

Desarrollo de Aplicaciones Móviles 2

Examen 01

Camila Sánchez, David Ruiz, Daniela Carvajal

Universidad Técnica Nacional

1. Configuración del entorno de desarrollo

Instalar dependencias necesarias

Node.js (descargar en nodejs.org)

Expo CLI (si usas Expo) o React Native CLI

Android Studio

Expo:

npx create-expo-app nombre-del-proyecto

cd nombre-del-proyecto

npm install

2. Instalación de paquetes adicionales:

npm install @react-native-picker/picker

npm install react-redux @reduxjs/toolkit

npm install @react-navigation/native

npm install react-native-sqlite-storage

npm install react-native-gesture-handler react-native-reanimated react-native-screens react-native-safe-area-context react-native-vector-icons

npm install @react-navigation/stack @react-navigation/bottom-tabs

3. Configurar SQLite para guardar favoritos

Crear la carpeta components/ y dentro de ella, el archivo creaDB.js:

```
import * as SQLite from "expo-sqlite";
```

```
const db = SQLite.openDatabase("favoritos.db");
```

```
export const getConnection = () => {
```

```
  return new Promise((resolve, reject) => {
```

```
    db.transaction((tx) => {
```

```
      tx.executeSql(
```

```
        "CREATE TABLE IF NOT EXISTS favoritos (id INTEGER PRIMARY KEY  
        AUTOINCREMENT, place_id TEXT, name TEXT, city TEXT);",
```

```

    [],
    () => resolve(db),
    (_, error) => reject(error)
  );
});
});
};

export const insertPlace = (db, place) => {
  return new Promise((resolve, reject) => {
    db.transaction((tx) => {
      tx.executeSql(
        "INSERT INTO favoritos (place_id, name, city) VALUES (?, ?, ?);",
        [place.place_id, place.name, place.city],
        (_, result) => resolve(result),
        (_, error) => reject(error)
      );
    });
  });
};

```

4. Crear la pantalla principal (Home.js)

```
import React, { useEffect, useState } from "react";
```

```
import { View, Text, TextInput, FlatList, TouchableOpacity, Button } from "react-native";  
  
import { Picker } from "@react-native-picker/picker";  
  
import { useDispatch, useSelector } from "react-redux";  
  
import { fetchLugares } from "../redux/lugaresSlice";  
  
import { useNavigation } from "@react-navigation/native";  
  
import styles from "../styles/parques";  
  
import { getConnection, insertPlace } from "../components/creaDB";
```

```
const Home = () => {  
  
  const dispatch = useDispatch();  
  
  const navigation = useNavigation();  
  
  const lugares = useSelector((state) => state.lugares.data);
```

```
  
  const [location, setLocation] = useState("Barcelona");  
  
  const [category, setCategory] = useState("poi");  
  
  const [keyword, setKeyword] = useState("");  
  
  const [source, setSource] = useState("GooglePlaces");
```

```
  
  useEffect(() => {  
  
    dispatch(fetchLugares({ location, category, keyword, source, limit: 10 }));  
  
  }, [location, category, keyword, source]);
```

```
  
  const handleAddFavorite = async (item) => {  
  
    try {
```

```

const db = await getConnection();

await insertPlace(db, { place_id: item.id.toString(), name: item.name, city: location });

alert("Lugar agregado a favoritos!");

} catch (error) {

  console.error("Error al agregar a favoritos:", error);

  alert(error.message);

}

};

return (

  <View style={styles.container}>

    <Text style={styles.title}>Lugares Turísticos</Text>

    <Button title="Ver Mis Favoritos" onPress={() => navigation.navigate("Favoritos")}
color="#4CAF50" />

    <Picker selectedValue={location} onValueChange={({value}) => setLocation(value)}
style={styles.picker}>

      {[ "Amsterdam", "Barcelona", "Berlin", "Dubai", "London", "Paris", "Rome",
"Tuscany"].map((city) => (

        <Picker.Item key={city} label={city} value={city} />

      )))

    </Picker>

    <TextInput style={styles.searchInput} placeholder="Buscar..." value={keyword}
onChangeText={setKeyword} />

```

```

<FlatList
  data={lugares}
  keyExtractor={(item) => item.id.toString()}
  renderItem={({ item }) => (
    <TouchableOpacity style={styles.card} onPress={() => navigation.navigate("Detalles",
    { lugar: item })}>
      <Text style={styles.cardTitle}>{item.name}</Text>
      <TouchableOpacity style={styles.favoriteButton} onPress={() =>
handleAddFavorite(item)}>
        <Text style={styles.favoriteButtonText}>★ Agregar a Favoritos</Text>
      </TouchableOpacity>
    </TouchableOpacity>
  )}
/>
</View>
);
};

export default Home;

```

6. Configurar Navegación

Crea el archivo navigation/AppNavigator.js:

```
import React from "react";
```

```
import { createStackNavigator } from "@react-navigation/stack";
import { NavigationContainer } from "@react-navigation/native";
import Home from "../screens/Home";
import Favoritos from "../screens/Favoritos";
import Detalles from "../screens/Detalles";

const Stack = createStackNavigator();

const AppNavigator = () => {
  return (
    <NavigationContainer>
      <Stack.Navigator>
        <Stack.Screen name="Home" component={Home} />
        <Stack.Screen name="Favoritos" component={Favoritos} />
        <Stack.Screen name="Detalles" component={Detalles} />
      </Stack.Navigator>
    </NavigationContainer>
  );
};

export default AppNavigator;
```

7. Crear la pantalla de Favoritos

Crea el archivo views /Favoritos.js:

```
import React, { useEffect, useState } from "react";

import { View, Text, FlatList, TouchableOpacity } from
"react-native";

import { getConnection } from "../components/creaDB";

const Favoritos = () => {
  const [favoritos, setFavoritos] = useState([]);

  useEffect(() => {
    const fetchFavoritos = async () => {
      const db = await getConnection();
      db.transaction((tx) => {
        tx.executeSql("SELECT * FROM favoritos", [], (
        { rows }) => {
          setFavoritos(rows._array);
        });
      });
    };
  });
};
```



```
    fetchFavoritos();  
  }, []);  
  
  return (  
    <View>  
      <Text>Mis Lugares Favoritos</Text>  
      <FlatList  
        data={favoritos}  
        keyExtractor={(item) => item.id.toString()}  
        renderItem={({ item }) => (  
          <TouchableOpacity>  
            <Text>{item.name} - {item.city}</Text>  
          </TouchableOpacity>  
        )}  
      />  
    </View>  
  );  
};
```

export default Favoritos;

8. run:

npm start

9. Be happy :)



