**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**APLICACIONES MÓVILES Y SERVICIOS TELEMÁTICOS**

Logotipo, Icono

Descripción generada automáticamente

**MANUAL TÉCNICO**

Rastreo de mochila v3.0

**INTEGRANTES:**

Lorena Sthefany Campoverde Vilela

Ramiro Sebastián Serrano Espinoza

Kevin Axell Concha Regatto

**GRUPO:** 5

**TUTORA:**

Msig. Adriana Collaguazo Jaramillo

TABLA DE CONTENIDO

[INTRODUCCIÓN 3](#_Toc62176336)

[DESCRIPCIÓN DEL SISTEMA 3](#_Toc62176337)

[RECURSOS 4](#_Toc62176338)

[DIAGRAMAS 4](#_Toc62176339)

[DIAGRAMA DEL PROYECTO 4](#_Toc62176340)

[DIAGRAMA DE ÁRBOL 5](#_Toc62176341)

[DIAGRAMA DE CLASES 5](#_Toc62176342)

[DIAGRAMA DE CASOS DE USO 6](#_Toc62176343)

[DIAGRAMA DE DESPLIEGUE 6](#_Toc62176344)

[DIAGRAMA DEL CIRCUITO 7](#_Toc62176345)

[DIAGRAMA ESQUEMÁTICO DEL CIRCUITO 7](#_Toc62176346)

[PROGRAMACIÓN 8](#_Toc62176347)

[CÓDIGO ARDUINO 8](#_Toc62176348)

[CÓDIGO JAVA 10](#_Toc62176349)

[Splash.java 10](#_Toc62176350)

[MainActivity.java 12](#_Toc62176351)

[Registro.java 16](#_Toc62176352)

[Home.java 19](#_Toc62176353)

[ModoEstatico.java 22](#_Toc62176354)

[ServicioModoEstatico.java 28](#_Toc62176355)

[Reminder.java 30](#_Toc62176356)

[ModoLive.java 31](#_Toc62176357)

[RegistroBaterias.java 35](#_Toc62176358)

[MyFirebaseInstanceService.java 38](#_Toc62176359)

[LAYOUTS XML 39](#_Toc62176360)

[Activity\_splah.xml 39](#_Toc62176361)

[Activity\_main.xml 39](#_Toc62176362)

[Activity\_registro.xml 41](#_Toc62176363)

[Activity\_home.xml 44](#_Toc62176364)

[Dialog\_telefono.xml 45](#_Toc62176365)

[Activity\_modo\_estatico.xml 46](#_Toc62176366)

[Activity\_modo\_live.xml 48](#_Toc62176367)

[Activity\_registro\_baterias.xml 49](#_Toc62176368)

# **INTRODUCCIÓN**

El presente manual técnico está enfocado en describir los recursos de hardware y software utilizados en la solución del proyecto.

Este proyecto tiene como objetivo ayudar a monitorear y rastrear las distintas mochilas con las que cuente un usuario, a través de un GPS, obteniendo alertas cada cierto tiempo o la ubicación en tiempo real según el modo elegido.

# **DESCRIPCIÓN DEL SISTEMA**

El sistema se basa en el uso de una aplicación móvil vinculada a una base de datos, la cual recibirá la información del dispositivo IoT a través de callbacks generados en Sigfox cada periodo de tiempo.

Esta app está dirigida para estudiantes de la Escuela Superior Politécnica del Litoral, los cuales suelen tener problemas de robos u olvidos de sus mochilas y por ende, pérdidas de pertenencias importantes.

Ante dicha problemática, el usuario será capaz de ubicar su mochila en un mapa en tiempo real y realizar distintas configuraciones que lo ayuden a encontrarla rápidamente en caso de un problema. Además, no será necesario mantener la aplicación abierta para monitorear los dispositivos gracias a que cuenta con un modo estático que alerta por medio de notificaciones o mensajes SMS que existe un movimiento de la dicho objeto.

# **RECURSOS**

Tabla 1. Recursos del proyecto

|  |  |  |
| --- | --- | --- |
| Hardware | Software | Extras |
| Arduino MEGA | Android Studio | Cautín |
| Módulo NODEMCU  ESP8266 MOD V3 | Sigfox | Jumpers |
| Relay 5V 1 Ch | Firebase | Estaño |
| Sensor de Voltaje | Arduino | Cable Arduino |
| Regulador LM7805 |  |  |
| iButton GPS |  |  |
| Baterias AA+ |  |  |
| Switch de 3 estados |  |  |
| Power Bank 5V |  |  |

# **DIAGRAMAS**

## **DIAGRAMA DEL PROYECTO**

Diagrama

Descripción generada automáticamente

Figura 1. Diagrama 1

## **DIAGRAMA DE ÁRBOL**

Diagrama

Descripción generada automáticamente

Figura 2. Diagrama de Árbol

## **DIAGRAMA DE CLASES**

Diagrama

Descripción generada automáticamente

Figura 3. Diagrama de clases

## **DIAGRAMA DE CASOS DE USO**

Gráfico, Gráfico de burbujas

Descripción generada automáticamente

Figura 4. Diagrama de casos de uso

## **DIAGRAMA DE DESPLIEGUE**

Diagrama

Descripción generada automáticamente

Figura 5. Diagrama de despliegue

## **DIAGRAMA DEL CIRCUITO**

Diagrama

Descripción generada automáticamente

Figura 6. Diagrama del circuito

## **DIAGRAMA ESQUEMÁTICO DEL CIRCUITO**

Diagrama, Esquemático

Descripción generada automáticamente

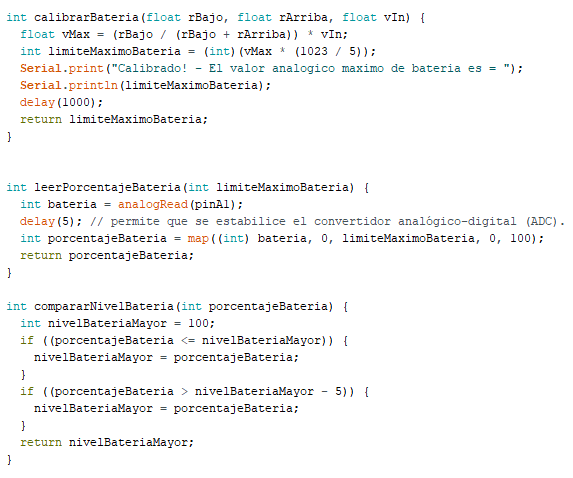
Figura 7. Diagrama esquemático del circuito

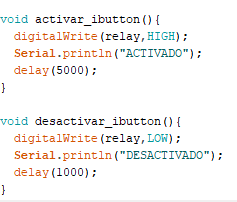
# **PROGRAMACIÓN**

## **CÓDIGO ARDUINO**









## **CÓDIGO JAVA**

### ***Splash.java***

package com.example.finder;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.content.pm.ActivityInfo;  
import android.os.Bundle;  
import android.os.Handler;  
import android.view.WindowManager;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
  
import java.util.HashMap;  
  
public class Splash extends AppCompatActivity {  
  
 private final int DURACION\_SPLASH = 3000;  
  
 @Override  
 public void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setRequestedOrientation(ActivityInfo.*SCREEN\_ORIENTATION\_PORTRAIT*);  
 this.getWindow().setFlags(WindowManager.LayoutParams.*FLAG\_FULLSCREEN*, WindowManager.LayoutParams.*FLAG\_FULLSCREEN*);  
  
 setContentView(R.layout.*activity\_splash*);  
  
 *// se presenta el splash por 3 segundos* new Handler().postDelayed(new Runnable(){  
 public void run(){  
  
 DatabaseReference db\_reference = FirebaseDatabase.*getInstance*().getReference();  
 FirebaseAuth mAuth = FirebaseAuth.*getInstance*();  
 FirebaseUser user = mAuth.getCurrentUser();  
 if (user != null) {  
  
 HashMap<String, String> map = new HashMap<>();  
 map.put("name", user.getDisplayName());  
 map.put("email", user.getEmail());  
 map.put("telefono",Home.*telefono*);  
  
 String id = mAuth.getCurrentUser().getUid();  
  
 db\_reference.child("Usuario").child(id).setValue(map).addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task2) {  
 if (task2.isSuccessful()) {  
   
 Intent intent =new Intent(Splash.this,Home.class);  
 intent.putExtra("map",map);  
 startActivity(intent);  
  
 finish();  
 } else {  
 Toast.*makeText*(Splash.this, "No se pudieron crear los datos correctamente.", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 }  
 });  
  
 } else {  
 Intent intent = new Intent(Splash.this, MainActivity.class);  
 startActivity(intent);  
 finish();  
 }  
 };  
 }, DURACION\_SPLASH);  
 }  
}

### ***MainActivity.java***

package com.example.finder;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.content.Context;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.graphics.Color;  
import android.graphics.drawable.ColorDrawable;  
import android.net.ConnectivityManager;  
import android.net.NetworkInfo;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.util.Log;  
import android.util.Patterns;  
import android.view.View;  
import android.view.WindowManager;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.google.android.gms.auth.api.signin.GoogleSignIn;  
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;  
import com.google.android.gms.auth.api.signin.GoogleSignInClient;  
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;  
import com.google.android.gms.common.api.ApiException;  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.AuthCredential;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.auth.GoogleAuthProvider;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
  
  
import java.util.HashMap;  
import java.util.Map;  
  
public class MainActivity extends AppCompatActivity {  
 *//Varibales públicas* static final int *GOOGLE\_SIGN\_IN* = 123;  
 FirebaseAuth mAuth;  
 GoogleSignInClient mGoogleSignInClient;  
 private Button btn\_login;  
 private ProgressDialog progressDialog;  
 DatabaseReference db\_reference;  
  
 private EditText EditT\_correo,EditT\_contrasena;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 mAuth = FirebaseAuth.*getInstance*();  
 db\_reference = FirebaseDatabase.*getInstance*().getReference();  
  
 GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.*DEFAULT\_SIGN\_IN*)  
 .requestIdToken(getString(R.string.*default\_web\_client\_id*))  
 .requestEmail()  
 .build();  
 mGoogleSignInClient = GoogleSignIn.*getClient*(this, gso);  
 Intent intent = getIntent();  
 String msg = intent.getStringExtra("msg");  
 if(msg != null){  
 if(msg.equals("cerrarSesion")){  
 cerrarSesion();  
 }  
 }  
  
 EditT\_correo= (EditText) findViewById(R.id.*Edit\_txtCorreo*);  
 EditT\_contrasena= (EditText) findViewById(R.id.*Edit\_txtContrasena*);  
 btn\_login= findViewById(R.id.*btn\_iniciarSesion*);  
 progressDialog=new ProgressDialog(this);  
 */\*FirebaseUser firebaseUser = mAuth.getCurrentUser();  
 if (firebaseUser != null) {  
 updateUI(firebaseUser);  
 }\*/* }  
  
 *// se valida que exista conexion a internet para iniciar sesion a traves de correo y contrasena o sino mostrar una alerta* public void Login(View view) {  
 ConnectivityManager manager = (ConnectivityManager) getApplicationContext().getSystemService(Context.*CONNECTIVITY\_SERVICE*);  
 NetworkInfo activeNetwork = manager.getActiveNetworkInfo();  
  
 if (null == activeNetwork || !activeNetwork.isConnected()|| !activeNetwork.isAvailable()) {  
 AlertaInternet();  
 }  
 else{  
 login\_user();  
 }  
 }  
  
 *// se valida que exista conexion a internet para iniciar sesion a traves de google o sino mostrar una alerta* public void iniciarSesion(View view) {  
 ConnectivityManager manager = (ConnectivityManager) getApplicationContext().getSystemService(Context.*CONNECTIVITY\_SERVICE*);  
 NetworkInfo activeNetwork = manager.getActiveNetworkInfo();  
  
 if (null == activeNetwork || !activeNetwork.isConnected()|| !activeNetwork.isAvailable()) {  
 AlertaInternet();  
 }  
 else{  
 Intent signInIntent = mGoogleSignInClient.getSignInIntent();  
 startActivityForResult(signInIntent, *GOOGLE\_SIGN\_IN*);  
 }  
 }  
  
 *// metodo para cerrar sesion de google* private void cerrarSesion() {  
 mGoogleSignInClient.signOut().addOnCompleteListener(this, task -> updateUI(null));  
 }  
  
 *//se valida que el usuario ingrese correctamente los datos y se verifica con la base de datos* private void login\_user() {  
  
 String email=EditT\_correo.getText().toString();  
 String password=EditT\_contrasena.getText().toString();  
  
 if(TextUtils.*isEmpty*(email)){  
 EditT\_correo.setError("Campo vacío.");  
 return;  
 }  
 else if(TextUtils.*isEmpty*(password)){  
 EditT\_contrasena.setError("Campo vacío.");  
 return;  
 }  
  
 progressDialog.setMessage("Validando...");  
 progressDialog.show();  
 progressDialog.setCanceledOnTouchOutside(false);  
 mAuth.signInWithEmailAndPassword(email,password).addOnCompleteListener(MainActivity.this, new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if(task.isSuccessful()){  
 Toast.*makeText*(MainActivity.this,"Ingreso Exitoso.",Toast.*LENGTH\_LONG*).show();  
 Intent intent= new Intent(MainActivity.this,Home.class);  
 startActivity(intent);  
 finish();  
 }  
 else{  
 Toast.*makeText*(MainActivity.this,"Usuario no registrado.",Toast.*LENGTH\_LONG*).show();  
 }  
 progressDialog.dismiss();  
 }  
 });  
  
 }  
  
 @Override  
 protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
 super.onActivityResult(requestCode, resultCode, data);  
 if (requestCode == *GOOGLE\_SIGN\_IN*) {  
 Task<GoogleSignInAccount> task = GoogleSignIn.*getSignedInAccountFromIntent*(data);  
 try {  
 GoogleSignInAccount account = task.getResult(ApiException.class);  
 if (account != null) firebaseAuthWithGoogle(account);  
 } catch (ApiException e) {  
 Log.*w*("TAG", "Fallo el inicio de sesión con google.", e);  
 }  
 }  
 }  
  
 *// Autenticacion a traves de google* private void firebaseAuthWithGoogle(GoogleSignInAccount acct) {  
 Log.*d*("TAG", "firebaseAuthWithGoogle:" + acct.getId());  
 AuthCredential credential = GoogleAuthProvider.*getCredential*(acct.getIdToken(),  
 null);  
 mAuth.signInWithCredential(credential)  
 .addOnCompleteListener(this, task -> {  
 if (task.isSuccessful()) {  
 FirebaseUser user = mAuth.getCurrentUser();  
 updateUI(user);  
 } else {  
 System.*out*.println("error");  
 updateUI(null);  
 }  
 });  
 }  
  
 *// se obtiene informacion del usuario en un HashMap* private void updateUI(FirebaseUser user) {  
 if (user != null) {  
  
 HashMap<String, String> map = new HashMap<>();  
 map.put("name", user.getDisplayName());  
 map.put("email", user.getEmail());  
 map.put("telefono",Home.*telefono*);  
  
 String id = mAuth.getCurrentUser().getUid();  
  
 db\_reference.child("Usuario").child(id).setValue(map).addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task2) {  
 if (task2.isSuccessful()) {  
 Toast.*makeText*(MainActivity.this, "Ingreso exitoso", Toast.*LENGTH\_SHORT*).show();  
 Intent intent =new Intent(MainActivity.this,Home.class);  
 intent.putExtra("map",map);  
 startActivity(intent);  
  
 finish();  
 } else {  
 Toast.*makeText*(MainActivity.this, "No se pudieron crear los datos correctamente.", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 }  
 });  
  
 } else {  
 System.*out*.println("Sin registro.");  
 }  
 }  
  
 *// metodo que permite ir a la pantalla de registro* public void registro (View view){  
 startActivity(new Intent(MainActivity.this,Registro.class));  
 finish();  
 }  
  
 *// se crea una alerta en caso de que el usuario no cuente con internet* private void AlertaInternet(){  
 AlertDialog.Builder builder= new AlertDialog.Builder(MainActivity.this);  
 builder.setTitle("ALERTA")  
 .setMessage("Error de Conexion.")  
 .setIcon(R.drawable.*warning*)  
 .setPositiveButton("Reintentar", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 recreate();  
 }  
  
 })  
 .setCancelable(false);  
  
 AlertDialog dialog= builder.create();  
 dialog.show();  
 }  
}

### ***Registro.java***

package com.example.finder;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.content.Context;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.net.ConnectivityManager;  
import android.net.NetworkInfo;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.util.Patterns;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.Task;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
  
import java.util.HashMap;  
import java.util.Map;  
  
public class Registro extends AppCompatActivity {  
  
 private EditText emailEt,passwordEt1,passwordEt2, telefonoEt;  
 private ProgressDialog progressDialog;  
 private FirebaseAuth firebaseAuth;  
 private Button btn\_registro;  
  
 String email="";  
 String password1="";  
 String password2="";  
 String telefono="";  
  
 DatabaseReference db\_reference;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_registro*);  
  
 firebaseAuth = FirebaseAuth.*getInstance*();  
 db\_reference = FirebaseDatabase.*getInstance*().getReference();  
  
 emailEt=findViewById(R.id.*et\_correo*);  
 passwordEt1=findViewById(R.id.*ed\_password1*);  
 passwordEt2=findViewById(R.id.*ed\_password2*);  
 telefonoEt=findViewById(R.id.*ed\_celular*);  
  
 progressDialog=new ProgressDialog(this);  
  
 btn\_registro=findViewById(R.id.*btn\_registro*);  
  
 }  
  
 *// se valida que exista conexion a internet para registrar al usuario o sino mostrar una alerta* public void Registro(View view) {  
 ConnectivityManager manager = (ConnectivityManager) getApplicationContext().getSystemService(Context.*CONNECTIVITY\_SERVICE*);  
 NetworkInfo activeNetwork = manager.getActiveNetworkInfo();  
  
 if (null == activeNetwork || !activeNetwork.isConnected()|| !activeNetwork.isAvailable()) {  
 AlertaInternet();  
 }  
 else{  
 RegistroUser();  
 }  
 }  
  
 *// se valida que el usuario ingrese correctamente los datos y se envia a la base de datos* private void RegistroUser(){  
 email=emailEt.getText().toString();  
 password1=passwordEt1.getText().toString();  
 password2=passwordEt2.getText().toString();  
 telefono=telefonoEt.getText().toString();  
  
 if(TextUtils.*isEmpty*(email)){  
 emailEt.setError("Campo vacío.");  
 return;  
 }  
 else if(TextUtils.*isEmpty*(password1)){  
 passwordEt1.setError("Campo vacío.");  
 return;  
 }  
 else if(TextUtils.*isEmpty*(password2)){  
 passwordEt2.setError("Campo vacío.");  
 return;  
 }  
 else if(TextUtils.*isEmpty*(telefono)){  
 telefonoEt.setError("Campo vacío.");  
 return;  
 }  
 else if(!password1.equals(password2)){  
 passwordEt2.setError("Las contraseñas no coinciden.");  
 return;  
 }  
 else if(password1.length()<6){  
 passwordEt1.setError("La contraseña debe ser mayor a 6 caracteres.");  
 return;  
 }  
 else if(telefono.length()!=10){  
 telefonoEt.setError("El teléfono debe tener 10 dígitos.");  
 return;  
 }  
 else if(!isValidEmail(email)){  
 emailEt.setError("Email inválido.");  
 return;  
 }  
  
 progressDialog.setMessage("Cargando...");  
 progressDialog.show();  
 progressDialog.setCanceledOnTouchOutside(false);  
 firebaseAuth.createUserWithEmailAndPassword(email, password1).addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if(task.isSuccessful()){  
  
