

Safe Routes App



Problematica

La delincuencia urbana se ha incrementado en los últimos años, generando inseguridad en las calles y temor en la población. La falta de iluminación, vigilancia y datos actualizados sobre zonas de riesgo agrava el problema. SafeRoute surge como una alternativa tecnológica para identificar rutas seguras y reducir la exposición a hechos delictivos.

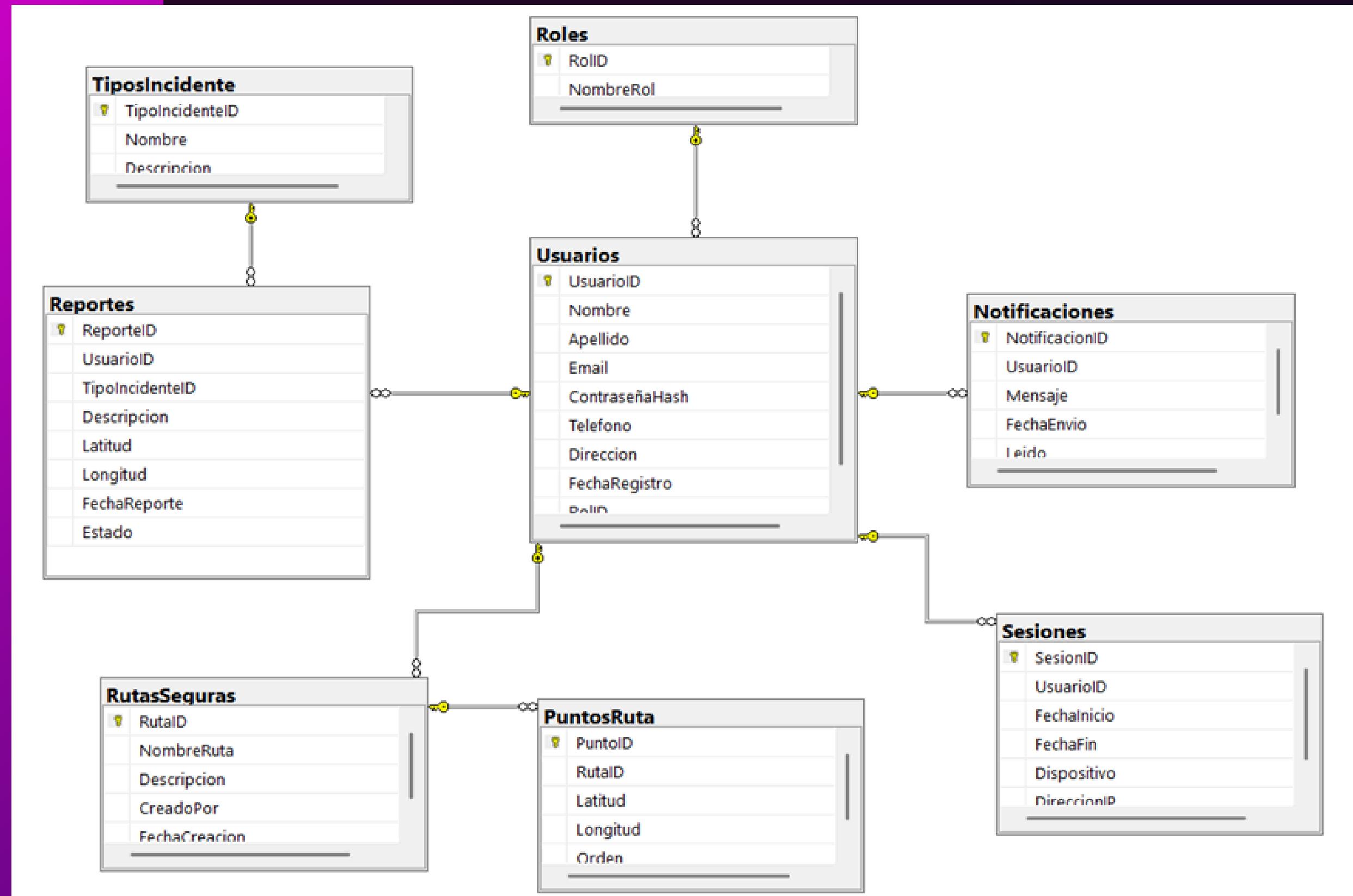


shutterstock.com · 1788751325

SPRINT

Nº1

Diseñar modelo entidad-relación y Crear tablas

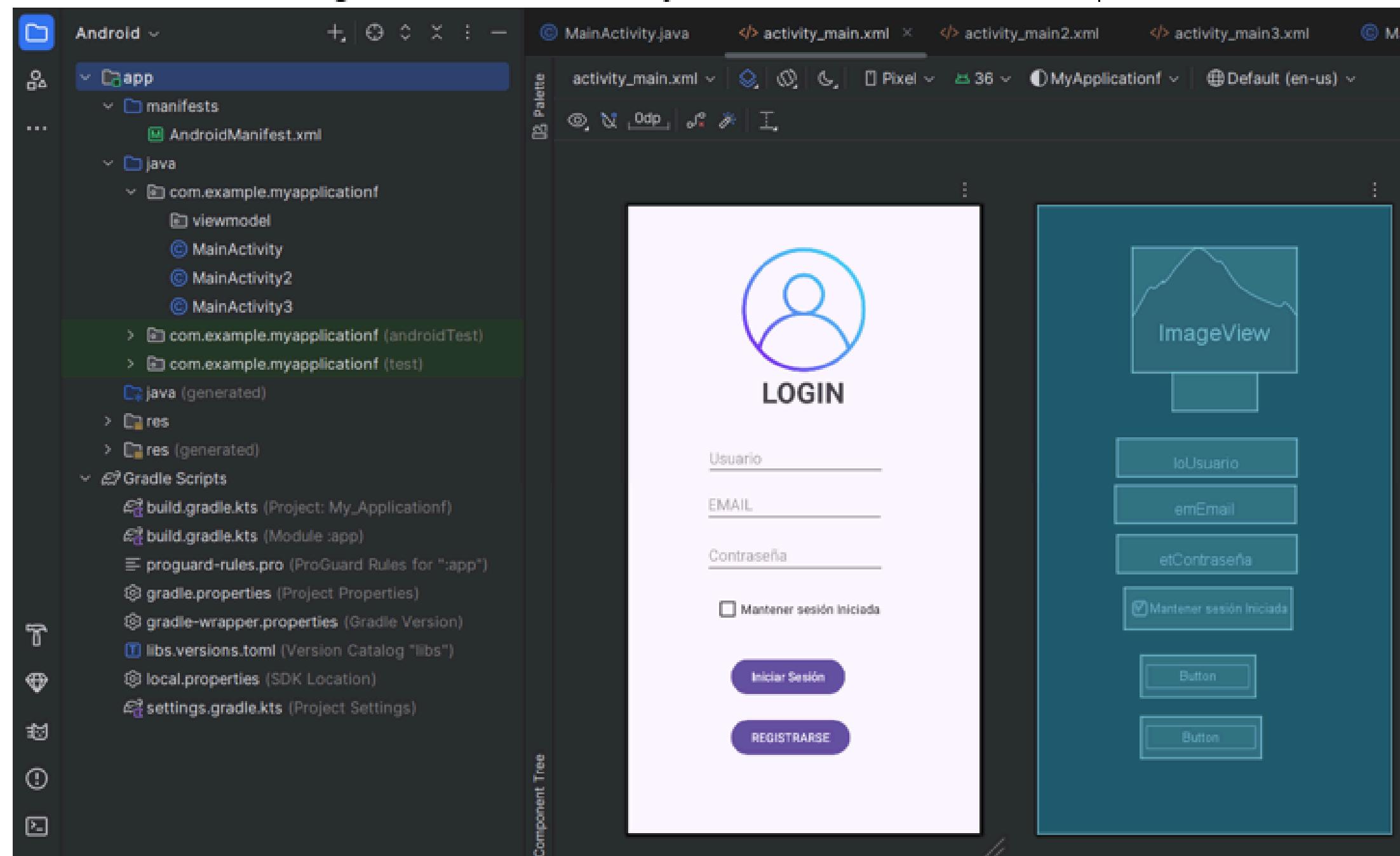


● Configurar conexión backend

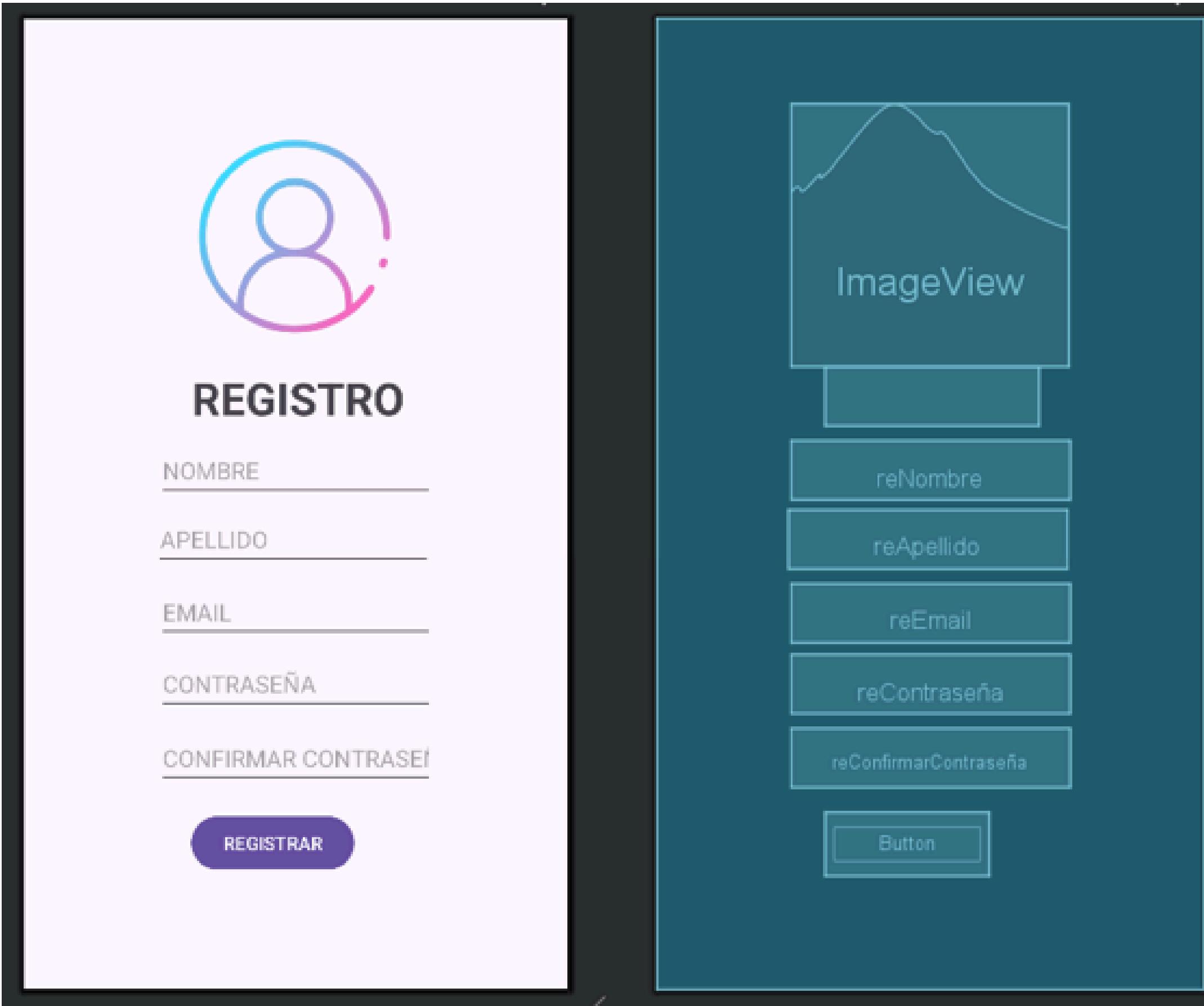
El proyecto es una aplicación móvil desarrollada en Android Studio con Java, que cuenta con 3 Activities principales y sus respectivos archivos XML de diseño:

MainActivity (activity_main.xml):

- Pantalla de inicio de sesión.
- Permite al usuario ingresar con correo y contraseña usando Firebase Authentication.
- Contiene la lógica de autenticación y validación de credenciales.



