HouseTIC

Manual Técnico

**Materia:**

Programación de Sistemas Telemáticos

**Título:**

Proyecto de ingeniería usando plataformas de prototipado con servicios basados en red

**Profesora:**

Msig. Adriana E. Collaguazo Jaramillo

**Grupo:** 6

**Integrantes:**

Juan Guillermo Vera García

Alfredo Andrés Ante Vargas

Hugo Martin Andérica Urquizo

José Daniel Moreno González

**CONTENIDO**

[1. RESUMEN EJECTUTIVO 3](#_Toc17625704)

[2. INTRODUCCION 3](#_Toc17625705)

[3. DESCRIPCION DEL PROBLEMA 3](#_Toc17625706)

[4. OBJETIVOS ESPECIFICOS 3](#_Toc17625707)

[5. PLANTEAMIENTO DE LA SOLUCION 4](#_Toc17625708)

[6. RECURSOS 4](#_Toc17625709)

[8. DIAGRAMAS 6](#_Toc17625710)

[8](#_Toc17625711)

[9. CODIGO FUENTE 8](#_Toc17625712)

[10. ANALISIS DE PRESUPUESTO 30](#_Toc17625713)

[11. CONCLUSIONES 31](#_Toc17625714)

[12. REFERENCIAS 31](#_Toc17625715)

## RESUMEN EJECTUTIVO

Este proyecto llamado HouseTIC busca mejorar las comodidades del hogar mediante la posibilidad de controlar los distintos dispositivos dentro del mismo de manera remota, tanto como par cuando el usuario se encuentre dentro del sitio o fuera de este.

Su funcionamiento consiste en un aplicativo móvil desarrollado para teléfonos inteligentes con sistema operativo android. Dentro del mismo se presentara una interfaz que permitirá interactuar con el sistema de control remoto de una manera muy simple e intuitiva. Esta aplicación trabajara en conjunto con una base de datos en línea que guardara el estado de los dispositivos y los presentara al usuario mediante la aplicación.

El sistema de control imlementado funcionara en base a un módulo Wifi ESP32.

El sistema se encuentra en fase de prototipado a menor escala, pero está basado en los mismos principios que corresponderían a una implementación a escala real.

El costo final del prototipo se estima en $65, lo cual es un precio razonable para una primera prueba del sistema.

Este dispositivo de control domótico esta enfocado a todas aquellas personas de clase media alta que busquen una mayor comodidad tanto en su hogar como en su zona de trabajo. La mayor utilidad de este sistema radica en que al estar conectado con la base de datos permite al usuario interactuar con los dispositivos de su hogar desde cualquier parte del mundo mientras tenga acceso a internet.

## INTRODUCCION

El presente manual muestra de forma detallada toda la información técnica acerca del funcionamiento y desarrollo del sistema empleado para el control domótico de una casa promedio.

Se detalla el procedimiento de implementación del aplicativo móvil para el sistema operativo Android en el software Android Studio. Incluye una descripción de la funcionalidad del sistema, tanto en hardware como en software, así como un detalle del presupuesto general del proyecto.

## DESCRIPCION DEL PROBLEMA

El avance tecnológico ha hecho que el hombre se vuelva dependiente en gran medida de los diversos dispositivos electrónicos de uso cotidiano. En la actualidad, la gran mayoría de los hogares tienen el inconveniente de que, para activar un dispositivo electrónico, se debe dirigirse al mismo y accionarlo manualmente. Esto puede resultar un poco incómodo en casas de grandes dimensiones o en casos en los que el usuario no se encuentre en su domicilio y sin embargo requiera encender o apagar un dispositivo.

Este proyecto proporciona un sistema capaz de poder controlar todos estos dispositivos desde cualquier lugar, sea dentro de la casa o en cualquier otro sitio.

## OBJETIVOS ESPECIFICOS

* Ofrecer información técnica acerca de la configuración de la aplicación móvil.
* Especificar detalladamente la estructura del aplicativo y su diseño.
* Definir especificaciones acerca de requerimientos de hardware y software necesarios para el correcto funcionamiento de la aplicación.
* Describir los métodos y herramientas empleados para el desarrollo del aplicativo

## PLANTEAMIENTO DE LA SOLUCION

La solución consiste en facilitar el día a día en el hogar o área de trabajo, brindando la oportunidad de controlar dispositivos de uso cotidiano mediante solo un clic desde un smartphone con conexión a internet.

Para la realización de este proyecto se implementó un sistema constituido por sensores, motores y leds controlados mediante un Arduino Uno. Se utilizó una base de datos remota RemoteMySql, en la cual se registraron los estados de los dispositivos a controlar (encendido o apagado). Paralelamente se diseñó una aplicación móvil en Android Studio, la cual permite al usuario interactuar con el sistema y definir manualmente el estado de determinado equipo de determinada área de la casa. La base de datos lee los cambios realizados por el usuario a través de la aplicación móvil, y comunica esta información con el Arduino para controlar un determinado dispositivo.

## RECURSOS

Para la implementación del prototipo se utilizó el siguiente hardware:

* 1 Arduino UNO
* 10 Leds
* 2 Motor DC
* 2 Servo Motor
* 11 Relés
* 1 Fuente 5V
* 1 Módulo Wifi ESP32

Las herramientas de software empleadas fueron:

* Android Studio
* FireBase
* Arduino
* MicroPython

1. **IMPLEMENTACIÓN DEL PROYECTO**

**Aplicativo móvil**

Se inició creando el aplicativo en Android Studio. Se crearon 8 diferentes activities de las cuales el primero se empleó para una interfaz de login que funciona con un ingreso a través de una cuenta de Google.

El segundo activity consiste en un menú principal el cual se agregó un TextView “estaciones”, seguido de 6 botones cada uno con imágenes que indican su funcionalidad; 5 de ellos abren un activity diferente, cada uno para el control de un área específica del hogar. Dichos activities constan de un TextView correspondiente a cada dispositivo que se quiera controlar acompañado de un switch para cambiar su estado a activado o desactivado.

Por último, el botón restante del menú principal tiene como función salir del menú principal hacia la pagina de login.

**Base de Datos**

Se creo una base de datos en la nube RemoteMySQL, en esta se crearon 4 tablas: cuarto, cuenta, dispositivo y hogar. La tabla Cuenta almacena información de usuario, contraseña correo y número de teléfono; se fijó como calve primaria el nombre de usuario.

La tabla Hogar contiene la información de un hogar especifico en el cual se instalará el sistema. Sus datos son: un id que se usara como clave primaria, dirección, terreno, numero de cuartos que tiene el domicilio y tiene como clave foránea el nombre de usuario de la tabla Cuenta.

La tabla Cuarto fue creada para almacenar cuartos o estaciones del domicilio. Los datos que contiene son: id, nombre, área, numero de dispositivos y como clave foránea el id del hogar correspondiente.

Por último, está la tabla Dispositivos, que almacena los dispositivos que se van a controlar en un cuarto determinado. Tiene como datos un id, el nombre, descripción, estado (encendido o apagado), y como clave foránea el id del cuarto.