 Map<String, String> map= new HashMap<>();  
 map.put("email",email);  
 map.put("password",password1);  
 map.put("telefono",telefono);  
  
  
 String id = firebaseAuth.getCurrentUser().getUid();  
 db\_reference.child("Usuario").child(id).setValue(map).addOnCompleteListener(new OnCompleteListener<Void>() {  
 @Override  
 public void onComplete(@NonNull Task<Void> task2) {  
 if (task2.isSuccessful()) {  
 Toast.*makeText*(Registro.this, "Registro Exitoso.", Toast.*LENGTH\_LONG*).show();  
 Intent intent = new Intent(Registro.this, Home.class);  
 startActivity(intent);  
 finish();  
 } else {  
 Toast.*makeText*(Registro.this, "Error al registrar datos.", Toast.*LENGTH\_LONG*).show();  
  
 }  
 }  
 });  
 }  
 else{  
 Toast.*makeText*(Registro.this,"Falla al registrar.",Toast.*LENGTH\_LONG*).show();  
 }  
 progressDialog.dismiss();  
 }  
 });  
  
 }  
  
 *// se valida que el correo tenga la estructura correcta* private Boolean isValidEmail(CharSequence target){  
 return (!TextUtils.*isEmpty*(target)&& Patterns.*EMAIL\_ADDRESS*.matcher(target).matches());  
 }  
  
 *// se crea una alerta en caso de que el usuario no cuente con internet* private void AlertaInternet(){  
 AlertDialog.Builder builder= new AlertDialog.Builder(Registro.this);  
 builder.setTitle("ALERTA")  
 .setMessage("Error de Conexión.")  
 .setIcon(R.drawable.*warning*)  
 .setPositiveButton("Reintentar", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 recreate();  
 }  
  
 })  
 .setCancelable(false);  
  
 AlertDialog dialog= builder.create();  
 dialog.show();  
 }

}

### ***Home.java***

package com.example.finder;  
  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.os.Bundle;  
import android.text.Html;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Spinner;  
import android.widget.Toast;  
import com.google.firebase.auth.FirebaseAuth;  
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.HashMap;  
import java.util.Map;  
  
public class Home extends AppCompatActivity implements AdapterView.OnItemSelectedListener{  
  
 DatabaseReference db\_reference;  
 FirebaseAuth mAuth;  
 EditText tlf;  
  
 public static String *telefono*;  
  
 private String dispositivos[] = {" ","Dispositivo 1", "Dispositivo 2", "Dispositivo 3", "Dispositivo 4", "Dispositivo 5", "Dispositivo 6"};  
 private String bateria [] = {"34%", "73%", "19%", "86%", "25%", "20%"};  
  
 private Button btn\_Modo;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_home*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
  
 Spinner Valores= findViewById(R.id.*spinner*);  
 Valores.setOnItemSelectedListener(this);  
  
 ArrayAdapter items= new ArrayAdapter(this, android.R.layout.*simple\_spinner\_item*,dispositivos);  
 items.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);  
 Valores.setAdapter(items);  
  
 btn\_Modo= (Button)findViewById(R.id.*btn\_modo*);  
  
 iniciarBaseDeDatos();  
 leerTelefono();  
  
 if(*telefono*==null) {  
 obtenerTelefono();  
 }  
  
 }  
  
 *// se inicia la base de datos* public void iniciarBaseDeDatos(){  
 db\_reference = FirebaseDatabase.*getInstance*().getReference();  
 }  
  
 *// al seleccionar un dispositivo del spinner se presenta un toast con su porcentaje de bateria* @Override  
 public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {  
 if (i>0) {  
 Toast.*makeText*(this,"La batería del " + adapterView.getSelectedItem().toString() + " es " + bateria[i-1],Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 @Override  
 public void onNothingSelected(AdapterView<?> adapterView) {  
  
 }  
  
 *// alerta que permite elegir entre modo estatico y modo live, y dirigirse a ellas* public void modo(View view) {  
 final CharSequence[] items= {"Modo Estático","Modo Live"};  
 AlertDialog.Builder alertaNotificaciones= new AlertDialog.Builder(this);  
 alertaNotificaciones.setTitle(Html.*fromHtml*("<font color= '#00BCD4'>"+"Elija una opción:"+"</font>"));  
  
 alertaNotificaciones.setItems(items, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 Toast mensaje= Toast.*makeText*(getApplicationContext(),"Opción seleccionada: "+ items[i],Toast.*LENGTH\_LONG*);  
 mensaje.show();  
  
 if (i==0){  
 startActivity(new Intent(getApplicationContext(), ModoEstatico.class));  
 }  
 else {  
 startActivity(new Intent(getApplicationContext(), ModoLive.class));  
 }  
  
 }  
 });  
 AlertDialog alertaModo = alertaNotificaciones.create();  
 alertaModo.show();  
 }  
  
 *// metodo que permite cerrar sesion de la aplicacion* public void cerrarSesion(View view){  
 FirebaseAuth.*getInstance*().signOut();  
 finish();  
 Intent intent = new Intent(this, MainActivity.class);  
 Toast.*makeText*(this,"Sesión Finalizada.", Toast.*LENGTH\_LONG*).show();  
 intent.putExtra("msg", "cerrarSesion");  
 startActivity(intent);  
 }  
  
 *// en caso de no contar con un telefono registrado en gmail, se presenta una alerta que pide el numero* private void obtenerTelefono(){  
 AlertDialog.Builder builder= new AlertDialog.Builder(Home.this);  
 LayoutInflater inflater= this.getLayoutInflater();  
 builder.setTitle("DATO NO REGISTRADO").setIcon(R.drawable.*tlf*);  
  
 View dialogView= inflater.inflate(R.layout.*dialog\_telefono*,null);  
 builder.setView(dialogView);  
  
 tlf= dialogView.findViewById(R.id.*edit\_tlf*);  
  
 builder.setPositiveButton("REGISTRAR", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 ingresoTelefono(tlf.getText().toString());  
 Toast.*makeText*(getApplicationContext(), "Registro exitoso.", Toast.*LENGTH\_LONG*).show();  
 }  
 })  
 .setCancelable(false);  
  
 builder.show();  
 }  
  
 *// en caso de no contar con un telefono registrado en gmail, se lo agrega en la base de datos* public void ingresoTelefono(String telefono){  
 Map<String, String> nuevoDato = new HashMap<String, String>();  
 nuevoDato.put("telefono", telefono);  
 DatabaseReference baseDatos = db\_reference.child("Usuario");  
 baseDatos.child(mAuth.getCurrentUser().getUid()).child("telefono").setValue(telefono);  
 }  
  
 *// metodo para ir a la pantalla del registro de baterias* public void registroBaterias(View view){  
 Intent intent = new Intent(this, RegistroBaterias.class);  
 startActivity(intent);  
 }  
  
 *// se obtiene el telefono de usuario desde la base de datos* public void leerTelefono(){  
 db\_reference.child("Usuario").child(mAuth.getCurrentUser().getUid()).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(DataSnapshot dataSnapshot) {  
 String numero = String.*valueOf*(dataSnapshot.child("telefono").getValue());  
 *telefono*=numero;  
 }  
  
 @Override  
 public void onCancelled(DatabaseError error) {  
 System.*out*.println(error.toException());  
 }  
 });  
  
 }  
}

### ***ModoEstatico.java***

package com.example.finder;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import androidx.core.app.NotificationCompat;  
import androidx.core.app.NotificationManagerCompat;  
  
import android.Manifest;  
import android.app.AlarmManager;  
import android.app.AlertDialog;  
import android.app.NotificationChannel;  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.content.Context;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.os.AsyncTask;  
import android.os.Build;  
import android.os.Bundle;  
import android.os.Handler;  
import android.telephony.SmsManager;  
import android.text.Html;  
import android.view.View;  
import android.widget.CompoundButton;  
import android.widget.NumberPicker;  
import android.widget.Switch;  
import android.widget.Toast;  
  
import com.google.firebase.auth.FirebaseAuth;  
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 ModoEstatico extends AppCompatActivity {  
  
 FirebaseAuth mAuth;  
 DatabaseReference db\_reference;  
  
 private int tiempo=0;  
 Intent intent;  
 public static Switch *switchSMS*;  
 public static time *time*;  
 public static tiempoSMS *tiempoSMS*;  
 public static Boolean *enviando*=false;  
  
 String telefono,ultimaLat,ultimaLon,longitud,latitud;  
 String dispElegido="5";  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_modo\_estatico*);  
  
 mAuth = FirebaseAuth.*getInstance*();  
  
 iniciarBaseDeDatos();  
 leerTelefono();  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*P*) {  
 ActivityCompat.*requestPermissions*(this,new String[]{Manifest.permission.*FOREGROUND\_SERVICE*}, PackageManager.*PERMISSION\_GRANTED*);  
 }  
  
 createNotificationChannel();  
  
 *switchSMS* = findViewById(R.id.*switchSMS*);  
  
 intent = new Intent(this, ServicioModoEstatico.class);  
  
 *switchSMS*.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
 @Override  
 public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {  
 if (isChecked) {  
  
 *time*.cancel(true);  
  
 if(ActivityCompat.*checkSelfPermission*(  
 ModoEstatico.this, Manifest.permission.*SEND\_SMS*)  
 != PackageManager.*PERMISSION\_GRANTED*&& ActivityCompat.*checkSelfPermission*(  
 ModoEstatico.this,Manifest  
 .permission.*SEND\_SMS*)!= PackageManager.*PERMISSION\_GRANTED*){  
 ActivityCompat.*requestPermissions*(ModoEstatico.this,new String[]  
 { Manifest.permission.*SEND\_SMS*,},1000);  
 }else{ };  
 *tiempoSMS* = new tiempoSMS();  
 *tiempoSMS*.execute();  
 } else {  
 *tiempoSMS*.cancel(true);  
 } } } );  
 actulizarDatos();  
 }  
  
