**MANUAL TÉCNICO DEL PROYECTO**

**PROGRAMACIÓN DE SISTEMAS TELEMÁTICOS**

**TEMA 8: SONIDO SECRETO PARA ABIR LA PUERTA DE UNA CASA, APLICACIÓN MOVIL**

A picture containing drawing, sign

Description automatically generated

**GRUPO 5**

**INTEGRANTES:**

* Sheyla Benavides Salmeron
* Alexis Cando Palma
* Alex Mora Hurel
* Jaime Moran Santos

**PROFESORA:** ADRIANA COLLAGUAZO JARAMILLO

PRIMER TÉRMINO 2020

1. **Resumen ejecutivo**

Con el pasar de los años los sistemas de seguridad se han ido acoplando a las nuevas tecnologías emergentes como biometría, a causa de que estas permiten incrementar su seguridad debido a su baja probabilidad de replicación, lo que significaría que un intento de alguien desconocido de acceder al sistema, el fraude sería evidenciado de inmediato, y en base al diseño del sistema, permitiría notificar al usuario de este intento malicioso. Usualmente esta clase de sistemas son utilizados en negocios, sin embargo, es esencial para la seguridad en los hogares, poder aplicar este tipo de sistemas en ellos, con el fin de asegurar la protección de los bienes de una forma cómoda y eficiente, mediante la instalación de un solenoide de cerradura programado y controlado por una aplicación, que puede ser usada mediante un dispositivo móvil Android. A partir de la aplicación, luego del registro del usuario administrador, este tiene la facultad de agregar más usuarios, además de administrar más de un dispositivo dentro de su propio hogar, y controlar las cerraduras mediante una clave ingresada por voz. Los usuarios agregados por el administrador podrán ejercer control sobre las puertas que el administrador registró, teniendo la opción de cerrarlas y abrirlas por voz mediante una clave fijada al momento de haber hecho su registro de cuenta. El usuario administrador recibirá notificaciones acerca de posibles intentos de ingreso no autorizado a la aplicación, a través de los correos ingresados en su registro, estos tipos de datos son almacenados en la base de datos de Firebase Google.

El costo de la implementación física de Smart Sound es de alrededor de $41, pudiéndose reducir gastos si se realiza a mayor escala, mientras que, para su control, se necesita un dispositivo Android que cuente con conexión a internet. Este producto es dirigido a hogares para afianzar la seguridad de sus pertenencias, pudiéndose extender el enfoque del negocio a la seguridad de empresas de pequeña y mediana escala. Debido a su fácil implementación y bajo costo de manufactura, este producto es accesible y práctico para hacer competencia a los diferentes sistemas de seguridad ofrecidos en el mercado actual.

1. **Descripción del problema**

Los equipos de seguridad convencionales de los hogares no brindan la seguridad necesaria para la protección de los bienes, debido a que pueden ser fácilmente vulnerados. Por ende, en la actualidad, se han implementado en los sistemas de seguridad tecnología biométrica, usando el escáner de huella dactilar, iris, activación de voz, entre otros. Esto se debe a que son sistemas seguros y que brindan retroalimentación al usuario por medio de notificaciones inmediatas a su dispositivo móvil, en el caso de un intento de acceso sospechoso al dispositivo, permitiéndole al usuario actuar ante la alarma y prevenir saqueo de sus bienes. Además, estos sistemas son interactivos, de fácil instalación y acceso ya que solo necesita un dispositivo móvil con conexión a internet para poder manejarlo.

1. **Objetivos específicos**

* Diseñar un sistema de apertura de cerrojos el cual pueda ser accedido por múltiples usuarios cuyas identidades y claves se encuentran almacenadas en la base de datos, siendo la clave una palabra predeterminada por cada usuario.
* Mejorar la seguridad del hogar mediante la gestión de los usuarios a través una interfaz creada en Android Studio que a su vez permite la recepción de alertas al usuario administrador directamente en el celular.
* Aplicar los conceptos aprendidos sobre bases de datos y aplicaciones usando Android Studio para hacerlas trabajar de forma conjunta con el objetivo de dar solución a un problema real.

1. **¿Cómo funciona la solución propuesta?**

Para la solución del problema se usará la base de datos de FireBase Google para poder guardar la información referente a los dispositivos usados, los usuarios administradores y los usuarios a los que el administrador desee dar acceso al dispositivo. Para el cierre y apertura de la puerta se utilizará un solenoide de 12V con un pestillo, que cierra o abre su seguro en base a que el usuario ingrese por voz, la clave del dispositivo, que determinó en su registro de cuenta. Previo al cierre del seguro, se valida mediante un led y fotorresistor que la puerta esta junta al marco de la puerta, el seguro no se cerrará al menos que esta se encuentre unida.

