Práctica 2. Widgets

Para esta tarea crearemos varias pantallas para una futura aplicación que proporcione información sobre las comarcas de la Comunitat Valenciana.

De momento, no añadiremos ninguna funcionalidad, solo el diseño de las pantallas para mostrar la información. Para ello se os proporcionará un proyecto de base (**comarquesguilnicial**), en el que se encuentra parte de la implementación hecha, y tendréis que completar y realizar algunas de esas pantallas.

En este proyecto de base, dispongo de la clase RepositoryExemple (fichero **lib/repository/repository_exemple.dart**) que contiene métodos estáticos para proporcionaros la información de provincias y comarcas a mostrar (en futuras tareas ya lo enlazaremos con la API).

Este proyecto, además, ya tiene implementada una pantalla de inicio con cuatro botones para abrir las diferentes vistas:

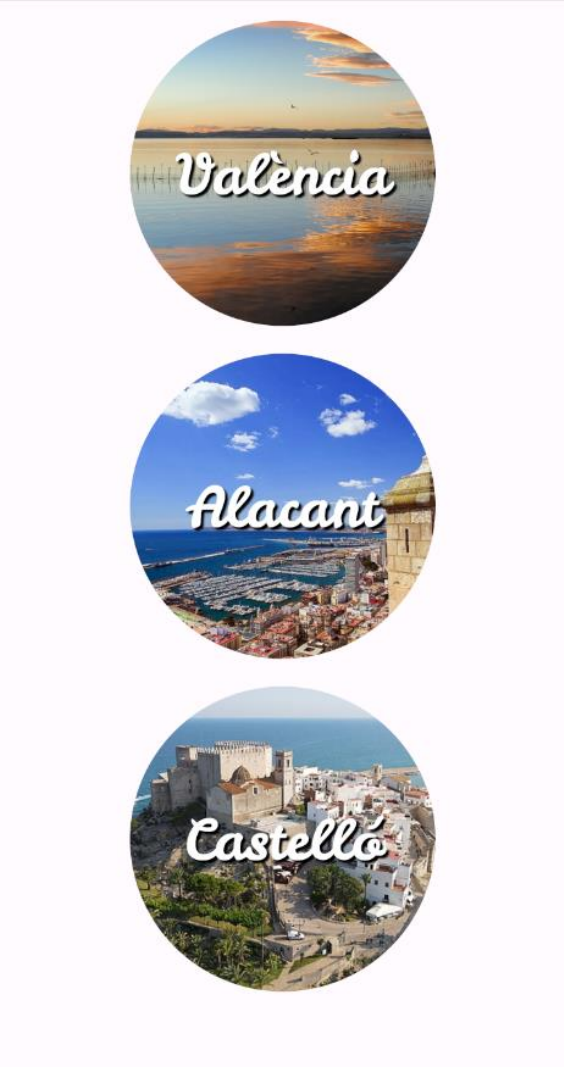
Pantalla inicial

Como veis, tendremos una pantalla donde se muestran las provincias, otra para comarcas, y un par más para información sobre las comarcas.

Vuestra tarea será completar la implementación de estas cuatro vistas.

Pantalla ProvinciesScreen

Esta pantalla mostrará las tres provincias haciendo uso del widget **CircleAvatar** con la imagen de cada provincia y su nombre dentro. La vista final será la siguiente:



Pantalla Provincias

Esta pantalla se implementa en la clase **ProvinciesScreen** (fichero lib/screens/provincies_screen.dart), y hace uso del repositorio para obtener la lista de provincias, concretamente, de la función estática **RepositoryExemple.obtenerProvincies()**, que nos devuelve una lista de objetos de tipo Provincia.

Como veréis en la clase, ya se os proporciona la mayor parte de la implementación, de manera que solo tendréis que implementar el método **build** del widget que representa cada provincia (**ProvinciaRoundButton**). El diseño general de la pantalla y la construcción de la lista de widgets, ya la tendréis creada. Echad un vistazo al resto de código y a los comentarios para entender bien su funcionamiento.

En esta clase, para simplificar, hemos incorporado el widget **ProvinciaRoundButton** al mismo fichero fuente que la clase **ProvinciesScreen**, pero podría perfectamente guardarse en un fichero externo, por ejemplo, ubicado en la carpeta **screens/widgets**.

Pantalla Comarques

En esta pantalla se mostrará una lista de comarcas, haciendo uso de **Cards** que combinan imagen y nombre de la comarca:

Pantalla Comarques

Esta pantalla se implementa en la clase **ComarcasScreen** (fichero lib/screens/comarques_screen.dart), y hace uso del repositorio para obtener la lista de objetos JSON con el nombre y la imagen de cada comarca de la lista. Concretamente, hace uso de la función estática **RepositoryExemple.obtenirComarques()**, que nos devuelve esta lista a modo de ejemplo para las comarcas de Alicante.

Como veis, y de igual manera que en la pantalla anterior, habrá que generar un widget personalizado para generar una tarjeta, e instanciarlo con la información de cada comarca.

En el proyecto de base se os proporciona parte de la estructura, de manera que tendréis que completar los TO-DOs que aparecen. Se os da más información al código.

