APLICACIONES MÓVILES Y SERVICIOS TELEMÁTICOS

1T - 2020

G1 - Integrantes:

Adriana Murrieta Sánchez  
Winter Álava Intriago  
Giancarlo Guerrero Burgos

Manual Técnico 2Report

Localización de guardias para denuncias



Escuela Superior Politécnica del Litoral

Tabla de contenido

[Introducción 2](#_Toc50698608)

[Descripción del sistema implementado 2](#_Toc50698609)

[Recursos 2](#_Toc50698610)

[Diagramas 3](#_Toc50698611)

[Diagrama del Sistema 3](#_Toc50698612)

[Diagrama de despliegue 3](#_Toc50698613)

[Modelo de la Base de datos 4](#_Toc50698614)

[Diagrama de clases 5](#_Toc50698615)

[Casos de uso 5](#_Toc50698616)

[Diagrama del circuito 6](#_Toc50698617)

[Explicación del código fuente 7](#_Toc50698618)

[Código fuente Arduino 7](#_Toc50698619)

[Archivo: CODIGO\_ARDUINO\_AMST.ino 7](#_Toc50698620)

[Archivo: CODIGO\_WIFI\_AMST.ino 8](#_Toc50698621)

[Código fuente java 10](#_Toc50698622)

[Layouts 40](#_Toc50698623)

[Login 40](#_Toc50698624)

[Registro 41](#_Toc50698625)

[Estudiante 45](#_Toc50698626)

[Guardia 51](#_Toc50698627)

[Administrador 52](#_Toc50698628)

[Reportes 53](#_Toc50698629)

# 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 la finalidad de permitir a los usuarios denunciar, a través de su Smartphone, algún delito ocurrido dentro de las instalaciones de ESPOL, de esta forma los guardias son alertados y pueden acudir a la ayuda del usuario. Además, el usuario tiene la posibilidad de localizar en tiempo real a los guardias más cercanos a su ubicación.

# Descripción del sistema implementado

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

Un usuario estudiante puede visulizar un mapa que indica la ubicación de los guardias más cercanos a él. Además, en caso de ser víctima de un delito, el estudiante cuenta con la posibilidad de reportarlo en un tiempo menor a 30 segundos y así alertar a los guardias cercanos para que puedan ir en su ayuda.

Un usuario guardia tiene acceso a la lista de todos los delitos que se han reportado dentro del campus y una descripción detallada de cada uno de ellos. Adicionalmente se desarrolló un dispositivo que permite sensar constantemente la ubicación de los guardias.

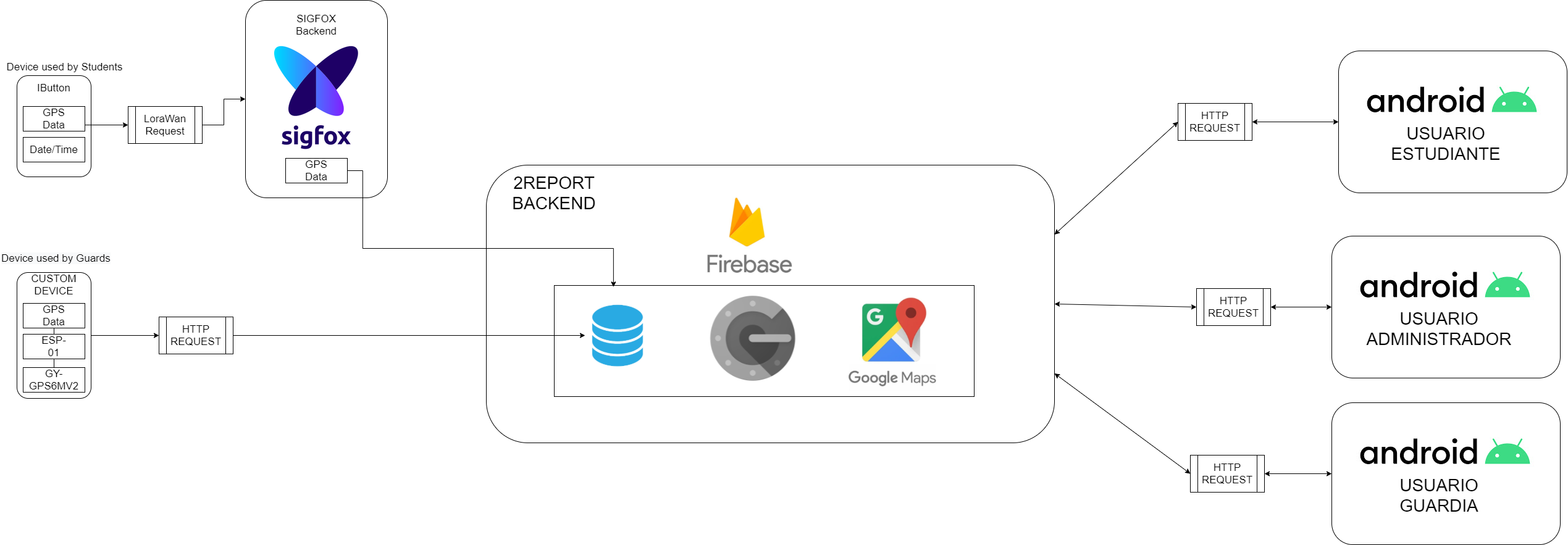
Por último, un usuario administrador pude acceder a la lista de delitos reportados y visualizarlo en un mapa permitiéndole conocer las áreas más afectadas.

# Recursos

|  |  |  |
| --- | --- | --- |
| Hardware | Software | Extras |
| Arduino UNO R3 | Android studio | Caja negra |
| Modulo de Wifi ESP8266-01 | Firebase | Cautín |
| Módulo GPS-gy-neo-6mv2 | Arduino V1.8.13 | Estaño |
| Regulador 3.3V lm1117T |  | Cable arduino |
| 2 borneras de dos puntos |  | Programador ESP8266-01 |
| Cables dupond |  |  |

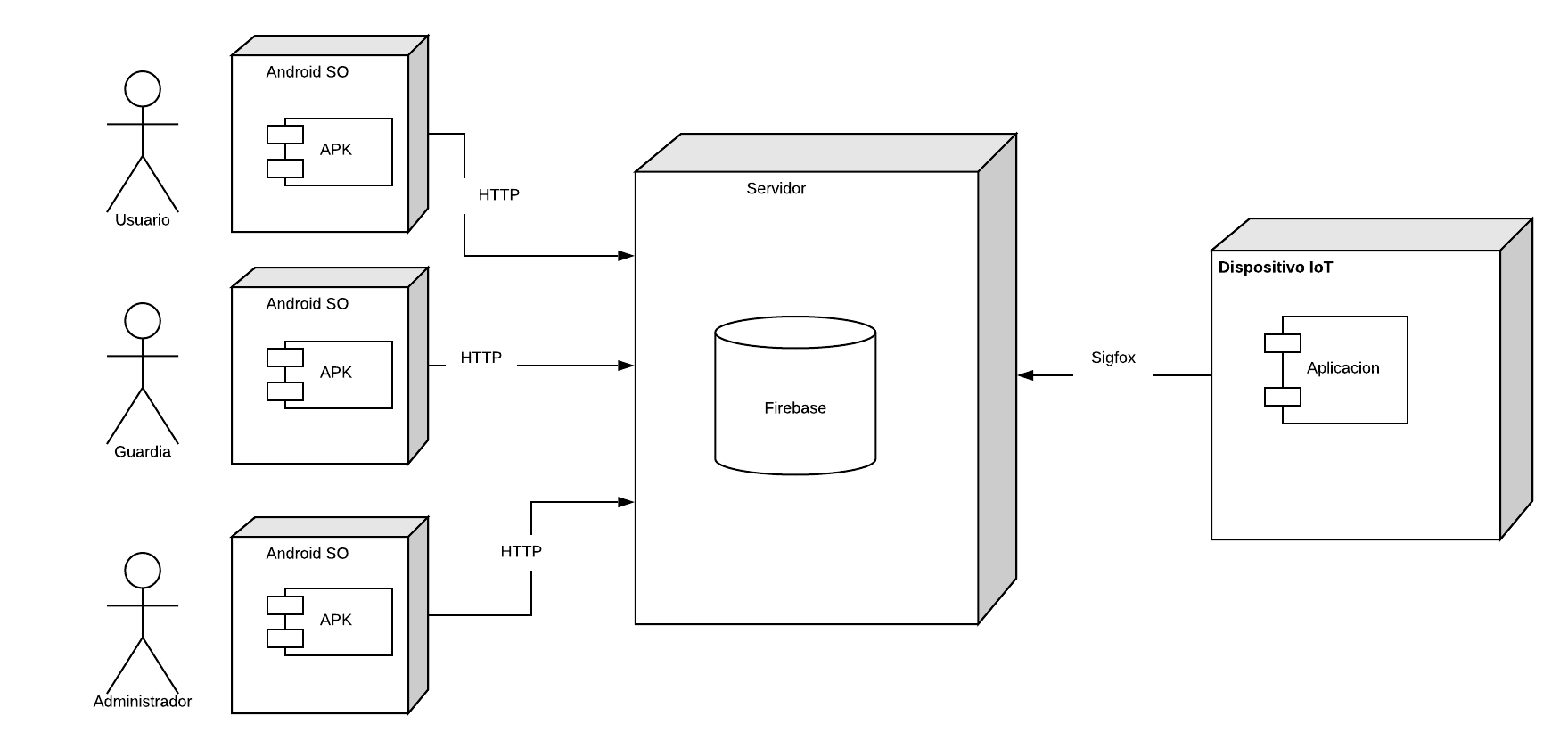
# Diagramas

## Diagrama del Sistema



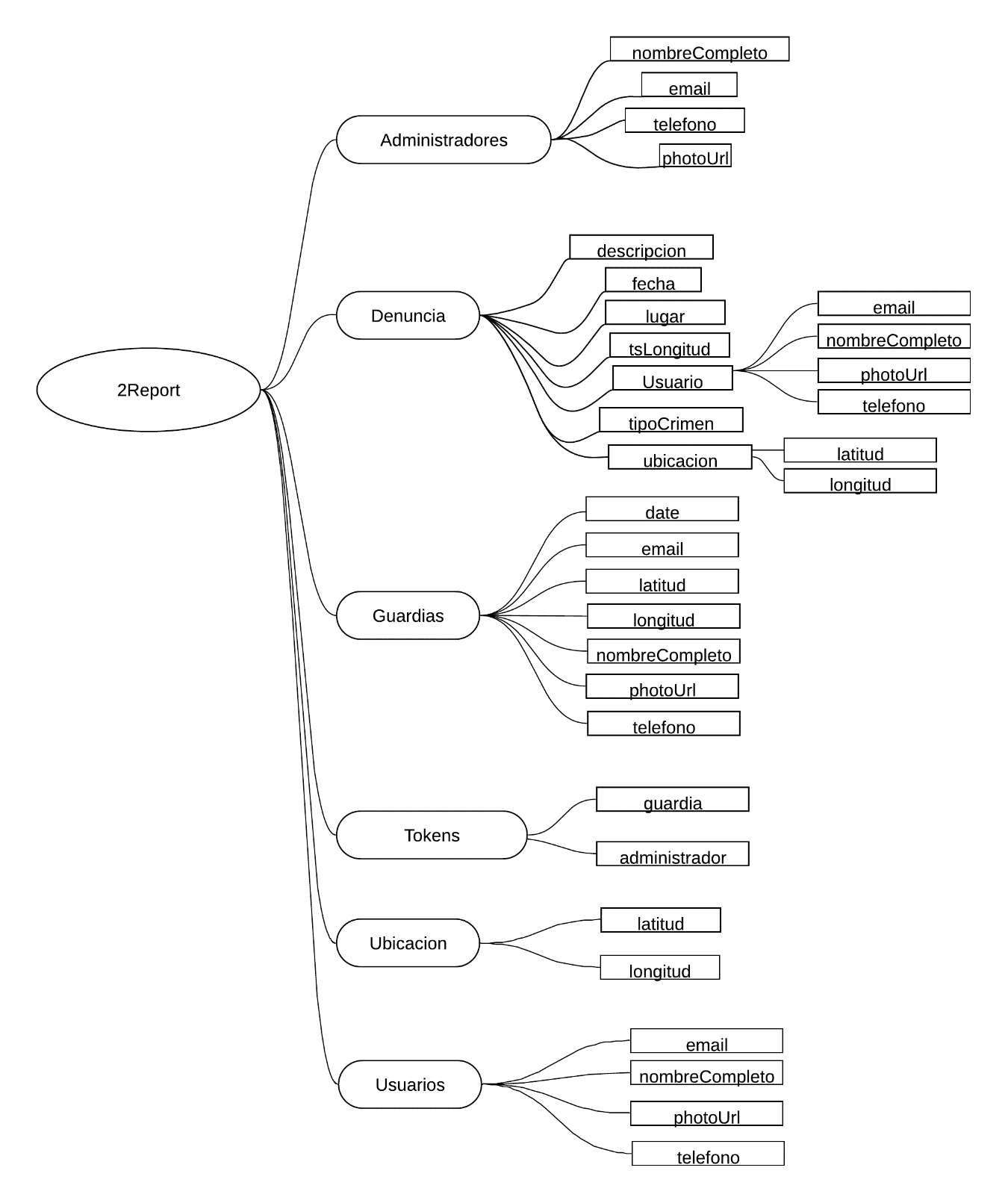
## Diagrama de despliegue

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 un guardia y es enviada mediante callbacks al BD en Firebase para luego ser consultada por los usuarios de la aplicación.