● Diseñar formulario de registro



● Validar campos obligatorios

About Us

Contact



10:52 □ 🔍 ⏪

10:52 □ 🔍 ⏪

10:52 □ 🔍 ⏪

100 %

100 %



LOGIN

Usuario

EMAIL

Contraseña

Mantener sesión Iniciada

Iniciar Sesión

REGISTRARSE

Ingrese email y contraseña



REGISTRO

NOMBRE

APELLIDO

EMAIL

CONTRASEÑA

CONFIRMAR CONTRASEÑA

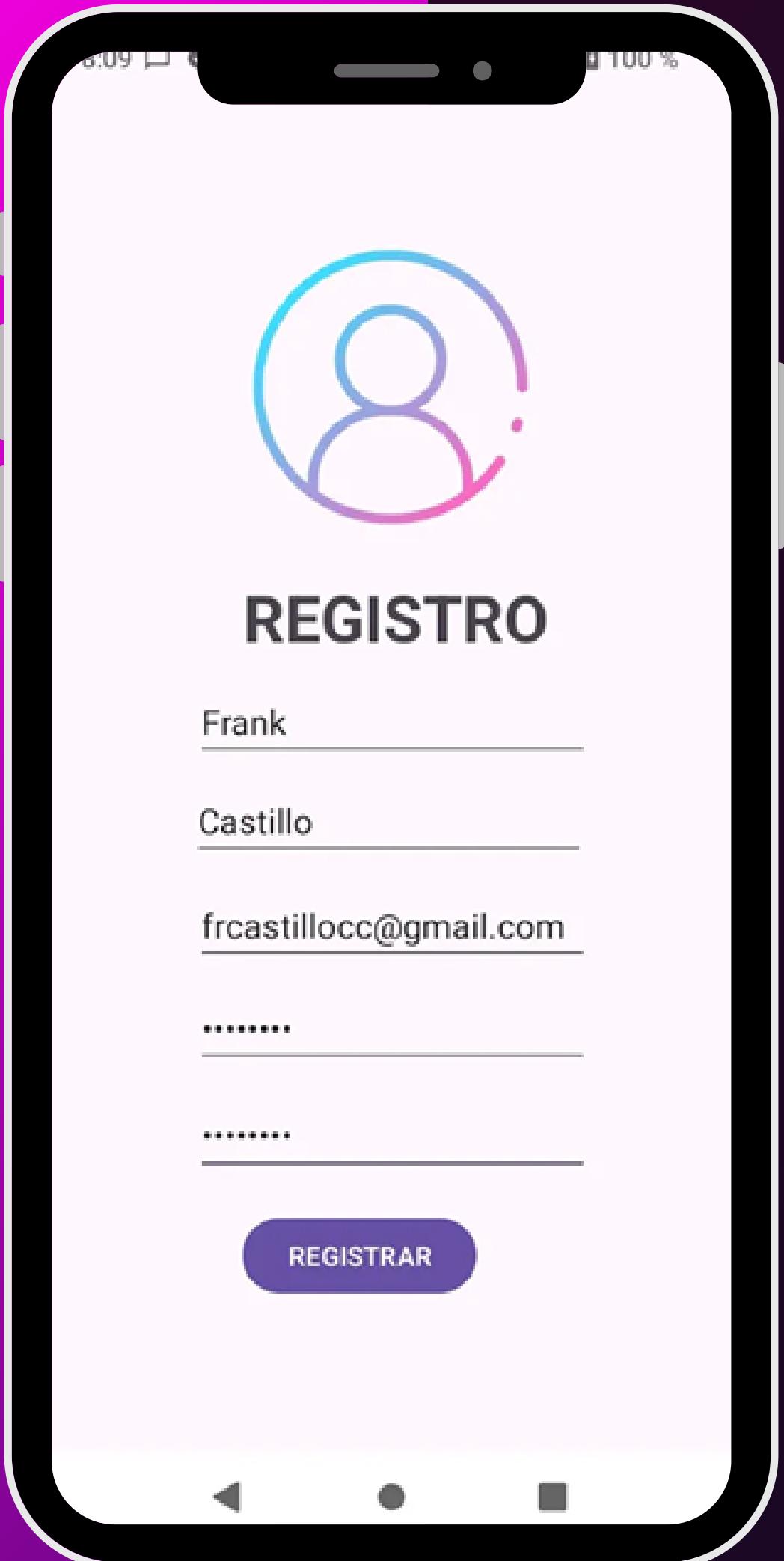
REGISTRAR

Complete todos los campos



● Conectar con BD

The screenshot shows the Firebase Database console for a project named "Safe Route App". The left sidebar includes links for "Descripción gen...", "Accessos directos a proyectos", "Authentication", "Firestore Database" (which is selected and highlighted in blue), "Novedades", "AI Logic (NUEVO)", "Categorías de producto", and "Herramientas de desarrollo relacionadas" (with a link to "Firebase Studio"). Below these are sections for "Compilación", "Ejecución", and "Analytics". At the bottom of the sidebar, there's a "Spark" section with the text "Sin costo (USD 0 al mes)" and a "Actualizar NUEVA" button. The main content area is titled "Base de datos" and features tabs for "Datos", "Reglas", "Índices", "Recuperación ante desastres", "Uso", and "Extensions". A prominent button labeled "Agregar base de datos" is visible. A callout bubble from "Gemini" suggests asking how to start using Firestore. Below this, a shield icon indicates protection against abuse, with a link to "Configurar la Verificación de aplicaciones". The interface also includes "Vista del panel", "Compilador de consultas", and a "Más funciones en Google Cloud" dropdown. The central workspace shows a collection named "(default)" with a sub-item "+ Iniciar colección". A circular icon with a grid pattern is displayed, and a message at the bottom states "Tu base de datos está lista. Solo tienes que agregar datos." The right side of the screen has a vertical toolbar with icons for user profile, notifications, and other tools.



Home

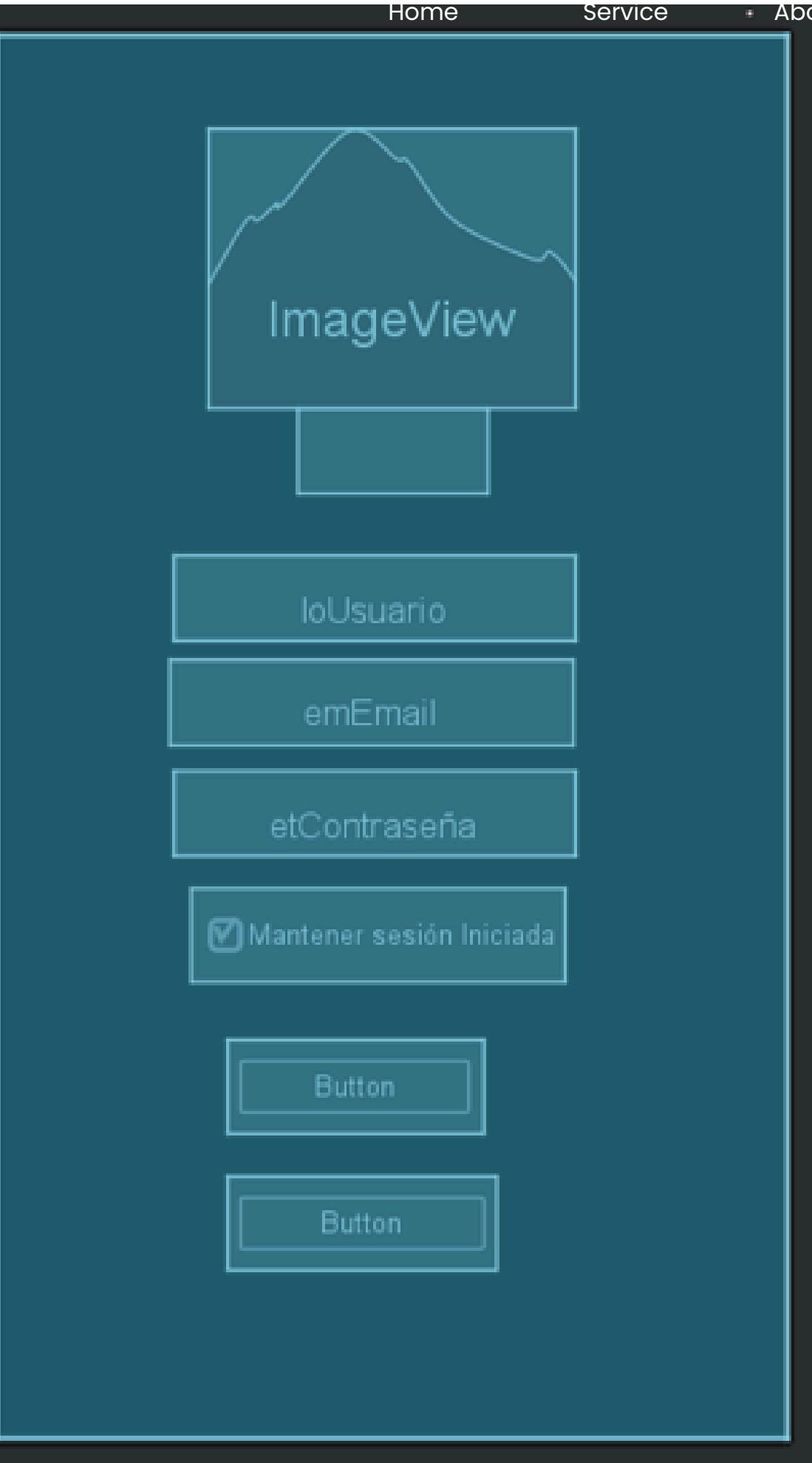
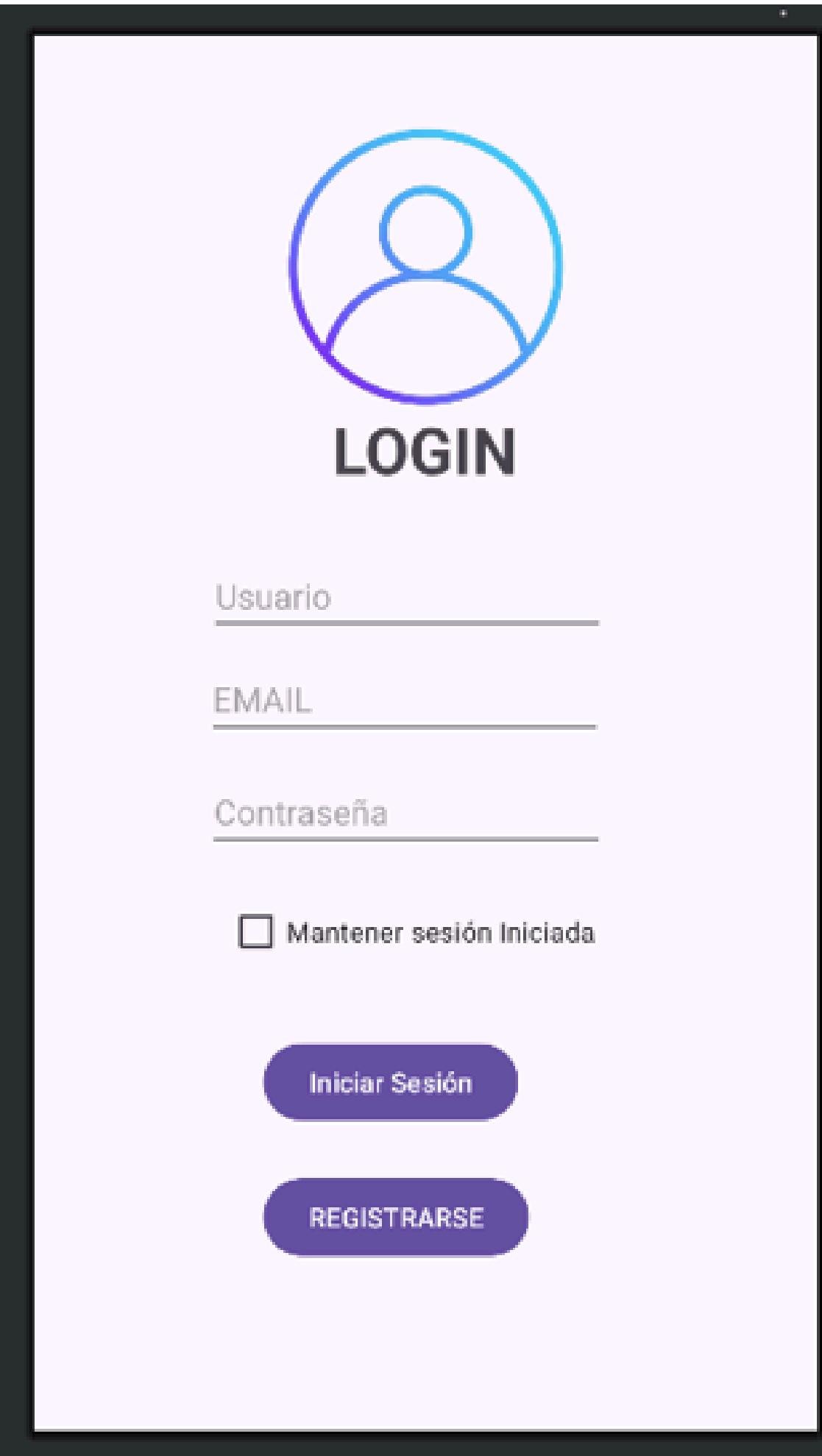
Service

About Us

Contact

● Cifrar contraseña

- Diseñar pantalla de login

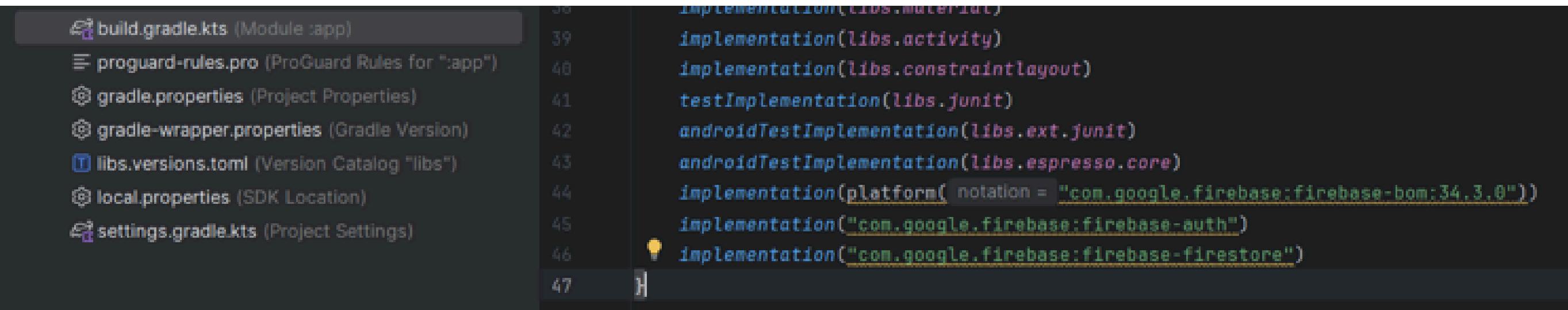
[Home](#)[Service](#)[Abc](#)[Contact](#)

● Implementar autenticación

→ `implementation("com.google.firebaseio:firebase-auth")`

Biblioteca para usar Firebase Authentication.

Permite el registro e inicio de sesión de usuarios con correo y contraseña,
Google, Facebook u otros proveedores de autenticación.



```
build.gradle.kts (Module :app)
proguard-rules.pro (ProGuard Rules for ":app")
gradle.properties (Project Properties)
gradle-wrapper.properties (Gradle Version)
libs.versions.toml (Version Catalog "libs")
local.properties (SDK Location)
settings.gradle.kts (Project Settings)

implementation('ios.material')
implementation(libs.activity)
implementation(libs.constraintlayout)
testImplementation(libs.junit)
androidTestImplementation(libs.ext.junit)
androidTestImplementation(libs.espresso.core)
implementation(platform( notation = "com.google.firebaseio:bom:34.3.0"))
implementation("com.google.firebaseio:firebase-auth")
implementation("com.google.firebaseio:firebase-firebase-store")
```

Obtiene la instancia de Firebase Authentication.

Con esto se puede:

Registrar usuarios con email y contraseña, google, facebook.

