DCC917A – TÓPICOS ESPECIAIS III: DESENVOLVIMENTO DE APLICATIVOS MÓVEIS

AULA 11

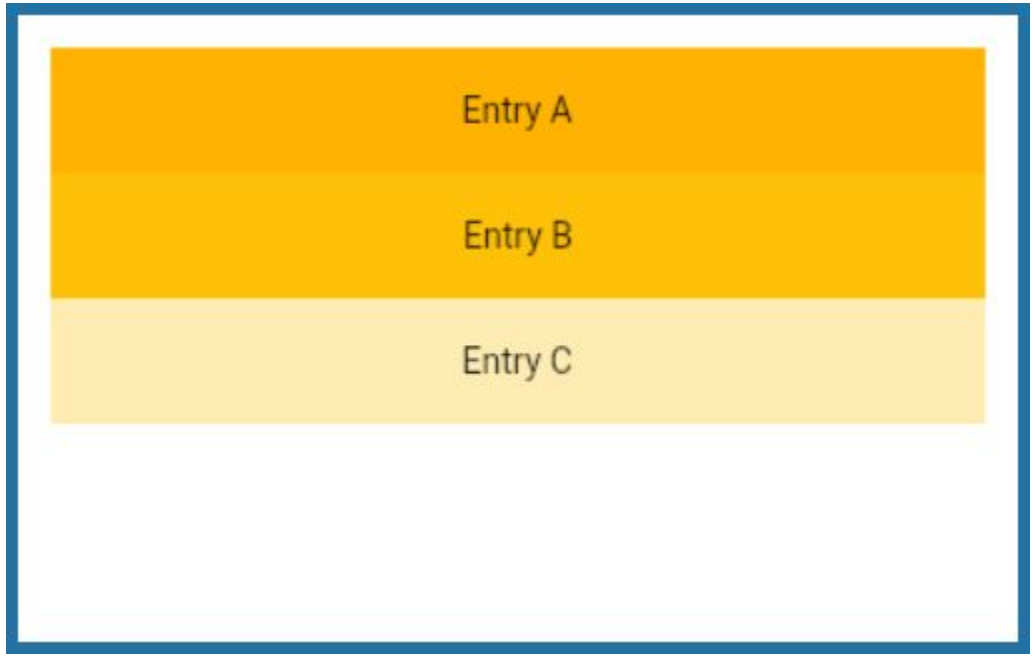
Carlos Bruno Oliveira Lopes

Engenheiro de Computação

Mestre em Ciência da Computação

Flutter (Listas - ListView)

- ListView é a widget de rolagem que exibe seus filhos um após o outro na direção de rolagem.

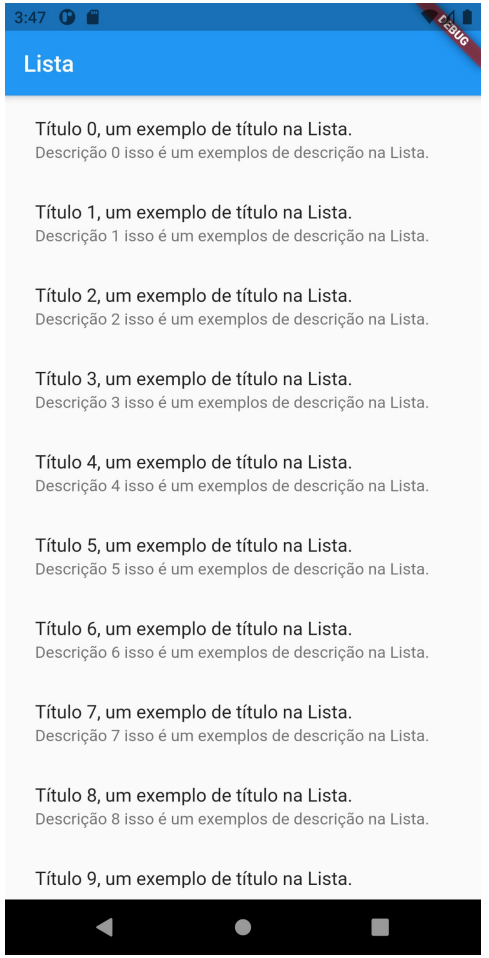



```
final List<String> entries = <String>['A', 'B', 'C'];
final List<int> colorCodes = <int>[600, 500, 100];

ListView.builder(
  padding: const EdgeInsets.all(8),
  itemCount: entries.length,
  itemBuilder: (BuildContext context, int index) {
    return Container(
      height: 50,
      color: Colors.amber[colorCodes[index]],
      child: Center(child: Text('Entry ${entries[index]}')),
    );
  }
);
```