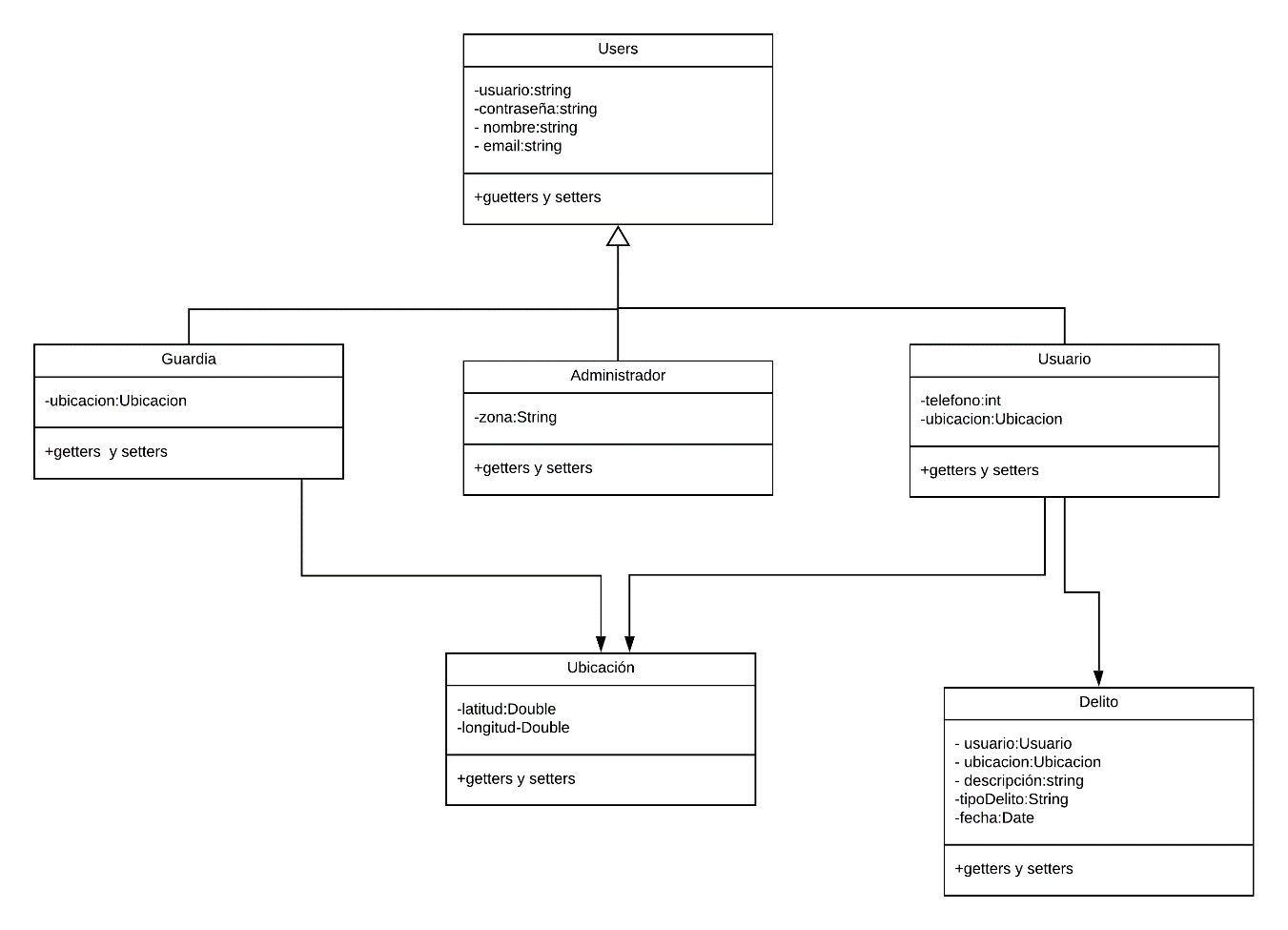


## Modelo de la Base de datos

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

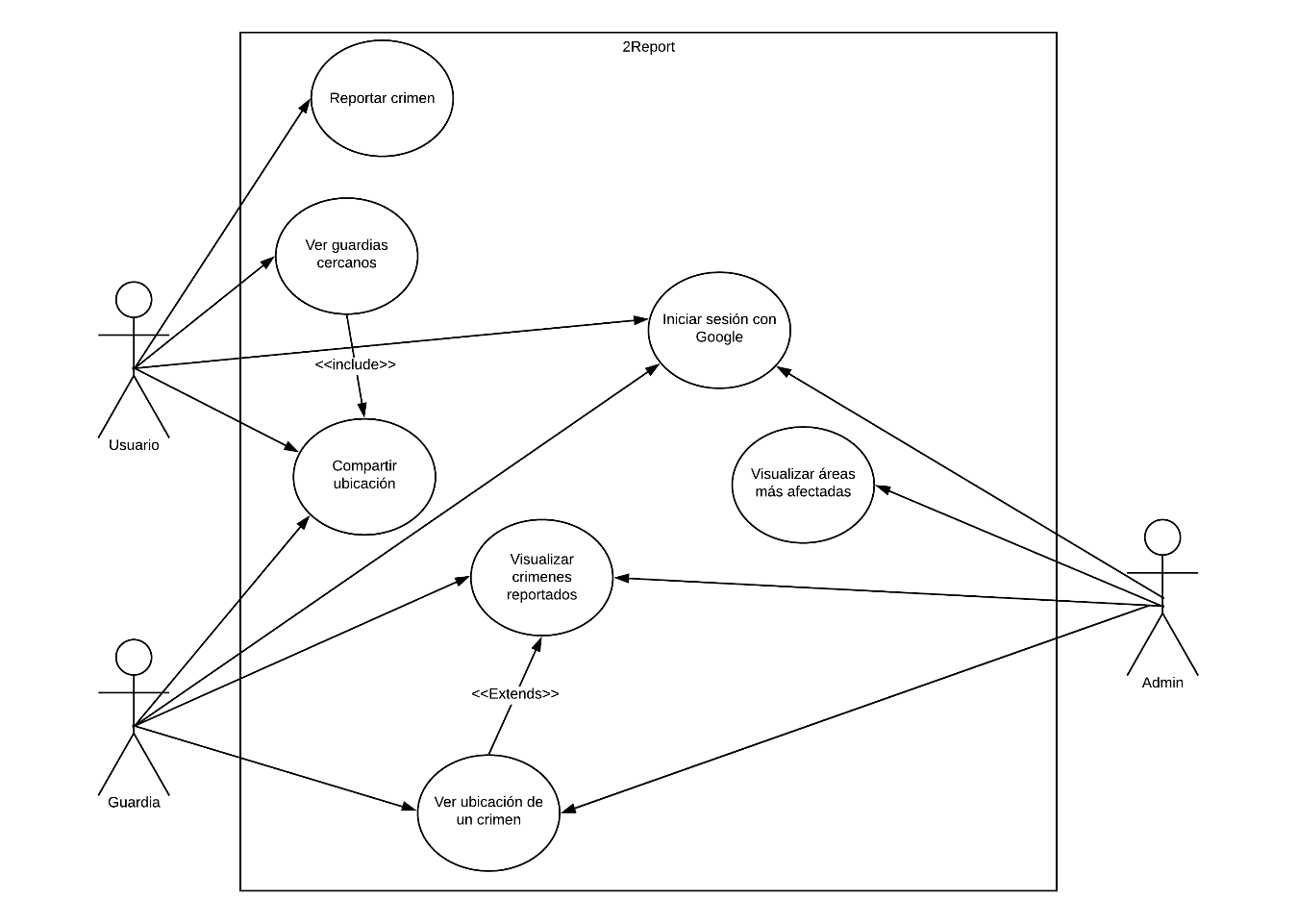
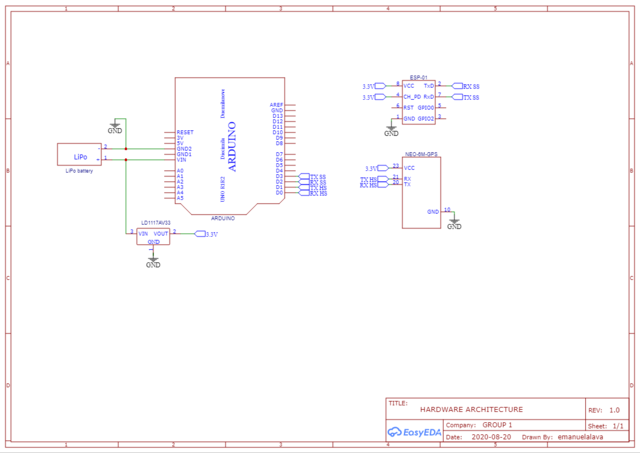


## Diagrama de clases



## 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 los guardias en tiempo real.

# Explicación del código fuente

## Código fuente Arduino

### Archivo: CODIGO\_ARDUINO\_AMST.ino

|  |  |
| --- | --- |
| Función: | void setup() |
| Descripción: |  |
| // Librerías  #include <SoftwareSerial.h>  #include <TinyGPS++.h>  // Definición de los pines del Wifi  int WifiRX = 2; // Pin 2 del arduino  int WifiTX = 3; // Pin 3 del arduino  static const uint32\_t WifiBaud = 9600;  SoftwareSerial WifiSerial(WifiRX, WifiTX);  //Definición del modulo gps  TinyGPSPlus gps;  static const uint32\_t GpsBaud = 9600;  // Trama de información que será enviada al Wifi  String trama;  void setup() {  WifiSerial.begin(WifiBaud);  Serial.begin(GpsBaud);  } | |

|  |  |
| --- | --- |
| Función: | void loop() |
| Descripción: |  |
| void loop() {  while (Serial.available() > 0)  if (gps.encode(Serial.read()))  displayInfo();  } | |

|  |  |
| --- | --- |
| Función: | void displayInfo() |
| Descripción: |  |
| void displayInfo()  {  if (gps.location.isValid() && (gps.date.isValid() && gps.time.isValid()))  {  trama = "@" + String(gps.location.lat(), 6) + "\*" + String(gps.location.lng(), 6) + "|";  String fecha = String(gps.date.day()) + "/" + String(gps.date.month()) + "/" + String(gps.date.year()) + " ";  String tiempo = "";  if (gps.time.hour() < 10) tiempo.concat("0");  tiempo.concat(gps.time.hour());  tiempo.concat(":");  if (gps.time.minute() < 10) tiempo.concat("0");  tiempo.concat(gps.time.minute());  tiempo.concat(":");  if (gps.time.second() < 10) tiempo.concat("0");  tiempo.concat(gps.time.second());  trama.concat(fecha);  trama.concat(tiempo);  WifiSerial.println(trama);  delay(3000);  }  } | |