4 usages

```
private FirebaseAuth mAuth;
```

2 usages

● Notificación de errores

[About Us](#)[Contact](#)



LOGIN

Usuario

EMAIL

Contraseña

Mantener sesión Iniciada

Iniciar Sesión

REGISTRARSE

 Ingrese email y contraseña



REGISTRO

NOMBRE

APELLIDO

EMAIL

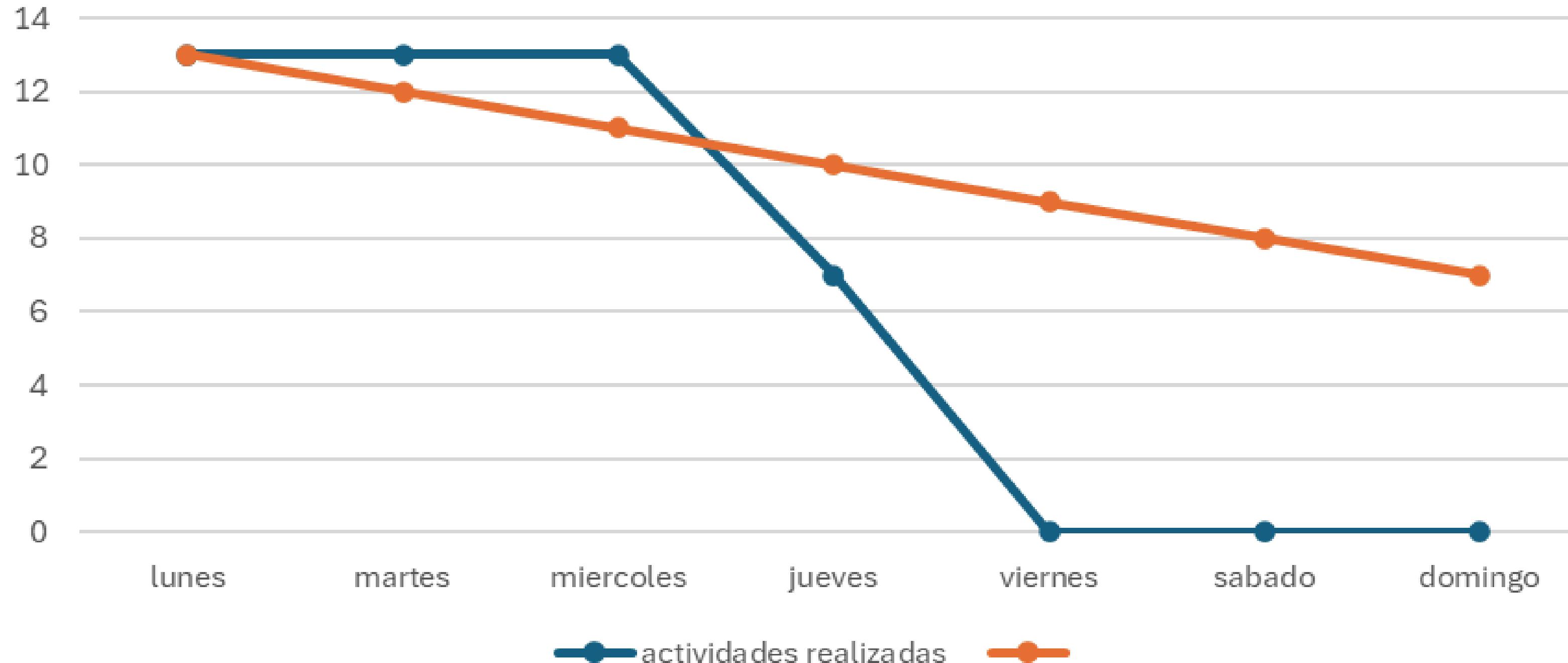
CONTRASEÑA

CONFIRMAR CONTRASEÑA

REGISTRAR

 Complete todos los campos

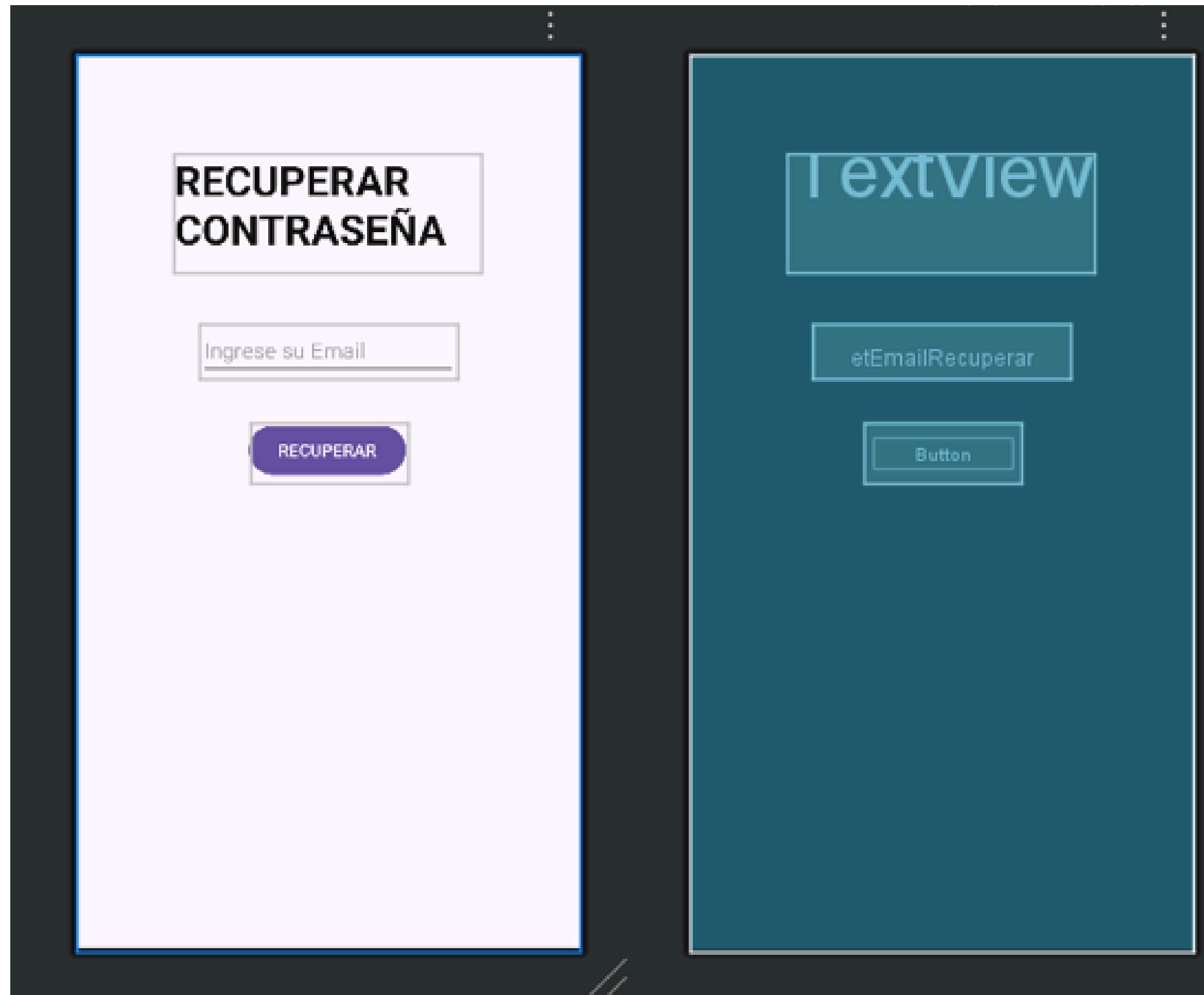
Sprint 1

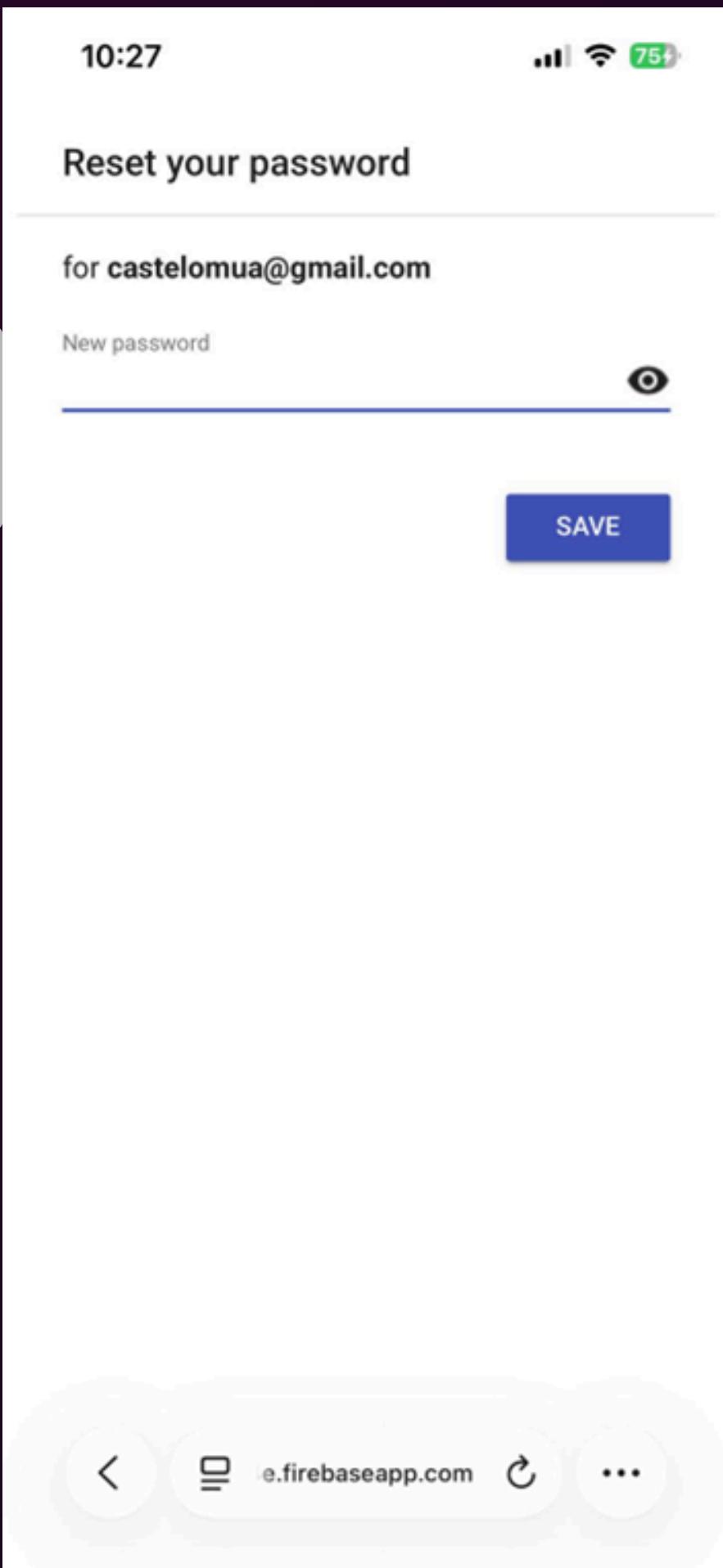
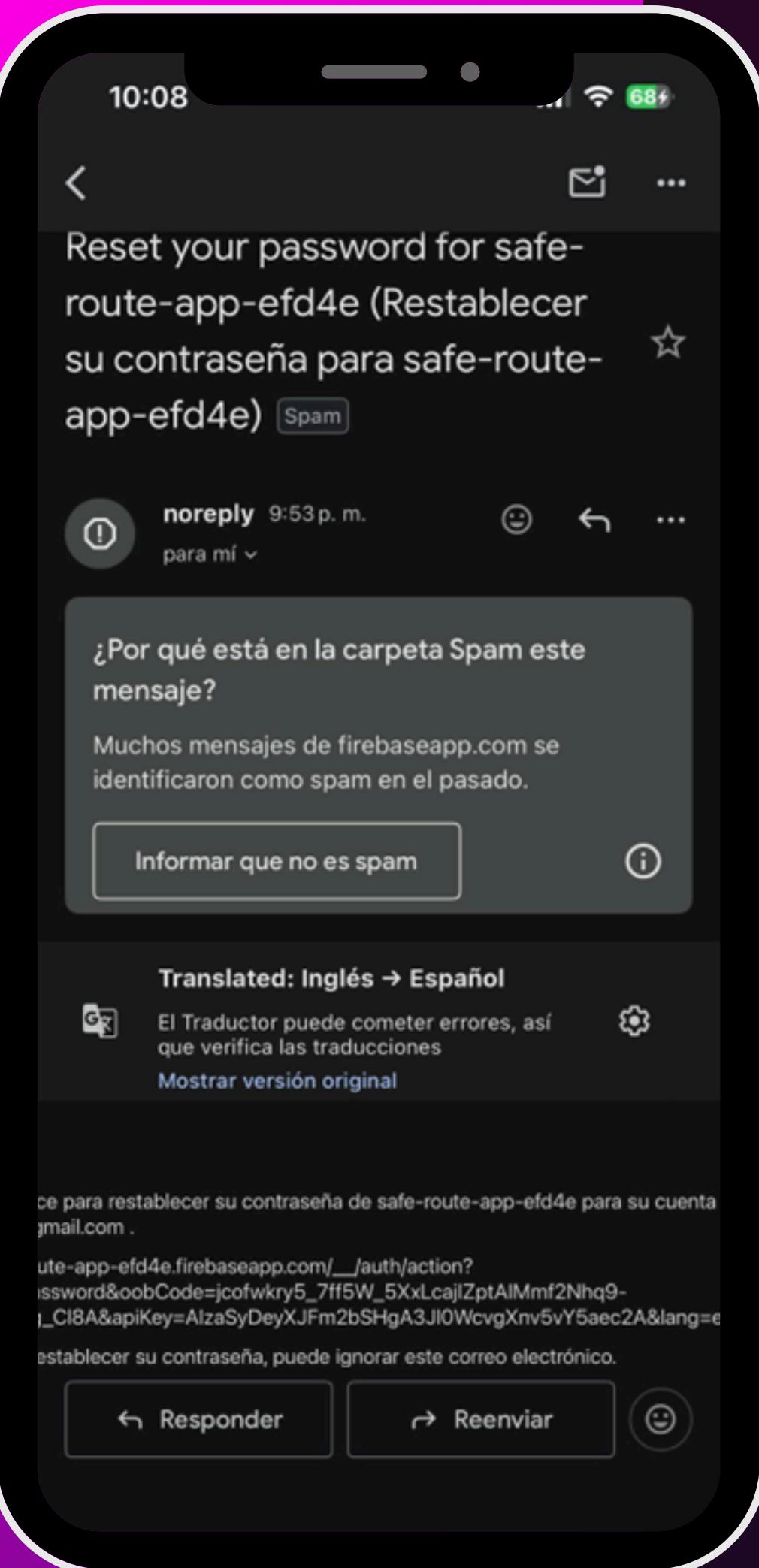