 *// se inicia base de datos* public void iniciarBaseDeDatos(){  
 db\_reference = FirebaseDatabase.*getInstance*().getReference();  
 }  
  
 public void hilo(){  
 try{  
 Thread.*sleep*(1000);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
  
 public void ejecutar(){  
 *time*= new time();  
 *time*.execute();  
 }  
  
 *// clase para verificar cada cierto tiempo la ubicacion y enviar una notificacion* public class time extends AsyncTask<Void,Integer,Boolean>{  
  
 @Override  
 protected Boolean doInBackground(Void... voids) {  
 for (int i=1; i<=10; i++){  
 hilo();  
 }  
 return true;  
 }  
  
 @Override  
 protected void onPostExecute(Boolean eBoolean){  
 ejecutar();  
 verificarMovNotif(longitud,ultimaLon,latitud,ultimaLat);  
 ultimaLon=longitud;  
 ultimaLat=latitud;  
 }  
 }  
  
 *// se comprueba si los datos de longitud y latitud son los mismos a los valores anteriores y segun eso enviar notificacion* private void verificarMovNotif(String datoLongitud, String ultimaLongitud, String datoLatitud, String ultimaLatitud){  
 if(datoLongitud!=ultimaLongitud || datoLatitud!=ultimaLatitud){  
 addNotification();  
 Toast.*makeText*(getApplicationContext(),"Notificación Recibida.",Toast.*LENGTH\_LONG*).show();  
 }  
 }  
  
 *// se crea la notificacion y su respectivo canal* private void addNotification() {  
 if(Build.VERSION.*SDK\_INT*>= Build.VERSION\_CODES.*O*){  
 NotificationChannel notificationChannel= new NotificationChannel("com.example.finder","com.example.finder",NotificationManager.*IMPORTANCE\_DEFAULT*);  
 NotificationManager manager=getSystemService(NotificationManager.class);  
 manager.createNotificationChannel(notificationChannel);  
 }  
  
 NotificationCompat.Builder builder= new NotificationCompat.Builder(this,"com.example.finder")  
 .setSmallIcon(R.drawable.*spot*)  
 .setContentTitle("FINDER ALERT")  
 .setContentText("Alerta de movimiento en dispositivo IoT.")  
 .setAutoCancel(true);  
 NotificationManagerCompat managerCompat = NotificationManagerCompat.*from*(this);  
 managerCompat.notify(998,builder.build());  
  
 }  
  
 *// metodo para regresar al menu principal* public void volverMenu(View view) {  
 startActivity(new Intent(getApplicationContext(), Home.class));  
 }  
  
 *// metodo para ir al modo live incluso antes de terminar el modo estatico* public void cambiarModo(View view) {  
 if(ServicioModoEstatico.*activado*){  
 cancelAlarm();  
 }  
 startActivity(new Intent(getApplicationContext(), ModoLive.class));  
 finish();  
 }  
  
 *// metodo para mostrar el number picker* public void aggTiempo(View view){  
 numberPickerDialog();  
 }  
  
 *// se crea el number picker y se agregan funcionalidades a los botones ok y cancel. Ademas, se inicia el servicio de modo estatico* private void numberPickerDialog(){  
 NumberPicker myNumberPicker= new NumberPicker(this);  
 myNumberPicker.setMaxValue(60);  
 myNumberPicker.setMinValue(0);  
 NumberPicker.OnValueChangeListener myValChangedListener= new NumberPicker.OnValueChangeListener() {  
 @Override  
 public void onValueChange(NumberPicker numberPicker, int i, int i1) {  
 tiempo=i1;  
 }  
 };  
 myNumberPicker.setOnValueChangedListener(myValChangedListener);  
 AlertDialog.Builder alertaTiempo= new AlertDialog.Builder(this).setView(myNumberPicker);  
 alertaTiempo.setTitle(Html.*fromHtml*("<font color= '#00BCD4'>"+"Seleccione la duración en minutos:"+"</font>"));  
 alertaTiempo.setPositiveButton(android.R.string.*ok*, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
  
 if(Build.VERSION.*SDK\_INT*>= Build.VERSION\_CODES.*O*){  
 startForegroundService(intent);  
 }else {  
 startService(intent);  
 }  
 Reminder.*activo*=true;  
  
 *time*= new time();  
 *time*.execute();  
  
 startAlarm(tiempo);  
 }  
 });  
 alertaTiempo.setNegativeButton(android.R.string.*cancel*, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 Toast.*makeText*(getApplicationContext(),"Operación cancelada.",Toast.*LENGTH\_LONG*).show();  
  
 }  
 });  
 alertaTiempo.show();  
 }  
  
 *// se inicia la alarma segun el tiempo elegido en el number picker* private void startAlarm(int tiempo2) {  
 AlarmManager alarmManager = (AlarmManager) getSystemService(Context.*ALARM\_SERVICE*);  
 Intent intent = new Intent(this, Reminder.class);  
 PendingIntent pendingIntent = PendingIntent.*getBroadcast*(this, 1, intent, 0);  
 long time1= System.*currentTimeMillis*();  
 long minutos= tiempo2\*60;  
 long segundos= 1000\*minutos;  
 alarmManager.set(AlarmManager.*RTC\_WAKEUP*, time1+segundos, pendingIntent);  
 Toast.*makeText*(getApplicationContext(),"Duración máxima de "+tiempo+" minutos.",Toast.*LENGTH\_LONG*).show();  
 }  
  
 *// se cancela la alarma y las tareas en caso de querer terminarlas antes* private void cancelAlarm() {  
 AlarmManager alarmManager = (AlarmManager) getSystemService(Context.*ALARM\_SERVICE*);  
 Intent intent = new Intent(this, ServicioModoEstatico.class);  
 PendingIntent pendingIntent = PendingIntent.*getBroadcast*(this, 1, intent, 0);  
 alarmManager.cancel(pendingIntent);  
 stopService(this.intent);  
 *//Reminder.activo=false;  
 time*.cancel(true);  
 if(*enviando*){  
 *tiempoSMS*.cancel(true);  
 *switchSMS*.setChecked(false);  
 }  
 Toast.*makeText*(this,"Modo estático detenido.",Toast.*LENGTH\_LONG*).show();  
 }  
  
 *// se crea el canal de notificacion de la notificacion que indica que se termino el modo estatico* private void createNotificationChannel(){  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {  
 NotificationChannel notificationChannel = new NotificationChannel("com.example.Finder1", "com.example.Finder1", NotificationManager.*IMPORTANCE\_DEFAULT*);  
 NotificationManager manager = getSystemService(NotificationManager.class);  
 manager.createNotificationChannel(notificationChannel);  
 }  
 }  
  
 public void ejecutar1(){  
 *tiempoSMS* = new tiempoSMS();  
 *tiempoSMS*.execute();  
 }  
  
 *//clase para verificar cada cierto tiempo la ubicacion y enviar un mensaje* public class tiempoSMS extends AsyncTask<Void,Integer,Boolean> {  
  
 @Override  
 protected Boolean doInBackground(Void... voids) {  
 for (int i=1; i<=10; i++){  
 hilo();  
 }  
 return true;  
 }  
  
 @Override  
 protected void onPostExecute(Boolean eBoolean){  
 ejecutar1();  
 verificarMovSMS(longitud,ultimaLon,latitud,ultimaLat,telefono);  
 ultimaLon=longitud;  
 ultimaLat=latitud;  
 *enviando*=true;  
 }  
 }  
  
 *// se comprueba si los datos de longitud y latitud son los mismos a los valores anteriores y segun eso enviar mensajes SMS* private void verificarMovSMS(String datoLongitud,String ultimaLongitud,String datoLatitud,String ultimaLatitud,String celular){  
 if(datoLongitud!=ultimaLongitud || datoLatitud!=ultimaLatitud){  
 EnviarSMS(celular);  
 Toast.*makeText*(getApplicationContext(),"Mensaje Enviado.",Toast.*LENGTH\_LONG*).show();  
 }  
 }  
  
 *// metodo que envia mensaje al numero obtenido de la base de datos* public void EnviarSMS(String numero) {  
 try{  
 String mensaje = "Alerta de movimiento en dispositivo IoT.";  
 SmsManager smsManager = SmsManager.*getDefault*();  
 smsManager.sendTextMessage(numero, null, mensaje, null, null);  
 }catch (Exception e){  
 Toast.*makeText*(getApplicationContext(), "Error al enviar mensaje.", Toast.*LENGTH\_LONG*).show();  
 e.printStackTrace();  
 }  
 }  
  
 *// metodo que obtiene el telefono del user de la base de datos* public void leerTelefono(){  
 db\_reference.child("Usuario").child(mAuth.getCurrentUser().getUid()).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(DataSnapshot dataSnapshot) {  
 String numero = String.*valueOf*(dataSnapshot.child("telefono").getValue());  
 telefono =numero;  
 }  
  
 @Override  
 public void onCancelled(DatabaseError error) {  
 System.*out*.println(error.toException());  
 }  
 });  
  
 }  
  
 *// hilo que se ejecuta cada 3 segundos obteniendo lectura de datos del dispositivo* public void actulizarDatos(){  
 final Handler handler = new Handler();  
 final Runnable runnable = new Runnable() {  
 @Override  
 public void run() {  
 leerDispositivo();  
 } };  
 handler.postDelayed(runnable, 3000);  
 }  
  
 *// metodo que permite entrar a la base de datos y obtener latitud y longitud del dispositivo seleccionado* public void leerDispositivo(){  
 db\_reference.child("Dispositivos").child(dispElegido).addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(DataSnapshot dataSnapshot) {  
 String longitudObt = String.*valueOf*(dataSnapshot.child("longitud").getValue());  
 String latitudObt = String.*valueOf*(dataSnapshot.child("latitud").getValue());  
  
 longitud=longitudObt;  
 latitud=latitudObt;  
 }  
 @Override  
 public void onCancelled(DatabaseError error) {  
 System.*out*.println(error.toException());  
 }  
 });  
 }  
  