### Archivo: CODIGO\_WIFI\_AMST.ino

|  |  |
| --- | --- |
| Función: | void setup() |
| Descripción: |  |
| #include <ESP8266WiFi.h>  #include <FirebaseArduino.h>  #define FIREBASE\_HOST "report-281701.firebaseio.com"  #define FIREBASE\_AUTH "RBZ3HocZ3bGaOyRupqLiCnGLl234CHjhz6XSkyHX"  //  #define WIFI\_SSID "INTERLIVE-INTRIAGO"  #define WIFI\_PASSWORD "cjia222020"  int i = 0;  void setup() {  Serial.begin(9600);  // put your setup code here, to run once:  delay(1000);  WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD); //try to connect with wifi  Serial.print("Connecting to ");  Serial.print(WIFI\_SSID);  while (WiFi.status() != WL\_CONNECTED) {  Serial.print(".");  delay(500);  }  Serial.println();  Serial.print("Connected to ");  Serial.println(WIFI\_SSID);  Serial.print("IP Address is : ");  Serial.println(WiFi.localIP()); //print local IP address  Firebase.begin(FIREBASE\_HOST, FIREBASE\_AUTH);  } | |

|  |  |
| --- | --- |
| Función: | void loop() |
| Descripción: |  |
| void loop() {  if (Serial.available()) {  String trama = Serial.readString();  double latitud = getLat(trama);  double longitud = getLon(trama);  String fecha = getDate(trama);  Firebase.setFloat("/Guardias/ExigiCwPrpMtCmcazMREiT4Wazc2/latitud", latitud); //setup path and send readings  if (Firebase.failed()) {  return;  }  Firebase.setFloat("/Guardias/ExigiCwPrpMtCmcazMREiT4Wazc2/longitud", longitud); //setup path and send readings  if (Firebase.failed()) {  return;  }  Firebase.setString("/Guardias/ExigiCwPrpMtCmcazMREiT4Wazc2/date", fecha); //setup path and send readings  if (Firebase.failed()) {  return;  }  }  } | |

|  |  |
| --- | --- |
| Función: | double getLat(String trama) |
| Descripción: |  |
| double getLat(String trama) {  int i = trama.indexOf("@");  int f = trama.indexOf("\*");  String latStr = trama.substring(i + 1, f);  double latitud = latStr.toDouble();  return latitud;  } | |

|  |  |
| --- | --- |
| Función: | double getLon(String trama) |
| Descripción: |  |
| double getLon(String trama) {  int f = trama.indexOf("\*");  int f2 = trama.indexOf("|");  String lonStr = trama.substring(f + 1, f2);  double longitud = lonStr.toDouble();  return longitud;  } | |

|  |  |
| --- | --- |
| Función: | String getDate(String trama) |
| Descripción: |  |
| String getDate(String trama) {  int f2 = trama.indexOf("|");  String fecha = trama.substring(f2 + 1);  return fecha;  } | |

# Código fuente java

|  |  |
| --- | --- |
| Archivo: | AdminMainActivity.java |
| Descripción: |  |
| public class AdminMainActivity extends AppCompatActivity {   String user\_name;  String user\_email;  String user\_photo;  String user\_id;  String user\_phone;  HashMap<String, String> info\_user;  TextView bannerAdminView;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_admin\_main*);   Intent i = getIntent();  info\_user = (HashMap<String, String>) i.getSerializableExtra("info\_user");  user\_name = info\_user.get("user\_name");  user\_email = info\_user.get("user\_email");  user\_photo = info\_user.get("user\_photo");  user\_id = info\_user.get("user\_id");  user\_phone = info\_user.get("user\_phone");  BottomNavigationView bottomNavigation= findViewById(*bottomNavigationViewAdmin*);  bottomNavigation.setOnNavigationItemSelectedListener(navListenerAdmin);  getSupportFragmentManager().beginTransaction().replace(R.id.*containerAdmin*, new MapFragment()).commit();  }   private BottomNavigationView.OnNavigationItemSelectedListener navListenerAdmin = new BottomNavigationView.OnNavigationItemSelectedListener() {  @Override  public boolean onNavigationItemSelected(@NonNull MenuItem item) {  Fragment selectedFragment = null;  FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();   switch (item.getItemId()){  case R.id.*mapsFragment*:  selectedFragment = new MapFragment();  transaction.replace(R.id.*containerAdmin*, selectedFragment);  transaction.addToBackStack(null);  transaction.commit();  break;  case R.id.*reportadminFragment*:  selectedFragment = new ReportesFragment(info\_user,getApplicationContext());  transaction.replace(R.id.*containerAdmin*, selectedFragment);  transaction.addToBackStack(null);  transaction.commit();  break;   case R.id.*outApp*:  cerrarSesion();  //finish();  //System.exit(0);  }   return true;  };  };   public void cerrarSesion(){  FirebaseAuth.*getInstance*().signOut();  finish();  Intent intent = new Intent(this, Login.class);  intent.putExtra("msg", "cerrarSesion");  startActivity(intent);  } } | |

|  |  |
| --- | --- |
| Archivo: | MapFragment.java |
| Descripción: |  |
| public class MapFragment extends Fragment implements OnMapReadyCallback {   Handler handler;  Runnable r;  TextView conn;  GoogleMap mMap;  private SupportMapFragment mMapFragment;     public MapFragment() {  // Required empty public constructor  }      @Override  public View onCreateView(LayoutInflater inflater, ViewGroup container,  Bundle savedInstanceState) {  // Inflate the layout for this fragment  View view = inflater.inflate(R.layout.*fragment\_map*, container, false);  conn = view.findViewById(R.id.*textConectivityAdmin*);  mMapFragment = (SupportMapFragment) getChildFragmentManager().findFragmentById(R.id.*mapAdmin*);  mMapFragment.getMapAsync(this);   checkConn();  return view;  }    @Override  public void onDestroyView() {  killHandler(handler,r);  super.onDestroyView();  }   public void killHandler(Handler handler, Runnable r){  handler.removeCallbacks(r);  }   public void checkConn(){  handler = new Handler();   r = new Runnable() {  public void run() {  ConnectivityManager cm = (ConnectivityManager) getContext().getSystemService(Context.*CONNECTIVITY\_SERVICE*);  NetworkInfo nInfo = cm.getActiveNetworkInfo();  boolean connected = nInfo != null && nInfo.isAvailable() && nInfo.isConnected();  if (connected == true){  conn.setVisibility(View.*INVISIBLE*);  conn.setHeight(0);  }else{  conn.setVisibility(View.*VISIBLE*);  conn.setHeight(80);  }    handler.postDelayed(this, 1000);  }  };   handler.postDelayed(r, 1000);  }   @Override  public void onMapReady(GoogleMap googleMap) {  DataBaseHandler db = new DataBaseHandler(FirebaseDatabase.*getInstance*());  mMap = googleMap;  mMap.setMapType(GoogleMap.*MAP\_TYPE\_SATELLITE*);  db.getMarksDenuncias(getContext(),mMap,getResources());  } } | |