**Dispositivos electrónicos**

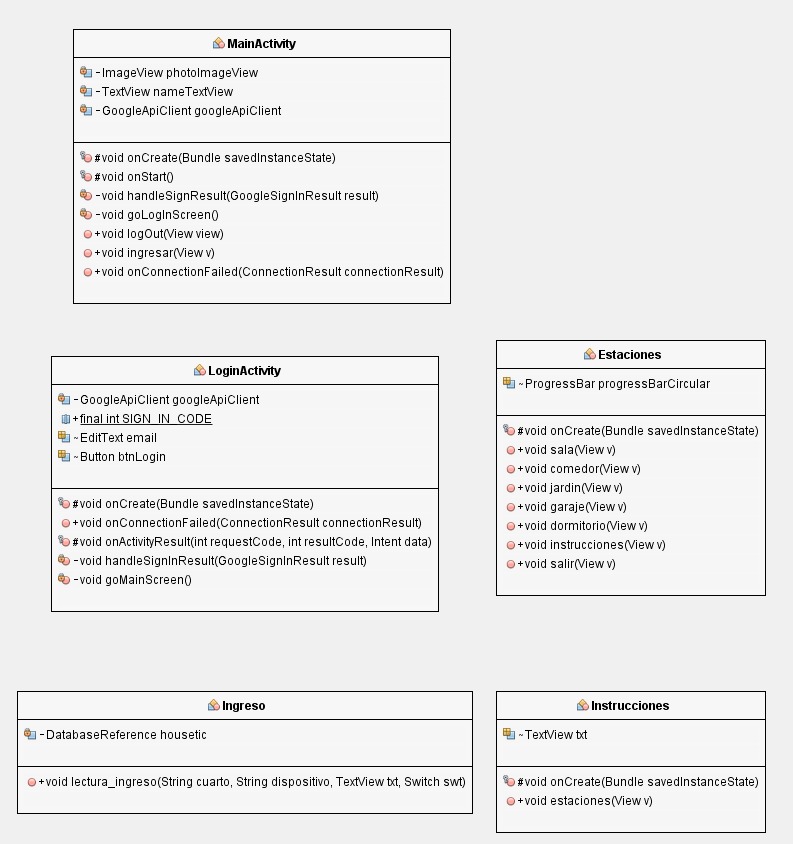
Para el circuito se utilizó un módulo ESP32 el cual se conectó por medio de 8 relés a los diferentes dispositivos. Se utilizó este módulo para darle conectividad a internet al sistema y permitirle acceder a la base de datos, para comunicarse con el aplicativo móvil y por ende, con el usuario.

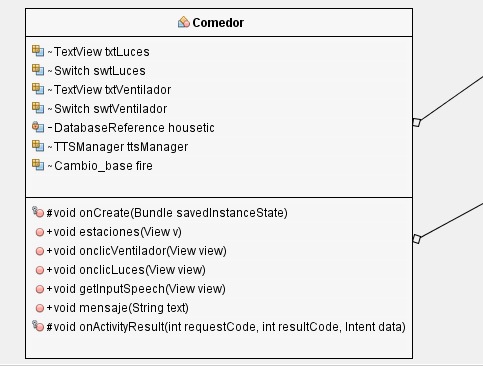
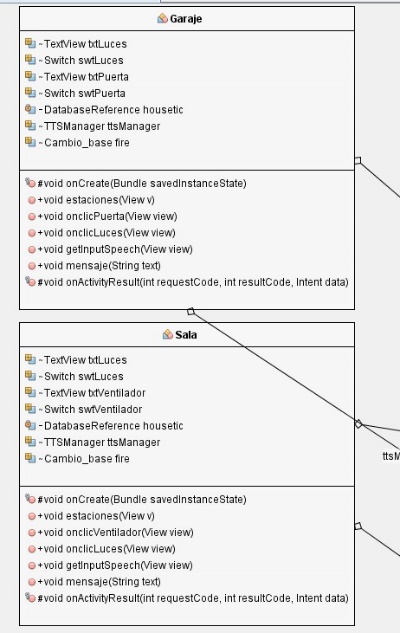
**Maqueta**

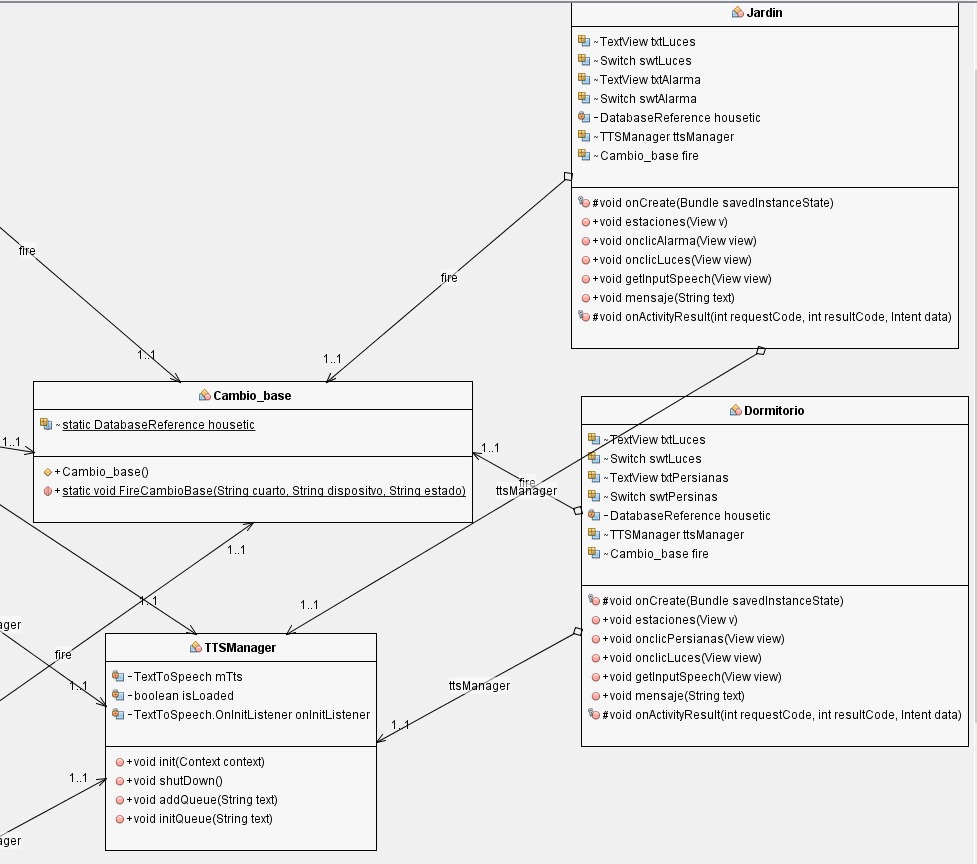
Para mostrar la funcionalidad del sistema implementado, se implementó una maqueta que consiste en una casa de juguete, separada por salas que representarían las estaciones y adaptada con dispositivos electrónicos demostrativos de funciones específicas del sistema, es decir, los leds para simular los focos eléctricos, motores dc para simular ventiladores, la alarma en conjunto con el sensor de proximidad, motores de paso y servo para controlar ventanas y puertas, etc.

