
3.2.2.1	Activación	86
3.3	ANÁLISIS Y DISCUSIÓN DE RESULTADOS.....	96
4	CONCLUSIONES Y RECOMENDACIONES.....	98
4.1	CONCLUSIONES	98
4.2	RECOMENDACIONES.....	99
5.	REFERENCIAS BIBLIOGRÁFICAS	100
ANEXOS1		

ÍNDICE DE FIGURAS

Figura 1.1 Componentes de la plataforma Android.....	4
Figura 1.2 Una ilustración simplificada del ciclo de vida de la actividad.....	6
Figura 1.3 Node MCU32	11
Figura 1.4 Node MCU32 distribución de pines	12
Figura 1.5 ESP32-CAM Distribución pines	13
Figura 1.6 Sensor PIR HC-SR501	14
Figura 1.7 Sensor Biométrico	15
Figura 1.8 Esquema de conexión Sensor biométrico	15
Figura 1.9 Representación de tablero Kanban	16
Figura 2.1 Resultados a la pregunta 1 de la Entrevista.....	18
Figura 2.2 Resultados a la pregunta 2 de la Entrevista.....	18
Figura 2.3 Resultados a la pregunta 3 de la Entrevista.....	18
Figura 2.4 Resultados a la pregunta 4 de la Entrevista.....	19
Figura 2.5 Resultados a la pregunta 5 de la Entrevista.....	19
Figura 2.6 Resultados a la pregunta 6 de la Entrevista.....	20
Figura 2.7 Resultados a la pregunta 7 de la Entrevista.....	20
Figura 2.8 Resultados a la pregunta 8 de la Entrevista.....	21
Figura 2.9 Resultados a la pregunta 9 de la Entrevista.....	21
Figura 2.10 Resultados a la pregunta 10 de la Entrevista	22
Figura 2.11 Resultados a la pregunta 11 de la Entrevista	22
Figura 2.12 Resultados a la pregunta 12 de la Entrevista	22
Figura 2.13 Diagrama de Caso de Uso del Sistema	24
Figura 2.14 Tablero Kanban en metodología.....	30
Figura 2.15 Diagrama de Flujo Cámara de Seguridad	32
Figura 2.16 Diagrama de Flujo de Acceso Puerta Principal desde el panel	33
Figura 2.17 Diagrama de Flujo de Acceso Puerta Principal desde el móvil.....	33
Figura 2.18 Diagrama de Secuencia Lista de Dispositivos.....	34
Figura 2.19 Diagrama de Secuencia Lista de Usuarios.....	35
Figura 2.20 Diagrama de Secuencia creación Usuario	36
Figura 2.21 Diagrama de Secuencia Conexión Cámara	37
Figura 2.22 Pantallas de Presentación, Inicio y Principal	38
Figura 2.23 Actividad de Cámara.....	39
Figura 2.24 Actividad perfil de usuario	39
Figura 2.25 Actividad Notificaciones	40
Figura 2.26 Actividad Control Acceso.....	40
Figura 2.27 Diagrama de flujo configuración inicial del control de apertura de la puerta .	41
Figura 2.28 Diagrama de flujo accionamiento del control de apertura de puerta.....	41
Figura 2.29 Diagrama de flujo de la configuración inicial del control de acceso	42
Figura 2.30 Diagrama de flujo funcionamiento del control de acceso.	43
Figura 2.31 UPS utilizada para el sistema de seguridad	43
Figura 2.32 Tablero Kanban en fase de implementación	44
Figura 2.33 Sección del código de AndroidManifest	46
Figura 2.34 Código XML de Actividad Principal	46
Figura 2.35 Código de Actividad Principal en JAVA.....	47
Figura 2.36 Clase Usuario	48
Figura 2.37 Adaptadores	49
Figura 2.38 Código Adaptador Dispositivos.....	49
Figura 2.39 Código de Evento Puerta Principal	50

Figura 2.40 Estructura e interfaz de la pantalla de inicio	51
Figura 2.41 Estructura e interfaz de pantalla de sesión.....	52
Figura 2.42 Estructura e interfaz de actividad principal.....	53
Figura 2.43 Menú principal para Navigation View (Fragmentos)	53
Figura 2.44 Estructura e interfaz del fragmento.....	54
Figura 2.45 Menú agregar dispositivos.....	54
Figura 2.46 Estructura e interfaz de ítem dispositivo.....	54
Figura 2.47 Estructura e interfaz de cámara de seguridad	55
Figura 2.48 Estructura e interfaz acceso puerta	56
Figura 2.49 Estructura e interfaz de perfil de usuario.....	58
Figura 2.50 Pantalla lista usuarios	58
Figura 2.51 Estructura e interfaz ingreso de nuevo usuario	59
Figura 2.52 Librería WiFiManager.....	60
Figura 2.53 Librería Firebase.....	61
Figura 2.54 Librerías de control de acceso.....	61
Figura 2.55 Codificación de panel de acceso	62
Figura 2.56 Codificación de apertura de cerradura.....	62
Figura 2.57 Codificación mensaje desde Firebase a aplicación móvil	63
Figura 2.58 Conexión PIR a ESP32-CAM	64
Figura 2.59 Placa PCB ESP32-CAM.....	64
Figura 2.60 Ensamblaje de cámara con sensor PIR.....	64
Figura 2.61 Vistas de Cámara, frontal y posterior.....	65
Figura 2.62 Conexión circuito falla de energía.....	65
Figura 2.63 Control bloqueo de llave de cerradura	66
Figura 2.64 Dispositivo de apertura de puerta principal	66
Figura 2.65 Diseño de placa PCB activador cerradura.....	66
Figura 2.66 Placa de activador cerradura.....	67
Figura 2.67 Conexión del teclado a la ESP32.....	67
Figura 2.68 Conexión de la ESP32 hacia la pantalla LCD 16x2.....	68
Figura 2.69 Circuito del control de acceso.....	69
Figura 2.70 Placa PCB de control de acceso	69
Figura 2.71 Parte posterior circuito control acceso	70
Figura 2.72 Parte frontal control acceso.....	70
Figura 2.73 Dispositivo Control de Acceso	71
Figura 2.74 Dispositivo activador de apertura.....	71
Figura 2.75 Dispositivo Cámara.....	72
Figura 2.76 Dispositivos del sistema de seguridad instalados	72
Figura 3.1 Tablero Kanban en Pruebas de Funcionamiento.....	73
Figura 3.2 Inicio de sesión y Creación de usuario desde la aplicación	74
Figura 3.3 Fallo de contraseñas.....	75
Figura 3.4 Campo vacío	75
Figura 3.5 Correo ya registrado en otro usuario	76
Figura 3.6 Registro de usuarios autenticados en Firebase.....	76
Figura 3.7 Creación de usuario en la base de datos Firebase.....	76
Figura 3.8 Barra de Navegación y Fragmentos en Actividad Principal.....	77
Figura 3.9 Perfil Usuario registrado.....	78
Figura 3.10 Usuarios Invitados.....	78
Figura 3.11 Creación usuario invitado.....	79
Figura 3.12 Página principal WiFiManager.....	80
Figura 3.13 Configuración de credenciales a la red	80

Figura 3.14 Plataforma Firebase dispositivo cámara No anclado	81
Figura 3.15 Dispositivo no anclado en aplicación móvil	81
Figura 3.16 Acceso a Pantalla de cámara	82
Figura 3.17 Imagen capturada por botón en aplicación móvil.....	82
Figura 3.18 Imagen Capturada por el usuario.....	83
Figura 3.19 Envío de notificación por sensor de movimiento.....	83
Figura 3.20 Código para error de contraseña hasta 3 veces.....	84
Figura 3.21 Solicitud Captura por contraseña de Emergencia.....	84
Figura 3.22 Captura de imagen por solicitud de control de acceso.....	85
Figura 3.23 Código de notificación de Contraseña errada.....	85
Figura 3.24 Notificación por error en la contraseña.....	86
Figura 3.25 Notificación en base de datos Firebase	86
Figura 3.26 Dispositivo no anclado a ningún usuario	87
Figura 3.27 Dispositivo anclado a una cuenta	87
Figura 3.28 Ingreso y confirmación de apertura desde aplicación	88
Figura 3.29 Encendido/apagado de relé de apertura	88
Figura 3.30 Variable activa para apertura de cerradura	89
Figura 3.31 Clave usuario principal y huella correcta.....	89
Figura 3.32 Notificación de apertura con clave principal	90
Figura 3.33 Clave de emergencia y huella digital.....	90
Figura 3.34 Notificación de emergencia y apertura.....	91
Figura 3.35 Error en la digitación de la contraseña.....	91
Figura 3.36 Error en la huella digital.....	92
Figura 3.37 Permiso de acceso y autorización de nuevo registro.....	92
Figura 3.38 Selección de usuario para la base de huellas digitales.....	93
Figura 3.39 Captura de primera imagen de huella digital	93
Figura 3.40 Captura de segunda imagen y validación.....	94
Figura 3.41 Almacenamiento y confirmación de nuevo registro.....	94
Figura 3.42 Huella digital incorrecta para nuevo registro	95
Figura 3.43 Ingreso erróneo en huella digital.....	95

RESUMEN

Este trabajo presenta la implementación de un sistema de seguridad, el cual consta de un subsistema de control de acceso, así como, un sistema de videovigilancia a través de cámara y un subsistema de gestión de usuarios. Además, un subsistema que manejará autenticación de usuarios y conexión hacia la base de datos, la cual está gestionada en la nube y tiene relación directa con Google.

En el primer capítulo, se realizan las entrevistas, con las cuales se analizan las diferentes necesidades y requerimientos del sistema de seguridad. Además, se presentan las características de los dispositivos a ensamblar utilizando tecnología WiFi para su conexión. También, se presentan los programas con los que se desarrolla la programación de los dispositivos y la aplicación móvil.

En el capítulo dos, se expone el diseño de la aplicación móvil, es decir: los tipos de usuario, el envío/recepción de información a la base de datos desde la aplicación y desde los dispositivos del sistema de seguridad. También se indican, los cálculos realizados para la placa que se construye para los elementos de seguridad, así como, las funciones *backend* que se utilizarán para los propósitos de seguridad.

En el capítulo 3, se realizan las pruebas de funcionamiento del sistema de seguridad para evidenciar que responda a las necesidades, así como se presentan los resultados y el análisis correspondiente.

En el último capítulo, se dan a conocer las conclusiones del sistema de seguridad, así como las recomendaciones obtenidas del presente Trabajo de Titulación.

Como parte del Trabajo de Titulación se adjuntan los anexos que corresponden a las entrevistas, requerimientos y códigos del sistema de seguridad

PALABRAS CLAVE: Sistema de Seguridad, Control de Acceso, Android, Firebase, ESP32.

ABSTRACT

This project presents the security system's implementation, which consists of an access control subsystem, as well as a video surveillance system through a camera and a user management subsystem. In addition, a subsystem that will handle user authentication and connection to the database, which is managed in the cloud and has a direct relationship with Google.

In the first chapter, the interviews are carried out, with which the different needs and requirements of the security system are analyzed. In addition, the characteristics of the devices to be assembled using Wi-Fi technology for their connection are presented. Also, the programs with which the programming of the devices and the mobile application are developed are presented.

In chapter two, the design of the mobile application is exposed, that is: the types of users, the sending/receiving of information to the database from the application and from the security system devices. The calculations made for the board that is built for the security elements are also indicated, as well as the backend functions that will be used for security purposes.

In chapter 3, the security system performance tests are carried out to show that it responds to the needs, as well as the results and the corresponding analysis.

In the last chapter, the conclusions of the security system are disclosed, as well as the recommendations obtained from this Degree Project.

As part of the Titling Work, the annexes corresponding to the interviews, requirements and codes of the security system are attached.

KEYWORDS: Security System, Access Control, Android, Firebase, ESP32.

1. INTRODUCCIÓN

Debido a la gran inseguridad que se enfrenta día a día en todos los ambientes, tanto residencial como de oficina, los sistemas de seguridad han ido avanzando a la par con las nuevas tecnologías. Estas mejoras en los sistemas de seguridad se han tomado con base en las comunicaciones, así como también, se han adaptado a las redes que se manejan actualmente.

La integración de los sistemas de seguridad con las aplicaciones móviles y su acceso remoto gracias a Internet, han logrado mejorar el confort en los usuarios, de tal manera que la transmisión de video o alertas de seguridad sean posibles en tiempo real.

Es por esto que resulta relevante el desarrollo de un sistema de seguridad que utilice los avances en tecnologías inalámbricas con módulos ESP32, así como también, la gestión a través de aplicaciones móviles que son muy utilizadas por su fácil manejo y control permanente.

1.1 OBJETIVOS

El objetivo general de este Trabajo de Titulación es:

- Desarrollar un sistema de seguridad inalámbrico mediante el uso de una aplicación móvil y módulos ESP32.

Sus objetivos específicos son:

- Estudiar la funcionalidad de los elementos y herramientas que permitirán desarrollar el sistema.
- Diseñar los dispositivos con módulos inalámbricos y la aplicación de control.
- Implementar el sistema de seguridad y control de acceso junto con la aplicación diseñados.
- Validar el sistema en base a pruebas de funcionamiento.

1.2 ALCANCE

Este Trabajo de Titulación está orientado a la implementación de un sistema de seguridad a través de módulos ESP debido a su bajo consumo energético y las grandes posibilidades de envío de datos de forma inalámbrica que presentan los dispositivos. Tiene como finalidad implementar: Un Subsistema de Seguridad que utilice claves que permitan la apertura de una cerradura eléctrica, además de una cámara y sensores de movimiento

para la vigilancia del interior del lugar; y, un Subsistema de Gestión que permita autenticar a los usuarios del sistema, así como el envío y recepción de notificaciones .

El Sistema de seguridad inalámbrico tendrá las siguientes características:

- Detección de movimiento en el área.
- Control de apertura de puerta principal.
- Vigilancia interna mediante detección de movimiento y uso de cámara.
- Aplicación Android para gestión del sistema.
- Uso de base de datos para almacenamiento de información.

Para la detección, se utiliza un sensor de movimiento PIR HC-SR501 [1] que envía una señal digital al módulo ESP32 si existe movimiento, que emitirá hacia la base de datos de la plataforma de Firebase un cambio en la variable y activará la toma de una imagen del instante en que se ha generado esta variación. También, se enviará una notificación y una imagen cuando se haya tenido un ingreso de contraseña errada por más de 2 veces o si se ingresa la clave de emergencia.

Para el control de la apertura de la puerta se instala una cerradura eléctrica que permite abrir la puerta mediante una señal de control. Esta chapa es modificada mediante un módulo ESP32 que permite enviar datos de forma inalámbrica para la apertura. También, se realiza el trancado del mecanismo de apertura y la instalación de un microswitch para la detección del ingreso de la llave en la cerradura evitando que se pueda abrir la puerta mediante la misma, ya que solo se abrirá si se ingresa la información necesaria solicitada. Cuando no exista energía eléctrica se desactiva el trancado del mecanismo permitiendo la apertura de la puerta mediante la llave. Como último paso si se ingresa la llave y se trata de forzar la apertura de la puerta ésta envía un aviso de intrusión hacia la aplicación.

Para el control de acceso se incluye un teclado matricial 4x3 en la puerta principal para el ingreso de las claves asignadas a cada usuario, dicha información aparecerá en una pantalla LCD 16x2 y será enviada a la base de datos para su verificación.

Dentro de la gestión de usuarios se debe tomar en cuenta la base de datos de la plataforma Firebase ya que, se dispone de un CRUD (*Create, Read, Update and Delete*) de los usuarios que están autorizados; éstos son clasificados como: Administradores, quienes pueden modificar a los usuarios en lista, así como también agregar o eliminar a otros usuarios, y los usuarios invitados que pueden acceder al sistema de seguridad.

La Aplicación cuenta con varias Actividades, cada una representa una interfaz de usuario. En Android Studio se puede realizar la programación de la aplicación tanto en Java como en Kotlin. Se considera que Java es un lenguaje muy conocido, por lo que será este lenguaje con el que se procede a codificar las actividades en la aplicación.

En la plataforma de Firebase, la base de datos cargará las variables con los valores obtenidos, registrando nuevos nodos. Por otra parte, el servicio de autenticación registra a los usuarios dentro de la plataforma para que puedan acceder a la aplicación.

En video vigilancia, la configuración de la cámara a utilizar dentro del sistema de seguridad emplea un módulo ESP32-CAM, el cual permite una conexión entre la cámara y la aplicación sin necesidad de una IP pública fija.

1.3 MARCO TEÓRICO

1.3.1 SISTEMAS DE SEGURIDAD

Un sistema es una combinación de recursos que actúan conjuntamente para alcanzar un objetivo. Se encuentra estructurado de diversas partes y funciones propias que aportan al objetivo principal.

Se caracteriza por tomar las variables que ingresan, realizar un proceso de validación y verificar las respuestas obtenidas; por ejemplo, en un sistema de alarmas, las respuestas de los sensores de movimiento o la digitación de una clave errónea.

A través de la inclusión del Internet en el hogar con las denominadas TIC (Tecnologías de la Información y las Comunicaciones), se ha construido una nueva manera de comprender los usos y aplicaciones de la tecnología en la vivienda. Por lo tanto, la tecnología se vuelve transparente al usuario. Gracias a esto se ha conseguido obtener un *mercado horizontal*, es decir, donde exista una convergencia entre todas las áreas dentro del hogar.

El desarrollo de este nuevo mercado horizontal requiere garantizar una comunicación eficiente entre todos los equipos, los cuales son llamados “equipos comunicantes” [2].

1.3.2 SISTEMA OPERATIVO ANDROID

Para el desarrollo de la aplicación que maneja el Sistema Integrado se usa Android Studio IDE; por lo tanto, es necesario conocer algunas características de este entorno de desarrollo, así, por ejemplo, que es un sistema operativo de código abierto, que está basado en Linux y que se programa principalmente en Java.

Las numerosas actualizaciones en sus versiones han ofrecido nuevas características que van mejorando su desempeño.

1.3.2.1 Arquitectura de la Plataforma Android [3]

Android es una pila de software de código abierto creada para una gran gama de dispositivos. En la Figura 1.1, se muestran los componentes principales de la plataforma Android.

- **Kernel de Linux**

La plataforma Android está basada en el kernel de Linux. Por ejemplo, el Tiempo de Ejecución de Android (ART) se basa en el kernel de Linux para funcionalidades subyacentes (subproceso, memoria, etc.)

Android aprovecha las funciones de seguridad del kernel de Linux.

- **Capa de Abstracción de Hardware (HAL)**

Brinda interfaces estándares que constan de varias bibliotecas, de tal manera que cuando es necesario el trabajo de una API (*Application Programming Interface*), llamará a una interfaz específica para acceder al hardware del dispositivo y podrá cargar el módulo de biblioteca en cuestión.

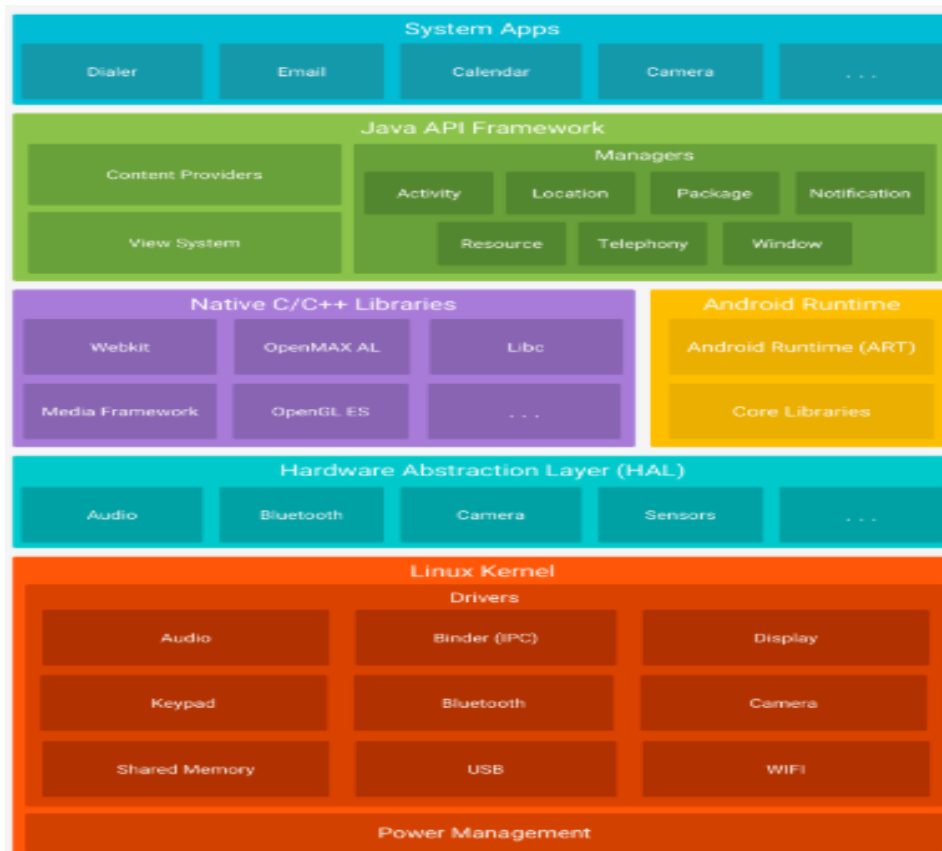


Figura 1.1 Componentes de la plataforma Android [3]

- **Tiempo de Ejecución de Android(ART)**

Cada aplicación ejecuta sus propios procesos e instancias del tiempo de ejecución de Android, éstos están escritos para ejecutar máquinas virtuales de memoria baja.

- **Bibliotecas C/C++ nativas**

Existen muchos componentes y servicios como ART y HAL, que requieren bibliotecas nativas en C/C++; la plataforma de Android provee un API de Java para entregar algunas funcionalidades de estas bibliotecas.

- **Framework de la API de Java**

Todas las funciones del SO Android está disponibles mediante las API de lenguaje Java. Estas API son los cimientos para crear aplicaciones en Android reutilizando los componentes del sistema.

- **Aplicaciones del sistema**

En Android se incluye un conjunto de aplicaciones centrales las cuales brindan capacidades claves a las que los desarrolladores pueden acceder desde sus propias aplicaciones.

1.3.2.2 Aspectos básicos de la Aplicación

1.3.2.2.1 Actividades

La aplicación móvil tiene diferencias notorias en comparación con las aplicaciones de escritorio, una de estas diferencias es la interacción con el usuario, ya que las aplicaciones móviles pueden comenzar de manera no determinista, es decir, que no siempre empiezan en el mismo lugar. Por ejemplo, si se abre la aplicación de contactos desde la pantalla principal estará predeterminada la lista de contactos. Pero, si se abre los contactos desde las redes sociales puede enviar directamente a la “*Activity*” de creación de un nuevo contacto.

La clase *Activity* está diseñada para facilitar la interacción entre las aplicaciones, por lo tanto, se llama a una actividad en lugar de la aplicación como un todo atómico. De esta manera, la actividad sirve como punto de entrada para la interacción de una aplicación con el usuario. Implementa una actividad como una subclase de la *Activityclass*.

La mayoría de las aplicaciones contienen múltiples pantallas, lo que significa que comprenden múltiples actividades. Al trabajar con varias actividades existe una relación

mínima de dependencia, ya que cada actividad puede trabajar en conjunto con las otras actividades [4].

Cada una de las *Activities* deben estar registradas en el manifiesto de la aplicación.

1.3.2.2.2 Ciclo de Vida de las Actividades

Para navegar transiciones entre etapas del ciclo de vida de la *Activity*, la clase de actividad muestra un conjunto básico de seis devoluciones de llamada: `onCreate()`, `onStart()`, `onResume()`, `onPause()`, `onStop()`, y `onDestroy()` [5].

La Figura 1.2 presenta una representación visual de este paradigma.

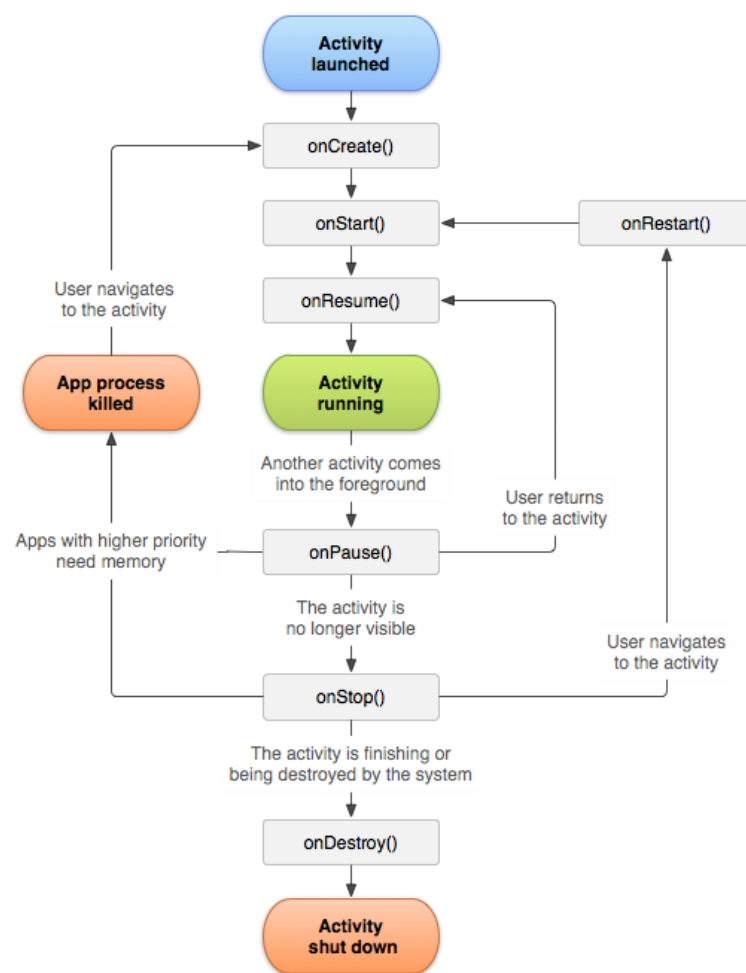


Figura 1.2 Una ilustración simplificada del ciclo de vida de la actividad [5]

La probabilidad de cerrar un proceso, junto con las *Activities* en él, depende del estado de la actividad en ese momento. El estado de actividad y la expulsión de la memoria proporcionan más información sobre la relación entre el estado y la vulnerabilidad

a la expulsión. Dependiendo de su actividad, es probable que no necesite implementar todos los métodos del ciclo de vida.

1.3.2.2.3 Servicios

Un servicio es un punto de entrada que sirve para ejecutar la aplicación en segundo plano por diversos motivos. Es un componente para realizar operaciones de ejecución prolongada o para realizar tareas de procesos remotos. Un servicio no proporciona una interfaz de usuario. Una actividad, puede iniciar el servicio y permitir que se ejecute o enlazarse a él para interactuar [6].

1.3.2.2.4 Receptores de emisiones

Un receptor de emisiones es un componente bien definido de entrada en la aplicación que permite que el sistema entregue eventos fuera de un flujo de usuarios habitual, permitiendo que se responda a los anuncios de emisión de todo el sistema. Los receptores pueden entregar emisiones incluso a las aplicaciones que no están disponibles en la ejecución.

Si bien los receptores de emisión pueden crear una notificación de la barra de estado para alertar al usuario cuando se produce algún evento, por lo general, un receptor de emisión es una *puerta de enlace* a otros componentes [6].

1.3.2.2.5 Proveedores de contenido

Un proveedor de contenido sirve para administrar datos almacenados que pueden ser compartidos de la aplicación, ya sea en una base de datos SQLite, en la Web o en cualquier otra ubicación de almacenamiento persistente a la que tenga acceso. A través del proveedor de contenido, otras aplicaciones pueden consultar o modificar los datos .

Los proveedores de contenido se activan a través de solicitudes de un ContentResolver, el cual aborda todas las transacciones directas con el proveedor de contenido [6].

1.3.2.2.6 Activación de componentes

Un aspecto exclusivo del diseño del sistema Android es que cualquier aplicación puede iniciar un componente de otra aplicación, es decir, que los componentes de las aplicaciones son reusables. Dentro de los componentes que han sido especificados, tres de ellos se generan tras la creación de una intención; ésta se manifiesta a través de un objeto *Intent* que activa el componente tras su ejecución. Para las actividades y los servicios, el objeto *Intent* se lo define como una acción a realizar y en donde se pueden especificar los parámetros o datos que se enviarán una vez que sea iniciada [6].

1.3.2.2.7 El Archivo de Manifiesto (Manifest)

Para iniciar un componente de la aplicación en el sistema Android, debe reconocer la existencia de ese componente leyendo el Archivo de Manifiesto de la aplicación (AndroidManifest.xml). Se debe declarar todos sus componentes en este archivo.

El Archivo de Manifiesto puede realizar ciertas acciones adicionales además de declarar los componentes de la aplicación, por ejemplo [7]:

- Identificar los permisos que requiere la aplicación.
- Declarar el nivel de API mínimo que requiere la aplicación en función de las API que usa.
- Declarar características de hardware y software.
- Declarar bibliotecas de la API a la que la aplicación necesita estar vinculada (además de las API del marco de trabajo de Android).

1.3.3 FIREBASE

Firebase es una plataforma desarrollada por Google que tiene varios servicios incorporados y facilita el desarrollo de aplicaciones. Por lo tanto, se puede comprender la importancia de explicar sus componentes y cómo va a ser útil dentro del desarrollo del sistema propuesto.

Firebase entrega una solución efectiva frente a problemas de desarrollo, disponibilidad y confiabilidad; con base en el crecimiento de la base de usuarios y dispositivos, ya que la infraestructura está dentro de Google [8].

Entre los servicios que dispone la plataforma Firebase se encuentran: autenticación, bases de datos, almacenamiento de datos, funciones *backend*, solución de errores, priorizar problemas de estabilidad, rendimiento y medidas estadísticas [9]. En el sistema de seguridad propuesto no se van a usar todas las funcionalidades de la plataforma, por lo tanto, se detallan a continuación los servicios que son utilizados:

- Base de Datos: Firebase posee una base de datos NoSQL que almacena y sincroniza los datos de acuerdo con la disponibilidad de la red, teniendo una estimación a tiempo real. Ésta ha sido utilizada para almacenar todos los datos de los usuarios que tendrán acceso al sistema integrado, así como, para el almacenamiento de la información de los dispositivos [10].
- Autenticación: Firebase provee métodos de registro e inicio de sesión a través de Google y también de algunos proveedores externos como: Facebook, Twitter,

Github e incluso a través de entidades anónimas para poder controlar los permisos que se pueden otorgar [11].

- Funciones *Backend*: Firebase provee un sistema de funciones de *backend*, es muy útil dentro del desarrollo del sistema, ya que a través de estas funciones se van a generar o actualizar la programación de apertura/cierre del sistema de seguridad, así como, el envío de notificaciones al ingresar al lugar [12].

Se profundizará cada uno de los aspectos en los capítulos donde se detalla su funcionalidad.

1.3.4 ARDUINO IDE

El Arduino *Integrated Development Environment*, o Arduino Software (IDE), contiene un editor de texto para escribir código, un área de mensajes, una consola de texto, una barra de herramientas con botones para funciones comunes y una serie de menús. Se conecta al hardware Arduino y Genuino para cargar programas y comunicarse con ellos [13].

Estos programas son denominados *bocetos*, los cuales se escriben en el editor de texto y se guardan con una extensión “.ino”. La consola muestra la salida del texto del software Arduino, incluyendo mensajes de error completos y de información.

El lenguaje de programación Arduino se divide en tres partes principales:

- Funciones: Para controlar la placa y realizar cálculos,
- Variables: Tipos de datos y constantes
- Estructura: Los elementos del código Arduino.

Una vez determinadas las tres partes principales se procede a compilar el código.

El entorno de Arduino se puede ampliar mediante el uso de bibliotecas, similar a la mayoría de las plataformas de programación. Estas bibliotecas añaden funcionalidades adicionales para su uso en bocetos; por ejemplo, el uso de nuevos hardware o programas que manipulen datos desde otras plataformas.

Varias bibliotecas vienen instaladas en el entorno IDE, pero también se pueden descargar o crear propias.

Las bibliotecas estándar son [14]:

- EEPROM : lectura y escritura en almacenamiento "permanente".

- Firmata : para comunicarse con aplicaciones en la computadora mediante un protocolo serie estándar.
- Liquid Crystal Display - para controlar pantallas de cristal líquido (LCD).
- Servo - para controlar servomotores.
- Software Serial: para comunicación en serie en cualquier pin digital. La versión 1.0 y posteriores de Arduino incorporan la biblioteca NewSoftSerial.
- WiFi : para conectarse a Internet utilizando el escudo Arduino WiFi.
- Cable : interfaz de dos cables (TWI / I2C) para enviar y recibir datos a través de una red de dispositivos o sensores.
- Las bibliotecas Matrix, WiFiManager y FirebaseESP32Client ya no son parte de la distribución principal.

