APLICACIONES MÓVILES Y SERVICIOS TELEMÁTICOS

1T - 2020

G2 - Integrantes:

Ronny Benitez  
Gabriel Murillo  
Eileen Sanchez

Manual Técnico  
G-Bag

Tracking de mochila v2.0



Escuela Superior Politécnica del Litoral

Tabla de contenido

[Introducción 2](#_Toc51346358)

[Descripción del sistema implementado 2](#_Toc51346359)

[Recursos 2](#_Toc51346360)

[Diagramas 2](#_Toc51346361)

[Diagrama del Sistema 2](#_Toc51346362)

[Diagrama de despliegue 3](#_Toc51346363)

[Modelo de la Base de datos 4](#_Toc51346364)

[Diagrama de clases 4](#_Toc51346365)

[Casos de uso 5](#_Toc51346366)

[Diagrama del circuito 5](#_Toc51346367)

[Explicación del código fuente 7](#_Toc51346368)

[Código fuente Arduino 7](#_Toc51346369)

[Archivo: bluettoharduino.ino 7](#_Toc51346370)

[Código fuente java 11](#_Toc51346371)

[Layouts 60](#_Toc51346372)

[activity\_modo\_mochila.xml 66](#_Toc51346373)

[activity\_registro.xml 70](#_Toc51346374)

[fragment\_mapa.xml 75](#_Toc51346375)

[fragment\_mochila.xml 76](#_Toc51346376)

[fragment\_perfil.xml 77](#_Toc51346377)

[informacion\_dipositivo.xml 82](#_Toc51346378)

[items\_list.xml 84](#_Toc51346379)

# Introducción

El presente manual técnico describe los recursos de hardware y software utilizados en la elaboración de la propuesta de solución del proyecto.

Este proyecto tiene como finalidad permitir a los usuarios rastrear sus mochilas en tiempo real a través de sus dispositivos Smartphone, poder mantener una mochila en un sitio y recibir notificaciones al momento de moverse.

# Descripción del sistema implementado

El sistema consta de una aplicación móvil dirigida para estudiantes de la Escuela Superior Politécnica del Litoral.

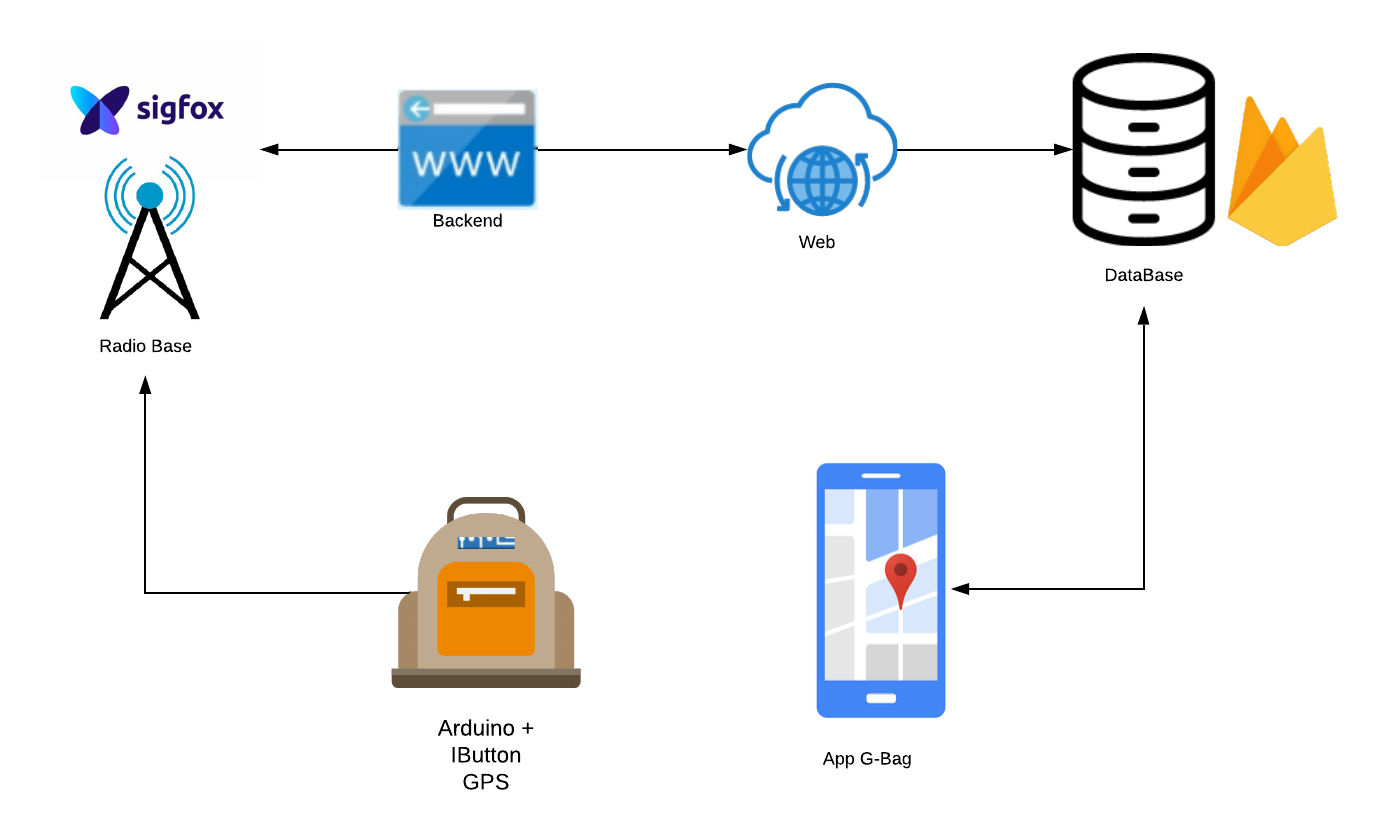
Un usuario estudiante puede visulizar un mapa que indica la ubicación de las mochilas registradas y poder cambiar a modo estatico y estar alerta si su mochila cambia de sitio.

# Recursos

|  |  |  |
| --- | --- | --- |
| Hardware | Software | Extras |
| Arduino UNO R3 | Android studio | Caja negra |
| Modulo de Bluetooth | Firebase | Cautín |
| Módulo GPS ibutton | Arduino V1.8.13 | Estaño |
| Modulo de bateria | Heroku | Cable arduino |
| Modulo SIM | ApiRest Express | Jumpers |
|  |  |  |

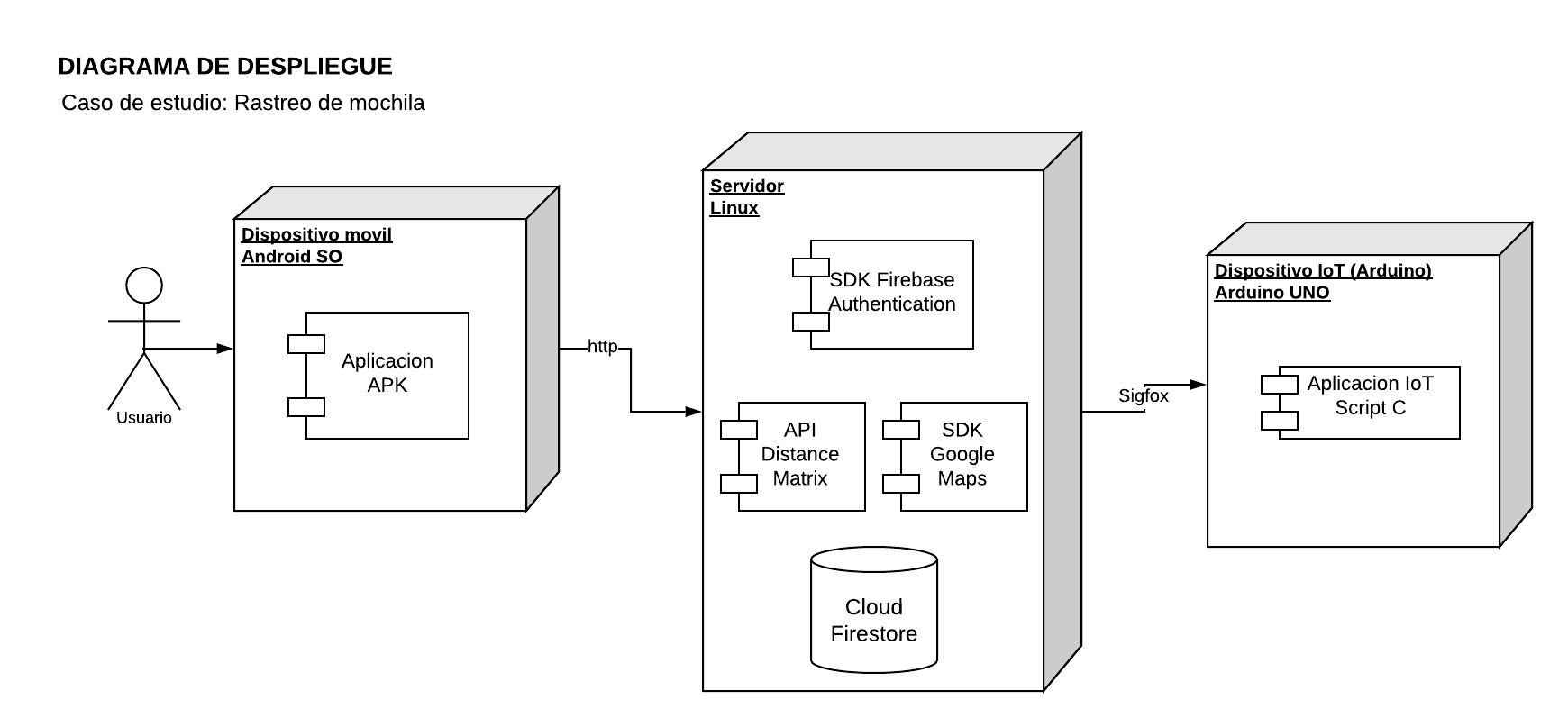
# Diagramas

## Diagrama del Sistema



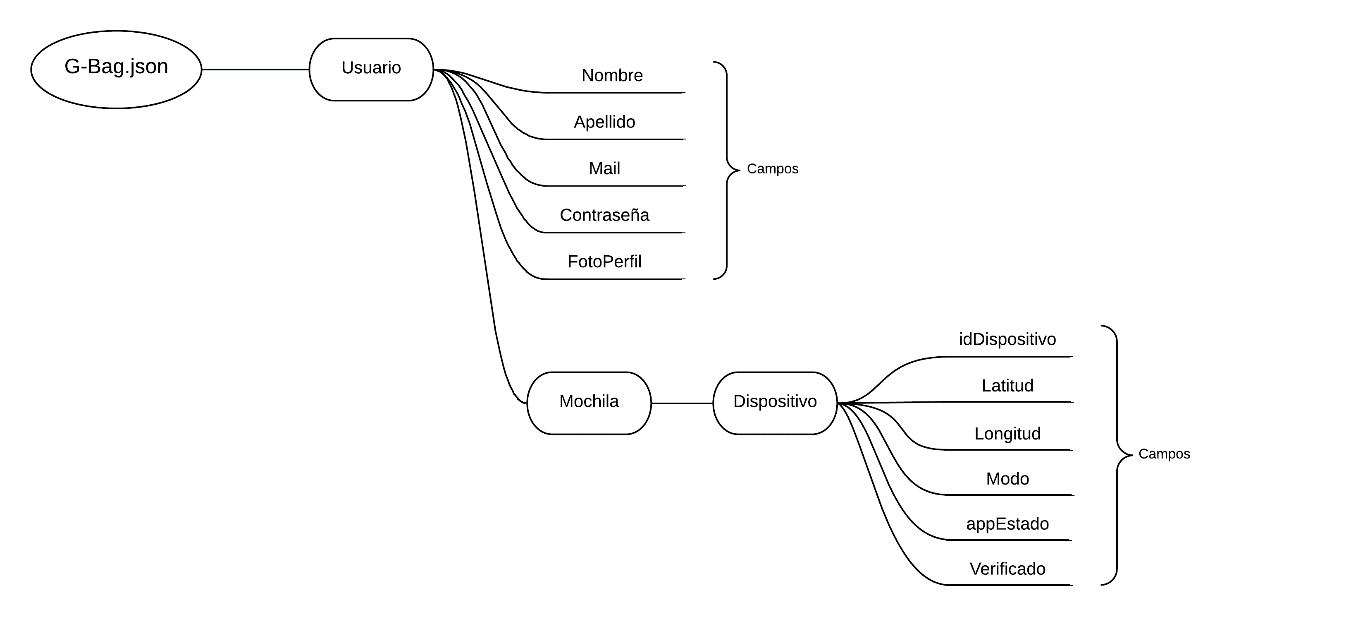
## Diagrama de despliegue

La aplicación sensa y envía datos en tiempo real mediane el dispositivo IoT desarrollado, a través de este, se obtiene cada cierto tiempo la ubicación de la mochila y es enviada mediante callbacks al BD en Firebase para luego ser consultada por los usuarios de la aplicación.

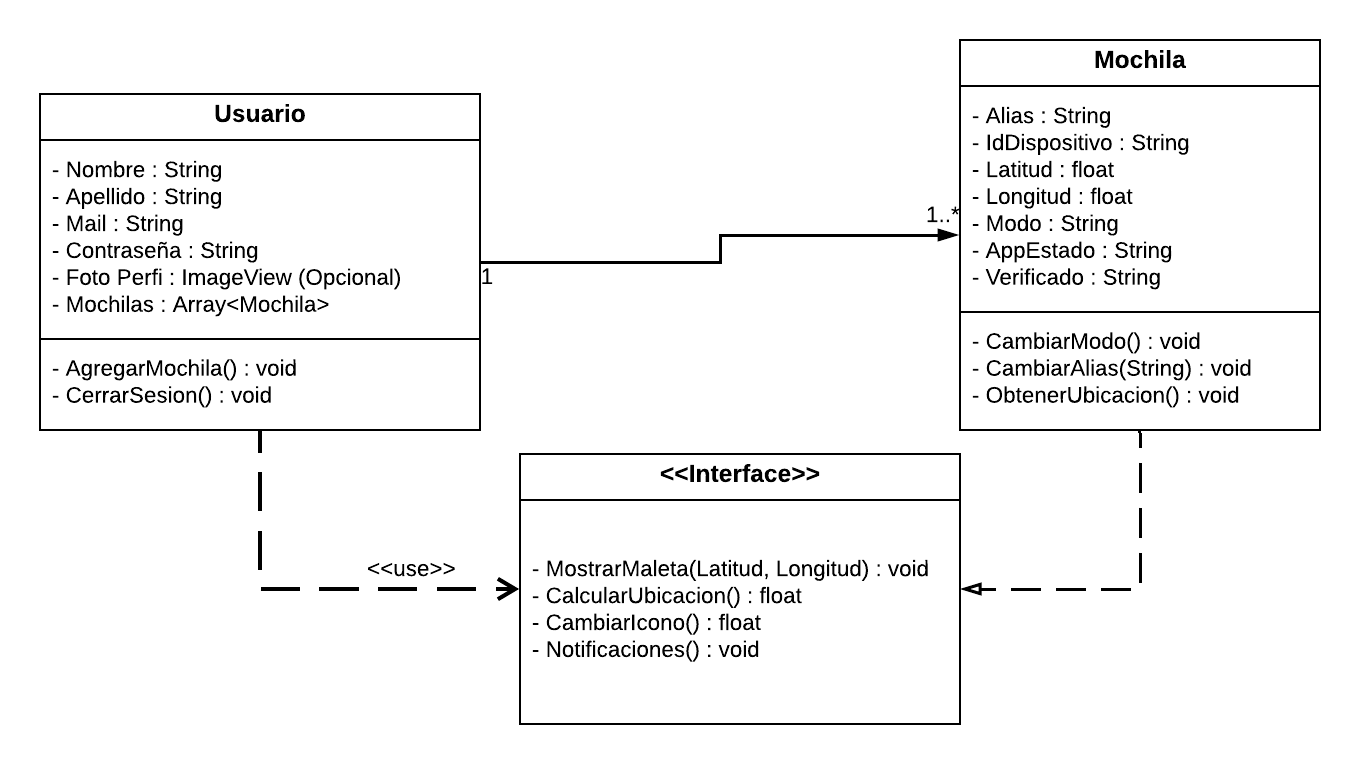


## Modelo de la Base de datos

Base de datos no relacional, representa la estructura de los datos almacenados en la base de datos de Firebase.