## DIAGRAMAS

**Diagrama UML**

****

****

****

**Diagrama de Despliegue**

## 

Luz

Garaje

Puerta

Luz

Luz

Alarma

Ventilador

Luz

Ventilador

Luz

Persiana

Sala

Dormitorio

Comedor

Jardín

## CODIGO FUENTE

**Arduino**

El código fue programado en el módulo ESP32. Tiene como funcionalidad estar constantemente leyendo la base de datos y notando cambios, para asi cambiar el estado de los distintos dispositivos de determinada sala o estación. El código fuente se puede ver a continuación:

#include <ArduinoJson.h> // esp32 library for reading json code

#include <ESP32Servo.h> // servo motor library ti open/close door

Servo servo1; // create instance of the servos for the door

Servo servo2;

int servo1Pin = 32; // create a variable to save the value of the digital pin that they will be attached to

int servo2Pin = 33;

int pos = 0;

#include <WiFi.h> // esp32 WiFi library

#include <IOXhop\_FirebaseESP32.h> // firebase library

#define FIREBASE\_HOST "https://housetic.firebaseio.com"

#define FIREBASE\_AUTH "W6EQPDdqioYizI6r7ig7039ffkwK22LWUxh7p0ig" // the secret key generated from firebase

#define WIFI\_SSID "HUAWEI Mate 10" // input your home or public wifi name

#define WIFI\_PASSWORD "0f1991634d4d" //password of wifi ssid

String fireStatus = ""; // led status received from firebase

int lPins[] = {5, // comedor/luces

18, // dormitorio/persianas

19, // garage/luces

21, // dormitorio/luces

13, // jardin/alarma

12, // jardin/luces

14, // sala/luces

27 // sala/ventilador

};

String lDisp[]={"comedor/luces","dormitorio/persianas","garage/luces","dormitorio/luces",

"jardin/alarma","jardin/luces","sala/luces","sala/ventilador"};

void recorrerDisps(){ // method used to iterate each device that works with an On/Off logic (lights,fans, etc)

for (int i=0;i<8;i++){

fireStatus = Firebase.getString(lDisp[i]); // get led status input from firebase

if (fireStatus == "ON") { // compare the input of led status received from firebase

Serial.println(lDisp[i]+" ON");

digitalWrite(lPins[i], LOW); // make output led ON

}

else if (fireStatus == "OFF") { // compare the input of led status received from firebase

Serial.println(lDisp[i]+" OFF");

digitalWrite(lPins[i], HIGH); // make output led OFF

}

else {

Serial.println("Wrong Credential! Please send ON/OFF"); // Validation that is throw if there is a manual change in data base

}

}

}

void controlPuerta(){

// cambiar estado de puerta (abierta o cerrada)

fireStatus = Firebase.getString("garage/puertas"); // get servo status input from firebase

if (fireStatus == "ON") { // compare the input of servos status received from firebase

Serial.println("Puerta abierta");

servo1.write(0); // make output servos go to position 0

servo2.write(0);

}

else if (fireStatus == "OFF") { // compare the input of servos status received from firebase

Serial.println("Puerta cerrada");

servo1.write(180); // make output servos go to position 180

servo2.write(180);

}

else {

Serial.println("Wrong Credential! Please send ON/OFF"); // Validation that is throw if there is a manual change in data base

}

}

void setup() { // this code will run once whe the code runs

Serial.begin(9600); // begin to work in Serial screen at a 9600 baud

delay(1000);

servo1.attach(servo1Pin); // attach the servos to the specified digital pins

servo2.attach(servo2Pin);

for (int i=0;i<8;i++){ // initialize each pin that works with On/Off logic as an output

pinMode(lPins[i],OUTPUT);

}

WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD); //try to connect with wifi

Serial.print("Connecting to ");

Serial.print(WIFI\_SSID);

while (WiFi.status() != WL\_CONNECTED) {

Serial.print(".");

delay(500);

}

Serial.println();

Serial.print("Connected to ");

Serial.println(WIFI\_SSID);

Serial.print("IP Address is : ");

Serial.println(WiFi.localIP()); //print local IP address

Firebase.begin(FIREBASE\_HOST, FIREBASE\_AUTH); // connect to firebase

}

void loop() { //loop that will run continiusly until the code stops

recorrerDisps(); // calling methos to iterate over all the devices and the servo

controlPuerta();

Serial.println("Fin"); // End of one of the void loops

delay(100);

}

**Android Studio**

Este código es el que rige el funcionamiento del aplicativo móvil. Su estructura se subdivide de la siguiente manera:

**Ingreso**

package com.example.domotic\_project;  
  
import android.content.Context;  
import android.widget.Switch;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
  
public class Ingreso {  
  
 private DatabaseReference housetic;//inicializacion de DatabaseRefenrence  
  
  
 public void lectura\_ingreso(final String cuarto, final String dispositivo, final TextView txt, final Switch swt) {//ingreso de funcion que hara lectura de base de datos  
  
 housetic = FirebaseDatabase.*getInstance*().getReference(); //ingreso a base de datos  
 housetic.child(cuarto);  
 housetic.addValueEventListener(new ValueEventListener() {  
 @Override  
  
 public void onDataChange(DataSnapshot dataSnapshot) {//buscar a traves del padre e hijo el estado del dispositivo  
 if (dataSnapshot.exists()) {  
 String resultadoConsulta = dataSnapshot.child(dispositivo).getValue().toString();  
 if (resultadoConsulta.equalsIgnoreCase("OFF")) {  
 swt.setChecked(false);  
 txt.setText("Desactivado");  
 } else {  
 swt.setChecked(true);  
 txt.setText("Activado");  
 }  
 }  
 }  
  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 }  
 });  
 }  
  
}

**Cambio de estado de la base de datos**

package com.example.domotic\_project;  
  
import android.view.View;  
import android.widget.Switch;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
  
import java.util.concurrent.ExecutionException;  
  
public class Cambio\_base {  
  
  
 static DatabaseReference *housetic*;  
  
 public Cambio\_base() {  
  
 }  
  
  
 public static void FireCambioBase(String cuarto,String dispositvo,String estado){//realizar cambio de estado en la base de datos  
  
 *housetic* = FirebaseDatabase.*getInstance*().getReference();  
 *housetic*.child(cuarto).child(dispositvo).setValue(estado);  
 }  
}