1.3.5 MÓDULO ESP32

El módulo ESP es un chip diseñado con características Bluetooth y WiFi 2.4 GHz con tecnología de ultra bajo consumo de energía. Está diseñado para mejorar el rendimiento, la confiabilidad, robustez, versatilidad y compatibilidad con una gran cantidad de aplicaciones [15].

Estos microcontroladores fueron integrados a placas de desarrollo las cuales poseen elementos electrónicos para el uso de los puertos de comunicación, de alimentación de voltaje y puertos de entrada y salida. Al implementar estos microcontroladores en las placas de desarrollo tomaron el nombre de Node MCU, permitiendo un mejor manejo y programación de los dispositivos.

Las placas de desarrollo más utilizadas en el mercado son los Node MCU ESP8266 y Node MCU ESP32, las cuales permiten utilizar una interfaz de desarrollo como Arduino IDE para su programación y configuración. Además, esta interfaz provee diferentes librerías *Open Source* que permiten la conexión con Firebase para el envío y recepción de datos desde un servidor Web.

El Node MCU ESP32 (ver Figura 1.3) es un potente módulo que integra la comunicación Bluetooth y WiFi en la misma placa de desarrollo, permitiendo crear dispositivos electrónicos con comunicación inalámbrica y de bajo costo. El uso de WiFi permite una comunicación de mediano alcance a través de una LAN mediante un *router* con conexión a Internet, mientras que la conexión Bluetooth permite la transmisión de datos a corta distancia entre dispositivos.



Figura 1.3 Node MCU32 [16]

1.3.5.1 Especificaciones Técnicas del Módulo ESP32

Entre las especificaciones del Node MCU ESP32 se puede destacar como la más importante su reducido tamaño, que es aproximadamente de 7 x 4 x 1 cm, lo cual permite crear dispositivos pequeños y que puedan ser adaptables a los sistemas eléctricos ya instalados dentro de los hogares.

Otra característica importante es la comunicación WiFi que permite el acceso a Internet del dispositivo y el envío y recepción de información hacia el exterior.

A continuación, se enlistan las principales características del Node MCU ESP32 [16]:

- Voltaje de Alimentación: 3.3V DC (2.7 ~ 3.6 v)
- Corriente de Operación: ~ 80 mA(fuente superior a 500 mA)
- Voltaje lógico (Entradas/Salidas): 3.3 V
- CPU: Doble núcleo Tensilica LX6 (32 bit)
- Frecuencia de Reloj: 240 MHz
- SRAM: 520 KB
- Pines digitales GPIO: 34(incluyendo todos los periféricos)
- UART: 2
- SPI:3
- I2C:2
- WiFi, Protocolo 802.11 b/g/n/e/i (802.11n carga hasta 150 Mbps)
- WiFi mode Station/SoftAP/SoftAP+Station/P2P
- WiFi Seguridad: WPA/WPA2/WPA2-Enterprise/WPS
- Protocolos de Red IPv4, IPv6, SSL, TCP/UDP/HTTP/FTP/MQTT
- Pila de Protocolo TCP/IP integrado.

En la Figura 1.4 se muestra la distribución de los diferentes pines de conexión de Node MCU32.

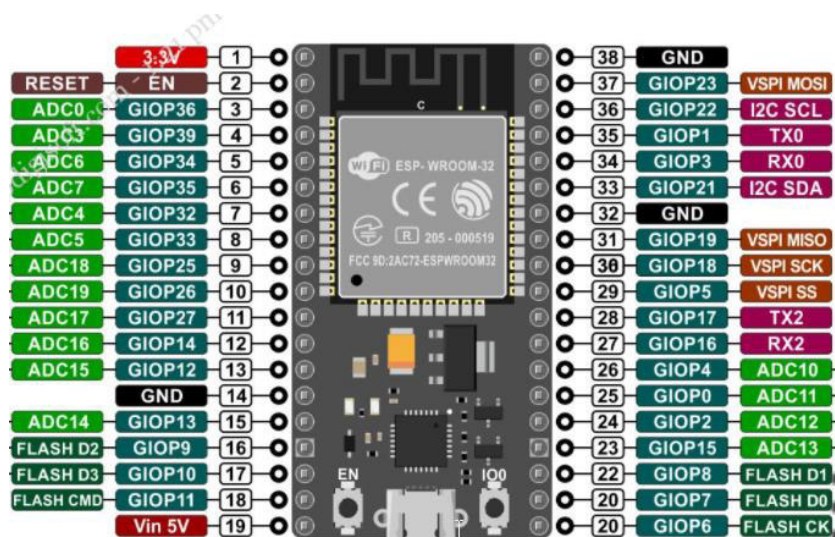


Figura 1.4 Node MCU32 distribución de pines [17]

1.3.5.1.1 Ventajas del Módulo ESP32

Una de las ventajas del ESP32 es que permite una conexión directa mediante WiFi a un *router* sin necesidad de otro elemento electrónico. Esta característica no tiene ningún otro microcontrolador en el mercado y que permite el desarrollo de dispositivos muy sofisticados que transmitan datos hacia la web.

Otra ventaja importante que provee este módulo es que puede ser programado en la plataforma Arduino IDE, la cual permite la utilización de librerías para sensores, pantallas, Firebase, etc., obteniendo una mayor flexibilidad para el desarrollo de aplicaciones.

1.3.5.2 ESP32-CAM

Se basa en el microcontrolador ESP32-S, incorpora una cámara y un led flash, con esto permite realizar el *streaming* de video, tomar fotos, aplicaciones de reconocimiento facial, entre otros.

Contiene todas las características de un módulo ESP32. Gracias a su formato DIP permite su fácil integración con cualquier aplicación y su montaje en protoboard es muy sencillo. Cabe recalcar que a mayor resolución la capacidad de cuadros por segundo transmitidos va a ser menor.

Para el ESP32-CAM es necesario un conversor USB-serial externo para probar las funcionalidades y cargar los archivos desde Arduino IDE. Puede alimentarse a 5 V o 3 V (ver Figura 1.5), pero es útil conectar capacitores para filtrar los picos de corriente. Los pines de entrada y salida trabajan con un voltaje de 3.3 V, por lo tanto, es necesario usar conversores a este voltaje [16].



Figura 1.5 ESP32-CAM Distribución pines [18]

1.3.6 SENSORES

Son dispositivos que transforman una variable física o química en una tensión o corriente eléctrica. Los sensores en las aplicaciones permiten analizar el lugar y de esta manera controlar los dispositivos para proveer al usuario el ambiente adecuado para sus actividades.

Los sensores existen en una gran variedad dependiendo de la variable física o química a medir, a continuación, se detalla los sensores que se están utilizando en el sistema de seguridad.

1.3.6.1 Sensor de movimiento PIR HC-SR501

Los sensores infrarrojos pasivos (PIR) son dispositivos que detectan el movimiento mediante la medición de radiación infrarroja de los cuerpos, la cual depende de su temperatura. A mayor temperatura los cuerpos emiten mayor radiación infrarroja [19].

Estos sensores se los utiliza comúnmente en la detección de objetos dentro de un lugar, ya que posee dos campos de detección que miden constantemente la radiación infrarroja. En condiciones normales, estos sensores reciben la misma radiación infrarroja en los dos campos por lo que existe un equilibrio; cuando pasa un cuerpo por el perímetro del sensor se provoca un desequilibrio en los campos de detección alertando de un movimiento dentro de su campo de visión.

A continuación, se indican las características técnicas del sensor PIR HC-SR501 [1] :

- Sensor piroeléctrico (Pasivo) infrarrojo (también llamado PIR)
- El módulo incluye el sensor, lente, controlador PIR BISS0001, regulador y todos los componentes de apoyo para una fácil utilización

- Rango de detección: 3 m a 7 m, ajustable mediante *trimmer* (Sx)
- Lente fresnel de 19 zonas, ángulo < 100°
- Salida activa alta a 3.3 V
- Tiempo en estado activo de la salida configurable mediante *trimmer* (Tx)
- Redisparo configurable mediante *jumper* de soldadura
- Consumo de corriente en reposo: < 50 μ A
- Voltaje de alimentación: 4.5 VDC a 20 VDC

En la Figura 1.6 se muestra el sensor PIR a utilizar en el sistema de seguridad.



Figura 1.6 Sensor PIR HC-SR501 [1]

1.3.6.2 Sensor Biométrico

Los sensores de huella o sensores biométricos (ver Figura 1.7) son capaces de identificar, leer y guardar huellas dactilares. Las huellas dactilares son únicas, de tal manera que así sean hermanos gemelos, las huellas van a ser únicas para cada una de las personas.

Los sensores biométricos en los últimos años se han hecho muy populares debido al uso en los teléfonos inteligentes. Debido a la gran cantidad de contraseñas que se usan hoy en día para poder guardar nuestra información, el uso de las huellas es una alternativa para obtener un nivel de seguridad más alto, ya que tiene un grado de dificultad bastante alto para poder ser falsificado.

Arduino decidió actualizarse con el uso de los sensores biométricos para aprovechar este sistema de autenticación a través de un módulo de reconocimiento y almacenamiento de huellas digitales. El módulo se comunica a mediante un puerto serial y necesita que la huella sea almacenada para poder ser comprobada [20].



Figura 1.7 Sensor Biométrico [20]

Como se observa en la Figura 1.8 su forma de conexión es bastante simple, son dos cables para la alimentación y dos cables para la conexión serie. Se puede conectar tanto de forma Software Serial, para no ocupar el puerto serial del dispositivo físico, así como también, se puede conectar de forma Hardware Serial, ocupando de esta manera el puerto serie físico del dispositivo.

A través de la librería Adafruit que se encuentra disponible en el IDE de Arduino la programación del sensor biométrico se simplifica. La librería se puede descargar e instalar en el IDE para su uso ya que no se encuentra preinstalada [20].



Figura 1.8 Esquema de conexión Sensor biométrico [20]

1.3.7 METODOLOGÍA DE DESARROLLO KANBAN

Kanban es una metodología de organización e información visual del progreso de cada proyecto, fue desarrollada a finales de la década de 1940 por la compañía Toyota para optimizar sus procesos. En el desarrollo de software del sistema de seguridad se puede aprovechar los principios de desarrollo ágil para hacer coincidir el trabajo con el equipo. Es un proceso de gestión que simplifica el desarrollo de productos para cumplir con las necesidades [21].

Todos los trabajos giran en torno al “tablero Kanban”, el cual se utiliza para visualizar el trabajo, de tal manera que el flujo esté estandarizado haciendo que los bloques y las dependencias se resuelvan lo antes posible. Como se puede observar en la Figura 1.9 Kanban trabaja con un esquema de 3 pasos, Pendiente, En curso y Hecho, aunque dependiendo del administrador se pueden añadir más columnas en el tablero.

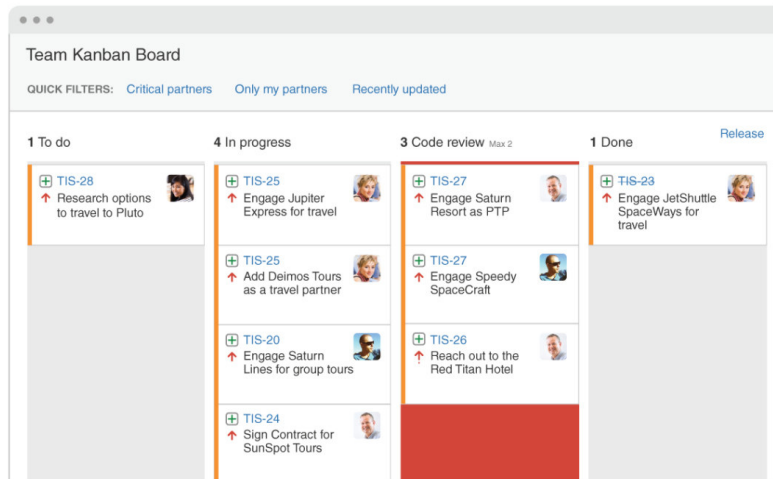


Figura 1.9 Representación de tablero Kanban [21]

Kanban basa su metodología en la transparencia y la comunicación, es decir, el tablero se considera como la única fuente de verdad para el trabajo del equipo. Kanban ofrece varias ventajas para la planificación y rendimiento como: flexibilidad, es decir, completar los trabajos de la lista de pendientes, pasando a otra, pero si es necesario regresar a una tarea en específico, teniendo el administrador la libertad de hacerlo. Posee ciclos de tiempo más cortos, gracias a la superposición de habilidades del grupo de trabajo y revisión de código a través de prácticas básicas para difundir el conocimiento. Entrega e integración continua, son prácticas combinadas frecuentemente que crean y prueban el código a lo largo del tiempo mientras se entrega el trabajo.

2. METODOLOGÍA

En este capítulo se presentan las dos fases: Diseño e Implementación del sistema de seguridad inalámbrico. Dentro del Diseño se definirá los requerimientos del proyecto, la estructura y comportamiento del sistema, así como también, la organización del trabajo por medio del sistema Kanban. Luego, se presentará la fase de Implementación, siguiendo la secuencia utilizada en el sistema Kanban para los componentes del sistema, así como, la implementación de los dispositivos electrónicos y la codificación del diseño.

2.1 DISEÑO DEL SISTEMA

Presenta el análisis de requerimientos del sistema, luego para el diseño de software se definen los requerimientos funcionales y no funcionales, las historias de usuario, los casos de uso, diagramas de secuencia, arquitectura del desarrollo y la organización del proyecto a través de la herramienta Kanban. Dentro de esta organización también se define el diseño electrónico en el software ISIS y el diseño de las PCB (*Printed Circuit Board*) en el software ARES de todos los dispositivos de seguridad implementados que son: cerradura eléctrica y cámara.

2.1.1 REQUERIMIENTOS DEL SISTEMA

Los requerimientos del sistema se realizan con base a entrevistas enviadas a los usuarios. El formato de la entrevista se encuentra en el Anexo A.

La primera pregunta es informativa, tiene como objetivo aclarar el uso de los sistemas de seguridad y qué tan importante considera el usuario su uso. Sus respuestas se muestran en la Figura 2.1.

La pregunta 2 busca conocer la aceptación de la administración remota desde una aplicación móvil. En la Figura 2.2 se presentan las respuestas a esta pregunta.

La pregunta 3 busca definir el método seguro de ingreso del usuario a la aplicación. Como se muestra en la Figura 2.3 la respuesta con mayor porcentaje es correo y contraseña.

La pregunta 4 tiene como objetivo modelar la clase usuario. Por lo tanto, con base en las respuestas, la entidad usuario contará con los atributos mencionados en la Tabla 2.1 y en la Figura 2.4.

La pregunta 5 menciona las acciones que deben ser realizadas por los diferentes usuarios, así se puede clasificar la interacción de los usuarios con la aplicación. En la pregunta 6 se observa el tipo de información que se puede entregar a cada uno. Se muestran los resultados a estas preguntas en las Figuras 2.5 y 2.6.

Cree que los sistemas de seguridad son indispensables?
16 respuestas

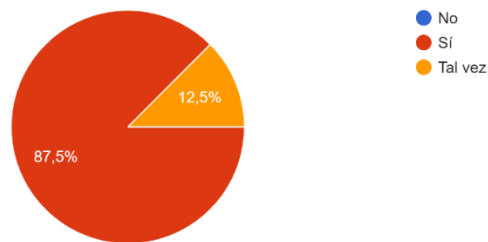


Figura 2.1 Resultados a la pregunta 1 de la Entrevista

Considera que es necesario el uso de una aplicación para el control y manejo de un sistema de seguridad
16 respuestas

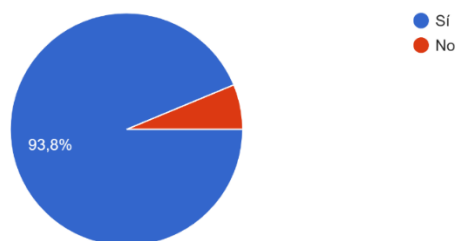


Figura 2.2 Resultados a la pregunta 2 de la Entrevista

En las aplicaciones móviles se hace uso de usuario y contraseña. De las siguientes opciones ¿Cuál preferiría como acceso?
16 respuestas

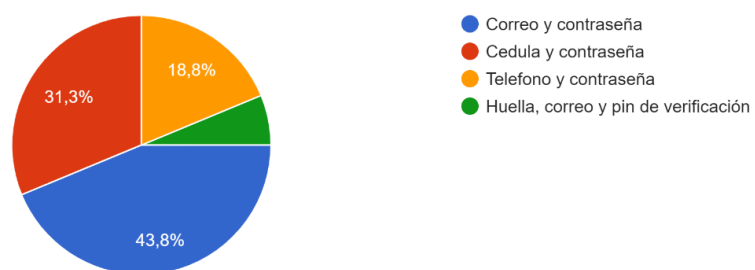


Figura 2.3 Resultados a la pregunta 3 de la Entrevista

¿ Qué información del usuario considera que se debe guardar en la aplicación?

16 respuestas

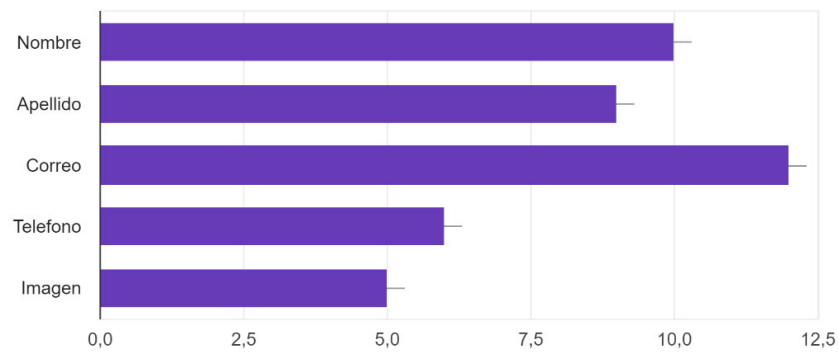


Figura 2.4 Resultados a la pregunta 4 de la Entrevista

¿Qué acciones considera que deben ser manejados en la aplicación?

16 respuestas

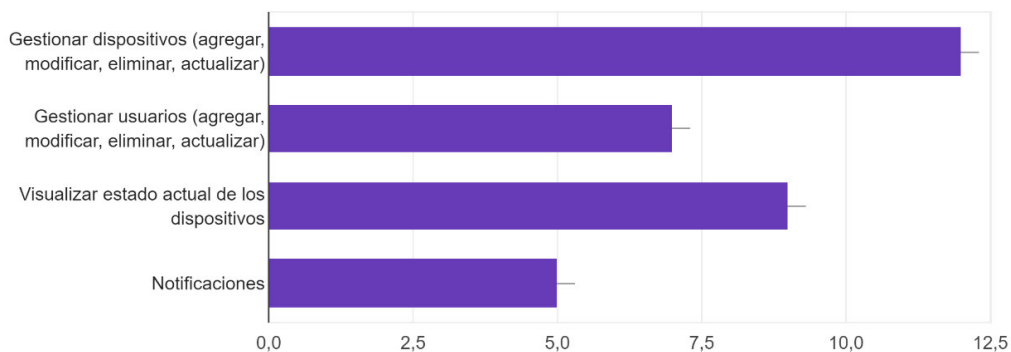


Figura 2.5 Resultados a la pregunta 5 de la Entrevista

La pregunta 7 es de carácter informativo, de esta manera en la Figura 2.7 se puede observar la importancia para los usuarios de una marca conocida y la aceptación a nuevas marcas del mercado.

La pregunta 8 define los tipos de dispositivos, ya que es necesario conocer la aceptación a tecnologías inalámbricas y cableadas. Adicionalmente, en la pregunta 9 se busca definir el tipo de seguridad que los clientes consideran de mayor utilidad al momento de colocar un sistema de seguridad en el ingreso principal. Se muestran los resultados a estas preguntas en las Figuras 2.8 y 2.9 respectivamente.

Tabla 2.1 Requisitos de datos de Usuario

Requisitos de datos: RD01: Usuario	
Identificador: RD01 Nombre: Usuario	Versión: 2
Tipo: Datos a mantener	
Datos específicos para mantener: id, nombre, apellido, url de foto de perfil, contraseña ingreso Puerta principal, huella biométrica, e-mail, token de notificaciones.	
Detalles: <ul style="list-style-type: none"> • Id: Cuentas vinculadas con correos electrónicos. • Url foto de perfil: Referencia a Firebase. • Contraseña ingreso puerta principal: clave generada para cada usuario para ingreso a la asociación. • Huella biométrica: huella digital de cada usuario. • Token de notificaciones: Identificador de Firebase para envío de notificaciones. 	

Dentro de la aplicación. ¿Qué información considera que se debe mostrar de los dispositivos?
16 respuestas

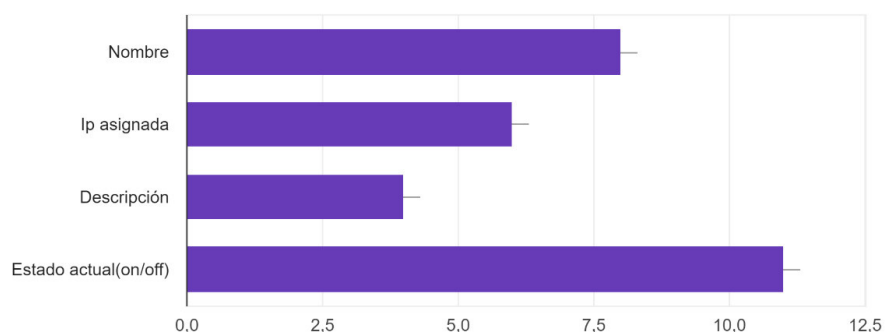


Figura 2.6 Resultados a la pregunta 6 de la Entrevista

¿Es importante la marca de los dispositivos a usar?
16 respuestas

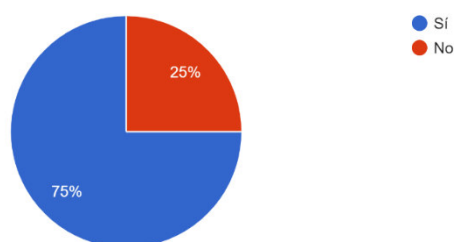


Figura 2.7 Resultados a la pregunta 7 de la Entrevista

Considera que los dispositivos deben ser:
16 respuestas

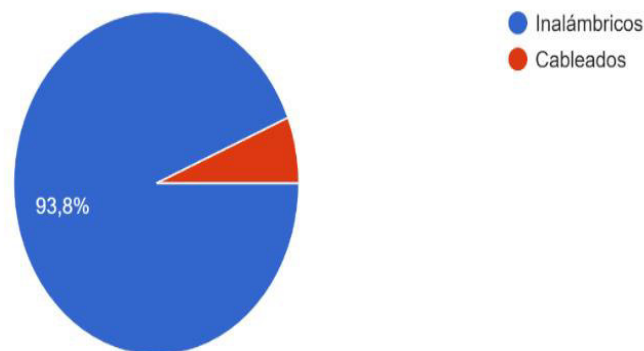


Figura 2.8 Resultados a la pregunta 8 de la Entrevista

En el caso de un control de puerta. ¿Cuál de las siguientes opciones considera mas seguro?
16 respuestas

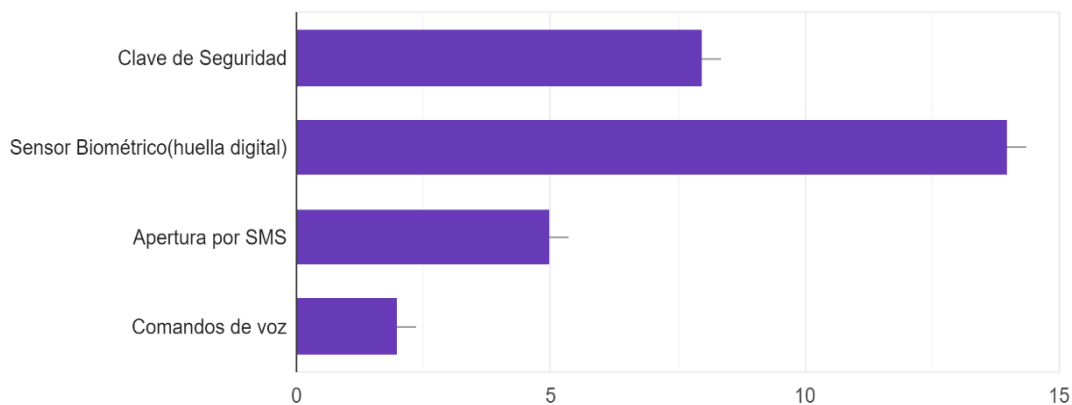


Figura 2.9 Resultados a la pregunta 9 de la Entrevista

Las preguntas 10 y 11 busca definir si es necesario ingresar la huella digital de cada persona en el hogar, así como también, una clave única personal por cada usuario dentro del sistema de seguridad o se permita una sola huella y clave por motivos de privacidad. Se muestran los resultados a estas preguntas en las Figuras 2.10 y 2.11.

En la pregunta 12 se define si los usuarios requieren respaldos en las cámaras de seguridad durante un evento de alerta o de ser el caso, un proceso continuo de grabado durante las 24 horas del día.

En el caso de uso de sensor biométrico. Considera que todas las personas dentro del hogar deben poder registrar su huella?

16 respuestas

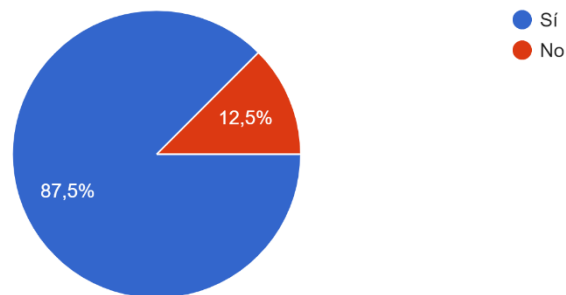


Figura 2.10 Resultados a la pregunta 10 de la Entrevista

El en caso de uso de clave. Considera que debe ser una clave por cada miembro de la familia?

16 respuestas

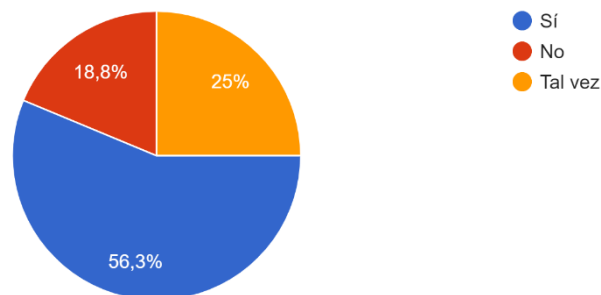


Figura 2.11 Resultados a la pregunta 11 de la Entrevista

¿Considera que las cámaras deben grabar solo los eventos que presenten alertas?

16 respuestas

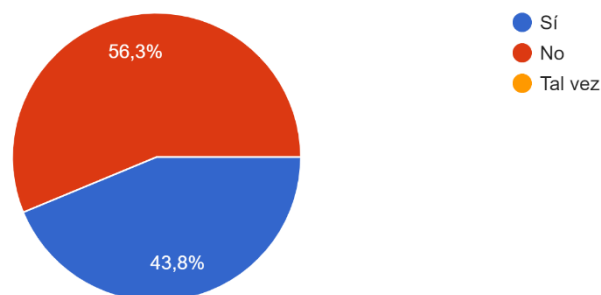


Figura 2.12 Resultados a la pregunta 12 de la Entrevista

Tabla 2.2 Requisito de datos de Dispositivo

Requisitos de datos: RD01: Dispositivo	
Identificador: RD01 Nombre: Dispositivo	Versión: 2
Tipo: Datos a mantener	
Datos específicos para mantener: id, nombre, descripción y estado actual.	
Detalles: <ul style="list-style-type: none">• Id: Número único del dispositivo• Nombre: Nombre del equipo• Descripción: Resumen del dispositivo• Estado actual: Indica si el dispositivo está encendido/apagado	

2.1.2 CASOS DE USO

Para describir los requisitos funcionales del sistema de seguridad se utilizan Casos de Uso. Un Caso de Uso representa una acción o describe las actividades que va a llevar a cabo algún proceso sin explicar el cómo los realiza. Permite describir los pasos que sigue el usuario para cumplir con un objetivo.

En la Figura 2.13 se pueden observar los Casos de Uso de las principales funciones de la sección de gestión de usuarios por parte del Administrador; se observan los Casos de Uso de las funciones de la sección de control de acceso, así como de la sección de seguridad.

La flecha representa una relación de inclusión que indica una relación de dependencia entre los dos escenarios y cómo denota la secuencia de eventos que genera un caso en el otro.

A partir de los casos de uso expuestos, se generan las tablas de Casos de Uso, las cuales detallan los requisitos funcionales del sistema integrado.

Para describir los Casos de Uso se incluyen los siguientes apartados:

- Id: Identificador del Caso de Uso. Su nomenclatura conlleva las letras CU seguidas de la numeración.
- Nombre: Indica el enfoque del Caso de Uso.
- Actor: Especifica un rol que una entidad externa toma cuando interactúa con el sistema integrado.
- Prioridad: Indica el nivel de importancia del Caso de Uso, puede ser: alta, media y baja.

- Descripción: Resumen del objetivo del Caso de Uso. No define el cómo, solo indica el qué hace.
- Flujo normal: Describe los pasos que van a seguirse en un contexto "ideal".
- Flujo alternativo: Son flujos independientes, describen desviaciones del flujo normal.

En la Tabla 2.3 se muestra un ejemplo de Caso de Uso para el ingreso de un nuevo usuario.

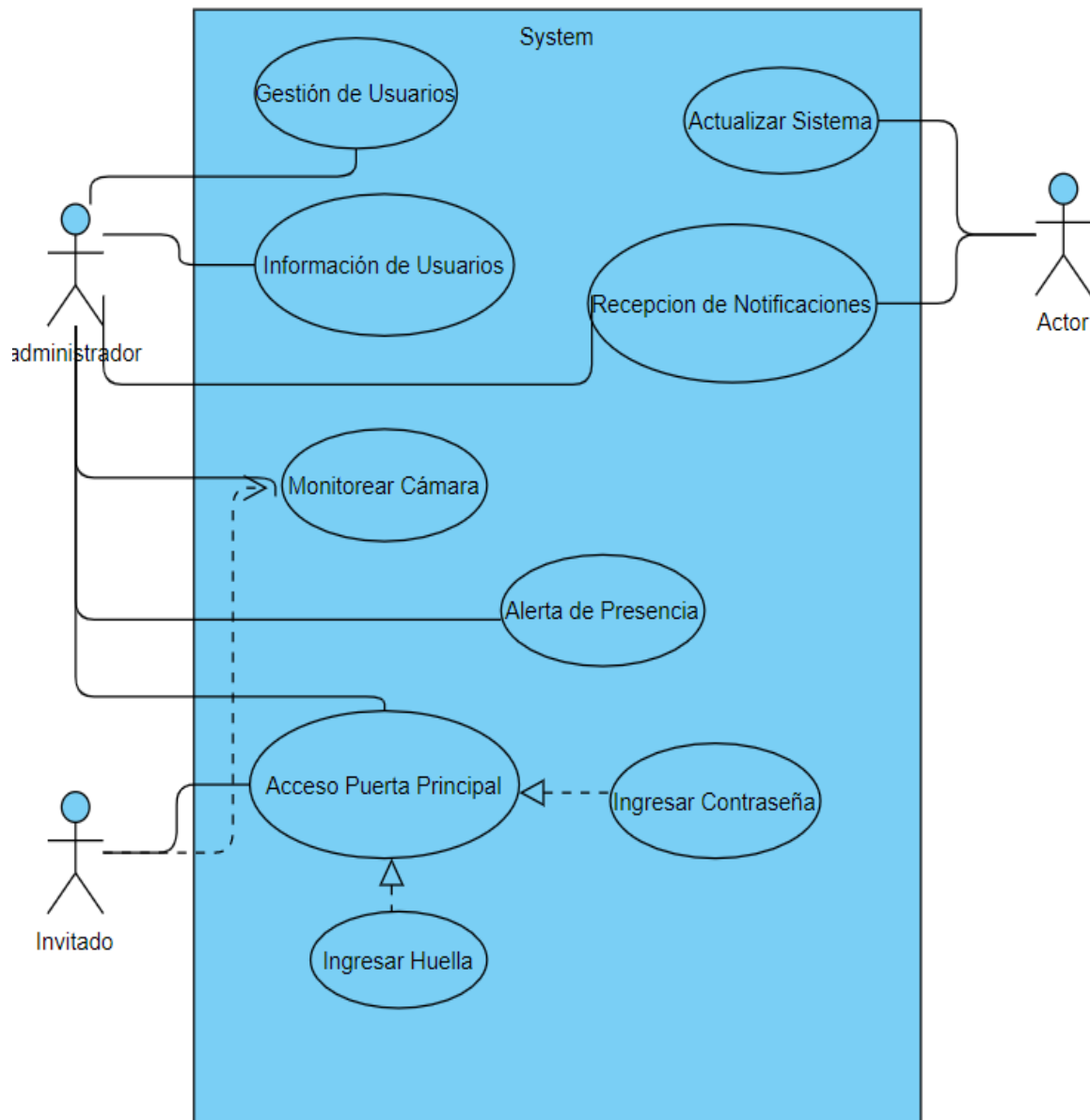


Figura 2.13 Diagrama de Caso de Uso del Sistema

Tabla 2.3 Caso de Uso Nuevo Usuario

Id: CU1	Nombre: Ingresar nuevo usuario al sistema
Actor: Usuario Administrador	Prioridad: Alta
Descripción: El usuario administrador ingresa un nuevo usuario a la base de datos a través de la aplicación móvil para que pueda tener acceso al sistema.	
Flujo Normal: 1.- El usuario administrador ingresa al sistema y se muestra los cuadros de acceso. 2.- El usuario administrador suministra los datos a la aplicación del nuevo usuario, dando permiso o no de usuario administrador. 3.- Se envía la información a la base de datos de Firebase y se actualiza.	
Flujo Alternativo: 1. El proceso de ingreso de nuevo usuario falla por falta de conexión entre la aplicación y la base de datos.	

