Flutter

ListViews

Carmelo Villegas Cruz

ListViews

Es el scrolling widget más usado. En muchas de las aplicaciones que usamos a diario nos encontramos con listas de elementos más comúnmente vertical aunque también nos podemos encontrar alguna horizontal.

Este widget contiene un listado de hijos que se dispondrán uno debajo de otro (vertical) o uno al lado de otro (horizontal).

ListViews

Toda la información la podemos encontrar en la documentación oficial de flutter:
https://api.flutter.dev/flutter/widgets/ListView-class.html

ListViews

Este widget se puede usar de cuatro diferentes maneras:

- Static ListView
- Dynamic ListView
- ListView separated
- ListView custom

Static ListView

Usamos ListView estáticos cuando tenemos un conjunto de valores pequeños y que no cambian.

```
ListView(
    children: <Widget>[
                    ListTile(title: Text('Sun'),),
                    ListTile(title: Text(Mercury),),
                    ListTile(title: Text(Venus),),
                    ],
            );
```

Static ListView

En el caso de que quisiéramos que los elementos de la static ListView tuvieran un separador tendríamos que usar: **ListTile.divideTiles**

```
ListView(
    children: ListTile.divideTiles(
        context: context,
        tiles: [ListTile(title: Text('Sun'),),
                ListTile(title: Text(Mercury),),
                ListTile(title: Text(Venus),)]).toList(),
    )
```

Dynamic ListView

Cuando la lista es demasiado grande como para crearla de forma estática o bien los valores son creado en tiempo de ejecución tendremos que usar ‘dynamic ListView’.

Los elementos en las dynamic ListView solo se crean cuando necesitan ser mostrados.

Tenemos que usar el constructor ListView.builder() para crear ListView dinámicamente.

Dynamic ListView

```
List jedis = [‘Obi-Wan’, ‘Yoda’, ‘Mace Windu’, ‘Anakin’,
‘Luke’, ‘Rey’, ‘Ben Solo’];
ListView.builder(
    itemCount: jedis.length,
    itemBuilder: (context, index) {
                 return ListTile(title: Text(jedis[index]),);
                 },);
```

ListView Separated

Este tipo de ListView tiene un comportamiento dinámico como el anterior y solo se diferencia de él en que los elementos están separados por un separador.

En este caso se usa el constructor ListView.separated

ListView Separated

```
ListView.separated(
          itemCount: 500,
          itemBuilder: (context, index) {
            return ListTile(
              title: Text('Element $index'),
            );
          },
          separatorBuilder: (context, index) {
            return Divider();
          },
        );
```

ListView custom

Cuando aplicando estilos y cambios a los elementos de una ListView no somos capaces de obtener el resultado visual que queremos tenemos que crear una ListView custom haciendo uso ListView.custom