Los usuarios administradores tendrán la opción de abrir puertas, administrar sus dispositivos, usuarios y agregar más usuarios para que accedan a los dispositivos. Mientras que los usuarios registrados por el administrador únicamente tendrán la posibilidad de cerrar o abrir los dispositivos mediante voz registrados por el usuario administrador. El usuario administrador recibirá una notificación a su correo registrado en el caso de que alguien haya intentado ingresar 4 veces y sin éxito, esto se tomará como un posible ingreso desautorizado, por lo que se lo notificará al administrador para que tome las acciones pertinentes.

1. **¿Qué van a construir para resolver el problema?**

Como interfaz entre el sistema y el usuario se realizará una aplicación usando Android Studio, la cual puede ser utilizada en un dispositivo Android. Adicionalmente, para la simulación del sistema se construirá una maqueta de la puerta hecha de balsa, y se instalará en este el solenoide de 12 V con el pestillo, en una carcaza impresa en 3D con material PLA. Para el control del solenoide, conexión con la base de datos y la validación de que la puerta se encuentre junta para poder cerrarla, se realizará un circuito electrónico, programado con Arduino. Toda la información referente a los usuarios, dispositivos, y demás datos, serán almacenados en una base de datos creada en Firebase Google.

1. **Recursos de hardware y de software**

**Hardware**

* Módulo ESP32 Dev Kit
* Solenoide cerradura de 12V
* Relay 5V
* Fotorresistor
* Leds
* Jumpers

**Software**

* Android Studio
* Arduino IDE
* Firebase Google

1. **Explicación paso a paso de la implementación del proyecto**

Para la implementación del proyecto, tanto su parte física como la aplicación que la administra, se realizaron los siguientes pasos:

* Diseño del modelo entidad-relación.
* Desarrollo del diagrama de diseño del proyecto.
* Creación de la base de datos del proyecto en FireBase Google.
* Creación del repositorio del proyecto en GitHub.
* Programación y diseño de la aplicación y cada una de las ventanas utilizadas mediante Android Studio.
* Programación del código en Arduino IDE que controla el circuito electrónico del solenoide que actúa como cerradura.
* Cotización y compra del hardware a utilizar en el circuito electrónico el solenoide de cerradura de 12V, para el cierre y apertura de la puerta.
* Construcción de la puerta y su integración a la maqueta.
* Acoplamiento del solenoide de cerradura de 12 V con su carcaza a la puerta de la maqueta, junto con su circuito electrónico de control.
* Comprobación del funcionamiento exitoso de la solución propuesta.

1. **Diagrama de diseño**



Ilustración 1. Diagrama de diseño.

En el diagrama de diseño se detalla la conexión del módulo ESP32 Dev Kit con la base de datos de Firebase Google, que sirve de intermediario entre el solenoide de 12 V de cerradura y la base, para poder cerrar o abrir la cerradura dependiendo de los valores dados por el circuito de control del solenoide, validándose si la puerta se encuentra unida al marco o si el usuario decidió abrirla o cerrarla mediante su clave de dispositivos, luego de ingresar correctamente su usuario y clave de la aplicación.