Los Casos de Uso del sistema desarrollado, debido a su extensión, se adjuntan en el Anexo C.

2.1.3 REQUERIMIENTOS DE HARDWARE

El Sistema de Seguridad además de sus requerimientos de software tiene requerimientos de hardware. Estos requerimientos han sido recolectados a través de la validación de sistemas de seguridad. Adicionalmente, en la Tabla 2.4, se muestran las características de corriente y voltaje de los dispositivos para su ensamblaje.

Tabla 2.4 Componentes de sistema hardware

Dispositivo	Cantidad	Características eléctricas	Aplicación
Detector de movimiento	1	Voltaje: 120 VAC Corriente: 10 A	Permite la detección de movimiento hasta en un rango de 5m
Cerradura eléctrica	1	Voltaje: 12 VDC Sensor de apertura	Permite la apertura y cierre de la puerta principal
Teclado	1	Voltaje: 12 VDC	Permite el ingreso de la clave y envío hacia base de datos
ESP32-CAM	1	Voltaje: 5 V	Permite la visualización de los eventos dentro del lugar
ESP32	3	Voltaje: 5 V	Permite la conexión de los dispositivos con la plataforma Firebase

2.1.4 REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

Los requerimientos funcionales describen el comportamiento del sistema a implementar, es decir, definen cuáles son los servicios o funcionalidades que deberán incluirse en el desarrollo del sistema. Por lo tanto, se definieron los siguientes:

- Iniciar sesión
- Al momento de ingresar correo electrónico y contraseña se debe validar que sea correcta.
- Los mensajes de notificación por campos vacíos se realizan a través de Toast.
- Mostrar un mensaje al tener una autenticación fallida.
- El usuario administrador podrá crear, modificar y eliminar a otros usuarios invitados dentro del sistema.
- El usuario administrador podrá añadir, modificar y eliminar dispositivos que se encuentren dentro del sistema integrado
- El usuario podrá visualizar la cámara de seguridad en la aplicación desarrollada.
- El usuario deberá configurar su perfil en la aplicación.
- El usuario deberá abrir la puerta principal a través de la aplicación o de manera manual.
- Para el ingreso de claves a través de la aplicación o teclado digital serán validados para que sean solo números.
- El usuario deberá recibir notificaciones.

Los requerimientos no funcionales son aquellos que definen las propiedades del sistema: rendimiento, seguridad, disponibilidad, es decir, el “cómo” se cumplirá el funcionamiento del sistema integrado. Se definieron:

- Cerrar sesión
- Las notificaciones para acceso a la puerta y sensor de presencia serán enviadas a través del sistema backend de Firebase.
- El tiempo de respuesta debe ser casi inmediato.

2.1.5 HISTORIA DE USUARIO

Las Historias de Usuario son utilizadas para describir lo que el usuario es capaz de hacer. Son herramientas utilizadas para especificar las funcionalidades del Sistema y describen los requerimientos del cliente a un nivel más fino. Para realizar las Historias de Usuario se tomará el formato especificado en la Tabla 2.5.

Tabla 2.5 Formato de historia de usuario

HISTORIA DE USUARIO		
ID:	Usuario:	
Programador Responsable:		
Nombre Historia:	Tipo:	
Descripción:		
Comentario:		

Los apartados de la tabla 2.4 son:

- ID: identificador de la Historia de Usuario.
- Usuario: Indica el usuario que realizará las funcionalidades.
- Programador Responsable: nombre de la persona encargada de programar o realizar esta Historia de Usuario.
- Nombre de historia: Nombre o título que describe la funcionalidad.
- Tipo: Indica si es un requerimiento funcional o no funcional.
- Descripción: se describe de manera concisa la funcionalidad.
- Comentario: información adicional y detallada de la descripción.

Tabla 2.6 Historia de usuario requerimiento funcional

HISTORIA DE USUARIO		
ID:HU 005	Usuario: administrador	
Programador Responsable: Juan Tapia		
Nombre Historia: Eliminar usuario	Tipo: Requerimiento Funcional	
Descripción: El usuario administrador podrá eliminar un usuario del sistema.		
Comentario: El usuario administrador podrá eliminar un usuario invitado del sistema.		

En la Tabla 2.6 se muestra un ejemplo de Historia de Usuario desglosada a partir de los requerimientos funcionales detallados; en la Tabla 2.7 se presenta un ejemplo de historia de usuario de un requerimiento no funcional.

Tabla 2.7 Historia de usuario con requerimiento no funcional

HISTORIA DE USUARIO		
ID:HU 016	Usuario: administrador	
Programador Responsable: Juan Tapia		
Nombre Historia: Tiempo de respuesta del sistema	Tipo: Requerimiento No Funcional	
Descripción: El sistema depende de la calidad del Internet para tiempo de respuesta		
Comentario: El sistema depende de la calidad del Internet para tiempo de respuesta, sin embargo, debe ser en tiempo real.		

Debido a la extensión de las historias de usuario del proyecto, se adjuntan en el Anexo B.

En la Tabla 2.8, se muestra la lista de los requerimientos funcionales y no funcionales, así como también, su respectiva Historia de Usuario. La nomenclatura respectiva dentro de la tabla es RF (requisito funcional), RNF (requisito no funcional), HU (Historia de Usuario).

Tabla 2.8 Requerimientos funcionales y no funcionales

Historia de Usuario	Tipo de Requerimiento	Descripción
HU001	RF001	Iniciar Sesión
HU002	RF002	Validación de credenciales
HU003	RF003	Administrar usuarios
HU004	RF004	Actualizar información de usuarios
HU005	RF005	Eliminar usuario
HU006	RF006	Buscar usuario
HU007	RF007	Administrar dispositivos
HU008	RF008	Actualizar información de dispositivos
HU009	RF009	Eliminar dispositivos
HU010	RF010	Buscar dispositivo
HU011	RF011	Visualizar cámara
HU012	RF012	Ingresar clave y huella para ingreso a puerta principal
HU013	RF014	Recibir notificaciones de sensor de presencia
HU014	RF015	Validación de claves de ingreso puerta
HU015	RNF001	Cerrar Sesión
HU016	RNF002	Tiempo de respuesta del sistema

2.1.6 DEFINICIÓN DE TABLERO KANBAN

Una vez que se han analizado todos los casos, se dividen las tareas del proyecto según la metodología Kanban. Dentro de cada tarjeta se desarrollan tareas específicas, de tal manera que se tiene la información necesaria y su estado, así como también, la fecha de entrega. Toda la información se ve reflejada en el tablero Kanban que administra e indica cómo está el proceso de cada tarea, tal como se indica en la Figura 2.14. En la Tabla 2.9, se muestra la organización de las tarjetas y sus procesos a desarrollar.

Tabla 2.9 Tareas en tablero Kanban

TARJETA	NOMBRE	REQUERIMIENTO	
Tarjeta 1	Autenticación de usuarios	1	Registro e inicio de sesión usuario.
		2	Cierre de sesión usuario.
Tarjeta 2	Administración de Usuarios	3	Administración de Usuarios(CRUD).
Tarjeta 3	Visualización de Cámara	4	Permitir visualización de cámaras.
		5	Permitir recibir notificaciones de presencia.
Tarjeta 4	Ensamblaje dispositivo cámara	6	Codificación dispositivo cámara.
		7	Conexión sensor PIR movimiento.
Tarjeta 5	Control acceso puerta principal	8	Obtener clave para puerta principal desde plataforma Firebase
		9	Identificar y confirmar huella digital almacenada en sensor biométrico.
Tarjeta 6	Control acceso puerta principal desde aplicación móvil	10	Autenticación biométrica en Android
		11	Ingreso de clave y confirmación en la plataforma Firebase
Tarjeta 7	Ensamblaje control acceso	12	Ensamblaje teclado y pantalla LCD
		13	Ensamblaje sensor biométrico.
Tarjeta 8	Ensamblaje control cerradura	14	Ensamblaje cerradura eléctrica y ESP
		15	Ensamblaje circuito detector fallas de energía
		16	Codificación en Arduino IDE

Como se muestra en la Figura 2.14, el tablero Kanban ayuda en la disposición de las tareas a realizar en tarjetas específicas, logrando tener un orden en la creación del sistema.



Figura 2.14 Tablero Kanban en metodología

2.1.7 ESTRUCTURA GENERAL DEL SISTEMA INTEGRADO

El Sistema fue desarrollado con una estructura cliente, Web API, base de datos, controladores y actuadores. Se lo describe de la siguiente manera:

- **Cliente:** aplicación móvil que gestiona el cliente con las opciones indicadas dentro del sistema integrado.
- **Web API:** donde se procesa las peticiones que realiza en cliente proveniente de la aplicación móvil.
- **Base de Datos:** donde se almacena los datos.
- **Actuador:** dispositivo que ejecuta las instrucciones para dar respuesta a las peticiones entregadas por el cliente a través de la aplicación móvil

2.1.7.1 Herramientas para el desarrollo del Sistema

En la Tabla 2.10, se muestran las herramientas necesarias para el desarrollo del Sistema Integrado, tanto en la parte de desarrollo de software como en el desarrollo de hardware.

Tabla 2.10 Herramientas para el desarrollo del sistema

HERRAMIENTA		DESCRIPCION
Servidor	Base de Datos	Para la base de datos se utilizará <i>Firebase Data base</i> , la cual brindará una actualización de los datos en la nube de manera directa y sin infraestructura adicional.
	Autenticación	La autenticación dentro de <i>Firebase</i> permite iniciar la aplicación de varias maneras; se utilizará autenticación con correo y contraseña
Controlador	Módulo ESP	Se va a utilizar como controlador principal para solicitud y envío de variables a la base de datos
Actuador	Dispositivos	Son los dispositivos que van a ser accionados una vez obtenidos los valores de la base de datos a través de los controladores
Cliente	Aplicación Móvil Android	El cliente se programa a través del IDE oficial <i>Android Studio</i> . La aplicación se desarrolla para el API 23 en adelante.

2.1.8 DIAGRAMA DE FLUJO

Este diagrama representa el flujo de actividad que realiza los diferentes procesos del Sistema Integrado como: ingreso a cámara de seguridad, administración de usuarios y control de acceso a la puerta principal.

2.1.8.1 Diagrama de Flujo de cámara de seguridad

Para poder acceder a las imágenes se debe ingresar en la base de datos de *Firebase*, así como al dispositivo anclado. Una vez cumplidas estas condiciones, se accede a la actividad que contiene el *web view*, el cual es el encargado de mostrar las imágenes de manera

secuencial, con un desfase de tiempo debido a los retardos de la red, así como también, al tiempo de retardo de la transmisión de cada una de las imágenes, tal como se muestra en la Figura 2.15

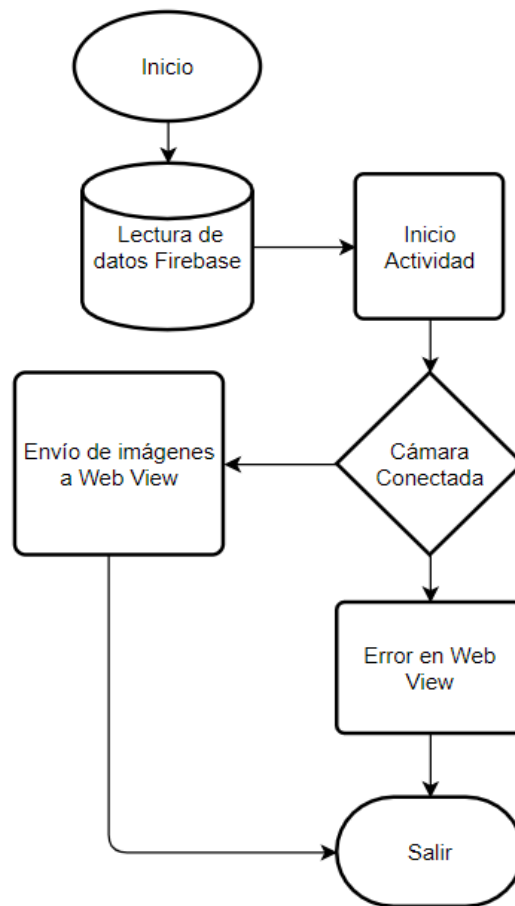


Figura 2.15 Diagrama de Flujo Cámara de Seguridad

2.1.8.2 Diagrama de Flujo de acceso puerta

Representa el acceso a la puerta principal, mediante la confirmación de la clave y la huella digital. Se tienen dos formas de acceder a la puerta principal de manera física y a través de la aplicación móvil. Si la apertura es de manera física (ver Figura 2.16) se solicita en primer lugar la contraseña personal de usuario; si es correcta, se procede a colocar la huella digital en el sensor biométrico. Una vez verificada la huella, la puerta se abre. Si el ingreso a la puerta se lo realiza a través de la aplicación móvil (ver Figura 2.17), la primera autenticación es la huella digital, si ésta correcta, se abre una actividad en la cual el usuario coloca su clave de acceso; una vez verificada la clave la puerta se abre.

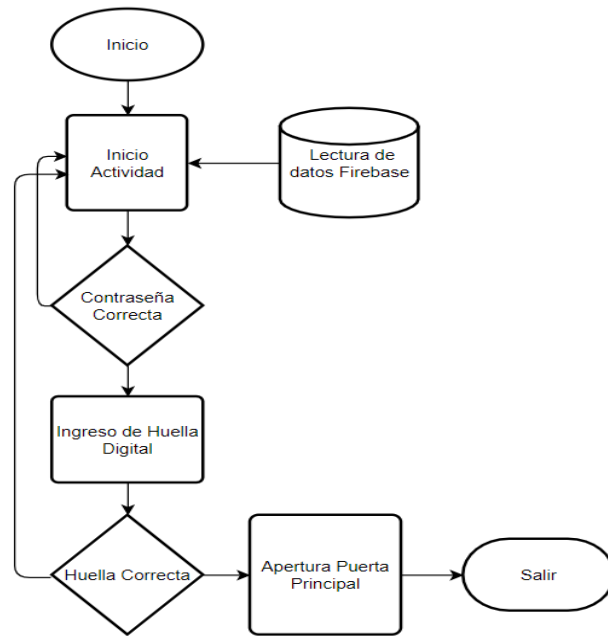


Figura 2.16 Diagrama de Flujo de Acceso Puerta Principal desde el panel

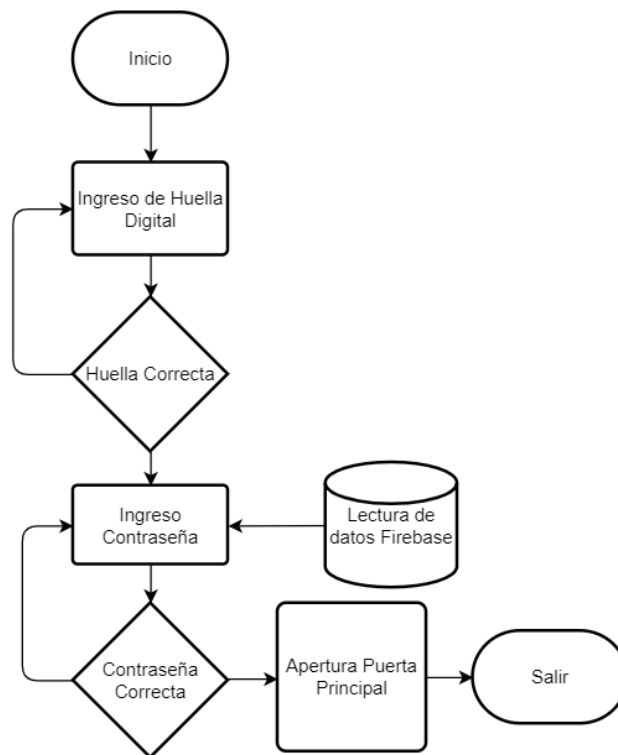


Figura 2.17 Diagrama de Flujo de Acceso Puerta Principal desde el móvil

2.1.9 DIAGRAMAS DE SECUENCIA

Los Diagramas de Secuencia representan la interacción entre los objetos y los componentes del Sistema durante la ejecución de un Caso de Uso, es decir, muestra la secuencia de mensajes entre instancias de clases, componentes, subsistemas o actores.

2.1.9.1 Diagrama de secuencia de lista de dispositivos

En la Figura 2.18 se muestra el diagrama de secuencia del cual se obtiene la lista de los dispositivos almacenados en Firebase dentro del sistema.

Una vez que el usuario se encuentra autenticado y confirmado en la base de datos, hace un pedido a la base de datos a través de una consulta, la misma que desencadena un evento llamado “ValueEventListener” que proporciona la información de la base de datos a la ruta específica de Firebase de los dispositivos. Se trae y almacena cada dispositivo en un objeto de clase “Producto”, que contiene los parámetros de cada dispositivo para que puedan ser configurados.

En la lista se presentan dos datos informativos: el nombre del dispositivo y la IP asignada.

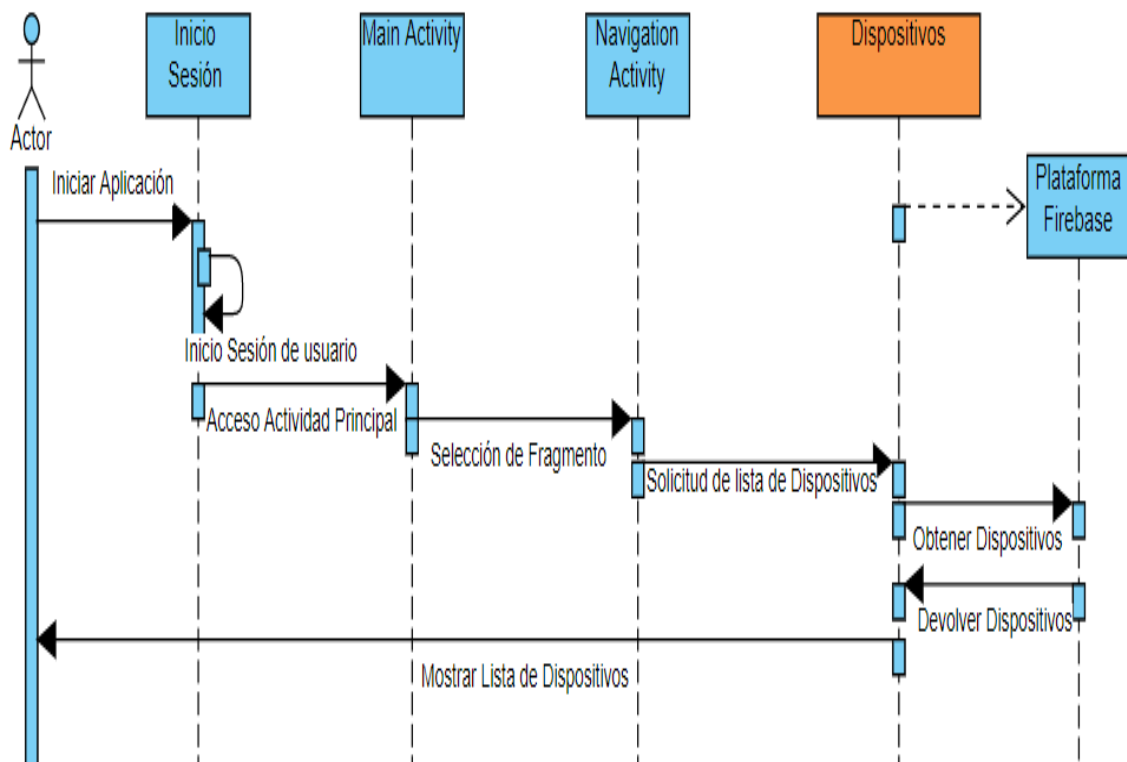


Figura 2.18 Diagrama de Secuencia Lista de Dispositivos

2.1.9.2 Diagrama de secuencia de lista de usuarios

En la Figura 2.19 se muestra el diagrama de secuencia del cual se obtiene la lista de usuarios almacenados en Firebase dentro del Sistema Integrado.

Una vez autenticado y confirmado el usuario en la base de datos, se hace un pedido a la base de datos a través de una consulta, la misma que desencadena un evento llamado “ValueEventListener” que proporciona la información de la base de datos a la ruta específica de Firebase de los usuarios. Cada usuario es sustraído y se almacena en un objeto de clase “Usuario”, que contiene los parámetros para que puedan ser configurados.

A esta lista de usuarios solo puede acceder el usuario Administrador, que será el encargado de realizar el CRUD de los usuarios dentro del Sistema Integrado y de dar los permisos correspondientes al escoger el tipo de usuario.

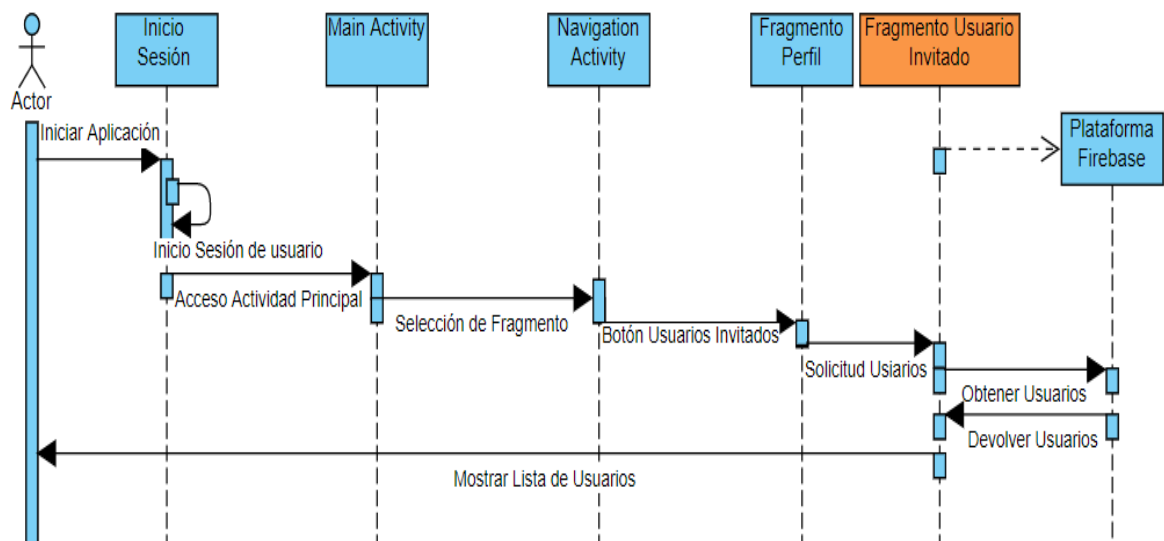


Figura 2.19 Diagrama de Secuencia Lista de Usuarios

2.1.9.3 Diagrama de secuencia de creación de usuarios

En la Figura 2.20 se muestra el diagrama de secuencia de creación de usuarios que van a ser almacenados en Firebase dentro del Sistema Integrado.

Una vez autenticado y confirmado el usuario en la base de datos como usuario administrador, podrá ingresar un nuevo usuario, para lo cual deberá ingresar los datos solicitados en la plantilla creada. Una vez que los datos se encuentran llenos, se hace un pedido a la base de datos a través de una consulta, la misma que desencadena un evento llamado “ValueEventListener” que creará al usuario con un id único descrito a través de un

“pushid”; los datos entregados por la aplicación serán guardados en el nodo “usuario” dentro de la base de datos de Firebase.

Una vez creado el usuario se regresará a la lista de usuarios, mostrando también al usuario reciente dentro de la misma.

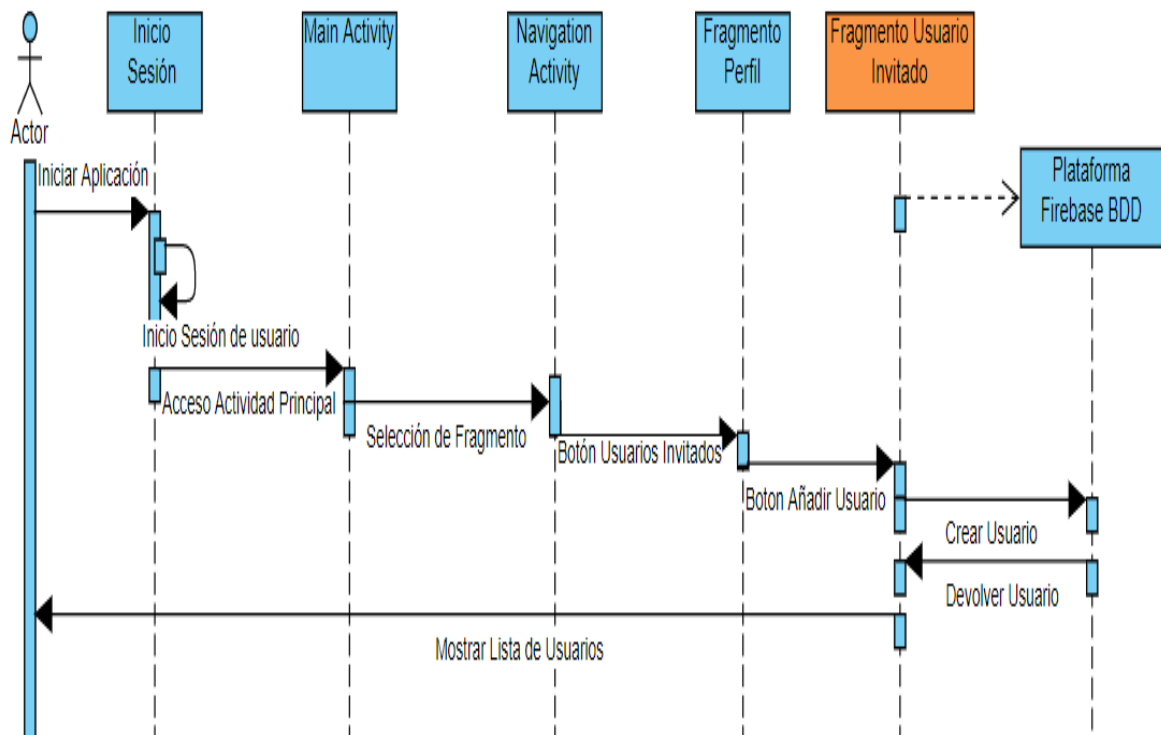


Figura 2.20 Diagrama de Secuencia creación Usuario

2.1.9.4 Diagrama de secuencia de cámara de seguridad

En la Figura 2.21 se indica el diagrama de secuencia en el cual se obtiene el video de la cámara de seguridad conectada dentro al sitio a supervisar.

Una vez autenticado y confirmado el usuario en la base de datos, genera un evento al presionar el botón de conectado dentro de la aplicación. Esta acción iniciará la conexión entre el usuario de la aplicación y la cámara de seguridad. Para este caso, se usa un protocolo llamado vidyio.io, el cual nos permitirá acceder a la cámara y verificar el estado del local a supervisar.

La conexión es TCP durante toda la transmisión del video y se accede a través de un *web view*, que se encuentra en la actividad de la cámara. Cada cámara tiene su propia actividad para ingreso y el *web view* muestra solo la imagen de la cámara que ha sido seleccionada.

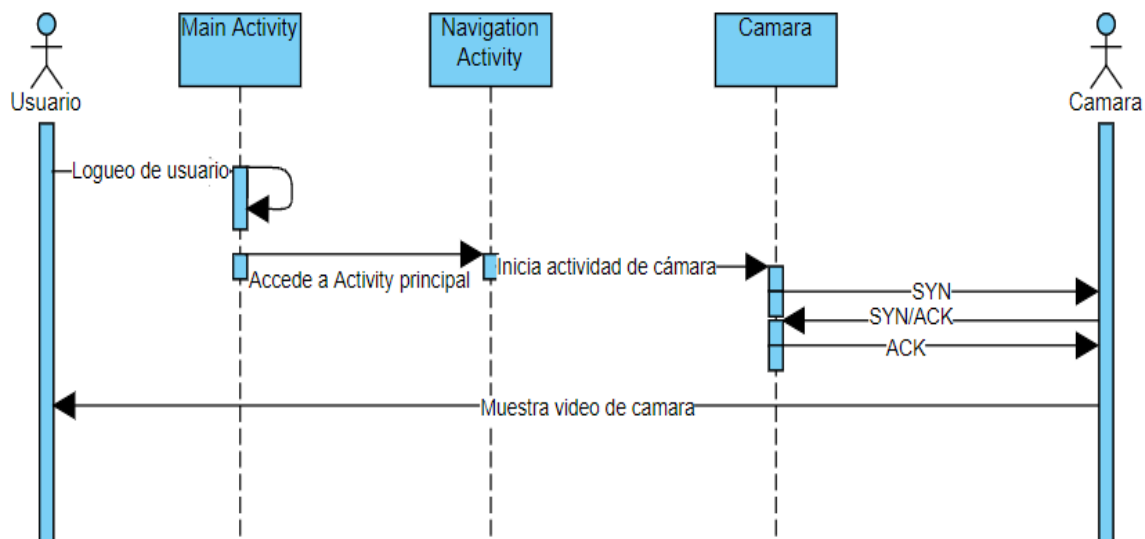


Figura 2.21 Diagrama de Secuencia Conexión Cámara

2.1.10 ESQUEMA DE AUTENTICACIÓN

El esquema de autenticación que se va a utilizar en el Sistema de Seguridad es de correo y contraseña ya que, en base a las entrevistas, los usuarios en su mayoría decidieron utilizar sus cuentas de correo electrónico como se puede observar en la Figura 2.3 para poder autenticarse en el sistema. Esto se debe a que la mayoría de las personas utilizan los correos electrónicos para las notificaciones de los servicios que contratan.

2.1.11 ESQUEMA DE BASE DE DATOS

Las bases de datos NoSQL (*Not only Structured Query Language*) son utilizadas para datos semi estructurados o no estructurados, ya que son más flexibles, tienen mejor rendimiento y son fáciles de desarrollar. La Plataforma Firebase maneja base de datos NoSQL, y se basa en un esquema de Documentos, donde los objetos se representan como documentos JSON ya que facilita el manejo de la información, así como el uso de clave-valor para poder gestionar sistemas de administración, perfiles de usuarios, y en el caso de Sistema de seguridad, la administración de dispositivos.

2.1.12 DISEÑO DE ACTIVIDADES Y FRAGMENTOS DE LA APLICACIÓN

Al usar una aplicación móvil para la gestión del sistema de seguridad se deben tomar en cuenta las diferentes interfaces de usuario que ayudarán a acceder a los dispositivos, así como, a las configuraciones de los usuarios. Por lo tanto, las siguientes imágenes muestran las plantillas (*mockups*) para la aplicación móvil Android.

En la Figura 2.22 se muestran las pantallas iniciales, es decir, la pantalla de presentación, de inicio de sesión o registro y la pantalla principal de la actividad. En la pantalla o actividad principal se muestra una barra de navegación, la misma que permite el desplazamiento de los fragmentos de dispositivos, acceso principal, notificaciones y perfil del usuario.

Después de la actividad principal se tienen actividades que son independientes, y son llamadas de acuerdo con el tipo de botón o imagen a ser pulsada en la aplicación. En la Figura 2.23 se puede observar el diseño de la actividad de la cámara, en la cual se captura una imagen de los eventos mostrados en la cámara y así almacenarlos.

La Figura 2.24 representa la actividad de usuario, en la cual se ubican los datos del usuario, por ejemplo: nombre, imagen, apellido, teléfono, correo. Además, se utiliza un campo en el cual se muestra la contraseña para el acceso a la puerta principal, así como la huella digital que se va a usar. En caso de tener alguna modificación se usa una actividad similar con campos a rellenar.

Por otra parte, en la Figura 2.25 se tiene el diseño del listado de notificaciones que muestra el título y una descripción de cada notificación. El contenido de estas notificaciones dependerá de cada uno de los dispositivos y de la acción que desencadena la misma.

