**SPRINT 5**

| Mostrar historial de rutas |
| --- |
| Crear formulario |

la implementación de la característica "Historial de Rutas". El objetivo es permitir que los usuarios guarden las rutas que calculan y puedan consultarlas más tarde. Al hacer clic en una ruta guardada en el historial, el usuario es redirigido al mapa, donde se traza automáticamente el trayecto seleccionado.

### **A. El Modelo de Datos: HistorialRuta.java**

Este archivo es la "plantilla" o POJO (Plain Old Java Object) que define la estructura de los datos que se guardarán en la colección historialRutas de Firestore.

package com.example.myapplicationf.Models;

import com.google.firebase.Timestamp;

import java.util.Date;

public class HistorialRuta {

private String userId;

private String origenNombre;

private String destinoNombre;

private double origenLat;

private double origenLng;

private double destinoLat;

private double destinoLng;

private Timestamp timestamp;

// Constructor vacío requerido por Firestore

public HistorialRuta() {}

// Constructor completo

public HistorialRuta(String userId, String origenNombre, String destinoNombre,

double origenLat, double origenLng, double destinoLat, double destinoLng,

Timestamp timestamp) {

this.userId = userId;

this.origenNombre = origenNombre;

this.destinoNombre = destinoNombre;

this.origenLat = origenLat;

this.origenLng = origenLng;

this.destinoLat = destinoLat;

this.destinoLng = destinoLng;

this.timestamp = timestamp;

}

// Getters y Setters para todos los campos

public String getUserId() { return userId; }

public void setUserId(String userId) { this.userId = userId; }

public String getOrigenNombre() { return origenNombre; }

public void setOrigenNombre(String origenNombre) { this.origenNombre = origenNombre; }

public String getDestinoNombre() { return destinoNombre; }

public void setDestinoNombre(String destinoNombre) { this.destinoNombre = destinoNombre; }

public double getOrigenLat() { return origenLat; }

public void setOrigenLat(double origenLat) { this.origenLat = origenLat; }

public double getOrigenLng() { return origenLng; }

public void setOrigenLng(double origenLng) { this.origenLng = origenLng; }

public double getDestinoLat() { return destinoLat; }

public void setDestinoLat(double destinoLat) { this.destinoLat = destinoLat; }

public double getDestinoLng() { return destinoLng; }

public void setDestinoLng(double destinoLng) { this.destinoLng = destinoLng; }

public Timestamp getTimestamp() { return timestamp; }

public void setTimestamp(Timestamp timestamp) { this.timestamp = timestamp; }

}

### **B. La Pantalla del Mapa: HomeFragment.java**

1. **Recibir y dibujar una ruta del historial:** Comprueba si recibe datos (coordenadas) al ser creado. Si es así, traza la ruta automáticamente.

### **\* Código para Guardar la Ruta en Firestore**

Esta funcionalidad se activa cuando el usuario presiona el botón "Guardar Ruta". Se divide en dos partes: el *listener* del botón y el método que hace el trabajo de guardar.

**-** El onClickListener del Botón btnGuardarRuta:

Este bloque de código, que se encuentra en el método onCreateView, es el que "escucha" cuándo el usuario presiona el botón. Verifica que haya una ruta válida antes de llamar al método que la guardará.

btnGuardarRuta.setOnClickListener(v -> {

if (origenLatLng != null && destinoLatLng != null && origenNombre != null && destinoNombre != null) {

// Si hay una ruta válida, llama al mé*todo para guardarla*

guardarRutaEnHistorial();

} else {

// Si no, muestra un mensaje de advertencia

Toast.makeText(requireContext(), getString(R.string.historial\_selecciona\_ruta), Toast.LENGTH\_SHORT).show();

}

});

**- El Método guardarRutaEnHistorial():**

Este es el método que se encarga de toda la lógica de guardado. Crea un objeto HistorialRuta con todos los datos necesarios (ID del usuario, nombres, coordenadas y fecha/hora) y lo envía a la colección historialRutas en tu base de datos de Firestore.

private void guardarRutaEnHistorial() {

// Obtiene el usuario actual para asignarle la ruta

FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();

if (user == null) {

Toast.makeText(getContext(), getString(R.string.historial\_no\_login), Toast.LENGTH\_SHORT).show();

return;