1. **Diagrama de circuito**

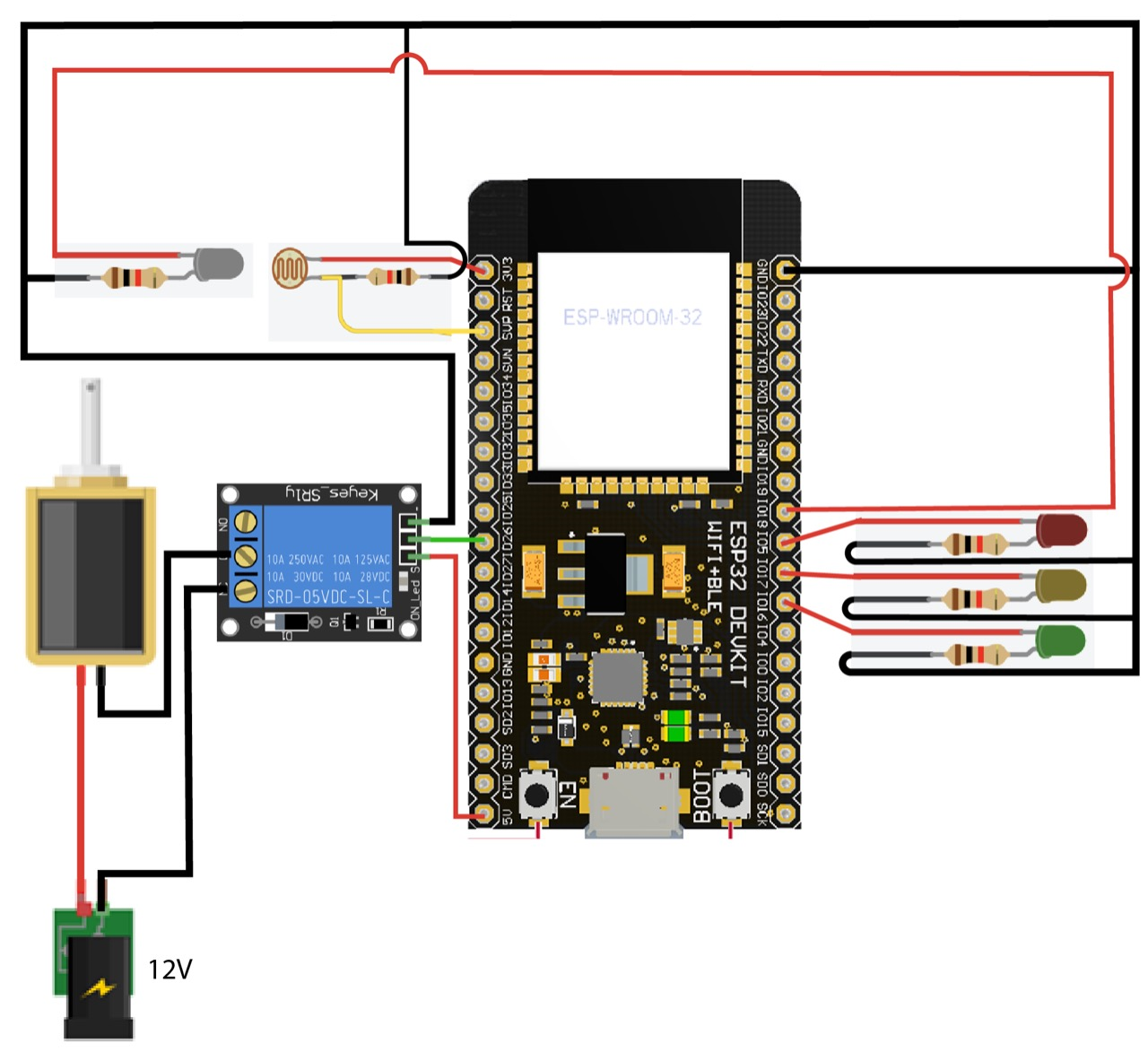


Ilustración 2. Diagrama del circuito.