**Sala**

package com.example.domotic\_project;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.speech.RecognizerIntent;  
import android.view.View;  
import android.widget.ProgressBar;  
import android.widget.Switch;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
  
import java.util.ArrayList;  
import java.util.Locale;  
  
public class Sala extends AppCompatActivity {  
 //Inicializacion de variables TextView, Switch, DatabaseRefence, Cambio\_base, TTSManager  
 TextView txtLuces;  
 Switch swtLuces;  
 TextView txtVentilador;  
 Switch swtVentilador;  
 private DatabaseReference housetic;  
 TTSManager ttsManager=null;  
 Cambio\_base fire = new Cambio\_base();  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_sala*);  
 ttsManager=new TTSManager(); //creacion del objeto que ejecutara la voz de la aplicacion  
 ttsManager.init(this);  
 txtLuces = (TextView) findViewById(R.id.*textView\_Luces1*); //llamado del TextView del activity  
 txtVentilador = (TextView) findViewById(R.id.*textView\_Ventilador1*);  
 swtVentilador= (Switch) findViewById(R.id.*switch\_Ventilador1*); //llamado del switch del activity  
 swtLuces = (Switch) findViewById(R.id.*switch\_Luces1*);  
  
  
 housetic = FirebaseDatabase.*getInstance*().getReference(); //ingreso a la base de datos  
 housetic.child("sala").addValueEventListener(new ValueEventListener() { //busqueada por padre e hijo del diapositivo  
 @Override  
  
 public void onDataChange(DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 String resultadoConsultaLuces= dataSnapshot.child("luces").getValue().toString();  
 if (resultadoConsultaLuces.equalsIgnoreCase("OFF")) {  
 swtLuces.setChecked(false);  
 txtLuces.setText("Desactivado");  
 Toast.*makeText*(Sala.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 } else {  
 swtLuces.setChecked(true);  
 txtLuces.setText("Activado");  
 Toast.*makeText*(Sala.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 }  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 }  
 });  
 housetic.child("sala").addValueEventListener(new ValueEventListener() { //busqueada por padre e hijo del diapositivo  
 @Override  
  
 public void onDataChange(DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 String resultadoConsultaVentilador= dataSnapshot.child("ventilador").getValue().toString();  
 if (resultadoConsultaVentilador.equalsIgnoreCase("OFF")) {  
 swtVentilador.setChecked(false);  
 txtVentilador.setText("Desactivado");  
 Toast.*makeText*(Sala.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 } else {  
 swtVentilador.setChecked(true);  
 txtVentilador.setText("Activado");  
 Toast.*makeText*(Sala.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 }  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 }  
 });  
  
 }  
  
 public void estaciones(View v) { //implementacion de funcion estaciones, redirrecionamiento del cuarto a estaciones  
 Intent i = new Intent(this, Estaciones.class);  
 startActivity(i);  
  
 }  
  
  
 public void onclicVentilador(View view){ //implementacion de funcion onClic para el ventilador  
  
 if (view.getId() == R.id.*switch\_Ventilador1*){  
 if (swtVentilador.isChecked()){  
 txtVentilador.setText("Activado");  
 fire.*FireCambioBase*("sala","ventilador","ON"); //implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a ON  
 ttsManager.initQueue("Ventilador activado"); //Ejecucion de voz del cambio de estado  
 Toast.*makeText*(Sala.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 }else{  
 txtVentilador.setText("Desactivado");  
 fire.*FireCambioBase*("sala","ventilador","OFF"); //implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a OFF  
 ttsManager.initQueue("Ventilador desactivado"); //Ejecucion de voz del cambio de estado  
 Toast.*makeText*(Sala.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 }  
  
 public void onclicLuces(View view){  
  
 if (view.getId() == R.id.*switch\_Luces1*){  
 if (swtLuces.isChecked()){  
 txtLuces.setText("Activado");  
 fire.*FireCambioBase*("sala","luces","ON"); //implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a ON  
 ttsManager.initQueue("Luces activadas"); //Ejecucion de voz del cambio de estado  
 Toast.*makeText*(Sala.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 }else{  
 txtLuces.setText("Desactivado");  
 fire.*FireCambioBase*("sala","luces","OFF"); //implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a OFF  
 ttsManager.initQueue("Luces desactivadas"); //Ejecucion de voz del cambio de estado  
 Toast.*makeText*(Sala.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 }  
  
  
 public void getInputSpeech(View view){  
 Intent intent = new Intent(RecognizerIntent.*ACTION\_RECOGNIZE\_SPEECH*);  
 intent.putExtra(RecognizerIntent.*EXTRA\_LANGUAGE\_MODEL*, RecognizerIntent.*LANGUAGE\_MODEL\_FREE\_FORM*);  
 intent.putExtra(RecognizerIntent.*EXTRA\_LANGUAGE*, Locale.*getDefault*());  
 if(intent.resolveActivity(getPackageManager())!= null){  
 startActivityForResult(intent, 10);  
 } else {  
 mensaje("Tu dispositivo no reconoce entrada de voz");  
 }  
 }  
  
 public void mensaje(String text){  
 Toast.*makeText*(Sala.this,text, Toast.*LENGTH\_LONG*).show();  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data){  
 super.onActivityResult(requestCode,resultCode,data);  
 switch (requestCode){  
 case 10:  
 if(resultCode == *RESULT\_OK* && data != null){  
 ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.*EXTRA\_RESULTS*);  
 String comando = result.get(0);  
 if(comando.equalsIgnoreCase("Encender luces")){  
 fire.*FireCambioBase*("sala","luces","ON");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a ON  
 ttsManager.initQueue("Luces encendidas"); //Ejecucion de voz del cambio de estado  
 mensaje("Luces encendidas");  
 } else if( comando.equalsIgnoreCase("Apagar luces")){  
 fire.*FireCambioBase*("sala","luces","OFF");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a OFF  
 ttsManager.initQueue("Luces apagadas"); //Ejecucion de voz del cambio de estado  
 mensaje("Luces apagadas");  
 }else if(comando.equalsIgnoreCase("Encender ventilador")){  
 fire.*FireCambioBase*("sala","ventilador","ON");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a ON  
 ttsManager.initQueue("Ventilador encendido"); //Ejecucion de voz del cambio de estado  
 mensaje("Ventilador encendido");  
 } else if( comando.equalsIgnoreCase("Apagar ventilador")){  
 fire.*FireCambioBase*("sala","ventilador","OFF");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a OFF  
 ttsManager.initQueue("Ventilador apagado"); //Ejecucion de voz del cambio de estado  
 mensaje("Ventilador apagado");  
 }  
 }  
 break;  
 }  
 }  
}

**Dormitorio**

package com.example.domotic\_project;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.speech.RecognizerIntent;  
import android.view.View;  
import android.widget.Switch;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
  
import java.util.ArrayList;  
import java.util.Locale;  
  
public class Dormitorio extends AppCompatActivity {  
 //Inicializacion de variables TextView, Switch, DatabaseRefence, Cambio\_base, TTSManager  
 TextView txtLuces;  
 Switch swtLuces;  
 TextView txtPersianas;  
 Switch swtPersinas;  
 private DatabaseReference housetic;  
 TTSManager ttsManager=null;  
 Cambio\_base fire = new Cambio\_base();  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 ttsManager=new TTSManager(); //creacion del objeto que ejecutara la voz de la aplicacion  
 ttsManager.init(this);  
 setContentView(R.layout.*activity\_dormitorio*);  
 txtLuces = (TextView) findViewById(R.id.*textView\_Luces2*); //llamado del TextView del activity  
 txtPersianas = (TextView) findViewById(R.id.*textView\_Persianas2*);  
 swtPersinas = (Switch) findViewById(R.id.*switch\_Persianas2*); //llamado del switch del activity  
 swtLuces = (Switch) findViewById(R.id.*switch\_Luces2*);  
  