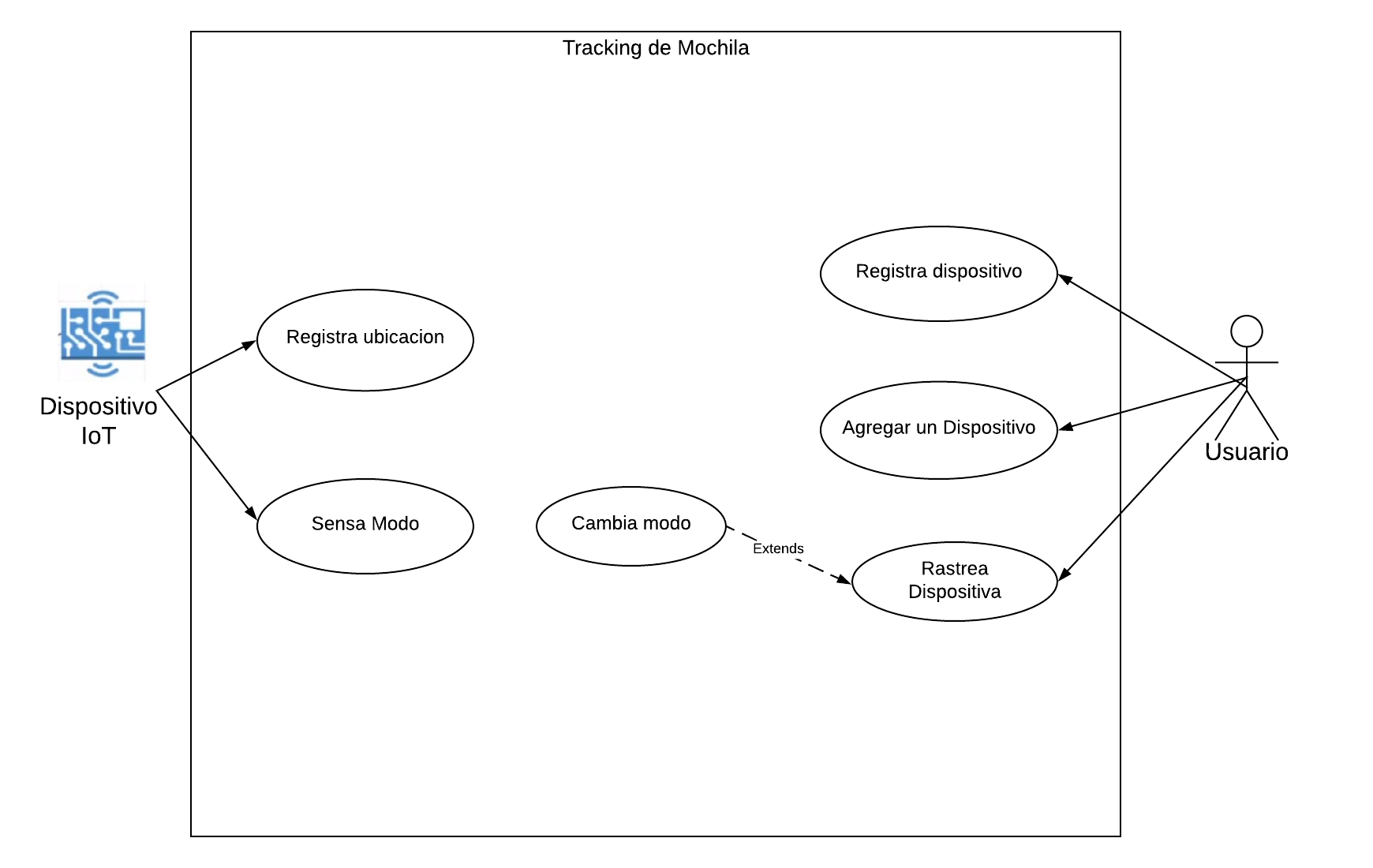


## Diagrama de clases



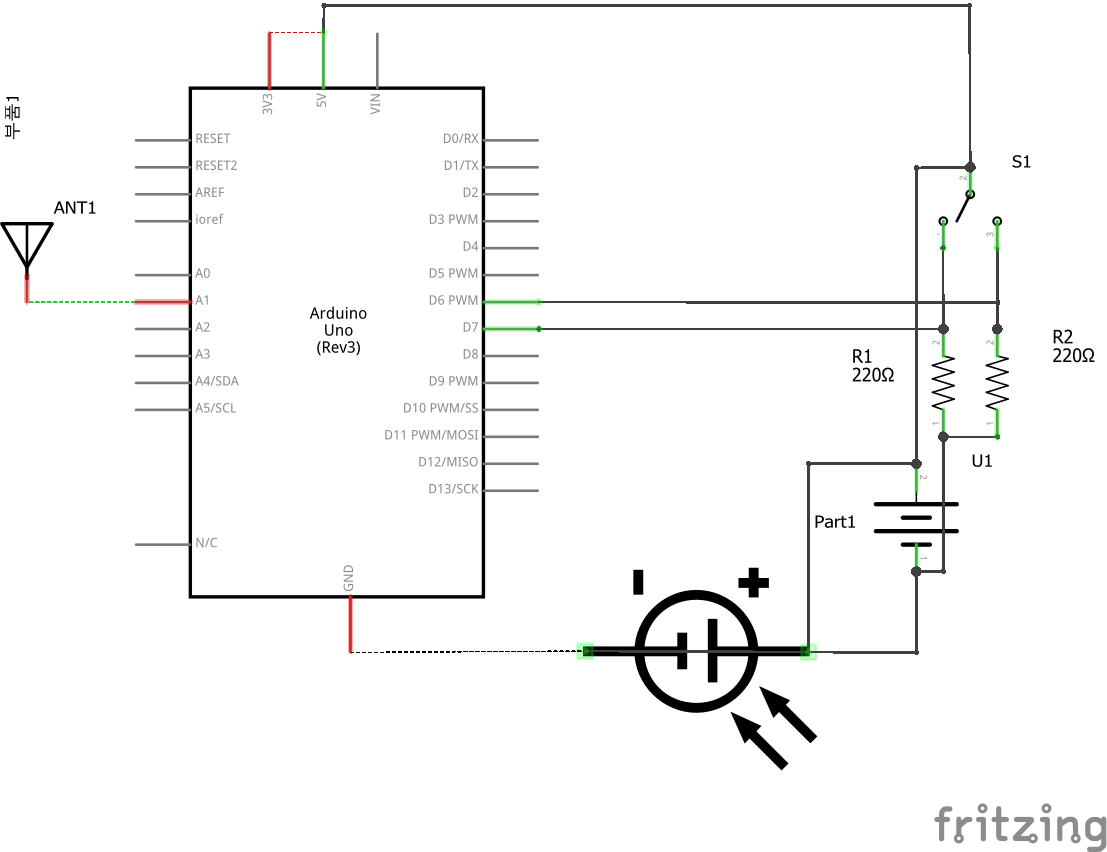
## Casos de uso

Detallan las diferentes interacciones entre las diferentes opciones que ofrece la aplicación y como se relacionen entre si dentro del sistema.



## Diagrama del circuito

Muestra las conexiones y los recursos utilizados para armar el dispositivo que obtiene la ubicación de las mochilas en tiempo real.



# Explicación del código fuente

## Código fuente Arduino

## Archivo: **bluettoharduino.ino**

|  |  |
| --- | --- |
| Función: | void setup() |
| Descripción: |  |
| pinMode(relay,OUTPUT);  int flagInit;  flagInit = -1;  while (flagInit == -1) {  Serial.println(""); // Make a clean restart  delay(1000);  flagInit = Isigfox->initSigfox();  Isigfox->testComms();  GetDeviceID();  //Isigfox->setPublicKey(); // set public key for usage with SNEK  }  tSensors->initSensors();  Serial.println(""); // Make a clean start  delay(1000);  blue.begin(9600);  pinMode(13, OUTPUT);  digitalWrite(13, HIGH);  delay(4000);  digitalWrite(13, LOW);  blue.print("AT");  delay(1000);  blue.print("AT+NAME=");  blue.print(NOMBRE);  delay(1000);  blue.print("AT+BAUD");  blue.print(BPS);  delay(1000);  blue.print("AT+PIN");  blue.print(PASS);  delay(1000); | |

|  |  |
| --- | --- |
| Función: | void loop() |
| Descripción: |  |
| void loop() {  analogValue = analogRead(AI\_BATTERY);  voltage = map(analogValue, 0, 1023, 0, 100);  if (blue.available()) {  usuario = blue.read();  Serial.println(usuario);  }  if (usuario == '+' || usuario == 'D' || usuario == 'I' || usuario == 'S' || usuario == 'C' || usuario == 'S' || usuario == 'U' || usuario == 'E' || usuario == 'O' || usuario == 'K' || usuario == ':' || usuario == '\n' || usuario == ' ') {  Serial.print("EPROM: ");  Serial.println(EEPROM.get(0, usuario));  usuario = EEPROM.get(0, usuario);  }  if (usuario == 'e') {  modo = 'e';  Serial.println("MODO ESTATICO");  } else if (usuario == 'r') {  modo = 'r';  Serial.println("MODO REAL");  }  if (modo == 'e') {  Serial.println("MODO ESTATICO");  delay(1000);  if (usuario == '0' || usuario == '1' || usuario == '2' || usuario == '3' || usuario == '4' || usuario == '5' || usuario == '6' || usuario == '7' || usuario == '8' || usuario == '9') {  int tiempo = String(usuario).toInt();  tiempo\_total = tiempo\_sis \* tiempo;  EEPROM.put(0, usuario);  if (i < tiempo\_total) {  obtener\_aceleracion();  } else {  modo = 'r';  i = 0;  }  i++;  Serial.println(i);  delay(1000);  }  //obtener\_aceleracion();  } else if (modo == 'r') {  Serial.println("MODO REAL");  enviarSigfox(voltage, 1.0);  activar\_ibutton();  desactivar\_ibutton();  //delay(600000);//tiempo de espera 10 minutos  delay(60000);  }  delay(1000);  } | |

|  |  |
| --- | --- |
| Función: | void obtener\_aceleracion () |
| Descripción: |  |
| void obtener\_aceleracion() {  INT16\_t x\_g, y\_g, z\_g;  acceleration\_xyz \*xyz\_g;  xyz\_g = (acceleration\_xyz \*)malloc(sizeof(acceleration\_xyz));  tSensors->getAccXYZ(xyz\_g);  x\_g.number = (int16\_t) (xyz\_g->x\_g \* 250);  y\_g.number = (int16\_t) (xyz\_g->y\_g \* 250);  z\_g.number = (int16\_t) (xyz\_g->z\_g \* 250);  float x = (float)x\_g.number / 250;  float y = (float)y\_g.number / 250;  Serial.print("Acc X: "); Serial.println(x);  Serial.print("Acc Y: "); Serial.println(y);  if (abs(x) > 1.5 or abs(y) > 0.1) {  if (movimiento == 0 or movimiento == 500UL) {  Serial.println("ENVIANDO DATOS...");  delay(5000);  enviarSigfox(voltage, 1.0);  activar\_ibutton();  desactivar\_ibutton();  //Serial.print("Acc X: "); Serial.println(x);  //Serial.print("Acc Y: "); Serial.println(y);  Serial.println("DATOS ENVIADO SATISFACTORIAMENTE...");  } else if (movimiento > 500UL) {  movimiento = 0;  }  movimiento++;  } else {  movimiento = 0;  }  Serial.print("\0");  free(xyz\_g);  delay(1000);  } | |

|  |  |
| --- | --- |
| Función: | void GetDeviceID () |
| Descripción: |  |
| void GetDeviceID() {  recvMsg \*RecvMsg;  const char msg[] = "AT$I=10";  RecvMsg = (recvMsg \*)malloc(sizeof(recvMsg));  Isigfox->sendMessage(msg, 7, RecvMsg);  Serial.print("Device ID: ");  for (int i = 0; i < RecvMsg->len; i++) {  Serial.print(RecvMsg->inData[i]);  }  Serial.println("");  free(RecvMsg);  } | |

|  |  |
| --- | --- |
| Función: | void enviarSigfox () |
| Descripción: |  |
| void enviarSigfox(float voltajeMedido, float activMovimiento) {  byte \*float\_volt = (byte \*)&voltajeMedido;  byte \*float\_movimiento = (byte \*)&activMovimiento;  const uint8\_t payloadSize = 9;  uint8\_t buf\_str[payloadSize];  buf\_str[0] = float\_volt[0];  buf\_str[1] = float\_volt[1];  buf\_str[2] = float\_volt[2];  buf\_str[3] = float\_volt[3];  buf\_str[4] = float\_movimiento[0];  buf\_str[5] = float\_movimiento[1];  buf\_str[6] = float\_movimiento[2];  buf\_str[7] = float\_movimiento[3];  uint8\_t \*sendData = buf\_str;  Send\_Pload(buf\_str, payloadSize);  } | |

|  |  |
| --- | --- |
| Función: | void Send\_Pload () |
| Descripción: |  |
| void Send\_Pload(uint8\_t \*sendData, const uint8\_t len) {  recvMsg \*RecvMsg;  RecvMsg = (recvMsg \*)malloc(sizeof(recvMsg));  Isigfox->sendPayload(sendData, len, 0, RecvMsg);  for (int i = 0; i < RecvMsg->len; i++) {  Serial.print(RecvMsg->inData[i]);  }  Serial.println("");  free(RecvMsg);  } | |

|  |  |
| --- | --- |
| Función: | void activar\_ibutton () |
| Descripción: |  |
| void activar\_ibutton(){  digitalWrite(relay,HIGH);  Serial.println("ACTIVADO");  delay(5000);  } | |

|  |  |
| --- | --- |
| Función: | void desactivar\_ibutton () |
| Descripción: |  |
| void desactivar\_ibutton(){  digitalWrite(relay,LOW);  Serial.println("DESACTIVADO");  delay(1000);  } | |

