

Flutter

Desarrollo multiplataforma

Prof. Juan Sanz

Scaffold

- El widget Scaffold nos ayuda a organizar la disposición de nuestra pantalla, a tratar la navegación y a implementar el diseño de Material Design.
- Propiedades importantes: appBar, body, bottomNavigationBar.
 - appBar propiedad que define la parte superior del Scaffold
 - Body, pantalla principal del Scaffold
 - bottomNavigationBar, menú inferior del Scaffold, recibirá el widget NavigationBar

AppBar

- Title
- backgroundColor
- centerTitle
- Leading, icono que aparece antes del título
 - Recibe un IconButton

NavigationBar

- backgroundColor
- onDestinationSelected, función que hará cambiar la página principal (currentPageIndex)
- selectedIndex, cambia la pestaña seleccionada
- Destinations: Lista de NavigationDestination (botones que se muestran en la parte inferior)
 - NavigationDestination(
 selectedIndex
 icon
 label)

NavigationBar

- En el Scaffold, body tendrá una lista de Widgets y será seleccionado el 'currentPageIndex'.
- Body: [
 Screen1(),
 Screen2()][currentPageIndex]

Crear clase de datos

- Crear carpeta models
- Crear un nuevo archivo .dart

```
class Gasto{  
  String nombre;  
  int coste;  
  
  Gasto(this.nombre, this.coste);  
}
```

Stateful Widget

- Son aquellos que tienen un estado el cual pueden gestionar.
- Tienen dos clases
 - Clase del widget: configuración del widget
 - Clase del estado: reside el estado y sus posibles cambios
- Al modificar el estado, hay que ejecutar `SetState()` para que el framework vuelva a repintar el widget.

ListView.builder

- Como un ListView, pero para datos dinámicos.
- itemCount : tamaño de los datos a mostrar (lista.length)
- itemBuilder : (BuildContext context, int index){
 return Widget();
}

Gestión del estado

- Con [Provider](#), podemos compartir información con todo el árbol de Widgets.
- Para añadirlo al proyecto
 - Flutter pub add privoder
 - Flutter pub get
- Crearemos una clase “superior” de la cual escucharán los widgets que nosotros indiquemos mediante Consumidores.

Gestión del estado

- Crearemos .dart donde se encontrarán unos gatos para compartir con toda la app.
- Nuestro runApp() en el main, estará englobado por ChangeNotifierProvider().

```
runApp(  
  ChangeNotifierProvider(  
    create: (context) => GastoProvider(),  
    child: const MainApp(),  
  ),  
);
```

- Los widgets hijos podrán acceder a nuestra clase creada.

```
final gastos = context.watch<GastoProvider>();
```