 String cuarto = "dormitorio"; //variables que se utilizara para ejecutar en la base de datos  
 final String dispositivo1 = "luces";  
 final String dispositivo2 = "persianas";  
 housetic = FirebaseDatabase.*getInstance*().getReference(); //ingreso a la base de datos  
 housetic.child(cuarto).addValueEventListener(new ValueEventListener() { //busqueada por padre e hijo del diapositivo  
 @Override  
  
 public void onDataChange(DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 String resultadoConsultaLuces= dataSnapshot.child(dispositivo1).getValue().toString();  
 if (resultadoConsultaLuces.equalsIgnoreCase("OFF")) {  
 swtLuces.setChecked(false);  
 txtLuces.setText("Desactivado");  
 mensaje("Conexión Establecida");  
 } else {  
 swtLuces.setChecked(true);  
 txtLuces.setText("Activado");  
 mensaje("Conexión Establecida");  
 }  
  
 }  
  
 }  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 }  
 });  
 housetic.child(cuarto).addValueEventListener(new ValueEventListener() { //busqueada por padre e hijo del diapositivo  
 @Override  
  
 public void onDataChange(DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 String resultadoConsultaPersianas= dataSnapshot.child(dispositivo2).getValue().toString();  
 if (resultadoConsultaPersianas.equalsIgnoreCase("OFF")) {  
 swtPersinas.setChecked(false);  
 txtPersianas.setText("Desactivado");  
 mensaje("Conexión Establecida");  
 } else {  
 swtPersinas.setChecked(true);  
 txtPersianas.setText("Activado" );  
 mensaje("Conexión Establecida");  
 }  
  
 }  
  
 }  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 }  
 });  
 }  
  
 public void estaciones(View v) { //implementacion de funcion estaciones, redirrecionamiento del cuarto a estaciones  
 Intent i = new Intent(this, Estaciones.class );  
 startActivity(i);  
  
 }  
  
 public void onclicPersianas(View view){ //implementacion de funcion onClic para las persianas  
  
 if (view.getId() == R.id.*switch\_Persianas2*){  
 if (swtPersinas.isChecked()){  
 txtPersianas.setText("Activado");  
 fire.*FireCambioBase*("dormitorio","persianas","ON"); //implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a ON  
 ttsManager.initQueue("Persianas elevadas"); //Ejecucion de voz del cambio de estado  
 mensaje("Persianas elevadas");  
 }else{  
 txtPersianas.setText("Desactivado");  
 fire.*FireCambioBase*("dormitorio","persianas","OFF");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a OFF  
 ttsManager.initQueue("Persianas desplegadas"); //Ejecucion de voz del cambio de estado  
 mensaje("Persianas desplegadas");  
 }  
 }  
 }  
  
 public void onclicLuces(View view){  
  
 if (view.getId() == R.id.*switch\_Luces2*){  
 if (swtLuces.isChecked()){  
 txtLuces.setText("Activado");  
 fire.*FireCambioBase*("dormitorio","luces","ON"); //implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a ON  
 ttsManager.initQueue("Luces encendidas"); //Ejecucion de voz del cambio de estado  
 mensaje("Luces encendidas");  
 }else{  
 txtLuces.setText("Desactivado");  
 fire.*FireCambioBase*("dormitorio","luces","OFF"); //implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a OFF  
 ttsManager.initQueue("Luces apagadas"); //Ejecucion de voz del cambio de estado  
 mensaje("Luces apagadas");  
 }  
 }  
  
 }  
  
 public void getInputSpeech(View view){  
 Intent intent = new Intent(RecognizerIntent.*ACTION\_RECOGNIZE\_SPEECH*);  
 intent.putExtra(RecognizerIntent.*EXTRA\_LANGUAGE\_MODEL*, RecognizerIntent.*LANGUAGE\_MODEL\_FREE\_FORM*);  
 intent.putExtra(RecognizerIntent.*EXTRA\_LANGUAGE*, Locale.*getDefault*());  
 if(intent.resolveActivity(getPackageManager())!= null){  
 startActivityForResult(intent, 10);  
 } else {  
 mensaje("Tu dispositivo no reconoce entrada de voz");  
 }  
 }  
  
 public void mensaje(String text){  
 Toast.*makeText*(Dormitorio.this,text, Toast.*LENGTH\_LONG*).show();  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data){  
 super.onActivityResult(requestCode,resultCode,data);  
 switch (requestCode){  
 case 10:  
 if(resultCode == *RESULT\_OK* && data != null){  
 ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.*EXTRA\_RESULTS*);  
 String comando = result.get(0);  
 if(comando.equalsIgnoreCase("Encender luces")){  
 fire.*FireCambioBase*("dormitorio","luces","ON");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a ON  
 ttsManager.initQueue("Luces encendidas"); //Ejecucion de voz del cambio de estado  
 mensaje("Luces encendidas");  
 } else if( comando.equalsIgnoreCase("Apagar luces")){  
 fire.*FireCambioBase*("dormitorio","luces","OFF");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a OFF  
 ttsManager.initQueue("Luces apagadas"); //Ejecucion de voz del cambio de estado  
 mensaje("Luces apagadas");  
 }else if(comando.equalsIgnoreCase("Elevar persianas")){  
 fire.*FireCambioBase*("dormitorio","persianas","ON");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a ON  
 ttsManager.initQueue("Persianas elevadas"); //Ejecucion de voz del cambio de estado  
 mensaje("Persianas elevadas");  
 } else if( comando.equalsIgnoreCase("Bajar persianas")){  
 fire.*FireCambioBase*("dormitorio","persianas","OFF");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a OFF  
 ttsManager.initQueue("Persianas desplegadas"); //Ejecucion de voz del cambio de estado  
 mensaje("Persianas desplegadas");  
 }  
 }  
 break;  
 }  
 }  
  
}

**Garaje**

package com.example.domotic\_project;  
  
 import androidx.annotation.NonNull;  
 import androidx.appcompat.app.AppCompatActivity;  
  
 import android.content.Intent;  
 import android.os.Bundle;  
 import android.speech.RecognizerIntent;  
 import android.view.View;  
 import android.widget.Switch;  
 import android.widget.TextView;  
 import android.widget.Toast;  
  
 import com.google.firebase.database.DataSnapshot;  
 import com.google.firebase.database.DatabaseError;  
 import com.google.firebase.database.DatabaseReference;  
 import com.google.firebase.database.FirebaseDatabase;  
 import com.google.firebase.database.ValueEventListener;  
  
 import java.util.ArrayList;  
 import java.util.Locale;  
  
public class Garaje extends AppCompatActivity {  
 //Inicializacion de variables TextView, Switch, DatabaseRefence, Cambio\_base, TTSManager  
 TextView txtLuces;  
 Switch swtLuces;  
 TextView txtPuerta;  
 Switch swtPuerta;  
 private DatabaseReference housetic;  
 TTSManager ttsManager=null;  
 Cambio\_base fire = new Cambio\_base();  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_garaje*);  
  
 txtLuces = (TextView) findViewById(R.id.*textView\_Luces3*); //llamado del TextView del activity  
 txtPuerta = (TextView) findViewById(R.id.*textView\_Puerta3*);  
 swtPuerta = (Switch) findViewById(R.id.*switch\_Puerta3*); //llamado del switch del activity  
 swtLuces = (Switch) findViewById(R.id.*switch\_Luces3*);  
 ttsManager=new TTSManager(); //creacion del objeto que ejecutara la voz de la aplicacion  
 ttsManager.init(this);  
  