SPRINT Nº2



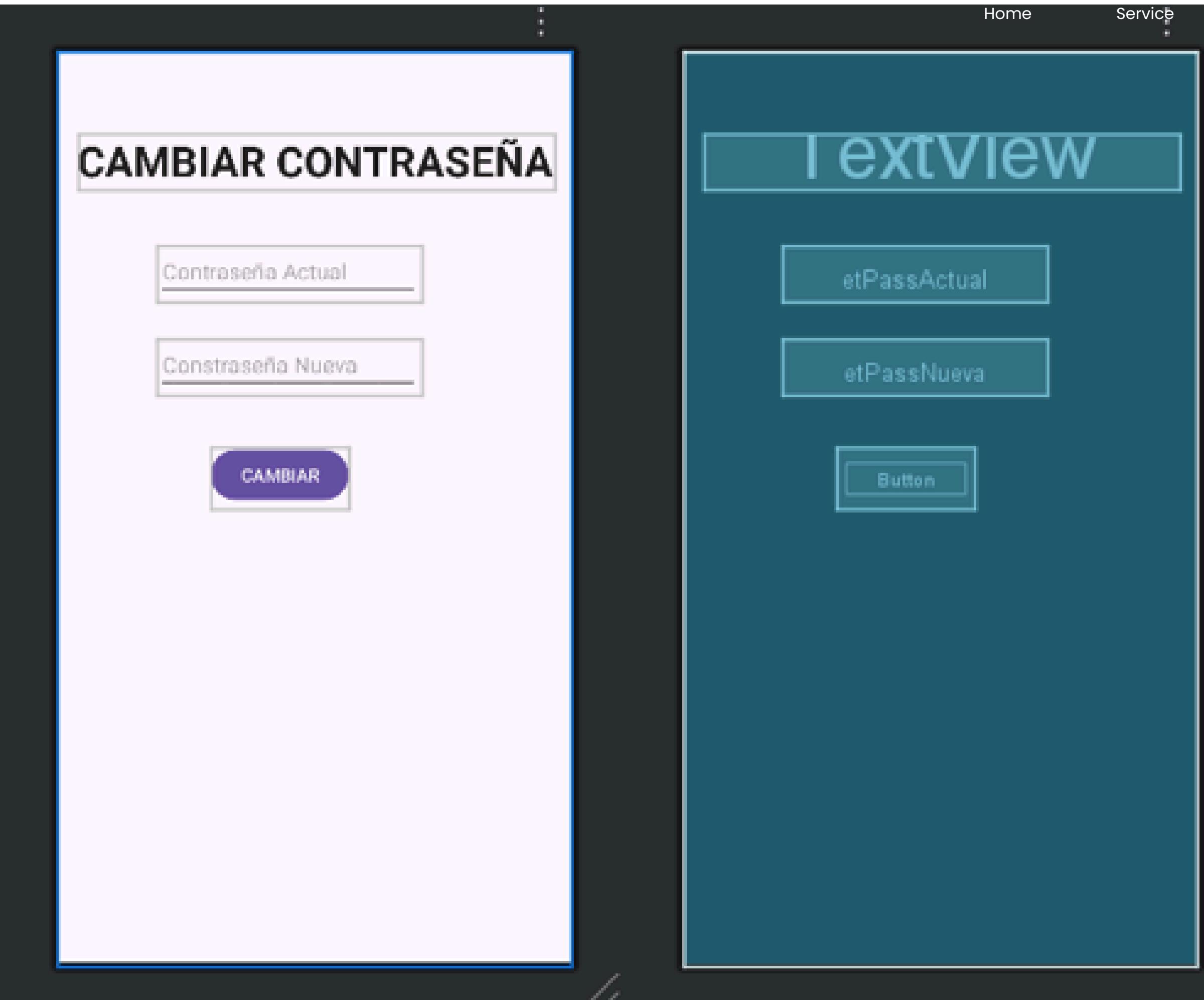
- Diseñar pantalla de recuperación





Enviar enlace/código

- Crear pantalla de cambio de contraseña



● Configurar permisos

Estado de prueba gratuita: crédito por S/.1,065.20 y 91 días restantes. Activa tu cuenta completa para obtener acceso ilimitado a todas las funciones de Google Cloud. Usa los créditos restantes y paga solo por lo que usas.

Descartar Activar

Google Cloud My First Project Buscar (I) recursos, documentos, productos y más Buscar

APIs y servicios Credenciales + Crear credenciales ▾ Eliminar Restablecer credenciales borradas

APIs y servicios habilitados

Biblioteca

Credenciales

Pantalla de consentimiento...

Acuerdos de uso de págin...

Crea credenciales para acceder a tus API habilitadas. [Más información](#)

⚠ Recuerda configurar la pantalla de consentimiento de OAuth con información sobre tu app. [Configurar pantalla de consentimiento](#)

Claves de API

<input type="checkbox"/>	Nombre	Fecha de creación	Restricciones	Acciones
<input type="checkbox"/>	Maps Platform API Key	25 sept 2025	Apps para Android, 3 API	Mostrar clave ...

IDs de clientes de OAuth 2.0

<input type="checkbox"/>	Nombre	Fecha de creación	Tipo	ID de cliente	Acciones
No hay clientes de OAuth para mostrar					

Cuentas de servicio

<input type="checkbox"/>	Correo electrónico	Nombre	Acciones
No hay cuentas de servicio para mostrar			

Estás viendo el proyecto "My First Project" en la organización "continental.edu.pe"

Cerrar mensaje [Administrador de cuentas de servicio](#)



Integrar API de geolocalización

Google Cloud SafeRouteProject

RPI APIs y servicios / Biblioteca de API

Filtro Escribe para filtrar

Visibilidad

- Pública (494)
- Privada (2)

Categoría

- Analítica (11)
- Macrodatos (22)
- Bases de datos (6)
- Aprendizaje automático (16)
- Mapas (32)
- DevOps (24)
- Procesamiento (13)
- Publicidad (15)
- Sociales (5)

Mapas

Ver todo (32)

- Maps SDK for Android Google Maps for your native Android app.
- Maps SDK for iOS Google Maps for your native iOS app.
- Maps JavaScript API Google Maps for your website
- Places API Google Enterprise API Get detailed information about 100 million places

Aprendizaje automático

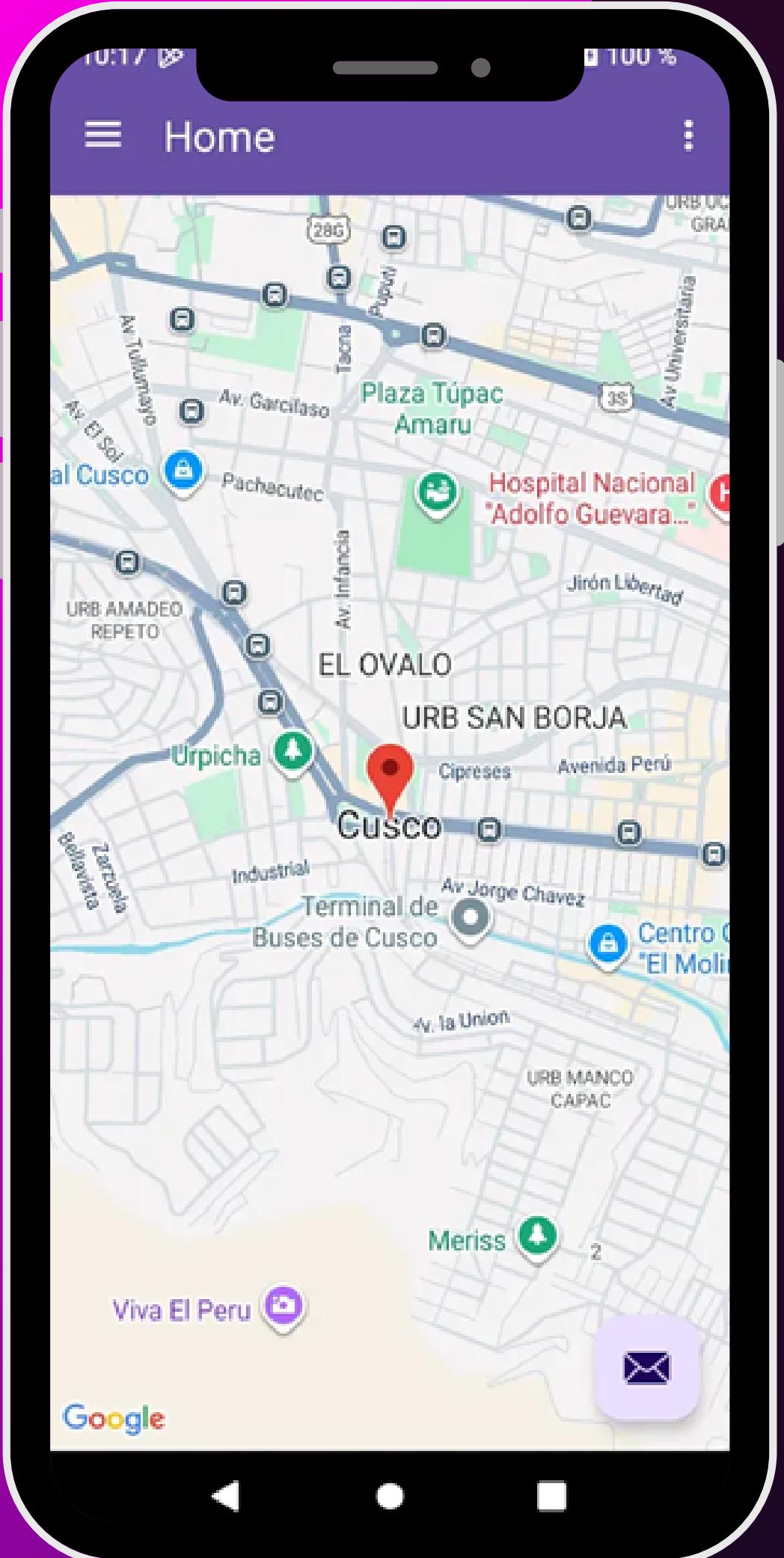
Ver todo (16)

- Dialogflow API Google Enterprise API Builds conversational interfaces
- Cloud Vision API Google Enterprise API Image Content Analysis
- Cloud Natural Language API Google Enterprise API Provides natural language understanding technologies, such as sentiment analysis, entity...
- Cloud Speech-to-Text API Google Enterprise API Speech recognition

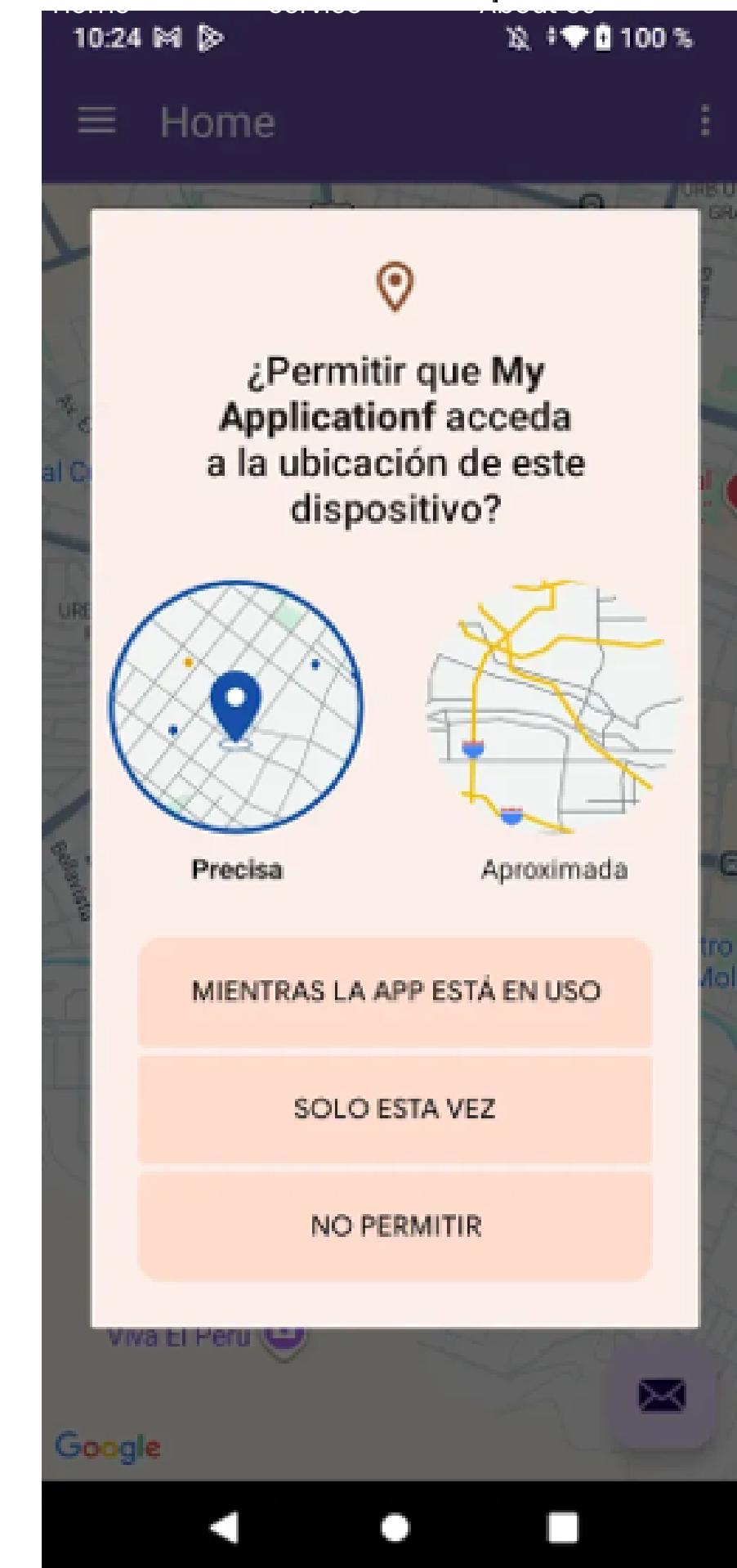
Estás viendo el proyecto "SafeRouteProject" en la organización "continental.edu.pe"

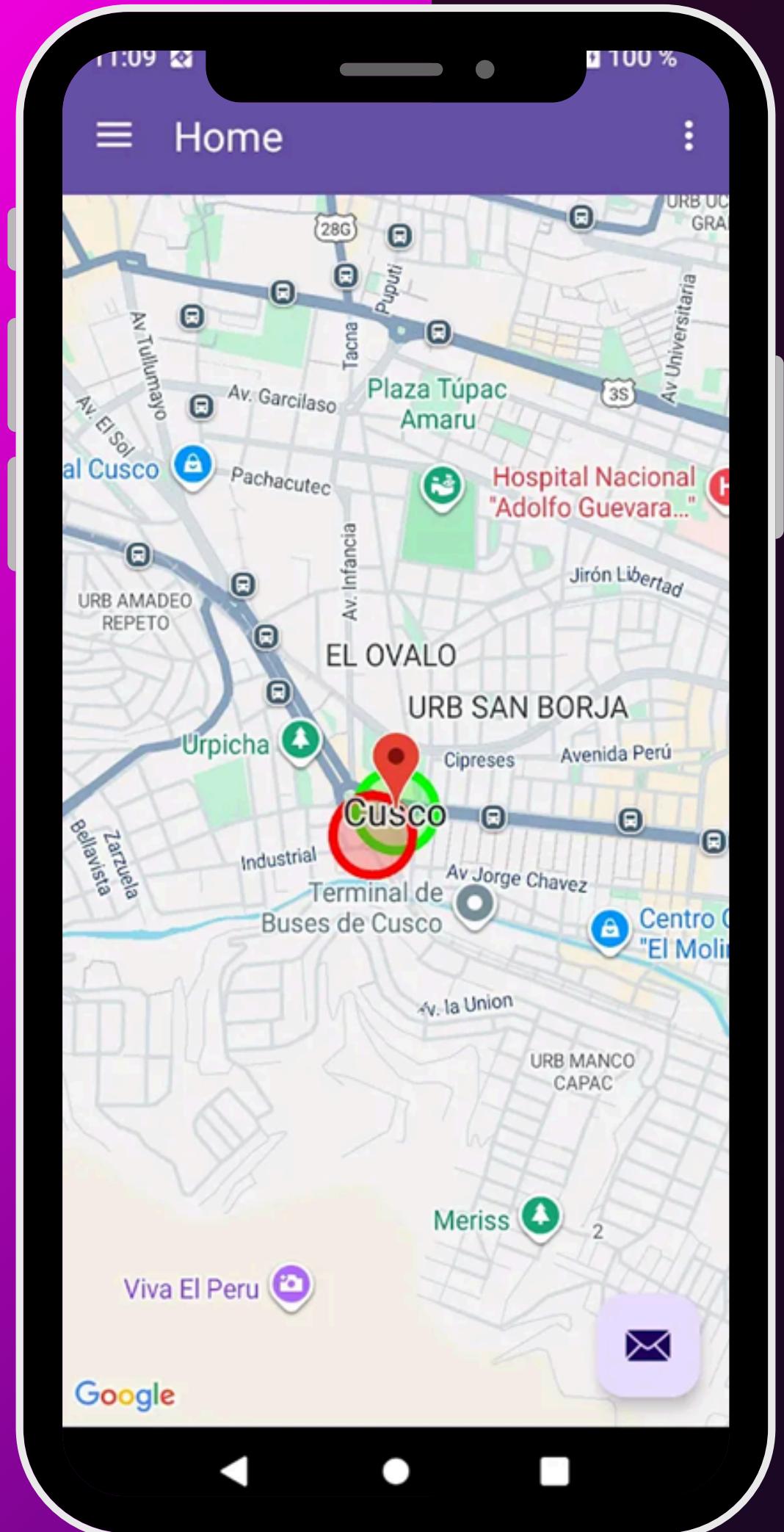
<https://console.cloud.google.com/apis/library/vision.googleapis.com?project=saferrouteproject-472300>