|  |  |
| --- | --- |
| Archivo: | DataBaseHandler |
| Descripción: |  |
| public class DataBaseHandler {  FirebaseDatabase db;   public DataBaseHandler(FirebaseDatabase db) {  this.db = db;  }   public void iniciarSesion(Context contexto,String uid,HashMap<String,String> info\_user,GoogleSignInClient mGoogleSignInClient){  DatabaseReference UserRef = this.db.getReference("Usuarios/"+uid);  DatabaseReference GuardRef = this.db.getReference("Guardias/"+uid);  DatabaseReference AdminRef = this.db.getReference("Administradores/"+uid);   UserRef.addListenerForSingleValueEvent(new ValueEventListener() {  @Override  public void onDataChange(@NonNull DataSnapshot snapshot) {  ConnectivityManager cm = (ConnectivityManager) contexto.getSystemService(Context.*CONNECTIVITY\_SERVICE*);  NetworkInfo nInfo = cm.getActiveNetworkInfo();  boolean connected = nInfo != null && nInfo.isAvailable() && nInfo.isConnected();  if (connected){  if (snapshot.exists()){  Intent intent = new Intent(contexto, MainActivity.class);  intent.putExtra("info\_user", info\_user);  intent.setFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  contexto.startActivity(intent);  }else{  GuardRef.addListenerForSingleValueEvent(new ValueEventListener() {  @Override  public void onDataChange(@NonNull DataSnapshot snapshot1) {  if (snapshot1.exists()){  Intent intent = new Intent(contexto, GuardMainActivity.class);  intent.putExtra("info\_user", info\_user);  intent.setFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  contexto.startActivity(intent);  }else{  AdminRef.addListenerForSingleValueEvent(new ValueEventListener() {  @Override  public void onDataChange(@NonNull DataSnapshot snapshot2) {  if(snapshot2.exists()){  Intent intent = new Intent(contexto, AdminMainActivity.class);  intent.putExtra("info\_user", info\_user);  intent.setFlags(Intent.*FLAG\_ACTIVITY\_NEW\_TASK* | Intent.*FLAG\_ACTIVITY\_CLEAR\_TASK*);  contexto.startActivity(intent);  }else{  Toast.*makeText*(contexto,"Usuario no registrado",Toast.*LENGTH\_LONG*).show();  FirebaseAuth.*getInstance*().signOut();  mGoogleSignInClient.signOut();  }  }   @Override  public void onCancelled(@NonNull DatabaseError error) {   }  });   }  }   @Override  public void onCancelled(@NonNull DatabaseError error) {   }  });  }  }else{  Toast.*makeText*(contexto,"No hay conexión a Internet",Toast.*LENGTH\_LONG*).show();  }  }   @Override  public void onCancelled(@NonNull DatabaseError error) {  Toast.*makeText*(contexto,"No hay conexión a Internet",Toast.*LENGTH\_LONG*).show();  }  });   }   public void registrarUsuario(String uid, Usuario usuario, Context contexto, HashMap<String, String> info\_user){  DatabaseReference userRef = this.db.getReference("Usuarios/"+uid);  userRef.addListenerForSingleValueEvent(new ValueEventListener() {  @Override  public void onDataChange(@NonNull DataSnapshot snapshot) {  ConnectivityManager cm = (ConnectivityManager) contexto.getSystemService(Context.*CONNECTIVITY\_SERVICE*);  NetworkInfo nInfo = cm.getActiveNetworkInfo();  boolean connected = nInfo != null && nInfo.isAvailable() && nInfo.isConnected();  if (connected){  if (snapshot.exists()){  Toast.*makeText*(contexto,"Este correo ya ha sido usado",Toast.*LENGTH\_LONG*).show();  }else{  userRef.setValue(usuario)  .addOnSuccessListener(new OnSuccessListener<Void>() {  @Override  public void onSuccess(Void aVoid) {  // Write was successful!  Toast.*makeText*(contexto,"Usuario registrado correctamente",Toast.*LENGTH\_LONG*).show();  Intent i = new Intent(contexto,MainActivity.class);  i.putExtra("info\_user",info\_user);  contexto.startActivity(i);  }  })  .addOnFailureListener(new OnFailureListener() {  @Override  public void onFailure(@NonNull Exception e) {  // Write failed  Toast.*makeText*(contexto,"Fallo en registro de Usuario",Toast.*LENGTH\_LONG*).show();   }  });  }  }else{  Toast.*makeText*(contexto,"No hay conexión a Internet",Toast.*LENGTH\_LONG*).show();  }  }  @Override  public void onCancelled(@NonNull DatabaseError error) {  Toast.*makeText*(contexto,"No hay conexión a Internet",Toast.*LENGTH\_LONG*).show();  }  });   }   public void registrarGuardia(String uid, Guardia guardia, Context contexto, String token, HashMap<String, String> info\_user){  DatabaseReference userRef = this.db.getReference("Guardias/"+uid);  DatabaseReference tokenRef = this.db.getReference("Tokens/"+token);  tokenRef.addListenerForSingleValueEvent(new ValueEventListener() {  @Override  public void onDataChange(@NonNull DataSnapshot snapshot0) {  ConnectivityManager cm = (ConnectivityManager) contexto.getSystemService(Context.*CONNECTIVITY\_SERVICE*);  NetworkInfo nInfo = cm.getActiveNetworkInfo();  boolean connected = nInfo != null && nInfo.isAvailable() && nInfo.isConnected();  if(connected){  if (snapshot0.exists()){  if (snapshot0.getValue().toString().equals("Guardias")){  userRef.addListenerForSingleValueEvent(new ValueEventListener() {  @Override  public void onDataChange(@NonNull DataSnapshot snapshot) {  if (snapshot.exists()){  Toast.*makeText*(contexto,"Este correo ya ha sido usado",Toast.*LENGTH\_LONG*).show();  }else{  userRef.setValue(guardia)  .addOnSuccessListener(new OnSuccessListener<Void>() {  @Override  public void onSuccess(Void aVoid) {  // Write was successful!  Toast.*makeText*(contexto,"Guardia registrado correctamente",Toast.*LENGTH\_LONG*).show();  Intent i = new Intent(contexto,GuardMainActivity.class);  i.putExtra("info\_user",info\_user);  contexto.startActivity(i);  }  })  .addOnFailureListener(new OnFailureListener() {  @Override  public void onFailure(@NonNull Exception e) {  // Write failed  Toast.*makeText*(contexto,"Fallo en registro de Guardia",Toast.*LENGTH\_LONG*).show();  }  });  }  }  @Override  public void onCancelled(@NonNull DatabaseError error) {   }  });  }  }  }else{  Toast.*makeText*(contexto,"No hay conexión a Internet",Toast.*LENGTH\_LONG*).show();  }   }  @Override  public void onCancelled(@NonNull DatabaseError error) {  Toast.*makeText*(contexto,"No hay conexión a Internet",Toast.*LENGTH\_LONG*).show();  }  });  }   public void registrarAdministrador(String uid, Administrador administrador, Context contexto, String token, HashMap<String, String> info\_user){  DatabaseReference userRef = this.db.getReference("Administradores/"+uid);  DatabaseReference tokenRef = this.db.getReference("Tokens/"+token);  tokenRef.addListenerForSingleValueEvent(new ValueEventListener() {  @Override  public void onDataChange(@NonNull DataSnapshot snapshot0) {  ConnectivityManager cm = (ConnectivityManager) contexto.getSystemService(Context.*CONNECTIVITY\_SERVICE*);  NetworkInfo nInfo = cm.getActiveNetworkInfo();  boolean connected = nInfo != null && nInfo.isAvailable() && nInfo.isConnected();  if (connected){  if (snapshot0.exists()){  if (snapshot0.getValue().toString().equals("Administradores")){  userRef.addListenerForSingleValueEvent(new ValueEventListener() {  @Override  public void onDataChange(@NonNull DataSnapshot snapshot) {  if (snapshot.exists()){  Toast.*makeText*(contexto,"Este correo ya ha sido usado",Toast.*LENGTH\_LONG*).show();  }else{  userRef.setValue(administrador)  .addOnSuccessListener(new OnSuccessListener<Void>() {  @Override  public void onSuccess(Void aVoid) {  // Write was successful!  Toast.*makeText*(contexto,"Administrador registrado correctamente",Toast.*LENGTH\_LONG*).show();  Intent i = new Intent(contexto,AdminMainActivity.class);  i.putExtra("info\_user",info\_user);  contexto.startActivity(i);  }  })  .addOnFailureListener(new OnFailureListener() {  @Override  public void onFailure(@NonNull Exception e) {  // Write failed  Toast.*makeText*(contexto,"Fallo en registro de Administrador",Toast.*LENGTH\_LONG*).show();  }  });  }  }  @Override  public void onCancelled(@NonNull DatabaseError error) {  }  });  }else{  Toast.*makeText*(contexto,"Token invalido",Toast.*LENGTH\_LONG*).show();  }  }else{  Toast.*makeText*(contexto,"Token invalido",Toast.*LENGTH\_LONG*).show();  }  }else{  Toast.*makeText*(contexto,"No hay conexión a Internet",Toast.*LENGTH\_LONG*).show();  }   }  @Override  public void onCancelled(@NonNull DatabaseError error) {  Toast.*makeText*(contexto,"No hay conexión a Internet",Toast.*LENGTH\_LONG*).show();  }  });   }   public void getPhoneNumer(String uid, TextView phoneView){  DatabaseReference userRef = this.db.getReference().child("Usuarios").child(uid).child("telefono");  userRef.addListenerForSingleValueEvent(new ValueEventListener() {  @Override  public void onDataChange(@NonNull DataSnapshot snapshot) {  String number = snapshot.getValue().toString();  phoneView.setText(number);  }   @Override  public void onCancelled(@NonNull DatabaseError error) {   }  });  }   public void getGeoLocalizacion(Context context, String uid, GoogleMap mMap, Resources resources, Location location, boolean incluirUsuario, ArrayList<Marker> guardiasMarkers, Marker[] markerArray, Location locationByGpsEmbed){  DatabaseReference ubicacionRef = this.db.getReference("Ubicacion/"+uid);  DatabaseReference guardiasRef = this.db.getReference("Guardias/");  ArrayList<Marker> markersGuardias = new ArrayList();  ValueEventListener valueEventListenerGuardias = new ValueEventListener() {  ArrayList<Marker> markers = new ArrayList();  @Override  public void onDataChange(@NonNull DataSnapshot snapshot) {   for (Marker m:markers){  m.remove();   } // for (Marker m1:guardiasMarkers){ // m1.remove(); // }  boolean cercanos = false;  for (DataSnapshot guardSnapShot:snapshot.getChildren()){  Double latitud = (Double) guardSnapShot.child("latitud").getValue();  Double longitud = (Double) guardSnapShot.child("longitud").getValue();  LatLng latLngGuard = new LatLng(latitud,longitud);  Marker marker = mMap.addMarker(new MarkerOptions().position(latLngGuard).icon(getBitmapDescriptor(R.drawable.*ic\_guardia*,resources)));  markers.add(marker);  guardiasMarkers.add(marker);  markersGuardias.add(marker);  Double lat1=marker.getPosition().latitude;  Double lat2=location.getLatitud();  Double long1=marker.getPosition().longitude;  Double long2=location.getLongitud();  Double R = 6371e3;  Double phi1 = lat1 \* Math.*PI*/180;  Double phi2 = lat2 \* Math.*PI*/180;  Double deltaphi = (lat2-lat1) \* Math.*PI*/180;  Double deltalambda = (long2-long1) \* Math.*PI*/180;   Double a = Math.*sin*(deltaphi/2) \* Math.*sin*(deltaphi/2) + Math.*cos*(phi1) \* Math.*cos*(phi2) \* Math.*sin*(deltalambda/2) \* Math.*sin*(deltalambda/2);  Double c = 2 \* Math.*atan2*(Math.*sqrt*(a), Math.*sqrt*(1-a));  Double d = R \* c;   cercanos = cercanos || (d<100);  }  if(cercanos==false){  if(incluirUsuario==true){  Toast.*makeText*(context,"No existen guardias cercanos a 100 mts",Toast.*LENGTH\_LONG*).show();  }  }     }  @Override  public void onCancelled(@NonNull DatabaseError error) {}  };   ValueEventListener valueEventListenerUsuario = new ValueEventListener() {   @Override  public void onDataChange(@NonNull DataSnapshot snapshot) {  if (snapshot.exists()) {  if (markerArray[0] != null){  markerArray[0].remove();  }  Double latitud = (Double) snapshot.child("latitud").getValue();  Double longitud = (Double) snapshot.child("longitud").getValue();  location.setLatitud(latitud);  location.setLongitud(longitud);  LatLng posicion = new LatLng(latitud, longitud);   markerArray[0] = mMap.addMarker(new MarkerOptions().position(posicion).title("Estas Aquí"));  mMap.moveCamera(CameraUpdateFactory.*newLatLng*(posicion));  mMap.setMinZoomPreference(18);  if(!incluirUsuario){  if(markerArray[0] != null){  markerArray[0].remove();} // location.setLongitud(locationByGpsEmbed.getLongitud()); // location.setLatitud(locationByGpsEmbed.getLatitud());  }     }  }  @Override  public void onCancelled(@NonNull DatabaseError error) {}   };   guardiasRef.addValueEventListener(valueEventListenerGuardias);  ubicacionRef.addValueEventListener(valueEventListenerUsuario);  }   public void isRegister(Context contexto, String uid, HashMap<String, String> info\_user, GoogleSignInClient googleSignInClient,FirebaseAuth mAuth){  DatabaseReference guardRef = this.db.getReference("Guardias/"+uid);  DatabaseReference adminRef = this.db.getReference("Administradores/"+uid);  DatabaseReference userRef = this.db.getReference("Usuarios/"+uid);   userRef.addListenerForSingleValueEvent(new ValueEventListener() {  @Override  public void onDataChange(@NonNull DataSnapshot snapshot) {  if (snapshot.exists()){  Toast.*makeText*(contexto,"El usuario ya está registrado", Toast.*LENGTH\_LONG*).show();  mAuth.signOut();  googleSignInClient.signOut();   }else{  guardRef.addListenerForSingleValueEvent(new ValueEventListener() {  @Override  public void onDataChange(@NonNull DataSnapshot snapshot1) {  if (snapshot1.exists()){  Toast.*makeText*(contexto,"El usuario ya está registrado", Toast.*LENGTH\_LONG*).show();  mAuth.signOut();  googleSignInClient.signOut();   }else{  adminRef.addListenerForSingleValueEvent(new ValueEventListener() {  @Override  public void onDataChange(@NonNull DataSnapshot snapshot2) {  if(snapshot2.exists()){  Toast.*makeText*(contexto,"El usuario ya está registrado", Toast.*LENGTH\_LONG*).show();  mAuth.signOut();  googleSignInClient.signOut();  }else{  Intent i = new Intent(contexto, RegisterActivity.class);  i.putExtra("info\_user", info\_user);  contexto.startActivity(i);  }  }   @Override  public void onCancelled(@NonNull DatabaseError error) {   }  });  }  }   @Override  public void onCancelled(@NonNull DatabaseError error) {   }  });  }  }   @Override  public void onCancelled(@NonNull DatabaseError error) {   }  });   }   public void enviarReporte(HashMap<String, String> info\_user, Location ubicacion, String descripcion,String lugar, Date date, String tipoDeCrimen,Context contexto){  Usuario user = new Usuario(info\_user.get("user\_name"),info\_user.get("user\_email"),info\_user.get("user\_phone"),info\_user.get("user\_photo"));  final String DATE\_FORMAT = "dd-MM-yyyy kk:mm:ss";  Long tsLong = System.*currentTimeMillis*()/1000;  SimpleDateFormat dateFormat = new SimpleDateFormat(DATE\_FORMAT); // dateFormat.setTimeZone(TimeZone.getTimeZone("UTC"));  String fecha = dateFormat.format(date);  CrimeInfo reporte = new CrimeInfo(user,ubicacion,descripcion,lugar,fecha,tipoDeCrimen,tsLong);  this.db.getReference().child("Denuncia").push().setValue(reporte)  .addOnSuccessListener(new OnSuccessListener<Void>() {  @Override  public void onSuccess(Void aVoid) {  Toast.*makeText*(contexto,"REPORTADO", Toast.*LENGTH\_LONG*).show();   Intent intent = new Intent(contexto, MainActivity.class);  intent.putExtra("info\_user",info\_user);  contexto.startActivity(intent);  }  });  }   public void getDenuncias(Context context, LinearLayout linearLayout, TextView noHayDelitosView,String uid){ // linearLayout.setPadding(0,0,0,50);  this.db.getReference().child("Denuncia").orderByChild("tsLong")  .addValueEventListener(new ValueEventListener() {  @Override  public void onDataChange(@NonNull DataSnapshot snapshot) {  if (snapshot.exists()){  linearLayout.removeAllViews();  for (DataSnapshot dataSnapshot:snapshot.getChildren()){  CrimeInfo crime = dataSnapshot.getValue(CrimeInfo.class);  CardCrime car = printCardView(context,crime);  car.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  Intent i = new Intent(context, CrimeInfoActivity.class);  HashMap<String, String> info\_crime = new HashMap<String, String>();   info\_crime.put("victim\_name",car.getCrime().getUsuario().getNombreCompleto());  info\_crime.put("victim\_email",car.getCrime().getUsuario().getEmail());  info\_crime.put("victim\_photo",car.getCrime().getUsuario().getPhotoUrl());  info\_crime.put("victim\_phone",car.getCrime().getUsuario().getTelefono());  info\_crime.put("victim\_issue",car.getCrime().getDescripcion());  info\_crime.put("victim\_lat",String.*valueOf*(car.getCrime().getUbicacion().getLatitud()));  info\_crime.put("victim\_lon",String.*valueOf*(car.getCrime().getUbicacion().getLongitud()));  info\_crime.put("guardia\_id",uid);  i.putExtra("info\_crime",info\_crime);  context.startActivity(i);  }  });  linearLayout.addView(car);  }  }else{  linearLayout.removeAllViews();  linearLayout.addView(noHayDelitosView);  }  }   @Override  public void onCancelled(@NonNull DatabaseError error) {  Toast.*makeText*(context,"No tiene conexión a internet",Toast.*LENGTH\_LONG*).show();  }  });   }   public CardCrime printCardView(Context context, CrimeInfo crime ){  String nombre = crime.getUsuario().getNombreCompleto();  String descripcion = crime.getDescripcion();  String lugar= crime.getLugar();  String uriPhoto = crime.getUsuario().getPhotoUrl();  String fecha = crime.getFecha();  CardCrime cardView = new CardCrime(context,crime);  RecyclerView.LayoutParams layoutParams = new RecyclerView.LayoutParams(RecyclerView.LayoutParams.*MATCH\_PARENT*, RecyclerView.LayoutParams.*WRAP\_CONTENT*);  layoutParams.bottomMargin = 80;   cardView.setLayoutParams(layoutParams);  cardView.setRadius(15);  cardView.setCardBackgroundColor(Color.*parseColor*("#E4E6E8")); // cardView.setMaxCardElevation(30); // cardView.setMaxCardElevation(6);  LinearLayout linearLayoutH = new LinearLayout(context);  linearLayoutH.setGravity(LinearLayout.*HORIZONTAL*);  LinearLayout.LayoutParams layoutParamsH = new LinearLayout.LayoutParams(LinearLayout.LayoutParams.*MATCH\_PARENT*, LinearLayout.LayoutParams.*WRAP\_CONTENT*);  linearLayoutH.setWeightSum(4);  linearLayoutH.setLayoutParams(layoutParamsH);   ImageView perfil = new ImageView(context);  LinearLayout.LayoutParams layoutParamsIV = new LinearLayout.LayoutParams(0, LinearLayout.LayoutParams.*WRAP\_CONTENT*);  layoutParamsIV.weight = 1;  perfil.setLayoutParams(layoutParamsIV);  linearLayoutH.addView(perfil);  perfil.setMinimumHeight(250); // layoutParamsIV.gravity = Gravity.START;  Picasso.*with*(context).load(uriPhoto).into(perfil);   LinearLayout linearLayoutH1 = new LinearLayout(context);  linearLayoutH1.setGravity(LinearLayout.*VERTICAL*);  LinearLayout.LayoutParams layoutParamsLL = new LinearLayout.LayoutParams(0, LinearLayout.LayoutParams.*WRAP\_CONTENT*);  layoutParamsLL.weight = 3;  linearLayoutH1.setPadding(40,15,5,0);  linearLayoutH1.setLayoutParams(layoutParamsLL);    TextView tx = new TextView(context);  tx.setLayoutParams(layoutParams);  String a = fecha+"\n"+nombre+"\n"+descripcion+"\n"+lugar;  tx.setText(a);  tx.setTextColor(Color.*parseColor*("#F47F25"));  tx.setGravity(Gravity.*CENTER*); // TextView txname = new TextView(getContext()); // txname.setLayoutParams(layoutParams); // txname.setTextColor(Color.BLACK); // txname.setText(nombre);   linearLayoutH1.addView(tx); // linearLayoutH1.addView(txname); // // TextView txdescripcion = new TextView(getContext()); // txdescripcion.setLayoutParams(layoutParams); // txdescripcion.setText(descripcion); // txdescripcion.setTextColor(Color.BLACK); // linearLayoutH1.addView(txdescripcion);   linearLayoutH.addView(linearLayoutH1);    cardView.addView(linearLayoutH);  return cardView; // linearLayoutCards.addView(cardView);  }   public void getMarksDenuncias(Context context, GoogleMap mMap, Resources resources){  DatabaseReference delitosRef = this.db.getReference().child("Denuncia");  delitosRef.addValueEventListener(new ValueEventListener() {  @Override  public void onDataChange(@NonNull DataSnapshot snapshot) {  mMap.clear();  if (snapshot.exists()){  for(DataSnapshot snapshot1:snapshot.getChildren()){   CrimeInfo denuncia = snapshot1.getValue(CrimeInfo.class);   Location location = denuncia.getUbicacion();  LatLng latLng = new LatLng(location.getLatitud(),location.getLongitud());  mMap.addMarker(new MarkerOptions().position(latLng).icon(getBitmapDescriptor(R.drawable.*ic\_person*,resources)).title(denuncia.getUsuario().getNombreCompleto()));  mMap.moveCamera(CameraUpdateFactory.*newLatLng*(latLng));  mMap.setMinZoomPreference(18);  }  }else{  Toast.*makeText*(context,"Aún no hay denuncias registradas",Toast.*LENGTH\_LONG*).show();  }  }   @Override  public void onCancelled(@NonNull DatabaseError error) {   }  });  }   private BitmapDescriptor getBitmapDescriptor(int id, Resources resources) {  if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  VectorDrawable vectorDrawable = (VectorDrawable) resources.getDrawable(id);  int h = vectorDrawable.getIntrinsicHeight();  int w = vectorDrawable.getIntrinsicWidth();  vectorDrawable.setBounds(0, 0, w, h);  Bitmap bm = Bitmap.*createBitmap*(w, h, Bitmap.Config.*ARGB\_8888*);  Canvas canvas = new Canvas(bm);  vectorDrawable.draw(canvas);  return BitmapDescriptorFactory.*fromBitmap*(bm);  } else {  return BitmapDescriptorFactory.*fromResource*(id);  }  }  } | |

