
Flutter

Actividad 5. Splash, Routes y delete city

Profesor

Pau Carrera

Vicente Català

Test unitario

1. Crea un splash screen con una imagen a modo de logotipo, un texto con un slogan y una animación de carga.
2. Organiza el proyecto para poder centralizar la gestión de rutas.
3. La aplicación tiene que empezar con la pantalla splash, pero se tiene que fijar en el main. La segunda pantalla será la de login, pero una vez se haya cargado no se podrá navegar a la pantalla splash, del mismo modo, una vez se inicie sesión tampoco se podrá volver al login.
4. En la pantalla de detalle del país tendrás que añadir un botón donde se mostrará un diálogo, permitiendo borrar la ciudad.
5. Al borrar una ciudad se realizará una navegación a la pantalla de listado de ciudades y se tendrá que actualizar el array de ciudades para que no se vuelva a mostrar.

1. Crea un Splash Screen con una imagen a modo de logotipo, un texto con un slogan y una animación de carga.

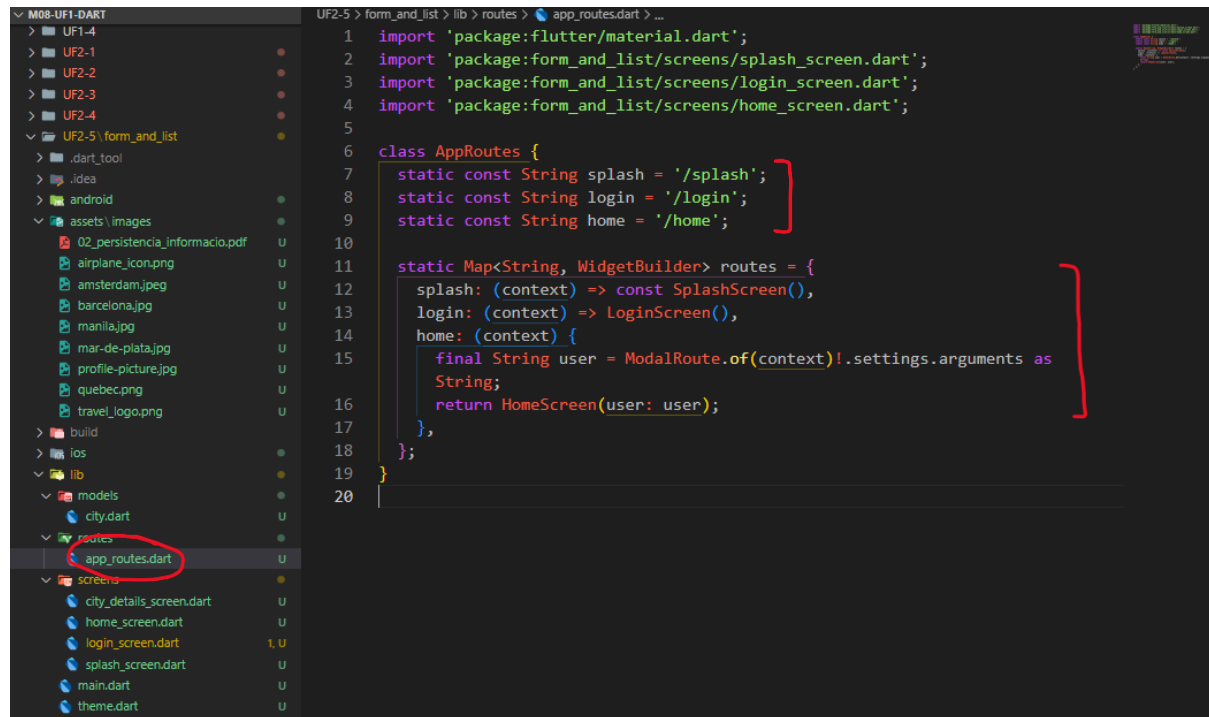
En esta actualización de la *SplashScreen*, además de la barra de carga animada y el avión en movimiento, he añadido el logo de la aplicación (travel_logo.png) y un eslogan para mejorar la presentación. Para ello, he utilizado un `Image.asset` con un tamaño de 150x150 píxeles para mostrar el logo en la parte superior. Justo debajo, he incluido un `Text` con el mensaje *"Explore the world, one city at a time."*, aplicando un estilo en cursiva con `fontStyle: FontStyle.italic`.



```
48     mainAxisSize: MainAxisSize.min,  
49     children: [  
50       Image.asset(  
51         'assets/images/travel_logo.png',  
52         width: 150,  
53         height: 150,  
54       ),  
55       const SizedBox(height: 10),  
56  
57       const Text(  
58         "Explore the world, one city at a time.",  
59         style: TextStyle(fontSize: 16, fontStyle: FontStyle.italic),  
60       ),  
61       const SizedBox(height: 30),  
62     ],
```

2. Organiza el proyecto para poder centralizar la gestión de rutas.

En la práctica anterior, ya había centralizado la gestión de rutas en la clase AppRoutes, lo que facilita la navegación dentro de la aplicación. Había definido constantes para cada ruta (splash, login y home), asegurando que los nombres sean consistentes en toda la aplicación. Además, había creado un Map<String, WidgetBuilder> donde cada ruta está asociada a su correspondiente pantalla. En el caso de home, he utilizado ModalRoute.of(context)!.settings.arguments para recibir el nombre de usuario desde la pantalla de inicio de sesión y pasarlo a HomeScreen.