● Obtener coordenadas en tiempo real

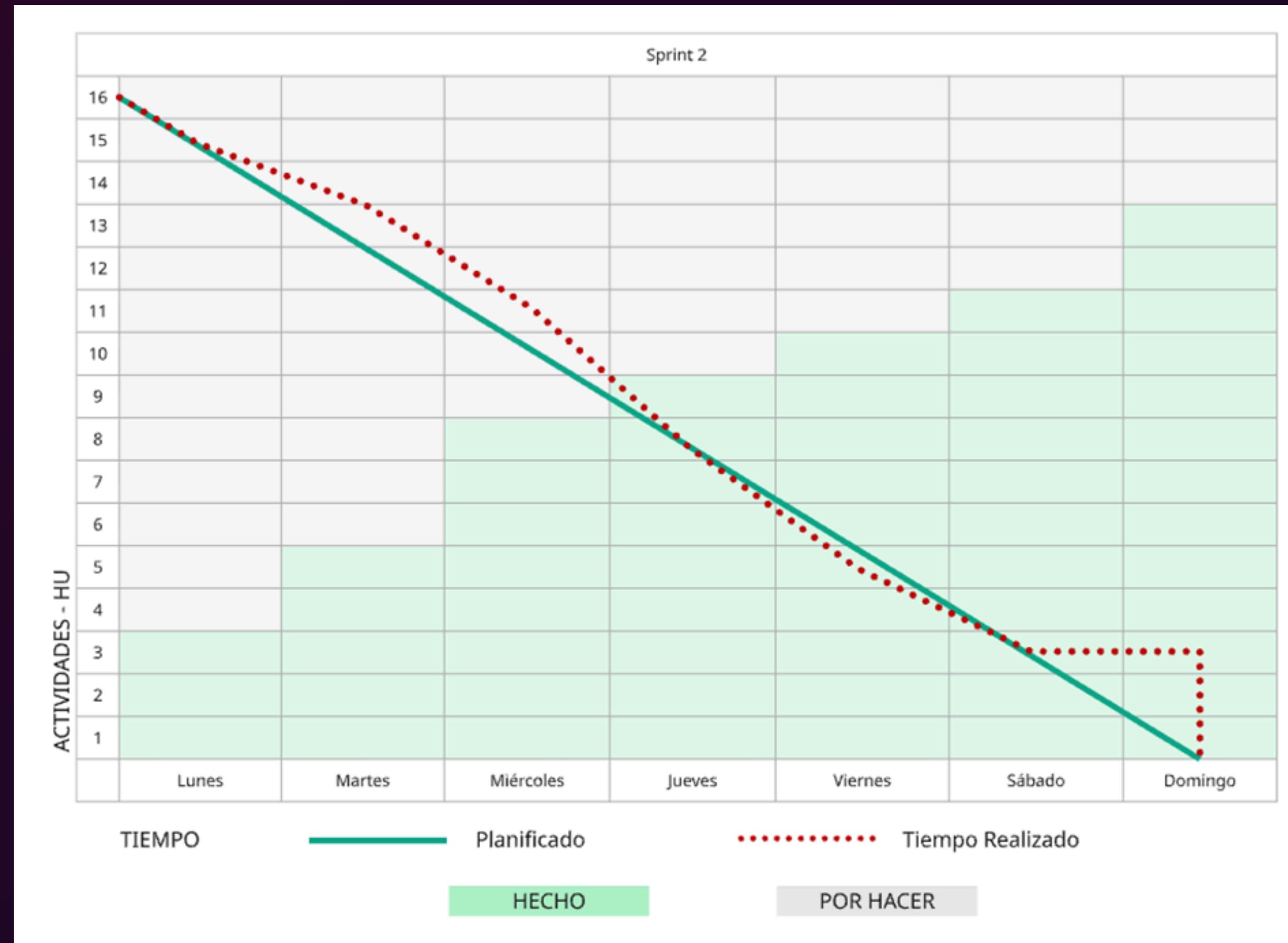


- Actualizar en tiempo real





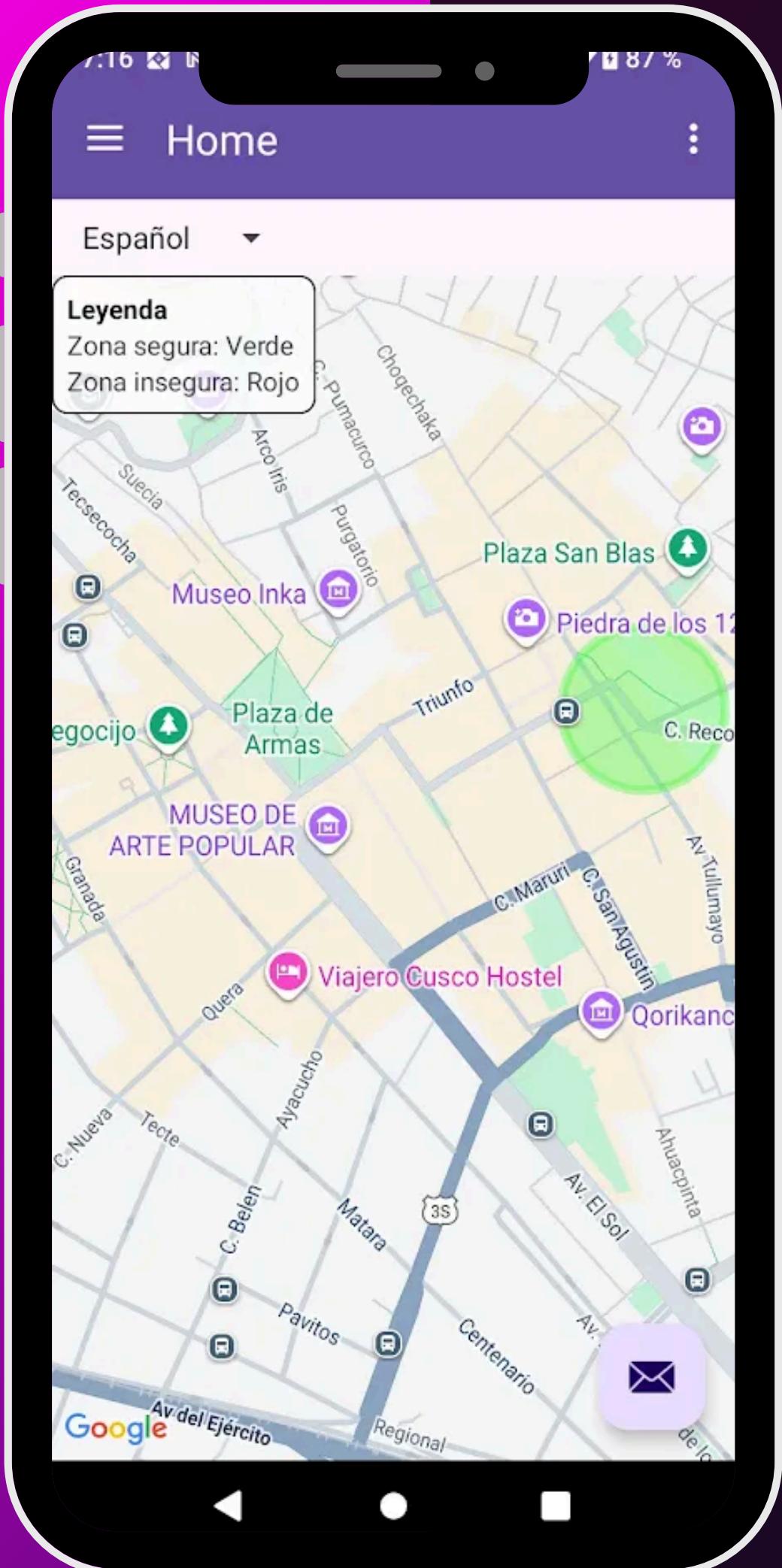
**Diseñar capas de mapa
seguras y Colores de
seguridad alta**



Encargado	N°	Actividades	Nombre	Por hacer	Haciendo	Terminado
Jhon	1	HU-04	Diseñar pantalla de recuperación			Diseñar pantalla de recuperación
Frank	2	HU-04	Validar correo			Validar correo
Luis	3	HU-04	Enviar enlace/código			Enviar enlace/código
Jhon	4	HU-04	Crear pantalla de cambio de contraseña			Crear pantalla de cambio de contraseña
Frank	5	HU-04	Pruebas de proceso completo			Pruebas de proceso completo
Luis	6	HU-05	Configurar permisos			Configurar permisos
Jhon	7	HU-05	Integrar API de geolocalización			Integrar API de geolocalización
Frank	8	HU-05	Obtener coordenadas en tiempo real			Obtener coordenadas en tiempo real
Luis	9	HU-05	Pruebas en dispositivos			Pruebas en dispositivos
Jhon	10	HU-06	Integrar mapa interactivo			Integrar mapa interactivo
Frank	11	HU-06	Mostrar ubicación con marcador			Mostrar ubicación con marcador
Luis	12	HU-06	Actualizar en tiempo real			Actualizar en tiempo real
Jhon	13	HU-07	Configurar frecuencia de actualización			Configurar frecuencia de actualización
Frank	14	HU-07	Notificación al cambiar zona		Notificación al cambiar zona	
Luis	15	HU-08	Diseñar capas de mapa seguras		Diseñar capas de mapa seguras	
Jhon	16	HU-08	Colores de seguridad alta		Colores de seguridad alta	

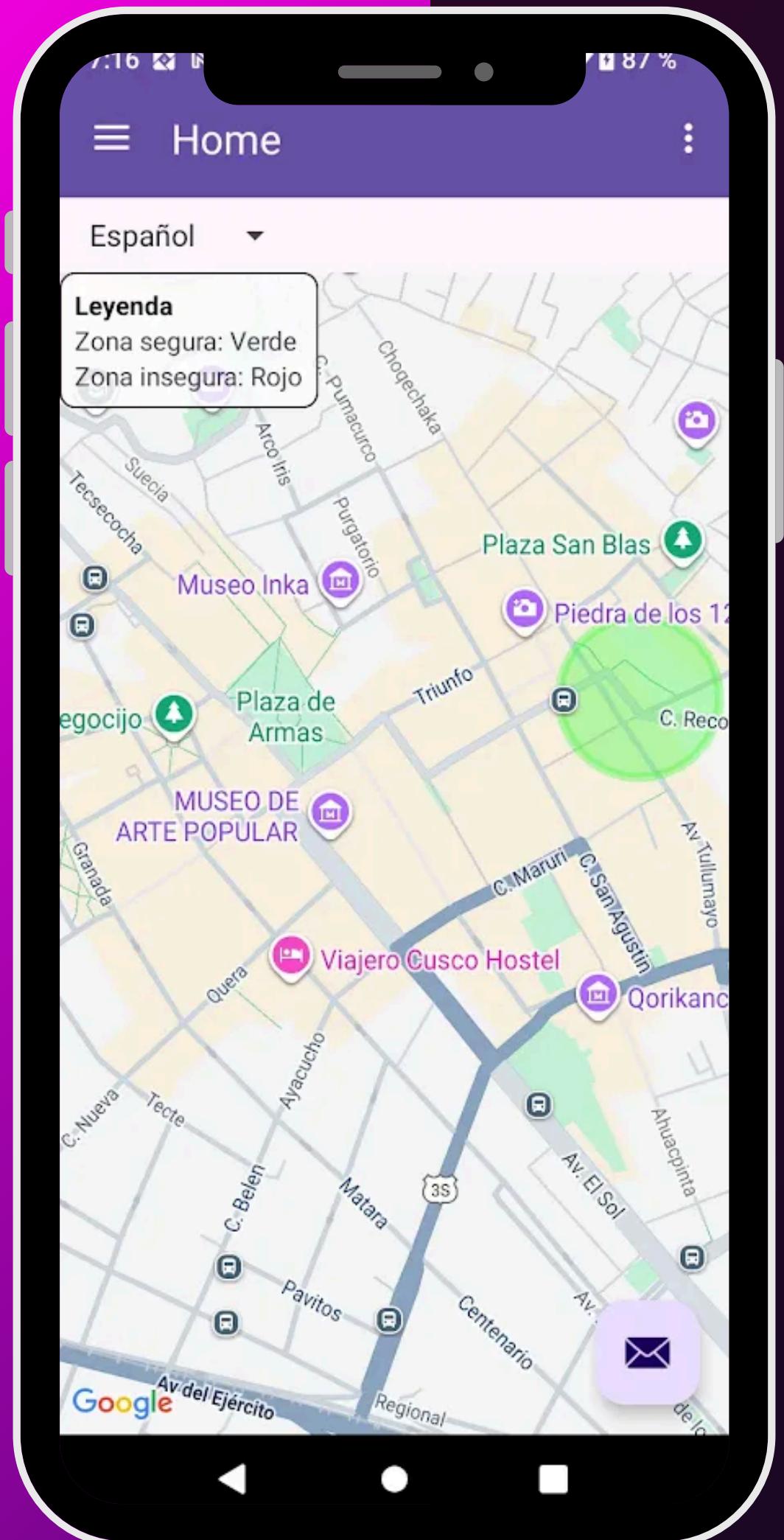
SPRINT Nº3



[Home](#)[Service](#)[About Us](#)[Contact](#)