 String cuarto = "garage"; //variables que se utilizara para ejecutar en la base de datos  
 final String dispositivo1 = "luces";  
 final String dispositivo2 = "puertas";  
 housetic = FirebaseDatabase.*getInstance*().getReference(); //ingreso a la base de datos  
 housetic.child(cuarto).addValueEventListener(new ValueEventListener() { //busqueada por padre e hijo del diapositivo  
 @Override  
  
 public void onDataChange(DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 String resultadoConsultaLuces= dataSnapshot.child(dispositivo1).getValue().toString();  
 if (resultadoConsultaLuces.equalsIgnoreCase("OFF")) {  
 swtLuces.setChecked(false);  
 txtLuces.setText("Desactivado");  
 Toast.*makeText*(Garaje.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 } else {  
 swtLuces.setChecked(true);  
 txtLuces.setText("Activado" );  
 Toast.*makeText*(Garaje.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 }  
  
 }  
  
 }  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 }  
 });  
 housetic.child(cuarto).addValueEventListener(new ValueEventListener() { //busqueada por padre e hijo del diapositivo  
 @Override  
  
 public void onDataChange(DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 String resultadoConsultaPuerta= dataSnapshot.child(dispositivo2).getValue().toString();  
 if (resultadoConsultaPuerta.equalsIgnoreCase("OFF")) {  
 swtPuerta.setChecked(false);  
 txtPuerta.setText("Desactivado");  
 Toast.*makeText*(Garaje.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 } else {  
 swtPuerta.setChecked(true);  
 txtPuerta.setText("Activado");  
 Toast.*makeText*(Garaje.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 }  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 }  
 });  
 }  
 public void estaciones(View v) { //implementacion de funcion estaciones, redirrecionamiento del cuarto a estaciones  
 Intent i = new Intent(this, Estaciones.class );  
 startActivity(i);  
  
 }  
 public void onclicPuerta(View view){ //implementacion de funcion onClic para las puertas  
  
 if (view.getId() == R.id.*switch\_Puerta3*){  
 if (swtPuerta.isChecked()){  
 txtPuerta.setText("Activado");  
  
 fire.*FireCambioBase*("garage","puertas","ON"); //implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a ON  
 ttsManager.initQueue("Puertas abiertas");  
 Toast.*makeText*(Garaje.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 }else{  
 txtPuerta.setText("Desactivado");  
  
 fire.*FireCambioBase*("garage","puertas","OFF");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a OFF  
 ttsManager.initQueue("Puertas cerradas");  
 Toast.*makeText*(Garaje.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 }  
  
 public void onclicLuces(View view){  
  
 if (view.getId() == R.id.*switch\_Luces3*){  
 if (swtLuces.isChecked()){  
 txtLuces.setText("Activado");  
  
 fire.*FireCambioBase*("garage","luces","ON"); //implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a ON  
 ttsManager.initQueue("Luces encendidas"); //Ejecucion de voz del cambio de estado  
 Toast.*makeText*(Garaje.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 }else{  
 txtLuces.setText("Desactivado");  
  
 fire.*FireCambioBase*("garage","luces","OFF"); //implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a OFF  
 ttsManager.initQueue("Luces apagadas"); //Ejecucion de voz del cambio de estado  
 Toast.*makeText*(Garaje.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 }  
  
 public void getInputSpeech(View view){  
 Intent intent = new Intent(RecognizerIntent.*ACTION\_RECOGNIZE\_SPEECH*);  
 intent.putExtra(RecognizerIntent.*EXTRA\_LANGUAGE\_MODEL*, RecognizerIntent.*LANGUAGE\_MODEL\_FREE\_FORM*);  
 intent.putExtra(RecognizerIntent.*EXTRA\_LANGUAGE*, Locale.*getDefault*());  
 if(intent.resolveActivity(getPackageManager())!= null){  
 startActivityForResult(intent, 10);  
 } else {  
 mensaje("Tu dispositivo no reconoce entrada de voz");  
 }  
 }  
  
 public void mensaje(String text){  
 Toast.*makeText*(Garaje.this,text, Toast.*LENGTH\_LONG*).show();  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data){  
 super.onActivityResult(requestCode,resultCode,data);  
 switch (requestCode){  
 case 10:  
 if(resultCode == *RESULT\_OK* && data != null){  
 ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.*EXTRA\_RESULTS*);  
 String comando = result.get(0);  
 if(comando.equalsIgnoreCase("Encender luces")){  
 fire.*FireCambioBase*("garage","luces","ON");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a ON  
 ttsManager.initQueue("Luces encendidas"); //Ejecucion de voz del cambio de estado  
 mensaje("Luces encendidas");  
 } else if( comando.equalsIgnoreCase("Apagar luces")){  
 fire.*FireCambioBase*("garage","luces","OFF");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a OFF  
 ttsManager.initQueue("Luces apagadas"); //Ejecucion de voz del cambio de estado  
 mensaje("Luces apagadas");  
 }else if(comando.equalsIgnoreCase("Abrir puertas")){  
 fire.*FireCambioBase*("garage","puertas","ON");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a ON  
 ttsManager.initQueue("Puertas abiertas"); //Ejecucion de voz del cambio de estado  
 mensaje("Puertas abiertas");  
 } else if( comando.equalsIgnoreCase("Cerrar puertas")){  
 fire.*FireCambioBase*("garage","puertas","OFF");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a OFF  
 ttsManager.initQueue("Puertas cerradas"); //Ejecucion de voz del cambio de estado  
 mensaje("Puertas cerradas");  
 }  
 }  
 break;  
 }  
 }  
}

**Comedor**

package com.example.domotic\_project;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.speech.RecognizerIntent;  
import android.view.View;  
import android.widget.Switch;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
  
import java.util.ArrayList;  
import java.util.Locale;  
  
public class Comedor extends AppCompatActivity {  
 //Inicializacion de variables TextView, Switch, DatabaseRefence, Cambio\_base, TTSManager  
 TextView txtLuces;  
 Switch swtLuces;  
 TextView txtVentilador;  
 Switch swtVentilador;  
 private DatabaseReference housetic;  
 TTSManager ttsManager=null;  
 Cambio\_base fire = new Cambio\_base();  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_comedor*);  
 txtLuces = (TextView) findViewById(R.id.*textView\_Luces5*); //llamado del TextView del activity  
 txtVentilador = (TextView) findViewById(R.id.*textView\_Ventilador5*);  
 swtVentilador= (Switch) findViewById(R.id.*switch\_Ventilador5*); //llamado del switch del activity  
 swtLuces = (Switch) findViewById(R.id.*switch\_Luces5*);  
 ttsManager=new TTSManager(); //creacion del objeto que ejecutara la voz de la aplicacion  
 ttsManager.init(this);  
  
 final String cuarto = "comedor"; //variables que se utilizara para ejecutar en la base de datos  
 final String dispositivo1 = "luces";  
 final String dispositivo2 = "ventilador";  
 housetic = FirebaseDatabase.*getInstance*().getReference(); //ingreso a la base de datos  
 housetic.child(cuarto).addValueEventListener(new ValueEventListener() { //busqueada por padre e hijo del diapositivo  
 @Override  
  