}

### ***ServicioModoEstatico.java***

package com.example.finder;  
  
import android.app.Notification;  
import android.app.NotificationChannel;  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.app.Service;  
import android.content.Intent;  
import android.os.Build;  
import android.os.IBinder;  
import android.widget.Toast;  
  
import androidx.annotation.Nullable;  
import androidx.core.app.NotificationCompat;  
  
import java.util.Timer;  
import java.util.TimerTask;  
  
public class ServicioModoEstatico extends Service {  
  
 public static Boolean *activado*=false;  
 public static Timer *t* = new Timer();  
  
 @Nullable  
 @Override  
 public IBinder onBind(Intent intent) {  
 return null;  
 }  
  
 *// se genera la notificacion primer plano del servicio inidicando que esta activo* @Override  
 public int onStartCommand(Intent intent, int flags, int startId) {  
  
 try {  
  
 createNotificationChannel();  
  
 Intent intent1=new Intent(this,ModoEstatico.class);  
  
 PendingIntent pendingIntent= PendingIntent.*getActivity*(this, 0, intent1,0);  
  
 Notification notification= new NotificationCompat.Builder(this,"ChannelId1")  
 .setSmallIcon(R.drawable.*bolsa*)  
 .setContentTitle("FINDER APP")  
 .setContentText("Modo Estático Activado.")  
 .setContentIntent(pendingIntent).build();  
  
 *activado*=true;  
  
 startForeground(1,notification);  
  
 }  
 catch (Exception e){  
 e.printStackTrace();  
 }  
  
 *// cuando se cumple el tiempo limite del modo estatico, se elimina la notificacion en primer plano  
  
 t*.scheduleAtFixedRate(new TimerTask() {  
 @Override  
 public void run() {  
 if (!Reminder.*activo*){  
 stopForeground(true);  
 }  
 }  
 }, 0, 2000);  
  
  
 return super.onStartCommand(intent, flags, startId);  
 }  
  
 *// se crea el canal de la notificacion en primer plano* private void createNotificationChannel() {  
 if(Build.VERSION.*SDK\_INT*>= Build.VERSION\_CODES.*O*){  
 NotificationChannel notificacionChannel= new NotificationChannel("ChannelId1","Foreground notification", NotificationManager.*IMPORTANCE\_DEFAULT*);  
 NotificationManager manager = getSystemService(NotificationManager.class);  
 manager.createNotificationChannel(notificacionChannel);  
 }  
  
 }  
  
 *// se destruye el servicio* @Override  
 public void onDestroy() {  
 *activado*=false;  
 stopForeground(true);  
 stopSelf();  
 super.onDestroy();  
 }  
  
}

### ***Reminder.java***

package com.example.finder;  
  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
  
import androidx.core.app.NotificationCompat;  
import androidx.core.app.NotificationManagerCompat;  
  
public class Reminder extends BroadcastReceiver {  
  
 public static Boolean *activo*;  
  
 *//se crea la notificacion de que se ha terminado el tiempo de modo estatico.* @Override  
 public void onReceive(Context context, Intent intent) {  
 if(ServicioModoEstatico.*activado*){  
 NotificationCompat.Builder builder= new NotificationCompat.Builder(context,"com.example.Finder1")  
 .setSmallIcon(R.drawable.*check*)  
 .setContentTitle("FINDER ALERT")  
 .setContentText("Modo Estático Desactivado.")  
 .setAutoCancel(true);  
 NotificationManagerCompat managerCompat = NotificationManagerCompat.*from*(context);  
 managerCompat.notify(999,builder.build());  
 }  
  
 ModoEstatico.*time*.cancel(true);  
  
 if(ModoEstatico.*enviando*){  
 ModoEstatico.*tiempoSMS*.cancel(true);  
 ModoEstatico.*enviando*=false;  
 ModoEstatico.*switchSMS*.setChecked(false);  
 }  
  
 *activo*= false;  
 }  
}

### ***ModoLive.java***

package com.example.finder;  
  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import androidx.fragment.app.FragmentActivity;  
  
import android.Manifest;  
import android.content.Context;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.location.Location;  
import android.location.LocationManager;  
import android.net.ConnectivityManager;  
import android.net.NetworkInfo;  
import android.os.Build;  
import android.os.Bundle;  
import android.os.Handler;  
import android.text.Html;  
import android.util.Log;  
import android.view.View;  
import android.widget.NumberPicker;  
import android.widget.Toast;  
  
import com.google.android.gms.maps.CameraUpdateFactory;  
import com.google.android.gms.maps.GoogleMap;  
import com.google.android.gms.maps.MapFragment;  
import com.google.android.gms.maps.OnMapReadyCallback;  
import com.google.android.gms.maps.model.BitmapDescriptorFactory;  
import com.google.android.gms.maps.model.LatLng;  
import com.google.android.gms.maps.model.Marker;  
import com.google.android.gms.maps.model.MarkerOptions;  
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 ModoLive extends AppCompatActivity implements OnMapReadyCallback {  
 GoogleMap map;  
 DatabaseReference db\_reference = FirebaseDatabase.*getInstance*().getReference();  
 double latitud = 0;  
 double longitud = 0;  
 Marker dispo;  
 Location loc;  
 LocationManager locManager;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_modo\_live*);  
  
 *//getSupportActionBar().hide();* MapFragment mapFragment = (MapFragment) getFragmentManager()  
 .findFragmentById(R.id.*map*);  
 mapFragment.getMapAsync(this);  
  
  
  
 }  
  
 @Override  
 protected void onStart() {  
 super.onStart();  
 ActivityCompat.*requestPermissions*(ModoLive.this, new String[]{Manifest.permission.*ACCESS\_FINE\_LOCATION*}, 1);  
 }  
  
 @Override  
 public void onMapReady(GoogleMap googlemap) {  
 if (ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_FINE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED* && ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_COARSE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED*) {  
 *// TODO: Consider calling  
 // ActivityCompat#requestPermissions  
 // here to request the missing permissions, and then overriding  
 // public void onRequestPermissionsResult(int requestCode, String[] permissions,  
 // int[] grantResults)  
 // to handle the case where the user grants the permission. See the documentation  
 // for ActivityCompat#requestPermissions for more details.* return;  
 }  
 map = googlemap;  
  
 db\_reference.child("Dispositivos").child("5").addValueEventListener(  
 new ValueEventListener() {  
 @Override  
 public void onDataChange(DataSnapshot dataSnapshot) {  
 *// Get Post object and use the values to update the UI* if (dispo != null) {  
 dispo.remove();  
 }  
  
 latitud = Double.*parseDouble*(String.*valueOf*(dataSnapshot.child("latitud").getValue()));  
 longitud= Double.*parseDouble*(String.*valueOf*(dataSnapshot.child("longitud").getValue()));  
  
 LatLng marca= new LatLng(latitud, longitud);  
 locManager = (LocationManager) getSystemService(Context.*LOCATION\_SERVICE*);  
 if (ActivityCompat.*checkSelfPermission*(ModoLive.this, Manifest.permission.*ACCESS\_FINE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED* && ActivityCompat.*checkSelfPermission*(ModoLive.this, Manifest.permission.*ACCESS\_COARSE\_LOCATION*) != PackageManager.*PERMISSION\_GRANTED*) {  
 *// TODO: Consider calling  
 // ActivityCompat#requestPermissions  
 // here to request the missing permissions, and then overriding  
 // public void onRequestPermissionsResult(int requestCode, String[] permissions,  
 // int[] grantResults)  
 // to handle the case where the user grants the permission. See the documentation  
 // for ActivityCompat#requestPermissions for more details.* return;  
 }  
 loc = locManager.getLastKnownLocation(LocationManager.*GPS\_PROVIDER*);  
 double lat2=loc.getLatitude();  
 double lng2=loc.getLongitude();  
 double radioTierra = 6371;*//en kilómetros* double dLat = Math.*toRadians*(lat2 - latitud);  
 double dLng = Math.*toRadians*(lng2 - longitud);  
 double sindLat = Math.*sin*(dLat / 2);  
 double sindLng = Math.*sin*(dLng / 2);  
 double va1 = Math.*pow*(sindLat, 2) + Math.*pow*(sindLng, 2)  
 \* Math.*cos*(Math.*toRadians*(latitud)) \* Math.*cos*(Math.*toRadians*(lat2));  
 double va2 = 2 \* Math.*atan2*(Math.*sqrt*(va1), Math.*sqrt*(1 - va1));  
 double distancia = (radioTierra \* va2) \* 1000;  
 int color=120;  
 if (distancia > Double.*parseDouble*(String.*valueOf*(dataSnapshot.child("distanciamax").getValue()))){  
 color=60;  
 }  
  
 dispo= map.addMarker(new MarkerOptions()  
 .position(marca)  
 .title(String.*format*("%.2f",distancia) + " m").icon(BitmapDescriptorFactory.*defaultMarker*(color)));  
 *// ...new MarkerOptions().position(pos)* }  
  
 @Override  
 public void onCancelled(DatabaseError databaseError) {  
 *// Getting Post failed, log a message* Log.*w*("aqui", "loadPost:onCancelled", databaseError.toException());  
 *// ...* }  
 }  
 );  
  
  
 map.setMyLocationEnabled(true);  
  
 LatLng marca= new LatLng(-2.1481404, -79.9666772);  
 map.addMarker(new MarkerOptions()  
 .position(marca)  
 .title("Marker"));  
 map.moveCamera(CameraUpdateFactory.*newLatLngZoom*(marca,17));  
  