|  |  |
| --- | --- |
| Archivo: | ReportesFragment.java |
| Descripción: |  |
| public class GuardMainActivity extends AppCompatActivity {  String user\_name;  String user\_email;  String user\_photo;  String user\_id;  String user\_phone;  HashMap<String, String> info\_user;  TextView bannerGuardView;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_guard\_main*);   Intent i = getIntent();  info\_user = (HashMap<String, String>) i.getSerializableExtra("info\_user");  user\_name = info\_user.get("user\_name");  user\_email = info\_user.get("user\_email");  user\_photo = info\_user.get("user\_photo");  user\_id = info\_user.get("user\_id");  user\_phone = info\_user.get("user\_phone");  BottomNavigationView bottomNavigation= findViewById(*bottomNavigationViewGuard*);  bottomNavigation.setOnNavigationItemSelectedListener(navListenerGuard);  getSupportFragmentManager().beginTransaction().replace(R.id.*containerGuard*, new ReportesFragment(info\_user,getApplicationContext())).commit();  }   private BottomNavigationView.OnNavigationItemSelectedListener navListenerGuard = new BottomNavigationView.OnNavigationItemSelectedListener() {  @Override  public boolean onNavigationItemSelected(@NonNull MenuItem item) {  Fragment selectedFragment = null;  FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();   switch (item.getItemId()){  case R.id.*reportesFragment*:  selectedFragment = new ReportesFragment(info\_user,getApplicationContext());  transaction.replace(R.id.*containerGuard*, selectedFragment);  transaction.addToBackStack(null);  transaction.commit();  break;   case R.id.*outApp*:  cerrarSesion();  //finish();  //System.exit(0);  }   return true;  };  };   public void cerrarSesion(){  FirebaseAuth.*getInstance*().signOut();  finish();  Intent intent = new Intent(this, Login.class);  intent.putExtra("msg", "cerrarSesion");  startActivity(intent);  }     } | |

|  |  |
| --- | --- |
| Archivo: | MainActivity.java |
| Descripción: |  |
| public class MainActivity extends AppCompatActivity {  String user\_name;  String user\_email;  String user\_photo;  String user\_id;  String user\_phone;  HashMap<String, String> info\_user;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_main*);   Intent i = getIntent();  info\_user = (HashMap<String, String>) i.getSerializableExtra("info\_user");  user\_name = info\_user.get("user\_name");  user\_email = info\_user.get("user\_email");  user\_photo = info\_user.get("user\_photo");  user\_id = info\_user.get("user\_id");  user\_phone = info\_user.get("user\_phone");  BottomNavigationView bottomNavigation= findViewById(*bottomNavigationView*);  bottomNavigation.setOnNavigationItemSelectedListener(navListener);  getSupportFragmentManager().beginTransaction().replace(R.id.*container*, new ReportFragment(info\_user)).commit();  }  private BottomNavigationView.OnNavigationItemSelectedListener navListener = new BottomNavigationView.OnNavigationItemSelectedListener() {  @Override  public boolean onNavigationItemSelected(@NonNull MenuItem item) {  Fragment selectedFragment = null;  FragmentTransaction transaction = getSupportFragmentManager().beginTransaction();   switch (item.getItemId()){  case R.id.*reportFragment*:  selectedFragment = new ReportFragment(info\_user);  transaction.replace(R.id.*container*, selectedFragment);  transaction.addToBackStack(null);  transaction.commit();  break;  case R.id.*profileFragment*:  selectedFragment = new ProfileFragment(user\_id,user\_name,user\_email,user\_photo);  transaction.replace(R.id.*container*, selectedFragment);  transaction.addToBackStack(null);  transaction.commit();  break;  case R.id.*outApp*:  cerrarSesion();  //finish();  //System.exit(0);  }   return true;  };  };   public void cerrarSesion(){  FirebaseAuth.*getInstance*().signOut();  finish();  Intent intent = new Intent(this, Login.class);  intent.putExtra("msg", "cerrarSesion");  startActivity(intent);  }  } | |

|  |  |
| --- | --- |
| Archivo: | ProfileFragment.java |
| Descripción: |  |
| public class ProfileFragment extends Fragment {  String user\_name;  String user\_email;  String user\_photo;  String user\_uid;  private TextView nombretxt;  private TextView emailtxt;  private ImageView photoView;  public TextView phoneView;  public ProfileFragment(String uid,String nombreCompleto,String emailCompleto, String uri) {  // Required empty public constructor  this.user\_name = nombreCompleto;  this.user\_email = emailCompleto;  this.user\_photo = uri;  this.user\_uid = uid;  }    @Override  public View onCreateView(LayoutInflater inflater, ViewGroup container,  Bundle savedInstanceState) {  // Inflate the layout for this fragment  View view = inflater.inflate(R.layout.*fragment\_profile*, container, false);  nombretxt = (TextView) view.findViewById(R.id.*txtNombreCompleto*);  emailtxt = (TextView) view.findViewById(R.id.*txtEmail*);  phoneView = (TextView) view.findViewById(R.id.*txtPhone*);  photoView = (ImageView) view.findViewById(R.id.*imv\_foto*);  nombretxt.setText(user\_name);  emailtxt.setText(user\_email); // Glide.with(getActivity()).load(uri).into(imageView);  Picasso.*with*(getContext()).load(user\_photo).into(photoView);  DataBaseHandler db = new DataBaseHandler(FirebaseDatabase.*getInstance*());  db.getPhoneNumer(user\_uid,phoneView);  return view;  } } | |