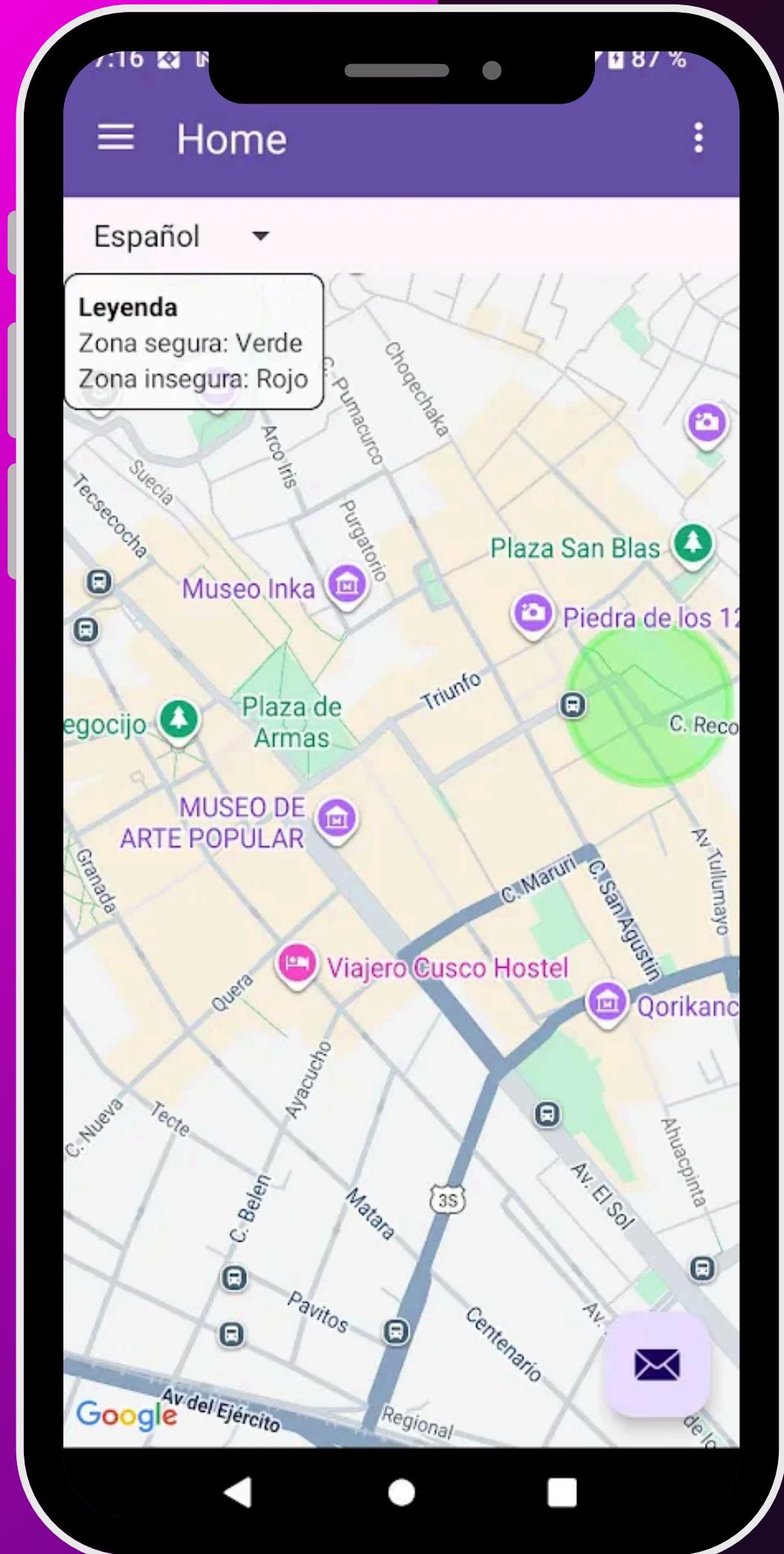
IMPLEMENTAR LEYENDA

```
@Override  
public void onMapReady(GoogleMap googleMap) {  
    FrameLayout mapContainer = getView().findViewById(R.id.mapContainer);  
    View legendView = LayoutInflater.from(getContext()).inflate(R.layout.legend_layout, mapContainer, false);  
    mapContainer.addView(legendView);  
  
    mMap = googleMap;  
  
    LatLng cusco = new LatLng(-13.53195, -71.967463);  
    mMap.addMarker(new MarkerOptions().position(cusco).title("Cusco Default"));  
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(cusco, 15));  
  
    dibujarZonasDesdeFirestore();  
}
```



Configurar capas inseguras

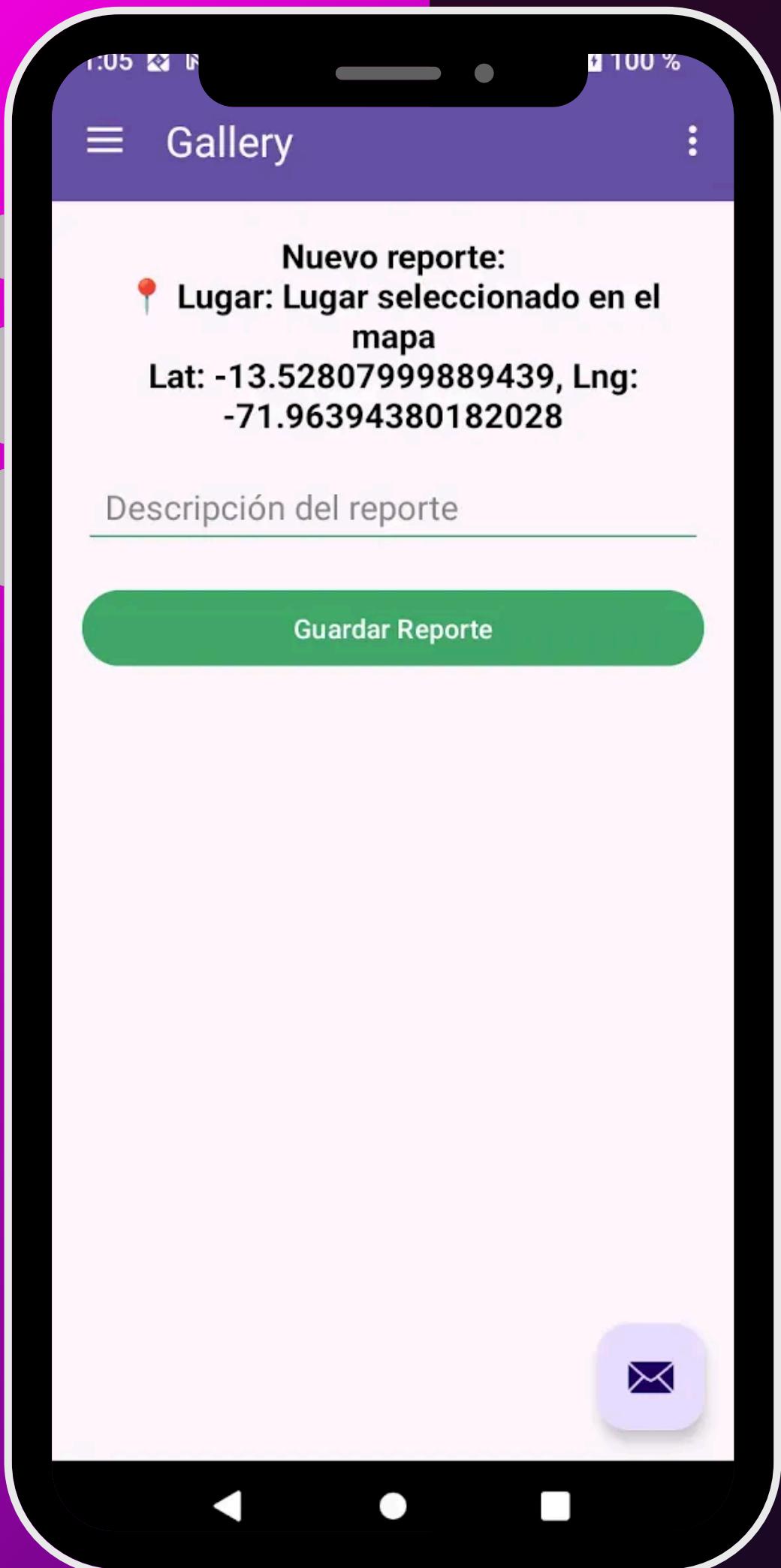
```
mMap.setOnMapClickListener(latLng -> {
    Reporte reporte = new Reporte(latLng.latitude, latLng.longitude, "Reportado por usuario");
    db.collection("reportes").add(reporte);
    mMap.addMarker(new MarkerOptions().position(latLng).title("Reporte enviado"));
    NotificacionHelper.mostrar(requireContext(),
        "Has reportado una zona: " + latLng.latitude + ", " + latLng.longitude);
});
```



Colorear mapa por riesgo

Se dibujan círculos sobre el mapa con colores distintos según el nivel de riesgo:

```
private void dibujarZonasPorRiesgo() {  
    if(mMap == null) return;  
  
    LatLng[] zonas = {zonaSegura, zonaPeligrosa};  
    int[] riesgos = {1, 3}; // 1 = bajo, 3 = alto  
  
    for (int i = 0; i < zonas.length; i++) {  
        int color;  
        switch (riesgos[i]) {  
            case 1: color = Color.parseColor("#5533FF33"); break; // Verde  
            case 2: color = Color.parseColor("#55FFAA50"); break; // Naranja  
            default: color = Color.parseColor("#55FF3333"); break; // Rojo  
        }  
  
        mMap.addCircle(new CircleOptions()  
            .center(zonas[i])  
            .radius(100)  
            .strokeColor(color)  
            .fillColor(color));  
    }  
}
```



Reportes Interfaz

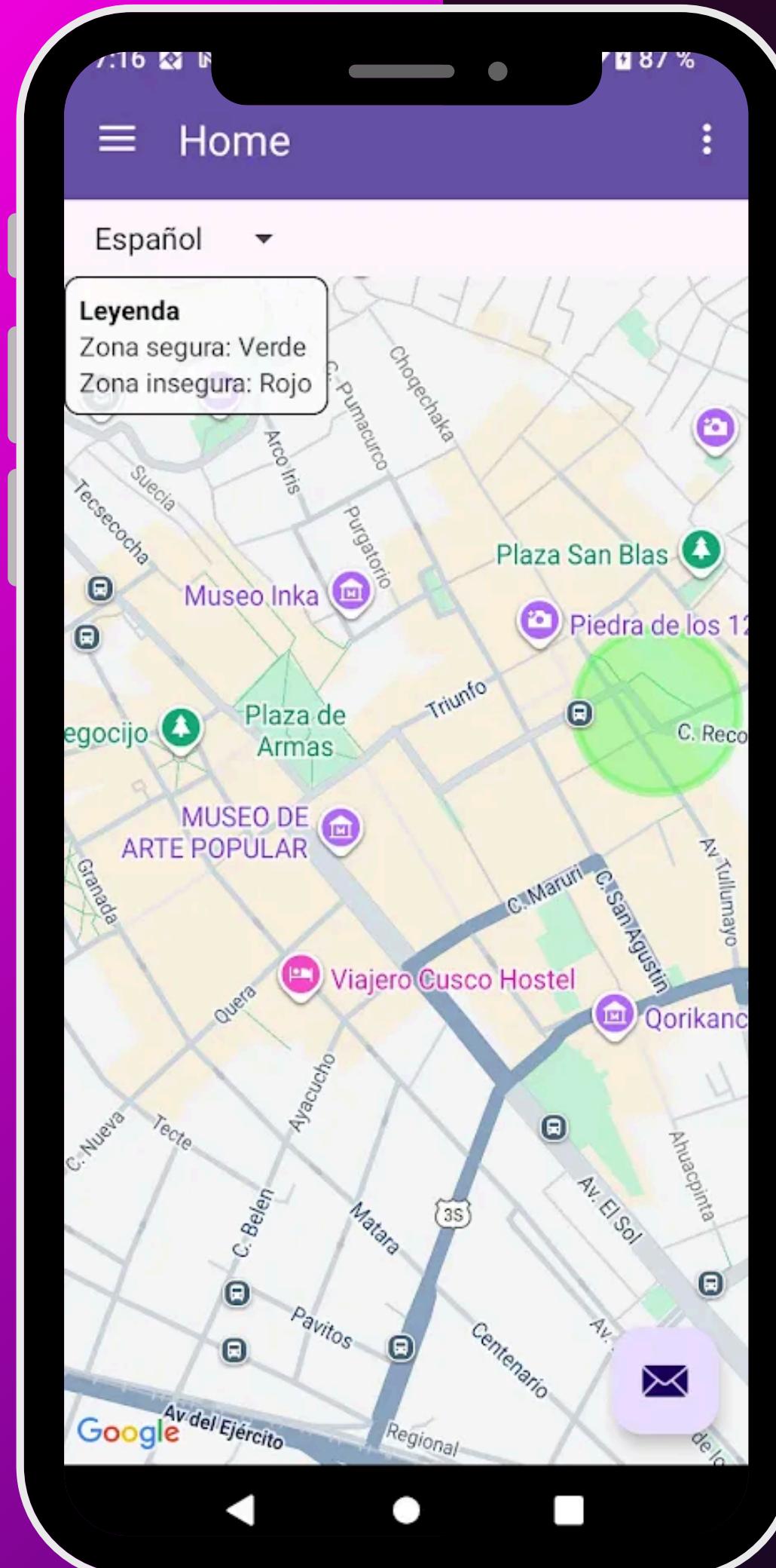
Android Studio code editor showing the `Reporte.java` file. The code defines a class `Reporte` with fields for latitude (`lat`), longitude (`lng`), a description (`descripcion`), and a place name (`nombreLugar`). It also includes a constructor for Firestore and a placeholder constructor.

```
package com.example.myapplication.Models;
public class Reporte {
    private double lat;
    private double lng;
    private String descripcion;
    private String nombreLugar; // nuevo campo para el lugar
    // Constructor vacio necesario para Firestore
    public Reporte() {}
    // Constructor completo
}
```

Android Studio code editor showing a portion of the `binding.setTextGallery` method. It receives arguments from the `HomeFragment` and sets text to a UI element. The code highlights several variables and methods in red, indicating they are not yet defined or are part of a different context.

```
// Recibir lat, lng y nombreLugar desde el HomeFragment
Bundle args = getArguments();
if (args != null) {
    lat = args.getDouble("lat", 0);
    lng = args.getDouble("lng", 0);
    nombreLugar = args.getString("nombreLugar", "Sin nombre");

    binding.textGallery.setText("Nuevo reporte:\n" +
        " Lugar: " + nombreLugar +
        "\nLat: " + lat +
        ", Lng: " + lng);
} else {
    binding.textGallery.setText("Lista de reportes:");
    cargarReportes(); // si no viene de un click en el mapa,
}
```