 }  
  
 public void volverMenu(View view) {  
 startActivity(new Intent(getApplicationContext(), Home.class));  
 }  
  
 public void irModoEstatico(View view) {  
 startActivity(new Intent(getApplicationContext(), ModoEstatico.class));  
 }  
  
 public void irSettings(View view) {  
 numberPickerDialog();  
 }  
  
 *// se crea el number picker y se agregan funcionalidades a los botones ok y cancel.* private void numberPickerDialog(){  
 NumberPicker myNumberPicker= new NumberPicker(this);  
 myNumberPicker.setMaxValue(20);  
 myNumberPicker.setMinValue(0);  
 NumberPicker.OnValueChangeListener myValChangedListener= new NumberPicker.OnValueChangeListener() {  
 @Override  
 public void onValueChange(NumberPicker numberPicker, int i, int i1) {  
  
 }  
 };  
 myNumberPicker.setOnValueChangedListener(myValChangedListener);  
 android.app.AlertDialog.Builder alertaTiempo= new android.app.AlertDialog.Builder(this).setView(myNumberPicker);  
 alertaTiempo.setTitle(Html.*fromHtml*("<font color= '#00BCD4'>"+"Seleccione distancia máxima en metros:"+"</font>"));  
 alertaTiempo.setPositiveButton(android.R.string.*ok*, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
  
 db\_reference.child("Dispositivos").child("5").child("distanciamax").setValue(myNumberPicker.getValue());  
  
  
 }  
 });  
 alertaTiempo.setNegativeButton(android.R.string.*cancel*, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 Toast.*makeText*(getApplicationContext(),"Operación cancelada.",Toast.*LENGTH\_LONG*).show();  
  
 }  
 });  
 alertaTiempo.show();  
 }  
  
 public void conexionFirebase(View view) {  
 ConnectivityManager manager = (ConnectivityManager) getApplicationContext().getSystemService(Context.*CONNECTIVITY\_SERVICE*);  
 NetworkInfo activeNetwork = manager.getActiveNetworkInfo();  
  
 if (null == activeNetwork || !activeNetwork.isConnected()|| !activeNetwork.isAvailable()) {  
 AlertaInternet();  
 }  
 else{  
 *// agregar metodo que inicie maps* }  
 }  
  
 private void AlertaInternet(){  
 AlertDialog.Builder builder= new AlertDialog.Builder(ModoLive.this);  
 builder.setTitle("ALERTA")  
 .setMessage("No se puede establecer conexión con la nube.")  
 .setIcon(R.drawable.*warning*)  
 .setPositiveButton("Reintentar", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {  
 recreate();  
 }  
  
 })  
 .setCancelable(false);  
  
 AlertDialog dialog= builder.create();  
 dialog.show();  
 }  
}

### ***RegistroBaterias.java***

package com.example.finder;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.Handler;  
import android.view.View;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
  
import com.android.volley.Request;  
import com.android.volley.RequestQueue;  
import com.android.volley.Response;  
import com.android.volley.VolleyError;  
import com.android.volley.toolbox.JsonArrayRequest;  
import com.android.volley.toolbox.Volley;  
import com.github.mikephil.charting.charts.BarChart;  
import com.github.mikephil.charting.components.XAxis;  
import com.github.mikephil.charting.data.BarData;  
import com.github.mikephil.charting.data.BarDataSet;  
import com.github.mikephil.charting.data.BarEntry;  
import com.github.mikephil.charting.interfaces.datasets.IBarDataSet;  
import com.github.mikephil.charting.utils.ColorTemplate;  
  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.Map;  
  
public class RegistroBaterias extends AppCompatActivity {  
  
 public BarChart graficoBarras;  
 private RequestQueue ListaRequest = null;  
 private String token = "eyJ0eXAi...........................-mMIArvMc";  
 LinearLayout contenedorBaterias;  
 private Map<String, TextView> bateriasTVs;  
 private Map<String, TextView> fechasTVs;  
 private RegistroBaterias contexto;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_registro\_baterias*);  
  
 setTitle("Grafico de barras");  
 bateriasTVs = new HashMap<String,TextView>();  
 fechasTVs = new HashMap<String,TextView>();  
 ListaRequest = Volley.*newRequestQueue*(this);  
 contexto= this;  
 */\* GRAFICO \*/* this.iniciarGrafico();  
 this.solicitarBaterias();  
 }  
  
 *// ajustes visuales para el grafico de barras* public void iniciarGrafico() {  
 graficoBarras = findViewById(R.id.*barChart*);  
 graficoBarras.getDescription().setEnabled(false);  
 graficoBarras.setMaxVisibleValueCount(60);  
 graficoBarras.setPinchZoom(false);  
 graficoBarras.setDrawBarShadow(false);  
 graficoBarras.setDrawGridBackground(false);  
 XAxis xAxis = graficoBarras.getXAxis();  
 xAxis.setPosition(XAxis.XAxisPosition.*BOTTOM*);  
 xAxis.setDrawGridLines(false);  
 graficoBarras.getAxisLeft().setDrawGridLines(false);  
 graficoBarras.animateY(1500);  
 graficoBarras.getLegend().setEnabled(false);  
 }  
  
 *// se crea un archivo json con la informacion de la url de la base de datos* public void solicitarBaterias(){  
 String url\_registros = "https://finder-fa909.firebaseio.com/Dispositivos.json";  
 JsonArrayRequest requestRegistros =  
 new JsonArrayRequest(Request.Method.*GET*,  
 url\_registros, null, new Response.Listener<JSONArray>() {  
 @Override  
 public void onResponse(JSONArray response) {  
 mostrarBaterias(response);  
 actualizarGrafico(response);  
 }  
 }, new Response.ErrorListener() {  
 @Override  
 public void onErrorResponse(VolleyError error) {  
 System.*out*.println(error);  
 } }  
 ){ @Override  
 public Map<String, String> getHeaders() {  
 Map<String, String> params = new HashMap<>();  
 params.put("Authorization", "JWT " + token);  
 return params;  
 } };  
 ListaRequest.add(requestRegistros);  
 }  
  
 *// se presenta el porcentaje de bateria y las fechas en los contenedores* private void mostrarBaterias(JSONArray baterias){  
 String registroId;  
 JSONObject registroBat;  
 LinearLayout nuevoRegistro;  
 TextView fechaRegistro;  
 TextView valorRegistro;  
 contenedorBaterias = findViewById(R.id.*cont\_baterias*);  
 LinearLayout.LayoutParams parametrosLayout = new LinearLayout.LayoutParams(  
 LinearLayout.LayoutParams.*WRAP\_CONTENT*,  
 LinearLayout.LayoutParams.*WRAP\_CONTENT*, 1);  
 try  
 {  
 for (int i = 0; i < baterias.length(); i++) {  
 registroBat = (JSONObject) baterias.get(i);  
 registroId = registroBat.getString("id");  
 if (bateriasTVs.containsKey(registroId) && fechasTVs.containsKey(registroId)) {  
 fechaRegistro = fechasTVs.get(registroId);  
 valorRegistro = bateriasTVs.get(registroId);  
 fechaRegistro.setText(registroBat.getString("fecha"));  
 valorRegistro.setText(registroBat.getString("bateria") + " %");  
 } else {  
 nuevoRegistro = new LinearLayout(this);  
 nuevoRegistro.setOrientation(LinearLayout.*HORIZONTAL*);  
 fechaRegistro = new TextView(this);  
 fechaRegistro.setLayoutParams(parametrosLayout);  
 fechaRegistro.setText(registroBat.getString("fecha"));  
 nuevoRegistro.addView(fechaRegistro);  
 valorRegistro = new TextView(this);  
 valorRegistro.setLayoutParams(parametrosLayout);  
 valorRegistro.setText(registroBat.getString("bateria") + " %");  
 nuevoRegistro.addView(valorRegistro);  
 contenedorBaterias.addView(nuevoRegistro);  
 fechasTVs.put(registroId, fechaRegistro);  
 bateriasTVs.put(registroId, valorRegistro);  
 }  
 }  
  
 } catch (JSONException e) {  
 e.printStackTrace();  
 System.*out*.println("error");  
 }  
 }  
  
 *// al obtener un nuevo valor en el archivo json se actualiza en el grafico* private void actualizarGrafico(JSONArray baterias){  
 JSONObject registro\_baterias;  
 String bat;  
 String date;  
 int count = 0;  
 float bat\_val;  
 ArrayList<BarEntry> dato\_temp = new ArrayList<>();  
 try  
 {  
 for (int i = 0; i < baterias.length(); i++) { registro\_baterias = (JSONObject) baterias.get(i);  
 bat = registro\_baterias.getString("bateria");  
 date = registro\_baterias.getString("fecha");  
 bat\_val = Float.*parseFloat*(bat);  
 dato\_temp.add(new BarEntry(count, bat\_val)); count++;  
 }  
 } catch (JSONException e) { e.printStackTrace(); System.*out*.println("error");  
 }  
 System.*out*.println(dato\_temp);  
 llenarGrafico(dato\_temp);  
 }  
  
 *// se colocan los datos de las baterias en el grafico de barras y se solicita la info a la base de datos cada 3 segundos* private void llenarGrafico(ArrayList<BarEntry> dato\_bat){  
 BarDataSet bateriasDataSet;  
 if ( graficoBarras.getData() != null &&  
 graficoBarras.getData().getDataSetCount() > 0) {  
 bateriasDataSet = (BarDataSet)  
 graficoBarras.getData().getDataSetByIndex(0);  
 bateriasDataSet.setValues(dato\_bat); graficoBarras.getData().notifyDataChanged();  
 graficoBarras.notifyDataSetChanged();  
 } else {  
 bateriasDataSet = new BarDataSet(dato\_bat, "Data Set");  
 bateriasDataSet.setColors(ColorTemplate.*VORDIPLOM\_COLORS*);  
 bateriasDataSet.setDrawValues(true);  
 ArrayList<IBarDataSet> dataSets = new ArrayList<>();  
 dataSets.add(bateriasDataSet);  
 BarData data = new BarData(dataSets);  
 graficoBarras.setData(data);  
 graficoBarras.setFitBars(true);  
 }  
 graficoBarras.invalidate();  
 final Handler handler = new Handler();  
 final Runnable runnable = new Runnable() {  
 @Override  
 public void run() {  
 solicitarBaterias();  
 } };  
 handler.postDelayed(runnable, 3000);  
 }  
  