|  |  |
| --- | --- |
| Archivo: | ReportFragment.java |
| Descripción: |  |
| public class ReportFragment extends Fragment implements OnMapReadyCallback {  private SupportMapFragment mMapFragment;  private GoogleMap mMap;  Geocoder geocoder;  List<Address> addresses;  HashMap<String, String> info\_user;  Location location;  TextView conn;  Handler handler;  Runnable r;  Handler handlerGps;  Runnable rGps;  FusedLocationProviderClient fusedLocationProviderClient;  Marker mainMarker = null;  Button embedGpsBtn;  Button iButtonGpsBtn;  ArrayList<Marker> GuardiasMarkers= new ArrayList();;  Marker ibuttonMarker = null;  final String COLOR\_PRESSED = "#3DDC84";  final String COLOR\_NOT\_PRESSED = "#7F7F7F";  Marker[] markerArray = {null};  Location locationByGpsEmbed = new Location();  boolean primeraVez = true;  String selected = "EMBED";   public ReportFragment(HashMap<String, String> info\_user){  this.info\_user = info\_user;  }   @Override  public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();  StrictMode.*setThreadPolicy*(policy);  View view = inflater.inflate(R.layout.*fragment\_report*, container, false);  embedGpsBtn = (Button) view.findViewById(R.id.*btnGps*);  iButtonGpsBtn =(Button) view.findViewById(R.id.*btnIbutton*);  Button report = (Button) view.findViewById(R.id.*btnReport*);  mMapFragment = (SupportMapFragment) getChildFragmentManager().findFragmentById(R.id.*map*);  mMapFragment.getMapAsync(this);  conn = (TextView) view.findViewById(R.id.*textConectivity*);  location = new Location(0,0);  embedGpsBtn.setPressed(true);  iButtonGpsBtn.setPressed(false);   report.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  killHandler(handler,r);  killHandler(handlerGps,rGps);  // getActivity().finish();  if(selected.equals("EMBED")){  Intent i = new Intent(getContext(), ReportCrime.class);  i.putExtra("info\_user",info\_user);  i.putExtra("user\_latitud",locationByGpsEmbed.getLatitud());  i.putExtra("user\_longitud",locationByGpsEmbed.getLongitud());  startActivity(i);  }else{  Intent i = new Intent(getContext(), ReportCrime.class);  i.putExtra("info\_user",info\_user);  i.putExtra("user\_latitud",location.getLatitud());  i.putExtra("user\_longitud",location.getLongitud());  startActivity(i);} // Toast.makeText(getActivity(), "REPORTADO", Toast.LENGTH\_LONG).show();  }  });  checkConn();    fusedLocationProviderClient = LocationServices.*getFusedLocationProviderClient*(getActivity());  checkGps();    return view;  }   @Override  public void onDestroyView() {  killHandler(handler,r);  killHandler(handlerGps,rGps);  super.onDestroyView();  }   public void killHandler(Handler handler,Runnable r){  handler.removeCallbacks(r);  }    @Override  public void onMapReady(GoogleMap googleMap) {  DataBaseHandler db = new DataBaseHandler(FirebaseDatabase.*getInstance*());  mMap = googleMap;  mMap.setMapType(GoogleMap.*MAP\_TYPE\_SATELLITE*);    embedGpsBtn.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  selected = "EMBED";  embedGpsBtn.setPressed(true);  iButtonGpsBtn.setPressed(false);  embedGpsBtn.setBackgroundColor(Color.*parseColor*(COLOR\_PRESSED));  iButtonGpsBtn.setBackgroundColor(Color.*parseColor*(COLOR\_NOT\_PRESSED));  Toast.*makeText*(getContext(),"Gps Embebido Activado",Toast.*LENGTH\_LONG*).show();  handlerGps.postDelayed(rGps, 1000);  db.getGeoLocalizacion(getContext(),info\_user.get("user\_id"),mMap,getResources(),location,false,GuardiasMarkers,markerArray,locationByGpsEmbed);  }  });   iButtonGpsBtn.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) { // mMap.clear();  selected = "IBUTTON";  embedGpsBtn.setPressed(false);  iButtonGpsBtn.setPressed(true);  embedGpsBtn.setBackgroundColor(Color.*parseColor*(COLOR\_NOT\_PRESSED));  iButtonGpsBtn.setBackgroundColor(Color.*parseColor*(COLOR\_PRESSED));  Toast.*makeText*(getContext(),"Gps iButton Activado",Toast.*LENGTH\_LONG*).show();  handlerGps.removeCallbacks(rGps);  if (mainMarker != null){  mainMarker.remove();  }   db.getGeoLocalizacion(getContext(),info\_user.get("user\_id"),mMap,getResources(),location,true,GuardiasMarkers,markerArray,locationByGpsEmbed);  }  });    }   public void checkGps(){  handlerGps = new Handler();  rGps = new Runnable() {  public void run() {  try{  if (ActivityCompat.*checkSelfPermission*(getActivity(), Manifest.permission.*ACCESS\_FINE\_LOCATION*)== PackageManager.*PERMISSION\_GRANTED*){  fusedLocationProviderClient.getLastLocation().addOnCompleteListener(new OnCompleteListener<android.location.Location>() {  @Override  public void onComplete(@NonNull Task<android.location.Location> task) {  android.location.Location location = task.getResult();  if (location != null){  LatLng posicion = new LatLng(location.getLatitude(),location.getLongitude());  locationByGpsEmbed.setLatitud(location.getLatitude());  locationByGpsEmbed.setLongitud(location.getLongitude());  if (!(location.getLatitude()==0.0) || !(location.getLongitude()==0.0)){  if (primeraVez) {  firstGpsEmbed();  mMap.setMinZoomPreference(18);  primeraVez = false;  }  }  if (mainMarker != null){  mainMarker.remove();  }  mainMarker = mMap.addMarker(new MarkerOptions().position(posicion).title("Estas En el Empalme"));  mMap.moveCamera(CameraUpdateFactory.*newLatLng*(posicion));  }  }  });  }else{  ActivityCompat.*requestPermissions*(getActivity(),new String[]{Manifest.permission.*ACCESS\_FINE\_LOCATION*},44);  }}catch (NullPointerException e){}    handlerGps.postDelayed(this, 1000);  }  };   handlerGps.postDelayed(rGps,1000);  }   public void firstGpsEmbed(){  DataBaseHandler db = new DataBaseHandler(FirebaseDatabase.*getInstance*());  embedGpsBtn.setPressed(true);  iButtonGpsBtn.setPressed(false);  embedGpsBtn.setBackgroundColor(Color.*parseColor*(COLOR\_PRESSED));  iButtonGpsBtn.setBackgroundColor(Color.*parseColor*(COLOR\_NOT\_PRESSED));  Toast.*makeText*(getContext(),"Gps Embebido Activado",Toast.*LENGTH\_LONG*).show();  db.getGeoLocalizacion(getContext(),info\_user.get("user\_id"),mMap,getResources(),location,false,GuardiasMarkers,markerArray,locationByGpsEmbed);  }   public void checkConn(){  handler = new Handler();   r = new Runnable() {  public void run() {  ConnectivityManager cm = (ConnectivityManager) getContext().getSystemService(Context.*CONNECTIVITY\_SERVICE*);  NetworkInfo nInfo = cm.getActiveNetworkInfo();  boolean connected = nInfo != null && nInfo.isAvailable() && nInfo.isConnected();  if (connected == true){  conn.setVisibility(View.*INVISIBLE*);  conn.setHeight(0);  }else{  conn.setVisibility(View.*VISIBLE*);  conn.setHeight(80);  }    handler.postDelayed(this, 1000);  }  };   handler.postDelayed(r, 1000);  }  } | |

|  |  |
| --- | --- |
| Archivo: | CrimeInfoActivity.java |
| Descripción: |  |
| public class CrimeInfoActivity extends AppCompatActivity implements OnMapReadyCallback {  String victim\_name;  String victim\_email;  String victim\_photo;  String victim\_phone;  String issue;  double lat;  double lon;   TextView textVictimName;  TextView textVictimEmail;  TextView textVictimIssue;  TextView textVictimPhone;  ImageView imagePhoto;   public String guardiaUid;  private SupportMapFragment mMapFragment;  private GoogleMap mMap;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_crime\_info*);   textVictimName = findViewById(R.id.*txtVictimName*);  textVictimEmail = findViewById(R.id.*txtVictimEmail*);  textVictimIssue = findViewById(R.id.*txtVictimIssue*);  textVictimPhone = findViewById(R.id.*txtVictimPhone*);  imagePhoto = findViewById(R.id.*imv\_foto*);   Intent i = getIntent();  HashMap<String, String> info\_crime = (HashMap<String, String>) i.getSerializableExtra("info\_crime");  victim\_name = info\_crime.get("victim\_name");  victim\_email = info\_crime.get("victim\_email");  victim\_photo = info\_crime.get("victim\_photo");  victim\_phone = info\_crime.get("victim\_phone");  issue = info\_crime.get("victim\_issue");  lat = Double.*parseDouble*(info\_crime.get("victim\_lat"));  lon = Double.*parseDouble*(info\_crime.get("victim\_lon"));  guardiaUid = info\_crime.get("guardia\_id");  textVictimName.setText(victim\_name);  textVictimEmail.setText(victim\_email);  textVictimPhone.setText(victim\_phone);  textVictimIssue.setText(issue);  Picasso.*with*(getApplicationContext()).load(victim\_photo).into(imagePhoto);  mMapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.*map*);  mMapFragment.getMapAsync(this);  }  @Override  public void onMapReady(GoogleMap googleMap) {  FirebaseDatabase.*getInstance*().getReference().child("Guardias").child(guardiaUid).addListenerForSingleValueEvent(new ValueEventListener() {  @Override  public void onDataChange(@NonNull DataSnapshot snapshot) {  Guardia guardia = snapshot.getValue(Guardia.class);  if (guardia.getLatitud() != null){  mMap = googleMap;  LatLng crime = new LatLng(lat, lon);  LatLng guard = new LatLng(guardia.getLatitud(),guardia.getLongitud());  mMap.addMarker(new MarkerOptions().position(crime).title("Marker in ESPOL"));  mMap.addMarker(new MarkerOptions().position(guard).icon(getBitmapDescriptor(R.drawable.*ic\_guardia*,getResources())).title("Guardia"));  mMap.moveCamera(CameraUpdateFactory.*newLatLng*(crime));  mMap.setMinZoomPreference(18);  }  }   @Override  public void onCancelled(@NonNull DatabaseError error) {   }  });   }  private BitmapDescriptor getBitmapDescriptor(int id, Resources resources) {  if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*LOLLIPOP*) {  VectorDrawable vectorDrawable = (VectorDrawable) resources.getDrawable(id);  int h = vectorDrawable.getIntrinsicHeight();  int w = vectorDrawable.getIntrinsicWidth();  vectorDrawable.setBounds(0, 0, w, h);  Bitmap bm = Bitmap.*createBitmap*(w, h, Bitmap.Config.*ARGB\_8888*);  Canvas canvas = new Canvas(bm);  vectorDrawable.draw(canvas);  return BitmapDescriptorFactory.*fromBitmap*(bm);  } else {  return BitmapDescriptorFactory.*fromResource*(id);  }  } } | |

