| Comparar con ruta rápida |
| --- |
| Mostrar (rápida vs segura) |
| Permitir selección |

**Comparar con ruta rápida**

### Decidiendo la Mejor Ruta

Lo que hace es:

1. Recorre todas las rutas alternativas que ofrece Google.
2. Para cada una, calcula la distancia total para encontrar la **más corta**.
3. También calcula un **"puntaje de riesgo"** revisando si la ruta pasa cerca de alguna zona reportada como insegura. La que tenga el puntaje más bajo es la **más segura**.
4. Finalmente, compara ambas y decide cómo dibujarlas.

// En HomeFragment.java

private void procesarRespuestaDeRutas(JSONObject response) {

try {

String status = response.getString("status");

if (!status.equals("OK")) {

// ... manejo de errores ...

return;

}

JSONArray routes = response.getJSONArray("routes");

if (routes.length() == 0) {

// ... manejo si no hay rutas ...

return;

}

int mejorPuntajeRiesgo = Integer.*MAX\_VALUE*;

int rutaMasSeguraIndex = -1;

int rutaMasCortaIndex = 0;

long menorDistancia = Long.*MAX\_VALUE*;

// --- 1. Bucle para analizar cada ruta alternativa ---

for (int i = 0; i < routes.length(); i++) {

JSONObject route = routes.getJSONObject(i);

JSONObject leg = route.getJSONArray("legs").getJSONObject(0);

// --- 2. Encontrar la ruta MÁS CORTA (por distancia) ---

long distanciaActual = leg.getJSONObject("distance").getLong("value");

if (distanciaActual < menorDistancia) {

menorDistancia = distanciaActual;

rutaMasCortaIndex = i;

}

// --- 3. Calcular el RIESGO de esta ruta ---

String polyline = route.getJSONObject("overview\_polyline").getString("points");

List<LatLng> puntos = decodePolyline(polyline);

int puntajeRiesgo = 0;

for (LatLng punto : puntos) {

for (Reporte reporte : listaDeReportes) {

float[] distancia = new float[1];

Location.*distanceBetween*(punto.latitude, punto.longitude, reporte.getLat(), reporte.getLng(), distancia);

if (distancia[0] < 75) { // Si el punto de la ruta está a menos de 75m de un reporte

if (reporte.getRiesgo() == 2) puntajeRiesgo += 1; // Suma 1 por riesgo moderado

if (reporte.getRiesgo() == 3) puntajeRiesgo += 5; // Suma 5 por riesgo alto

}

}

}

// --- 4. Encontrar la ruta MÁS SEGURA (menor puntaje de riesgo) ---

if (puntajeRiesgo < mejorPuntajeRiesgo) {

mejorPuntajeRiesgo = puntajeRiesgo;

rutaMasSeguraIndex = i;

}

}

String tiempoRutaCorta = routes.getJSONObject(rutaMasCortaIndex).getJSONArray("legs").getJSONObject(0).getJSONObject("duration").getString("text");

String tiempoRutaSegura = routes.getJSONObject(rutaMasSeguraIndex).getJSONArray("legs").getJSONObject(0).getJSONObject("duration").getString("text");

// --- 5. Lógica para MOSTRAR LAS OPCIONES ---

if (rutaMasCortaIndex == rutaMasSeguraIndex) {

// Si la más corta es también la más segura, solo se muestra una en verde

tvTiempo.setText("Ruta Óptima (corta y segura): " + tiempoRutaCorta);

String polyline = routes.getJSONObject(rutaMasCortaIndex).getJSONObject("overview\_polyline").getString("points");

String tagUnica = "Ruta Óptima: " + tiempoRutaCorta + " (Riesgo: " + mejorPuntajeRiesgo + ")";

dibujarRuta(decodePolyline(polyline), Color.*GREEN*, 20, tagUnica);

} else {

// Si son diferentes, se muestran ambas opciones

tvTiempo.setText("Comparando Rutas | Azul: Más Corta (" + tiempoRutaCorta + ") | Verde: Más Segura (" + tiempoRutaSegura + ")");

// Dibuja la ruta corta en azul

String polylineCorta = routes.getJSONObject(rutaMasCortaIndex).getJSONObject("overview\_polyline").getString("points");

String tagCorta = "Ruta más corta: " + tiempoRutaCorta;

dibujarRuta(decodePolyline(polylineCorta), Color.*BLUE*, 15, tagCorta);

// Dibuja la ruta segura en verde

String polylineSegura = routes.getJSONObject(rutaMasSeguraIndex).getJSONObject("overview\_polyline").getString("points");

String tagSegura = "Ruta más segura: " + tiempoRutaSegura + " (Riesgo: " + mejorPuntajeRiesgo + ")";

dibujarRuta(decodePolyline(polylineSegura), Color.*GREEN*, 20, tagSegura);

}

} catch (Exception e) {

// ... manejo de errores ...

}

}

**Mostrar (rápida vs segura)**

Se añadió un tag para que, cuando se toque la línea, te muestre un mensaje con su información.

// En HomeFragment.java

private void dibujarRuta(List<LatLng> puntos, int color, float ancho, String tag) {

if (mMap == null) return;

Polyline polyline = mMap.addPolyline(new PolylineOptions()

.addAll(puntos)

.color(color) // Color (Verde o Azul)

.width(ancho) // Grosor de la línea

.clickable(true)); // Hacemos que se pueda tocar

polyline.setTag(tag); // Le asignamos un texto para mostrarlo al hacer clic

polylinesActuales.add(polyline);

}