 *// metodo que permite regresar al menu principal* public void regresarHome(View view){  
 startActivity(new Intent(getApplicationContext(), Home.class));  
 }  
  
}

### ***MyFirebaseInstanceService.java***

package com.example.finder.Services;  
  
import android.app.Notification;  
import android.app.NotificationChannel;  
import android.app.NotificationManager;  
import android.content.Context;  
import android.graphics.Color;  
import android.os.Build;  
import android.util.Log;  
  
import androidx.core.app.NotificationCompat;  
  
import com.example.finder.R;  
import com.google.firebase.messaging.FirebaseMessagingService;  
import com.google.firebase.messaging.RemoteMessage;  
  
import java.util.Random;  
  
public class MyFirebaseInstanceService extends FirebaseMessagingService {  
  
 @Override  
 public void onNewToken(String s) {  
 super.onNewToken(s);  
 Log.*d*("TOKENFIREBASE", s);  
 }  
  
 public void onMessageReceived(RemoteMessage remoteMessage) {  
 super.onMessageReceived(remoteMessage);  
 showNotification(remoteMessage.getNotification().getTitle(), remoteMessage.getNotification().getBody());  
 }  
  
 private void showNotification(String title, String body) {  
 NotificationManager notificationManager =  
 (NotificationManager) getSystemService(Context.*NOTIFICATION\_SERVICE*);  
 String NOTIFICATION\_CHANNEL\_ID = "com.amst.firebasenotify.test";  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {  
 NotificationChannel notificationChannel =  
 new NotificationChannel(NOTIFICATION\_CHANNEL\_ID, "Notification", NotificationManager.*IMPORTANCE\_DEFAULT*);  
 notificationChannel.setDescription("EDMT Channel");  
 notificationChannel.enableLights(true);  
 notificationChannel.setLightColor(Color.*BLUE*);  
 notificationChannel.setVibrationPattern(new long[]{0, 1000, 500, 1000});  
 notificationManager.createNotificationChannel(notificationChannel);  
 }  
 NotificationCompat.Builder notificationBuilder =  
 new NotificationCompat.Builder(this, NOTIFICATION\_CHANNEL\_ID);  
 notificationBuilder.setAutoCancel(true)  
 .setDefaults(Notification.*DEFAULT\_ALL*)  
 .setWhen(System.*currentTimeMillis*())  
 .setContentTitle(title)  
 .setContentText(body)  
 .setSmallIcon(R.drawable.*mensaje*)  
 .setContentInfo("info");  
 notificationManager.notify(  
 new Random().nextInt(), notificationBuilder.build());  
  
 }  
  
}

## **LAYOUTS XML**

### ***Activity\_splah.xml***

*<?*xml version="1.0" encoding="utf-8"*?>*<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".Splash">  
  
 <ImageView  
 android:id="@+id/imageView4"  
 android:layout\_width="367dp"  
 android:layout\_height="655dp"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_alignParentEnd="true"  
 android:layout\_alignParentRight="true"  
 android:layout\_alignParentBottom="true"  
 android:layout\_marginStart="23dp"  
 android:layout\_marginLeft="23dp"  
 android:layout\_marginTop="38dp"  
 android:layout\_marginEnd="23dp"  
 android:layout\_marginRight="23dp"  
 android:layout\_marginBottom="38dp"  
 app:srcCompat="@drawable/logo2" />  
</RelativeLayout>

### ***Activity\_main.xml***

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:padding="35dp"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/txt\_"  
 android:layout\_width="match\_parent"  
 android:layout\_height="100dp"  
 android:fontFamily="sans-serif-smallcaps"  
 android:gravity="center"  
 android:text="¡Bienvenido a Finder!"  
 android:textAppearance="@style/TextAppearance.AppCompat.Body1"  
 android:textColor="@color/teal\_700"  
 android:textSize="34sp"  
 android:textStyle="bold"  
 tools:text="¡Bienvenido a Finder!" />  
  
 <TextView  
 android:id="@+id/txtCorreo"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:text="Ingrese correo electrónico:"  
 android:textColor="@color/cardview\_dark\_background"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/Edit\_txtCorreo"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="Correo electrónico"  
 android:inputType="textPersonName" />  
  
 <TextView  
 android:id="@+id/txtVacio3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
  
 <TextView  
 android:id="@+id/txtContrasena"  
 android:layout\_width="match\_parent"  
 android:layout\_height="35dp"  
 android:text="Ingrese contraseña:"  
 android:textColor="@color/cardview\_dark\_background"  
 android:textSize="18sp"  
 android:textStyle="bold" />  
  
 <EditText  
 android:id="@+id/Edit\_txtContrasena"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:ems="10"  
 android:hint="Contraseña"  
 android:inputType="textPassword" />  
  
 <TextView  
 android:id="@+id/txtVacio4"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
  
 <Button  
 android:id="@+id/btn\_iniciarSesion"  
 android:layout\_width="300dp"  
 android:layout\_height="57dp"  
 android:layout\_gravity="center"  
 android:onClick="Login"  
 android:text="Iniciar Sesión"  
 android:textStyle="bold"  
 app:backgroundTint="@android:color/holo\_blue\_dark" />  
  
 <TextView  
 android:id="@+id/txtVacio5"  
 android:layout\_width="match\_parent"  
 android:layout\_height="8dp" />  
  
 <Button  
 android:id="@+id/btn\_login"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:layout\_gravity="center"  
 android:fontFamily="sans-serif-black"  
 android:onClick="iniciarSesion"  
 android:text="Iniciar sesión con Google"  
 app:backgroundTint="?attr/colorAccent"  
 app:icon="@drawable/google"  
 app:iconGravity="start"  
 app:iconSize="40dp" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:padding="20dp">  
  
 <TextView  
 android:id="@+id/txt\_vacio8"  
 android:layout\_width="5dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="0.1" />  
  
 <TextView  
 android:id="@+id/txt\_info"  
 android:layout\_width="69dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_weight="1"  
 android:text="No tiene una cuenta?"  
 android:textColor="@color/cardview\_dark\_background"  
 android:textSize="15sp" />  
  
 <TextView  
 android:id="@+id/txt\_registrarse"  
 android:layout\_width="36dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:layout\_weight="1"  
 android:onClick="registro"  
 android:text="Regístrese aquí"  
 android:textColor="@color/teal\_700"  
 android:textSize="16sp"  
 android:textStyle="bold" />  
 </LinearLayout>  
  
</LinearLayout>

### ***Activity\_registro.xml***

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="30dp"  
 tools:context=".Registro">  
  
 <TextView  
 android:id="@+id/txt\_registro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="46dp"  
 android:fontFamily="sans-serif-smallcaps"  
 android:gravity="center"  
 android:text="Registro"  
 android:textColor="@color/teal\_700"  
 android:textSize="34sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/txt\_vacio6"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="70dp"  
 android:orientation="horizontal">  
  
 <ImageView  
 android:id="@+id/imageView\_correo"  
 android:layout\_width="1dp"  
 android:layout\_height="45dp"  
 android:layout\_weight="0.76"  
 app:srcCompat="@android:drawable/sym\_action\_email" />  
  
 <EditText  
 android:id="@+id/et\_correo"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:ems="10"  
 android:hint="Ingrese correo electrónico"  
 android:inputType="textEmailAddress"  
 android:textSize="16sp" />  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="70dp"  
 android:orientation="horizontal">  
  
 <ImageView  
 android:id="@+id/im\_password1"  
 android:layout\_width="1dp"  
 android:layout\_height="45dp"  
 android:layout\_weight="0.76"  
 app:srcCompat="@android:drawable/ic\_lock\_idle\_lock" />  
  
 <EditText  
 android:id="@+id/ed\_password1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:ems="10"  
 android:hint="Ingrese contraseña"  
 android:textSize="16sp"  
 android:inputType="textPassword" />  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="70dp"  
 android:orientation="horizontal">  
  
 <ImageView  
 android:id="@+id/im\_password2"  
 android:layout\_width="1dp"  
 android:layout\_height="45dp"  
 android:layout\_weight="0.76"  
 app:srcCompat="@android:drawable/ic\_lock\_idle\_lock" />  
  
 <EditText  
 android:id="@+id/ed\_password2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:ems="10"  
 android:hint="Confirme contraseña"  
 android:textSize="16sp"  
 android:inputType="textPassword" />  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="70dp"  
 android:orientation="horizontal">  
  
 <ImageView  
 android:id="@+id/im\_celular"  
 android:layout\_width="1dp"  
 android:layout\_height="45dp"  
 android:layout\_weight="0.76"  
 app:srcCompat="@android:drawable/ic\_menu\_call" />  
  
 <EditText  
 android:id="@+id/ed\_celular"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:ems="10"  
 android:hint="Ingrese número de teléfono"  
 android:textSize="16sp"  
 android:inputType="number" />  
  
 </LinearLayout>  
  
 <Button  
 android:id="@+id/btn\_registro"  
 android:layout\_width="280dp"  
 android:layout\_height="50dp"  
 android:layout\_gravity="center"  
 android:onClick="Registro"  
 android:text="registrarse"  
 android:textStyle="bold"  
 app:backgroundTint="@android:color/holo\_blue\_dark" />  
  
</LinearLayout>

### ***Activity\_home.xml***

*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".Home">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="30dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <TextView  
 android:id="@+id/txt\_menu"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:fontFamily="sans-serif-smallcaps"  
 android:gravity="center"  
 android:text="Menú Principal"  
 android:textColor="@color/teal\_700"  
 android:textSize="34sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
  