Adicionalmente, en la Figura 2.26 se encuentra el diseño de la actividad del control de acceso, el cual pasa por la autenticación biométrica para acceder a la digitación de la clave personal para la apertura de la puerta principal.

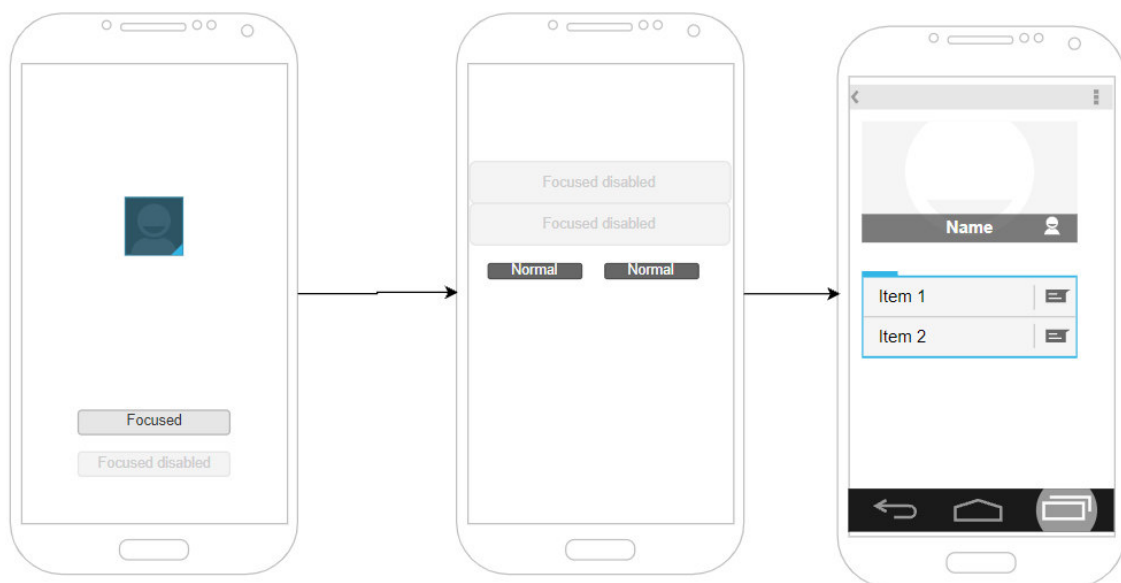


Figura 2.22 Pantallas de Presentación, Inicio y Principal

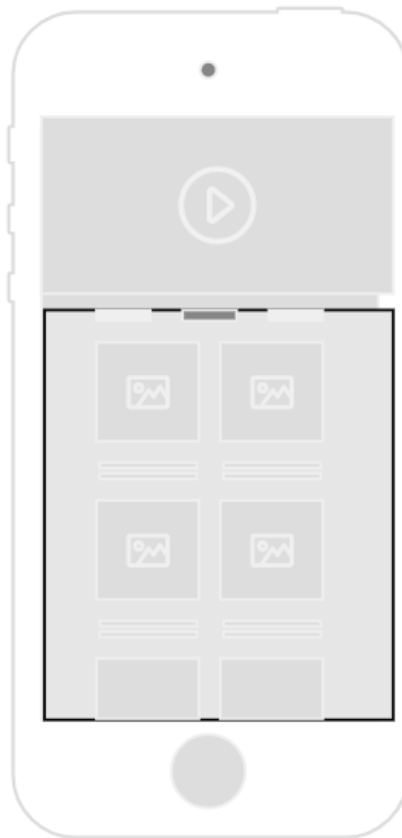


Figura 2.23 Actividad de Cámara

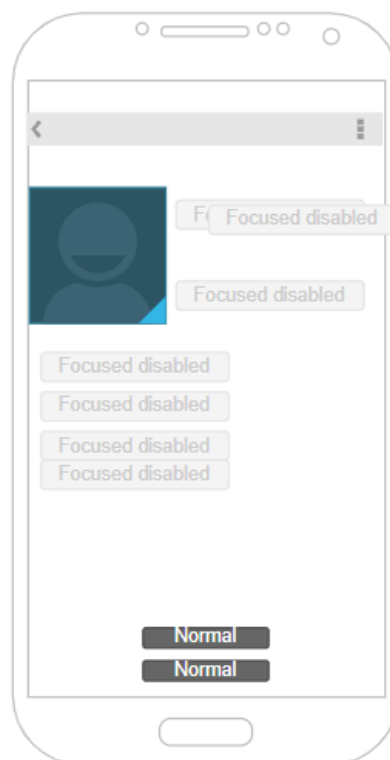


Figura 2.24 Actividad perfil de usuario



Figura 2.25 Actividad Notificaciones

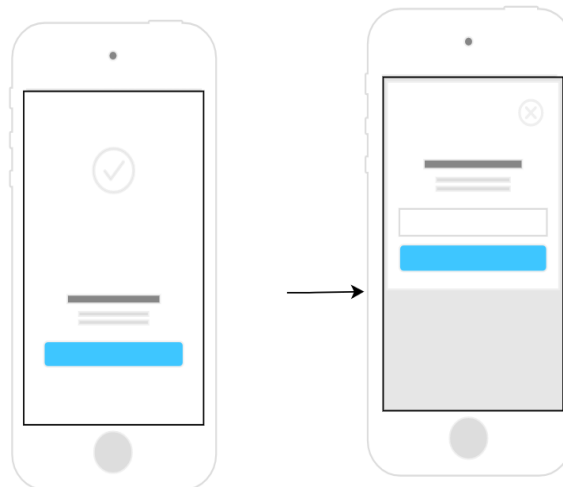


Figura 2.26 Actividad Control Acceso

2.1.13 PROGRAMACIÓN EN SOFTWARE ARDUINO IDE

2.1.13.1 Dispositivo de apertura de puerta principal

Después de realizar la configuración inicial (ver Figura 2.27) e incluir las librerías como: la conexión a Internet y a la plataforma Firebase, el programa debe conectarse a la base de datos de Firebase para obtener y revisar la variable “control_apertura” de la base de datos, si se encuentra en “abierto” abre la puerta y desactiva las celdas de carga. Si se encuentra en “cerrado” se activan tanto las celdas de carga como el relé para el cierre de la puerta como se muestra en la Figura 2.28.

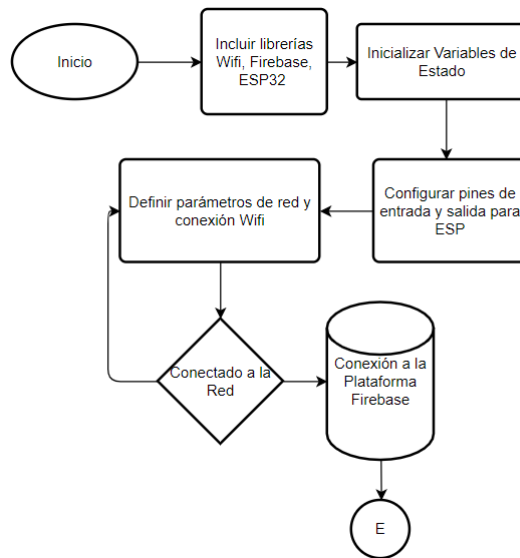


Figura 2.27 Diagrama de flujo configuración inicial del control de apertura de la puerta

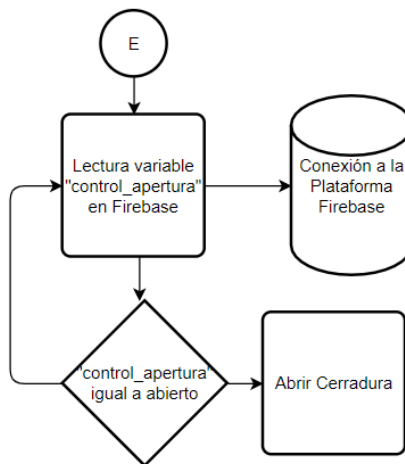


Figura 2.28 Diagrama de flujo accionamiento del control de apertura de puerta

2.1.13.2 Dispositivo de control de acceso

Para el dispositivo de control de acceso se deben configurar las librerías e inicialización de los pines para el manejo del teclado de 4x3, la pantalla LCD de 16x2 y el sensor biométrico. Para la comunicación de la pantalla se utiliza el bus I2C para disminuir la cantidad de conexiones para el manejo de la LCD que se conectan con los pines SDA y SCL del ESP32. Por otra parte, para la configuración del teclado matricial 4x3 se utiliza siete entradas digitales, que se definen en los pines del ESP32. Adicionalmente, se definen dos pines: transmisión y recepción para el sensor biométrico en el módulo ESP32 para lograr la identificación, almacenamiento de huellas digitales. En la figura 2.29 se muestra la configuración inicial.

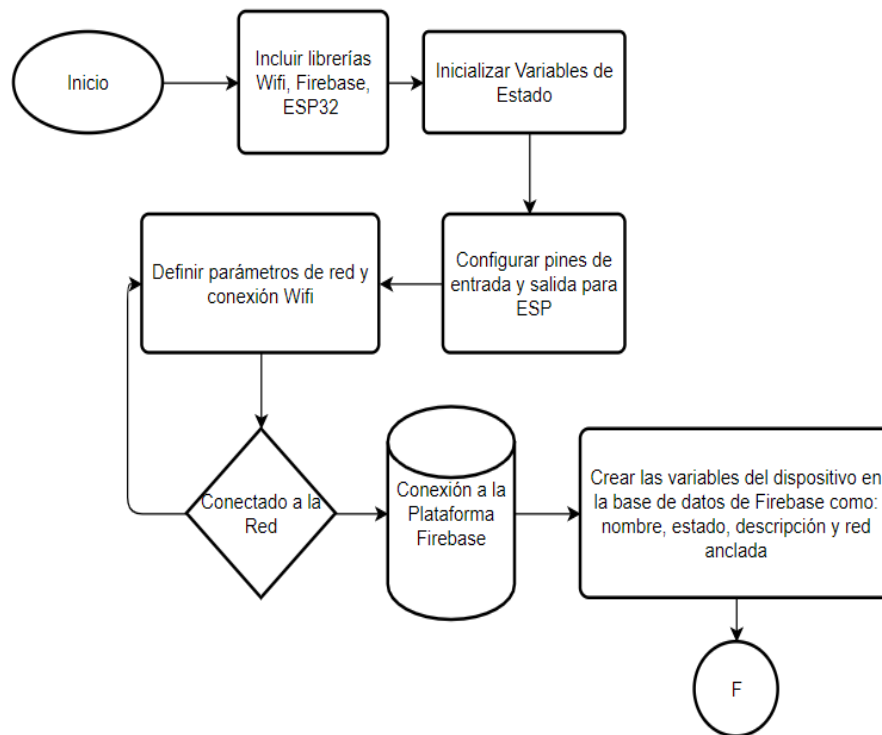


Figura 2.29 Diagrama de flujo de la configuración inicial del control de acceso

Como se observa en la Figura 2.30, en el programa de control, se muestra inicialmente en la pantalla “Ingrese la clave”; mediante el teclado se ingresa la clave de cinco dígitos que se almacenan en la variable “clave” y va a ser comparada con la clave que se toma de la base de datos de la plataforma Firebase. Una vez validada la clave se procede con la solicitud de ingreso de la huella digital a través de la pantalla, y se procede a colocar el dedo en el sensor biométrico. A continuación, si la huella ingresada coincide con una de las huellas almacenadas en el módulo se abre la puerta, caso contrario, se solicita nuevamente la clave para poder verificar los datos nuevamente, esto incluye el ingreso de la clave inicial.

2.1.14 SISTEMA DE ALIMENTACIÓN (UPS)

Para el diseño de la UPS (*Uninterruptable Power Supply*) se debe tener en cuenta que la carga aproximada en la instalación del Sistema de seguridad es de 90 W por esta razón, por disponibilidad en el mercado, se utilizará una UPS que pueda abastecer de 1000 W durante una hora para que permanezca en funcionamiento los elementos del sistema de seguridad. En la Figura 2.31 se muestra la UPS.

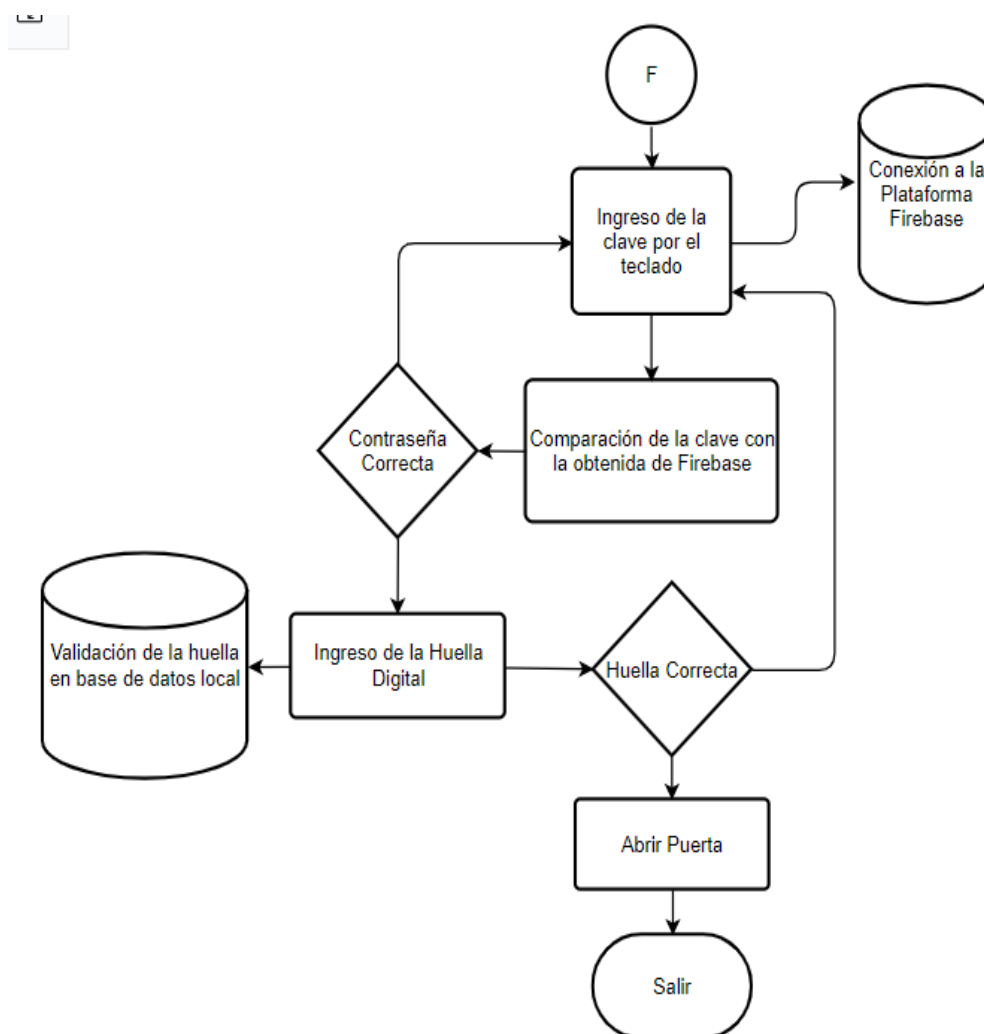


Figura 2.30 Diagrama de flujo funcionamiento del control de acceso.



Figura 2.31 UPS utilizada para el sistema de seguridad

2.2 FASE DE IMPLEMENTACIÓN

En esta sección se muestra el proceso de implementación de la aplicación para el sistema integrado, es decir, aplicación móvil, módulos controladores ESP y el ensamblaje del sistema en el local a proveer seguridad.

Como se presenta en la Figura 2.32, las tareas a realizar en la fase de Implementación se encuentran en desarrollo y las tareas de Diseño ya se han completado hasta el momento, teniendo pendientes las tareas de prueba.



Figura 2.32 Tablero Kanban en fase de implementación

2.2.1 IMPLEMENTACIÓN DEL SOFTWARE

2.2.1.1 Implementación de la aplicación móvil

El diseño de la interfaz de usuario indica la estructura de la aplicación, es decir, la forma en la que se encuentra constituida y la cantidad de elementos necesarios para los requerimientos que han sido especificados por el usuario. Todos estos elementos o diseños tienen un orden jerárquico de objetos.

El diseño se puede declarar de dos formas:

- Declarar elementos de la interfaz de usuario en XML: Android suministra un vocabulario XML que tiene coincidencia con las clases y subclases de vistas.
- Crear instancias de elementos de diseño en tiempo de ejecución: Se pueden crear objetos de clase View y Group View de forma programática.

En el presente Trabajo de Titulación la programación se realiza a través de XML que permite separar la presentación de la interfaz de la codificación de la aplicación, así como también, la creación de diseños para diferentes tamaños de dispositivos Android.

2.2.1.1.1 *Especificación de la interfaz de usuario.*

Se describen las UI (*User Interface*) de la aplicación, asociadas a los diferentes procesos que se utilizan dentro del Sistema Integrado y la estructura de los archivos que forman parte. El código fuente de la aplicación se encuentra descrito en el Anexo E.

Al crear el proyecto, Android Studio separa automáticamente las clases Java y los recursos usados en XML para la interfaz gráfica. Por lo tanto, dentro del fichero Java se pueden encontrar las clases, que son el *backend* de la aplicación para la interfaz del usuario y los que forman parte de la extracción de datos de la base en la plataforma y sirven para guardar la información del sistema. Por otra parte, se encuentra el fichero res, que proporcionan el espacio de trabajo para la interfaz de usuario (XML).

Para el desarrollo de la aplicación se tiene un fichero principal en el cual se declaran todos los permisos necesarios y todas las actividades que se van a utilizar para el correcto funcionamiento. Este archivo es de tipo .XML y se llama "AndroidManifest"(ver Figura 2.33). En el caso de la presente aplicación existen varios permisos que se van a usar, como son: Internet, acceso a la red, uso del sensor biométrico, entre otros.

El código se desarrolla en la carpeta "Main", dentro de ésta se encuentran dos carpetas que contienen la información de la interfaz y el código que son: "java" y "res", dentro de java se encuentran los paquetes con todo el código desarrollado como su nombre lo indica, es decir, Java. La primera carpeta contiene Activitys que están organizadas para la interfaz del usuario, las cuales contienen métodos que realizan operaciones. Por otro lado, se encuentra la carpeta "res", que alberga todos los archivos XML donde se diseña las interfaces visuales de la aplicación, así como también, se tiene las imágenes y recursos dibujados o agregados desde los iconos predeterminados de Android (carpetas como: *drawable*, *mipmap*, *navigation*). Las Figuras 2.34 y 2.35 presentan el desarrollo del XML y la codificación en Java de la actividad principal.

```

15 <uses-permission android:name="android.permission.USE_BIOMETRIC" />
16
17 <application
18     android:allowBackup="true"
19     android:icon="@mipmap/ic_launcher"
20     android:label="SistemaSeguridad"
21     android:roundIcon="@mipmap/ic_launcher_round"
22     android:supportsRtl="true"
23     android:theme="@style/AppTheme">
24     <activity
25         android:name=".PuertaPrincipal"
26         android:label="PuertaPrincipal"
27         android:theme="@style/AppTheme.NoActionBar"></activity>
28     <activity android:name=".DispositivosNoAnclados" />
29     <activity android:name=".UsuariosInvitados" />
30     <activity android:name=".UsuarioInvitado" />
31     <activity android:name=".ConfiguracionCamara" />
32
33     <service
34         android:name=".MensajeriaFirebase"
35         android:enabled="true"
36         android:exported="true"
37         android:stopWithTask="false">
38         <intent-filter>
39             <action android:name="com.google.firebase.MESSAGING_EVENT" />
40         </intent-filter>
41     </service>
42
43     <activity
44         android:name=".Inicio"
45         android:theme="@style/AppTheme.NoActionBar">
46         <intent-filter>
47             <action android:name="android.intent.action.MAIN" />

```

Figura 2.33 Sección del código de AndroidManifest

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/container"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent">
8
9     <com.google.android.material.bottomnavigation.BottomNavigationView
10         android:id="@+id/nav_view"
11         android:layout_width="0dp"
12         android:layout_height="wrap_content"
13         android:background="?android:attr/windowBackground"
14         app:layout_constraintBottom_toBottomOf="parent"
15         app:layout_constraintHorizontal_bias="0.0"
16         app:layout_constraintLeft_toLeftOf="parent"
17         app:layout_constraintRight_toRightOf="parent"
18         app:menu="@menu/bottom_nav_menu" />
19
20     <fragment
21         android:id="@+id/nav_host_fragment"
22         android:name="androidx.navigation.fragment.NavHostFragment"
23         android:layout_width="match_parent"
24         android:layout_height="match_parent"
25         app:defaultNavHost="true"
26         app:layout_constraintHorizontal_bias="0.0"
27         app:layout_constraintLeft_toLeftOf="parent"
28         app:layout_constraintRight_toRightOf="parent"
29         app:layout_constraintTop_toTopOf="parent"
30         app:navGraph="@navigation/mobile_navigation" />
31

```

Figura 2.34 Código XML de Actividad Principal

```

34
35 public class MainActivity extends AppCompatActivity {
36
37     private FirebaseAuth firebaseAuth;
38     private FirebaseAuth.AuthStateListener firebaseAuthListener;
39     FirebaseDatabase usuariosDatabase;
40     DatabaseReference Seg_ref;
41     TextView tx1;
42     @Override
43     protected void onCreate(Bundle savedInstanceState) {
44         super.onCreate(savedInstanceState);
45         setContentView(R.layout.activity_main);
46         BottomNavigationView navView = findViewById(R.id.nav_view);
47         // Passing each menu ID as a set of Ids because each
48         // menu should be considered as top level destinations.
49         AppBarConfiguration appBarConfiguration = new AppBarConfiguration.Builder(
50             R.id.navigation_home, R.id.navigation_notifications, R.id.navigation_perfil)
51             .build();
52         NavController navController = Navigation.findNavController( activity: this, R.id.nav_host_fragment);
53         NavigationUI.setupActionBarWithNavController( activity: this, navController, appBarConfiguration);
54         NavigationUI.setupWithNavController(navView, navController);
55         usuariosDatabase = FirebaseDatabase.getInstance();
56         Seg_ref = usuariosDatabase.getReference(Firebase_References.SEGURIDAD_REFERENCES);
57         firebaseAuth = FirebaseAuth.getInstance();
58         final FirebaseUser user = firebaseAuth.getCurrentUser();
59         if (user != null) {
60             Query q = Seg_ref.child(Firebase_References.USER_REFERENCES).orderByChild("correo").equalTo(user.getEmail());
61             q.addListenerForSingleValueEvent(new ValueEventListener() {
62                 @Override
63                 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
64                     if (dataSnapshot.exists()) {
65                         for (DataSnapshot datasnapsho :
66                             dataSnapshot.getChildren()) {
67                             Usuario user1 = datasnapsho.getValue(Usuario.class);
68                             ObtenerToken(user1.getCorreo().toString());

```

Figura 2.35 Código de Actividad Principal en JAVA

La Figura 2.36 muestra parte del código utilizado para la creación del objeto “Usuario” y los métodos a utilizar para cada evento que se vaya a desencadenar en la aplicación. Adicional a éstas, se tienen clases que son exclusivamente para el manejo de variables dentro del entorno de la aplicación, así como su relación con la base de datos NoSQL para guardar datos.

La guía de Android en su API (*Application Programming Interface*), muestra a los elementos como objetos *View* o *ViewGroup*. Una *View* o vista es un objeto con el cual el usuario es capaz de interactuar; mientras que un *ViewGroup*, es aquel que contiene varias vistas.

El diseño es posible realizarlo solo con objetos *View* en el código de las *Activitys*, sin embargo, es mucho más sencillo utilizar los archivos XML para el diseño de los interfaces.


```

1 package com.jgtsecurity.seguridad.objetos;
2
3 public class Usuario {
4     private String id;
5     private String nombre;
6     private String apellido;
7     private String correo;
8     private int telefono;
9     private String llave_notificaciones;
10    private String contrasena_puerta;
11    private String contrasena_inicio_sesion;
12    private String llave_mensajeria;
13    private String clave_emergencia;
14
15    public Usuario() {}
16
17    public Usuario(String nombre, String apellido, String contrasena_puerta) {
18        this.nombre = nombre;
19        this.apellido = apellido;
20        this.contrasena_puerta = contrasena_puerta;
21    }
22
23    public Usuario(String id, String nombre, String apellido, String correo, String contra
24        this.id = id;
25        this.nombre = nombre;
26        this.apellido = apellido;
27        this.correo = correo;
28        this.contrasena_inicio_sesion = contrasena_inicio_sesion;
29        this.contrasena_puerta = contrasena_puerta;
30    }
31
32    public Usuario(String id, String nombre, String apellido, String correo, int telefono,
33        this.id = id;

```

Figura 2.36 Clase Usuario

2.2.1.1.2 Uso de adaptadores para listar datos de una vista

Para rellenar ciertas vistas como listas, es necesario el uso de adaptadores como se muestra en la Figura 2.37, los cuales se enlazan a la *Activity* con dicha lista. Estos datos se recuperan de bases de datos que pueden ser relacionales o no. En el caso del sistema integrado en la plataforma Firebase se tiene una base NoSQL.

Los datos obtenidos de Firebase son personalizados, se crea el adaptador personalizado tomando los datos principales a mostrarse (ver Figura 2.38). Se ha utilizado un constructor con los siguientes parámetros: contexto de la aplicación, referencia al recurso, y lista de datos a adaptar.

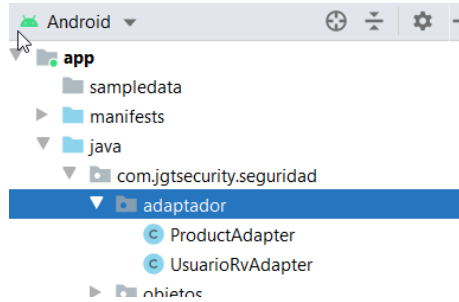


Figura 2.37 Adaptadores

```

public class ProductAdapter extends RecyclerView.Adapter<ProductAdapter.ProductoViewHolder> {

    private ArrayList<Producto> mrvlist;
    private OnItemClickListener mListener;
    static Context context;

    public ProductAdapter(ArrayList<Producto> rvlist) { this.mrvlist = rvlist; }

    public interface OnItemClickListener{
        void OnItemClick(int posicion);
        void OnDeleteClick(int posicion);
    }

    public void setOnItemClickListener(OnItemClickListener listener) { mListener = listener; }

    public static class ProductoViewHolder extends RecyclerView.ViewHolder {
        public TextView txtNombre_producto, txtDescripcion;
        public ImageView imgIcono, imgdelete;

        public ProductoViewHolder(@NonNull View itemView, final OnItemClickListener listener) {
            super(itemView);
            txtNombre_producto = itemView.findViewById(R.id.txtVNombreDisp);
            txtDescripcion = itemView.findViewById(R.id.txtView_descriptionDispositivo);
            //imgIcono = itemView.findViewById(R.id.img_producto);
            imgdelete = itemView.findViewById(R.id.img_delete);
            itemView.setOnClickListener((v) -> {
                if(listener!=null){
                    int position = getAdapterPosition();
                    if(position!=RecyclerView.NO_POSITION){
                        listener.OnItemClick(position);
                    }else{
                        Toast.makeText(context, text: "No hay dispositivos a seleccionar", Toast.LENGTH_SHORT).show();
                    }
                }
            });
        }
    }
}

```

Figura 2.38 Código Adaptador Dispositivos

2.2.1.1.3 Eventos

Los eventos son interfaces en la clase View que contienen métodos de llamados a las funciones, los cuales son llamados por el marco de trabajo de Android cuando una determinada acción ocurre en el objeto [22]. Los eventos son elaborados de manera generalizada, por lo tanto, para su uso se debe extender la clase y reemplazarlo con las acciones que se desea.

