# **Documentación – Visualización de Zonas y Leyenda**

## **Objetivo**

Mostrar en el mapa las zonas seguras e inseguras, colorear según nivel de riesgo y agregar una leyenda para que el usuario entienda los colores.

## **1. Implementación**

### **a) Colorear mapa según 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("#55FFA500"); break; // Naranja

default: color = Color.parseColor("#55FF3333"); break; // Rojo

}

mMap.addCircle(new CircleOptions()

.center(zonas[i])

.radius(100)

.strokeColor(color)

.fillColor(color));

}

}

### **b) Configurar capas inseguras**

Se agregan marcadores y se permite reportar nuevas zonas inseguras al tocar el mapa:

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);

});

### **c) Implementar leyenda**

Se agrega una vista flotante con leyenda sobre el mapa:

FrameLayout mapContainer = getView().findViewById(R.id.mapContainer);

View legendView = LayoutInflater.from(getContext())

.inflate(R.layout.legend\_layout, mapContainer, false);

mapContainer.addView(legendView);

## **2. Observaciones**

* Verde = zona segura, rojo = zona de alto riesgo, naranja = riesgo medio.
* Los reportes de zonas inseguras se almacenan en Firestore.
* La leyenda permite al usuario interpretar fácilmente los colores del mapa.