Università
degli Studi
di Padova

FLUTTER

Mobile Programming and Multimedia

Final examination

M.S. in Computer Science

Andrea Favero ID 291825497

Flutter

- Open source framework by Google
- Cross compiled
- Applications written in DART
- Difficult to distinguish between a native and a Flutter App
- Very good compiler

Dart

Flutter

- Unique codebase → Android, iOS, web, embedded deployment
- Last version: 3.10.2 released on 24/05/2023
  - better web/mobile integration, better graphics performances, ...
- Developed and used by big companies around the world

Performances

- Very similar to **native** applications performances
- OS features (icons, scrolling, fonts, etc…) are all incorporated inside widgets

User interface

- **Native look and feel** according to the platform
- Expressive and flexible:
  - low level frameworks: create your **personal UI**

Fast development

- **reuse** widgets and library provided by Google & community
- **Hot reload**:
  - no need to rebuild from scratch after small modifications (e.g. moving a button)
  - state “preservations” for widgets
  - useful to test just some parts of a complex interaction (e.g.: buying steps in an ecommerce app)
- **Code forking**: allowed to gain data from sensors or use a specific API
- webapp deployment useful for quick tests

Pros

- Native performances
- Single codebase
- Hot reload
- Accessibility (big fonts, contrasts, screen reader)
- Documentation
- Community
- FOSS
- Firebase

Cons

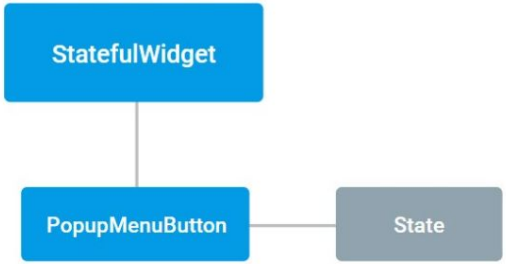
- Dart knowledge is required
- Animations

Flutter's architecture

- **Engine**: runtime environment which runs the application
- **Framework**: divided in levels
- **Dart:ui** allow communications with the flutter engine (e.g.: render the shape of a button and its behavior)

Widgets

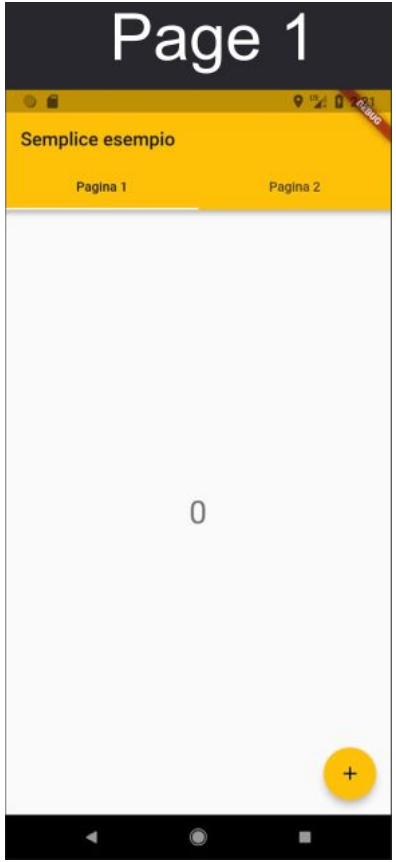
- **Everything is a widget**: button, fonts, padding
- **Hierarchical** organization (in a **tree**)
- **build()** method
- They can be **stateful** and **stateless**

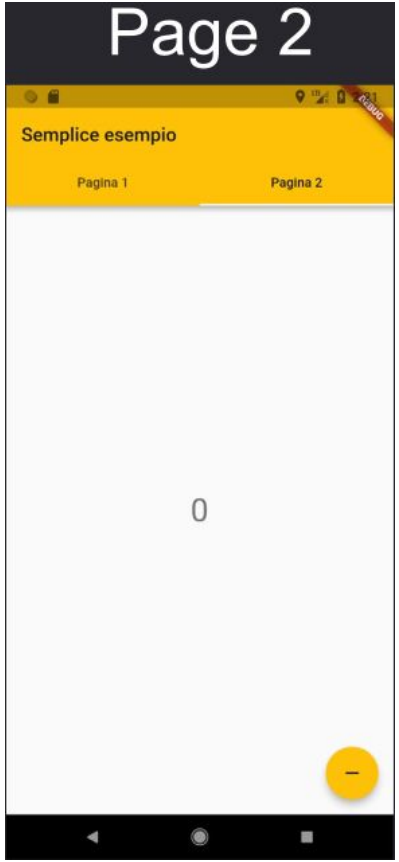


Dart

- OOP by Google
- Every dart app is a library
- Lazy library loading
- variables like Java: store references, null as default
- Compilation:
  - Ahead Of Time (AOT)
    - compile a component only when it is actually needed at runtime (like Hot reload)
  - Just In Time (JIT)
    - used for deploy. Makes cross compilation possible

Example






```
class MyApp extends StatelessWidget {...}
class FirstPage extends StatefulWidget {...}
class SecondPage extends StatefulWidget {...}
class _FirstPageState extends State<FirstPage> {...}
class _SecondPageState extends State<SecondPage> {...}
```