# Código fuente java

|  |  |
| --- | --- |
| Archivo: | Bienvenida.java |
| Descripción: |  |
| public class Bienvenida extends AppCompatActivity {  Handler handler;  ImageView img;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_bienvenida);  img = findViewById(R.id.logo\_portada);  img.animate().alpha(4000).setDuration(0);  handler = new Handler();  handler.postDelayed(new Runnable() {  @Override  public void run() {  Intent log = new Intent(Bienvenida.this, Login.class);  startActivity(log);  finish();  }  },4000);  }  } | |

|  |  |
| --- | --- |
| Archivo: | Login.java |
| Descripción: |  |
| public class Login extends AppCompatActivity implements View.OnClickListener{  Button btnRegistro, btnLogin;  SignInButton signInButton;  GoogleSignInClient mGoogleSignInClient;  FirebaseAuth firebaseAuth;  private ProgressDialog progressDialog;  private FirebaseFirestore db;  EditText edtextmail, edtextpass;  int RC\_SIGN\_IN = 123;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_login);  //Declaracion de variables  autologin();  edtextmail = findViewById(R.id.Edtxt\_login);  edtextpass = findViewById(R.id.Edtxt\_pass);  signInButton = findViewById(R.id.signButton);  firebaseAuth = FirebaseAuth.getInstance();  db = FirebaseFirestore.getInstance();  btnRegistro = findViewById(R.id.btnRegistro);  btnLogin = findViewById(R.id.btnLogin);  progressDialog = new ProgressDialog(this);  GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT\_SIGN\_IN)  .requestIdToken(getString(R.string.default\_web\_client\_id))  .requestEmail()  .build();  mGoogleSignInClient = GoogleSignIn.getClient(this,gso);  edtextpass.setOnEditorActionListener(new TextView.OnEditorActionListener() {  @Override  public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {  if(actionId == EditorInfo.IME\_ACTION\_SEND){  edtextpass.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  logueado();  }  });  }  return false;  }  });  signInButton.setOnClickListener(this);  btnRegistro.setOnClickListener(this);  btnLogin.setOnClickListener(this);  }  @Override  public void onClick(View view) {  switch (view.getId()){  case R.id.signButton:  if(verConexioInternet()){  singIn();  }  break;  case R.id.btnRegistro:  registro();  break;  case R.id.btnLogin:  if(verConexioInternet()){  logueado();  }  break;  }  }  private void singIn() {  Intent signInIntent = mGoogleSignInClient.getSignInIntent();  startActivityForResult(signInIntent, RC\_SIGN\_IN);  }  @Override  protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  super.onActivityResult(requestCode, resultCode, data);  if(requestCode == RC\_SIGN\_IN){  Task<GoogleSignInAccount> task = GoogleSignIn.getSignedInAccountFromIntent(data);  try {  GoogleSignInAccount account = task.getResult(ApiException.class);  if (account != null) FirebaseGoogleAuth(account);  Toast.makeText(Login.this,"Inicio de sesion saisfactoria",Toast.LENGTH\_SHORT).show();  } catch (ApiException e) {  Log.w("Login", "signInResult:failed code=" + e.getStatusCode());  }  }  }  private void FirebaseGoogleAuth(GoogleSignInAccount account){  AuthCredential authCredential = GoogleAuthProvider.getCredential(account.getIdToken(),null);  firebaseAuth.signInWithCredential(authCredential).addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {  @Override  public void onComplete(@NonNull Task<AuthResult> task) {  if(task.isSuccessful()){  FirebaseUser user = firebaseAuth.getCurrentUser();  updateUI(user);  }else{  System.out.println("Error");  updateUI(null);  }  }  });  }  private void updateUI(final FirebaseUser user){  if (user != null) {  final Map<String, Object> map\_user = new HashMap<>();  map\_user.put("dispvinculados", Collections.emptyList());  map\_user.put("nombre", user.getDisplayName());  map\_user.put("apellido", "");  if(user.getPhoneNumber()==null || user.getPhoneNumber().isEmpty()){  map\_user.put("celular","");  }else{  map\_user.put("celular",user.getPhoneNumber());  }  progressDialog.setMessage("Iniciando sesion"); //Muestra un progressDialog con ese mensaje  progressDialog.show();  db.collection("usuarios").document(Objects.requireNonNull(user.getEmail())).set(map\_user).addOnCompleteListener(new OnCompleteListener<Void>() {  @Override  public void onComplete(@NonNull Task<Void> task) {  if(task.isSuccessful()){  Preferences.SaveCredenciales(Login.this,user.getEmail(),"email"); //guarda un String con llave user  Preferences.SaveCredenciales(Login.this,user.getDisplayName(),"nom\_usuario");//guarda un String con llave nombre  Toast.makeText(Login.this,"Inicio de Sesion Satisfactorio",Toast.LENGTH\_SHORT).show();  irSistema();  progressDialog.dismiss(); // termina el progressDialog  }else{  Toast.makeText(Login.this,"Fallo en el registro",Toast.LENGTH\_SHORT).show();  }  }  });  } else {  System.out.println("Fallo al registrarse");  }  }  public void registro(){  Intent regis = new Intent(Login.this,Registro.class);  startActivity(regis);  }  public void logueado(){  final String email = edtextmail.getText().toString().trim();  final String pass = edtextpass.getText().toString().trim();  if(TextUtils.isEmpty(email)|| TextUtils.isEmpty(pass)){  if(TextUtils.isEmpty(email)){  edtextmail.setError("Este campo no puede estar vacio");  }  if(TextUtils.isEmpty(pass)){  edtextpass.setError("Este campo no puede estar vacio");  }  }else{  progressDialog.setMessage("Iniciando sesion"); //Muestra un progressDialog con ese mensaje  progressDialog.show();  firebaseAuth.signInWithEmailAndPassword(email,pass). // inicia sesion con correo y contraseña  addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {  @Override  public void onComplete(@NonNull Task<AuthResult> task) {  if(task.isSuccessful()){  if(Objects.requireNonNull(firebaseAuth.getCurrentUser()).isEmailVerified()){ // permite ingresar si el correo ha sido verificado  FirebaseUser firebaseUser = firebaseAuth.getCurrentUser(); // obtengo los parametros del usuario del Firebase Authentication  final String usdisplay = Objects.requireNonNull(firebaseUser.getDisplayName()); // obtengo el nombre a mostrar que esta en los parametros del usuario  final DocumentReference docRef = db.collection("usuarios").document(email);  docRef.get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {  @Override  public void onComplete(@NonNull Task<DocumentSnapshot> task) {  if(task.isSuccessful()){  DocumentSnapshot document = task.getResult();  assert document != null;  if (document.exists()) {  Preferences.SaveCredenciales(Login.this,email,"email"); //guarda un String con llave user  Preferences.SaveCredenciales(Login.this,usdisplay,"nom\_usuario");//guarda un String con llave nombre  Toast.makeText(Login.this,"Se ha ingresado correctamente",Toast.LENGTH\_LONG).show();  irSistema();  }else {  Log.d("ERROR", "Error a obtener el documento");  }  }else{  Log.d("ERROR", "fallas al obtener con ", task.getException());  }  }  });  }else{  Toast.makeText(Login.this,"Por favor verifque su correo electronico",Toast.LENGTH\_LONG).show();  }  }else{  Toast.makeText(Login.this,"Correo y/o contraseña incorrecta",Toast.LENGTH\_LONG).show();  }  progressDialog.dismiss(); // termina el progressDialog  }  });  }  }  public void autologin(){  String correo = Preferences.ObtenerCredenciales(this,"email", (String) null); //obtiene String de las credenciales guardadas  String nom\_usuario = Preferences.ObtenerCredenciales(this,"nom\_usuario", (String) null); //obtiene String de las credenciales guardadas  if(correo!=null && nom\_usuario!= null){  irSistema(); // ingresa al sistema si existen valores guardados en el SharedPreferences  }  }  public void irSistema(){  Intent i = new Intent(Login.this,MainActivity.class);  startActivity(i);  finish(); // destruye la actividad  }  public boolean verConexioInternet(){  try {  ConnectivityManager con = (ConnectivityManager) this.getSystemService(Context.CONNECTIVITY\_SERVICE);  assert con != null;  NetworkInfo networkInfo = con.getActiveNetworkInfo();  if (networkInfo != null && networkInfo.isConnected()) {  return true;  }else{  Toast.makeText(Login.this,"Verifique su conexion de Internet",Toast.LENGTH\_SHORT).show();  return false;  }  }catch (NullPointerException n){  return false;  }  }  } | |

|  |  |
| --- | --- |
| Archivo: | MainActivity.java |
| Descripción: |  |
| public class MainActivity extends AppCompatActivity {  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);  BottomNavigationView navView = findViewById(R.id.nav\_view);  // Passing each menu ID as a set of Ids because each  // menu should be considered as top level destinations.  AppBarConfiguration appBarConfiguration = new AppBarConfiguration.Builder(  R.id.navigation\_home, R.id.navigation\_dashboard, R.id.navigation\_notifications)  .build();  NavController navController = Navigation.findNavController(this, R.id.nav\_host\_fragment);  NavigationUI.setupActionBarWithNavController(this, navController, appBarConfiguration);  NavigationUI.setupWithNavController(navView, navController);  }  } | |

|  |  |
| --- | --- |
| Archivo: | Preferences.java |
| Descripción: |  |
| public class Preferences {  public static final String PREFERENCES\_CREDENCIALES = "credenciales.usuario";  public static void SaveCredenciales(Context c, String s, String key){  SharedPreferences preferences = c.getSharedPreferences(PREFERENCES\_CREDENCIALES,c.MODE\_PRIVATE);  preferences.edit().putString(key,s).apply();  }  public static String ObtenerCredenciales(Context c, String key,String defecto){  SharedPreferences preferences = c.getSharedPreferences(PREFERENCES\_CREDENCIALES,c.MODE\_PRIVATE);  return preferences.getString(key,defecto);  }  public static void SaveCredenciales(Context c, boolean b, String key){  SharedPreferences preferences = c.getSharedPreferences(PREFERENCES\_CREDENCIALES,c.MODE\_PRIVATE);  preferences.edit().putBoolean(key,b).apply();  }  public static Boolean ObtenerCredenciales(Context c, String key, Boolean defecto){  SharedPreferences preferences = c.getSharedPreferences(PREFERENCES\_CREDENCIALES,c.MODE\_PRIVATE);  return preferences.getBoolean(key,defecto);  }  public static void save(Context c,Usuario usuario,String key) {  SharedPreferences.Editor edit = c.getSharedPreferences(PREFERENCES\_CREDENCIALES,c.MODE\_PRIVATE).edit();  String json = new Gson().toJson(usuario);  edit.putString(key,json);  edit.apply();  }  public static Usuario getUsuario(Context c,String key){  String json = c.getSharedPreferences(PREFERENCES\_CREDENCIALES,c.MODE\_PRIVATE).getString(key,null);  if(json == null){  return null;  }  return new Gson().fromJson(json, Usuario.class);  }  public static void LimpiarCredenciales(Context c){  SharedPreferences preferences = c.getSharedPreferences(PREFERENCES\_CREDENCIALES,c.MODE\_PRIVATE);  preferences.edit().clear().apply();  }  } | |

|  |  |
| --- | --- |
| Archivo: | Registro.java |
| Descripción: |  |
| public class Registro extends AppCompatActivity {  TextView txtpasslong,txtpassdig,txtpassmayus,txtpassesp;  EditText ednom, edapell, edmail, edpass, edpassconf, edphone;  ProgressDialog progressDialog;  private FirebaseAuth firebaseAuth;  private FirebaseFirestore db;  private static final ArrayList<String> correo\_permitidos = new ArrayList<>();  private static final Pattern PASSWORD\_PATTERN =  Pattern.compile("^" +  "(?=.\*[0-9])" + //al menos 1 digito  //"(?=.\*[a-z])" + //al menos 1 letra minuscula  "(?=.\*[A-Z])" + //al menos 1 letra mayuscula  "(?=.\*[a-zA-Z])" + //cualquier letra  //"(?=.\*[@#$%^&+=])" + //al menos 1 caracter especial  "(?=\\S+$)" + //sin espacios  ".{7,}" + //al menos 7 caracteres  "$");  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_registro);  //inicializar variables  ednom = findViewById(R.id.EdTnombre\_registro);  edapell = findViewById(R.id.EdTapellido\_registro);  edmail = findViewById(R.id.EdTemail\_registro);  edpass = findViewById(R.id.EdTpass\_registro);  edpassconf = findViewById(R.id.EdTpassconfir\_registro);  edphone = findViewById(R.id.Edcelular);  txtpasslong = findViewById(R.id.Txtpass\_longitud);  txtpassmayus = findViewById(R.id.Txtpass\_mayuscula);  txtpassesp = findViewById(R.id.Txtpass\_espacio);  txtpassdig = findViewById(R.id.Txtpass\_digito);  //agregando dominio permitido  correo\_permitidos.add("gmail.com");  correo\_permitidos.add("outlook.com");  correo\_permitidos.add("outlook.es");  correo\_permitidos.add("hotmail.com");  correo\_permitidos.add("hotmail.es");  correo\_permitidos.add("yahoo.com");  progressDialog = new ProgressDialog(this);  firebaseAuth = FirebaseAuth.getInstance();  db = FirebaseFirestore.getInstance();  //Muestra al usuario si el pass cumple con los parametros especificos  edpass.addTextChangedListener(new TextWatcher() {  @Override  public void beforeTextChanged(CharSequence s, int start, int count, int after) {  }  @Override  public void onTextChanged(CharSequence s, int start, int before, int count) {  }  @Override  public void afterTextChanged(Editable s) {  if(!PASSWORD\_PATTERN.matcher(s).matches()){  txtpasslong.setVisibility(View.VISIBLE);  txtpassesp.setVisibility(View.VISIBLE);  txtpassdig.setVisibility(View.VISIBLE);  txtpassmayus.setVisibility(View.VISIBLE);  txtpasslong.setCompoundDrawablesWithIntrinsicBounds(R.drawable.ic\_cancel\_red\_24dp,0,0,0);  txtpassesp.setCompoundDrawablesWithIntrinsicBounds(R.drawable.ic\_cancel\_red\_24dp,0,0,0);  txtpassmayus.setCompoundDrawablesWithIntrinsicBounds(R.drawable.ic\_cancel\_red\_24dp,0,0,0);  }else{  txtpassdig.setCompoundDrawablesWithIntrinsicBounds(R.drawable.ic\_check\_green\_24dp,0,0,0);  txtpasslong.setCompoundDrawablesWithIntrinsicBounds(R.drawable.ic\_check\_green\_24dp,0,0,0);  txtpassesp.setCompoundDrawablesWithIntrinsicBounds(R.drawable.ic\_check\_green\_24dp,0,0,0);  txtpassmayus.setCompoundDrawablesWithIntrinsicBounds(R.drawable.ic\_check\_green\_24dp,0,0,0);  }  }  });  edpassconf.addTextChangedListener(new TextWatcher() {  @Override  public void beforeTextChanged(CharSequence s, int start, int count, int after) {  if(PASSWORD\_PATTERN.matcher(edpass.getText().toString()).matches()){  txtpasslong.setVisibility(View.GONE);  txtpassesp.setVisibility(View.GONE);  txtpassdig.setVisibility(View.GONE);  txtpassmayus.setVisibility(View.GONE);  }else{  txtpasslong.setVisibility(View.VISIBLE);  txtpassesp.setVisibility(View.VISIBLE);  txtpassdig.setVisibility(View.VISIBLE);  txtpassmayus.setVisibility(View.VISIBLE);  }  }  @Override  public void onTextChanged(CharSequence s, int start, int before, int count) {  }  @Override  public void afterTextChanged(Editable s) {  }  });  }  private boolean validar\_mail(String email){  if(TextUtils.isEmpty(email)){  edmail.setError("Este campo no puede estar vacio");  return false;  }else{  if(!Patterns.EMAIL\_ADDRESS.matcher(email).matches()){  edmail.setError("Correo invalido");  return false;  }else{  String[] tempdom = email.split("@");  if(tempdom.length<3){  String dominio = tempdom[1];  if(correo\_permitidos.contains(dominio)){  return true;  }else{  edmail.setError("Correo invalido");  return false;  }  }else{  edmail.setError("Correo invalido");  return false;  }  }  }  }  private boolean validar\_pass(String pass){  String pass\_conf = edpassconf.getText().toString().trim();  if(TextUtils.isEmpty(pass)|| TextUtils.isEmpty(pass\_conf)){  edpass.setError("Este campo no puede estar vacio");  edpassconf.setError("Este campo no puede estar vacio");  return false;  }else{  if(!PASSWORD\_PATTERN.matcher(pass).matches()){  edpass.setError("Contraseña invalida");  return false;  }else{  if(!pass.equals(pass\_conf)){  edpassconf.setError("Contraseña mal escrita");  return false;  }else{  return true;  }  }  }  }  private boolean validar\_vacios(){  String nombre = ednom.getText().toString().trim();  String apellido = edapell.getText().toString().trim();  String email = edmail.getText().toString().trim();  String pass = edpass.getText().toString().trim();  String pass\_conf = edpassconf.getText().toString().trim();  String phone = edphone.getText().toString().trim();  if(TextUtils.isEmpty(nombre) || TextUtils.isEmpty(apellido)||  TextUtils.isEmpty(email)|| TextUtils.isEmpty(pass)||  TextUtils.isEmpty(pass\_conf) || TextUtils.isEmpty(phone)) {  if(TextUtils.isEmpty(nombre)){  ednom.setError("Este campo no puede estar vacio");  }  if(TextUtils.isEmpty(apellido)){  edapell.setError("Este campo no puede estar vacio");  }  if(TextUtils.isEmpty(email)){  edmail.setError("Este campo no puede estar vacio");  }  if(TextUtils.isEmpty(pass)){  edpass.setError("Este campo no puede estar vacio");  }  if(TextUtils.isEmpty(pass\_conf)){  edpassconf.setError("Este campo no puede estar vacio");  }  if(TextUtils.isEmpty(phone)){  edphone.setError("Este campo no puede estar vacio");  }  return false;  }else{  return true;  }  }  public void registro(final View view) {  final String email = edmail.getText().toString().trim();  String pass = edpass.getText().toString().trim();  final String nombre = ednom.getText().toString().trim();  final String apellido = edapell.getText().toString().trim();  String phone = edphone.getText().toString().trim();  if (validar\_vacios() && validar\_mail(email) && validar\_pass(pass)) {  final Map<String, Object> map\_user = new HashMap<>();  map\_user.put("dispvinculados", Collections.emptyList());  map\_user.put("nombre", nombre);  map\_user.put("apellido", apellido);  map\_user.put("celular",phone);  try {  ConnectivityManager con = (ConnectivityManager) this.getSystemService(Context.CONNECTIVITY\_SERVICE);  assert con != null;  NetworkInfo networkInfo = con.getActiveNetworkInfo();  if (networkInfo != null && networkInfo.isConnected()) {  progressDialog.setMessage("Realizando registro en linea");  progressDialog.show();  firebaseAuth.createUserWithEmailAndPassword(email,pass)  .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {  @Override  public void onComplete(@NonNull Task<AuthResult> task) {  if(task.isSuccessful()){  Objects.requireNonNull(firebaseAuth.getCurrentUser()).sendEmailVerification().addOnCompleteListener(new OnCompleteListener<Void>() {  @Override  public void onComplete(@NonNull Task<Void> task) {  if(task.isSuccessful()){  UserProfileChangeRequest profileChangeRequest = new UserProfileChangeRequest.Builder().setDisplayName(nombre+" "+apellido).build();  firebaseAuth.getCurrentUser().updateProfile(profileChangeRequest).addOnCompleteListener(new OnCompleteListener<Void>() {  @Override  public void onComplete(@NonNull Task<Void> task) {  if(task.isSuccessful()){  db.collection("usuarios").document(email).set(map\_user).addOnCompleteListener(new OnCompleteListener<Void>() {  @Override  public void onComplete(@NonNull Task<Void> task) {  if(task.isSuccessful()){  Toast.makeText(Registro.this,"Se ha registrado correctamente. Por favor verificar correo de verficacion",Toast.LENGTH\_LONG).show();  InputMethodManager imm = (InputMethodManager) getSystemService(INPUT\_METHOD\_SERVICE);  Objects.requireNonNull(imm).hideSoftInputFromWindow(view.getWindowToken(), 0);  Intent i = new Intent(Registro.this,Login.class);  startActivity(i);  FirebaseAuth.getInstance().signOut(); //cerrar sesion  finish();  }else{  Toast.makeText(Registro.this,"No se pudo registrar correctamente",Toast.LENGTH\_LONG).show();  }  }  });  }else{  Toast.makeText(Registro.this,"No se pudo registrar correctamente",Toast.LENGTH\_LONG).show();  }  }  });  }else{  Toast.makeText(Registro.this,"No se pudo registrar correctamente",Toast.LENGTH\_LONG).show();  }  }  });  }else{  if(task.getException() instanceof FirebaseAuthUserCollisionException){  edmail.setError("Correo ya registrado");  }else{  Toast.makeText(Registro.this,"No se pudo registrar correctamente",Toast.LENGTH\_LONG).show();  }  }  progressDialog.dismiss();  }  });  }else {  Toast.makeText(this, "No se pudo conectar, verifique" +  " el acceso a Internet e Intente de nuevamente", Toast.LENGTH\_LONG).show();  }  }catch (NullPointerException n){  Log.w("Error",n.toString());  }  }  }  } | |

