

ISW-011

Desenv. para Dispositivos Móveis 1 (DDM1)

DDM-1 – ISW-011
Prof. Davi dos Reis

Aula 07 – Flutter – componentes básicos

Componentes básicos

Flutter – TextField

- No Flutter, para criarmos um campo onde o usuário digita uma informação utilizamos o widget **TextField**.

Prof. Davi

```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      theme: ThemeData(primaryColor: Colors.blue),
      home: Scaffold(
        appBar: _titulo(),
        body: _body(),
      ),
    );
  }

  _titulo() {
    return AppBar(
      title: Text("Flutter - TextField"),
      centerTitle: true,
    );
  }
}
```

Flutter – TextField

- No Flutter, para criarmos um campo onde o usuário digita uma informação utilizamos o widget **TextField**.

Prof. Davi

```
_body() {
  return Container(
    width: double.infinity,
    color: Colors.white10,
    child: Column(
      mainAxisAlignment: MainAxisAlignment.max,
      mainAxisSize: MainAxisSize.max,
      crossAxisAlignment: CrossAxisAlignment.center,
      children: <Widget>[
        _campo(),
        _botao(),
      ],
    ),
  );
}

_campo() {
  return TextField(
    keyboardType: TextInputType.text,
    decoration: InputDecoration(
      labelText: "Digite o seu nome",
      labelStyle: TextStyle(color: Colors.green)),
    textAlign: TextAlign.center,
    style: TextStyle(color: Colors.green, fontSize: 25.0),
  );
}
```

Flutter – TextField

- No Flutter, para criarmos um campo onde o usuário digita uma informação utilizamos o widget **TextField**.

```
_botao() {
  return ElevatedButton(
    style: ElevatedButton.styleFrom(backgroundColor: Colors.green),
    onPressed: metodoClicar,
    child: Text(
      "OK",
      style: TextStyle(
        color: Colors.white,
        fontSize: 20,
      ),
    ),
  );
}

metodoClicar() {
  print("Clicou aqui");
}
```

Flutter – TextField

- O exemplo anterior exibe um valor fixo no Console.
- Para conseguir receber o valor do usuário e exibi-lo (ainda no Console), é necessário realizar os 3 passos abaixo:
 - Adicionar “final TextEditingController _controller = TextEditingController();” abaixo do construtor;
 - Configurar esse novo elemento na propriedade “controller” do método “_campo()”;
 - Alterar o método “metodoClicar()” para ficar com as duas linhas abaixo:


```
String texto = _controller.text;
print("Texto inserido: $texto");
```

Flutter – StatefulWidget e setState

- Quando trabalhamos com StatefulWidget e States, significa que precisamos guardar as informações que manipulamos em nosso App.
- Ao criar um StatefulWidget são criadas 2 classes:
 - A primeira classe herda de StatefulWidget
 - A segunda classe herda de State<Sua Primeira Classe>
- Ex.:

Prof. Davi

ISW-011 (DDM-1)

8

Flutter – StatefulWidget

```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return TelaInicial();
  }
}
```

```
// Primeira Classe
class TelaInicial extends StatefulWidget {
  const TelaInicial({Key? key}) : super(key: key);

  @override
  _TelaInicialState createState() => _TelaInicialState();
}
```

```
// Segunda Classe
class _TelaInicialState extends State<TelaInicial> {
  @override
  Widget build(BuildContext context) {
    return Container();
  }
}
```

Prof. Davi

ISW-011 (DDM-1)

9

Flutter – Stateful

- Na primeira classe temos apenas um método **createState** que chama outro objeto que herda de classe State de forma tipada. Nessa primeira classe não vamos desenvolver nada.
- Na segunda classe é onde vamos desenvolver nosso código. Por essa classe herdar de State, ela tem a capacidade de armazenar informações durante a execução do App – além de atualizar a tela do App durante sua execução.

```
// Primeira Classe
class TelaInicial extends StatefulWidget {
  const TelaInicial({Key? key}) : super(key: key);

  @override
  _TelaInicialState createState() => _TelaInicialState();
}

// Segunda Classe
class _TelaInicialState extends State<TelaInicial> {
  @override
  Widget build(BuildContext context) {
    return Container();
  }
}
```

Flutter – Stateful

- Vamos agora ver um exemplo de utilização de State.

```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return TelaInicial();
  }
}

class TelaInicial extends StatefulWidget {
  const TelaInicial({Key? key}) : super(key: key);

  @override
  _TelaInicialState createState() => _TelaInicialState();
}
```

Flutter – Stateful

- Vamos agora ver um exemplo de utilização de State.

Prof. Davi

```
class _TelaInicialState extends State<TelaInicial> {
  TextEditingController nomeController = TextEditingController();

  String exibicaoNome = "Seu nome será exibido aqui";

  void _exibeNome() {
    setState(() {
      String nome = nomeController.text;

      exibicaoNome = "Seu nome é: $nome";
    });
  }

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      theme: ThemeData(primaryColor: Colors.blue),
      home: Scaffold(
        appBar: AppBar(
          title: Text("Flutter - Stateful"),
          centerTitle: true,
        ),
        body: _body(),
      ),
    );
  }
}
```

Flutter – Stateful

- Vamos agora ver um exemplo de utilização de State.

Prof. Davi

```
_body() {
  return Container(
    width: double.infinity,
    color: Colors.white10,
    child: Column(
      mainAxisAlignment: MainAxisAlignment.max,
      mainAxisSize: MainAxisSize.max,
      crossAxisAlignment: CrossAxisAlignment.center,
      children: <Widget>[
        _campo(),
        _button(),
        _texto(),
      ],
    ),
  );
}

_campo() {
  return TextField(
    keyboardType: TextInputType.text,
    decoration: InputDecoration(
      labelText: "Digite o seu nome",
      labelStyle: TextStyle(color: Colors.green)),
    textAlign: TextAlign.center,
    style: TextStyle(color: Colors.green, fontSize: 25.0),
    controller: nomeController,
  );
}
```

Flutter – Stateful

- Vamos agora ver um exemplo de utilização de State.

```
_texto() {
  return Text(exibicaoNome);
}

_button() {
  return ElevatedButton(
    style: ElevatedButton.styleFrom(backgroundColor: Colors.green),
    onPressed: _exibeNome,
    child: Text(
      "OK",
      style: TextStyle(
        color: Colors.white,
        fontSize: 20,
      ),
    ),
  );
}
```

Flutter – Stateful

- Perceba que neste exemplo temos:
 1. O que o usuário digitou foi capturado através do objeto **nomeController**, que foi atribuído no **controller** do **TextField**; Declaramos e instanciamos o objeto **nomeController** no início da nossa classe;
 2. O **botão**, ao ser clicado, está chamando o método **exibeNome**.
 3. Criamos uma variável chamada **exibicaoNome** que está sendo exibida no método **_texto()**.
 4. O método **exibeNome** possui em seu corpo o método **setState**, que está capturando o conteúdo do controller **nomeController** e o colocando na variável **nome**, que por sua vez está sendo utilizada na variável **exibicaoNome**.
 5. Quando o método **setState** é chamado, todo o conteúdo do método **build** é atualizado (renderizado novamente em tempo de execução); com isso a variável **exibicaoNome** tem seu valor atualizado e é mostrado logo abaixo do botão no App.

Obrigado!