1. **Diagrama del modelo entidad-relación**

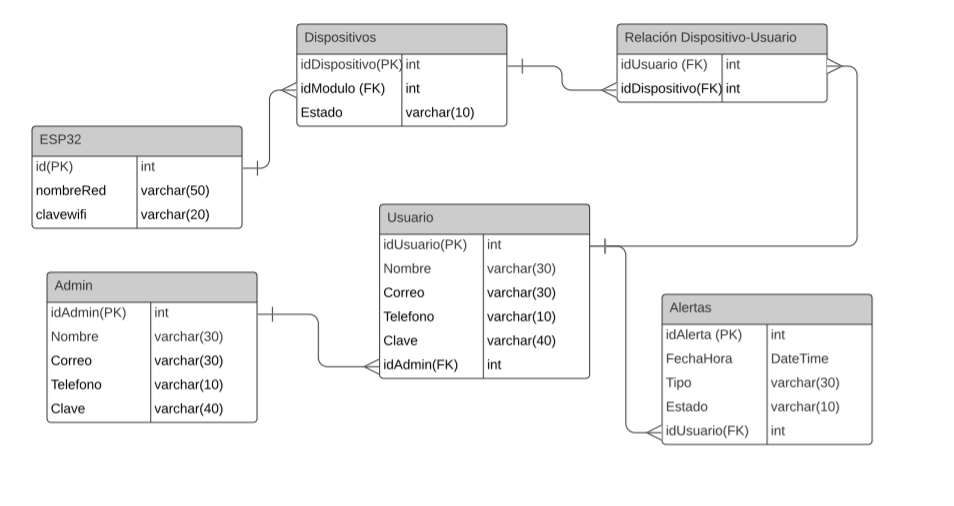


Ilustración 3. Modelo entidad-relación

1. **Diagrama de casos UML**

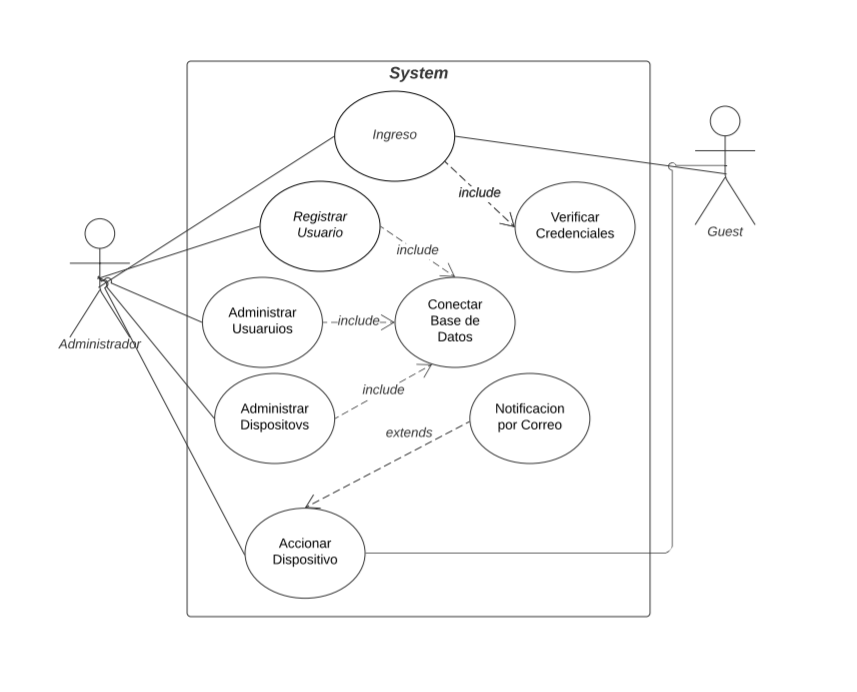


Ilustración 4. Diagrama de casos.

1. **Diagrama de despliegue**

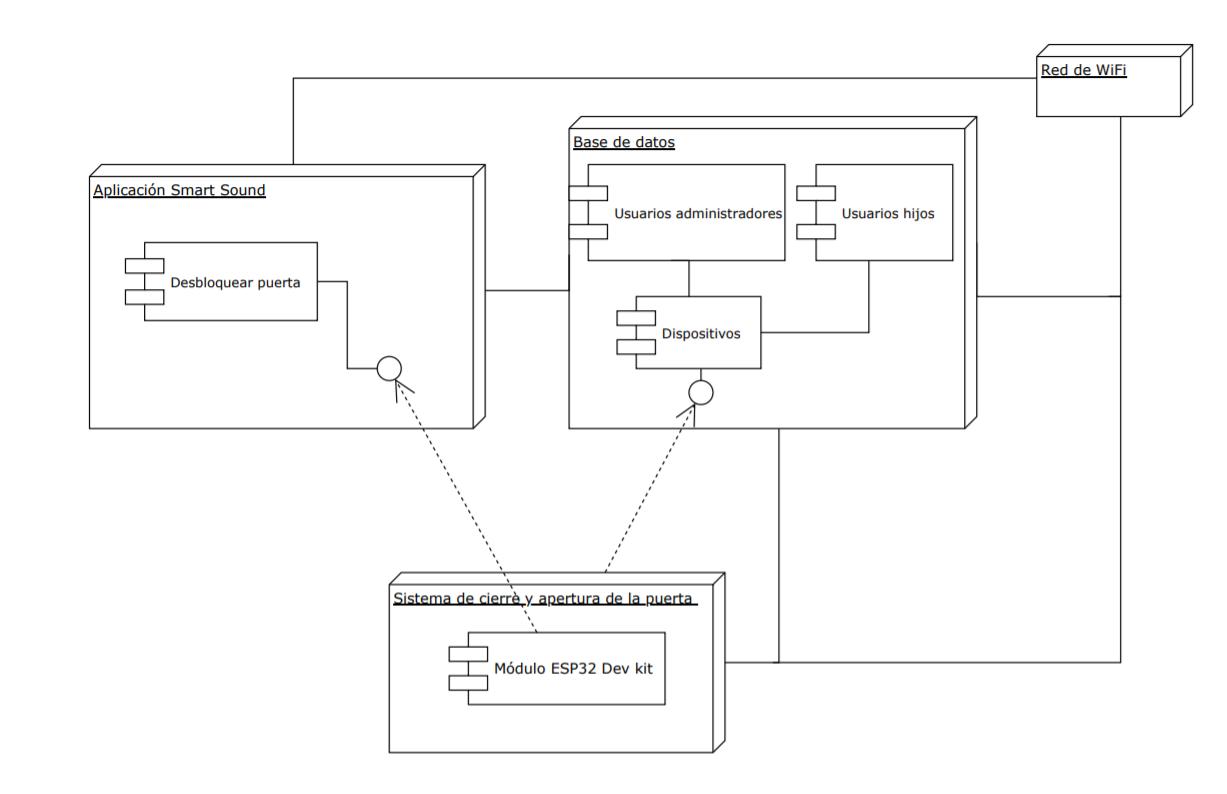


Ilustración 5. Diagrama de despliegue.