|  |  |
| --- | --- |
| Archivo: | Usuario.java |
| Descripción: |  |
| public class Usuario {  @SerializedName("mochilas")  final ArrayList<Mochila> mochilas;  @SerializedName("nom\_usuario")  String nom\_usuario;  @SerializedName("correo")  String correo;  public Usuario() {  mochilas = new ArrayList<>();  this.correo = "";  }  public ArrayList<Mochila> getMochilas() {  return mochilas;  }  public void setMochilas(Mochila mochila) {  this.mochilas.add(mochila);  }  public String getCorreo() {  return correo;  }  public void setCorreo(String correo) {  this.correo = correo;  }  public String getNom\_usuario() {  return nom\_usuario;  }  public void setNom\_usuario(String nom\_usuario) {  this.nom\_usuario = nom\_usuario;  }  public int cant\_mochilas(){  return this.mochilas.size();  }  public String ObteneridMochila(String tipo, String id\_alias){  if(tipo.equals("alias")){  for(Mochila mochila: getMochilas()){  if(mochila.getAlias().equals(id\_alias)){  return mochila.getId\_dispositivo();  }  }  }  return id\_alias;  }  public void setModoMochila(int position,String modo){  this.mochilas.get(position).setEncd\_apagado(modo);  }  public ArrayList<Mochila> obtenerMochilasEnc(){  ArrayList<Mochila> mochilastemp = new ArrayList<>();  for(Mochila mochila: this.mochilas){  if(mochila.getEncd\_apagado().equalsIgnoreCase("on")){  mochilastemp.add(mochila);  }  }  return mochilastemp;  }  } | |

|  |  |
| --- | --- |
| Archivo: | MapaFragment.java |
| Descripción: |  |
| public class MapaFragment extends Fragment implements OnMapReadyCallback, GoogleApiClient.ConnectionCallbacks,  GoogleApiClient.OnConnectionFailedListener, LocationListener {  GoogleMap mapaGoogle;  DatabaseReference db\_reference;  private ArrayList<Marker> temprealTimeMarker = new ArrayList<>();  private ArrayList<Marker> realTimeMarker = new ArrayList<>();  LocationManager manejadorLocalizacion;  public LatLng UBICACIONU = new LatLng(-0.947334, -80.732324);  double milongitudeGPS, milatitudeGPS;  private GoogleApiClient clienteGoogleApi;  private Location ultimaLocalizacion;  public static final int REQUEST\_LOCATION = 1;  public static final int REQUEST\_CHECK\_SETTINGS = 2;  public static final long INTERVALO\_ACTUALIZACION = 2000;  public static final long INTERVALO\_ACTUALIZACION\_RAPIDA = INTERVALO\_ACTUALIZACION / 2;  private LocationRequest requisitoLocalizacion;  private LocationSettingsRequest configresquisitoLocalizacion;  FloatingActionButton floatingSearch;  String distanciaMaxima;  Usuario usuario;  @Override  public void onCreate(@Nullable Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  }  public View onCreateView(@NonNull LayoutInflater inflater,  ViewGroup container, Bundle savedInstanceState) {  View root = inflater.inflate(R.layout.fragment\_mapa, container, false);  manejadorLocalizacion = (LocationManager) getActivity().getApplicationContext().getSystemService(Context.LOCATION\_SERVICE);  db\_reference = FirebaseDatabase.getInstance().getReference();  floatingSearch = root.findViewById(R.id.floatingSearch);  SupportMapFragment supportMapFragment = (SupportMapFragment) getChildFragmentManager().findFragmentById(R.id.map);  supportMapFragment.getMapAsync(this);  usuario = Preferences.getUsuario(getActivity().getApplicationContext(),"obusuario");  //inicializacion variables  clienteGoogleApi = new GoogleApiClient.Builder(getActivity())  .addConnectionCallbacks(this)  .addOnConnectionFailedListener(this)  .addApi(LocationServices.API)  .enableAutoManage(getActivity(), this)  .build();  requisitoLocalizacion = new LocationRequest()  .setInterval(INTERVALO\_ACTUALIZACION)  .setFastestInterval(INTERVALO\_ACTUALIZACION\_RAPIDA)  .setPriority(LocationRequest.PRIORITY\_HIGH\_ACCURACY);  LocationSettingsRequest.Builder builder = new LocationSettingsRequest.Builder()  .addLocationRequest(requisitoLocalizacion)  .setAlwaysShow(true);  configresquisitoLocalizacion = builder.build();  final PendingResult<LocationSettingsResult> result = LocationServices.SettingsApi.checkLocationSettings(  clienteGoogleApi, builder.build()  );  result.setResultCallback(new ResultCallback<LocationSettingsResult>() {  @Override  public void onResult(@NonNull LocationSettingsResult result) {  Status status = result.getStatus();  switch (status.getStatusCode()) {  case LocationSettingsStatusCodes.SUCCESS:  System.out.println("Los ajustes de ubicación satisfacen la configuración.");  processLastLocation();  break;  case LocationSettingsStatusCodes.RESOLUTION\_REQUIRED:  try {  System.out.println("Los ajustes de ubicación no satisfacen la configuración. " +  "Se mostrará un diálogo de ayuda.");  status.startResolutionForResult(  getActivity(),  REQUEST\_CHECK\_SETTINGS);  } catch (IntentSender.SendIntentException e) {  System.out.println("El Intent del diálogo no funcionó.");  // Sin operaciones  }  break;  case LocationSettingsStatusCodes.SETTINGS\_CHANGE\_UNAVAILABLE:  System.out.println("Los ajustes de ubicación no son apropiados.");  break;  }  }  });  floatingSearch.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  AlertDialog.Builder dialogo = new AlertDialog.Builder(getActivity());  dialogo.setTitle("Establezca la distancia maxima en Km");  final EditText edtxDistancia = new EditText(getActivity());  edtxDistancia.setInputType(InputType.TYPE\_CLASS\_NUMBER);  dialogo.setView(edtxDistancia);  dialogo.setPositiveButton("BUSCAR", new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface dialogInterface, int i) {  if(TextUtils.isEmpty(edtxDistancia.getText().toString().trim())){  edtxDistancia.setError("Este campo no puede estar vacio");  }else{  mapaGoogle.clear();  LatLng milatLng = new LatLng(milatitudeGPS, milongitudeGPS);  distanciaMaxima = edtxDistancia.getText().toString().trim();  for(Mochila mochila: usuario.getMochilas()){  if(mochila.getEncd\_apagado().equalsIgnoreCase("on")){  LatLng LatLng\_dispositivo = new LatLng(mochila.getLatitud(),mochila.getLongitud());  //realTimeMarker.get(indice).remove();  MarkerOptions markerOptions = new MarkerOptions();  double distancia\_ub\_dispo = CalculationByDistance(milatLng,LatLng\_dispositivo);  if(Double.parseDouble(distanciaMaxima)>=distancia\_ub\_dispo){  markerOptions.icon(BitmapDescriptorFactory.fromResource(R.mipmap.mochila\_green)).anchor(0.0f,0.5f);  mochila.setRango("aceptable");  }else{  markerOptions.icon(BitmapDescriptorFactory.fromResource(R.mipmap.mochila\_red)).anchor(0.0f,0.5f);  mochila.setRango("fuera");  }  Preferences.save(getActivity().getApplicationContext(),usuario,"obusuario");  markerOptions.position(LatLng\_dispositivo);  mapaGoogle.addMarker(markerOptions);  }  }  }  }  });  dialogo.setNegativeButton("CANCELAR",null);  dialogo.show();  }  });  return root;  }  @Override  public void onMapReady(GoogleMap googleMap) {  mapaGoogle = googleMap;  if (ActivityCompat.checkSelfPermission(getContext(), Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED &&  ActivityCompat.checkSelfPermission(getContext(), Manifest.permission.ACCESS\_COARSE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {  return;  }  mapaGoogle.setMyLocationEnabled(true);  //UiSettings settings = mapaGoogle.getUiSettings();  //settings.setZoomControlsEnabled(true);  //setMarkerDragListener(mapaGoogle);  }  public void obtenerMochila(DatabaseReference databaseReference, final Mochila mochila){  databaseReference.addValueEventListener(new ValueEventListener() {  @Override  public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  try {  String latitud = String.valueOf(dataSnapshot.child("ubicacion").child("latitud").getValue());  String longitud = String.valueOf(dataSnapshot.child("ubicacion").child("longitud").getValue());  if(!latitud.isEmpty() && !longitud.isEmpty()){  if(Double.parseDouble(latitud)!=0.0&&Double.parseDouble(longitud)!=0.0){  mochila.setLatitud(Double.parseDouble(latitud));  mochila.setLongitud(Double.parseDouble(longitud));  MarkerOptions markerOptions = new MarkerOptions();  if(mochila.getRango().equalsIgnoreCase("neutral")){  markerOptions.icon(BitmapDescriptorFactory.fromResource(R.mipmap.mochila\_black)).anchor(0.0f,1.0f);  }else if(mochila.getRango().equalsIgnoreCase("fuera")){  markerOptions.icon(BitmapDescriptorFactory.fromResource(R.mipmap.mochila\_red)).anchor(0.0f,1.0f);  }else{  markerOptions.icon(BitmapDescriptorFactory.fromResource(R.mipmap.mochila\_green)).anchor(0.0f,1.0f);  }  Preferences.save(getActivity().getApplicationContext(),usuario,"obusuario");  markerOptions.position(new LatLng(Double.parseDouble(latitud),Double.parseDouble(longitud)));  temprealTimeMarker.add(mapaGoogle.addMarker(markerOptions));  }else{  Toast.makeText(getActivity(),"Dispositivo "+mochila.getId\_dispositivo()+" fuera de alcance para enviar datos",Toast.LENGTH\_SHORT).show();  }  }else{  Toast.makeText(getActivity(),"Active el dispositivo en un rango permitido",Toast.LENGTH\_SHORT).show();  }  }catch (NullPointerException n){  Toast.makeText(getActivity(),"Error al obtener la ubicacion del dispositivo",Toast.LENGTH\_SHORT).show();  }  }  @Override  public void onCancelled(@NonNull DatabaseError databaseError) {  System.out.println(databaseError.toException());  }  });  }  @Override  public void onStart() {  super.onStart();  clienteGoogleApi.connect();  }  @Override  public void onStop() {  super.onStop();  clienteGoogleApi.disconnect();  if(usuario!=null){  for(Mochila mochila: usuario.getMochilas()){  mochila.setRango("neutral");  }  Preferences.save(getActivity().getApplicationContext(),usuario,"obusuario");  }  }  @Override  public void onPause() {  super.onPause();  if (clienteGoogleApi.isConnected()) {  stopLocationUpdates();  }  clienteGoogleApi.stopAutoManage(getActivity());  clienteGoogleApi.disconnect();  if(usuario!=null){  for(Mochila mochila: usuario.getMochilas()){  mochila.setRango("neutral");  }  Preferences.save(getActivity().getApplicationContext(),usuario,"obusuario");  }  }  @Override  public void onResume() {  super.onResume();  if (clienteGoogleApi.isConnected()) {  startLocationUpdates();  }  }  @Override  public void onConnected(@Nullable Bundle bundle) {  if(verConexioInternet()){  // Obtenemos la última ubicación al ser la primera vez  processLastLocation();  // Iniciamos las actualizaciones de ubicación  startLocationUpdates();  }  }  private void startLocationUpdates() {  if (isLocationPermissionGranted()) {  if (ActivityCompat.checkSelfPermission(getContext(), android.Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {  return;  }  LocationServices.FusedLocationApi.requestLocationUpdates(  clienteGoogleApi, requisitoLocalizacion, this);  } else {  manageDeniedPermission();  }  }  private void stopLocationUpdates() {  LocationServices.FusedLocationApi  .removeLocationUpdates(clienteGoogleApi, this);  }  @Override  public void onConnectionSuspended(int i) {  System.out.println("Conexión suspendida");  clienteGoogleApi.connect();  }  @Override  public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {  Toast.makeText(getActivity(),"Error de conexión con el código:" + connectionResult.getErrorCode(), Toast.LENGTH\_LONG).show();  }  //Procesando la ultima ubicacion  private void processLastLocation() {  getLastLocation();  if (ultimaLocalizacion != null) {  updateLocationUI();  }  }  private void getLastLocation() {  if (isLocationPermissionGranted()) {  if (ActivityCompat.checkSelfPermission(getContext(), android.Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {  }  System.out.println("ACEPTADO");  ultimaLocalizacion = LocationServices.FusedLocationApi.getLastLocation(clienteGoogleApi);  } else {  manageDeniedPermission();  }  }  private void updateLocationUI() {  mi\_marcador(ultimaLocalizacion.getLatitude(), ultimaLocalizacion.getLongitude());  }  private void mi\_marcador(double latitude, double longitude) {  mapaGoogle.clear(); //limpio el mapa  if(this.milatitudeGPS==0.0 && this.milongitudeGPS==0.0){  milatitudeGPS = latitude;  milongitudeGPS = longitude;  LatLng latLng = new LatLng(milatitudeGPS, milongitudeGPS);  mapaGoogle.moveCamera(CameraUpdateFactory.newLatLng(latLng));  mapaGoogle.addMarker(new MarkerOptions().position(latLng).title("Mi ubicacion"));  }  LatLng latLng = new LatLng(milatitudeGPS, milongitudeGPS);  mapaGoogle.addMarker(new MarkerOptions().position(latLng).title("Mi ubicacion"));  //mapaGoogle.addMarker(new MarkerOptions().icon(BitmapDescriptorFactory.fromResource(R.mipmap.ic\_logotipo)).anchor(0.0f,1.0f).position(latLng).title("Mi ubicacion")); //añade un marcador con un icono  for(Marker marker: realTimeMarker){  marker.remove();  }  if(usuario!=null){  if(usuario.getMochilas().size()!=0){  if(usuario.obtenerMochilasEnc().size()!=0){  for(Mochila mochila: usuario.getMochilas()){  if(mochila.getEncd\_apagado().equalsIgnoreCase("on")){  DatabaseReference db\_dispositivos = db\_reference.child("dispositivos").child(mochila.getId\_dispositivo());  obtenerMochila(db\_dispositivos,mochila);  }  }  }else{  Toast.makeText(getActivity(),"Active al menos un dispositivo",Toast.LENGTH\_SHORT).show();  }  }else{  Toast.makeText(getActivity(),"Dirijase a la seccion mochila para agregar un dispositivo",Toast.LENGTH\_SHORT).show();  }  }  realTimeMarker.clear();  realTimeMarker.addAll(temprealTimeMarker);  }  //Calculando distancia radial de un punto a otro punto  public double CalculationByDistance(LatLng StartP, LatLng EndP) {  int Radius = 6371;// radius of earth in Km  double lat1 = StartP.latitude;  double lat2 = EndP.latitude;  double lon1 = StartP.longitude;  double lon2 = EndP.longitude;  double dLat = Math.toRadians(lat2 - lat1);  double dLon = Math.toRadians(lon2 - lon1);  double a = Math.sin(dLat / 2) \* Math.sin(dLat / 2)  + Math.cos(Math.toRadians(lat1))  \* Math.cos(Math.toRadians(lat2)) \* Math.sin(dLon / 2)  \* Math.sin(dLon / 2);  double c = 2 \* Math.asin(Math.sqrt(a));  double valueResult = Radius \* c;  double km = valueResult / 1;  DecimalFormat newFormat = new DecimalFormat("####");  int kmInDec = Integer.valueOf(newFormat.format(km));  double meter = valueResult % 1000;  int meterInDec = Integer.valueOf(newFormat.format(meter));  Log.i("Radius Value", "" + valueResult + " KM " + kmInDec  + " Meter " + meterInDec);  return Radius \* c;  }  public void setMarkerDragListener(GoogleMap map) {  map.setOnMarkerDragListener(new GoogleMap.OnMarkerDragListener() {  @Override  public void onMarkerDragStart(Marker marker) {  }  @Override  public void onMarkerDrag(Marker marker) {  }  @Override  public void onMarkerDragEnd(Marker marker) {  LatLng p = marker.getPosition();  marker.setTitle("Mochila " + String.valueOf(CalculationByDistance(p, UBICACIONU) + " Km"));  }  });  }  //Permisos de CONEXION INTERNET - LOCALIZACIÓN  public boolean verConexioInternet() {  try {  ConnectivityManager con = (ConnectivityManager) getActivity().getSystemService(Context.CONNECTIVITY\_SERVICE);  assert con != null;  NetworkInfo networkInfo = con.getActiveNetworkInfo();  if (networkInfo != null && networkInfo.isConnected()) {  return true;  } else {  Toast.makeText(getActivity(), "Verifique su conexion de Internet", Toast.LENGTH\_SHORT).show();  return false;  }  } catch (NullPointerException n) {  return false;  }  }  //Verifica y pide permisos de localizacion  private boolean isLocationPermissionGranted() {  int permission = ActivityCompat.checkSelfPermission(  getActivity(),  Manifest.permission.ACCESS\_FINE\_LOCATION);  return permission == PackageManager.PERMISSION\_GRANTED;  }  private void manageDeniedPermission() {  if (ActivityCompat.shouldShowRequestPermissionRationale(getActivity(),  Manifest.permission.ACCESS\_FINE\_LOCATION)) {  // Aquí muestras confirmación explicativa al usuario  // por si rechazó los permisos anteriormente  } else {  requestPermissions(new String[]{Manifest.permission.ACCESS\_FINE\_LOCATION}, REQUEST\_LOCATION);  }  }  public void onRequestPermissionsResult(int requestCode,  String[] permissions,  int[] grantResults) {  if (requestCode == REQUEST\_LOCATION) {  if (grantResults.length == 1  && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {  if (ActivityCompat.checkSelfPermission(getActivity(), Manifest.permission.ACCESS\_FINE\_LOCATION) != PackageManager.PERMISSION\_GRANTED) {  return;  }  ultimaLocalizacion = LocationServices.FusedLocationApi.getLastLocation(clienteGoogleApi);  if (ultimaLocalizacion != null) {  mi\_marcador(ultimaLocalizacion.getLatitude(),ultimaLocalizacion.getLongitude());  } else {  Toast.makeText(getActivity(), "Ubicación no encontrada", Toast.LENGTH\_LONG).show();  }  } else {  Toast.makeText(getActivity(), "Permisos no otorgados", Toast.LENGTH\_LONG).show();  }  }  }  //Escucha los cambios de la localizacion, es decir actualiza la localizacion ultima actual  @Override  public void onLocationChanged(Location location) {  ultimaLocalizacion = location;  updateLocationUI();  }  } | |

