

Programación multimedia y dispositivos móviles

Actividad 4.4. App del tiempo:
menú Drawer.

Francisco José García Cutillas | 2FPGS_DAM



Índice

Ejercicio 1	3
-------------------	---

Ejercicio 1

Se pide incorporar un menú tipo Drawer con tres opciones: Previsión, Selección de Municipio y Créditos.

La primera opción será la pantalla principal de la aplicación, y mostrará la pantalla con la previsión meteorológica, realizada anteriormente.

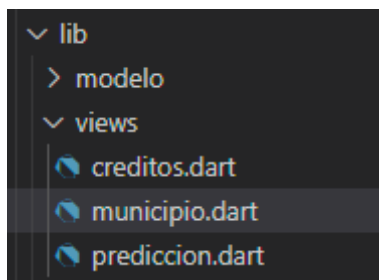
La segunda mostrará una lista con los diferentes municipios para que el usuario pueda seleccionar uno y este se guarde en la configuración de la aplicación.

La tercera simplemente mostrará una vista con los créditos y vuestro nombre.

Para la realización de esta práctica vamos a partir del proyecto ya creado para el ejercicio 4.3, por lo que aquí sólo se mostrarán los pasos de todos aquellos componentes que se han añadido al mismo.

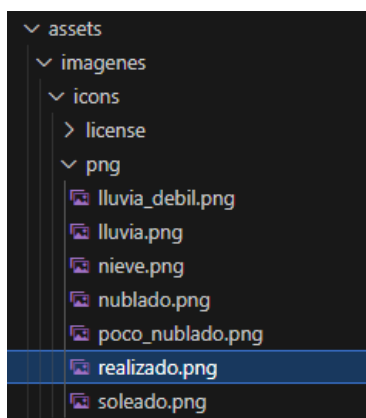
Para comenzar vamos a crear las pantallas de selección de municipio, de créditos y de predicción, pero sin mostrar nada por ellas de momento.

Lo vamos a organizar de la siguiente manera:



Para tenerlo todo más organizado, creamos el directorio “views” que será donde se guardarán las vistas de nuestra aplicación.

Para los créditos hemos añadido un icono nuevo, el cual hemos metido en el directorio assets/imagenes/icons/png



También lo tenemos que añadir al pubspec.yaml.

```

54
55 # The following section is specific to Flutter packages.
56 flutter:
57
58 # The following line ensures that the Material Icons font is
59 # included with your application, so that you can use the icons in
60 # the material Icons class.
61 uses-material-design: true
62
63 # To add assets to your application, add an assets section, like this:
64 assets:
65   - assets/imagenes/background.jpg
66   - assets/imagenes/icons/png/lluvia_debil.png
67   - assets/imagenes/icons/png/lluvia.png
68   - assets/imagenes/icons/png/nieve.png
69   - assets/imagenes/icons/png/nublado.png
70   - assets/imagenes/icons/png/poco_nublado.png
71   - assets/imagenes/icons/png/soleado.png
72   - assets/imagenes/icons/png/realizado.png
73   - assets/municipios_por_provincia.json
74

```

Comenzamos con la clase prediccion.dart, que contendrá lo siguiente.

```

lib > views > predicción.dart > MyAppTiempo
Click here to ask Blackbox to help you code faster
1 import 'package:flutter/material.dart';
2 import 'package:tiempo/modelo/prevision.dart';
3 import 'package:tiempo/obtener_prevision.dart';
4 import '../MainDrawer.dart';
5
6 //Clase principal
7 class MyAppTiempo extends StatelessWidget {
8   const MyAppTiempo({Key? key, required this.title}) : super(key: key);
9   final String title;
10
11   //Widget principal de la aplicación
12   @override
13   Widget build(BuildContext context) {
14     return Scaffold(
15       drawer: const MainDrawer(),
16       appBar: AppBar(
17         title: Text(title),
18       ), // AppBar
19       body: InfoTiempo(),
20     ); // Scaffold
21   }
22 }
23
24 //Widget para generar un estado a través de la información recibida de _InfoTiempo
25 class InfoTiempo extends StatefulWidget {
26   const InfoTiempo({Key? key}) : super(key: key);
27
28   @override
29   _InfoTiempo createState() => _InfoTiempo();
30 }
31
32 //Estado determinado por un objeto de tipo Prevision
33 class _InfoTiempo extends State<InfoTiempo> {
34   late Future<Prevision> _value;
35