Implementar selector de idioma

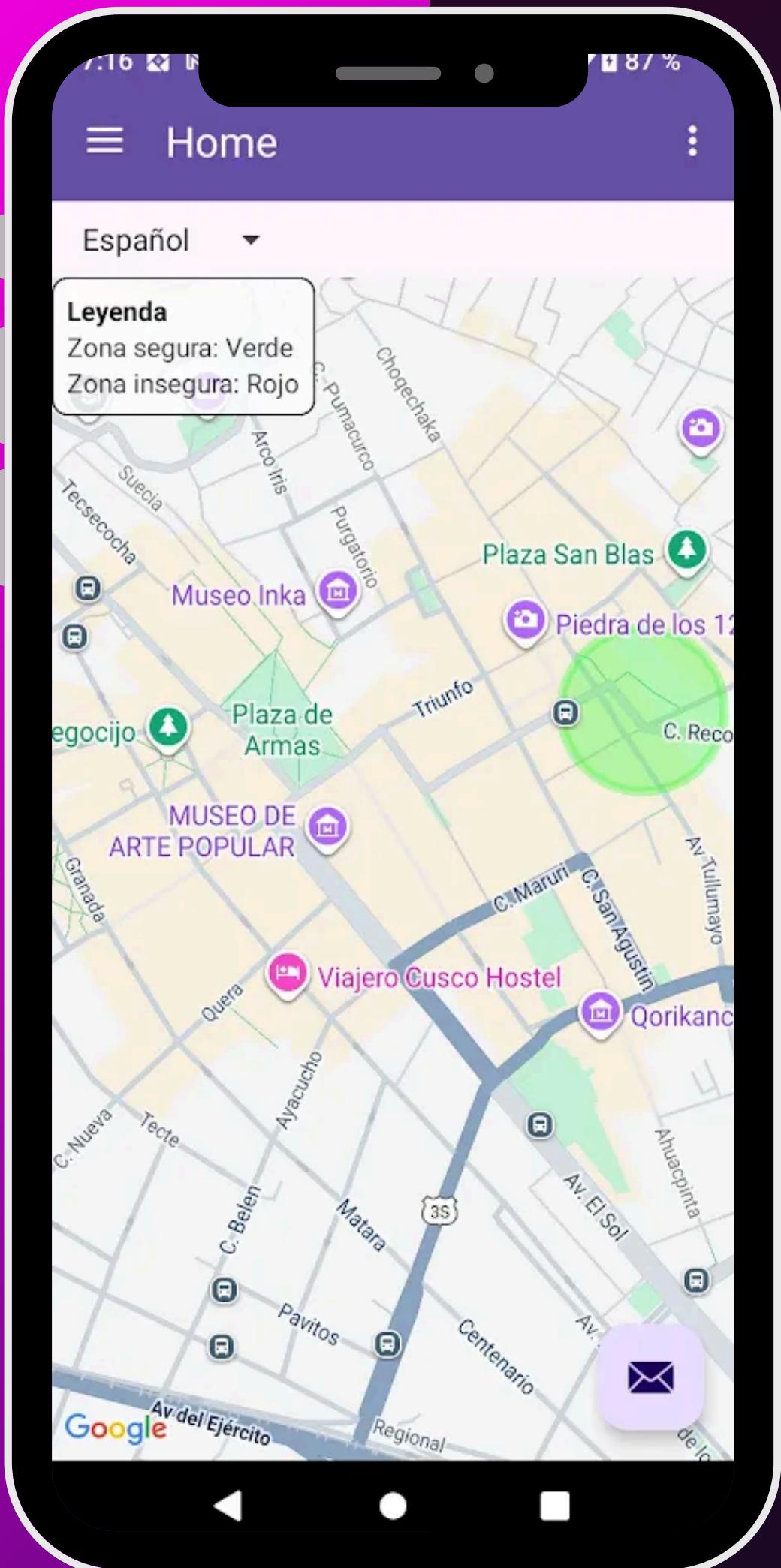
The screenshot shows the Google Cloud Platform interface for the Cloud Translation API. At the top, it says "Google Cloud" and "My First Project". Below that, it says "Detalles del producto" and "Cloud Translation API" with a "Google Enterprise API" link. It describes the API as "Integrates text translation into your website or application." There are three buttons: "Administrar", "Probar esta API" (which is highlighted), and "API habilitada" (with a checked checkbox). Below these buttons are tabs for "Descripción general", "Precios", "Documentación", and "Productos relacionados".

The screenshot shows an Android code editor with Java code. The code implements the Google Cloud Translation API and sets up a spinner for selecting languages. The code includes imports for Context, Spinner, ArrayAdapter, and R. It defines a string array of languages ("Espanol", "Ingles") and creates an ArrayAdapter to set it as the adapter for a spinner.

```
implementation("com.google.cloud:google-cloud-translate:2.24.0")

// Spinner para seleccionar idioma
Spinner spinner = root.findViewById(R.id.spinnerIdiomas);
String[] idiomas = {"Espanol", "Ingles"};
ArrayAdapter<String> adapter = new
ArrayAdapter<>(requireContext(),
android.R.layout.simple_spinner_item, idiomas);

adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spinner.setAdapter(adapter);
```

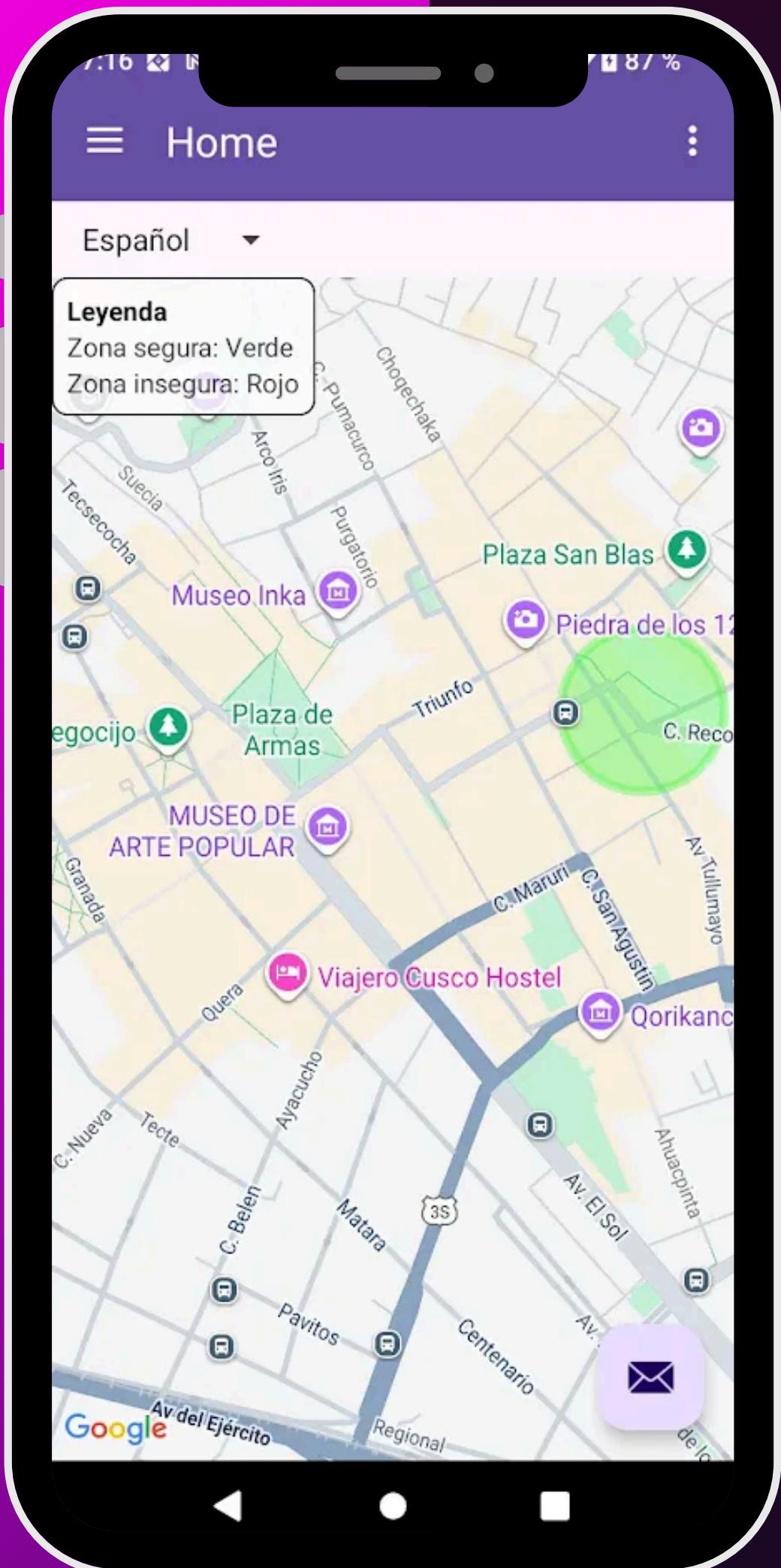


Mostrar lugares turísticos

```
// * Mostrar lugares turísticos cercanos
private void mostrarLugaresTurísticos(LatLng location) {
    String url =
"https://maps.googleapis.com/maps/api/place/nearbysearch/json?" +
        "location=" + location.latitude + "," + location.longitude +
        "&radius=2000&type=tourist_attraction&key=" + apiKey;

    RequestQueue queue = Volley.newRequestQueue(requireContext());
    JsonObjectRequest request = new
JsonObjectRequest(Request.Method.GET, url, null,
        response -> {
        try {
            JSONArray results =
response.getJSONArray("results");
            for (int i = 0; i < results.length(); i++) {
                JSONObject place = results.getJSONObject(i);
                String name = place.getString("name");
                JSONObject geometry =
place.getJSONObject("geometry").getJSONObject("location");
                double lat = geometry.getDouble("lat");
                double lng = geometry.getDouble("lng");

                LatLng pos = new LatLng(lat, lng);
                mMap.addMarker(new
MarkerOptions().position(pos).title(name));
            }
        } catch (Exception e) {
```



Filtrar info relevante

```
// Spinner Filtro

String[] filtros = {"Ninguno", "Zonas de riesgo", "Lugares turísticos",
"Ambos"};

ArrayAdapter<String> adapterFiltro = new ArrayAdapter<>(requireContext(),
    android.R.layout.simple_spinner_item, filtros);

adapterFiltro.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

spinnerFiltro.setAdapter(adapterFiltro);

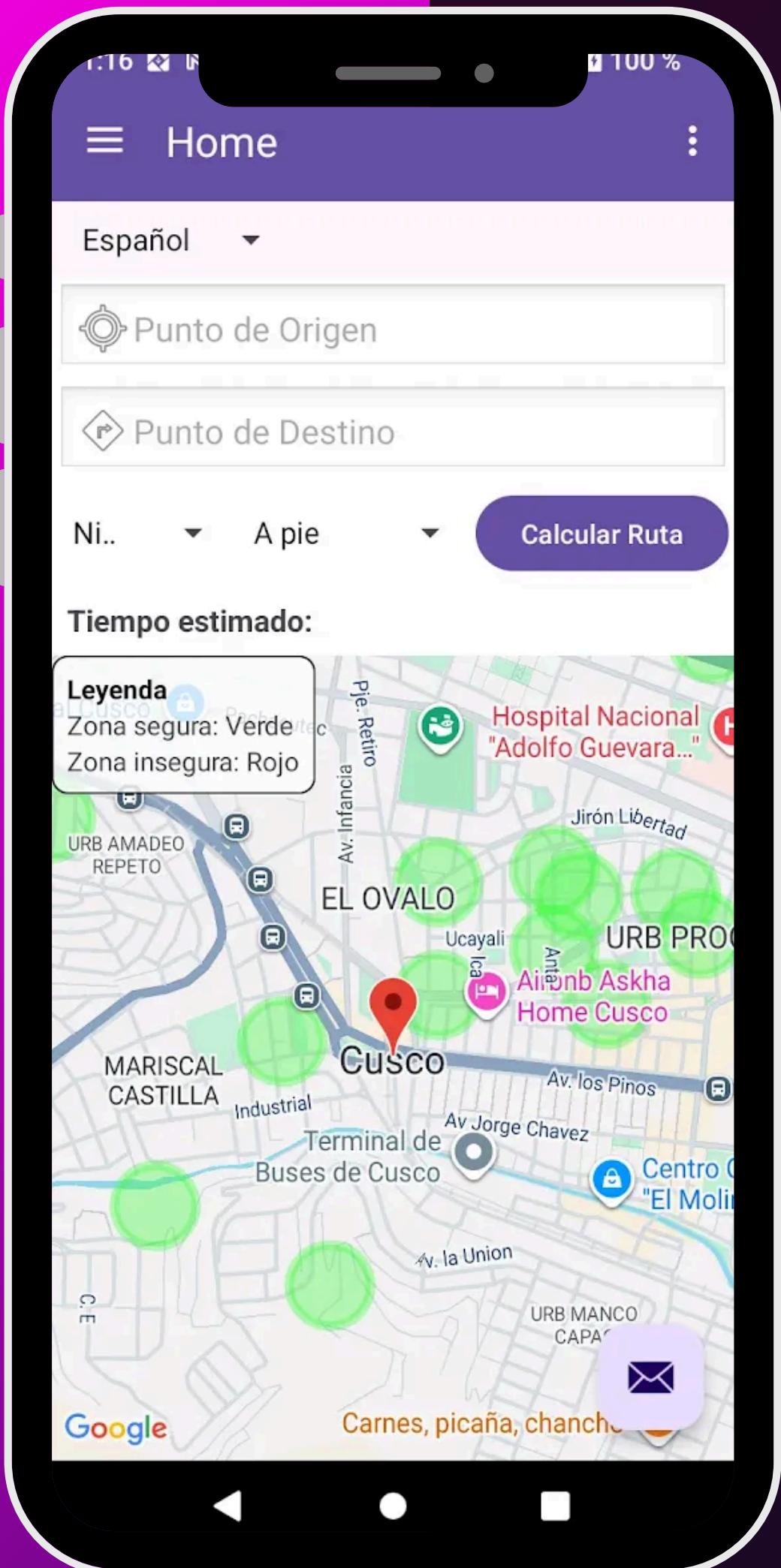
spinnerFiltro.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {

    @Override

    public void onItemSelected(AdapterView<?> parent, View view, int
position, long id) {

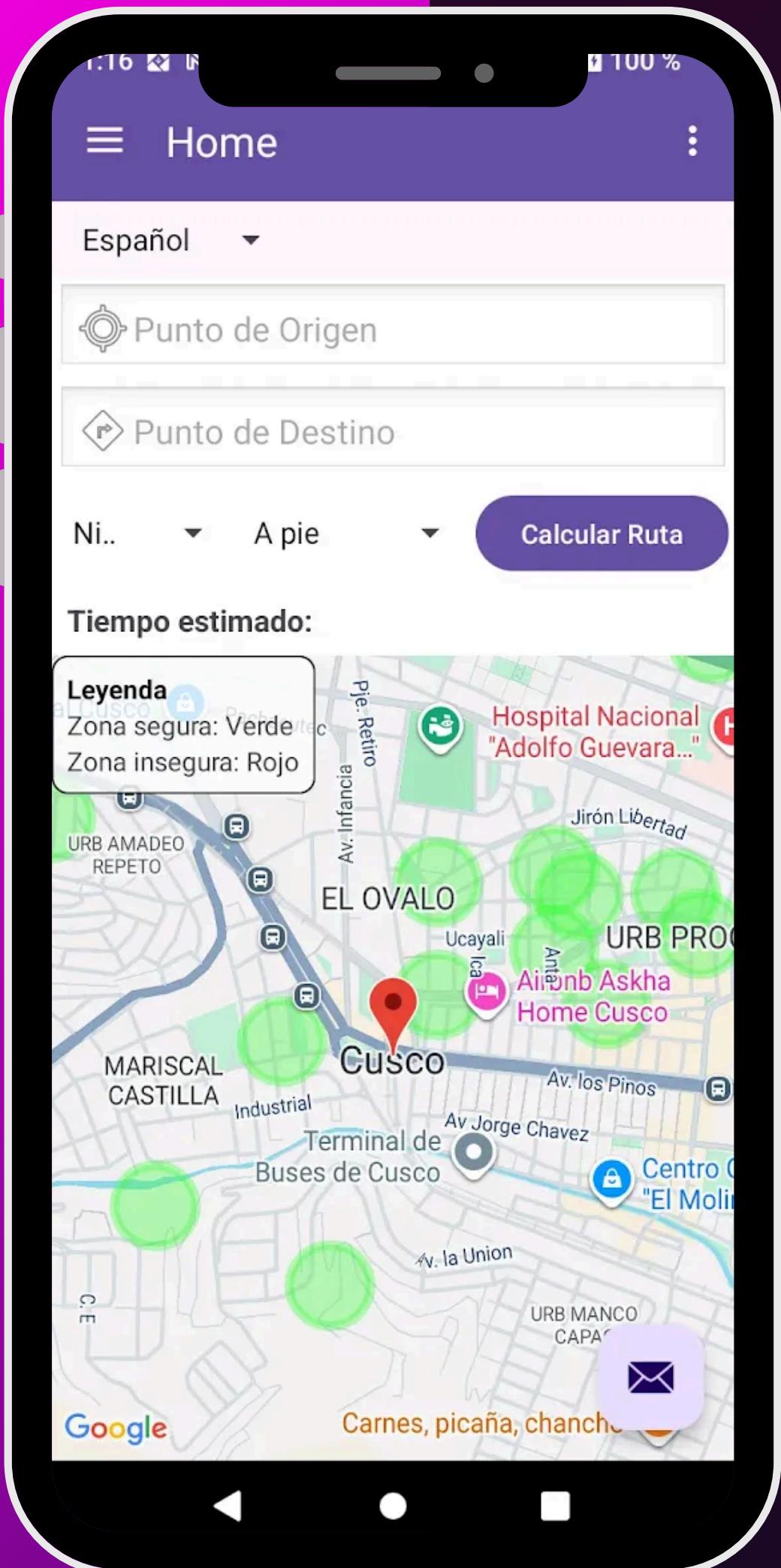
        if (mMap == null || ultimaUbicacion == null) return;

        if (position == 0) {
            mMap.setMapType(0);
            ultimaUbicacion.setZonaRiesgo("Ninguno");
        } else if (position == 1) {
            mMap.setMapType(1);
            ultimaUbicacion.setZonaRiesgo("Zonas de riesgo");
        } else if (position == 2) {
            mMap.setMapType(2);
            ultimaUbicacion.setZonaRiesgo("Lugares turísticos");
        } else if (position == 3) {
            mMap.setMapType(0);
            ultimaUbicacion.setZonaRiesgo("Ambos");
        }
    }
});
```