|  |  |
| --- | --- |
| Archivo: | MapsActivity.java |
| Descripción: |  |
| public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {  private GoogleMap mMap;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_maps);  // Obtain the SupportMapFragment and get notified when the map is ready to be used.  SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()  .findFragmentById(R.id.map);  mapFragment.getMapAsync(this);  }  @Override  public void onMapReady(GoogleMap googleMap) {  mMap = googleMap;  }  } | |

|  |  |
| --- | --- |
| Archivo: | AdapterDatos.java |
| Descripción: |  |
| public class AdapterDatos extends RecyclerView.Adapter<AdapterDatos.ViewHolderDatos> {  ArrayList<String> listaDatos ;  Context context;  final Usuario usuario;  View mview;  FragmentActivity activity;  DatabaseReference db\_reference;  int indice = 1;  public AdapterDatos(ArrayList<String> listaDatos, Context context, FragmentActivity activity) {  this.listaDatos = listaDatos;  this.context = context;  this.activity=activity;  usuario = Preferences.getUsuario(context,"obusuario");  }  @NonNull  @Override  public ViewHolderDatos onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {  LayoutInflater layout = LayoutInflater.from(context);  View view = layout.inflate(R.layout.items\_list,parent,false);  return new ViewHolderDatos(view);  }  @Override  public void onBindViewHolder(@NonNull ViewHolderDatos holder, int position) {  holder.asignarDato(listaDatos.get(position));  }  @Override  public int getItemCount() {  return listaDatos.size();  }  public class ViewHolderDatos extends RecyclerView.ViewHolder {  TextView dato ;  Switch switchon\_off;  Button info;  public ViewHolderDatos(@NonNull final View itemView) {  super(itemView);  dato = (TextView) itemView.findViewById(R.id.idDato);  switchon\_off = (Switch) itemView.findViewById(R.id.switchOnOff);  info = (Button) itemView.findViewById(R.id.imgButtonInfo);  //Cambia el modo de la mochila  indice+=getAdapterPosition();  if(-1\*(indice)>=0){  if(usuario.getMochilas().get(-1\*(indice)).getEncd\_apagado().equals("on")){  switchon\_off.setChecked(true);  }else if(usuario.getMochilas().get(-1\*(indice)).getEncd\_apagado().equals("off")){  switchon\_off.setChecked(false);  }  }  switchon\_off.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  if(usuario!=null){  if(switchon\_off.isChecked()){  usuario.setModoMochila(getAdapterPosition(),"on");  }else{  usuario.setModoMochila(getAdapterPosition(),"off");  }  Preferences.save(view.getContext(),usuario,"obusuario");  }  }  });  info.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  db\_reference = FirebaseDatabase.getInstance().getReference();  AlertDialog.Builder mBuilder = new AlertDialog.Builder(context);  mview = activity.getLayoutInflater().inflate(R.layout.informacion\_dipositivo,null);  DatabaseReference db\_dispositivos = db\_reference.child("dispositivos")  .child(usuario.getMochilas().get(getAdapterPosition()).getId\_dispositivo());  TextView dispositivo = (TextView) mview.findViewById(R.id.info\_id);  final TextView bateria = (TextView) mview.findViewById(R.id.info\_bat);  TextView modo = (TextView) mview.findViewById(R.id.info\_modo);  dispositivo.setText(usuario.getMochilas().get(getAdapterPosition()).getId\_dispositivo());  db\_dispositivos.addValueEventListener(new ValueEventListener() {  @Override  public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  bateria.setText(String.valueOf(dataSnapshot.child("informacion").child("bateria").getValue())+"%");  }  @Override  public void onCancelled(@NonNull DatabaseError databaseError) {  }  });  modo.setText(usuario.getMochilas().get(getAdapterPosition()).getModo());  mBuilder.setView(mview);  AlertDialog dialog = mBuilder.create();  Objects.requireNonNull(dialog.getWindow()).setBackgroundDrawable(new ColorDrawable(Color.TRANSPARENT));  dialog.show();  }  });  }  public void asignarDato(String s) {  dato.setText(s);  }  }  } | |

|  |  |
| --- | --- |
| Archivo: | AddMochilas.java |
| Descripción: |  |
| public class AddMochilas extends AppCompatActivity {  private FirebaseFirestore db;  TextView aliasmochila,idmochila;  Usuario usuario;  ProgressDialog progressDialog;  DatabaseReference db\_reference;  boolean bandera = false;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_add\_mochilas);  db = FirebaseFirestore.getInstance();  db\_reference = FirebaseDatabase.getInstance().getReference();  aliasmochila = findViewById(R.id.EdtxtAliasMochila);  idmochila = findViewById(R.id.EdtxViewIdmochila);  progressDialog = new ProgressDialog(this);  usuario = Preferences.getUsuario(getApplicationContext(),"obusuario");  FloatingActionButton fabreturn = findViewById(R.id.fabReturn);  fabreturn.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  if(bandera){  Toast.makeText(AddMochilas.this,"Click de nuevo en el Boton Mochila para recargar",Toast.LENGTH\_LONG).show();  }  onBackPressed();  finish();  }  });  Button btnregistroMochila =(Button) findViewById(R.id.btnEnviarMochila);  btnregistroMochila.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  if(verConexioInternet(view)){  if(validar\_vacios()){  final String alias = aliasmochila.getText().toString().trim();  final String id = idmochila.getText().toString().trim();  final Map<String, Object> mochila\_cloud = new HashMap<>();  if(TextUtils.isEmpty(alias)){  mochila\_cloud.put("alias","");  }else{  mochila\_cloud.put("alias",alias);  }  if(usuario!=null){  mochila\_cloud.put("pertenece",usuario.getCorreo());  }  progressDialog.setMessage("Realizando registro en linea");  progressDialog.show();  db.collection("dispositivos").get().addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {  @Override  public void onComplete(@NonNull Task<QuerySnapshot> task) {  if(task.isSuccessful()){  ArrayList<String> ids\_totales = new ArrayList<String>();  for(QueryDocumentSnapshot documentSnapshots: task.getResult()){  System.out.println(documentSnapshots.getId());  ids\_totales.add(documentSnapshots.getId());  }  if(ids\_totales.contains(id)){  db.collection("dispositivos").document(id).get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {  @Override  public void onComplete(@NonNull Task<DocumentSnapshot> task) {  if(task.isSuccessful()){  DocumentSnapshot documentReference = task.getResult();  String perteneciente = (String) documentReference.get("pertenece");  if(TextUtils.isEmpty(perteneciente)){  db.collection("dispositivos").document(id).set(mochila\_cloud).addOnCompleteListener(new OnCompleteListener<Void>() {  @Override  public void onComplete(@NonNull Task<Void> task) {  if(task.isSuccessful()){  DatabaseReference mochila\_database = db\_reference.child("dispositivos").child(id);  mochila\_database.child("ubicacion").child("latitud").setValue("");  mochila\_database.child("ubicacion").child("longitud").setValue("");  mochila\_database.child("informacion").child("bateria").setValue(100);  bandera=true;  Toast.makeText(getApplicationContext(),"Dispositivo registrado correctamente",Toast.LENGTH\_SHORT).show();  }else{  bandera=false;  Toast.makeText(getApplicationContext(),"Hubo un error en el registro. Intentelo nuevamente",Toast.LENGTH\_SHORT).show();  }  }  });  }else{  bandera=false;  idmochila.setError("Ingrese Id no registrado");  Toast.makeText(getApplicationContext(),"Dispositivo ya registrado",Toast.LENGTH\_SHORT).show();  }  }else{  bandera=false;  Toast.makeText(getApplicationContext(),"Hubo un error en el registro. Intentelo nuevamente",Toast.LENGTH\_SHORT).show();  }  }  });  }else{  bandera=false;  Toast.makeText(getApplicationContext(),"Autentificacion de Dispositivo Incorrecta",Toast.LENGTH\_SHORT).show();  idmochila.setError("Id dispositivo incorrecto");  }  }else{  bandera=false;  Toast.makeText(getApplicationContext(),"Fallo de conexión en la nube",Toast.LENGTH\_SHORT).show();  }  progressDialog.dismiss();  }  });  }  }  }  });  }  public boolean verConexioInternet(View view){  try {  ConnectivityManager con = (ConnectivityManager) this.getSystemService(Context.CONNECTIVITY\_SERVICE);  assert con != null;  NetworkInfo networkInfo = con.getActiveNetworkInfo();  if (networkInfo != null && networkInfo.isConnected()) {  return true;  }else{  Snackbar.make(view, "Verifique su conexion de Internet", Snackbar.LENGTH\_LONG)  .setAction("Action", null).show();  return false;  }  }catch (NullPointerException n){  return false;  }  }  public boolean validar\_vacios(){  String id = idmochila.getText().toString().trim();  if(TextUtils.isEmpty(id)){  idmochila.setError("Este campo no puede estar vacio");  return false;  }  return true;  }  } | |