```

public void ConfirmarClave(View view) {
    conf_contr= con_puerta.getText().toString();
    Query q = Seg_Ref.child(Firebase_References.USER_REFERENCES);
    q.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot datasnapshot) {
            try {
                for (DataSnapshot dato: datasnapshot.getChildren()){
                    Usuario user = dato.getValue(Usuario.class);
                    if(user.getContrasena_puerta().equals(conf_contr)){
                        Toast.makeText( context: PuertaPrincipal.this,user.getNombre(),Toast.LENGTH_SHORT).show();
                        AbrirPuertaEsp32();
                        finish();
                    }else{
                        BuscarUsuariosInvitados();
                        Toast.makeText( context: PuertaPrincipal.this, text: "no existe coincidencia",Toast.LENGTH_SHORT).show();
                    }
                }
            }catch(Exception e){
                Toast.makeText( context: PuertaPrincipal.this, text: "no existe coincidencia total",Toast.LENGTH_SHORT).show();
            }
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {
        }
    });
}
}

```

Figura 2.39 Código de Evento Puerta Principal

2.2.1.1.4 Actividades de la aplicación

a. Pantalla de Inicio

Es la primera pantalla o *Activity* que se muestra al usuario. Esta pantalla muestra el logo o cualquier otro elemento que permita identificar el sistema. La vista de la pantalla de inicio se indica en la Figura 2.40. Se muestra cuando el usuario abre la aplicación y consta de los botones para el ingreso o registro de un nuevo usuario para el acceso a la aplicación.

Tabla 2.11 Descripción pantalla inicio

id	Tipo de Elemento	Descripción
1	Image View	Imagen que muestra logo de la aplicación
2	Button	Botón usado para el ingreso de un usuario existente
3	Text View	Texto usado para crear un nuevo usuario

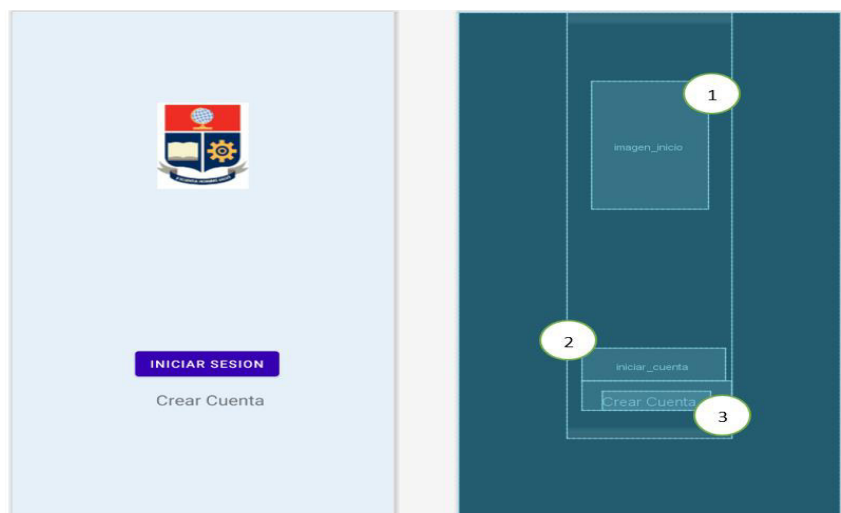


Figura 2.40 Estructura e interfaz de la pantalla de inicio

b. Pantalla de Sesión

Este interfaz permite el ingreso a la aplicación y a sus herramientas; se puede realizar el inicio de sesión mediante correo electrónico y contraseña. En este *Activity* se cuenta con el ingreso para crear un usuario y para el inicio de sesión. Por lo tanto, dependiendo del tipo de ingreso se activan y validan los campos correspondientes.

Después de la pantalla de inicio se muestra la UI de inicio de sesión de usuarios, tal como se puede observar en la Figura 2.41.

Tabla 2.12 Elementos de la interfaz de Inicio de sesión

Id	Tipo de elemento	Descripción
1	Image View	Imagen que muestra logo de la aplicación
2	Edit. Text	Texto a editar para colocar el nombre
3	Edit. Text	Texto a editar para colocar el apellido
4	Edit. Text	Texto a editar para colocar el correo
5	Edit. Text	Texto a editar para colocar la contraseña
6	Edit. Text	Texto a editar para colocar la confirmación de la contraseña
3	Botón	Inicio de sesión o registro para el ingreso
4	Botón	Para salir a la pantalla de inicio

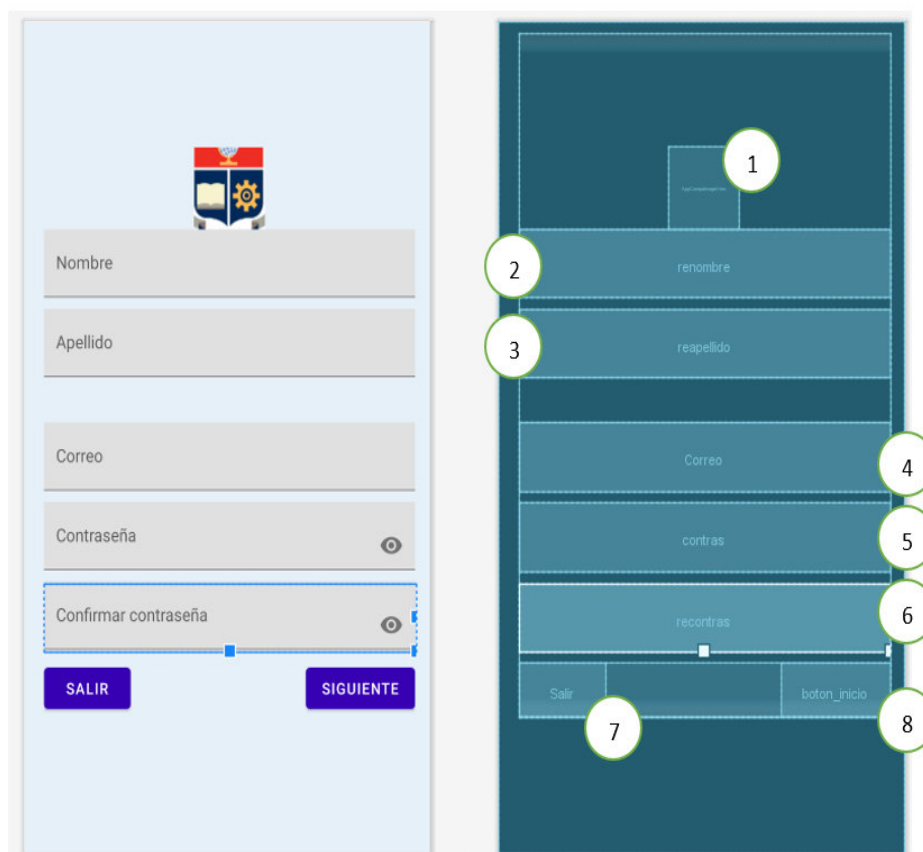


Figura 2.41 Estructura e interfaz de pantalla de sesión

c. Interfaz de Consola

Es la interfaz principal del sistema de seguridad en la cual se van a alojar los fragmentos que se usan para poder visualizar los dispositivos que se encuentran conectados, así como también, las notificaciones de los diferentes eventos que sucedan y el perfil del usuario. En las Figuras 2.42 y 2.43, se muestran los componentes de la interfaz y el menú de acciones a realizar respectivamente.

En la Tabla 2.13 se presentan los componentes utilizados para esta interfaz.

Tabla 2.13 Elementos de interfaz de consola

Id	Tipo de elemento	Descripción
1	Nav host Fragment	Receptor de fragmentos para la aplicación móvil
2	Button Navigation View	Botones para paso de un fragmento a otro

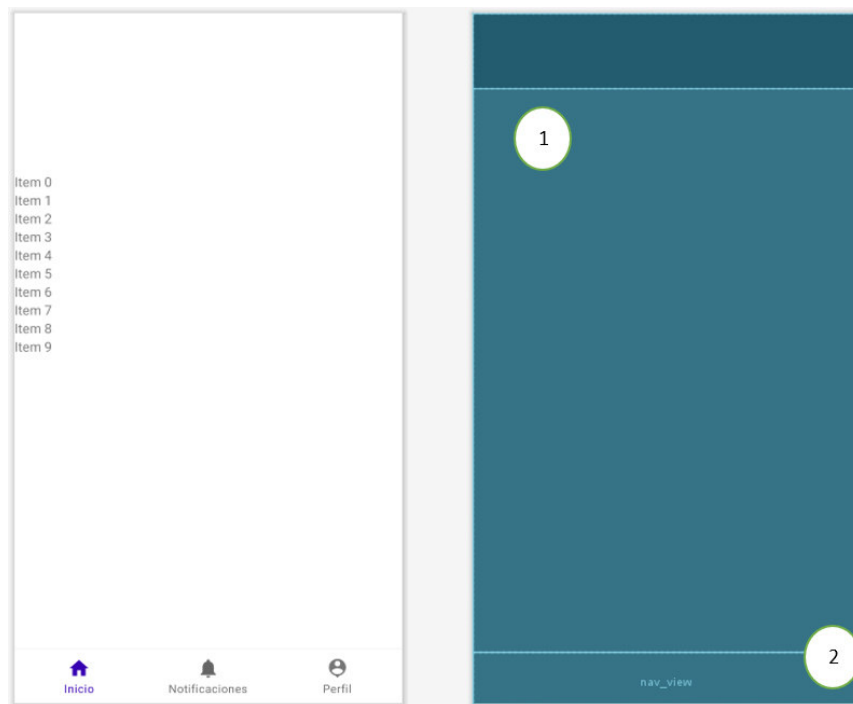


Figura 2.42 Estructura e interfaz de actividad principal

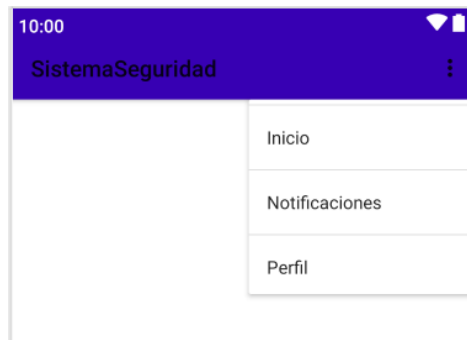


Figura 2.43 Menú principal para Navigation View (Fragmentos)

d. Fragmento Inicio (Sistema Seguridad)

En este fragmento (ver Figura 2.44) se observa una lista de los dispositivos conectados al sistema de seguridad en el cual se puede seleccionar al dispositivo para ingresar a la configuración. Además, cuenta con un ícono superior para añadir nuevos dispositivos. En la Figura 2.45 se encuentra especificado el ítem de agregar dispositivos nuevos.

Para la lista de los elementos se toman en cuenta solo tres características que sirven para identificar a cada dispositivo. En la Figura 2.46 se especifica la estructura e interfaz de cada ítem.

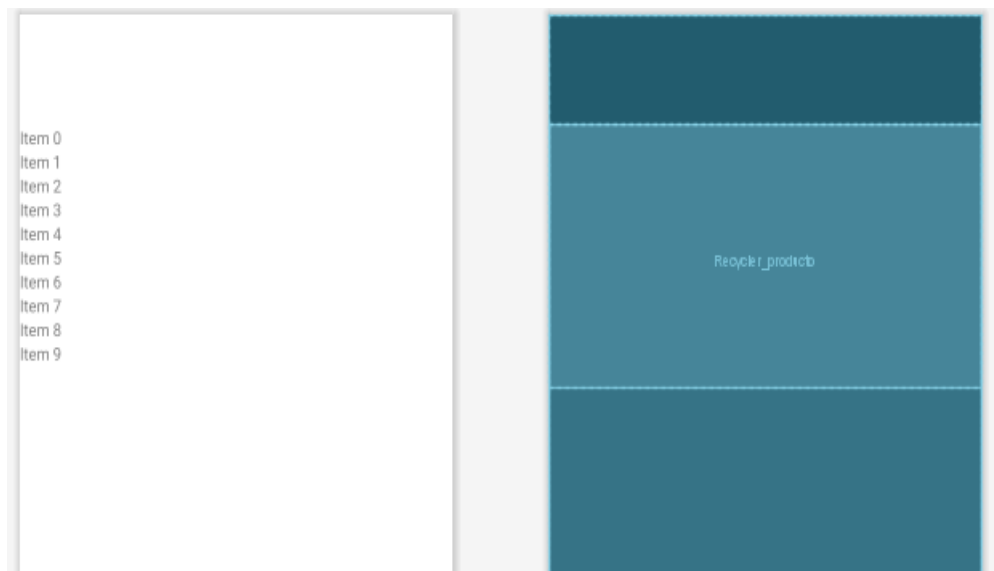


Figura 2.44 Estructura e interfaz del fragmento



Figura 2.45 Menú agregar dispositivos



Figura 2.46 Estructura e interfaz de ítem dispositivo

Tabla 2.14 Elementos del interfaz ítem dispositivo

Id	Tipo de elemento	Descripción
1	Image View	Icono de imagen principal
2	Text View	Texto que indica el nombre del dispositivo
3	Text View	Texto que indica descripción del dispositivo
4	Image View	Icono de imagen para borrar el elemento del sistema

e. Interfaz de Cámara de Seguridad

En esta interfaz se puede acceder a la cámara, para observar las actividades que se realizan en el instante que se conecta. Utiliza solo video, ya que se encuentra almacenando y enviando las imágenes desde la cámara hacia la plataforma Firebase y desde la plataforma a la aplicación como un conjunto de imágenes consecutivas.

En la Figura 2.47 se puede observar su diseño y estructura; mientras que en la Tabla 2.15 los componentes.



Figura 2.47 Estructura e interfaz de cámara de seguridad

Tabla 2.15 Elementos pantalla cámara de seguridad

Id	Tipo de elemento	Descripción
1	Text View	Texto para titulo
2	Text View	Texto que muestra en estado del dispositivo
3	Web View	Muestra las imágenes de video.
4	Text View	Muestra la fecha y hora actual
5	Button	Captura la imagen de ese instante para almacenarla
6	Button	Salir de la actividad

f. Interfaz de Control de Acceso a Puerta

La interfaz de control de acceso se utiliza para ingresar a la puerta principal. Contiene dos partes, la primera es la autenticación a través de la huella digital; mientras que la segunda se muestra una vez confirmada la huella, en la cual se debe ingresar la clave de seguridad

autorizada. La Figura 2.48 muestra la vista de la interfaz y su estructura. Los componentes están descritos en la Tabla 2.16.

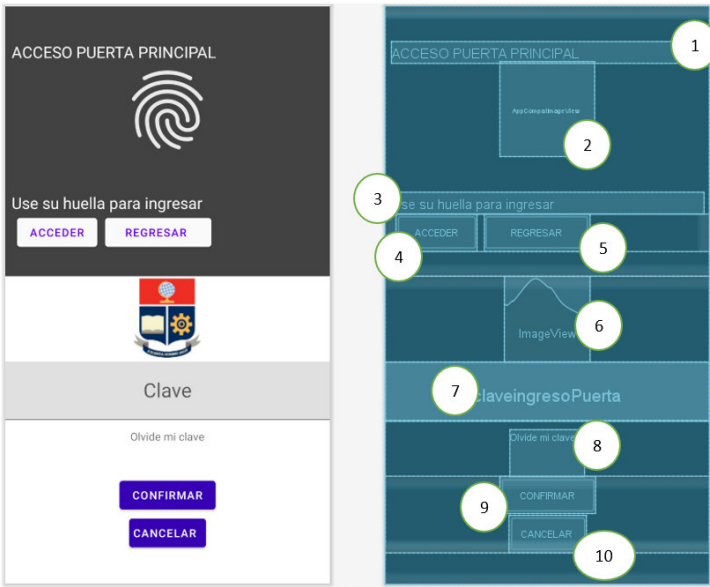


Figura 2.48 Estructura e interfaz acceso puerta

Tabla 2.16 Elementos de interfaz de control de acceso

Id	Tipo de elemento	Descripción
1	Text View	Texto descriptivo para ingreso
2	Image View	Muestra imagen inicial para autenticación de huella
3	Text View	Muestra descripción para el ingreso con autenticación de huellas
4	Button	Botón para ingresar a la autenticación con huella
5	Button	Botón para regresar a la actividad principal
6	Image View	Muestra imagen de logo
7	Edit. Text	Ingreso de clave
8	Text View	Texto para solicitar nueva clave
9	Button	Botón para confirmar la clave de acceso
10	Button	Botón para salir a la actividad principal

g. Fragmento Perfil

Este fragmento muestra el perfil del usuario, es decir, los datos del usuario principal. Adicionalmente, lleva a la actividad donde se encuentra la lista de los usuarios invitados para poder acceder a la puerta principal. En la Figura 2.49 se observa la estructura de la interfaz gráfica del fragmento y en la Tabla 2.17 se muestran los elementos que la constituyen.

Tabla 2.17 Elementos de la pantalla de perfil

Id	Tipo de elemento	Descripción
1	Image View	Muestra una imagen de perfil
2	Text View	Muestra el nombre del usuario
3	Text View	Muestra el apellido del usuario
4	Text View	Texto descriptivo
5	Text View	Muestra el correo del usuario
6	Text View	Texto descriptivo
7	Text View	Muestra el teléfono del usuario
8	Text View	Texto descriptivo
9	Text View	Muestra la contraseña del usuario
10	Button	Muestra activity con lista de usuarios invitados

h. Interfaz de administración de Usuarios

Permite gestionar los usuarios invitados que se encuentran dentro del sistema, aquí se agregan o eliminan a los usuarios que pueden acceder al sistema integrado una vez que se autenticuen. En la Figura 2.50, se muestra el diseño de la interfaz y en la Tabla 2.19 se presentan los componentes.

Tabla 2.18 Elementos pantalla usuarios

Id	Tipo de elemento	Descripción
1	Recycler View	Muestra la lista de usuarios del sistema
2	Floating Buttom	Botón para añadir nuevo usuario al sistema



Figura 2.49 Estructura e interfaz de perfil de usuario

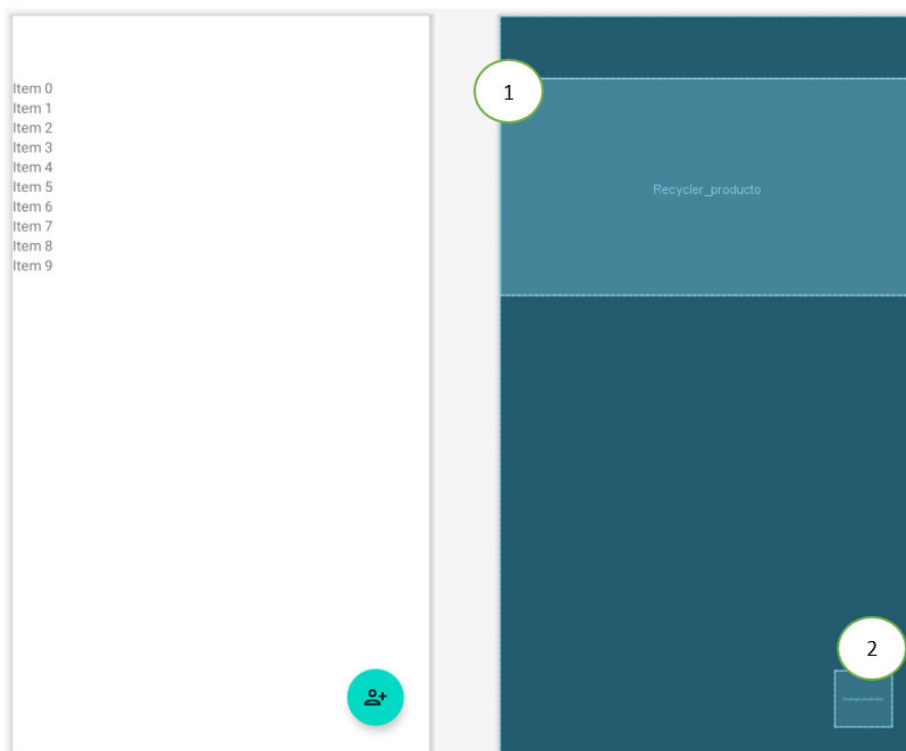


Figura 2.50 Pantalla lista usuarios

i. Pantalla ingreso usuarios

En la pantalla de administración de usuarios es necesario presentar también la interfaz de creación de usuarios invitados dentro de la cual se van a colocar los datos más relevantes para el ingreso del usuario al Sistema integrado. Además, se ingresa por *backend* una variable adicional la cual será enviada al usuario una vez que se haya creado con éxito.

Tabla 2.19 Elementos de pantalla ingreso nuevo usuario

Id	Tipo de elemento	Descripción
1	Image View	Muestra imagen descriptiva
2	Edit. Text	Ingreso de texto del nombre de usuario
3	Edit. Text	Ingreso de texto del apellido de usuario
4	Text View	Texto descriptivo
5	Text View	Texto descriptivo con caracteres para ocultar contraseña
6	Button	Botón para agregar los usuarios a la lista
7	Button	Botón para salir

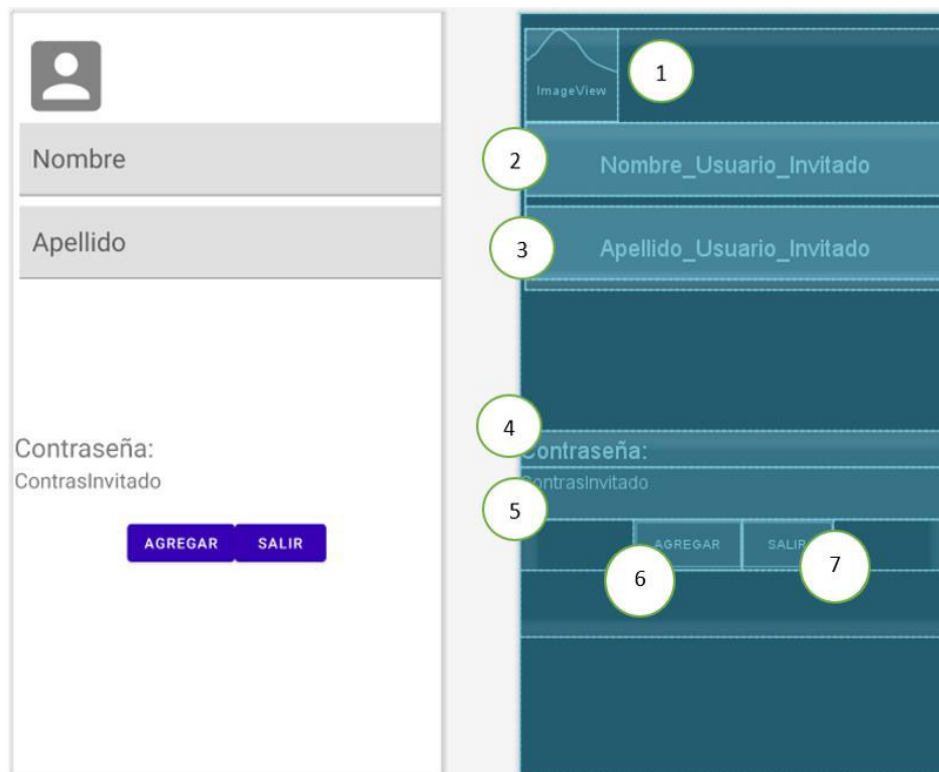


Figura 2.51 Estructura e interfaz ingreso de nuevo usuario

Como se observa en la Figura 2.51 y se detalla en la Tabla 2.19, la pantalla realiza el ingreso de nuevos usuarios invitados dentro del sistema, los cuales tendrán acceso a la puerta principal.

2.2.1.2 Implementación de Software de módulos ESP Arduino IDE

Dentro de los módulos ESP se definen algunas características de programación en Arduino IDE que son generales para todos los dispositivos tales como: librerías para la plataforma Firebase y librerías de WiFiManager, las cuales dan acceso a las credenciales y autenticación en los diferentes sistemas y plataformas.

Adicionalmente, dependiendo del dispositivo, se utilizan diferentes librerías para sensores que van a ser controlados por medio de los módulos ESP.

En la Figura 2.52 se presenta la inclusión de la librería de conexión a WiFi y su codificación para el acceso a Internet. En la Figura 2.53 se muestra la librería que conecta con la plataforma de Firebase, tanto para su autenticación, como para el envío de las notificaciones a la aplicación móvil.

```
#include <WiFi.h>
#include <WiFiManager.h>

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
WiFiManager wm;
bool res;
// res = wm.autoConnect(); // auto generated AP name from chipid
res = wm.autoConnect("AutoConnectAP"); // anonymous ap
//res = wm.autoConnect("AutoConnectAP","password"); // password protected ap

if(!res) {
    Serial.println("Failed to connect");
    // ESP.restart();
}
else {
    //if you get here you have connected to the WiFi
    Serial.println("connected...yeey :)");
}
```

Figura 2.52 Librería WiFiManager

```

#include <FirebaseESP32.h>

String FIREBASE_HOST = "sistema-de-seguridad-3a89d-default-rtdb.firebaseio.com";
String FIREBASE_AUTH = "qBAeyVPvWodcpSO2KQYYTCLoSPOSVn4YVxLS8wb3";
#define FIREBASE_FCM_SERVER_KEY "AAAAVImYOsY:APA91bGiCGgS93w9pXqiGYdsZqsrq_SXtkB
#define FIREBASE_TOKEN "f55CovlPRJaUu94kYjjqfc:APA91bE5D2lFBOYN_-Bw6wkKgOSY0kO4V

Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
Firebase.reconnectWiFi(true);
Firebase.setMaxRetry(firebaseData, 3);
Firebase.setMaxErrorQueue(firebaseData, 30);
Firebase.enableClassicRequest(firebaseData, true);

//agregando el servicio de notificaciones FCM
firebaseData.fcm.begin(FIREBASE_FCM_SERVER_KEY);
firebaseData.fcm.addDeviceToken(FIREBASE_TOKEN);
firebaseData.fcm.setPriority("high");
firebaseData.fcm.setTimeToLive(1000);

```

Figura 2.53 Librería Firebase

En el dispositivo de control de acceso, el módulo que controla tanto la pantalla LCD16x2, como el sensor biométrico y el teclado matricial 4x3 contiene las librerías necesarias para acceder al control de cada uno. En la Figura 2.54 se muestran las bibliotecas ancladas en el programa para el control de sus respectivas variables y el uso adecuado de los mismos.

En este dispositivo se va a realizar el ingreso de todas las variables que se van a comparar para la apertura de la cerradura. Por lo tanto, la conexión, digitación y validación de los datos depende directamente de este módulo ESP. También, este módulo guiará de manera visual, sea para confirmación de acceso o denegación del ingreso por diferentes motivos. En la Figura 2.55 se observa parte de la codificación del ingreso del teclado y parte de la codificación de los botones para su uso, así como la comparación y verificación de la huella digital en el sensor biométrico.

```

#include "Keypad.h"
#include "Wire.h"
#include "LiquidCrystal_I2C.h"
#include <Adafruit_Fingerprint.h>
#include <HardwareSerial.h>

```

Figura 2.54 Librerías de control de acceso

```

char tecla_pulsada = keypad.getKey();
lcd.setCursor(z-1,1);
lcd.print("*");
if (tecla_pulsada){
    Serial.println(tecla_pulsada);
    switch(tecla_pulsada){
        case '*':
            z=0;
            break;
        case '#':
            delay(300);
            checkKEY();
            break;
        default:
            attempt_key[z]=tecla_pulsada;
            z++;
    }
}
////////////////////////////////////
if(digitalRead(scan_pin) && !id_ad)
{
    scanning = true;
    lcd.setCursor(0,0);
    lcd.print("  Ingrese su huella ");
    lcd.setCursor(0,1);
    lcd.print("Esperando -----");
}

while(scanning && counter <= 60)
{
    getFingerprintID();
    delay(100);
    counter = counter + 1;
    if(counter == 10)

```

Figura 2.55 Codificación de panel de acceso

Para la codificación del activador de la cerradura principal, se observa que se utilizan las bibliotecas generales para activar los pines que ejecutan la acción de apertura de la cerradura. No es necesario utilizar más bibliotecas, ya que se toman las variables directamente de la plataforma Firebase para su comparación y validación. También, se usa para el envío de los mensajes de notificación. En la Figura 2.56 se presenta parte de la codificación de la captura de información de la plataforma Firebase; mientras que en la Figura 2.57 se muestra parte de la codificación para el envío de los mensajes de notificación.

```

void loop() {
    String Path = "//seguridadJGT/dispositivos_anclados/138512";
    Firebase.getString(firebaseData, Path+"/control_apertura");
    String estado= firebaseData.stringData();
    if(estado == "abierto"){
        digitalWrite(2,LOW);
        digitalWrite(32,HIGH);
        Serial.println("la puerta se abrio");
        delay(2000);
        String Path = "//seguridadJGT/dispositivos_anclados/138512";
        Firebase.setString(firebaseData, Path+"/control_apertura", "cerrado");
    }
}

```

Figura 2.56 Codificación de apertura de cerradura

```

void enviarmensaje(){
    Serial.println("-----");
    Serial.println("Send Firebase Cloud Messaging...");

    firebaseData.fcm.setNotifyMessage("Notificacion", "Se ha abierto la puerta principal ");

    firebaseData.fcm.setDataMessage("{\"myData\":\" " + String(count) + " }");