1. **Descripción de los campos y tipos de datos creados en base de datos**

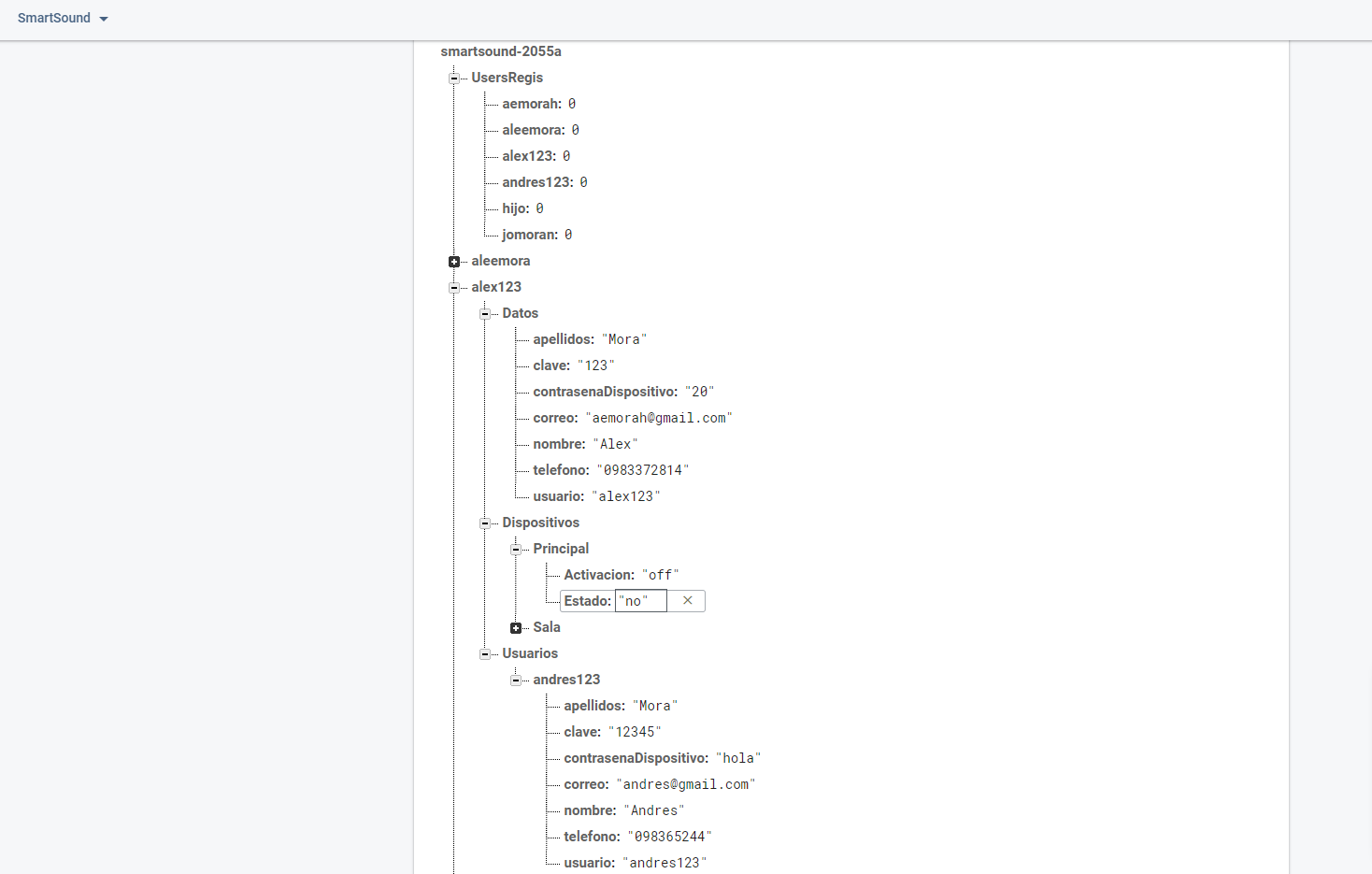


Ilustración 6. Base de datos en Firebase Google.