Pantallas con información sobre la comarca

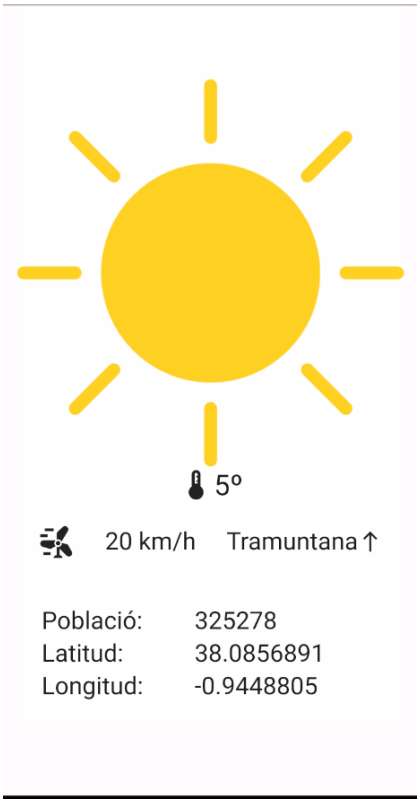
Finalmente, generaremos dos pantallas con información sobre una comarca en concreto. De momento, esta comarca será fija, y la obtendremos con el método **RepositoryExemple.obtenirInfoComarca()** del repositorio.

En estas dos pantallas mostraremos la información de la **Comarca** que nos devuelve el método anterior: comarca, capital, poblacio, imagen representativa (img), descripción (desc), y latitud y longitud (vector coordenadas).

Además, también maquetaremos un espacio en una de ellas para añadir una imagen sobre el tiempo (de momento será fija), así como la temperatura actual y la dirección del viento (esta información la obtendremos también en tareas posteriores)

El aspecto de estas dos pantallas podrá ser parecido a las siguientes:

Información básica sobre la comarca



Información ampliada sobre la comarca

Las clases para generar estas pantallas son **InfoComarcaGeneral** (fichero lib/screens/infocomarca_general.dart) e **InfoComarcaDetall** (fichero lib/screens/infocomarca_detall.dart), respectivamente.

Además, para la información detallada, se os proporciona un widget personalizado **MyWeatherInfo** (fichero lib/screens/widhets/my_weather_info.dart) para mostrar la información del clima en la comarca.

Incluyendo los Assets

Tal y como se os proporciona el proyecto, no tenéis acceso a las **tipografías** o a las imágenes para representar el tiempo. Estos recursos se os proporcionan en la carpeta **assets**, pero hay que registrarlos en el fichero del proyecto **pubspec.yaml**. Modificad este fichero de manera que podáis hacer uso de estos recursos en vuestra aplicación.

Estructura de la aplicación

Cuando empezamos a incorporar widgets a nuestras aplicaciones, es fácil que el código acabe considerablemente desordenado, y por eso es importante que procuremos llevar una organización lo más coherente y sencilla posible. Intentamos que nuestros widgets sean cortos y modulares, y que el código de la función **main** sea lo más simple posible.

El código que se os proporciona está dividido en varias carpetas, que posteriormente ampliaremos para seguir los patrones BLoC o Provider. Si conocéis la arquitectura Modelo-Vista-Controlador, o Model-View-ViewModel, puede que os suenen algunas de estas capas o carpetas.

La organización del código es la siguiente:

```
.
├── assets
│   ├── fonts
│   │   └── ...
│   ├── icons
│   │   └── ...
│   └── img
│       └── ...
├── lib
│   ├── main.dart
│   ├── models
│   │   ├── comarca.dart
│   │   └── provincia.dart
│   ├── repository
│   │   └── repository_exemple.dart
│   ├── screens
│   │   ├── comarques_screen.dart
│   │   ├── infocomarca_detall.dart
│   │   ├── infocomarca_general.dart
│   │   ├── launcher_screen.dart
│   │   ├── provincies_screen.dart
│   │   └── widgets
│   │       └── my_weather_info.dart
└── pubspec.yaml
```

Como vemos, tenemos:

- La carpeta **models** contiene los Modelos de nuestra aplicación, es decir, las clases **Comarca** y **Provincia**.
- La carpeta **repository** contiene la clase **RepositoryExemple**, que nos servirá posteriormente como punto de acceso a los datos de la aplicación. De momento, hacemos uso de esta clase con métodos estáticos para obtener la información de ejemplo.
- La carpeta **screens**, con las diferentes pantallas de la aplicación. Dentro de ella tenemos la carpeta widgets, con widgets auxiliares que vamos a utilizar.

Después tenemos el fichero main.dart, con la función main, y la carpeta de **assets**, con los diferentes recursos.

Entrega de la práctica

Para entregar la práctica recordad hacer **flutter clean** para eliminar todos los posibles build que se hayan construido y de esta forma hacer que el proyecto sea mucho menos pesado.

Se debe entregar además un pequeño **pdf** en el que se comenten las clases y widgets empleados para hacer las **implementaciones nuevas solicitadas en la práctica** con un apartado en el que se comenten las **dificultades encontradas**.

A continuación, comprimir la carpeta raíz junto con el pdf en un fichero .zip o .rar para hacer la entrega.