    //if (Firebase.broadcastMessage(firebaseData))
    //Firebase.sendTopic(firebaseData);
    if (Firebase.sendMessage(firebaseData, 0))//send message to recipient index 0
    {
        guardarnotificacion();
        Serial.println("PASSED");
        Serial.println(firebaseData.fcm.getSendResult());
        Serial.println("-----");
        Serial.println();
    }
    else
    {
        Serial.println("FAILED");
        Serial.println("REASON: " + firebaseData.errorReason());
    }
}

```

Figura 2.57 Codificación mensaje desde Firebase a aplicación móvil

En la codificación de la cámara de seguridad, la plataforma de Firebase usa el envío de las imágenes secuenciales para recibir desde el módulo ESP y transmitir a la aplicación móvil. Por lo tanto, no se usan más librerías adicionales que las de conexión a Internet y a la plataforma.

2.2.2 FASE IMPLEMENTACIÓN HARDWARE

2.2.2.1 Implementación Dispositivos Electrónicos

2.2.2.1.1 Cámara de Videovigilancia

En este dispositivo se utiliza una ESP32-CAM, que permite el envío de las imágenes hacia la plataforma Firebase para luego ser tomadas por la aplicación móvil. Este dispositivo cuenta con un sensor de presencia PIR, el cual, al detectar cualquier movimiento enviará notificaciones. Usa una fuente a 5 V para el encendido del circuito electrónico.

En este caso se realiza la conexión del sensor PIR con la cámara como se muestra en la Figura 2.58, de tal manera que al detectar un movimiento de manera externa a través de GPIO 15 (HIGH), se capture la imagen y se envía una notificación a la aplicación móvil. A través del GPIO 4, se accede a la luz led que incorpora el ESP32, así, se puede capturar la escena en movimiento con mayor claridad. En la Figura 2.59 se observa el diseño de la placa PCB para la conexión de la cámara antes de su ensamblaje.

Para ensamblar la cámara de seguridad se usa la placa y cable molex para el sensor PIR; en la Figura 2.60 se muestra el módulo y la conexión al sensor, siendo éstos ubicados en su respectiva caja. En la Figura 2.61 se muestra la parte frontal y posterior de la cámara respectivamente.

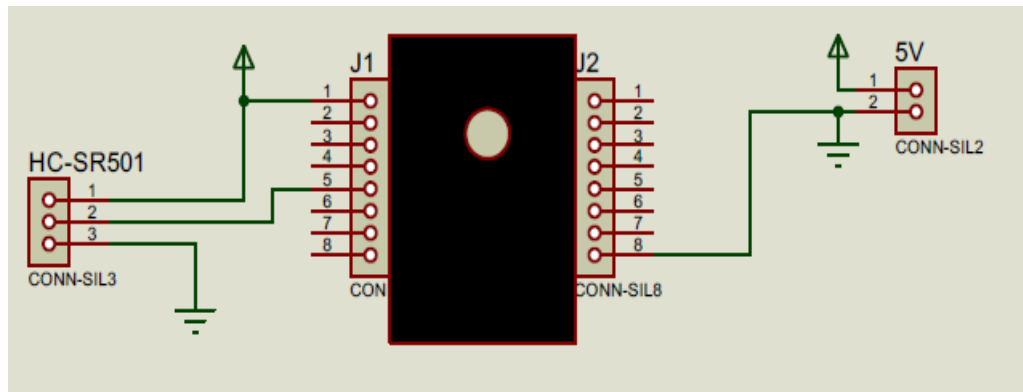


Figura 2.58 Conexión PIR a ESP32-CAM

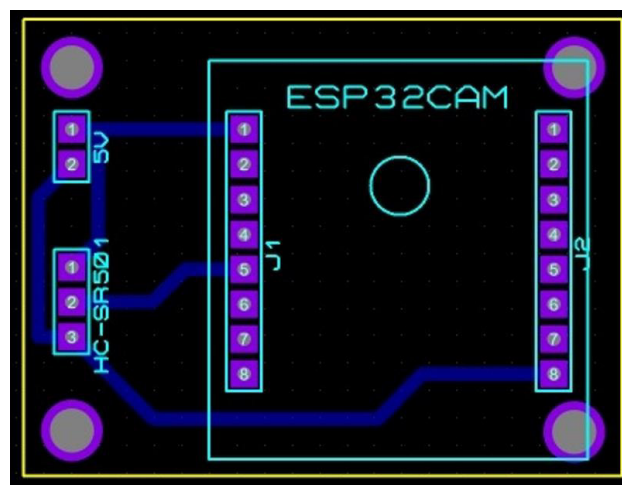


Figura 2.59 Placa PCB ESP32-CAM



Figura 2.60 Ensamblaje de cámara con sensor PIR

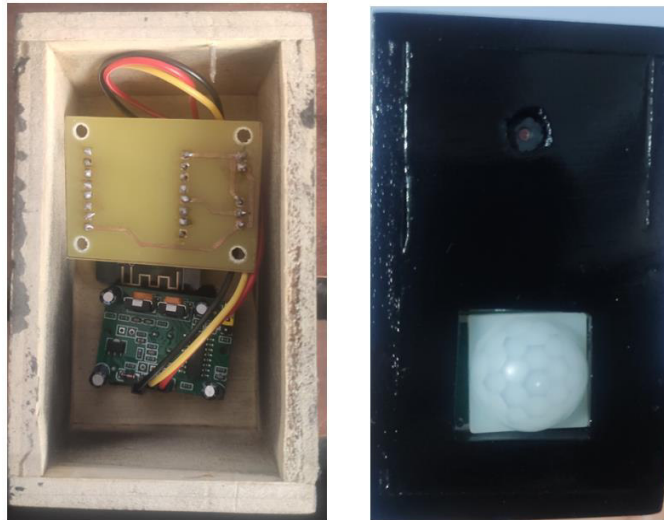


Figura 2.61 Vistas de Cámara, frontal y posterior

2.2.2.1.2 Control Activador Puerta Principal

En este dispositivo se utiliza un módulo ESP32 que permite la adquisición de los datos del sensor de la cerradura, activación y apertura de la chapa. También usa una fuente de 5V para el encendido del circuito electrónico.

Como primer paso se realizaron las conexiones de los sensores en modo *pull-down*, esto quiere decir que al estar el circuito en modo de reposo la caída de tensión es de 0V (LOW) y al detectar la activación del sensor se tiene un voltaje de 5V (HIGH). Lo que permite este circuito es si por alguna razón se corta el cable de los sensores éste detectaría un voltaje de 0V que significa el desbloqueo de la llave de las puertas o cerradura para su apertura manual.

Después de realizar las conexiones de los sensores se realiza el diseño de los circuitos para la activación de los relés los cuales controlarán la apertura de la chapa y la apertura de las celdas de carga que están instaladas en las puertas. Este circuito es similar a los anteriores implementados en los otros dispositivos y será conectado al pin GPIO 2 de la ESP32 como se muestra en la Figura 2.64.

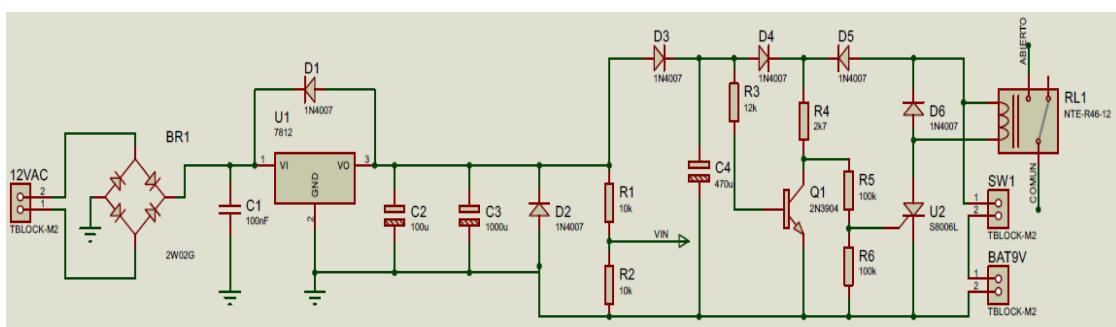


Figura 2.62 Conexión circuito falla de energía

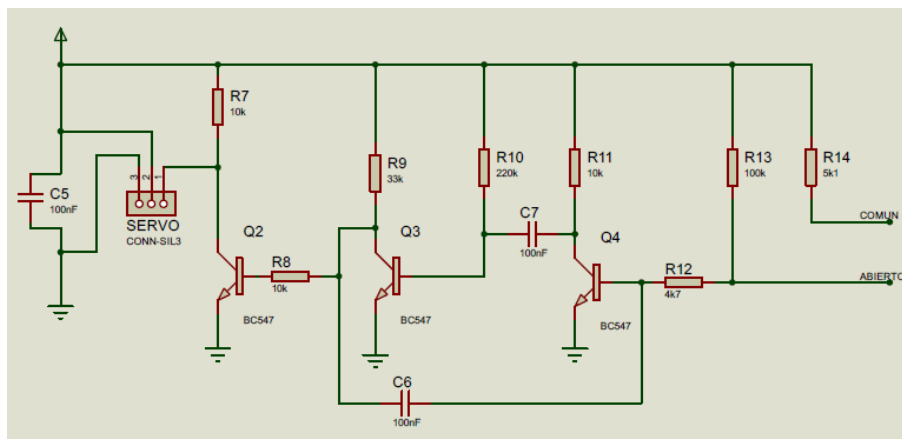


Figura 2.63 Control bloqueo de llave de cerradura

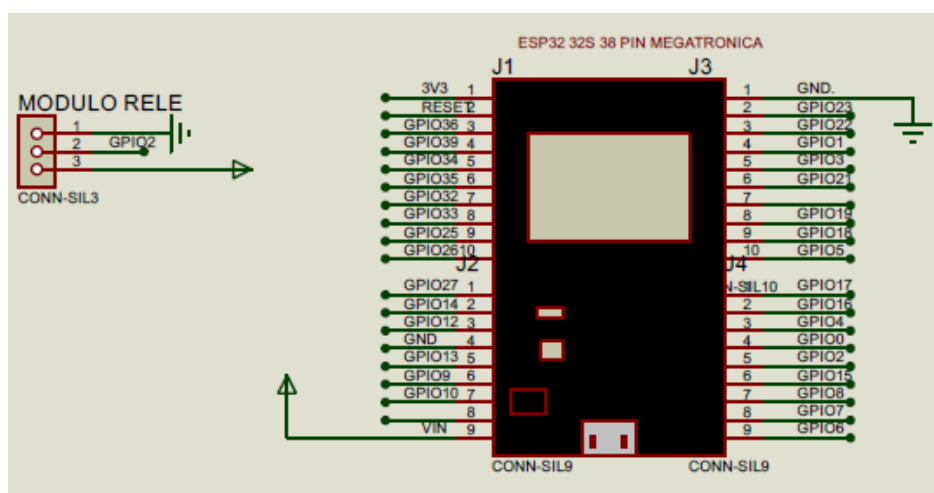


Figura 2.64 Dispositivo de apertura de puerta principal

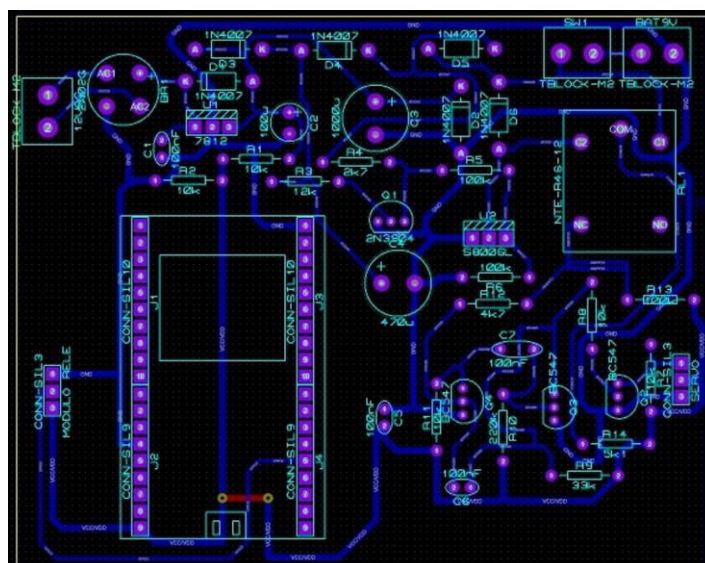


Figura 2.65 Diseño de placa PCB activador cerradura

En la Figura 2.66 se indica como queda la placa una vez instalada junto a los elementos para el activador de la puerta principal. Se incluye el sistema para desactivar el bloqueo del seguro en caso de fallas de energía para poder acceder a la puerta principal a través de la llave.

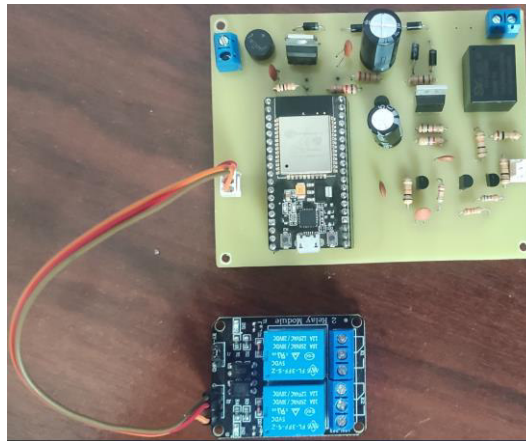


Figura 2.66 Placa de activador cerradura

2.2.2.1.3 *Diseño del dispositivo de control de acceso*

En este dispositivo se utiliza un módulo ESP32 que permitirá el manejo del teclado de 4x3, una pantalla LCD de 16x2 y un sensor biométrico. También usa una fuente de 5 V y 1000 mA para el encendido del circuito electrónico y un led indicador del estado de la red WiFi.

Para el manejo del teclado de 4x3 se utilizó 7 pines digitales los cuales decodifican el pulsador presionado. Para esta conexión se usó los pines GPIO 32,33,25,26,27,14 y 12 como se muestra en la Figura 2.67.

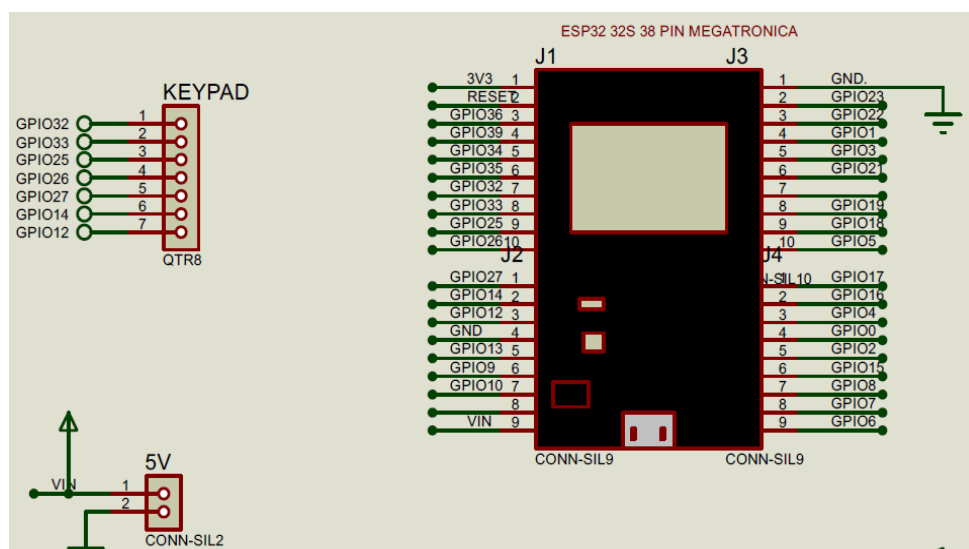


Figura 2.67 Conexión del teclado a la ESP32

Para el manejo de la LCD de 16 x 2 se utiliza un módulo de conexión I2C el cual necesita los pines SDA, SCL, GND y VCC. Para esto se conecta el pin 22 de la ESP hacia el SDA y el pin 21 al SCL de la pantalla LCD. En la Figura 2.68 se indican las conexiones de la pantalla LCD.

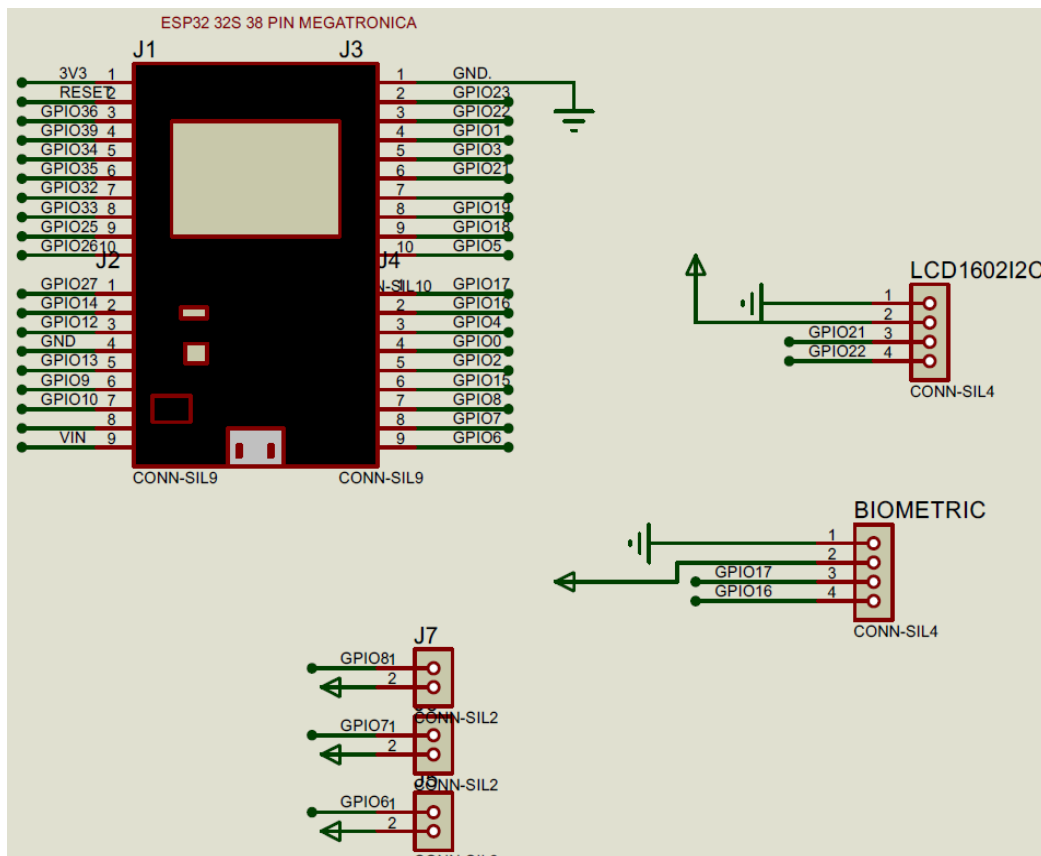


Figura 2.68 Conexión de la ESP32 hacia la pantalla LCD 16x2.

Adicionalmente, en la Figura 2.68, se muestra la conexión del sensor biométrico a los pines GPIO 16,17, los cuales funcionan como receptor y transmisor para el envío de las imágenes de la huella digital para su posterior comparación o ingreso. Por otra parte, se tiene pulsadores conectados en modo “pull-down” para la actualización de la tabla de huellas y el ingreso a posterior de nuevas entradas.

En la Figura 2.69 se observa el circuito completo para el manejo del dispositivo de control de acceso del sistema de seguridad.

Una vez obtenido el esquema general del circuito para el control de acceso se procede a realizar el diseño de la placa PCB para la conexión a posterior de los elementos y su ensamblaje. En la Figura 2.70 se muestra el diseño.

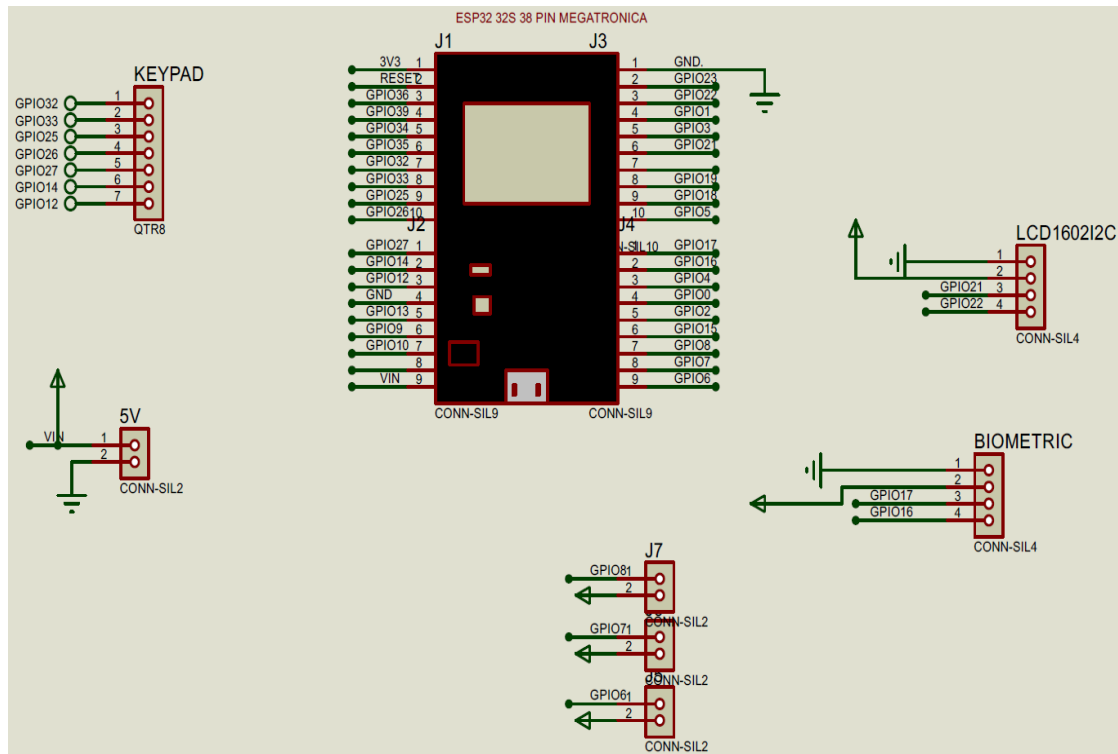


Figura 2.69 Circuito del control de acceso.

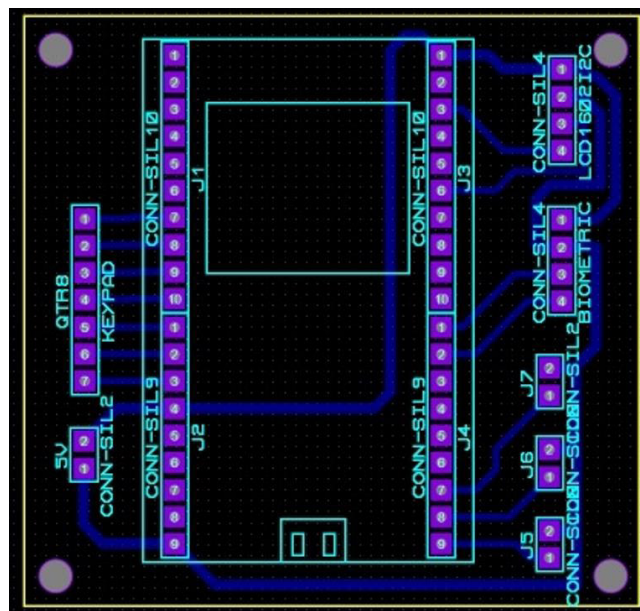


Figura 2.70 Placa PCB de control de acceso

Una vez impreso el circuito, se procede a soldar los elementos en la placa junto con los demás sensores y pantalla en su respectiva caja. En las Figuras 2.71 y 2.72 se muestran cómo queda desde la parte frontal y posterior el circuito instalado.



Figura 2.71 Parte posterior circuito control acceso



Figura 2.72 Parte frontal control acceso

2.2.3 MONTAJE EN EL LUGAR A SUPERVISAR

Una vez que los dispositivos se encuentran ensamblados y probados, se instalan en el ambiente a supervisar. Como se puede observar en la Figura 2.73, el montaje del dispositivo de control se lo realiza a un lado de la puerta principal y su conexión se toma desde la parte interna del domicilio, evitando que se pueda manipular los cables de energía.



Figura 2.73 Dispositivo Control de Acceso

El dispositivo activador de la puerta, con el cual se maneja la apertura, se lo ubica en la parte interna del domicilio, como se muestra en la Figura 2.74; las conexiones se toman desde el transformador a 12 VAC, ya que este voltaje es el utilizado por la cerradura magnética como mecanismo de apertura.



Figura 2.74 Dispositivo activador de apertura

La cámara de videovigilancia se ubica a un costado de la puerta principal para garantizar que se tenga una imagen clara al momento de revisar su video, así como para la captura

de imágenes. El dispositivo cuenta con una caja bastante pequeña, de tal manera que se hace menos perceptible en el ambiente ubicado (ver Figura 2.75).



Figura 2.75 Dispositivo Cámara

En el lugar a supervisar todo el sistema de seguridad se encuentra propuesto como se muestra en la Figura 2.76 en la que se observa el dispositivo de control de acceso y la cámara de seguridad, ya que el dispositivo activador se encuentra ubicado en la parte interna del lugar.



Figura 2.76 Dispositivos del sistema de seguridad instalados

3. RESULTADOS Y DISCUSIÓN

En esta sección del Trabajo de Titulación ya se ha concretado el montaje del Sistema de Seguridad en el ambiente a supervisar, por lo tanto, se procede con la fase de pruebas de funcionamiento. Como se muestra en la Figura 3.1 el tablero Kanban se encuentra con las tarjetas de prueba en la sección de desarrollo.



Figura 3.1 Tablero Kanban en Pruebas de Funcionamiento

3.1 PRUEBAS EN APLICACIÓN MÓVIL

3.1.1 REGISTRO Y ACCESO USUARIO

El uso de la plataforma de Firebase se realiza a partir de cualquier cuenta de Gmail, no es necesario utilizar un plan de pago, a menos que se requiera cierta cantidad de descargas o cargas, al igual que la sección de funciones automáticas en la plataforma. Por lo tanto, la interfaz de la plataforma Firebase y sus componentes ya se encuentran habilitados.

Para el acceso al sistema los usuarios pueden registrarse con un correo electrónico y contraseña, permitiendo su autenticación en la plataforma Firebase. Esta autenticación da paso a la creación del usuario en la base de datos.

Se debe crear la cuenta en el sistema para su autenticación a posterior. En la Figura 3.2 se muestra la pantalla principal con las opciones de inicio de sesión y de registro y/o creación de una nueva cuenta.

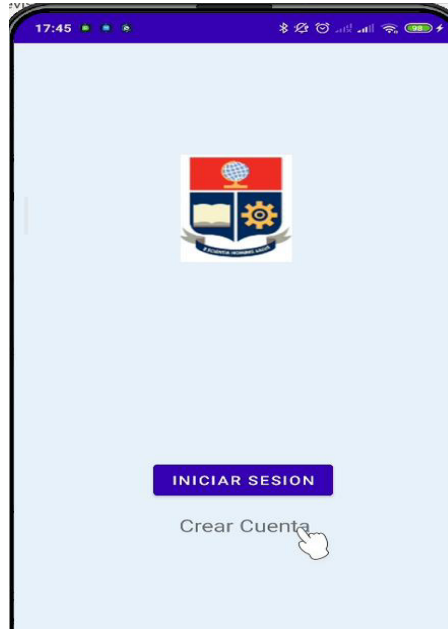



Figura 3.2 Inicio de sesión y Creación de usuario desde la aplicación

En la creación de la cuenta se puede tener varios casos de error. En la Figura 3.3 se muestra cuando las contraseñas que se están ingresando no coinciden. En la Figura 3.4 se observa el caso cuando un usuario no ha completado uno de los campos del formulario de solicitud. En caso de que se quiera utilizar el mismo correo para dos cuentas diferentes, el sistema no permite dicha autenticación dentro de la plataforma de Firebase. En la Figura 3.5 se presenta el mensaje necesario para la información al usuario.

El registro de autenticación de los usuarios se lo realiza a través de la aplicación, pero se añade también al control de autenticación de Firebase, ya que para el acceso hacia la base de datos se tienen reglas de seguridad, las cuales incluyen que el usuario se haya autenticado dentro de la plataforma. A continuación, se muestra cómo se genera el registro de los usuarios en la plataforma de Firebase en el registro de autenticación.

La plataforma de Firebase trabaja con bases de datos no relacionales, por lo tanto, la estructura de los datos se presenta en la Figura 3.6. Cada uno de los nodos que se encuentran especificados en la Figura 3.7 son determinados al momento de la programación en Android Studio o en los dispositivos directamente asociados.



Nombre
Rosalia

Apellido
Tapia

Correo
rosaliajll3@gmail.com

Contraseña
.....


Confirmar contraseña
.....

SALIR REGISTRAR

Password no coincide

Falló el ingreso

Figura 3.3 Fallo de contraseñas



Nombre
Rosalia

Apellido

Correo
rosaliajll3@gmail.com

Contraseña
.....

Confirmar contraseña
.....

SALIR REGISTRAR

Falló el ingreso

Figura 3.4 Campo vacío



Formulario de registro de usuario con los siguientes campos:

- Nombre: Rosalia
- Apellido: Tapia
- Correo: rosal@gmail.com
- Contraseña: [oculto]
- Confirmar contraseña: [oculto]

Botones: SALIR, REGISTRAR

Mensaje: El usuario ya esta registrado

Figura 3.5 Correo ya registrado en otro usuario

Buscar por dirección de correo electrónico, número de teléfono o UID de usuario					Agregar usuario	
Identificador	Proveedores	Fecha de creación	Fecha de acceso	UID de usuario		
jdav@gmail.com	✉	25 sep. 2021	25 sep. 2021	zCxs7PUXKYPvweIQnW2rbsihTKH3		
rosal@gmail.com	✉	24 sep. 2021	4 oct. 2021	R3onjUYybrYiDDO2DrBrOdoGbLt1		

Figura 3.6 Registro de usuarios autenticados en Firebase

<https://sistema-de-seguridad-3a89d-default-rtdb.firebaseio.com/>

sistema-de-seguridad-3a89d-default-rtdb

- esp32-cam
- seguridadJGT
 - dispositivos_no_anclados
 - usuario
 - R3onjUYybrYiDDO2DrBrOdoGbLt1
 - apellido: "Lopez"
 - contrasena_iniciosesion: "123456789"
 - contrasena_puerta: "12341"
 - correo: "rosal@gmail.com"
 - dispositivo
 - id: "R3onjUYybrYiDDO2DrBrOdoGbLt1"
 - llave_notificaciones: "f55Cov1PRJaUu94kYjjqfc:APA91bHufKou20cA7gU2LDbf"
 - nombre: "Rosal"
 - telefono: 0
 - usuario_invitado

Figura 3.7 Creación de usuario en la base de datos Firebase

Cada uno de los usuarios que se añaden como invitados dentro del sistema tienen su contraseña para el ingreso a la puerta principal y el almacenamiento de la huella digital se lo realiza de manera directa en el tablero.

3.1.2 NAVEGACIÓN EN PANTALLA PRINCIPAL

La navegación entre pantallas dentro de la actividad principal se realiza a través de un *navigation buttom*. Se tienen tres opciones (ver Figura 3.8) en las cuales el usuario puede seleccionar para dirigirse a su vista como son: El fragmento “Inicio”, que es donde se aloja la lista de los dispositivos que se encuentran anclados dentro del sistema de seguridad. El fragmento “Notificaciones” donde se aloja la lista de alertas enviadas por los dispositivos al usuario en caso de tener algún evento que produzca un cambio en su estado. El fragmento “Perfil” que se encarga de mostrar los datos del usuario y el botón para agregar un usuario invitado en caso de ser requerido.

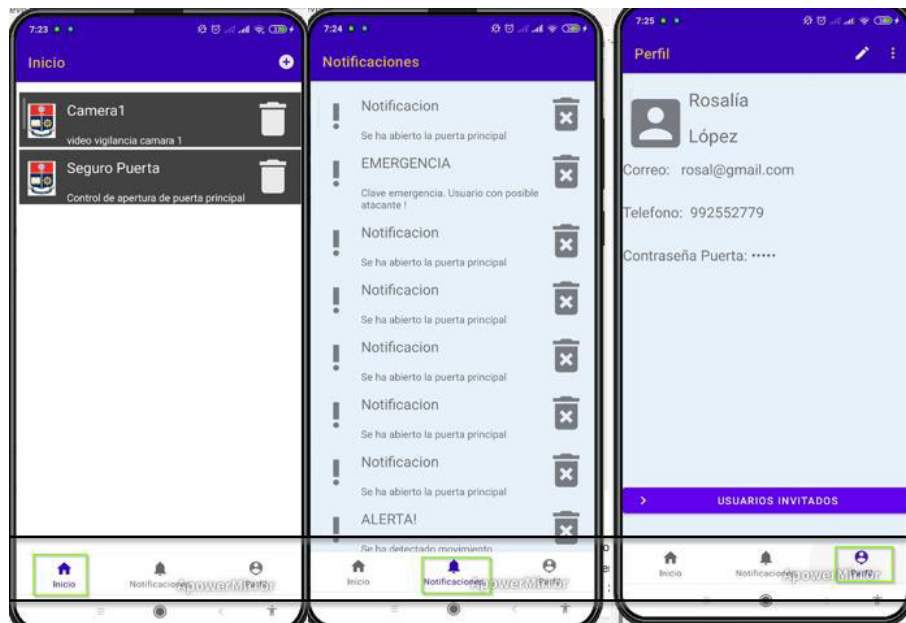


Figura 3.8 Barra de Navegación y Fragmentos en Actividad Principal

3.1.3 CREACIÓN DE USUARIOS INVITADOS

Se tiene un fragmento que direcciona a los datos del perfil de usuario administrador tal como se muestra en la Figura 3.9, así como también, un botón para observar la lista de los usuarios invitados y la posibilidad de añadir más usuarios. En la Figura 3.11 se observa la creación de un usuario invitado, quien tendrá su propia clave de acceso para la puerta principal. Por otra parte, en la Figura 3.10 se presenta la lista de los usuarios invitados.

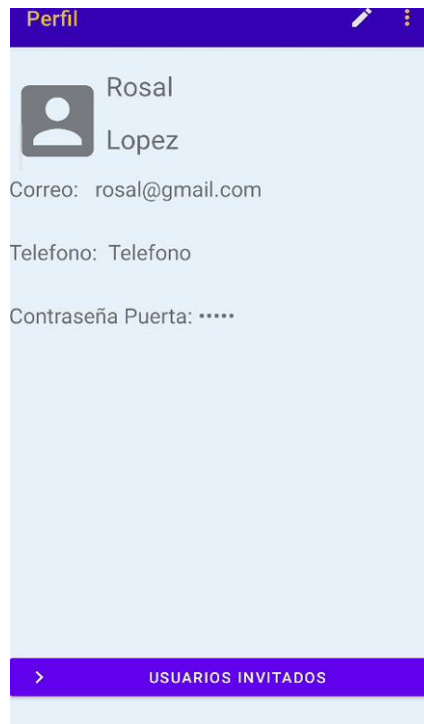


Figura 3.9 Perfil Usuario registrado

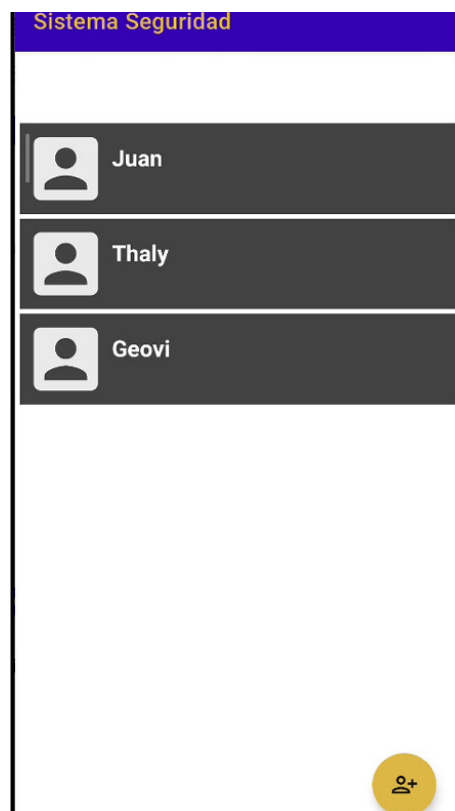


Figura 3.10 Usuarios Invitados

Sistema Seguridad

Nombre

Apellido

Contraseña:
12676

Contraseña puerta principal para usuario invitado

AGREGAR SALIR

Figura 3.11 Creación usuario invitado

3.2 PRUEBAS EN DISPOSITIVOS

Todos los dispositivos son ensamblados a través del módulo ESP32, el cual envía los datos hacia la plataforma Firebase. Los dispositivos tienen variables generales para su reconocimiento, así como: id, nombre, descripción, estado, las cuales indican el dispositivo que se ha conectado a la red y reconocido en la base de datos. Adicionalmente, se tienen las variables por el tipo de dispositivo, es decir, las variables de manejo del dispositivo.

Para el acceso a la red WiFi, se utiliza la librería WiFiManager, la cual permite usar la red disponible más cercana para anclar los módulos ESP sin tener una configuración previa en el módulo. Con esta librería se accede a través de la interfaz web y se selecciona la red a la cual se quiere conectar, luego se ingresa la clave de la red y el dispositivo se reinicia con las nuevas credenciales.

En la Figura 3.12 se tiene la página principal para acceder a las redes disponibles. Por otro lado, en la Figura 3.13 se observa la configuración de las credenciales de la red a la cual se va a acceder, así como también, el reinicio del dispositivo y la autenticación con las nuevas credenciales.

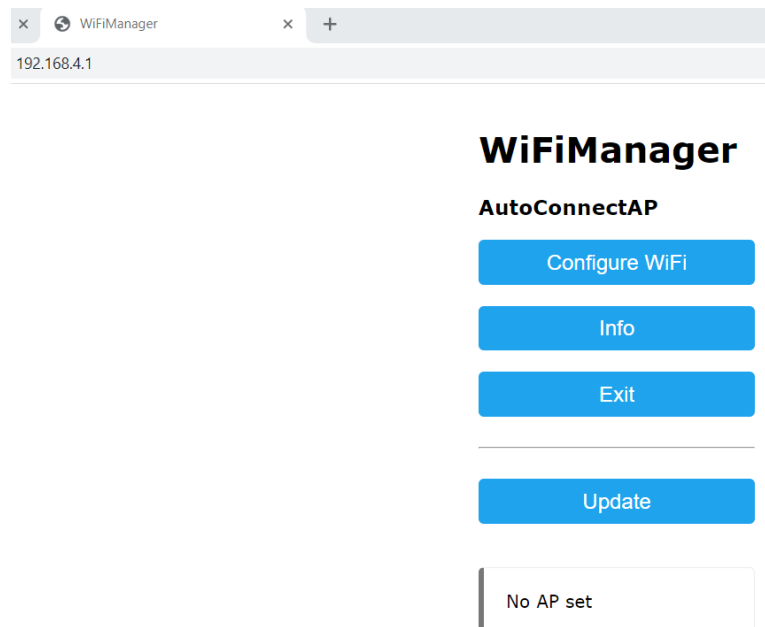


Figura 3.12 Página principal WiFiManager

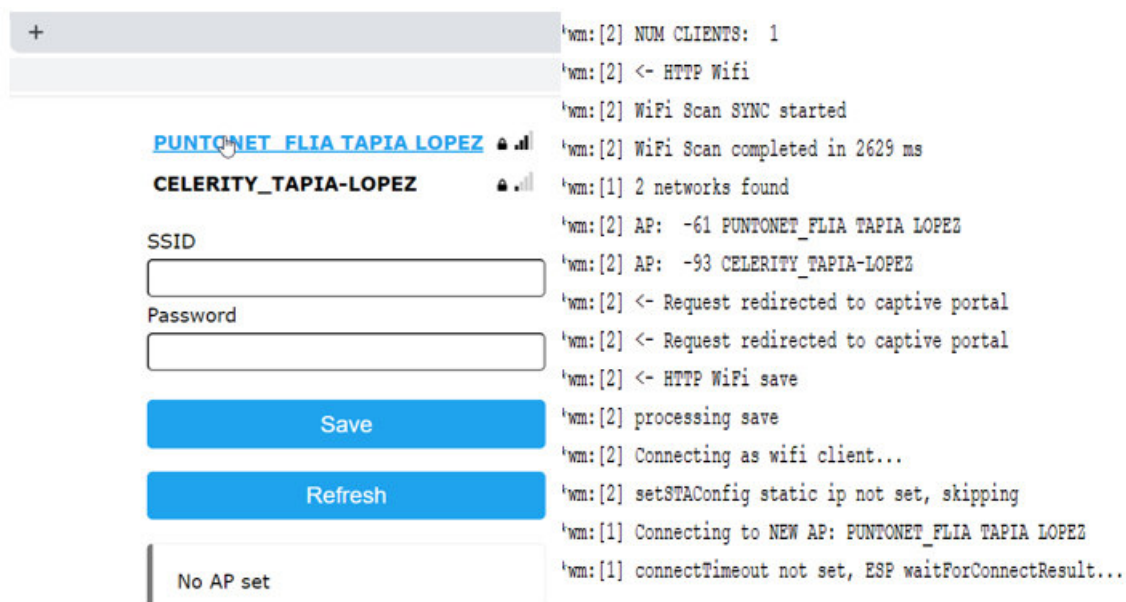


Figura 3.13 Configuración de credenciales a la red

3.2.1 CÁMARA DE SEGURIDAD

En la cámara de seguridad se tiene una variable en donde se procesan las imágenes que se cargan y descargan dentro de la plataforma Firebase. Por lo tanto, se van a observar los sucesos que ocurren en el ambiente donde se encuentra instalada.

Los dispositivos se generan dentro de la base de datos en la sección de “dispositivos no anclados” (ver Figura 3.14) una vez que se han conectado a la energía eléctrica. Como se muestra en la Figura 3.15 se presenta un mensaje de confirmación antes de que los dispositivos sean agregados a la lista de “dispositivos_anclados” y poder configurarlos.



Figura 3.14 Plataforma Firebase dispositivo cámara No anclado

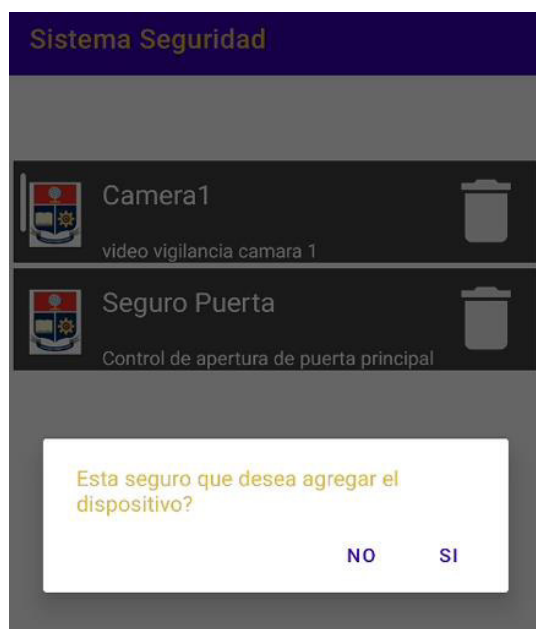


Figura 3.15 Dispositivo no anclado en aplicación móvil

Una vez que se acepta el ingreso del dispositivo en la aplicación se puede tener acceso a la pantalla de la cámara de seguridad. En esta actividad se pueden visualizar las imágenes del video y capturar imágenes a través del botón de “captura” para mantener esas imágenes en la plataforma Firebase.

En la base de datos de Firebase, se tienen los campos para la emisión de las imágenes del video mostrado en la aplicación (ver Figura 3.16). Por otra parte, se tiene la variable que almacena cada una de las capturas que se quieran realizar mientras se está visualizando el video en la aplicación móvil. En la Figura 3.17 se puede observar cómo se encuentran almacenadas las imágenes del botón captura.



Figura 3.16 Acceso a Pantalla de cámara

```
id: "R3onjUYybrYiDD02DrBr0doGbLt1"
imagen_capturada
  -MqLtQMG6EwFzMflrCmf: "data:image/jpeg;base64,%2F9j%2F4AAQSkZJRgABAQEA..."
```

Figura 3.17 Imagen capturada por botón en aplicación móvil

Las imágenes capturadas que se muestran en la Figura 3.17 dentro de la plataforma Firebase en la base de datos, también se muestran en la aplicación móvil en un contenedor bajo la proyección del video de la cámara. Se puede observar en la Figura. 3.18 una de las imágenes capturadas por parte del usuario.

En la Figura 3.19 se observa la acción del sensor de movimiento en la cámara para el envío de notificaciones hacia la aplicación móvil. Este proceso es repetitivo correspondiente a los movimientos dentro del sistema.



Figura 3.18 Imagen Capturada por el usuario

```
connected...yeey :)
Camera Stream Ready! Go to: http://192.168.1.102

sensor activo
-----
Send Firebase Cloud Messaging...
PASSED
{"multicast_id":7401055675619718190,"success":1,"failure":0,'
-----
```

Figura 3.19 Envío de notificación por sensor de movimiento

Se tiene una acción combinada dentro del sistema que permite tomar capturas de imágenes y mostrarlas en las notificaciones cuando se ha ingresado de manera errónea la contraseña por 3 veces, o si se ha ingresado la clave de emergencia por medio del teclado.

Como se muestra en la Figura 3.20, se tiene un contador que irá aumentando conforme la contraseña haya sido ingresada de manera incorrecta, una vez que llega a la tercera ocasión, se accede al método que solicita una captura de la imagen por parte de la cámara de seguridad y el envío de la notificación hacia la aplicación móvil. De la misma manera, se envía la solicitud hacia la cámara de vigilancia si la contraseña ingresada es la señalada como emergencia y se muestra en la Figura 3.21.