La base de datos creada en Firebase Google para el desarrollo del proyecto (<https://smartsound-2055a.firebaseio.com/> ) tiene como primer dato UsersRegis, dentro de la cual se tienen los usuarios registrados contabilizados y almacenados para que no se repita un registro bajo el mismo usuario. Luego se almacenan todos los usuarios administradores registrados en la base de datos, y dentro de estos, se encuentran almacenados sus datos referentes a apellidos, clave, contrasenaDispositivo, correo, nombre, telefono y usuario, siendo todos datos de tipo String. Adicionalmente, dentro de los usuarios administradores, se encuentra sus dispositivos registrados, teniendo en cada uno de estos una variable activación y otra llamada estado, ambas de tipo String. La variable activación indica el cierre de la cerradura (“off”) y su apertura mediante el string (“on”), mientras que la variable estado, indica que la puerta esta unida al marco usando el String “si” o no se encuentra unida mediante el String “no”. Por último, también dentro de cada usuario administrador, se tiene usuarios, en el cual se almacena los usuarios hijos que son registrados por el usuario administrador, dentro de este se almacena los mismos datos de tipo String solicitados al usuario administrador en su registro.

1. **Explicación del código fuente completo desarrollado con los comentarios correspondientes**

AdministrarDisp: Activity que se encarga de cargar los datos en un ListView para poder mostrar todos los dispositivos que tiene registrado el administrador respectivo, adicionalmente presenta las opciones de eliminar y agregar nuevos dispositivos, siempre mostrando un Dialog para confirmar la opción solicitada.

AdministrarUsuarios: Activity que también carga los datos dentro de un ListView para mostrar los usarios (hijos) que tiene registrado el administrador, de igual forma al querer eliminar uno de los usuarios se presenta un Dialog de confirmación.

Bienvenida: Activity donde se muestra el logo representativo de la aplicación y mediante un Timer se mantiene para luego cambiar a la ventana de Login.

DesbloquearPuerta: Ventana donde se muestra los dispositivos que se pueden abrir mediante un ListView y adicionalmente en la parte inferior dos botones, uno para iniciar el reconocimiento de la voz y otro para cerrar el dispositivo seleccionado.

IngresoGuest: Ventana que solo se muestra para los usuarios (hijos), con los dispositivos que tienen disponibles para poder abrir o cerrar cada uno de ellos.

Login: Activity donde se solicita el usuario y la contraseña, se validan con la conexión a la base de datos y en caso de no existir o estar incorrecta, se muestra un toast indicando que no se pudo ingresar y el motivo.

MainActivity: Activity donde se permite el registro del usuario administrador con todos los sus atributos como lo son: nombre, correo, apellido, clave, celular, contraseña del dispositivo.

MenuIngreso: Ventana que se muestra cuando un usuario administrador ingresa, tiene un botón para cada una de sus opciones que son: administrar dispositivos, administrar usuarios, registrar usuarios hijos y realizar el desbloqueo de la puerta.

RegistarUsuario: Activity en la cual se solicitan todos los parámetros del usuario hijo para poder registrarlo en la base da datos. Los parámetros solicitados son: nombre, apellido, usuario, correo, celular y la contraseña del diapositivo.

Adicionalmente tenemos dos clases de Java creadas para el funcionamiento interno de la aplicación.

Persona: Crea objeto de tipo Persona para guardar los parámetros solicitados dentro de los atributos de la clase. Los atributos están puestos en private por lo tanto cada uno de ello requiere el método set y get sobreescritos.

GuardadoUsuario: Clase que contiene variables estáticas para que puedan ser utilizadas en cualquier clase de la app, estas variables son: “usuarioUsando” guarda el usuario con el que se ingresó y la variable “parent” se actualiza solo cuando ingresa un usuario hijo y se coloca como valor el usuario administrador al que pertenece.

* **Código Arduino con comentarios.**

En el código comentado a continuación, se realiza la conexión de la base de datos al Arduino y además, como el Arduino controla el cierre y apertura de la puerta mediante la validación de que se encuentre junta un led y un fotorresistor.