|  |  |
| --- | --- |
| Archivo: | Login |
| Descripción: |  |
| public class Login extends AppCompatActivity {  static final int *GOOGLE\_SIGN\_IN* = 123;  FirebaseAuth mAuth;  GoogleSignInClient mGoogleSignInClient;  Button btn\_login;  boolean estaRegistrando = false;  boolean estaLogeando = false;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_login*);  mAuth = FirebaseAuth.*getInstance*();  GoogleSignInOptions gso = new GoogleSignInOptions.Builder(GoogleSignInOptions.*DEFAULT\_SIGN\_IN*)  .requestIdToken(getString(R.string.*default\_web\_client\_id*))  .requestEmail()  .build();  mGoogleSignInClient = GoogleSignIn.*getClient*(this, gso);   Intent intent = getIntent();  String msg = intent.getStringExtra("msg");  if(msg != null){  if(msg.equals("cerrarSesion")){  cerrarSesion();  }  }  if (mAuth.getCurrentUser() != null) {  // User is signed in  DataBaseHandler db = new DataBaseHandler(FirebaseDatabase.*getInstance*());  FirebaseUser user = mAuth.getCurrentUser();  HashMap<String, String> info\_user = new HashMap<String, String>();  info\_user.put("user\_name", user.getDisplayName());  info\_user.put("user\_email", user.getEmail());  info\_user.put("user\_photo", String.*valueOf*(user.getPhotoUrl()));  info\_user.put("user\_id", user.getUid());  info\_user.put("user\_phone", user.getPhoneNumber());  db.iniciarSesion(this,info\_user.get("user\_id"),info\_user,mGoogleSignInClient); // Intent i = new Intent(this, MainActivity.class); // i.putExtra("info\_user",info\_user); // i.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK | Intent.FLAG\_ACTIVITY\_CLEAR\_TASK); // startActivity(i);  }  }   private void cerrarSesion() {  mGoogleSignInClient.signOut().addOnCompleteListener(this,  task -> updateUI(null));  }   public void iniciarSesion(View view) {  ConnectivityManager cm = (ConnectivityManager) getApplicationContext().getSystemService(Context.*CONNECTIVITY\_SERVICE*);  NetworkInfo nInfo = cm.getActiveNetworkInfo();  boolean connected = nInfo != null && nInfo.isAvailable() && nInfo.isConnected();  if(connected){  estaLogeando = true;  Intent signInIntent = mGoogleSignInClient.getSignInIntent();  startActivityForResult(signInIntent, *GOOGLE\_SIGN\_IN*);  }else{  Toast.*makeText*(getApplicationContext(),"No hay Conexión a Internet",Toast.*LENGTH\_LONG*).show();  }  }   public void registrarUsuario(View v){  ConnectivityManager cm = (ConnectivityManager) getApplicationContext().getSystemService(Context.*CONNECTIVITY\_SERVICE*);  NetworkInfo nInfo = cm.getActiveNetworkInfo();  boolean connected = nInfo != null && nInfo.isAvailable() && nInfo.isConnected();  if(connected){  estaRegistrando = true;  Intent signInIntent = mGoogleSignInClient.getSignInIntent();  startActivityForResult(signInIntent, *GOOGLE\_SIGN\_IN*);  }else{  Toast.*makeText*(getApplicationContext(),"No hay Conexión a Internet",Toast.*LENGTH\_LONG*).show();  }  }   @Override  protected void onActivityResult(int requestCode, int resultCode, Intent data) {  super.onActivityResult(requestCode, resultCode, data);  if (requestCode == *GOOGLE\_SIGN\_IN*) {  Task<GoogleSignInAccount> task = GoogleSignIn.*getSignedInAccountFromIntent*(  data);  try {  GoogleSignInAccount account = task.getResult(ApiException.class);  if (account != null) firebaseAuthWithGoogle(account);  } catch (ApiException e) {  Log.*w*("TAG", "Fallo el inicio de sesión con google.", e);  }  }  }   private void firebaseAuthWithGoogle(GoogleSignInAccount acct) {  Log.*d*("TAG", "firebaseAuthWithGoogle:" + acct.getId());  AuthCredential credential = GoogleAuthProvider.*getCredential*(acct.getIdToken(),  null);  mAuth.signInWithCredential(credential)  .addOnCompleteListener(this, task -> {  if (task.isSuccessful()) {  FirebaseUser user = mAuth.getCurrentUser();  updateUI(user);  } else {  updateUI(null);  }  });  }   private void updateUI(FirebaseUser user) { | |

|  |  |
| --- | --- |
| Archivo: | RegisterActivity |
| Descripción: |  |
| public class RegisterActivity extends AppCompatActivity implements AdapterView.OnItemSelectedListener {  String user\_name;  String user\_email;  String user\_photo;  String user\_id;  String user\_phone;  private TextView nombretxt;  private TextView emailtxt;  private ImageView photoView;  private EditText phoneView;  private Spinner spinner;  public String tipoDeUsuarioSeleccionado;  private EditText tokenText;  private TextView tokenView;  HashMap<String, String> info\_user;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_register*);   Intent i = getIntent();  info\_user = (HashMap<String, String>) i.getSerializableExtra("info\_user");  user\_name = info\_user.get("user\_name");  user\_email = info\_user.get("user\_email");  user\_photo = info\_user.get("user\_photo");  user\_id = info\_user.get("user\_id");  user\_phone = info\_user.get("user\_phone");   nombretxt = (TextView) findViewById(R.id.*txtNombreCompletoRA*);  emailtxt = (TextView) findViewById(R.id.*txtEmailRA*);  photoView = (ImageView) findViewById(R.id.*imv\_fotoRA*);  phoneView = (EditText) findViewById(R.id.*txtPhoneRA*);  nombretxt.setText(user\_name);  emailtxt.setText(user\_email);  phoneView.setText(user\_phone);  Picasso.*with*(this).load(user\_photo).into(photoView);   spinner = (Spinner) findViewById(R.id.*spinerTipoUsuario*);  ArrayAdapter<CharSequence> adapter = ArrayAdapter.*createFromResource*(this,R.array.*tiposDeUsuario*, android.R.layout.*simple\_spinner\_item*);  adapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);  spinner.setAdapter(adapter);  spinner.setOnItemSelectedListener(this);   tokenText = findViewById(R.id.*txtToken*);  tokenView = findViewById(R.id.*tokenViews*);    }    public void registrar(View v){  DataBaseHandler db = new DataBaseHandler(FirebaseDatabase.*getInstance*());  if (spinner.getSelectedItem().toString().equals("Usuarios")){  Usuario user;  if (user\_phone == null){  user = new Usuario(user\_name,user\_email,phoneView.getText().toString(),user\_photo);  }else{  user = new Usuario(user\_name,user\_email,user\_phone,user\_photo);  }  db.registrarUsuario(user\_id,user,this,info\_user);  }else if(spinner.getSelectedItem().toString().equals("Guardias")){  Guardia guardia;  if (user\_phone == null){  guardia = new Guardia(user\_name,user\_email,phoneView.getText().toString(),user\_photo,-2.145983, -79.965755);  }else{  guardia = new Guardia(user\_name,user\_email,user\_phone,user\_photo,-2.145983, -79.965755);  }  db.registrarGuardia(user\_id,guardia,this,tokenText.getText().toString(),info\_user);  }else if(spinner.getSelectedItem().toString().equals("Administradores")){  Administrador admin;  if (user\_phone == null){  admin = new Administrador(user\_name,user\_email,phoneView.getText().toString(),user\_photo);  }else{  admin = new Administrador(user\_name,user\_email,user\_phone,user\_photo);  }  db.registrarAdministrador(user\_id,admin,this,tokenText.getText().toString(),info\_user);  }  }  @Override  public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {  tipoDeUsuarioSeleccionado = (String) adapterView.getItemAtPosition(i);  if (tipoDeUsuarioSeleccionado.equals("Usuarios")){  tokenView.setVisibility(View.INVISIBLE);  tokenText.setVisibility(View.INVISIBLE);  }  else{  tokenView.setVisibility(View.VISIBLE);  tokenText.setVisibility(View.VISIBLE);  }  }   @Override  public void onNothingSelected(AdapterView<?> adapterView) {   } } | |

# Layouts

## Login

|  |  |
| --- | --- |
| Nombre: | Activity\_login.xml |
| Descripción: | Pantalla donde el usuario puede acceder a la aplicación con su cuenta de Google o acceder al registro. |
| <?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="#00ACC1"  android:gravity="center"  android:orientation="vertical"  tools:context="com.twoReport.project.Login">   <ImageView  android:id="@+id/imgLogin"  android:layout\_width="match\_parent"  android:layout\_height="230dp"  android:layout\_weight="1"  android:src="@drawable/logof" />   <EditText  android:id="@+id/txtUsuario"  android:layout\_width="300dp"  android:layout\_height="45dp"  android:background="#D9FDFBFB"  android:drawableLeft="@drawable/ic\_profile"  android:paddingLeft="20dp"  android:ems="10"  android:hint="Usuario"  android:inputType="textPersonName"  android:textAlignment="center"  android:textColor="#FFFFFFFF"/>   <EditText  android:id="@+id/txtContra"  android:layout\_width="300dp"  android:layout\_height="45dp"  android:layout\_marginTop="20dp"  android:background="#D9FDFBFB"  android:paddingLeft="20dp"  android:drawableLeft="@android:drawable/ic\_secure"  android:hint="Contraseña"  android:inputType="textPassword"  android:textColor="#FFFFFFFF"  android:textAlignment="center" />    <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:gravity="center"  android:orientation="vertical">   <Button  android:id="@+id/btn\_login"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:onClick="iniciarSesion"  android:layout\_marginTop="20dp"  android:layout\_marginBottom="20dp"  android:textAlignment="center"  android:text="Iniciar sesion con Google" />   <Button  android:id="@+id/btn\_register"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:onClick="registrarUsuario"  android:layout\_marginTop="0dp"  android:layout\_marginBottom="50dp"  android:textAlignment="center"  android:text="Registrarse con Google" />   </LinearLayout> </LinearLayout> | |

## Registro

|  |  |
| --- | --- |
| Nombre: | Activity\_register.xml |
| Descripción: | Pantalla que permite el registro de un usuario a través de una cuenta de Google. |
| <?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:paddingBottom="@dimen/activity\_vertical\_margin"  android:paddingLeft="@dimen/activity\_horizontal\_margin"  android:paddingRight="@dimen/activity\_horizontal\_margin"  android:paddingTop="@dimen/activity\_vertical\_margin"  android:orientation="vertical"  tools:context="com.twoReport.project.RegisterActivity" >    <TextView  android:id="@+id/txtProfile"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:gravity="center\_horizontal"  android:padding="10dp"  android:textSize="24dp"  android:elevation="8dp"  android:background="@android:color/white"  android:text="@string/register\_title" />   <ImageView  android:id="@+id/imv\_fotoRA"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:minWidth="150dp"  android:minHeight="150dp"  android:layout\_gravity="center"  tools:srcCompat="@tools:sample/avatars" />   <ScrollView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1">  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_margin="0dp"  android:orientation="vertical"  android:padding="0dp"  android:layout\_weight="1">   <TextView  android:id="@+id/textView2"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:paddingLeft="20sp"  android:layout\_marginTop="20dp"  android:text="@string/nombre\_completo\_text"  android:textColor="#2196F3"  android:textSize="18sp" />   <TextView  android:id="@+id/txtNombreCompletoRA"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:ems="10"  android:inputType="textPersonName"  android:paddingLeft="40dp"  android:text="NOMBRE APELLIDO" />   <TextView  android:id="@+id/textView3"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:paddingLeft="20dp"  android:text="@string/email\_text"  android:textColor="#2196F3"  android:textSize="18sp" />   <TextView  android:id="@+id/txtEmailRA"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:ems="10"  android:inputType="textPersonName"  android:paddingLeft="40dp"  android:text="CORREO ELECTRONICO" />     <TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:paddingLeft="20dp"  android:text="@string/telefono\_contacto\_text"  android:textColor="#2196F3"  android:textSize="18sp" />   <EditText  android:id="@+id/txtPhoneRA"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="40dp"  android:layout\_marginTop="0dp"  android:layout\_marginRight="40dp"  android:ems="10"  android:hint="Teléfono de emergencia"  android:inputType="textPersonName"  android:textColor="#000000"  android:textColorLink="#FFFFFF" />   <TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:paddingLeft="20dp"  android:text="@string/tipo\_usuario\_text"  android:textColor="#2196F3"  android:textSize="18sp" />   <Spinner  android:id="@+id/spinerTipoUsuario"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_alignParentTop="true"  android:layout\_centerHorizontal="true"  android:layout\_marginTop="20dp"  android:layout\_marginLeft="40dp"  android:layout\_marginRight="40dp"/>  <TextView  android:id="@+id/tokenViews"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:paddingLeft="20dp"  android:text="@string/token\_text"  android:textColor="#2196F3"  android:textSize="18sp" />   <EditText  android:id="@+id/txtToken"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="40dp"  android:layout\_marginTop="0dp"  android:layout\_marginRight="40dp"  android:ems="10"  android:hint="@string/token\_input\_text"  android:inputType="textPersonName"  android:textColor="#000000" />   </LinearLayout>  </ScrollView>   <Button  android:id="@+id/button"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="20dp"  android:onClick="registrar"  android:text="@string/registrar\_usuario" />  </LinearLayout> | |