 <ImageView  
 android:id="@+id/imageView2"  
 android:layout\_width="292dp"  
 android:layout\_height="80dp"  
 android:layout\_gravity="center"  
 app:srcCompat="@drawable/bolsoazul" />  
  
 <TextView  
 android:id="@+id/txt\_nadaHome"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
  
 <TextView  
 android:id="@+id/txt\_texto"  
 android:layout\_width="263dp"  
 android:layout\_height="33dp"  
 android:gravity="center"  
 android:text="Elija un dispositivo activo:"  
 android:textSize="16sp"  
 android:textStyle="bold" />  
  
 <Spinner  
 android:id="@+id/spinner"  
 android:layout\_width="294dp"  
 android:layout\_height="43dp"  
 android:layout\_gravity="center" />  
  
  
 <Button  
 android:id="@+id/btn\_modo"  
 android:layout\_width="280dp"  
 android:layout\_height="47dp"  
 android:layout\_gravity="center"  
 android:gravity="center"  
 android:onClick="modo"  
 android:text="Modo Aplicación"  
 android:textStyle="bold"  
 app:backgroundTint="@android:color/holo\_blue\_dark" />  
  
 <Button  
 android:id="@+id/btn\_barraBateria"  
 android:layout\_width="280dp"  
 android:layout\_height="47dp"  
 android:layout\_gravity="center"  
 android:gravity="center"  
 android:onClick="registroBaterias"  
 android:text="Registro de Baterías"  
 android:textStyle="bold"  
 app:backgroundTint="@android:color/holo\_blue\_dark" />  
  
 <TextView  
 android:id="@+id/txt\_vacio"  
 android:layout\_width="match\_parent"  
 android:layout\_height="17dp" />  
  
 <Button  
 android:id="@+id/btn\_salir"  
 android:layout\_width="280dp"  
 android:layout\_height="47dp"  
 android:layout\_gravity="center"  
 android:gravity="center"  
 android:onClick="cerrarSesion"  
 android:text="cerrar sesión"  
 android:textStyle="bold"  
 app:backgroundTint="@color/teal\_200"  
 app:icon="@drawable/common\_full\_open\_on\_phone"  
 app:iconPadding="1dp"  
 app:iconSize="35dp" />  
  
  
 </LinearLayout>  
</androidx.constraintlayout.widget.ConstraintLayout>

### ***Dialog\_telefono.xml***

*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="200dp">  
  
 <TextView  
 android:id="@+id/txt\_tlf"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Ingrese su número de teléfono:"  
 android:textColor="@color/teal\_700"  
 android:textColorHighlight="@color/black"  
 android:textSize="16sp"  
 android:textStyle="bold"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.247"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.342" />  
  
 <EditText  
 android:id="@+id/edit\_tlf"  
 android:layout\_width="279dp"  
 android:layout\_height="40dp"  
 android:ems="10"  
 android:inputType="phone"  
 android:textSize="16sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.348"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/txt\_tlf"  
 app:layout\_constraintVertical\_bias="0.285" />  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

### ***Activity\_modo\_estatico.xml***

*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/fondo"  
 tools:context=".ModoEstatico">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="20dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <TextView  
 android:id="@+id/txt\_modoEstatico"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:fontFamily="sans-serif-smallcaps"  
 android:gravity="center"  
 android:text="Modo Estático"  
 android:textColor="@color/teal\_700"  
 android:textSize="34sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/txt\_nada"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
  
 <ImageView  
 android:id="@+id/img\_puntero"  
 android:layout\_width="match\_parent"  
 android:layout\_height="87dp"  
 app:srcCompat="@drawable/puntero" />  
  
 <TextView  
 android:id="@+id/txt\_nada2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
  
 <Button  
 android:id="@+id/btn\_duracion"  
 android:layout\_width="280dp"  
 android:layout\_height="47dp"  
 android:layout\_gravity="center"  
 android:onClick="aggTiempo"  
 android:text="Duración Máxima"  
 android:textStyle="bold"  
 app:backgroundTint="@android:color/holo\_blue\_dark" />  
  
 <TextView  
 android:id="@+id/txt\_nada3"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
  
 <Switch  
 android:id="@+id/switchSMS"  
 android:layout\_width="279dp"  
 android:layout\_height="38dp"  
 android:layout\_gravity="center"  
 android:fontFamily="sans-serif-black"  
 android:text="NOTIFICACIONES SMS"  
 android:textColor="@color/cardview\_dark\_background"  
 android:textSize="16sp" />  
  
 <TextView  
 android:id="@+id/txt\_nada6"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
  
 <Button  
 android:id="@+id/btn\_cambio"  
 android:layout\_width="280dp"  
 android:layout\_height="47dp"  
 android:layout\_gravity="center"  
 android:onClick="cambiarModo"  
 android:text="Cambiar modo"  
 android:textStyle="bold"  
 app:backgroundTint="@android:color/holo\_blue\_dark" />  
  
 <TextView  
 android:id="@+id/txt\_nada4"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
  
 <Button  
 android:id="@+id/btn\_volver"  
 android:layout\_width="280dp"  
 android:layout\_height="47dp"  
 android:layout\_gravity="center"  
 android:onClick="volverMenu"  
 android:text="Volver al menú"  
 android:textStyle="bold"  
 app:backgroundTint="@color/teal\_200" />  
  
 <TextView  
 android:id="@+id/txt\_nada5"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
  
 </LinearLayout>  
</androidx.constraintlayout.widget.ConstraintLayout>

### ***Activity\_modo\_live.xml***

*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center"  
 android:orientation="vertical"  
 android:background="#ECE8E9"  
 tools:context=".ModoLive">  
  
 <fragment  
 android:id="@+id/map"  
 android:name="com.google.android.gms.maps.MapFragment"  
 android:layout\_width="match\_parent"  
  
 android:layout\_height="0dp"  
 app:layout\_constraintBottom\_toTopOf="@+id/line"  
 app:layout\_constraintTop\_toTopOf="parent" />  
 <LinearLayout  
 android:id="@+id/line"  
 android:layout\_width="match\_parent"  
 android:layout\_height="55dp"  
 android:orientation="vertical"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.6"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:background="#fff"  
 app:layout\_constraintVertical\_bias="1.0">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="2dp"  
 android:layout\_weight="1"  
 android:orientation="vertical">  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="30dp"  
 android:layout\_weight="1"  
 android:orientation="horizontal">  
  
 <ImageView  
 android:id="@+id/imgMark"  
 android:layout\_width="35dp"  
 android:layout\_height="35dp"  
 android:layout\_marginHorizontal="53dp"  
 android:onClick="irModoEstatico"  
 app:srcCompat="@drawable/ic\_mark" />  
  
 <ImageView  
 android:id="@+id/imgHome"  
 android:layout\_width="35dp"  
 android:layout\_height="35dp"  
 android:layout\_marginHorizontal="52dp"  
 android:onClick="volverMenu"  
 app:srcCompat="@drawable/ic\_home" />  
  
 <ImageView  
 android:id="@+id/imgSet"  
 android:layout\_width="35dp"  
 android:layout\_height="35dp"  
 android:layout\_marginHorizontal="40dp"  
 android:onClick="irSettings"  
 app:srcCompat="@drawable/ic\_setting" />  
  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="20dp"  
 android:layout\_weight="1"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/textModo"  
 android:layout\_width="100dp"  
 android:layout\_height="20dp"  
 android:layout\_marginHorizontal="25dp"  
 android:gravity="center"  
 android:text="Modo Estático"  
 android:textColor="@color/common\_google\_signin\_btn\_text\_light\_focused"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textHome"  
 android:layout\_width="50dp"  
 android:layout\_height="20dp"  
 android:layout\_marginHorizontal="36dp"  
 android:gravity="center"  
 android:text="Home"  
 android:textColor="@color/common\_google\_signin\_btn\_text\_light\_default"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/textSettings"  
 android:layout\_width="100dp"  
 android:layout\_height="20dp"  
 android:layout\_marginHorizontal="14dp"  
 android:gravity="center"  
 android:text="Configuración"  
 android:textColor="@color/common\_google\_signin\_btn\_text\_light\_default"  
 android:textStyle="bold" />  
 </LinearLayout>  
 </LinearLayout>  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

### ***Activity\_registro\_baterias.xml***

*<?*xml version="1.0" encoding="utf-8"*?>*<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:fillViewport="true"  
 android:scrollbars="vertical"  
 android:padding="20dp"  
 android:scrollbarStyle="insideInset">  
  
 <LinearLayout  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"  
 android:orientation="vertical"  
 android:padding="10dp"  
 android:id="@+id/contenedor">  
  
 <com.github.mikephil.charting.charts.BarChart  
 android:id="@+id/barChart"  
 android:layout\_width="match\_parent"  
 android:layout\_height="260dp"></com.github.mikephil.charting.charts.BarChart>  
  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="50dp"  
 android:gravity="bottom|center"  
 android:text="Registro de Baterias"  
 android:textSize="24sp"  
 android:textStyle="bold" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/textView3"  
 android:layout\_width="48dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:background="#635D5D"  
 android:text="Fecha obtenida"  
 android:textColor="#FFFFFF"  
 android:textStyle="bold" />  
 <TextView  
 android:id="@+id/textView5"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:background="#635D5D"  
 android:text="Valor"  
 android:textColor="#FFFFFF"  
 android:textStyle="bold" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:id="@+id/cont\_baterias"  
 android:layout\_width="match\_parent"  
 android:layout\_height="138dp"  
 android:orientation="vertical">  
  
 </LinearLayout>  
  
 <Button  
 android:id="@+id/btn\_regresoHome"  
 android:layout\_width="280dp"  
 android:layout\_height="47dp"  
 android:layout\_gravity="center"  
 android:gravity="center"  
 android:onClick="regresarHome"  
 android:text="VOLVER AL MENÚ"  
 android:textStyle="bold"  
 app:backgroundTint="@color/teal\_200" />  
  
 </LinearLayout>  
  
</ScrollView>