```

```

36 //Estado inicial
37 @override
38 initState() {
39   super.initState();
40
41   //Obtenemos la previsión
42   _value = obtenerPrevision();
43 }
44
45 // Constructor del Widget que nos muestra la imagen de fondo
46 @override
47 Widget build(BuildContext context) {
48   return Container(
49     constraints: const BoxConstraints.expand(),
50     decoration: const BoxDecoration(
51       image: DecorationImage(
52         image: AssetImage("assets/imagenes/background.jpg"),
53         fit: BoxFit.cover)), // DecorationImage // BoxDecoration
54
55     //Detector de gestos
56     child: GestureDetector(
57       // Detección de la velocidad en vertical
58       onPanEnd: (DragEndDetails downDetails) {
59         //Si hacemos un deslizado en la pantalla hacia abajo, actualizamos previsión
60         if (downDetails.velocity.pixelsPerSecond.dy > 0) {
61           setState(() {
62             _value = obtenerPrevision();
63           });
64         }
65       },

```

```

66
67     //Padding de los widgets
68     child: Padding(
69       padding: const EdgeInsets.all(24.0),
70       child: Center(
71         child: FutureBuilder(
72           future: _value,
73
74           /*Con el builder establecemos la manera de formar el widget cuando
75           obtenemos la respuesta del Future*/
76           builder: (BuildContext context, AsyncSnapshot snapshot) {
77             //Si no tenemos respuesta, realizamos una animación de espera
78             if (snapshot.connectionState == ConnectionState.waiting) {
79               return const Center(
80                 child: SizedBox(
81                   height: 150.0,
82                   width: 150.0,
83                   child: CircularProgressIndicator(
84                     color: Colors.white,
85                   ), // CircularProgressIndicator
86                 ), // SizedBox
87               ); // Center
88             } else if (snapshot.connectionState == ConnectionState.done) {
89               //Comprobación de error
90               if (snapshot.hasError) {
91                 return const Text('Error');
92               }
93               //Si no hay error, generamos el widget con la previsión
94             } else if (snapshot.hasData) {
95               return MyInfoWeather(prevision: snapshot.data);
96             } else {
97               // Si no recibimos datos
98               return const Text('Sin datos');
99             }

```

```

100         } else {
101             // Mostramos cualquier otro estado
102             return Text('Estado: ${snapshot.connectionState}');
103         }
104     },
105     )), // FutureBuilder // Center // Padding
106     ), // GestureDetector
107     ); // Container
108 }
109 }
110
111 // Widget para el contenido central con la previsión
112 class MyInfoWeather extends StatefulWidget {
113     final Prevision prevision;
114     MyInfoWeather({Key? key, required this.prevision}) : super(key: key);
115
116     @override
117     State<MyInfoWeather> createState() => _MyInfoWeatherState();
118 }
119
120 class _MyInfoWeatherState extends State<MyInfoWeather> {
121     //Widget que nos mostrará la información en un layout vertical
122     @override
123     Widget build(BuildContext context) {
124         return Column(
125             children: <Widget>[
126                 //Imagen de la previsión
127                 Image(
128                     image: AssetImage(
129                         "assets/imagenes/icons/png/${widget.prevision.imagen}"), // AssetImage
130                     height: 220.0), // Image
131                 Center(
132                     //Nombre del municipio
133                     child: Text(
134                         widget.prevision.municipio,
135                         style: Theme.of(context).textTheme.displaySmall,
136                     ), // Text
137                 ), // Center

```

```

138                 Text(
139                     //Previsión
140                     "${widget.prevision.estado}",
141                     style: Theme.of(context).textTheme.headlineSmall,
142                 ), // Text
143                 Text(
144                     //Probabilidad de precipitaciones
145                     "\n Probabilidad de lluvia ${widget.prevision.precipitacion}%",
146                     style: Theme.of(context).textTheme.headlineSmall,
147                 ), // Text
148                 Padding(
149                     //Temperaturas
150                     padding: const EdgeInsets.all(24.0),
151                     child: Row(
152                         children: [
153                             Expanded(
154                                 child: Center(
155                                     child: Text(
156                                         "${widget.prevision.tMin}°",
157                                         style: Theme.of(context).textTheme.displayMedium,
158                                     ), // Text
159                                 ), // Center
160                             ), // Expanded
161                             Expanded(
162                                 child: Center(
163                                     child: Text(
164                                         "${widget.prevision.tMax}°",
165                                         style: Theme.of(context).textTheme.displayMedium,
166                                     ), // Text
167                                 ), // Center
168                             ), // Expanded
169                         ],
170                     ), // Row
171                 ), // Padding
172             ], // <Widget>[]
173         ); // Column
174     }
175 }