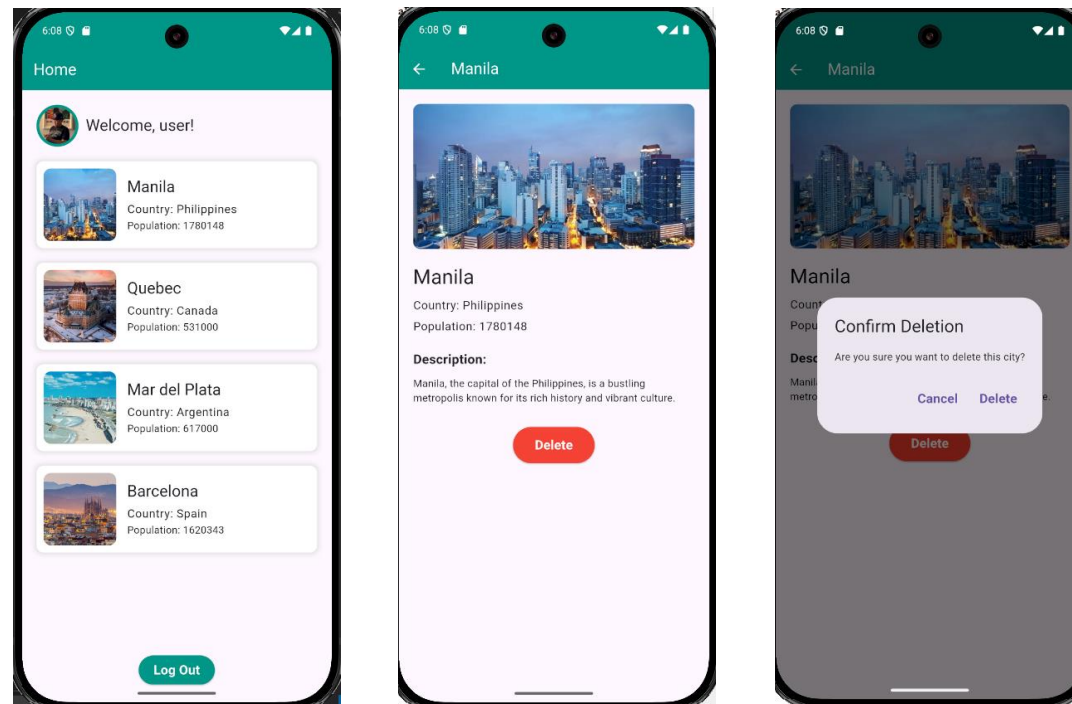
3. La aplicación tiene que empezar con la pantalla splash, pero se tiene que fijar en el main. La segunda pantalla será la de login, pero una vez se haya cargado no se podrá navegar a la pantalla splash, del mismo modo, una vez se inicie sesión tampoco se podrá volver al login.

En la aplicación, la navegación comienza desde la pantalla de carga (SplashScreen) y está controlada por las rutas definidas en AppRoutes. Una vez la pantalla splash ha cargado, se redirige automáticamente a la pantalla de inicio de sesión (LoginScreen) utilizando la ruta definida para ello. Para garantizar que el usuario no pueda regresar a la pantalla splash después de que se haya cargado ni a la pantalla de login después de iniciar sesión, se utiliza Navigator.pushReplacementNamed. Esto asegura que, al navegar a la pantalla de inicio de sesión o a la pantalla principal (HomeScreen), la pantalla anterior es reemplazada, evitando que el usuario pueda ir atrás en la navegación

```
1  import 'package:flutter/material.dart';
2  import 'package:form_and_list/theme.dart';
3  import 'package:form_and_list/routes/app_routes.dart';
4
5  Run | Debug | Profile
6  void main() {
7    runApp(const MainApp());
8  }
9
10 class MainApp extends StatelessWidget {
11   const MainApp({super.key});
12
13   @override
14   Widget build(BuildContext context) {
15     return MaterialApp(
16       debugShowCheckedModeBanner: false,
17       theme: appTheme,
18       initialRoute: AppRoutes.splash, ←
19       routes: AppRoutes.routes,
20     );
21   }
22 }
```

4. En la pantalla de detalle del país tendrás que añadir un botón dónde se mostrará un diálogo, permitiendo borrar la ciudad.

Se ha agregado un botón de "Delete" en la pantalla de detalles de la ciudad (CityDetailsScreen). Al presionar este botón, se muestra un cuadro de diálogo de confirmación que pregunta al usuario si está seguro de querer eliminar la ciudad seleccionada. Para esto, se utiliza el método showDialog junto con un AlertDialog, que contiene dos botones: "Cancel" para cerrar el cuadro de diálogo sin realizar ninguna acción, y "Delete" para proceder con la eliminación de la ciudad. Al confirmar la eliminación, se utiliza el método Navigator.pop para regresar a la pantalla anterior, pasando el objeto city eliminado como resultado.



5. Al borrar una ciudad se realizará una navegación a la pantalla de listado de ciudades y se tendrá que actualizar el array de ciudades para que no se vuelva a mostrar.

En la pantalla principal (HomeScreen), se maneja este resultado dentro del ListView.builder, donde, si se recibe una ciudad eliminada, se actualiza la lista de ciudades mediante setState(), removiendo la ciudad de la lista.