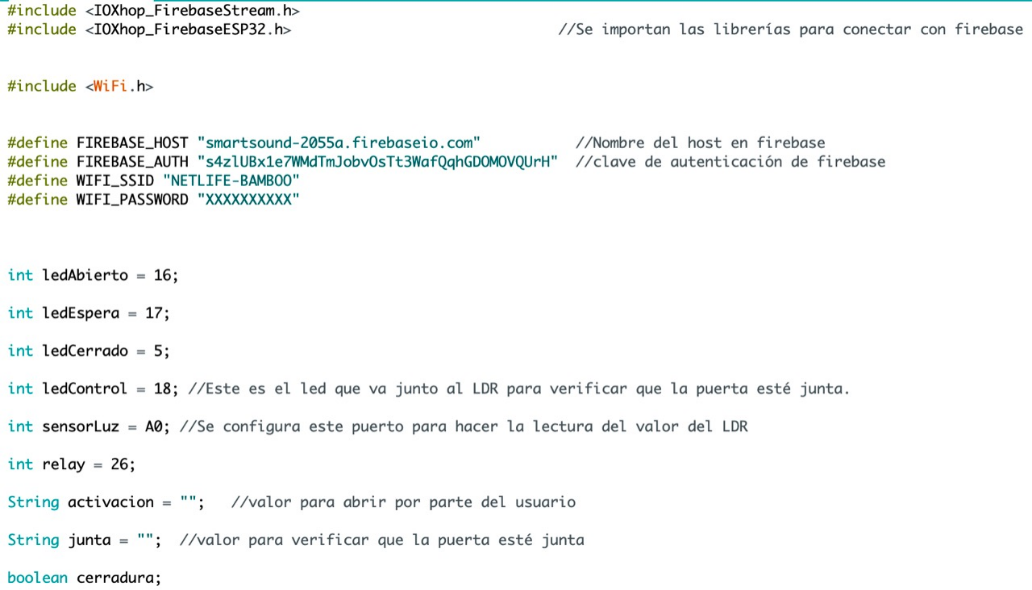


Ilustración 7. Código Arduino.

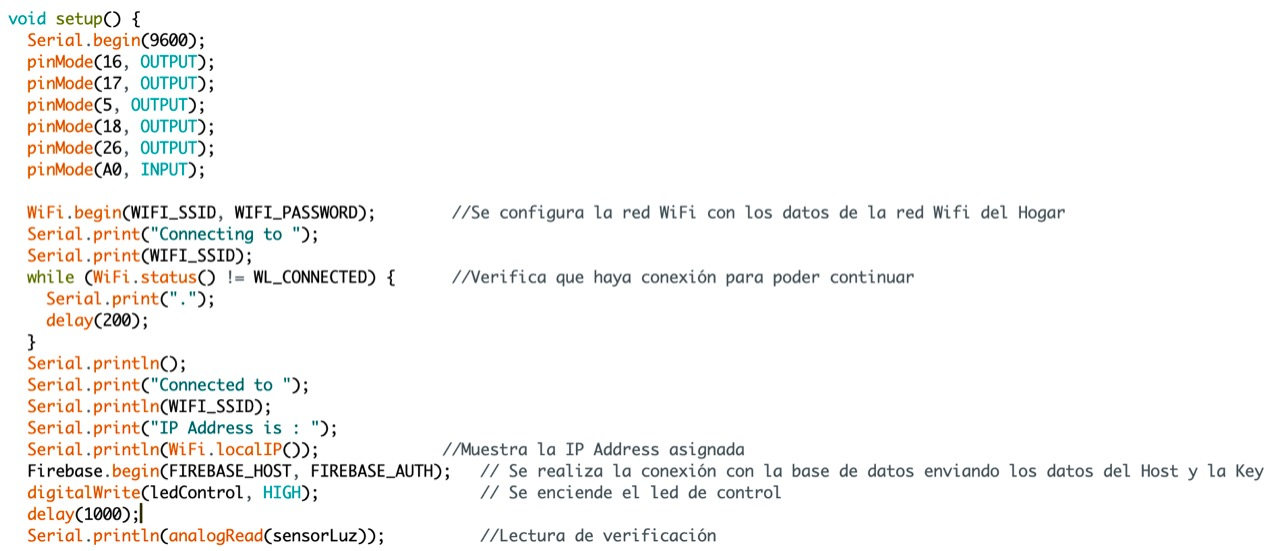


Ilustración 8. Código Arduino.