```

{
  lcd.setCursor(0,1);
  lcd.print("Clave Incorrecta");
  delay(3000);
  z=0;
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Ingrese su clave");
  if(contador_error!=3){
    contador_error++;
  }else{
    envioSolicitudCaptura();
    contador_error=0;
  }
}

```

Figura 3.20 Código para error de contraseña hasta 3 veces

```

}else if(contras_emergencia == clave_numerica){
  Serial.println("La comparacion es correcta");
  lcd.setCursor(0,1);
  lcd.print("Clave correcta");
  mensajeEmergencia();
  delay(2000);
  getFingerprintID();
  z=0;
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Ingrese su clave");
  envioSolicitudCaptura();
}

```

Figura 3.21 Solicitud Captura por contraseña de Emergencia

En la Figura 3.22 se muestra el lazo en el dispositivo de la cámara de seguridad que reacciona a la solicitud de captura del control de acceso para la toma de captura de imágenes según la opción de contraseña errada o la contraseña de emergencia. Una vez que se captura la imagen, se envía a través de una notificación hacia la aplicación móvil.

Las notificaciones que se envían cuando existe una solicitud por parte del control de acceso guardan una captura de la imagen del momento en que se encuentra realizando la acción el usuario. Una vez que se ha digitado por tercera vez consecutiva la contraseña de forma errada, se captura la imagen para luego ser guardada en la plataforma Firebase. En la Figura 3.23 se muestra el código de la notificación de contraseña errada, así como también el envío a la plataforma Firebase con un formato JSON de la notificación a guardarse.

```

void loop() {
  String Path = "//seguridadJGT/dispositivos_anclados/84651";
  Firebase.getString(firebaseData, Path+"/capturar_emergencia");
  String estadoem= firebaseData.stringData();
  if(estadoem == "true"){
    envioMensajeError();
    delay(2000);
    String Path = "//seguridadJGT/dispositivos_anclados/138512";
    Firebase.setString(firebaseData, Path+"/capturar_emergencia", "false");
  }
  Firebase.getString(firebaseData, Path+"/capturar_err_contrasena");
  String estadoerr= firebaseData.stringData();
  if(estadoerr == "true"){
    envioMensajeEm();
    delay(2000);
    String Path = "//seguridadJGT/dispositivos_anclados/138512";
    Firebase.setString(firebaseData, Path+"/capturar_err_contrasena", "false");
  }
  delay(1000);
}

```

Figura 3.22 Captura de imagen por solicitud de control de acceso

En la Figura 3.24 se observa la notificación recibida en el teléfono Android, así como también, la notificación en el panel de notificaciones de la aplicación móvil ; mientras que en la Figura 3.25 se presenta la notificación guardada en la plataforma Firebase para su lectura a posterior.

```

void envioMensajeError() {
  Serial.println("-----");
  Serial.println("Send Firebase Cloud Messaging...");

  firebaseData.fcm.setNotifyMessage("ERROR", "La contraseña ha sido errada por 3 ocasiones");

  firebaseData.fcm.setDataMessage("{\"myData\": " + String(count) + "}");

  //if (Firebase.broadcastMessage(firebaseData))
  Firebase.sendTopic(firebaseData);
  if (Firebase.sendMessage(firebaseData, 0))//send message to recipient index 0
  {
    guardarNotificacionError();
    Serial.println("PASSED");
    Serial.println(firebaseData.fcm.getSendResult());
    Serial.println("-----");
    Serial.println();
  }
  else
  {
    Serial.println("FAILED");
    Serial.println("REASON: " + firebaseData.errorReason());
    Serial.println("-----");
    Serial.println();
  }
  count++;
}