}

String uid = user.getUid();

// Crea un nuevo objeto "HistorialRuta" con todos los datos

HistorialRuta nuevaRuta = new HistorialRuta(

uid,

origenNombre,

destinoNombre,

origenLatLng.latitude,

origenLatLng.longitude,

destinoLatLng.latitude,

destinoLatLng.longitude,

new Timestamp(new Date()) // Añade la fecha y hora actual

);

// Guarda el objeto en la colección "historialRutas" de Firestore

db.collection("historialRutas")

.add(nuevaRuta)

.addOnSuccessListener(documentReference -> Toast.makeText(getContext(), getString(R.string.historial\_ruta\_guardada), Toast.LENGTH\_SHORT).show())

.addOnFailureListener(e -> Toast.makeText(getContext(), getString(R.string.historial\_error\_guardar), Toast.LENGTH\_SHORT).show());

}

### **\* Código para Recibir y Dibujar la Ruta del Historial**

Esta es la parte que se activa cuando el usuario hace clic en un elemento del historial y es redirigido al mapa. Se encuentra dentro del método onMapReady.

**A. Bloque para Recibir los Datos:**

Este código comprueba si el HomeFragment recibió un "paquete" de datos (Bundle) con las coordenadas de la ruta. Si los encuentra:

1. Extrae las coordenadas y los nombres.
2. Rellena los campos de texto de origen y destino en la pantalla.
3. Simula un clic en el botón "Calcular Ruta" (btnCalcularRuta.performClick()) para que la ruta se trace automáticamente.

if (getArguments() != null && getArguments().containsKey("origenLat")) {

Bundle args = getArguments();

// Extrae todos los datos del Bundle

origenLatLng = new LatLng(args.getDouble("origenLat"), args.getDouble("origenLng"));

destinoLatLng = new LatLng(args.getDouble("destinoLat"), args.getDouble("destinoLng"));

origenNombre = args.getString("origenNombre");

destinoNombre = args.getString("destinoNombre");

// Rellena los campos de texto para que el usuario vea la ruta seleccionada

etOrigen.setText(origenNombre);

etDestino.setText(destinoNombre);

// Dibujar la ruta automáticamente después de un breve retraso

new Handler().postDelayed(() -> {

if (mMap != null) {

// Simula un clic en el botón para trazar la ruta

btnCalcularRuta.performClick();

}

}, 500); // 500ms de espera para asegurar que el mapa esté listo

}

### **C. El Diseño de la Lista: item\_historial\_ruta.xml y fragment\_slideshow.xml**

Estos archivos definen la apariencia de la pantalla de historial.

### **item\_historial\_ruta.xml**

<?xml version="1.0" encoding="utf-8"?>

<androidx.cardview.widget.CardView xmlns:android="[http://schemas.android.com/apk/res/android](http://schemas.android.com/apk/res/android)"

xmlns:app="[http://schemas.android.com/apk/res-auto](http://schemas.android.com/apk/res-auto)"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginHorizontal="8dp"

android:layout\_marginVertical="4dp"

app:cardCornerRadius="8dp"

app:cardElevation="2dp">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:padding="16dp">

<TextView

android:id="@+id/tvOrigenHistorial"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textStyle="bold"

android:textSize="16sp"

android:textColor="@android:color/black"/>

<TextView

android:id="@+id/tvDestinoHistorial"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="4dp"

android:textSize="16sp"

android:textColor="@android:color/black"/>

<TextView

android:id="@+id/tvFechaHistorial"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="12dp"

android:textSize="12sp"

android:textColor="@android:color/darker\_gray"/>

</LinearLayout>

</androidx.cardview.widget.CardView>

### **fragment\_slideshow.xml**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="[http://schemas.android.com/apk/res/android](http://schemas.android.com/apk/res/android)"

xmlns:tools="[http://schemas.android.com/tools](http://schemas.android.com/tools)"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".ui.slideshow.SlideshowFragment">

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/recyclerViewHistorial"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="8dp"

tools:listitem="@layout/item\_historial\_ruta" />

<TextView

android:id="@+id/tvEmptyHistory"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:text="@string/historial\_vacio"

android:textSize="18sp"

android:visibility="gone"

tools:visibility="visible"/>

</RelativeLayout>