```

Una vez realizada la clase predicción.dart, nos pasamos a la clase municipio.dart, que contendrá lo siguiente.

```
lib > views > municipio.dart > ...
Click here to ask Blackbox to help you code faster
1 import 'dart:convert';
2 import 'package:flutter/material.dart';
3 import 'package:flutter/services.dart';
4 import 'package:tiempo/modelo/config.dart';
5 import '../MainDrawer.dart';
6
7 //Clase para crear la vista de la selección de municipio
8 class SeleccionMunicipio extends StatelessWidget {
9   const SeleccionMunicipio({Key? key}) : super(key: key);
10
11   //Constructor del widget
12   @override
13   Widget build(BuildContext context) {
14     return Scaffold(
15       drawer: const MainDrawer(),
16       appBar: AppBar(
17         title: const Text("Seleccione un municipio"),
18       ), // AppBar
19       body: const SelectorMunicipio(),
20     ); // Scaffold
21   }
22 }
23
24 //Clase crea el esqueleto del widget
25 class SelectorMunicipio extends StatefulWidget {
26   const SelectorMunicipio({Key? key}) : super(key: key);
27
28   @override
29   _EstadoSelector createState() => _EstadoSelector();
30 }
31
32 //Clase que obtiene los municipios y crea el widget
33 class _EstadoSelector extends State<SelectorMunicipio> {
34   late Future _municipios;
35
36   //Iniciamos obteniendo los municipios
37   @override
38   void initState() {
39     _municipios = obtenerMunicipios();
40     super.initState();
41   }
42 }
```

```

43 //Obtenemos la lista de municipios y hasta que no los tenemos no seguimos
44 Future<String> obtenerMunicipios() async =>
45 |   await rootBundle.loadString('assets/municipios_por_provincia.json');
46
47 //Constructor del widget
48 @override
49 Widget build(BuildContext context) {
50   return FutureBuilder(
51     future: _municipios,
52     builder: (context, AsyncSnapshot snapshot) {
53       if (!snapshot.hasData) {
54         return Text('Cargando municipios...');
55       }
56
57       //Cuando los datos están cargados, generamos una lista con los municipios
58       var municipios = json.decode(snapshot.data);
59       List lista_municipios = [];
60
61       for (var provincia in municipios["municipios"]) {
62         for (var municipio in provincia["municipios"]) {
63           lista_municipios.add([
64             municipio["nombre"],
65             municipio["codigo"],
66             provincia["provincia"]
67           ]);
68         }
69       }
70
71       //Ordenamos los municipios por nombre
72       lista_municipios.sort((m1, m2) => m1[0].compareTo(m2[0]));
73
74       //Creamos la vista
75       return ListView.builder(
76         itemCount: lista_municipios.length,
77         itemBuilder: (BuildContext context, int index) {
78           return ListTile(
79             title: Text(lista_municipios[index][0]),
80             subtitle: Text(lista_municipios[index][2]),
81
82           /*Cuando pulsemos sobre un municipio, cambiará el
83            código de config.dart y nos llevará a
84            la ventana de la predicción */
85           onTap: () {
86             Config.codigo = lista_municipios[index][1];
87             Navigator.of(context).pushReplacementNamed('/');
88           },
89         ); // ListTile
90       }); // ListView.builder
91     },
92   ); // FutureBuilder
93 }
94 }
95

```


Finalmente, del paquete views, generamos la clase créditos.dart. Que mostrará información sobre el desarrollador de la aplicación y un icono.

```
lib > views > credits.dart > Credits > build
Click here to ask Blackbox to help you code faster
1 import 'package:flutter/material.dart';
2
3 import '../MainDrawer.dart';
4
5 class Credits extends StatelessWidget {
6   const Credits({Key? key}) : super(key: key);
7
8   @override
9   Widget build(BuildContext context) {
10    return Scaffold(
11      drawer: const MainDrawer(),
12      appBar: AppBar(
13        title: const Text("Créditos"),
14      ), // AppBar
15      body: Container(
16        margin: const EdgeInsets.all(12.0),
17        decoration: BoxDecoration(
18          color: Colors.blueGrey[50],
19          shape: BoxShape.rectangle,
20          borderRadius: const BorderRadius.all(Radius.circular(25)), // BoxDecoration
21        alignment: Alignment.center,
22        padding: const EdgeInsets.all(16.0),
23        child: Column(children: [
24          Center(
25            child: Text("Aplicación del tiempo",
26              textAlign: TextAlign.center,
27              style: Theme.of(context).textTheme.headlineMedium)), // Text // Center
28          Center(
29            child: Text("\nRealizada por: Francisco José García Cutillas\n",
30              textAlign: TextAlign.center,
31              style: Theme.of(context).textTheme.titleLarge)), // Text // Center
32          const Center(
33            child: Expanded(
34              child: Padding(
35                padding: EdgeInsets.only(left: 10, right: 10),
36                child: Image(
37                  image: AssetImage("assets/imagenes/icons/png/realizado.png"),
38                ), // Image
39              ), // Padding
40            ), // Expanded
41          ), // Center
42        ]), // Column
43      ), // Container
44    ); // Scaffold
45  }
46 }
47
```