void guardarNotificacionError() {
  idrandom=random(100,99999999);
  String id = (String) idrandom;
  String pathNoti = "/seguridadJGT/usuario/mensajes_notificacion";
  String jsonDato = id;
  String title = "ERROR";
  String smssec = "Error de Clave. Verificar usuario !";
  String img = "asvds";
  json2.add("id", jsonDato);
  json2.add("titulo", title );
  json2.add("mensajeSecundario", smssec );
  json2.add("imagen", img);
  Firebase.pushJSON(firebaseData, pathNoti, json2);
}

```

Figura 3.23 Código de notificación de Contraseña errada

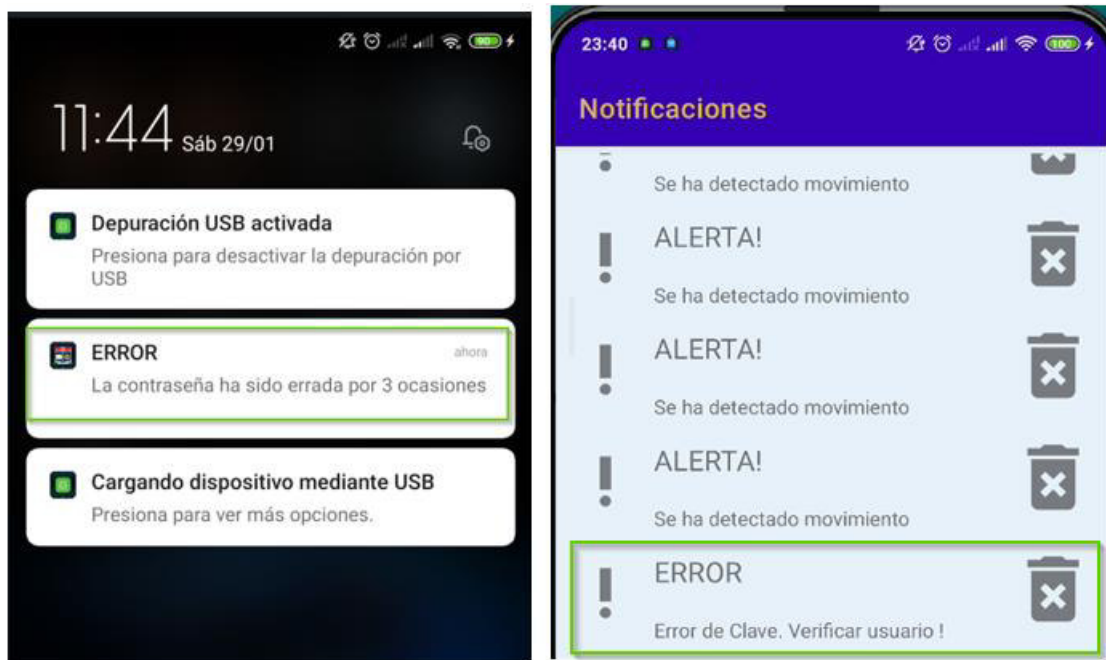


Figura 3.24 Notificación por error en la contraseña

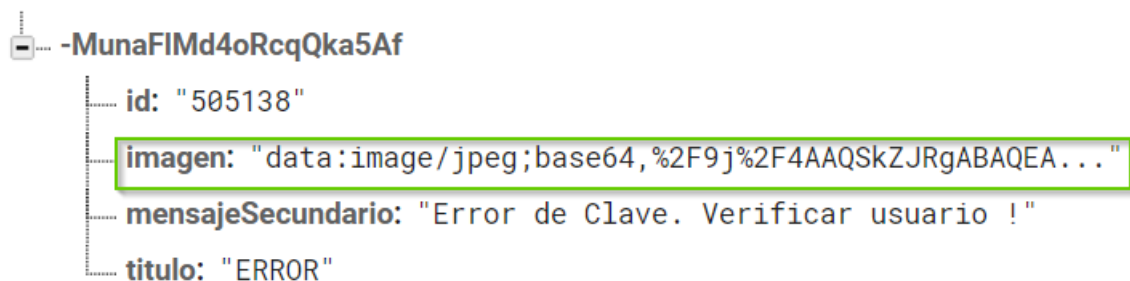


Figura 3.25 Notificación en base de datos Firebase

De la misma manera se realiza para las notificaciones por clave de emergencia. Por lo tanto, se va a recibir notificaciones desde el control de acceso y desde la cámara en caso de que sea la clave de emergencia la que se haya ingresado, siendo las dos primeras notificaciones desde el control de acceso y la tercera notificación desde la cámara de seguridad, en la cual se guarda la imagen capturada.

3.2.2 CERRADURA PUERTA PRINCIPAL

3.2.2.1 Activación

La cerradura de la puerta principal se encuentra gestionada por dos ESP32. En el primer módulo se conectan los mandos de control de la puerta como la pantalla digital, el sensor biométrico para huella digital y el teclado matricial para el ingreso de la clave personal. En el segundo módulo se encuentra conectada la cerradura eléctrica por medio de un relé, el cual acciona su apertura de manera automática, así como el paso de corriente durante un

periodo de tiempo. Como parte del segundo módulo se tiene un detector de fallas de corriente para poder desbloquear el acceso con la llave de la cerradura.

En la Figura 3.26 se puede observar el estado del dispositivo cuando se encuentra recién ingresado a la base de datos al momento de su conexión física en la sección de dispositivos no anclados; mientras que en la Figura 3.27 se muestra cuando el dispositivo ya se encuentra anclado a un cliente.

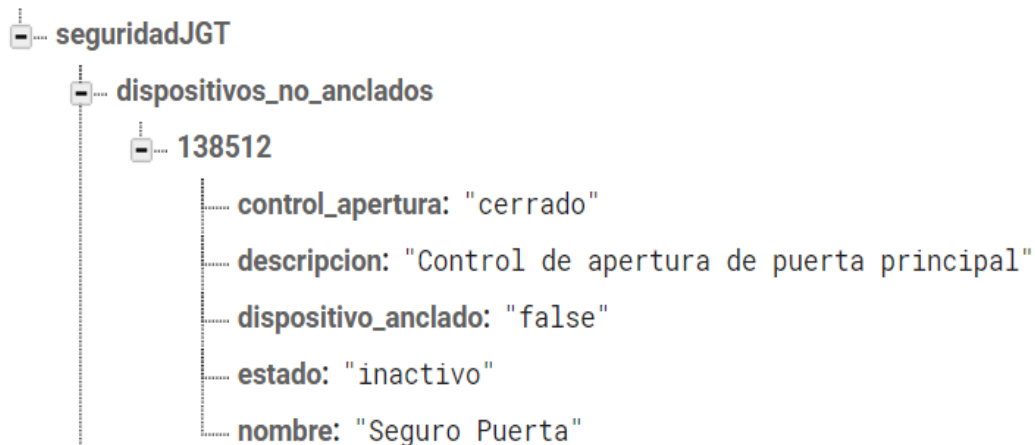


Figura 3.26 Dispositivo no anclado a ningún usuario

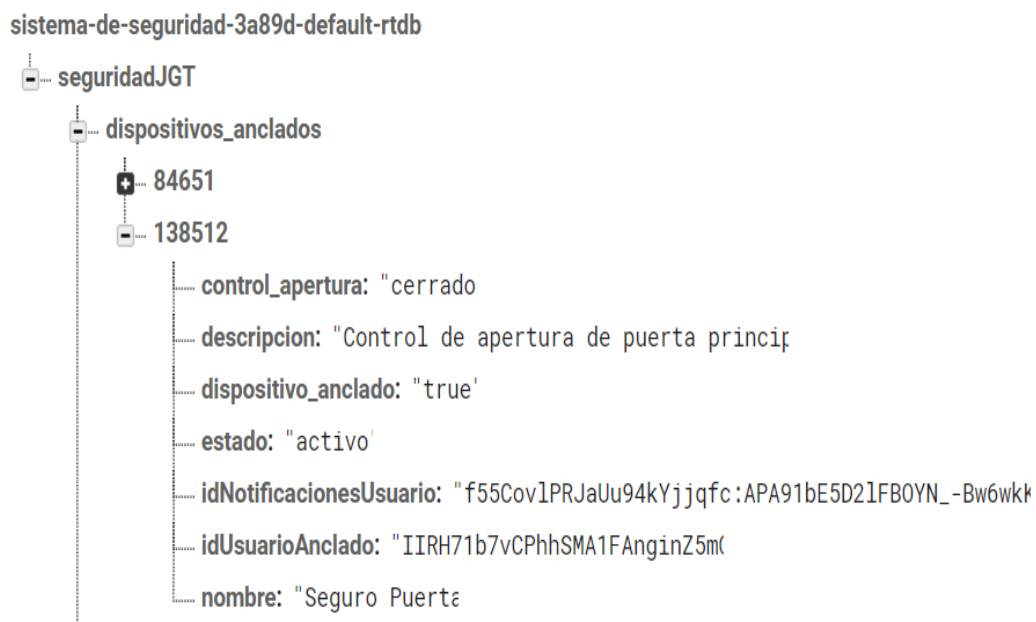


Figura 3.27 Dispositivo anclado a una cuenta

Dentro del esquema, el relé de funcionamiento se activa y desactiva tanto por el módulo de acceso principal que será el teclado y el sensor biométrico, así como también por medio de la aplicación móvil.

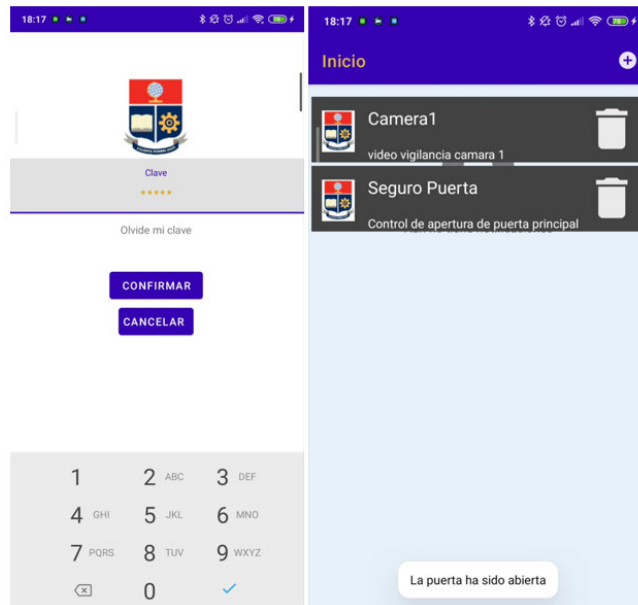


Figura 3.28 Ingreso y confirmación de apertura desde aplicación

En la Figura 3.28 se observa la pantalla de autenticación biométrica del teléfono Android, donde se puede apreciar el panel para el ingreso de la clave personal; cuando ésta es correcta se cambia el valor de la variable “control_apertura” en la base de datos de la Plataforma Firebase, de tal manera que el relé sea activado como se muestra en las Figuras 3.29 y 3.30.



Figura 3.29 Encendido/apagado de relé de apertura

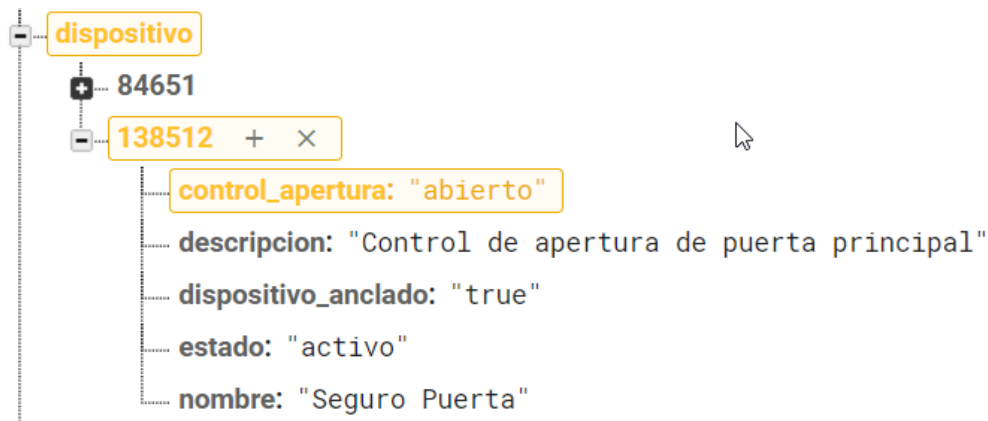


Figura 3.30 Variable activa para apertura de cerradura

Al realizar el ingreso de manera local, es decir a través del teclado y el sensor biométrico en la pantalla se muestra a través de “*” el ingreso de los números digitados para la clave. A continuación, por medio de la tecla “#” se confirma el ingreso completo de la clave y se procede a su verificación.

En el control de acceso de la cerradura se tienen 3 opciones para el ingreso de la contraseña:

1. Se ingresa la contraseña del usuario principal, en este caso “22334” que sirve de acceso y puede ser cambiada en la aplicación, además, se ingresa la captura de la huella digital. En la Figura 3.31, se puede observar el ingreso de la clave y la autenticación de la huella, mostrando del porcentaje de compatibilidad; en la Figura 3.32, se observa la notificación de acceso enviada a la aplicación desde el circuito de apertura.



Figura 3.31 Clave usuario principal y huella correcta



Figura 3.32 Notificación de apertura con clave principal

2. Se ingresa la contraseña de emergencia que se encuentra inicializada con “14725”, la cual antes de solicitar la huella digital envía un mensaje de alerta, se ingresa la huella y se realiza la apertura de la puerta principal. En la Figura 3.33, se observa el ingreso de la clave, cambiando a la clave de emergencia y la comparación de la huella digital; la diferencia se obtiene en el tipo de mensaje que se envía a la aplicación como se observa en la Figura 3.34, la cual muestra la notificación de emergencia o alerta seguido de la notificación de apertura.



Figura 3.33 Clave de emergencia y huella digital

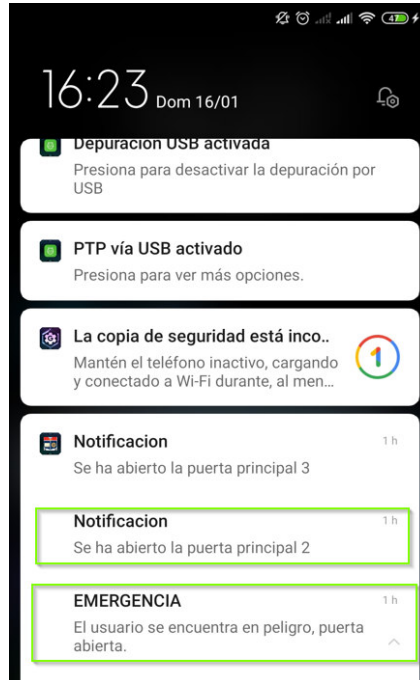


Figura 3.34 Notificación de emergencia y apertura

3. Contraseña o huella errónea: para este caso se tienen dos opciones, en la primera que la contraseña sea incorrecta, por lo tanto, no permite acceder a la verificación de la huella digital y reinicia con el mensaje principal. En la Figura 3.35 se muestra el mensaje de error que se proporciona al ingresar mal la clave desde un principio. La segunda opción, es que la huella no se encuentre registrada o se ingrese de manera incorrecta; en ambas alternativas, se muestra el mismo mensaje de error que se presenta en la Figura 3.36.



Figura 3.35 Error en la digitación de la contraseña



Figura 3.36 Error en la huella digital

3.2.2.2 Ingreso de Nuevos Registros

Adicionalmente, el ingreso de nuevas huellas digitales para el acceso se lo debe realizar de manera local en el sistema de control. Por medio de un botón en el lado izquierdo del módulo implementado, al presionarse envía a la opción de nuevo registro. Para validar que no sea realizado por cualquier usuario, se solicita la huella del usuario principal el cual se lo toma como el usuario "1". En la Figura 3.37 se muestra la solicitud de huella del usuario principal, así como el mensaje de autorización para el ingreso del nuevo usuario. Como se observa en la Figura 3.38 se solicita escoger en qué usuario de la base de datos local se quiere guardar la nueva huella digital.



Figura 3.37 Permiso de acceso y autorización de nuevo registro



Figura 3.38 Selección de usuario para la base de huellas digitales.

En la Figura 3.39 se muestra la solicitud de escaneo de la nueva huella digital para posteriormente solicitar un nuevo escaneo de la misma huella digital y capturar la imagen tal como se presenta en la Figura 3.40, tomando dos capturas de la misma huella para poder tener luego la comparación de éstas.



Figura 3.39 Captura de primera imagen de huella digital

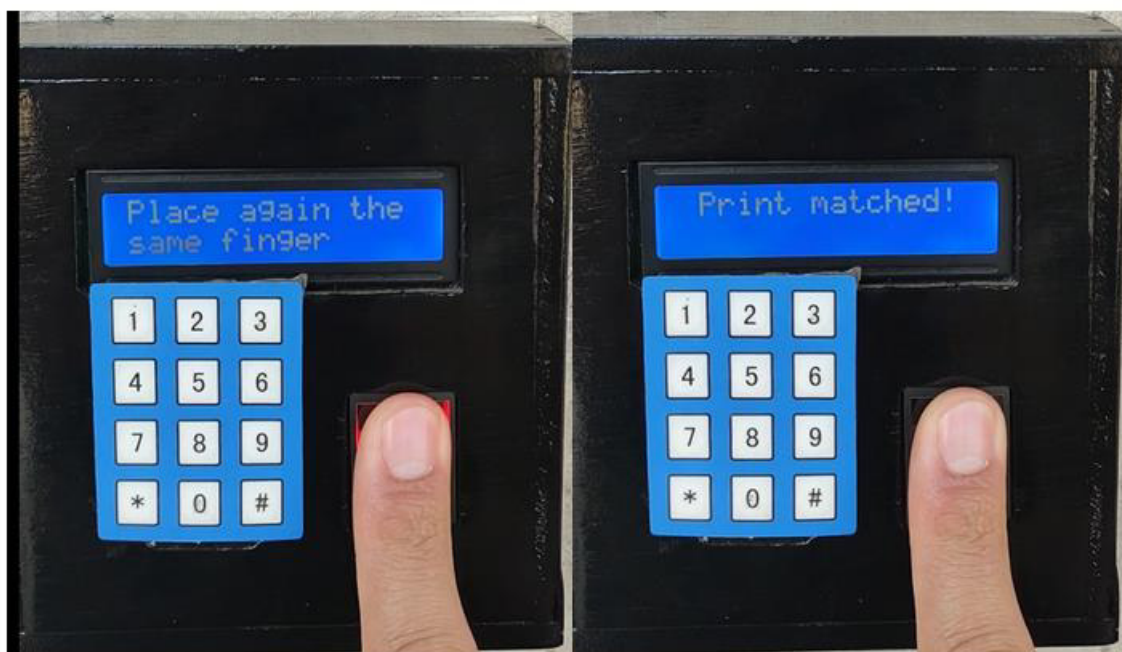


Figura 3.40 Captura de segunda imagen y validación.

En la Figura 3.41 se observa que la huella digital ha sido almacenada y se muestra un mensaje de confirmación indicando el número de usuario que logró el registro con éxito.



Figura 3.41 Almacenamiento y confirmación de nuevo registro

3.2.2.3 Errores en Registros

Se pueden tener errores al momento de ingresar un nuevo registro, tal como que la huella que se solicita del usuario principal no es la correcta, como se muestra en la Figura 3.42 por lo tanto, se regresa a la pantalla de inicio.



Figura 3.42 Huella digital incorrecta para nuevo registro

Por otra parte, cuando se está ingresando la huella digital y en la segunda captura se ingresa una huella digital distinta se imprime otro mensaje de error como se muestra en la Figura 3.43 y el registro del usuario no va a ser almacenado dentro del sistema.



Figura 3.43 Ingreso erróneo en huella digital

3.3 ANÁLISIS Y DISCUSIÓN DE RESULTADOS

En primer lugar, dentro del control de acceso se puede validar que el ingreso y comparación de las huellas digitales se realiza de manera exitosa, ya que se solicita que por cada huella ingresada se tomen dos capturas para poder compararlas y validar con la base de datos local. Por otra parte, el ingreso de nuevos usuarios solo se puede realizar siempre y cuando el usuario administrador del sistema ingresa su huella digital. Esto asegura que no se tendrá acceso desde otros usuarios y se protege el uso del sistema de control. Hay que considerar que, si un usuario tiene su huella digital en mal estado, puede ocasionar conflictos al momento de la comparación, ya que el sistema de control no va a reconocer las huellas, y por ende, no va a abrir la puerta.

Se realizaron intentos con usuarios que tenían las huellas digitales bastante gastadas por el tipo de trabajo que realizan y el detector de huellas tuvo inconvenientes al momento de leer sus resultados. Luego, se colocaron las huellas en el sensor biométrico de los mismos usuarios, pero tomando otro ángulo se pudo realizar la validación de las huellas de manera correcta.

También se aprecia que la toma de datos por parte del control de acceso desde la plataforma Firebase es bastante eficiente, pero la espera por las funciones de envío de notificaciones puede retardar el tiempo de respuesta de la apertura de la puerta.

En segundo lugar, la cámara de seguridad que se encuentra instalada permite observar el lugar a supervisar. La cámara es la encargada de generar notificaciones de alerta por movimiento, contraseña errada o clave de emergencia. Las notificaciones por contraseña incorrecta y clave de emergencia contienen una captura de imagen del momento en que se está realizando la acción. Para esto en las cámaras de seguridad se usaron dos ESP32-CAM, ya que la una funciona como emisor del video del ambiente y la otra realiza las capturas de las imágenes y envía las notificaciones de seguridad. Cuando se realizan las acciones en un solo módulo ESP32 se llega a tener conflictos con el envío y procesamiento del video.

En tercer lugar, la gestión del sistema de seguridad a través de la aplicación móvil es eficiente, ya que maneja las notificaciones, la administración de los dispositivos y la gestión del perfil de usuario para ingreso de usuarios invitados en fragmentos separados, haciendo que la aplicación sea mucho más fácil manejar. El sistema de autenticación de la huella digital funciona solamente con la huella del usuario del teléfono, ya que es una autenticación propia del sistema Android y no permite que se pueda tener lectura de huellas fuera de las registradas en su configuración. Por lo tanto, si se desea acceder al sistema

de control de la puerta principal solo se lo realizará con la huella del usuario que maneja el dispositivo.

4 CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- Al finalizar el presente Trabajo de Titulación y una vez que se ha implementado y probado el funcionamiento del prototipo en el ambiente a supervisar, se puede afirmar que se cumple plenamente con los objetivos propuestos.
- El Sistema inalámbrico desarrollado es escalable, ya que se puede ingresar más cámaras de seguridad y controles de acceso en el domicilio. El usuario puede observar los dispositivos en la lista principal para su manejo.
- El sistema implementado se lo puede mejorar haciendo posible que se instalen dispositivos no solo de seguridad sino de automatización del domicilio o lugar a instalar; es decir, puede escalar a un sistema domótico, lo cual no se lo realiza en el presente trabajo por el alcance aprobado en el Proyecto de Titulación.
- Los dispositivos implementados se pueden instalar en cualquier ambiente inalámbrico sin tener que ingresar a su programación gracias a librerías implementadas, pudiendo cambiar su red de conexión.
- El tener un usuario administrador dentro del control de acceso y de la aplicación móvil permite tener una mejor administración, gestión y seguridad del sistema desarrollado.
- Enviar diferentes notificaciones en el control de acceso por los casos propuestos al inicio, permite a los usuarios tomar acciones en caso de ser necesario.
- Para el ingreso de nuevos usuarios en el sensor biométrico, tener un usuario administrador proporciona mayor seguridad y evita que personas no autorizadas ingresen sus credenciales.
- Los materiales que se utilizaron para ensamblar los dispositivos se encuentran en cualquier casa comercial y son de bajo costo, lo cual hace que el sistema sea económico para su elaboración.
- Se concluye que la seguridad del sistema es bastante alta, ya que, al tener los dos tipos de autenticación de forma secuencial hace que no se pueda vulnerar el ingreso de manera fácil, es decir, aunque se logre adivinar la contraseña, la huella digital es única.

- El uso del sensor biométrico en la aplicación tiene limitantes ya que utiliza las huellas digitales registradas en la configuración del teléfono Android.

4.2 RECOMENDACIONES

- Mejorar la seguridad del ingreso a la puerta principal, cambiando el sistema del teclado por tarjetas RFID, lo que mejorará las características de seguridad.
- Mostrar el nombre del usuario que accede a la puerta principal para tener una bitácora más precisa de los usuarios que ingresan en el domicilio o lugar de instalación.
- No exceder la distancia permitida por el *router* para la conexión de los dispositivos, ya que se pueden generar retardos de transmisión y recepción de datos por falta de señal.
- Tener una fuente de energía de respaldo, lo cual se logra a través de un UPS. Si es necesario se puede añadir celdas de carga para aumentar el tiempo de respaldo de energía.
- Se recomienda notificar al usuario en caso de fallos de energía externa.
- Se debe considerar a los sistemas secundarios en caso de fallas eléctricas, de tal manera que no afecte en la comodidad de ingreso hacia el domicilio o la privacidad de los datos, imágenes y demás del usuario que ha creado la cuenta.
- Implementar luces de funcionamiento en cada dispositivo, para conocer si están funcionando de manera correcta o si es necesario realizar algún mantenimiento.
- Implementar leds indicadores de conexión exitosa hacia el *router*, para informar si se están transmitiendo datos hacia la plataforma.
- Considerar la velocidad contratada al Proveedor de Internet para evitar “cuellos de botella” en la salida de datos o información, ya que los dispositivos mantienen una conexión permanente.
- Realizar todos los cambios necesarios a la aplicación mientras aún no ha sido lanzada a su producción.
- Se recomienda el uso del sistema de seguridad ya que en el domicilio del autor donde se encuentra instalado funciona de manera correcta.

5. REFERENCIAS BIBLIOGRÁFICAS

- [1] Electronilab, «Sensor de movimiento PIR,» 20 Enero 2017. [En línea]. Available: <https://electronilab.co/tienda/sensor-de-movimiento-pir-hc-sr501/>. [Último acceso: 15 Octubre 2019].
- [2] X. P. D. V. Stefan Junestrand, Domotica y Hogar digital, Madrid: Paraninfo, 2005.
- [3] Android, «Arquitectura de la plataforma,» [En línea]. Available: <https://developer.android.com/guide/platform/index.html>. [Último acceso: 10 2019].
- [4] Android, «Android Developers- Activitys,» [En línea]. Available: <https://developer.android.com/guide/components/activities/intro-activities>. [Último acceso: 05 10 2019].
- [5] Android, «Ciclo de una Actividad,» [En línea]. Available: https://developer.android.com/guide/components/activities/activity-lifecycle?hl=es_419. [Último acceso: 1 10 2019].
- [6] Android, «Aspectos fundamentales de las App,» [En línea]. Available: https://developer.android.com/guide/components/fundamentals?hl=es_419. [Último acceso: 10 2019].
- [7] Android, «Descripcion Archivo de Manifiesto,» [En línea]. Available: https://developer.android.com/guide/topics/manifest/manifest-intro?hl=es_419. [Último acceso: 30 09 2019].
- [8] Y. Muradas, «OpenWebinars.net,» [En línea]. Available: <https://openwebinars.net/blog/que-es-firebase-de-google/>. [Último acceso: 4 10 2021].
- [9] G. Cloud, «Firebase,» 03 12 2019. [En línea]. Available: <https://firebase.google.com/docs/android/setup?hl=es-419>. [Último acceso: 25 01 2021].
- [10] Firebase, «Instalacion y Configuracion Base de Datos,» [En línea]. Available: <https://firebase.google.com/docs/database/android/start?hl=es-419>. [Último acceso: 12 5 2021].
- [11] Firebase, «Primeros pasos con Autenticacion,» [En línea]. Available: <https://firebase.google.com/docs/auth/android/start?hl=es-419>. [Último acceso: 16 4 221].
- [12] Firebase, «Llama funciones desde tu App,» [En línea]. Available: https://firebase.google.com/docs/functions/callable?hl=es-419#call_the_function. [Último acceso: 19 4 2021].
- [13] Arduino, «Software Arduino IDE,» Arduino, 2019. [En línea]. Available: <https://www.arduino.cc/en/guide/environment>. [Último acceso: 30 3 2021].

- [14] Arduino, «Arduino,» [En línea]. Available: <https://www.arduino.cc/en/Tutorial/LibraryExamples>. [Último acceso: 15 10 2021].
- [15] E. SYSTEMS, «espressif,» 2008. [En línea]. Available: <https://www.espressif.com/en/products/socs/esp32>. [Último acceso: 06 06 2021].
- [16] Naylamp mechatronics, «Naylamp mechatronics,» 20 Julio 2018. [En línea]. Available: <https://naylampmechatronics.com/espressif-esp/382-modulo-esp32-esp-wroom-32.html>. [Último acceso: 15 Octubre 2019].
- [17] Microdigisoft, «microdigisoft.com,» [En línea]. Available: <https://microdigisoft.com/getting-started-with-the-esp32-development-board/>. [Último acceso: 15 06 2021].
- [18] Randomnerdtutorials, «randomnerdtutorials,» [En línea]. Available: <https://randomnerdtutorials.com/esp32-cam-ai-thinker-pinout/>.
- [19] L. Llamas, «luisllamas.es,» [En línea]. Available: <https://www.luisllamas.es/detector-de-movimiento-con-arduino-y-sensor-pir/>.
- [20] Prometec, «PROMETEC,» [En línea]. Available: <https://www.prometec.net/lector-de-huellas/>. [Último acceso: 09 09 2021].
- [21] D. RAGIDAN, «¿Qué es Kanban?,» ¿Qué es Kanban?, 2021. [En línea]. Available: <https://www.atlassian.com/agile/kanban>. [Último acceso: 2 11 2021].
- [22] Android, «Android Developer,» [En línea]. Available: <https://developer.android.com/guide/platform/index.html>. [Último acceso: 08 10 2019].
- [23] naylampmechatronics, «naylampmechatronics,» [En línea]. Available: <https://naylampmechatronics.com/espressif-esp/384-nodemcu-32-30-pin-esp32-WiFi.html>. [Último acceso: 5 06 2021].

ANEXOS

ANEXO A: ENTREVISTA

ANEXO B: HISTORIA DE USUARIO

ANEXO C: CASOS DE USO

ANEXO D: MANUAL DE USUARIO

ANEXO E: CODIFICACIÓN DEL SISTEMA DE SEGURIDAD

ANEXO A

Entrevista

Entrevista realizada para determinar los parámetros iniciales del Sistema de seguridad.

Entrevista Sistema de Seguridad

Con el objetivo de brindar una mejor experiencia con el servicio de seguridad, solicitamos que conteste las siguientes preguntas

Cree que los sistemas de seguridad son indispensables? *

- ☐ No
- ☐ Sí
- ☐ Tal vez

Considera que es necesario el uso de una aplicación para el control y manejo de un sistema de seguridad *

- ☐ Sí
- ☐ No

En las aplicaciones móviles se hace uso de usuario y contraseña. De las siguientes opciones
¿Cuál preferiría como acceso? *

- ☐ Correo y contraseña
- ☐ Cedula y contraseña
- ☐ Telefono y contraseña
- ☐ Otra...

¿ Qué información del usuario considera que se debe guardar en la aplicación? *

- ☐ Nombre
- ☐ Apellido
- ☐ Correo
- ☐ Telefono
- ☐ Imagen

¿Qué acciones considera que deben ser manejados en la aplicación? *

- ☐ Gestionar dispositivos (agregar, modificar, eliminar, actualizar)
- ☐ Gestionar usuarios (agregar, modificar, eliminar, actualizar)
- ☐ Visualizar estado actual de los dispositivos
- ☐ Notificaciones

Dentro de la aplicación. ¿Qué información considera que se debe mostrar de los dispositivos? *

- ☐ Nombre
- ☐ Ip asignada
- ☐ Descripción
- ☐ Estado actual(on/off)

¿Es importante la marca de los dispositivos a usar? *

☐ Sí

☐ No

Considera que los dispositivos deben ser: *

☐ Inalámbricos

☐ Cableados

En el caso de un control de puerta. ¿Cuál de las siguientes opciones considera mas seguro? *

☐ Clave de Seguridad

☐ Sensor Biométrico(huella digital)

☐ Apertura por SMS

☐ Comandos de voz

En el caso de uso de sensor biométrico. Considera que todas las personas dentro del hogar deben poder registrar su huella? *

☐ Sí

☐ No

El en caso de uso de clave. Considera que debe ser una clave por cada miembro de la familia? *

☐ Sí

☐ No

☐ Tal vez

¿Considera que las cámaras deben grabar solo los eventos que presenten alertas? *

☐ Sí

☐ No

☐ Tal vez

ANEXO B

HISTORIA DE USUARIO

En este anexo se adjuntan las historias de usuarios.

HISTORIA DE USUARIO		
ID:HU 001	Usuario: administrador	
Programador Responsable: Juan Tapia		
Nombre Historia: Iniciar Sesión	Tipo: Requerimiento Funcional	
Descripción: autenticación del usuario en la aplicación		
Comentario: El usuario podrá autenticarse en la aplicación a través del sistema Firebase		

HISTORIA DE USUARIO		
ID:HU 002	Usuario: administrador	
Programador Responsable: Juan Tapia		
Nombre Historia: Cerrar Sesión	Tipo: Requerimiento no Funcional	
Descripción: El usuario podrá cerrar sesión en la aplicación del sistema integrado		
Comentario: El usuario podrá salir del sistema		

HISTORIA DE USUARIO		
ID:HU 003	Usuario: administrador	
Programador Responsable: Juan Tapia		
Nombre Historia: Administrar Usuario	Tipo: Requerimiento Funcional	
Descripción: El usuario administrador podrá agregar usuarios		
Comentario: El usuario administrador podrá añadir usuarios tanto administradores como invitados		

HISTORIA DE USUARIO		
ID:HU 004	Usuario: administrador	
Programador Responsable: Juan Tapia		
Nombre Historia: Actualizar Usuario	Tipo: Requerimiento Funcional	
Descripción: El usuario administrador podrá actualizar la información de un usuario		
Comentario: El usuario administrador actualizar la información de un usuario, así como un usuario invitado actualizar su perfil		

HISTORIA DE USUARIO		
ID:HU015	Usuario: usuario invitado	
Programador Responsable: Juan Tapia		
Nombre Historia: Validación de credenciales	Tipo: Requerimiento no funcional	
Descripción: validar que los usuarios ingresen con correo y contraseña.		
Comentario: La autenticación se la realiza en el sistema Firebase, una vez registrados, el sistema valida las credenciales para ingreso.		

HISTORIA DE USUARIO		
ID:HU 006		Usuario: administrador
Programador Juan Tapia	Responsable:	
Nombre Historia: Buscar Usuario		Tipo: Requerimiento Funcional
Descripción: El usuario administrador podrá buscar un usuario		
Comentario: El usuario administrador podrá buscar dentro de la lista a un usuario en particular		

HISTORIA DE USUARIO		
ID:HU 007	Usuario: administrador	
Programador Responsable: Juan Tapia		
Nombre Historia: Administrar dispositivos	Tipo: Requerimiento Funcional	
Descripción: El usuario administrador podrá añadir un nuevo dispositivo		
Comentario: El usuario administrador podrá añadir un nuevo dispositivo que se haya instalado y aparezca en la lista de dispositivos a agregar		

HISTORIA DE USUARIO		
ID:HU 008	Usuario: administrador/invitado	
Programador Responsable: Juan Tapia		
Actualizar información de dispositivos	Tipo: Requerimiento Funcional	
Descripción: El usuario podrá actualizar la información del dispositivo.		
Comentario: El usuario actualizar la información del dispositivo, tanto su estado actual como la información de horarios automáticos si estos se encuentran disponibles		

HISTORIA DE USUARIO		
ID:HU 009	Usuario: administrador	
Programador Responsable: Juan Tapia		
Nombre Historia: Eliminar dispositivos	Tipo: Requerimiento Funcional	
Descripción: El usuario administrador podrá eliminar un dispositivo del sistema.		
Comentario: El usuario administrador podrá eliminar dispositivos de sistema. Esto quiere decir que no podrá configurar los mismos.		

HISTORIA DE USUARIO		
ID:HU 010	Usuario: administrador	
Programador Responsable: Juan Tapia		
Nombre Historia: Buscar dispositivos	Tipo: Requerimiento Funcional	
Descripción: El usuario administrador podrá buscar un dispositivo		
Comentario: El usuario administrador podrá buscar un dispositivo		

HISTORIA DE USUARIO		
ID:HU 011	Usuario: administrador	
Programador Responsable: Juan Tapia		
Nombre Historia: Visualizar cámara	Tipo: Requerimiento Funcional	
Descripción: El usuario podrá visualizar la cámara		
Comentario: El usuario podrá visualizar la cámara a través de la aplicación móvil.		

HISTORIA DE USUARIO		
ID:HU 012	Usuario: administrador	
Programador Responsable: Juan Tapia		
Nombre Historia:	Tipo: Requerimiento Funcional	
Ingresar clave y huella digital para ingreso a puerta principal		
Descripción: El usuario podrá ingresar la clave de acceso.		
Comentario: El usuario podrá ingresar las claves de acceso a través de la aplicación o de manera manual.		

HISTORIA DE USUARIO		
ID:HU 013	Usuario: administrador	
Programador Responsable: Juan Tapia		
Nombre Historia: Recibir notificaciones de ingreso	Tipo: Requerimiento Funcional	
Descripción: El usuario recibirá en la aplicación notificaciones con pin de ingreso		
Comentario: El usuario recibirá en la aplicación notificaciones con pin de ingreso a la aplicación una vez validada la primera clave.		

HISTORIA DE USUARIO		
ID:HU 014	Usuario: administrador	
Programador Responsable: Juan Tapia		
Nombre Historia:	Tipo: Requerimiento Funcional	
Recibir notificación de presencia		
Descripción: El usuario recibirá notificación de movimiento		
Comentario: El usuario recibirá notificación de movimiento dentro de la asociación de estudiantes fuera de horarios.		

HISTORIA DE USUARIO		
ID:HU 016	Usuario: administrador	
Programador Responsable: Juan Tapia		
Nombre Historia: Tiempo de respuesta del sistema	Tipo: Requerimiento No Funcional	
Descripción: El sistema depende de la calidad del internet para tiempo de respuesta		
Comentario: El sistema depende de la calidad del internet para tiempo de respuesta, sin embargo, debe ser casi inmediato.		

HISTORIA DE USUARIO		
ID:HU 017	Usuario: administrador	
Programador Responsable: Juan Tapia		
Nombre Historia:	Tipo: Requerimiento Funcional	
Validación de claves de ingreso puerta		
Descripción: El sistema validara las contraseñas		
Comentario: El sistema validara las contraseñas enviadas a través de peticiones por parte del usuario		

ANEXO C

CASOS DE USO

Id: CU1	Nombre: Acceder a Puerta Principal
Actor: Usuario	Prioridad: Media
Descripción: El usuario accede a la asociación por la puerta principal con parámetro de doble seguridad.	
Flujo Normal: <ol style="list-style-type: none"> 1.- El usuario ingresa contraseña en aplicación o teclado virtual. 2.- El usuario coloca su huella biométrica. 3.- Las cerraduras son abiertas. 4.- El sistema es notificado del nuevo ingreso. 	
Flujo Alternativo: <ol style="list-style-type: none"> 1. La contraseña o pin proporcionada es incorrecta y se notifica al usuario. 	

Id: CU2	Nombre: Actualizar información del usuario al sistema
Actor: Usuario	Prioridad: Alta
Descripción: El usuario actualiza la información que se encuentra dentro del sistema.	
Flujo Normal: <ol style="list-style-type: none"> 1.- El usuario se autentica y accede al sistema. 2.- El usuario cambia su información dentro de la aplicación. 3.- El sistema actualiza la información en Firebase. 4.- El usuario finaliza sesión. 	
Flujo Alternativo: <ol style="list-style-type: none"> 1. El proceso de autenticación falla y el usuario no puede ingresar al sistema. Se muestra mensaje de error. 2. La actualización de los datos del usuario no es posible por falta de conectividad. 	

Id: CU3	Nombre: Monitorear Cámara
Actor: Usuario	Prioridad: Media
Descripción: El usuario accede a la cámara de seguridad para verificar actividades en cualquier horario.	
Flujo Normal: <ol style="list-style-type: none"> 1.- El usuario accede al sistema. 2.- El usuario ingresa a la cámara. 	
Flujo Alternativo: <ol style="list-style-type: none"> 1. El acceso a la cámara de seguridad falla por falta de conexión en la aplicación. 	

Id: CU4	Nombre: Alertar Presencia
Actor: Usuario	Prioridad: Media
Descripción: El sistema notifica al usuario de movimiento dentro del sistema.	
Flujo Normal: 1.- El sistema notifica al usuario de movimiento en los casilleros y activa luminaria.	
Flujo Alternativo: 1. El sistema no notifica del movimiento.	

ANEXO D

MANUAL DE USUARIO

En esta sección se realiza el manual de usuario para la instalación de los dispositivos hacia el sistema de seguridad.

MANUAL DE USUARIO

SISTEMA DE SEGURIDAD

Introducción

1.1 Objetivo 2

1.2 Requerimientos..... 2

Opciones del Sistema

2.1 Ingreso/ Registro en el Sistema..... 2

2.2 Configuración de Perfil..... 3

2.3 Agregación de usuarios..... 4

2.4 Instalación física de los dispositivos..... 5

2.5 Anclaje a la red de los dispositivos 5

2.6 Integración y eliminación de los dispositivos en la aplicación..... 7

2.7 Manejo de las configuraciones de los dispositivos..... 9

1 INTRODUCCIÓN

1.1 Objetivo

Otorgar soporte a los usuarios para el registro y autenticación en la aplicación móvil, así como para la instalación de los dispositivos de seguridad a la red y su agregación a la plataforma de control.

1.2 Requerimientos

- Teléfono con sistema Android 7 o superior
- Conexión estable a Internet
- Conexión de tomacorrientes para los dispositivos.

2 OPCIONES DEL SISTEMA

El presente manual se encuentra organizado conforme la secuencia de conexión, anclaje y administración de los dispositivos, así como también a la gestión de usuarios.

- Ingreso/ Registro en el sistema
- Configuración de Perfil
- Agregación de usuarios invitados
- Instalación física de los dispositivos
- Anclaje a la red de los dispositivos físicos
- Integración y eliminación de los dispositivos en la aplicación
- Manejo de las configuraciones de los dispositivos

2.1 Ingreso/ Registro en el Sistema

En la pantalla inicial el usuario debe presionar el botón Crear Cuenta que le llevara a la pantalla donde se va a solicitar su información para el ingreso al sistema, con el fin de acceder a la plataforma y poder recibir las notificaciones. Todos los campos solicitados deben se completados.

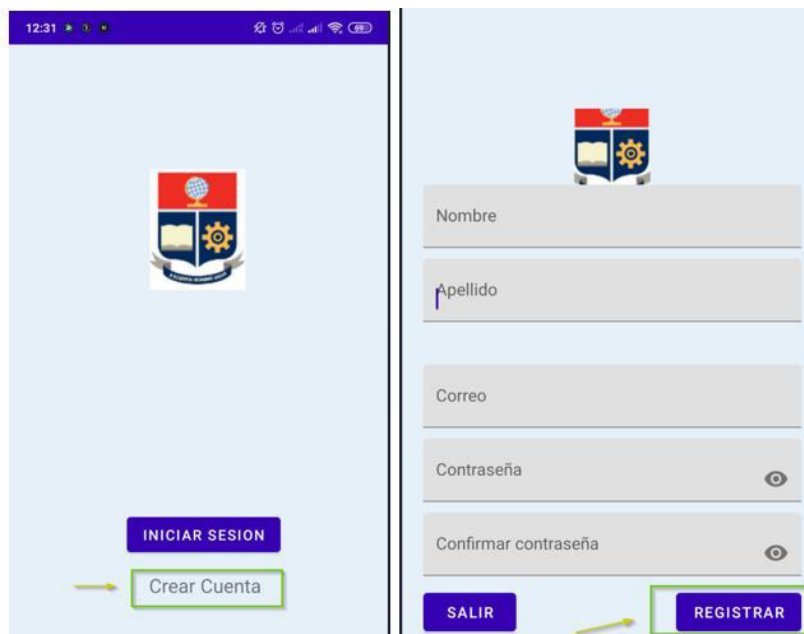


Figura D.1 Creación y registro de Usuario

2.2 Configuración de Perfil

En la configuración del Perfil se puede editar los campos del usuario a excepción del correo electrónico, ya que este campo se encuentra seteado para la recepción de notificaciones, así como para la autenticación en la plataforma. En esta pantalla se tiene la opción de cierre de sesión.

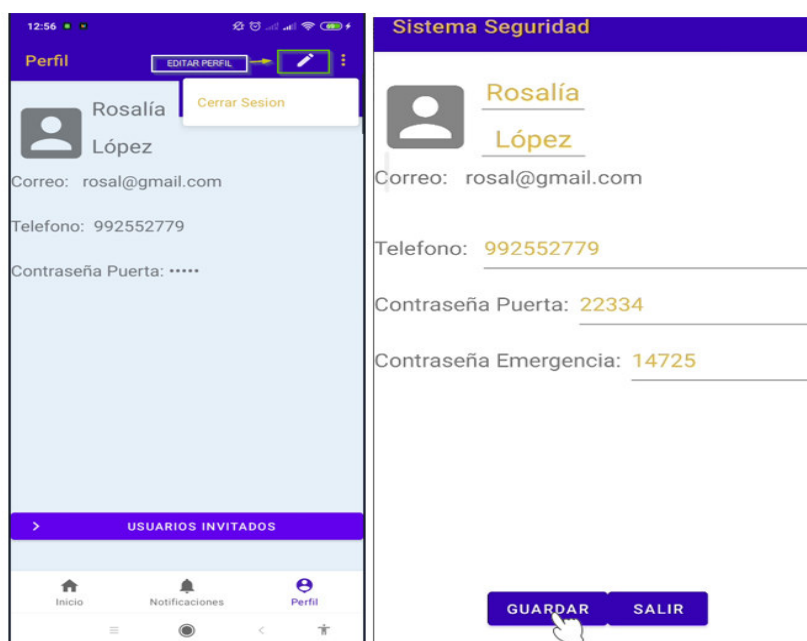


Figura D.2 Perfil y Edición de Usuario

2.3 Ingreso de usuarios invitados

Dentro de la configuración del perfil de usuario presionamos en botón Usuarios invitados, que redirige a la pantalla de lista de usuarios invitados creados, en esta pantalla seleccionamos el icono + para poder ir a la creación de un nuevo usuario.

En la pantalla de creación de usuario invitado, se solicita el nombre y apellido. Además, se observa la creación de la clave de acceso para la puerta principal.

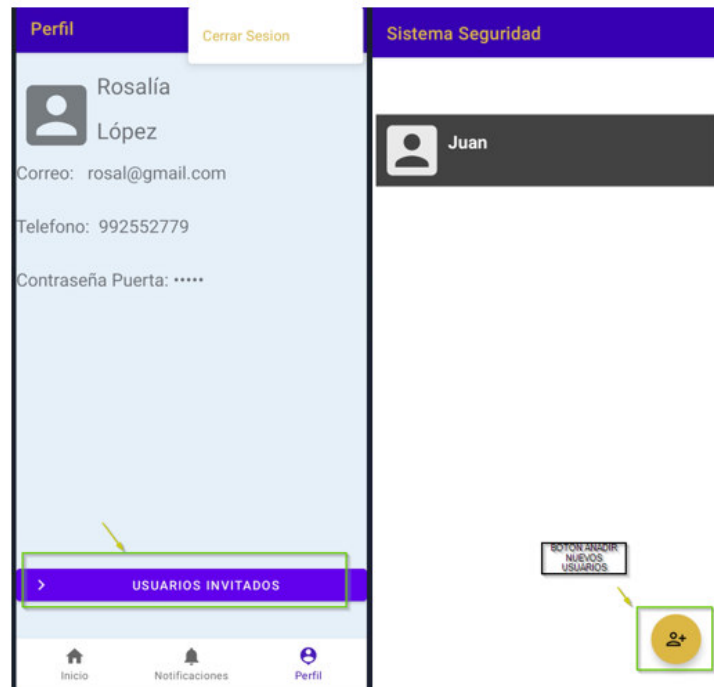


Figura D.3 Selección de botón de usuarios y botón añadir usuario

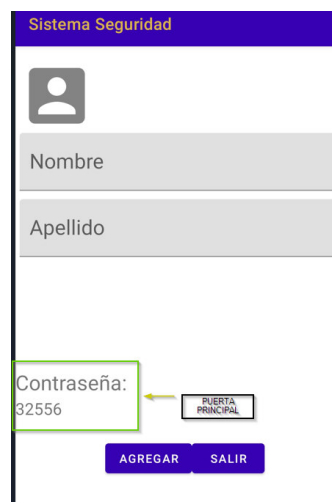


Figura D.4 Creación de nuevo usuario invitado

2.4 Instalación física de los dispositivos

Para la instalación física de los dispositivos se tiene una conexión directa a los tomacorrientes, ya que cuentan con su propio cable USB y su cargador. No tienen conexión de respaldo interno de energía eléctrica.



Figura D.5 Dispositivo para conexión.



Figura D.6 Montaje del dispositivo

2.5 Anclaje a la red de los dispositivos

Una vez que los dispositivos se encuentran conectados a la energía eléctrica vamos a ir a las redes de conexión WiFi, donde podemos observar que se encuentra una nueva red temporal con el nombre del dispositivo que acaba de ser conectado; seleccionamos la red con el nombre del dispositivo y redirige a una página web, donde muestra las redes WiFi-

disponibles. Por lo tanto, se selecciona la red domiciliaria a la cual van a ser conectados los dispositivos y se ingresa la contraseña de seguridad.

Una vez que se ha procedido de esta manera, los dispositivos se van a poder visualizar en la aplicación.

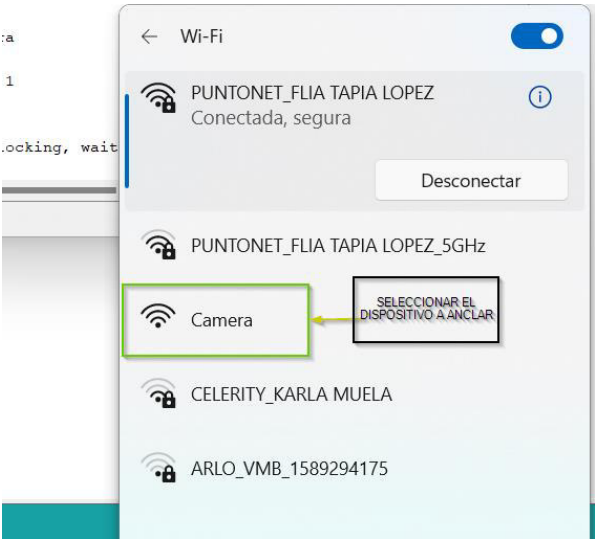


Figura D.7 Dispositivo conectado y solicitando acceso a la red

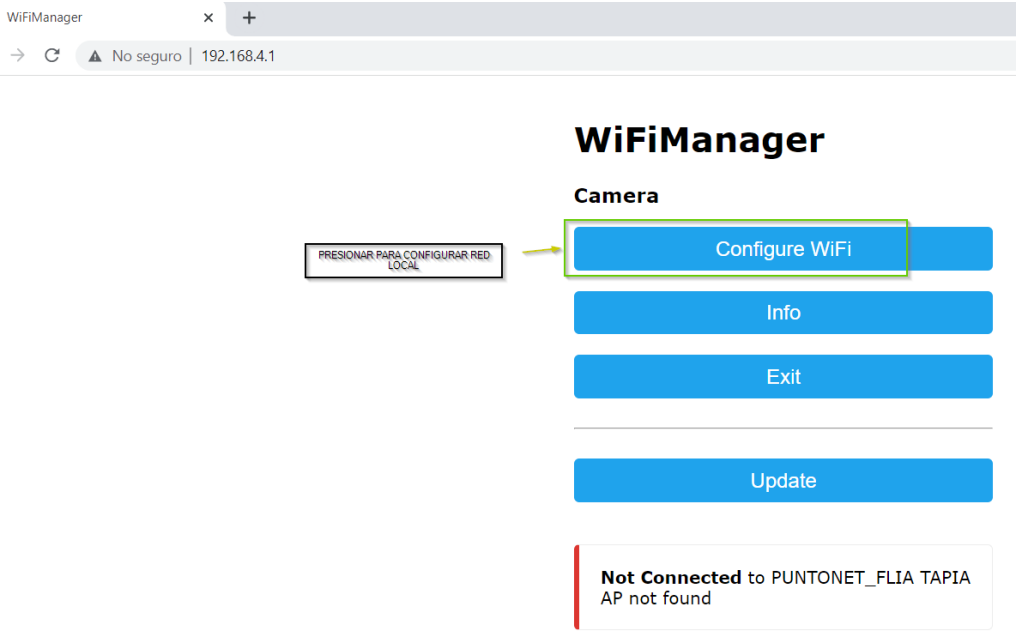


Figura D.8 Solicitud de acceso para redes disponibles

Config ESP

No seguro | 192.168.4.1/wifi?#p

PUNTONET_FLIA TAPIA LOPEZ

CELERITY_KARLA MUELA

SSID

PUNTONET_FLIA TAPIA LOPEZ

Password

Save

Refresh

Figura D. 9 Selección de Red Local y credenciales de autenticación

2.6 Integración y eliminación de los dispositivos en la aplicación

En la aplicación móvil, una vez que los dispositivos han sido anclados a la red domiciliaria hay que agregarlos. Para esto, se ubica en la opción “Inicio”, en la parte superior derecha se encuentra el botón de agregar con el símbolo “+”. Una vez que se presiona la opción, aparece una nueva pantalla, donde están los dispositivos a agregarse. Al seleccionar el dispositivo solicita confirmación de aceptación, se procede con la confirmación y el dispositivo ha sido movido a nuestra lista de control de manera satisfactoria.



Figura D. 10 Botón para agregar nuevos dispositivos



Figura D. 11 Escoger el dispositivo anclado

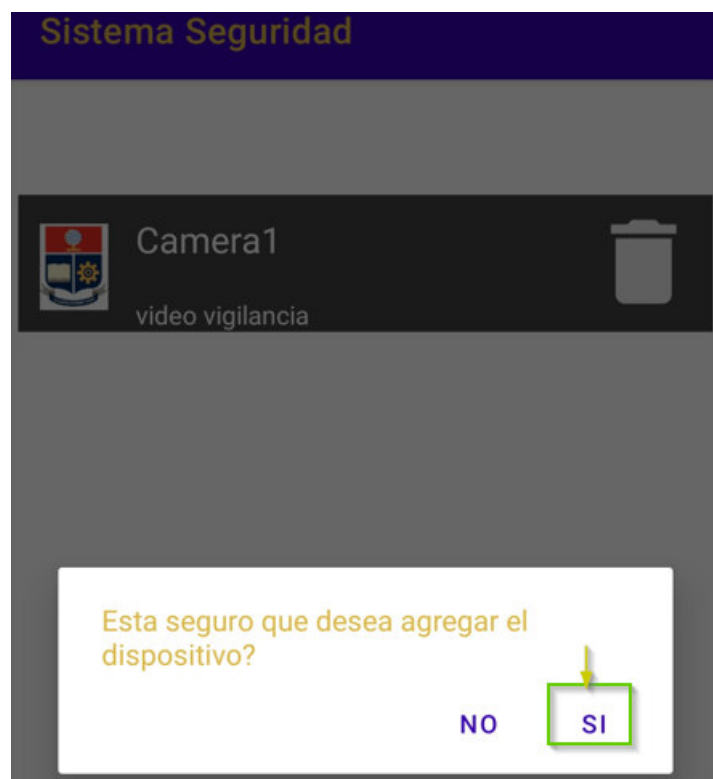


Figura D. 12 Confirmación de agregación del dispositivo

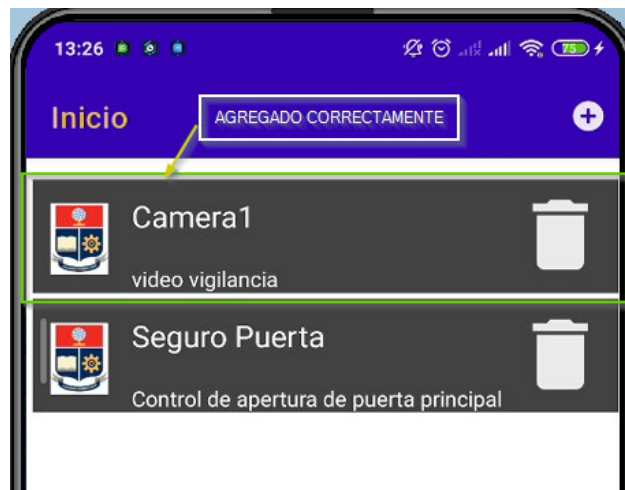


Figura D. 13 Dispositivo agregado correctamente

2.7 Manejo de las configuraciones de los dispositivos

Cuando ya se encuentran agregados los dispositivos y se desea acceder a los diferentes dispositivos se usa la opción de “Inicio” y se selecciona el dispositivo a mostrar, esta acción lleva a la pantalla de configuración que, dependiendo del dispositivo va a ser diferente la interfaz.



Figura D. 14 Ingreso afirmativo a dispositivo cámara

ANEXO E

CODIFICACIÓN DEL SISTEMA DE SEGURIDAD

Debido a su extensión, la codificación del sistema de seguridad se encuentra en adjunto en *One Drive* para su verificación.