Example - Application

```
class MyApp extends StatelessWidget {
 // This widget is the root of your
application.
 @override
 Widget build(BuildContext context) {
   return MaterialApp(
     title: 'Flutter Demo',
     theme: ThemeData(
       primarySwatch: Colors.amber;
     ),
     home: DefaultTabController(
         length: 2,
```

```
child: Scaffold(
    appBar: AppBar(
        bottom: TabBar(
            tabs: [ Tab(text: ”Page 1"),
                    Tab(text: ”Page 2")]
        ),
        title: Text(”Simple example"),
    ),
    body: TabBarView(
        children: [
            FirstPage(title: ”First page"),
            SecondPage(title: ”Second page")
        ]),)),);}}
```

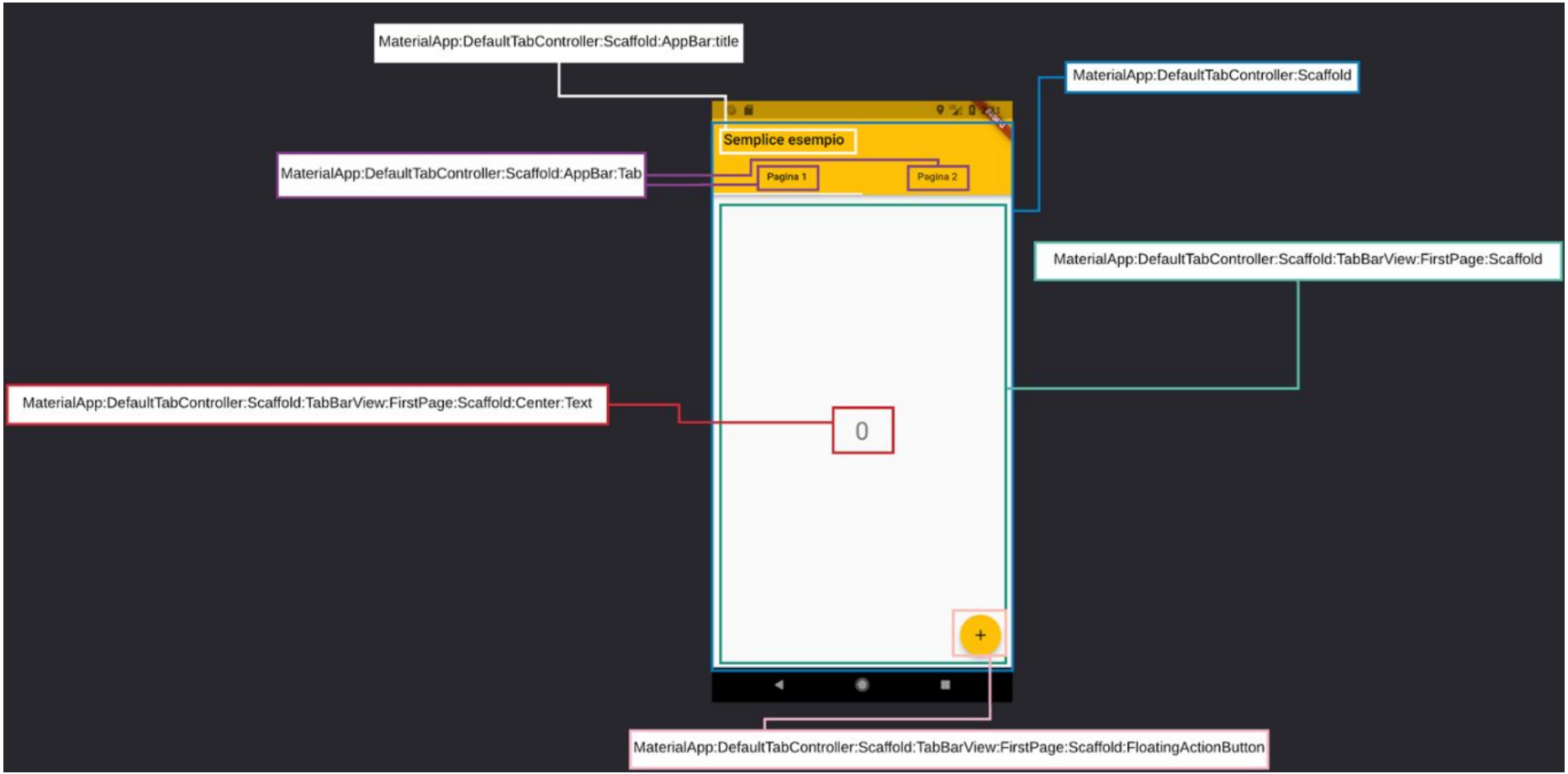
Example - State of first page

```
class _FirstPageState extends State<FirstPage> {
 int _counter1 = 0;
 void _incrementCounter() {
   setState(() {
     _counter1++;
   });
 }
```

```
@override
 Widget build(BuildContext context) {
   return Scaffold(
     body: Center(
       child: Text(
             '$_counter1',
       ),
     ),
```

```
floatingActionButton: FloatingActionButton(
        onPressed: _incrementCounter,
        tooltip: 'Increment',
        child: Icon(Icons.add),
      ),
    );
  }
}
```

Example - Interface schema



References

- https://medium.com/flutter/whats-new-in-flutter-3-10-b21db2c38c73
- https://repository.tudelft.nl/islandora/object/uuid:8708c82f-1100-4607-ba28-eaebae8b03ce/datastream/OBJ/download
- https://dev.to/sudarasach/intro-to-flutter-2odk
- Slides from the lectures