Configurar API de rutas

```
private void calcularRuta(LatLng origen, LatLng destino) {  
    // Limpiar ruta anterior  
  
    if (rutaActual != null) {  
  
        rutaActual.remove();  
  
        rutaActual = null;  
    }  
  
    // Limpiar marcadores anteriores  
  
    if (marcadorOrigen != null) marcadorOrigen.remove();  
  
    if (marcadorDestino != null) marcadorDestino.remove();  
  
    // Agregar nuevos marcadores  
  
    marcadorOrigen = mMap.addMarker(new  
    MarkerOptions().position(origen).title("Origen"));  
  
    marcadorDestino = mMap.addMarker(new  
    MarkerOptions().position(destino).title("Destino"));  
  
    // Construir URL según modo de transporte  
  
    String url = construirUrlDirecciones(origen, destino, modoTransporte);
```

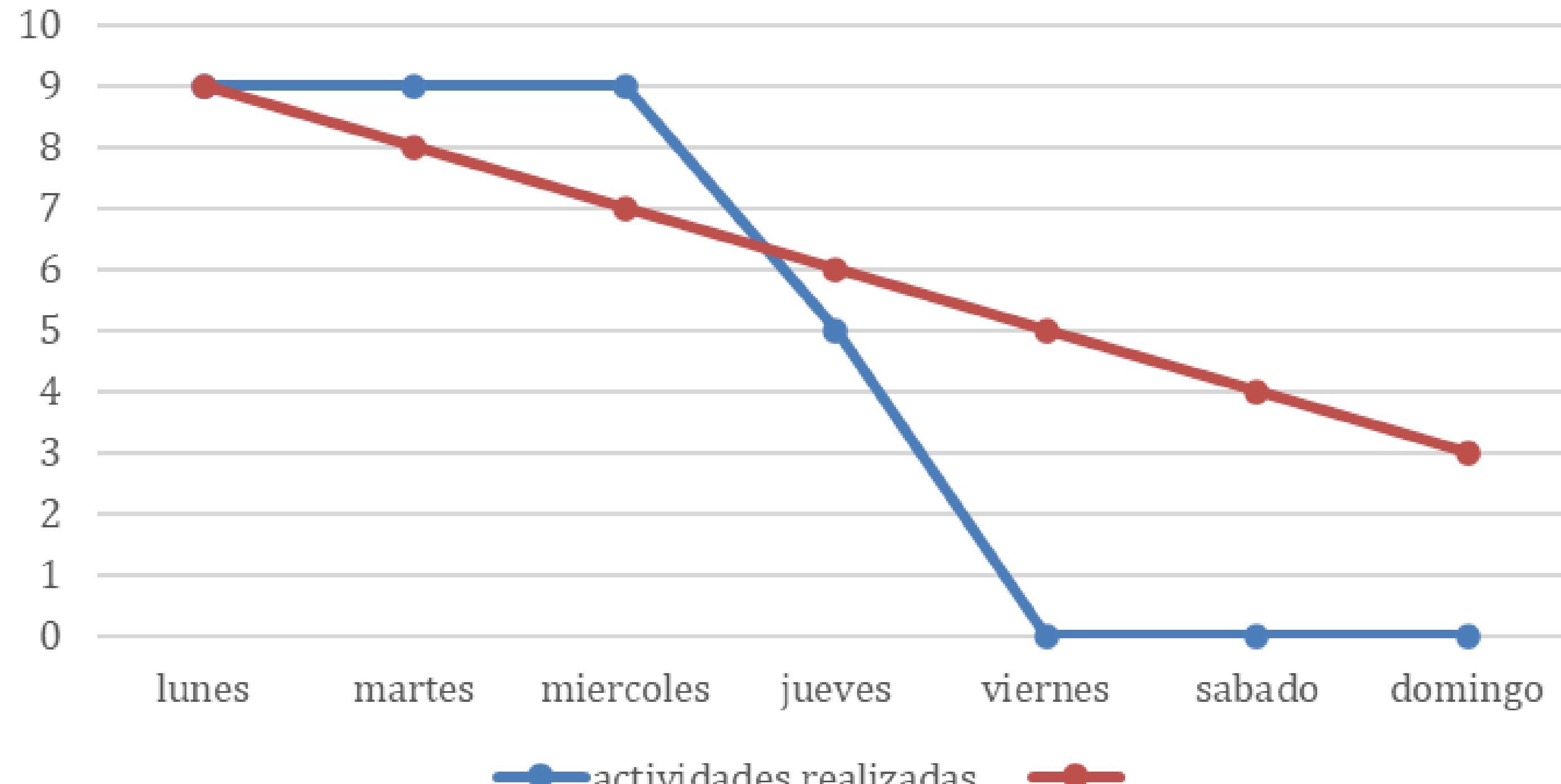
[Home](#)[Service](#)[About Us](#)[Contact](#)

Calcular tiempo

```
private void calcularTiempoAproximado(LatLng origen, LatLng destino) {  
    float[] distancia = new float[1];  
    Location.distanceBetween(  
        origen.latitude, origen.longitude,  
        destino.latitude, destino.longitude,  
        distancia  
    );  
  
    double velocidadMps;  
    switch (modoTransporte) {  
        case "driving":  
            velocidadMps = 4.17; // ~15 km/h promedio urbano realista  
            break;  
        case "bicycling":  
            velocidadMps = 2.78; // ~10 km/h  
            break;  
        default:  
    }  
}
```

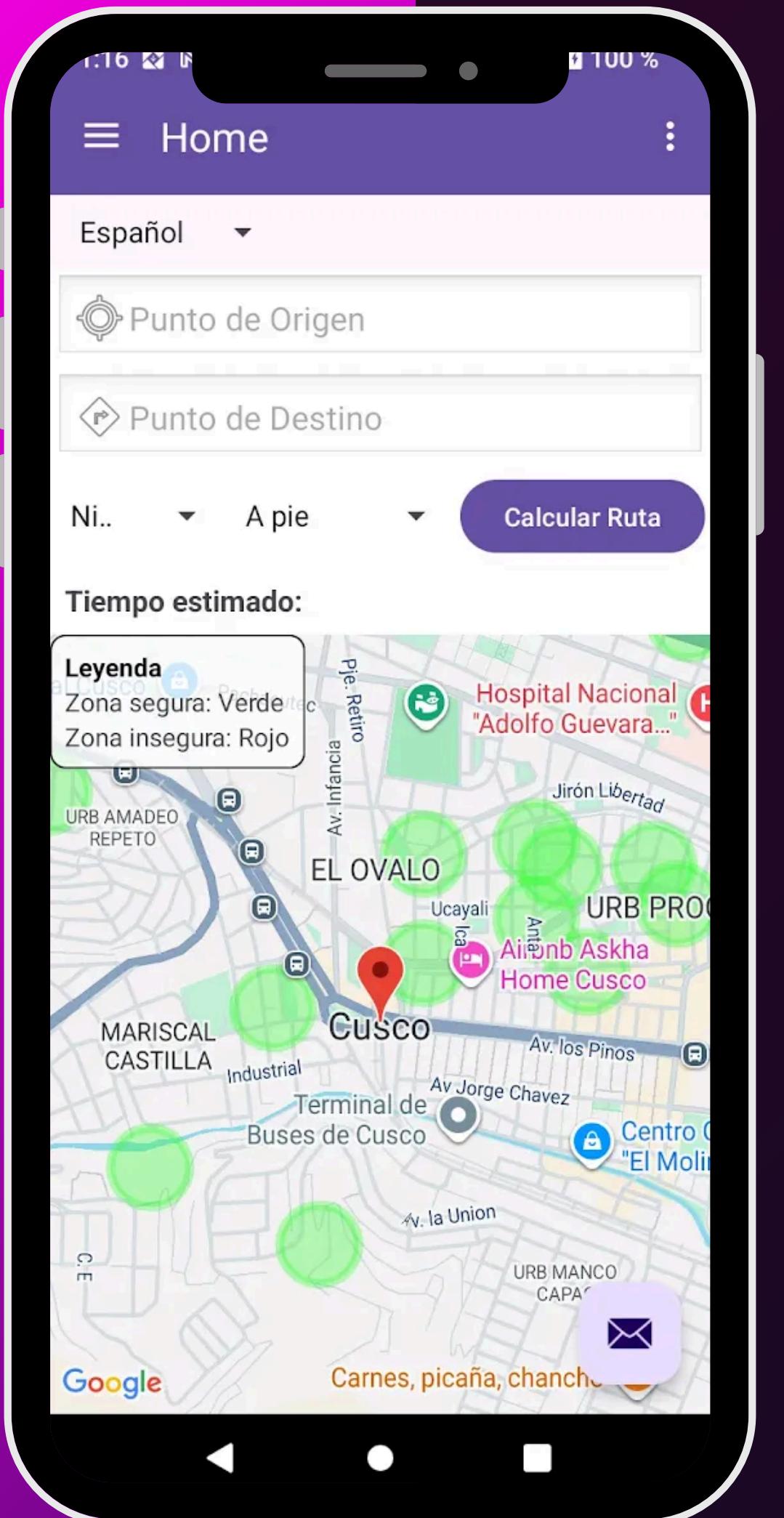
Nº	Responsable	Actividades - HU	Nombre	Por hacer	Haciendo	Hecho
1	Luis Cristian	HU-08	Implementar leyenda			Implementar leyenda
2	Luis Cristian	HU-09	Configurar capas inseguras			Configurar capas inseguras
3	Luis Cristian	HU-09	Colorear mapa por riesgo			Colorear mapa por riesgo
4	Frank	HU-09	Asociar datos con reportes			Asociar datos con reportes
5	Frank	HU-31	Implementar selector de idioma			Implementar selector de idioma
6	Frank	HU-31	Mostrar lugares turísticos			Mostrar lugares turísticos
7	Jhon Cristian	HU-31	Filtrar info relevante			Filtrar info relevante
8	Jhon Cristian	HU-10	Configurar API de rutas			Configurar API de rutas
9	Jhon Cristian	HU-10	Calcular tiempo			Calcular tiempo

Sprint 3



SPRINT N°4





- **Mostrar en mapa(ruta rápida)**
- **Integrar reportes ciudadanos**
- **Algoritmo de rutas seguras**
- **Comparar con ruta rápida**
- **Mostrar opciones (rápida vs segura)**
- **Permitir selección**
- **Mostrar métricas (tiempo/distancia/riesgo)**
- **Configurar notificaciones**
- **Asociar a nivel de riesgo**

SPRINT

Nº4

Nº	Responsable	Actividades HU	Nombre	Por hacer	Haciendo	Hecho
1	Luis Cristian	HU-08	Mostrar en mapa			Mostrar en mapa
2	Luis Cristian	HU-09	Integrar reportes ciudadanos			Integrar reportes ciudadanos
3	Luis Cristian	HU-09	Algoritmo de rutas seguras			Algoritmo de rutas seguras
4	Frank	HU-09	Comparar con ruta rápida			Comparar con ruta rápida
5	Frank	HU-31	Mostrar opciones (rápida vs segura)			Mostrar opciones (rápida vs segura)
6	Frank	HU-31	Permitir selección			Permitir selección
7	Jhon Cristian	HU-31	Mostrar métricas (tiempo/distancia/riesgo)			Mostrar métricas (tiempo/distancia/riesgo)
8	Jhon Cristian	HU-10	Configurar notificaciones	Configurar notificaciones		
9	Jhon Cristian	HU-10	Asociar a nivel de riesgo	Asociar a nivel de riesgo		

SPRINT

Nº4