 public void onDataChange(DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 String resultadoConsultaLuces= dataSnapshot.child(dispositivo1).getValue().toString();  
 if (resultadoConsultaLuces.equalsIgnoreCase("OFF")) {  
 swtLuces.setChecked(false);  
 txtLuces.setText("Desactivado");  
 Toast.*makeText*(Comedor.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 } else {  
 swtLuces.setChecked(true);  
 txtLuces.setText("Activado");  
 Toast.*makeText*(Comedor.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 }  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 }  
 });  
 housetic.child(cuarto).addValueEventListener(new ValueEventListener() { //busqueada por padre e hijo del diapositivo  
 @Override  
  
 public void onDataChange(DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 String resultadoConsultaVentilador= dataSnapshot.child(dispositivo2).getValue().toString();  
 if (resultadoConsultaVentilador.equalsIgnoreCase("OFF")) {  
 swtVentilador.setChecked(false);  
 txtVentilador.setText("Desactivado");  
 Toast.*makeText*(Comedor.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 } else {  
 swtVentilador.setChecked(true);  
 txtVentilador.setText("Activado");  
 Toast.*makeText*(Comedor.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 }  
  
 }  
  
 }  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 }  
 });  
  
  
 }  
 public void estaciones(View v) { //implementacion de funcion estaciones, redirrecionamiento del cuarto a estaciones  
 Intent i = new Intent(this, Estaciones.class );  
 startActivity(i);  
  
 }  
  
 public void onclicVentilador(View view){ //implementacion de funcion onClic para el ventilador  
  
 if (view.getId() == R.id.*switch\_Ventilador5*){  
 if (swtVentilador.isChecked()){  
 txtVentilador.setText("Activado");  
  
 fire.*FireCambioBase*("comedor","ventilador","ON"); //implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a ON  
 ttsManager.initQueue("Ventilador encendido");  
  
 Toast.*makeText*(Comedor.this,"Ventilador encendido", Toast.*LENGTH\_LONG*).show();  
 }else{  
 txtVentilador.setText("Desactivado");  
  
 fire.*FireCambioBase*("comedor","ventilador","OFF");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a OFF  
 ttsManager.initQueue("Ventilador apagado");  
 Toast.*makeText*(Comedor.this,"Ventilador apagado", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 }  
  
 public void onclicLuces(View view){  
  
 if (view.getId() == R.id.*switch\_Luces5*){  
 if (swtLuces.isChecked()){  
 txtLuces.setText("Activado");  
  
 fire.*FireCambioBase*("comedor","luces","ON"); //implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a ON  
 ttsManager.initQueue("Luces encendidas"); //Ejecucion de voz del cambio de estado  
 Toast.*makeText*(Comedor.this,"Luces encendidas", Toast.*LENGTH\_LONG*).show();  
 }else{  
 txtLuces.setText("Desactivado");  
  
 fire.*FireCambioBase*("comedor","luces","OFF");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a OFF  
 ttsManager.initQueue("Luces apagadas"); //Ejecucion de voz del cambio de estado  
 Toast.*makeText*(Comedor.this,"Luces apagadas", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
  
 }  
  
  
 public void getInputSpeech(View view){  
 Intent intent = new Intent(RecognizerIntent.*ACTION\_RECOGNIZE\_SPEECH*);  
 intent.putExtra(RecognizerIntent.*EXTRA\_LANGUAGE\_MODEL*, RecognizerIntent.*LANGUAGE\_MODEL\_FREE\_FORM*);  
 intent.putExtra(RecognizerIntent.*EXTRA\_LANGUAGE*, Locale.*getDefault*());  
 if(intent.resolveActivity(getPackageManager())!= null){  
 startActivityForResult(intent, 10);  
 } else {  
 mensaje("Tu dispositivo no reconoce entrada de voz");  
 }  
 }  
  
 public void mensaje(String text){  
 Toast.*makeText*(Comedor.this,text, Toast.*LENGTH\_LONG*).show();  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data){  
 super.onActivityResult(requestCode,resultCode,data);  
 switch (requestCode){  
 case 10:  
 if(resultCode == *RESULT\_OK* && data != null){  
 ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.*EXTRA\_RESULTS*);  
 String comando = result.get(0);  
 if(comando.equalsIgnoreCase("Encender luces")){// validacion de comando de voz  
 fire.*FireCambioBase*("comedor","luces","ON");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a ON  
 ttsManager.initQueue("Luces encendidas"); //Ejecucion de voz del cambio de estado  
 mensaje("Luces encendidas");  
 } else if( comando.equalsIgnoreCase("Apagar luces")){// validacion de comando de voz  
 fire.*FireCambioBase*("comedor","luces","OFF");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a OFF  
 ttsManager.initQueue("Luces apagadas"); //Ejecucion de voz del cambio de estado  
 mensaje("Luces apagadas");  
 }else if(comando.equalsIgnoreCase("Encender ventilador")){// validacion de comando de voz  
 fire.*FireCambioBase*("comedor","ventilador","ON");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a ON  
 ttsManager.initQueue("Ventilador encendido"); //Ejecucion de voz del cambio de estado  
 mensaje("Ventilador encendido");  
 } else if( comando.equalsIgnoreCase("Apagar ventilador")){// validacion de comando de voz  
 fire.*FireCambioBase*("comedor","ventilador","OFF");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a OFF  
 ttsManager.initQueue("Ventilador apagado"); //Ejecucion de voz del cambio de estado  
 mensaje("Ventilador apagado");  
 }  
 }  
 break;  
 }  
 }  
}

**Jardín**

package com.example.domotic\_project;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.speech.RecognizerIntent;  
import android.view.View;  
import android.widget.Switch;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;  
  
import java.util.ArrayList;  
import java.util.Locale;  
  
public class Jardin extends AppCompatActivity {  
 //Inicializacion de variables TextView, Switch, DatabaseRefence, Cambio\_base, TTSManager  
 TextView txtLuces;  
 Switch swtLuces;  
 TextView txtAlarma;  
 Switch swtAlarma;  
 private DatabaseReference housetic;  
 TTSManager ttsManager=null;  
 Cambio\_base fire = new Cambio\_base();  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_jardin*);  
 txtLuces = (TextView) findViewById(R.id.*textView\_Luces4*); //llamado del TextView del activity  
 txtAlarma = (TextView) findViewById(R.id.*textView\_Alarma4*);  
 swtAlarma= (Switch) findViewById(R.id.*switch\_Alarma4*); //llamado del switch del activity  
 swtLuces = (Switch) findViewById(R.id.*switch\_Luces4*);  
 ttsManager=new TTSManager(); //creacion del objeto que ejecutara la voz de la aplicacion  
 ttsManager.init(this);  
  