Ahora vamos a volver al raíz del paquete “lib” y creamos la clase MainDrawer.dart, que será la encargada de gestionar el menú de la aplicación.

```
lib > MainDrawer.dart > MainDrawer > build
Click here to ask Blackbox to help you code faster
1 import 'package:flutter/material.dart';
2
3 //Clase que define el Drawer
4 class MainDrawer extends StatelessWidget {
5   const MainDrawer({Key? key}) : super(key: key);
6
7   //Constructor del widget
8   @override
9   Widget build(BuildContext context) {
10    return Drawer(
11      child: ListView(
12        padding: EdgeInsets.zero,
13        children: [
14          DrawerHeader(
15            decoration: const BoxDecoration(
16              image: DecorationImage(
17                fit: BoxFit.cover,
18                image: AssetImage('assets/imagenes/background.jpg')), // DecorationImage // BoxDecoration
19            child: Text('Aplicación del tiempo',
20              style: Theme.of(context).textTheme.displaySmall), // Text
21            ), // DrawerHeader
22          ListTile(
23            title: Text('Predicción',
24              style: Theme.of(context).textTheme.titleLarge), // Text
25            onTap: () {
26              String currentRoute =
27                (ModalRoute.of(context)?.settings.name).toString();
28              if (currentRoute != '/') {
29                Navigator.of(context).pushReplacementNamed('/');
30              }
31            },
32          ), // ListTile
33          ListTile(
34            title: Text('Selección de municipio',
35              style: Theme.of(context).textTheme.titleLarge), // Text
36            onTap: () {
37              String currentRoute =
38                (ModalRoute.of(context)?.settings.name).toString();
39              if (currentRoute != '/selectorMunicipio') {
40                Navigator.of(context)
41                  .pushReplacementNamed('/selectorMunicipio');
42              }
43            },
44          ), // ListTile
45          ListTile(
46            title:
47              Text('Créditos', style: Theme.of(context).textTheme.titleLarge),
48            onTap: () {
49              String currentRoute =
50                (ModalRoute.of(context)?.settings.name).toString();
51              if (currentRoute != '/creditos') {
52                Navigator.of(context).pushReplacementNamed('/creditos');
53              }
54            },
55          ), // ListTile
56        ],
57      ), // ListView
58    ); // Drawer
59  }
60 }
61
```

Finalmente generamos la clase main.dart encargada de gestionar el conjunto de la aplicación.

```
lib > main.dart > ...
1 import 'package:flutter/material.dart';
2 import 'package:tiempo/views/creditos.dart';
3 import 'package:tiempo/views/prediccion.dart';
4 import 'package:tiempo/views/municipio.dart';
5
6 void main() {
7   runApp(const MyApp());
8 }
9
10 //Clase principal que ejecuta nuestra aplicación
11 class MyApp extends StatelessWidget {
12   const MyApp({Key? key}) : super(key: key);
13
14   //Constructor del widget de la aplicación
15   @override
16   Widget build(BuildContext context) {
17     return MaterialApp(
18       debugShowCheckedModeBanner: false,
19       theme: ThemeData(
20         colorScheme:
21           ColorScheme.fromSwatch().copyWith(primary: Colors.blueGrey),
22         fontFamily: 'Roboto',
23         textTheme: const TextTheme(
24           headlineSmall:
25             TextStyle(fontWeight: FontWeight.w300, color: Colors.white),
26           headlineMedium:
27             TextStyle(fontWeight: FontWeight.w400, color: Colors.white),
28           displaySmall:
29             TextStyle(fontWeight: FontWeight.w400, color: Colors.white),
30           displayMedium:
31             TextStyle(fontWeight: FontWeight.w300, color: Colors.white),
32         ), // TextTheme
33       ), // ThemeData
34
35       title: 'Aplicación del tiempo',
36       //Ruta inicial de la aplicación
37       initialRoute: '/',
38       //Rutas para las tres pantallas
39       routes: {
40         '/': (context) => const MyAppTiempo(title: 'Aplicación del tiempo'),
41         '/selectorMunicipio': (context) => const SeleccionMunicipio(),
42         '/creditos': (context) => const Creditos(),
43       },
44       //home: const MiAppTiempo(title: 'Mi App del Tiempo'),
45     ); // MaterialApp
46   }
47 }
```

Resumen de cómo quedaría la estructura del proyecto.