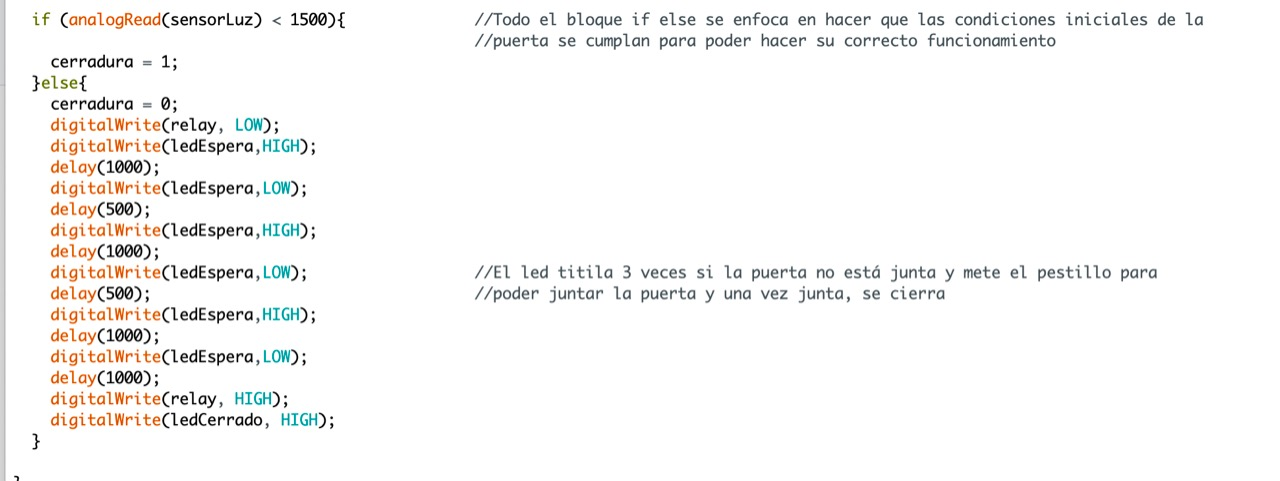


Ilustración 9. Código Arduino.

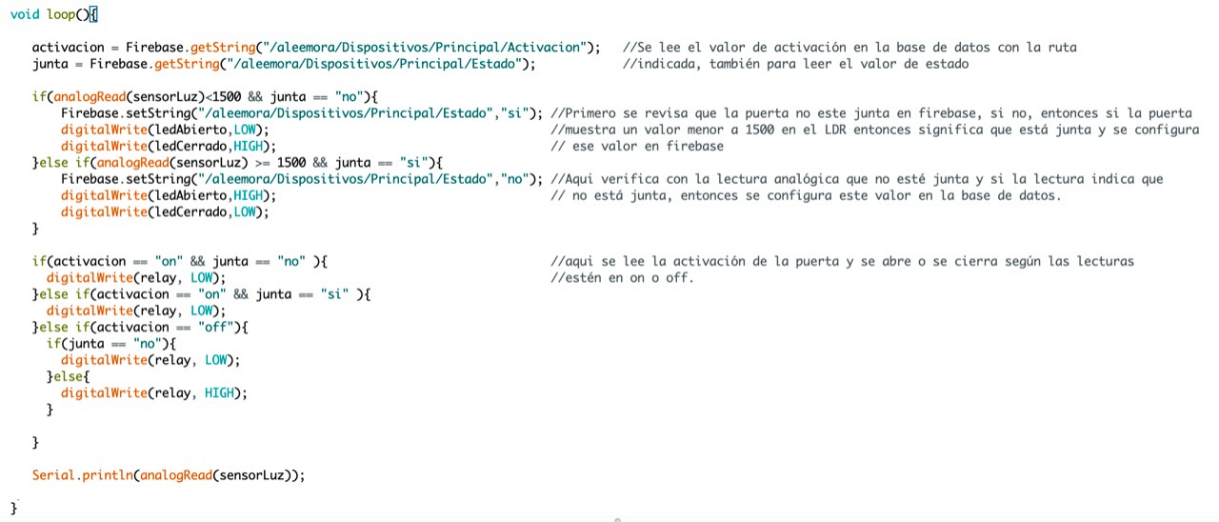


Ilustración 10. Código Arduino.

1. **Análisis de presupuesto**

|  |  |
| --- | --- |
| **Componente** | **Precio** |
| Módulo ESP32 Dev Kit | $25.75 |
| Solenoide cerradura 12 V | $8 |
| Visagra | $0.90 |
| Madera | $2 |
| Leds | $0.30 |
| jumpers | $1 |
| Relay 5V | $2.50 |
| Fotorresistor | $0.60 |
| **Total** | $41.05 |

Tabla 1. Tabla de análisis de presupuesto del proyecto.

1. **Conclusiones**

* La base de datos remota llamada Firebase Google fue utilizada para el almacenamiento de los datos de los dispositivos y sobre los usuarios registrados en la aplicación web.
* La aplicación móvil para el control de cierre y apertura de cerradura de una puerta en tiempo real bajo un usuario con clave registrado y su respectiva clave válida de control de dispositivos ingresada por voz, fue desarrollada mediante Android Studio y el circuito que controla la cerradura mediante Arduino IDE.
* El usuario administrador registrado en la aplicación recibió de forma automática notificaciones de alerta en caso de que se haya detectado actividad sospechosa a causa de un intento de apertura de la puerta fallido 4 veces seguidas.

1. **Referencias bibliográficas**

[1] A. Oduroye, “Voice Recognition Door Access Control System,” 2019, doi: 10.9790/0661-2105010112.

[2] W. Astuti and S. Mohamed, “Biometric Voice Recognition in Security System.,” *Indian J. Comput. Sci.*, vol. 3, 2007.

[3] L. Kamelia, M. Sanjaa, and W. S. E. Mulyana, “Door automation system using bluetooth-based android for mobile phone,” *J. Eng. Appl. Sci.*, vol. 9, 2014.