 String cuarto = "jardin"; //variables que se utilizara para ejecutar en la base de datos  
 final String dispositivo1 = "luces";  
 final String dispositivo2 = "alarma";  
 housetic = FirebaseDatabase.*getInstance*().getReference();//ingreso a la base de datos  
 housetic.child(cuarto).addValueEventListener(new ValueEventListener() { //busqueada por padre e hijo del diapositivo  
 @Override  
  
 public void onDataChange(DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 String resultadoConsultaLuces= dataSnapshot.child(dispositivo1).getValue().toString();  
 if (resultadoConsultaLuces.equalsIgnoreCase("OFF")) {  
 swtLuces.setChecked(false);  
 txtLuces.setText("Desactivado");  
 Toast.*makeText*(Jardin.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 } else {  
 swtLuces.setChecked(true);  
 txtLuces.setText("Activado" );  
 Toast.*makeText*(Jardin.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 }  
  
 }  
  
 }  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 }  
 });  
 housetic.child(cuarto).addValueEventListener(new ValueEventListener() { //busqueada por padre e hijo del diapositivo  
 @Override  
  
 public void onDataChange(DataSnapshot dataSnapshot) {  
 if(dataSnapshot.exists()){  
 String resultadoConsultaAlarma= dataSnapshot.child(dispositivo2).getValue().toString();  
 if (resultadoConsultaAlarma.equalsIgnoreCase("OFF")) {  
 swtAlarma.setChecked(false);  
 txtAlarma.setText("Desactivado");  
 Toast.*makeText*(Jardin.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 } else {  
 swtAlarma.setChecked(true);  
 txtAlarma.setText("Activado");  
 Toast.*makeText*(Jardin.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 }  
  
 }  
  
 }  
 @Override  
 public void onCancelled(@NonNull DatabaseError databaseError) {  
 }  
 });  
 }  
 public void estaciones(View v) { //implementacion de funcion estaciones, redirrecionamiento del cuarto a estaciones  
 Intent i = new Intent(this, Estaciones.class );  
 startActivity(i);  
  
 }  
 public void onclicAlarma(View view){ //implementacion de funcion onClic para la alarma  
  
 if (view.getId() == R.id.*switch\_Alarma4*){  
 if (swtAlarma.isChecked()){  
 txtAlarma.setText("Activado");  
  
 fire.*FireCambioBase*("jardin","alarma","ON"); //implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a ON  
 ttsManager.initQueue("Alarma activada");//Ejecucion de voz del cambio de estado  
 Toast.*makeText*(Jardin.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 }else{  
 txtAlarma.setText("Desactivado");  
 ttsManager.initQueue("Alarma desactivada");//Ejecucion de voz del cambio de estado  
 fire.*FireCambioBase*("jardin","alarma","OFF");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a OFF  
 Toast.*makeText*(Jardin.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 }  
  
 public void onclicLuces(View view){  
  
 if (view.getId() == R.id.*switch\_Luces4*){  
 if (swtLuces.isChecked()){  
 txtLuces.setText("Activado");  
  
 fire.*FireCambioBase*("jardin","luces","ON"); //implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a ON  
 ttsManager.initQueue("Luces encendidas"); //Ejecucion de voz del cambio de estado  
 Toast.*makeText*(Jardin.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 }else{  
 txtLuces.setText("Desactivado");  
  
 fire.*FireCambioBase*("jardin","luces","OFF");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a OFF  
 ttsManager.initQueue("Luces apagadas");//Ejecucion de voz del cambio de estado  
 Toast.*makeText*(Jardin.this,"Conexión Establecida", Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 }  
  
  
 public void getInputSpeech(View view){  
 Intent intent = new Intent(RecognizerIntent.*ACTION\_RECOGNIZE\_SPEECH*);  
 intent.putExtra(RecognizerIntent.*EXTRA\_LANGUAGE\_MODEL*, RecognizerIntent.*LANGUAGE\_MODEL\_FREE\_FORM*);  
 intent.putExtra(RecognizerIntent.*EXTRA\_LANGUAGE*, Locale.*getDefault*());  
 if(intent.resolveActivity(getPackageManager())!= null){  
 startActivityForResult(intent, 10);  
 } else {  
 mensaje("Tu dispositivo no reconoce entrada de voz");  
 }  
 }  
  
 public void mensaje(String text){  
 Toast.*makeText*(Jardin.this,text, Toast.*LENGTH\_LONG*).show();  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data){  
 super.onActivityResult(requestCode,resultCode,data);  
 switch (requestCode){  
 case 10:  
 if(resultCode == *RESULT\_OK* && data != null){  
 ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.*EXTRA\_RESULTS*);  
 String comando = result.get(0);  
 if(comando.equalsIgnoreCase("Encender luces")){//Ejecucion de voz del cambio de estado  
 fire.*FireCambioBase*("jardin","luces","ON");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a ON  
 ttsManager.initQueue("Luces encendidas");// validacion de comando de voz  
 mensaje("Luces encendidas");  
 } else if( comando.equalsIgnoreCase("Apagar luces")){//Ejecucion de voz del cambio de estado  
 fire.*FireCambioBase*("jardin","luces","OFF");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a OFF  
 ttsManager.initQueue("Luces apagadas");// validacion de comando de voz  
 mensaje("Luces apagadas");  
 }else if(comando.equalsIgnoreCase("Activar alarma")){//Ejecucion de voz del cambio de estado  
 fire.*FireCambioBase*("jardin","alarma","ON");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a ON  
 ttsManager.initQueue("Alarma activada");// validacion de comando de voz  
 mensaje("Alarma activada");  
 } else if( comando.equalsIgnoreCase("Desactivar alarma")){//Ejecucion de voz del cambio de estado  
 fire.*FireCambioBase*("jardin","alarma","OFF");//implementacion del FireCambioBase, cambiara el estado del disposito en la base de datos a OFF  
 ttsManager.initQueue("Alarma desactivada");// validacion de comando de voz  
 mensaje("Alarma desactivada");  
 }  
 }  
 break;  
 }  
 }  
}

## ANALISIS DE PRESUPUESTO

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Cantidad** | **Precio unidad** | **Precio total** |
| Relés | 11 | $1,75 | $14 |
| Sensor proximidad | 1 | $2,5 | $2,5 |
| Motor dc | 3 | $1,5 | $4,5 |
| Luces | 10 | $0,15 | $1,5 |
| Servo | 2 | $3 | $6 |
| Pulsador | 1 | $1 | $1 |
| Bocina | 1 | $2 | $2 |
| Arduino mega | 1 | $20 | $20 |
| Fuente 5V | 1 | $9 | $9 |
| Módulo Wifi | 1 | $4,5 | $4,5 |
|  |  | **Total** | **$65** |

## CONCLUSIONES

* El proyecto muestra de manera clara como es la implementación de un sistema de control domótico para un hogar promedio. Se pudo conocer la complejidad de su instalación tanto del hardware como de la creación del software requerido y su interacción con la base de datos remota.
* La conexión tanto del aplicativo móvil como del Arduino para la comunicación con la base de datos es un proceso complejo ya que es muy distinto a utilizar un servidor local.
* El proyecto es factible de realizar ya que resulta relativamente económico en comparación a dispositivos implementados hoy en día.

## REFERENCIAS

* Base De Datos Con Android: <https://codigofacilito.com/articulos/base-de-datos-con-android>
* Exploring Home Automation and Domotics: <https://www.thoughtco.com/what-is-a-smart-house-domotics-177572>
* De MySQL a Arduino, leer datos de la base de datos: <https://www.rinconingenieril.es/de-mysql-a-arduino/>