|  |  |
| --- | --- |
| Archivo: | Mochila.java |
| Descripción: |  |
| public class Mochila {  String id\_dispositivo,alias,encd\_apagado,bateria;  double latitud,longitud;  String modo,rango;  public Mochila(String id\_dispositivo) {  this.id\_dispositivo = id\_dispositivo;  this.alias="";  this.latitud=0.0;  this.longitud=0.0;  this.encd\_apagado="off";  this.bateria="";  this.modo = "REAL";  this.rango ="neutral";  }  public String getAlias() {  return alias;  }  public void setAlias(String alias) {  this.alias = alias;  }  public String getEncd\_apagado() {  return encd\_apagado;  }  public void setEncd\_apagado(String encd\_apagado) {  this.encd\_apagado = encd\_apagado;  }  public String getBateria() {  return bateria;  }  public void setBateria(String bateria) {  this.bateria = bateria;  }  public double getLatitud() {  return latitud;  }  public void setLatitud(double latitud) {  this.latitud = latitud;  }  public double getLongitud() {  return longitud;  }  public void setLongitud(double longitud) {  this.longitud = longitud;  }  public String getId\_dispositivo() {  return id\_dispositivo;  }  public void setId\_dispositivo(String id\_dispositivo) {  this.id\_dispositivo = id\_dispositivo;  }  public String getModo() {  return modo;  }  public void setModo(String modo) {  this.modo = modo;  }  public String getRango() {  return rango;  }  public void setRango(String rango) {  this.rango = rango;  }  } | |

|  |  |
| --- | --- |
| Archivo: | MochilaFragment.java |
| Descripción: |  |
| public class MochilaFragment extends Fragment {  private RecyclerView recyclerView;  Usuario usuario;  ArrayList<String> listaDatos;  AdapterDatos adapter;  private FirebaseFirestore db;  Button BtnAgregarmochilas,BtncambiarModo;  ViewGroup viewGroup;  ArrayList<String> ids;  public MochilaFragment() {  }  public void setListaDatos(String s) {  this.listaDatos.add(s);  }  public View onCreateView(@NonNull LayoutInflater inflater,  ViewGroup container, Bundle savedInstanceState) {  View root = inflater.inflate(R.layout.fragment\_mochila, container, false);  this.viewGroup = container;  recyclerView = (RecyclerView) root.findViewById(R.id.recyclerMochilas);  BtnAgregarmochilas = root.findViewById(R.id.btnAgregarMochila);  BtncambiarModo = root.findViewById(R.id.btnCambiarModo);  BtnAgregarmochilas.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  Intent intent = new Intent(getActivity(),AddMochilas.class);  startActivity(intent);  }  });  BtncambiarModo.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  //Se mueve a la actividad Modo Mochila  Intent intent = new Intent(getActivity(),ModoMochila.class);  startActivity(intent);  }  });  return root;  }  @Override  public void onCreate(@Nullable Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  usuario = Preferences.getUsuario(getActivity().getApplicationContext(),"obusuario");  ids = new ArrayList<>();  if(usuario==null){  usuario = new Usuario();  String email = Preferences.ObtenerCredenciales(getActivity().getApplicationContext(),"email","null");  if(!email.equals("null")){  usuario.setCorreo(email);  }  }else{  for(Mochila mochila: usuario.getMochilas()){  ids.add(mochila.getId\_dispositivo());  mochila.setRango("neutral");  }  Preferences.save(getActivity().getApplicationContext(),usuario,"obusuario");  }  listaDatos = new ArrayList<>();  if(verConexioInternet()){  db = FirebaseFirestore.getInstance();  db.collection("dispositivos").whereEqualTo("pertenece",usuario.getCorreo()).get().addOnCompleteListener(new OnCompleteListener<QuerySnapshot>() {  @Override  public void onComplete(@NonNull Task<QuerySnapshot> task) {  if(task.isSuccessful()){  for (QueryDocumentSnapshot document : task.getResult()) {  if(usuario.getMochilas().isEmpty()){  Mochila mochila = new Mochila(document.getId());  mochila.setAlias((String) document.get("alias"));  usuario.setMochilas(mochila);  }else{  if(!ids.isEmpty()){  if(!ids.contains(document.getId())){  Mochila mochila = new Mochila(document.getId());  mochila.setAlias((String) document.get("alias"));  usuario.setMochilas(mochila);  }  }  }  if(TextUtils.isEmpty((String) document.get("alias"))){  setListaDatos(document.getId());  }else{  setListaDatos((String) document.get("alias"));  }  }  }  Preferences.save(getActivity().getApplicationContext(),usuario,"obusuario");  adapter = new AdapterDatos(listaDatos,getContext(),getActivity());  recyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));  recyclerView.setAdapter(adapter);  if(usuario.getMochilas().size()==0){  Toast.makeText(getActivity(),"Registre al menos una mochila",Toast.LENGTH\_SHORT).show();  }  }  });  }  }  public boolean verConexioInternet(){  try {  ConnectivityManager con = (ConnectivityManager) getActivity().getSystemService(Context.CONNECTIVITY\_SERVICE);  assert con != null;  NetworkInfo networkInfo = con.getActiveNetworkInfo();  if (networkInfo != null && networkInfo.isConnected()) {  return true;  }else{  Toast.makeText(getActivity(),"Verifique su conexion de Internet",Toast.LENGTH\_SHORT).show();  return false;  }  }catch (NullPointerException n){  return false;  }  }  } | |

|  |  |
| --- | --- |
| Archivo: | ModoMochila.java |
| Descripción: |  |
| public class ModoMochila extends AppCompatActivity {  private static final int REQUEST\_ENABLE\_BT=0;  ArrayList<String> alias\_id = new ArrayList<>();  ArrayList<String> mochilas\_activas = new ArrayList<>();  Spinner mochilas\_spinner;  Button botonEncenderBt,botonApagarBt,btnconectarBlue,btnActivarModo,btnModifTelefono,btnEnviarBlue;  BluetoothAdapter mbluetoothAdapter = null;  Usuario usuario;  Switch estado\_modo;  ImageView bluetothView;  TextView nom\_bluetooth\_Ref,con\_nom\_bluetooh,txMensaje;  String nom\_bluetooth;  private static String adress\_bluetooth;  EditText edtxtelefonaenviar,edtxTiempo\_programado;  Handler bluetoothIn;  final int handlerState = 0;  private BluetoothSocket btSocket = null;  private StringBuilder DataStringIN = new StringBuilder();  private ConnectedThread MyConexionBT;  // Identificador unico de servicio - SPP UUID  private static final UUID BTMODULEUUID = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");  // String para la direccion MAC  @Override  protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  if (requestCode == REQUEST\_ENABLE\_BT) {  if (resultCode == RESULT\_OK) {  bluetothView.setImageResource(R.drawable.ic\_bluetooth\_on);  Toast.makeText(getApplicationContext(), "Bluetooth esta encendido", Toast.LENGTH\_SHORT).show();  } else {  Toast.makeText(getApplicationContext(), "No se pudo encender Bluetooth", Toast.LENGTH\_SHORT).show();  }  }  super.onActivityResult(requestCode, resultCode, data);  }  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_modo\_mochila);  estado\_modo = findViewById(R.id.switchModo);  bluetothView = findViewById(R.id.bluetoothView);  mochilas\_spinner = findViewById(R.id.mochilas\_act\_spinner);  botonEncenderBt = findViewById(R.id.btnModoEncender);  botonApagarBt = findViewById(R.id.btnModoApagar);  nom\_bluetooth\_Ref = findViewById(R.id.RefBluetooh);  con\_nom\_bluetooh = findViewById(R.id.NombBluetooh);  btnconectarBlue = findViewById(R.id.btnConectarBlue);  txMensaje = findViewById(R.id.textMensaje);  btnActivarModo = findViewById(R.id.btnActivarModo);  edtxtelefonaenviar = findViewById(R.id.EdtxTelefonEnviar);  btnModifTelefono = findViewById(R.id.btnModificarTelf);  edtxTiempo\_programado = findViewById(R.id.edtxTiempo);  btnEnviarBlue = findViewById(R.id.btnEnviarBlue);  txMensaje.setVisibility(View.GONE);  usuario = Preferences.getUsuario(getApplicationContext(),"obusuario");  bluetoothIn = new Handler() {  public void handleMessage(android.os.Message msg) {  if (msg.what == handlerState) {  //Interacción con los datos de ingreso  }  }  };  if(verificadorEstadoBt()){  mbluetoothAdapter=BluetoothAdapter.getDefaultAdapter();  if(mbluetoothAdapter.isEnabled()){  bluetothView.setImageResource(R.drawable.ic\_bluetooth\_on);  Toast.makeText(this,"Bluetooh activado",Toast.LENGTH\_SHORT).show();  }else{  bluetothView.setImageResource(R.drawable.ic\_bluetooth\_off);  Toast.makeText(this,"Bluetooh desactivado",Toast.LENGTH\_SHORT).show();  }  }  FloatingActionButton fabreturn = findViewById(R.id.fabReturn2);  fabreturn.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  onPause();  onBackPressed();  finish();  }  });  if(usuario!=null) {  String telefono = Preferences.ObtenerCredenciales(this,"telefono", (String) null);  if(telefono!=null){  edtxtelefonaenviar.setText(telefono);  edtxtelefonaenviar.setEnabled(false);  }  for (Mochila m : usuario.getMochilas()) {  if (!m.getAlias().isEmpty() && !m.getEncd\_apagado().equals("off")) {  mochilas\_activas.add(m.getAlias());  alias\_id.add("alias");  } else {  if(!m.getEncd\_apagado().equals("off")){  mochilas\_activas.add(m.getId\_dispositivo());  alias\_id.add("id");  }  }  }  }  if(TextUtils.isEmpty(edtxtelefonaenviar.getText().toString())){  btnModifTelefono.setText("ACEPTAR");  edtxtelefonaenviar.setEnabled(true);  }else{  btnModifTelefono.setText("MODIFICAR");  }  ArrayAdapter<CharSequence> adapter = new ArrayAdapter(this,android.R.layout.simple\_spinner\_item,mochilas\_activas);  mochilas\_spinner.setAdapter(adapter);  botonEncenderBt.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  if(!mbluetoothAdapter.isEnabled()){  Intent intent = new Intent(BluetoothAdapter.ACTION\_REQUEST\_ENABLE);  startActivityForResult(intent,REQUEST\_ENABLE\_BT);  }else{  Toast.makeText(getApplication(),"Bluetooth ya esta encendido",Toast.LENGTH\_SHORT).show();  }  }  });  botonApagarBt.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  if(mbluetoothAdapter.isEnabled()){  mbluetoothAdapter.disable();  Toast.makeText(getApplication(),"Apagando Bluetooth",Toast.LENGTH\_SHORT).show();  bluetothView.setImageResource(R.drawable.ic\_bluetooth\_off);  btnActivarModo.setEnabled(false);  estado\_modo.setEnabled(false);  con\_nom\_bluetooh.setText("");  }else{  Toast.makeText(getApplication(),"Bluetooth ya esta apagado",Toast.LENGTH\_SHORT).show();  }  if (btSocket!=null)  {  try {btSocket.close();}  catch (IOException e)  { Toast.makeText(getApplication(), "Error", Toast.LENGTH\_SHORT).show();;}  }  }  });  estado\_modo.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  if(MyConexionBT!=null){  if(estado\_modo.isChecked()){  MyConexionBT.write("e");  }else{  MyConexionBT.write("r");  }  }  if(estado\_modo.isChecked()){  edtxTiempo\_programado.setVisibility(View.VISIBLE);  btnEnviarBlue.setVisibility(View.VISIBLE);  edtxTiempo\_programado.setEnabled(true);  btnEnviarBlue.setEnabled(true);  }else{  edtxTiempo\_programado.setVisibility(View.GONE);  btnEnviarBlue.setVisibility(View.GONE);  edtxTiempo\_programado.setEnabled(false);  btnEnviarBlue.setEnabled(false);  Toast.makeText(getApplication(),"Real",Toast.LENGTH\_SHORT).show();  }  }  });  btnconectarBlue.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  txMensaje.setVisibility(View.GONE);  Set<BluetoothDevice> pairedDevices = mbluetoothAdapter.getBondedDevices();  if(mbluetoothAdapter.isEnabled()){  if(pairedDevices.size()>0){  for(BluetoothDevice device: pairedDevices){  if(nom\_bluetooth\_Ref.getText().toString().equalsIgnoreCase(device.getName())){  con\_nom\_bluetooh.setText(device.getName());  nom\_bluetooth = device.getName();  adress\_bluetooth = device.getAddress();  btnActivarModo.setEnabled(true);  }  }if(TextUtils.isEmpty(con\_nom\_bluetooh.getText().toString())){  Toast.makeText(getApplicationContext(),"Bluetooth No Vinculado ",Toast.LENGTH\_LONG).show();  txMensaje.setVisibility(View.VISIBLE);  txMensaje.setText("Dispositivo NO VINCULADO\nVaya a configuraciones y vincule el dispositivo\nY vuelva aplastar el boton CONECTAR BLUETOOTH");  btnActivarModo.setEnabled(false);  }  }  }else{  Toast.makeText(getApplicationContext(),"Active el Bluetooth ",Toast.LENGTH\_SHORT).show();  btnActivarModo.setEnabled(false);  }  }  });  btnModifTelefono.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  if(btnModifTelefono.getText().toString().equalsIgnoreCase("MODIFICAR")){  edtxtelefonaenviar.setEnabled(true);  btnModifTelefono.setText("ACEPTAR");  }else{  if(TextUtils.isEmpty(edtxtelefonaenviar.getText().toString())){  edtxtelefonaenviar.setError("Este campo no puede estar vacio");  }else{  edtxtelefonaenviar.setEnabled(false);  btnModifTelefono.setText("MODIFICAR");  }  }  }  });  btnActivarModo.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  if(MyConexionBT==null) {  onResume();  }  if(verficarMochias\_activas(mochilas\_spinner.getSelectedItemPosition())){  if(nom\_bluetooth!=null&&adress\_bluetooth!=null&&!edtxtelefonaenviar.getText().toString().isEmpty()&&  !con\_nom\_bluetooh.getText().toString().isEmpty()){  estado\_modo.setEnabled(true);  }else{  if(nom\_bluetooth==null||adress\_bluetooth==null){  Toast.makeText(getApplicationContext(),"Vincule con el dispositivo Bluetooth ",Toast.LENGTH\_SHORT).show();  }else if(TextUtils.isEmpty(edtxtelefonaenviar.getText().toString())){  edtxtelefonaenviar.setError("Este campo no puede estar vacio");  }  }  }  }  });  btnEnviarBlue.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  if(!TextUtils.isEmpty(edtxTiempo\_programado.getText().toString().trim())){  if(Integer.parseInt(edtxTiempo\_programado.getText().toString().trim())!=0){  if(MyConexionBT==null){  onResume();  }else{  if(verificadorEstadoBt()){  System.out.println(MyConexionBT);  System.out.println(adress\_bluetooth);  System.out.println(edtxtelefonaenviar.getText().toString().trim());  String dataenviar = edtxtelefonaenviar.getText().toString().trim();  String tiempo = edtxTiempo\_programado.getText().toString().trim();  //MyConexionBT.write(dataenviar.getBytes());  MyConexionBT.write(tiempo);  }  }  }else{  edtxTiempo\_programado.setError("El tiempo debe ser mayor que 0");  }  }else{  edtxTiempo\_programado.setError("Este campo no puede estar vacio");  }  }  });  }  public boolean verficarMochias\_activas(int valor){  if(valor<0){  Toast.makeText(this,"Active al menos una mochila",Toast.LENGTH\_SHORT).show();  return false;  }return true;  }  public boolean verificadorEstadoBt(){  mbluetoothAdapter = BluetoothAdapter.getDefaultAdapter();  if(mbluetoothAdapter==null){  Toast.makeText(this,"El dispositivo no soporta Bluetooth",Toast.LENGTH\_SHORT).show();  return false;  }else{  if(!mbluetoothAdapter.isEnabled()){  Intent enableBlue = new Intent(BluetoothAdapter.ACTION\_REQUEST\_ENABLE);  startActivityForResult(enableBlue,1);  bluetothView.setImageResource(R.drawable.ic\_bluetooth\_on);  return false;  }  return true;  }  }  private BluetoothSocket createBluetoothSocket(BluetoothDevice device) throws IOException  {  //crea un conexion de salida segura para el dispositivo usando el servicio UUID  return device.createRfcommSocketToServiceRecord(BTMODULEUUID);  }  @Override  public void onResume()  {  super.onResume();  if(estado\_modo.isEnabled()){  BluetoothDevice device = mbluetoothAdapter.getRemoteDevice(adress\_bluetooth);  try  {  btSocket = createBluetoothSocket(device);  } catch (IOException e) {  Toast.makeText(getBaseContext(), "La creacción del Socket fallo", Toast.LENGTH\_LONG).show();  }  // Establece la conexión con el socket Bluetooth.  try  {  btSocket.connect();  } catch (IOException e) {  try {  btSocket.close();  } catch (IOException e2) {  System.out.println(e2.getMessage());  }  }  MyConexionBT = new ConnectedThread(btSocket);  MyConexionBT.start();  }  }  @Override  public void onPause()  {  super.onPause();  if(btSocket!=null){  try  { // Cuando se sale de la aplicación esta parte permite que no se deje abierto el socket  btSocket.close();  }  catch (IOException e2) {  System.out.println(e2.getMessage());  }  }  }  // Defines several constants used when transmitting messages between the  // service and the UI.  private interface MessageConstants {  public static final int MESSAGE\_READ = 0;  public static final int MESSAGE\_WRITE = 1;  public static final int MESSAGE\_TOAST = 2;  // ... (Add other message types here as needed.)  }  //Crea la clase que permite crear el evento de conexion  private class ConnectedThread extends Thread  {  private final BluetoothSocket mmSocket;  private final InputStream mmInStream;  private final OutputStream mmOutStream;  private byte[] mmBuffer; // mmBuffer store for the stream  public ConnectedThread(BluetoothSocket socket)  {  mmSocket = socket;  InputStream tmpIn = null;  OutputStream tmpOut = null;  try  {  tmpIn = socket.getInputStream();  tmpOut = socket.getOutputStream();  } catch (IOException e) {  System.out.println(e.getMessage());  }  mmInStream = tmpIn;  mmOutStream = tmpOut;  }  public void run()  {  byte[] byte\_in = new byte[1];  int numBytes; // bytes returned from read()  // Se mantiene en modo escucha para determinar el ingreso de datos  while (true) {  try {  // Read from the InputStream.  /\* = mmInStream.read(mmBuffer);  // Send the obtained bytes to the UI activity.  Message readMsg = bluetoothIn.obtainMessage(  MessageConstants.MESSAGE\_READ, numBytes, -1,  mmBuffer);  readMsg.sendToTarget();\*/  mmInStream.read(byte\_in);  char ch = (char) byte\_in[0];  bluetoothIn.obtainMessage(handlerState, ch).sendToTarget();  } catch (IOException e) {  System.out.println(e.getMessage());  break;  }  }  }  //Envio de trama  public void write(String input) {  try {  mmOutStream.write(input.getBytes());  }  catch (IOException e)  {  //si no es posible enviar datos se cierra la conexión  Toast.makeText(getBaseContext(), "La Conexión fallo", Toast.LENGTH\_LONG).show();  }  }  /\*public void write(byte[] bytes) {  try {  mmOutStream.write(bytes);  // Share the sent message with the UI activity.  Message writtenMsg = bluetoothIn.obtainMessage(  MessageConstants.MESSAGE\_WRITE, -1, -1, mmBuffer);  writtenMsg.sendToTarget();  } catch (IOException e) {  System.out.println(e.getMessage());  // Send a failure message back to the activity.  Message writeErrorMsg =  bluetoothIn.obtainMessage(MessageConstants.MESSAGE\_TOAST);  Bundle bundle = new Bundle();  bundle.putString("toast", "No puede enviar la data al otro dispositivo");  writeErrorMsg.setData(bundle);  bluetoothIn.sendMessage(writeErrorMsg);  }  }\*/  public void cancel() {  try {  mmSocket.close();  } catch (IOException e) {  System.out.println(e.getMessage());  }  }  }  } | |