## Estudiante

|  |  |
| --- | --- |
| Nombre: | activity\_main.xml |
| Descripción: | Bottom navigation bar con las opciones disponibles para un usuario estudiante. |
| <?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="com.twoReport.project.UsuarioPackage.MainActivity">   <FrameLayout  android:layout\_width="match\_parent"  android:layout\_gravity="top"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:id="@+id/container" />   <com.google.android.material.bottomnavigation.BottomNavigationView  android:id="@+id/bottomNavigationView"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:background="#fff"  android:layout\_below="@+id/container"  app:menu="@menu/bottom\_menu" />  </LinearLayout> | |

|  |  |
| --- | --- |
| Método: | activity\_report\_crime.xml |
| Descripción: | Pantalla que muestra el formulario que el estudiante debe llenar para reportar un crimen. |
| <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"  tools:context="com.twoReport.project.ReportCrime">   <TextView  android:id="@+id/txtProfile"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:gravity="center\_horizontal"  android:padding="10dp"  android:textSize="24dp"  android:elevation="8dp"  android:background="@android:color/white"  android:text="@string/report\_crime\_text" />  <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_margin="0dp"  android:orientation="vertical"  android:padding="0dp"  android:layout\_weight="1">   <TextView  android:id="@+id/textView2"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:paddingLeft="20sp"  android:text="@string/tipo\_delito\_text"  android:layout\_marginTop="50dp"  android:textColor="#2196F3"  android:textSize="18sp" />   <Spinner  android:id="@+id/spinnerTipoDelito"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="10dp"/>   <TextView  android:id="@+id/textView6"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:paddingLeft="20sp"  android:text="@string/descripcion\_text"  android:textColor="#2196F3"  android:textSize="18sp"  android:layout\_marginTop="20dp" />   <EditText  android:id="@+id/txtDescripcion"  android:layout\_width="match\_parent"  android:layout\_height="100dp"  android:layout\_marginLeft="20dp"  android:layout\_marginRight="20dp"  android:ems="10"  android:gravity="start|top"  android:inputType="textMultiLine" />   <TextView  android:id="@+id/textView7"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:paddingLeft="20sp"  android:text="@string/location\_descripcion\_text"  android:textColor="#2196F3"  android:textSize="18sp"  android:layout\_marginTop="20dp" />   <EditText  android:id="@+id/txtDescripcionLugar"  android:layout\_width="match\_parent"  android:layout\_height="100dp"  android:layout\_marginLeft="20dp"  android:layout\_marginRight="20dp"  android:ems="10"  android:gravity="start|top"  android:inputType="textMultiLine" />   <Button  android:id="@+id/btnReportar"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginVertical="30dp"  android:onClick="reportar"  android:text="REPORTAR AHORA"  android:layout\_marginRight="20dp"  android:layout\_marginLeft="20dp"/>/>   </LinearLayout>    </LinearLayout> | |

|  |  |
| --- | --- |
| Nombre: | Fragment\_profile.xml |
| Descripción: | Pantalla para mostrar la información del perfil del usuario. |
| <?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="com.twoReport.project.UsuarioPackage.ProfileFragment" >   <TextView  android:id="@+id/txtProfile"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:gravity="center\_horizontal"  android:padding="10dp"  android:textSize="24dp"  android:elevation="8dp"  android:background="@android:color/white"  android:text="@string/profile\_title" />   <ImageView  android:id="@+id/imv\_foto"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:minWidth="150dp"  android:minHeight="150dp"  android:layout\_gravity="center"  tools:srcCompat="@tools:sample/avatars" />   <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_margin="0dp"  android:orientation="vertical"  android:padding="0dp"  android:layout\_weight="1">   <TextView  android:id="@+id/textView2"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:paddingLeft="20sp"  android:text="@string/nombre\_completo\_text"  android:textColor="#2196F3"  android:textSize="18sp" />   <TextView  android:id="@+id/txtNombreCompleto"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:ems="10"  android:inputType="textPersonName"  android:paddingLeft="40dp"  android:text="NOMBRE APELLIDO" />   <TextView  android:id="@+id/textView3"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:paddingLeft="20dp"  android:text="@string/email\_text"  android:textColor="#2196F3"  android:textSize="18sp" />   <TextView  android:id="@+id/txtEmail"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:ems="10"  android:inputType="textPersonName"  android:paddingLeft="40dp"  android:text="CORREO ELECTRONICO" />     <TextView  android:id="@+id/textView5"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:paddingLeft="20dp"  android:text="@string/telefono\_contacto\_text"  android:textColor="#2196F3"  android:textSize="18sp" />   <TextView  android:id="@+id/txtPhone"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="40dp"  android:layout\_marginTop="20dp"  android:layout\_marginRight="40dp"  android:ems="10"  android:hint="Teléfono de emergencia"  android:inputType="textPersonName" />  </LinearLayout>   <Button  android:id="@+id/button"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="50dp"  android:text="@string/update\_perfil" />  </LinearLayout> | |

|  |  |
| --- | --- |
| Nombre: | Fragment\_report.xml |
| Descripción: | Pantalla principal del usuario estudiante. |
| <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="com.twoReport.project.UsuarioPackage.ReportFragment">   <!-- *TODO: Update blank fragment layout* -->  <LinearLayout  android:layout\_height="wrap\_content"  android:layout\_width="match\_parent"  android:orientation="horizontal"  android:weightSum="2">   <Button  android:id="@+id/btnGps"  android:layout\_width="0dp"  android:layout\_weight="1"  android:layout\_height="wrap\_content"  android:layout\_margin="0dp"  android:insetTop="0dp"  android:insetBottom="0dp"  android:text="@string/SelectionGpsEmbed"  app:backgroundTint="#3DDC84"/>  <Button  android:id="@+id/btnIbutton"  android:layout\_width="0dp"  android:layout\_weight="1"  android:layout\_height="wrap\_content"  android:layout\_margin="0dp"  android:insetTop="0dp"  android:insetBottom="0dp"  android:text="@string/SelectionIbutton"  app:backgroundTint="#7F7F7F"/>  </LinearLayout>  <TextView  android:id="@+id/textConectivity"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="0dp"  android:paddingLeft="20sp"  android:text="No hay conexión a internet"  android:textAlignment="center"  android:textColor="#F44336"  android:textSize="18sp" />   <fragment  android:id="@+id/map"  android:name="com.google.android.gms.maps.SupportMapFragment"  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_gravity="top"  android:layout\_margin="0dp"  android:layout\_weight="1"  android:orientation="vertical" />   <LinearLayout  android:id="@+id/linear"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_margin="0dp"  android:orientation="vertical"  android:padding="0dp">   <Button  android:id="@+id/btnReport"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_margin="0dp"  android:insetTop="0dp"  android:insetBottom="0dp"  android:text="@string/reportar"  app:backgroundTint="#F44336"  app:elevation="2dp"  app:iconPadding="0dp" />  </LinearLayout> </LinearLayout> | |

## Guardia

|  |  |
| --- | --- |
| Nombre: | Activity\_guard\_main.xml |
| Descripción: | Bottom navigation bar de un usuario guardia |
| <?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="com.twoReport.project.GuardiaPackage.GuardMainActivity">   <FrameLayout  android:layout\_width="match\_parent"  android:layout\_gravity="top"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:id="@+id/containerGuard" />   <com.google.android.material.bottomnavigation.BottomNavigationView  android:id="@+id/bottomNavigationViewGuard"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:background="#fff"  android:layout\_below="@+id/containerGuard"  app:menu="@menu/bottom\_menu\_guard" /> </LinearLayout> | |

|  |  |
| --- | --- |
| Nombre: | Fragment\_reportes.xml |
| Descripción: | Pantalla que muestra la lista de crímenes reportados |
| <?xml version="1.0" encoding="utf-8"?> <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"  tools:context="com.twoReport.project.GuardiaPackage.ReportesFragment">   <TextView  android:id="@+id/txtBannerGuard"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:gravity="center\_horizontal"  android:padding="10dp"  android:textSize="24dp"  android:elevation="8dp"  android:background="@android:color/white" />  <TextView  android:id="@+id/textConectivityGuard"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="No hay conexión a internet"  android:textAlignment="center"  android:textColor="#F44336"  android:textSize="18sp" />  <ScrollView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:paddingLeft="20dp"  android:paddingRight="20dp"  android:paddingTop="20dp">   <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:id="@+id/linearLayoutCards"  android:orientation="vertical">   </LinearLayout>  </ScrollView>  </LinearLayout> | |

## Administrador

|  |  |
| --- | --- |
| Nombre: | Activity\_admin\_main.xml |
| Descripción: | Botton navigation bar de un usuario administrador |
| <?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="com.twoReport.project.AdministradorPackage.AdminMainActivity">   <FrameLayout  android:layout\_width="match\_parent"  android:layout\_gravity="top"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:id="@+id/containerAdmin" />   <com.google.android.material.bottomnavigation.BottomNavigationView  android:id="@+id/bottomNavigationViewAdmin"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:background="#fff"  android:layout\_below="@+id/containerAdmin"  app:menu="@menu/bottom\_menu\_admin" />  </LinearLayout> | |

|  |  |
| --- | --- |
| Nombre: | Fragment\_map.xml |
| Descripción: | Mapa |
| <?xml version="1.0" encoding="utf-8"?> <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"  tools:context="com.twoReport.project.AdministradorPackage.MapFragment">   <!-- *TODO: Update blank fragment layout* -->  <TextView  android:id="@+id/textConectivityAdmin"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="0dp"  android:paddingLeft="20sp"  android:text="No hay conexión a internet"  android:textAlignment="center"  android:textColor="#F44336"  android:textSize="18sp" />   <fragment  android:id="@+id/mapAdmin"  android:name="com.google.android.gms.maps.SupportMapFragment"  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_gravity="top"  android:layout\_margin="0dp"  android:layout\_weight="1"  android:orientation="vertical" />  </LinearLayout> | |

## Reportes

|  |  |
| --- | --- |
| Nombre: | Activity\_crime\_info.xml |
| Descripción: | Pantalla que muestra la información de un delito reportado |
| <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="com.twoReport.project.CrimeInfoActivity" >   <TextView  android:id="@+id/txtProfile"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:gravity="center\_horizontal"  android:padding="10dp"  android:textSize="24dp"  android:elevation="8dp"  android:background="@android:color/white"  android:text="@string/crime\_detail" />   <ImageView  android:id="@+id/imv\_foto"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_gravity="center"  android:minWidth="150dp"  android:minHeight="150dp"  tools:srcCompat="@tools:sample/avatars" />   <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_margin="0dp"  android:orientation="vertical"  android:padding="0dp"  android:layout\_weight="1">   <TextView  android:id="@+id/textView2"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:paddingLeft="20sp"  android:text="@string/victim\_name"  android:textColor="#2196F3"  android:textSize="18sp" />   <TextView  android:id="@+id/txtVictimName"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:ems="10"  android:paddingStart="40dp"  android:text="NOMBRE APELLIDO" />   <TextView  android:id="@+id/textView3"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:paddingStart="20dp"  android:text="@string/victim\_email"  android:textColor="#2196F3"  android:textSize="18sp" />   <TextView  android:id="@+id/txtVictimEmail"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:ems="10"  android:inputType="textPersonName"  android:paddingLeft="40dp"  android:text="CORREO ELECTRONICO" />   <TextView  android:id="@+id/textView5"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:paddingLeft="20dp"  android:text="@string/telefono\_contacto\_text"  android:textColor="#2196F3"  android:textSize="18sp" />   <TextView  android:id="@+id/txtVictimPhone"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:ems="10"  android:paddingStart="40dp"  android:text="TELEFONO DE EMERGENCIA" />   <TextView  android:id="@+id/textView4"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:paddingStart="20dp"  android:text="@string/victim\_issue"  android:textColor="#2196F3"  android:textSize="18sp" />   <TextView  android:id="@+id/txtVictimIssue"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="20dp"  android:layout\_marginBottom="20dp"  android:ems="10"  android:paddingStart="40dp"  android:text="DEMANDA" />   <TextView  android:id="@+id/location"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="0dp"  android:paddingStart="20dp"  android:text="@string/locate"  android:textColor="#2196F3"  android:textSize="18sp" />   <fragment  android:id="@+id/map"  android:name="com.google.android.gms.maps.SupportMapFragment"  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_gravity="top"  android:layout\_margin="20dp"  android:layout\_weight="1"  android:orientation="vertical" />    </LinearLayout>  </LinearLayout> | |

Buenos días, somos el grupo #1 de la materia Itinerario Aplicaciones móviles y servicios telemáticos, nuestro proyecto es Localización de guarias para denuncias. Este proyecto surge debido a que no existe una forma inmediata de denunciar los delitos que se pueden llegar a suscitar dentro de las instalaciones de Espol.

De aquí nace 2Report una aplicación que muestra la localización en tiempo real de los guardias que se encuentren más cercanos a tu ubicación y en caso de ser víctima de algún delito, se puede realizar denuncias en un tiempo menor a 30segundos con el fin de alertar a los guardias más cercanos para que puedan acudir a tu ayuda.