|  |  |
| --- | --- |
| Archivo: | PerfilFragment.java |
| Descripción: |  |
| public class PerfilFragment extends Fragment {  Button btnCerrarSesion;  CircleImageView photo\_user;  View root;  TextView numMochilas,nombreUsuario,textviewtelefono,textViewcorreo;  public View onCreateView(@NonNull LayoutInflater inflater,  ViewGroup container, Bundle savedInstanceState) {  root = inflater.inflate(R.layout.fragment\_perfil, container, false);  //Inicializando Variables  btnCerrarSesion = root.findViewById(R.id.btnCerrarSesion);  photo\_user = root.findViewById(R.id.imgViewPhoto);  numMochilas = root.findViewById(R.id.TextViewMochilas);  nombreUsuario = root.findViewById(R.id.txtUserPerfil);  textviewtelefono = root.findViewById(R.id.TextView\_telefono);  textViewcorreo = root.findViewById(R.id.TextView\_mail);  //Se borra las preferencias guardadas y se regresa a la pagina de Login  btnCerrarSesion.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  Preferences.LimpiarCredenciales(getContext());  Intent intent = new Intent(getActivity(), Login.class);  startActivity(intent);  getActivity().finish();  }  });  //Obteniendo preferencias  Usuario usuario = Preferences.getUsuario(getActivity().getApplicationContext(),"obusuario");  String photo = Preferences.ObtenerCredenciales(getActivity().getApplicationContext(),"photo\_user","null");  String email = Preferences.ObtenerCredenciales(getActivity().getApplicationContext(),"email","null");  String nomuser = Preferences.ObtenerCredenciales(getActivity().getApplicationContext(),"nom\_usuario","null");  String telefono = Preferences.ObtenerCredenciales(getActivity().getApplicationContext(),"telefono","null");  //Modificando parametros no nulos  if(usuario!=null){  numMochilas.setText(String.valueOf(usuario.getMochilas().size()));  }  if(!photo.equals("null")){  Picasso.with(getActivity().getApplicationContext()).load(photo).into(photo\_user);  }if(!nomuser.equals("null")){  nombreUsuario.setText(nomuser);  }if(!email.equals("null")){  textViewcorreo.setText(email);  }if(!telefono.equals("null")){  textviewtelefono.setText(telefono);  }  return root;  }  } | |

|  |  |
| --- | --- |
| Archivo: | MyFirebaseInstanceService.java |
| Descripción: |  |
| public class MyFirebaseInstanceService extends FirebaseMessagingService {  //Ctrl + O  @Override  public void onNewToken(String s) {  super.onNewToken(s); Log.d("TOKENFIREBASE",  s);  }  @Override  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)  .setContentInfo("info");  notificationManager.notify(  new Random().nextInt(), notificationBuilder.build());  }  } | |

# Layouts

|  |  |
| --- | --- |
| Nombre: | activity\_add\_mochilas.xml |
| Descripción: | Pantalla donde el usuario agrega mochilas para rastrear |
| <?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"  tools:context=".ui.mochila.AddMochilas">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:paddingBottom="10dp">  <com.google.android.material.floatingactionbutton.FloatingActionButton  android:id="@+id/fabReturn"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_gravity="bottom|end"  android:layout\_marginStart="@dimen/fab\_margin"  android:layout\_marginTop="@dimen/fab\_margin"  android:layout\_marginEnd="@dimen/fab\_margin"  android:layout\_marginBottom="@dimen/fab\_margin"  android:src="@android:drawable/ic\_menu\_revert" />  <TextView  android:id="@+id/textView3"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_gravity="center"  android:layout\_marginTop="8dp"  android:layout\_marginBottom="16dp"  android:fontFamily="sans-serif-black"  android:gravity="center"  android:text="Registro de Mochila"  android:textSize="24sp" />  </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="vertical">  <EditText  android:id="@+id/EdtxtAliasMochila"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:ems="10"  android:hint="Alias"  android:inputType="textPersonName" />  <EditText  android:id="@+id/EdtxViewIdmochila"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:ems="10"  android:hint="Id Dispositvo"  android:inputType="textPersonName" />  <Button  android:id="@+id/btnEnviarMochila"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:text="Registrar Mochila" />  </LinearLayout>  </LinearLayout> | |

|  |  |
| --- | --- |
| Nombre: | activity\_bienvenida.xml |
| Descripción: |  |
| <?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"  android:background="#094275"  android:gravity="center"  tools:context=".Bienvenida">  <ImageView  android:id="@+id/logo\_portada"  android:layout\_width="359dp"  android:layout\_height="360dp"  android:layout\_gravity="center"  app:srcCompat="@drawable/logotipo" />  </RelativeLayout> | |

|  |  |
| --- | --- |
| Nombre: | activity\_login.xml |
| Descripción: |  |
| <?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="#094275"  tools:context=".Login">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="600dp"  android:background="#094275"  android:orientation="vertical"  app:layout\_constraintTop\_toTopOf="parent">  <ImageView  android:layout\_width="164dp"  android:layout\_height="131dp"  android:layout\_gravity="center"  android:layout\_marginTop="30dp"  android:src="@drawable/logotipo" />  <ScrollView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:scrollbarAlwaysDrawVerticalTrack="false">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_gravity="center"  android:gravity="center"  android:orientation="vertical">  <com.google.android.material.textfield.TextInputLayout  style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"  android:layout\_width="330dp"  android:layout\_height="54dp"  android:layout\_marginTop="30dp"  app:boxStrokeColor="@color/colorPrimaryDark">  <com.google.android.material.textfield.TextInputEditText  android:id="@+id/Edtxt\_login"  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:layout\_gravity="center\_vertical"  android:layout\_marginTop="0dp"  android:background="#1e1a1a"  android:drawableStart="@drawable/ic\_baseline\_person\_outline\_24"  android:hint="EMAIL"  android:inputType="textPersonName"  android:padding="5dp"  android:singleLine="true"  android:textColor="#ffffff"  android:textColorHint="#b4b7b7"  android:textSize="22sp" />  </com.google.android.material.textfield.TextInputLayout>  <com.google.android.material.textfield.TextInputLayout  style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"  android:layout\_width="330dp"  android:layout\_height="54dp"  android:layout\_marginTop="30dp"  app:boxStrokeColor="@color/colorPrimaryDark">  <com.google.android.material.textfield.TextInputEditText  android:id="@+id/Edtxt\_pass"  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:background="#1e1a1a"  android:drawableStart="@drawable/ic\_outline\_lock\_white\_24"  android:hint="PASSWORD"  android:imeOptions="actionSend"  android:inputType="textPassword"  android:maxLength="20"  android:padding="5dp"  android:singleLine="true"  android:textColor="#ffffff"  android:textColorHint="#b4b7b7"  android:textSize="22sp" />  </com.google.android.material.textfield.TextInputLayout>  <Button  android:id="@+id/btnLogin"  android:layout\_width="330dp"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:background="@drawable/back\_loginbutton"  android:text="LOGIN"  android:textColor="#ffffff"  android:textSize="22sp"  android:textStyle="normal|bold" />  <Button  android:id="@+id/btnRegistro"  android:layout\_width="330dp"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:background="@drawable/back\_registbutton"  android:text="REGISTRARSE"  android:textColor="#221F1F"  android:textSize="22sp" />  <TextView  android:id="@+id/TxtO"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_gravity="center"  android:layout\_marginTop="20dp"  android:gravity="center"  android:text="O"  android:textSize="18sp"  android:textStyle="normal|bold" />  <com.google.android.gms.common.SignInButton  android:id="@+id/signButton"  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:layout\_marginTop="15dp"  app:buttonSize="wide">  </com.google.android.gms.common.SignInButton>  </LinearLayout>  </ScrollView>  </LinearLayout>  </androidx.constraintlayout.widget.ConstraintLayout> | |

|  |  |
| --- | --- |
| Nombre: | activity\_main.xml |
| Descripción: |  |
| <?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:id="@+id/container"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="#094275">  <com.google.android.material.bottomnavigation.BottomNavigationView  android:id="@+id/nav\_view"  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:background="#dc7364"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintLeft\_toLeftOf="parent"  app:layout\_constraintRight\_toRightOf="parent"  app:menu="@menu/bottom\_nav\_menu" />  <fragment  android:id="@+id/nav\_host\_fragment"  android:name="androidx.navigation.fragment.NavHostFragment"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  app:defaultNavHost="true"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintHorizontal\_bias="0.0"  app:layout\_constraintLeft\_toLeftOf="parent"  app:layout\_constraintRight\_toRightOf="parent"  app:layout\_constraintTop\_toTopOf="parent"  app:layout\_constraintVertical\_bias="0.0"  app:navGraph="@navigation/mobile\_navigation" />  </androidx.constraintlayout.widget.ConstraintLayout> | |

|  |  |
| --- | --- |
| Nombre: | activity\_maps.xml |
| Descripción: |  |
| <?xml version="1.0" encoding="utf-8"?>  <fragment xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:map="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:id="@+id/map"  android:name="com.google.android.gms.maps.SupportMapFragment"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="#094275"  tools:context=".ui.mapa.MapsActivity" /> | |

|  |  |
| --- | --- |
| Nombre: | **activity\_modo\_mochila.xml** |
| Descripción: |  |
| <?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"  tools:context=".ui.mochila.ModoMochila">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:paddingBottom="10dp">  <com.google.android.material.floatingactionbutton.FloatingActionButton  android:id="@+id/fabReturn2"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_gravity="bottom|end"  android:layout\_marginStart="@dimen/fab\_margin"  android:layout\_marginTop="@dimen/fab\_margin"  android:layout\_marginEnd="@dimen/fab\_margin"  android:layout\_marginBottom="@dimen/fab\_margin"  android:src="@android:drawable/ic\_menu\_revert" />  <TextView  android:id="@+id/textView31"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_gravity="center"  android:layout\_marginTop="8dp"  android:layout\_marginBottom="16dp"  android:fontFamily="sans-serif-black"  android:gravity="center"  android:text="Cambiar Modo"  android:textSize="24sp" />  <ImageView  android:id="@+id/bluetoothView"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_weight="1" />  </LinearLayout>  <Spinner  android:id="@+id/mochilas\_act\_spinner"  android:layout\_width="fill\_parent"  android:layout\_height="wrap\_content" />  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal">  <EditText  android:id="@+id/EdtxTelefonEnviar"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:ems="10"  android:hint="Telefono a Enviar SMS"  android:inputType="textPersonName" />  <Button  android:id="@+id/btnModificarTelf"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="Modificar" />  </LinearLayout>  <Button  android:id="@+id/btnModoEncender"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:layout\_marginBottom="20dp"  android:text="ENCENDER BLUETOOH" />  <Button  android:id="@+id/btnModoApagar"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="APAGAR BLUETOOTH" />  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal">  <Switch  android:id="@+id/switchModo"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:enabled="false"  android:text="Live" />  <TextView  android:id="@+id/txtViewEstatico"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Estatico" />  </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal">  <TextView  android:id="@+id/textView10"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Nombre del Bluetooh a conectar:" />  <TextView  android:id="@+id/RefBluetooh"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="8dp"  android:layout\_weight="1"  android:text="HC-06" />  </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal">  <TextView  android:id="@+id/textView30"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Contraseña del Bluetooh a conectar:" />  <TextView  android:id="@+id/PassBluetooh"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="8dp"  android:layout\_weight="1"  android:text="1234" />  </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal">  <TextView  android:id="@+id/textView4"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Nombre del Bluetooh conectado:" />  <TextView  android:id="@+id/NombBluetooh"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="8dp"  android:layout\_weight="1" />  </LinearLayout>  <Button  android:id="@+id/btnConectarBlue"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="CONECTAR BLUETOOTH" />  <Button  android:id="@+id/btnActivarModo"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:enabled="false"  android:text="ACTIVAR MODO" />  <TextView  android:id="@+id/textMensaje"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content" />  <EditText  android:id="@+id/edtxTiempo"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:ems="10"  android:hint="Tiempo programado en segundos"  android:inputType="number"  android:visibility="gone" />  <Button  android:id="@+id/btnEnviarBlue"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:enabled="false"  android:text="ENVIAR"  android:visibility="gone" />  </LinearLayout>  </LinearLayout> | |

|  |  |
| --- | --- |
| Nombre: | **activity\_registro.xml** |
| Descripción: |  |
| <?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:background="#094275"  android:orientation="vertical"  tools:context=".Registro">  <TextView  android:id="@+id/textView2"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_gravity="center"  android:layout\_marginTop="20dp"  android:fontFamily="sans-serif-medium"  android:text="REGISTRO"  android:textColor="#FFFFFF"  android:textSize="36dp" />  <ScrollView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:scrollbarAlwaysDrawVerticalTrack="false">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="vertical">  <com.google.android.material.textfield.TextInputLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  app:boxStrokeColor="@color/colorPrimaryDark">  <com.google.android.material.textfield.TextInputEditText  android:id="@+id/EdTnombre\_registro"  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:layout\_marginStart="8dp"  android:layout\_marginTop="30dp"  android:layout\_marginEnd="8dp"  android:ems="10"  android:hint="NOMBRE"  android:inputType="textPersonName"  android:maxLength="20"  android:singleLine="true"  android:text=""  android:textColor="#ffffff"  android:textColorHint="#b4b7b7"  android:textSize="22sp" />  </com.google.android.material.textfield.TextInputLayout>  <com.google.android.material.textfield.TextInputLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  app:boxStrokeColor="@color/colorPrimaryDark">  <com.google.android.material.textfield.TextInputEditText  android:id="@+id/EdTapellido\_registro"  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:layout\_marginStart="8dp"  android:layout\_marginTop="30dp"  android:layout\_marginEnd="8dp"  android:ems="10"  android:hint="APELLIDO"  android:inputType="textPersonName"  android:maxLength="20"  android:singleLine="true"  android:text=""  android:textColor="#ffffff"  android:textColorHint="#b4b7b7"  android:textSize="22sp" />  </com.google.android.material.textfield.TextInputLayout>  <com.google.android.material.textfield.TextInputLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  app:boxStrokeColor="@color/colorPrimaryDark">  <com.google.android.material.textfield.TextInputEditText  android:id="@+id/EdTemail\_registro"  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:layout\_marginStart="8dp"  android:layout\_marginTop="20dp"  android:layout\_marginEnd="8dp"  android:ems="10"  android:hint="CORREO"  android:inputType="textEmailAddress"  android:singleLine="true"  android:text=""  android:textColor="#ffffff"  android:textColorHint="#b4b7b7"  android:textSize="22sp" />  </com.google.android.material.textfield.TextInputLayout>  <com.google.android.material.textfield.TextInputLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  app:boxStrokeColor="@color/colorPrimaryDark">  <com.google.android.material.textfield.TextInputEditText  android:id="@+id/Edcelular"  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:layout\_marginStart="8dp"  android:layout\_marginTop="20dp"  android:layout\_marginEnd="8dp"  android:digits="0123456789"  android:ems="10"  android:hint="CELULAR"  android:inputType="phone"  android:maxLength="10"  android:singleLine="true"  android:text=""  android:textColor="#ffffff"  android:textColorHint="#b4b7b7"  android:textSize="22sp" />  </com.google.android.material.textfield.TextInputLayout>  <com.google.android.material.textfield.TextInputLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  app:boxStrokeColor="@color/colorPrimaryDark">  <com.google.android.material.textfield.TextInputEditText  android:id="@+id/EdTpass\_registro"  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:layout\_marginStart="8dp"  android:layout\_marginTop="20dp"  android:layout\_marginEnd="8dp"  android:ems="10"  android:hint="CONTRASEÑA"  android:inputType="textPassword"  android:maxLength="20"  android:singleLine="true"  android:text=""  android:textColor="#ffffff"  android:textColorHint="#b4b7b7"  android:textSize="22sp" />  </com.google.android.material.textfield.TextInputLayout>  <TextView  android:id="@+id/Txtpass\_longitud"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginStart="8dp"  android:layout\_marginTop="10dp"  android:layout\_marginEnd="8dp"  android:drawableStart="@drawable/ic\_cancel\_red\_24dp"  android:padding="5dp"  android:text="Al menos 7 caracteres"  android:visibility="gone" />  <TextView  android:id="@+id/Txtpass\_espacio"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginStart="8dp"  android:layout\_marginTop="10dp"  android:layout\_marginEnd="8dp"  android:drawableStart="@drawable/ic\_cancel\_red\_24dp"  android:padding="5dp"  android:text="Sin espacio"  android:visibility="gone" />  <TextView  android:id="@+id/Txtpass\_mayuscula"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginStart="8dp"  android:layout\_marginTop="10dp"  android:layout\_marginEnd="8dp"  android:drawableStart="@drawable/ic\_cancel\_red\_24dp"  android:padding="5dp"  android:text="Al menos 1 mayuscula"  android:visibility="gone" />  <TextView  android:id="@+id/Txtpass\_digito"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginStart="8dp"  android:layout\_marginTop="10dp"  android:layout\_marginEnd="8dp"  android:drawableStart="@drawable/ic\_cancel\_red\_24dp"  android:padding="5dp"  android:text="Al menos 1 digito"  android:visibility="gone" />  <com.google.android.material.textfield.TextInputLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  app:boxStrokeColor="@color/colorPrimaryDark">  <com.google.android.material.textfield.TextInputEditText  android:id="@+id/EdTpassconfir\_registro"  android:layout\_width="match\_parent"  android:layout\_height="50dp"  android:layout\_marginStart="8dp"  android:layout\_marginTop="20dp"  android:layout\_marginEnd="8dp"  android:ems="10"  android:hint="CONFIRMAR CONTRASEÑA"  android:inputType="textPassword"  android:maxLength="20"  android:singleLine="true"  android:text=""  android:textColor="#ffffff"  android:textColorHint="#b4b7b7"  android:textSize="22sp" />  </com.google.android.material.textfield.TextInputLayout>  <com.google.android.material.textfield.TextInputLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  app:boxStrokeColor="@color/colorPrimaryDark" />  <Button  android:id="@+id/btnRegistrarse"  android:layout\_width="330dp"  android:layout\_height="50dp"  android:layout\_gravity="center"  android:layout\_marginTop="20dp"  android:background="@drawable/back\_loginbutton"  android:onClick="registro"  android:text="REGISTRARSE"  android:textColor="#100F0F"  android:textSize="22sp" />  </LinearLayout>  </ScrollView>  </LinearLayout> | |

|  |  |
| --- | --- |
| Nombre: | **fragment\_mapa.xml** |
| Descripción: |  |
| <?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="#094275"  tools:context=".ui.mapa.MapaFragment">  <fragment xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:map="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:id="@+id/map"  android:name="com.google.android.gms.maps.SupportMapFragment"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".ui.mapa.MapsActivity" />  <com.google.android.material.floatingactionbutton.FloatingActionButton  android:id="@+id/floatingSearch"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginEnd="8dp"  android:layout\_marginBottom="72dp"  android:clickable="true"  android:foregroundGravity="bottom|right"  android:src="@android:drawable/ic\_menu\_search"  app:layout\_constraintBottom\_toBottomOf="@+id/map"  app:layout\_constraintEnd\_toEndOf="parent" />  </androidx.constraintlayout.widget.ConstraintLayout> | |

|  |  |
| --- | --- |
| Nombre: | **fragment\_mochila.xml** |
| Descripción: |  |
| <?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:background="#094275"  android:orientation="vertical"  tools:context=".ui.mochila.MochilaFragment">  <TextView  android:id="@+id/textView"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="16dp"  android:text="Mis Dispositivos"  android:textAlignment="center"  android:textSize="30sp"  android:textStyle="normal|bold" />  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal">  <androidx.recyclerview.widget.RecyclerView  android:id="@+id/recyclerMochilas"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_margin="16dp" />  </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal">  <Button  android:id="@+id/btnAgregarMochila"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="Agregar Mochila" />  <Button  android:id="@+id/btnCambiarModo"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="Cambiar Modo" />  </LinearLayout>  </LinearLayout> | |

|  |  |
| --- | --- |
| Nombre: | **fragment\_perfil.xml** |
| Descripción: |  |
| <?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"  android:background="#FFFFFF"  tools:context=".ui.perfil.PerfilFragment">  <RelativeLayout  android:id="@+id/container\_img"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:background="#094275"  android:paddingBottom="20sp">  <RelativeLayout  android:id="@+id/imgUser"  android:layout\_marginTop="8dp"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerHorizontal="true">  <de.hdodenhof.circleimageview.CircleImageView  android:id="@+id/imgViewPhoto"  android:layout\_width="100dp"  android:layout\_height="100dp"  android:layout\_marginStart="6dp"  android:layout\_marginTop="6dp"  android:layout\_marginEnd="6dp"  android:layout\_marginBottom="6dp"  android:src="@drawable/sinfoto"  app:civ\_border\_color="#000000"  app:civ\_border\_width="2dp" />  </RelativeLayout>  <TextView  android:id="@+id/txtUserPerfil"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_below="@+id/imgUser"  android:layout\_centerHorizontal="true"  android:fontFamily="sans-serif-light"  android:text="Usuario"  android:textColor="@color/white"  android:textSize="24sp" />  </RelativeLayout>  <LinearLayout  android:id="@+id/display"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  android:layout\_below="@+id/container\_img">  <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:background="#0A4478"  android:layout\_gravity="center"  android:gravity="center"  android:layout\_weight="1"  android:paddingTop="5dp"  android:paddingBottom="5dp">  <LinearLayout  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="vertical">  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:fontFamily="sans-serif-light"  android:text="Mochilas"  android:textColor="@color/white"  android:textSize="18sp"  android:textStyle="bold" />  <TextView  android:id="@+id/TextViewMochilas"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_gravity="center"  android:fontFamily="sans-serif-light"  android:text="0"  android:textAllCaps="true"  android:textColor="@color/white"  android:textSize="15sp" />  </LinearLayout>  </RelativeLayout>  <RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:background="#073259"  android:layout\_gravity="center"  android:gravity="center"  android:layout\_weight="1"  android:paddingTop="5dp"  android:paddingBottom="5dp">  <LinearLayout  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:orientation="vertical">  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:fontFamily="sans-serif-light"  android:text="COPYRIGTH"  android:textColor="@color/white"  android:textSize="18sp"  android:textStyle="bold" />  <TextView  android:layout\_gravity="center"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:fontFamily="sans-serif-light"  android:textAllCaps="true"  android:text="GBAG"  android:textColor="@color/white"  android:textSize="15sp"/>  </LinearLayout>  </RelativeLayout>  </LinearLayout>  <ScrollView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_below="@+id/display">  <RelativeLayout  android:id="@+id/display\_datos\_personal"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="vertical"  android:layout\_centerInParent="true"  android:layout\_marginLeft="30dp"  android:layout\_marginRight="10dp"  android:layout\_marginTop="20dp"  android:layout\_marginBottom="20dp">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="vertical">  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Email"  android:textStyle="bold"  android:textColor="#5C5A5A"  android:textSize="15sp"/>  <TextView  android:id="@+id/TextView\_mail"  android:layout\_marginTop="6dp"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:fontFamily="sans-serif-light"  android:text="example@gmail.com"  android:textColor="#000000"  android:textSize="18sp"/>  <View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:layout\_marginTop="6dp"  android:background="@color/grey"/>  </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="vertical"  android:layout\_marginTop="20dp">  <TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Telofono"  android:textColor="#5C5A5A"  android:textStyle="bold"  android:textSize="15sp"/>  <TextView  android:id="@+id/TextView\_telefono"  android:layout\_marginTop="6dp"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:fontFamily="sans-serif-light"  android:text="09XXXXXXXX"  android:textColor="#000000"  android:textSize="18sp"/>  <View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:layout\_marginTop="6dp"  android:background="@color/grey"/>  </LinearLayout>  <Button  android:id="@+id/btnCerrarSesion"  android:layout\_width="330dp"  android:layout\_height="wrap\_content"  android:layout\_gravity="center"  android:layout\_marginTop="40dp"  android:background="@drawable/back\_loginbutton"  android:text="Cerrar Sesion"  android:textColor="#000000"  android:textStyle="bold" />  </LinearLayout>  </RelativeLayout>  </ScrollView>  </RelativeLayout> | |

|  |  |
| --- | --- |
| Nombre: | **informacion\_dipositivo.xml** |
| Descripción: |  |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="vertical">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="vertical">  <TextView  android:id="@+id/textView9"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="16dp"  android:text="Informacion del Dispositivo"  android:textAlignment="center"  android:textSize="30sp" />  </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal">  <Space  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1" />  <TextView  android:id="@+id/textView3"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="ID:"  android:textSize="24sp" />  <TextView  android:id="@+id/info\_id"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:textSize="24sp" />  <Space  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1" />  </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal">  <Space  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1" />  <TextView  android:id="@+id/textView5"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="% Bateria:"  android:textSize="24sp" />  <TextView  android:id="@+id/info\_bat"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:textSize="24sp" />  <Space  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1" />  </LinearLayout>  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal">  <Space  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1" />  <TextView  android:id="@+id/textView7"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="Modo:"  android:textSize="24sp" />  <TextView  android:id="@+id/info\_modo"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:textSize="24sp" />  <Space  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_weight="1" />  </LinearLayout>  </LinearLayout> | |

|  |  |
| --- | --- |
| Nombre: | **items\_list.xml** |
| Descripción: |  |
| <?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="wrap\_content"  android:background="#094275"  android:paddingTop="8dp"  android:paddingBottom="8dp">  <TextView  android:id="@+id/idDato"  android:layout\_width="150dp"  android:layout\_height="wrap\_content"  android:layout\_margin="8dp"  android:background="#094275"  android:text="Mochila"  android:textSize="18sp" />  <Button  android:id="@+id/imgButtonInfo"  android:layout\_width="28dp"  android:layout\_height="28dp"  android:layout\_marginTop="8dp"  android:layout\_weight="0"  android:background="@android:drawable/ic\_dialog\_info" />  <Switch  android:id="@+id/switchOnOff"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="20dp"  android:layout\_marginTop="8dp"  android:layout\_weight="0"  android:text="ON/OFF"  android:textSize="14sp" />  </LinearLayout> | |