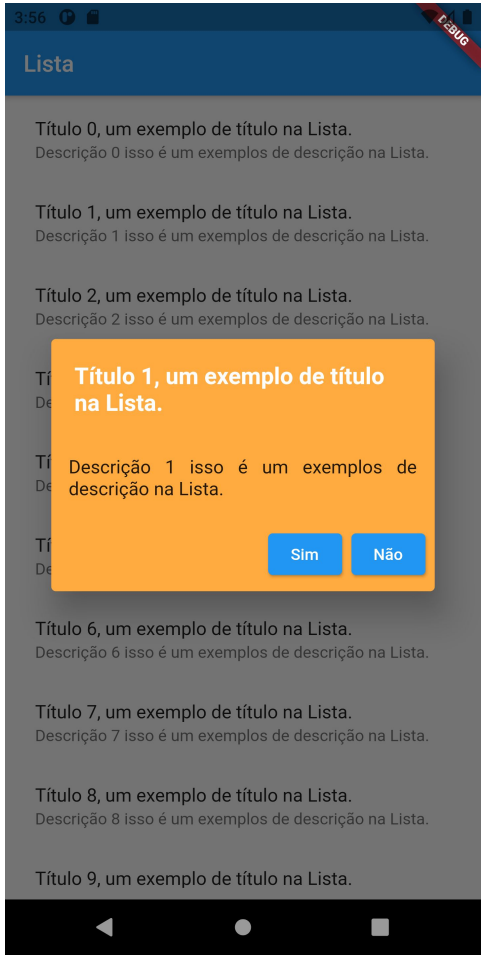
Flutter (Listas - ListView)

```
child: ListView.builder(
  itemCount: _itens!.length,
  itemBuilder: (context, index) {
    return ListTile(
      title: Text(_itens![index]['titulo']),
      subtitle: Text(_itens![index]['descricao']),
      onTap: (){
        print("clique com OnTap $index");
      },
      onLongPress: (){
        print("clique com onLongPress $index");
      },
    );
  },
),
```



Flutter (AlertDialog)

```
builder: (context) => AlertDialog(
  title: Text(_itens![index]['titulo']),
  titlePadding: EdgeInsets.all(20),
  titleTextStyle: TextStyle(fontSize: 20, fontWeight: FontWeight.bold,color: Colors.White),
  content: Text(_itens![index]['descricao'], textAlign: TextAlign.justify),
  contentPadding: EdgeInsets.all(15),
  contentTextStyle: TextStyle(fontSize: 16, color: Colors.black87),
  backgroundColor: Colors.orangeAccent,
  actions: [
    ElevatedButton(
      onPressed: (){
        print("Foi slecionado sim");
        Navigator.pop(context); //fecha a tela
      },
      child: Text("Sim"),
    ),
    ElevatedButton(
      onPressed: (){print("Foi selecionado não");},
      child: Text("Não"),
    ),
  ],
),
```



Flutter (Requisições – Classe Future)

Future

- Classe que representa um objeto que será processado com atraso (*delayed computation*)
- Ela é usada para representar um valor em potencial, ou erro, que estará disponível em algum tempo no futuro.
- Em gera, ele é usado para operações assíncronas, tais como:
  - Busca de dados sobre a rede;
  - Escrita em banco de dados;
  - Leitura de dados em arquivos;
- Ela usada com palavras chaves `async` e `await`.

Flutter (Requisições – Classe Future)

Future

- Por meio da classe podemos fazer um controle mais robusto durantes as requisições que são feitos na rede, como controle de status:
  - **`ConnectionState.none`**: Indica que atualmente não há nenhuma computação assíncrona conectada.
  - **`ConnectionState.waiting`**: Indica que esta conectado há uma computação assíncrona e que esta esperando pela interação.
  - **`ConnectionState.active`**: Indica que esta conectado há uma computação assíncrona (Conexão em Stream).
  - **`ConnectionState.done`**: Indica que esta conectado há uma computação assíncrona encerrada.

Flutter (Requisições – Classe Future)

```
Future<Map> _recuperarPreco() async {
  var url = Uri.parse("https://blockchain.info/ticker");
  http.Response response = await http.get(url);
  return json.decode(response.body);
}

Widget build(BuildContext context) {
  return FutureBuilder<Map>(
    future: _recuperarPreco(),
    builder: (context, snapshot) {
      String resultado = "";
      switch(snapshot.connectionState){
        case ConnectionState.none:
          print ("conexão none");
          break;
```

Flutter (Requisições – Classe Future)

```
      case ConnectionState.waiting:
        print ("conexão waiting");
        resultado = "Carregando...";
        // return CircularProgressIndicator();
        break;
      case ConnectionState.active:
        print ("conexão active");
        break;
      case ConnectionState.done:
        print ("conexão done");
        if(snapshot.hasError){
          resultado = "Erro ao carregar os dados";
        } else {
          double valor = snapshot.data!['BRL']['buy'];
          resultado = "Preço do bitcoin: ${valor.toString()}";
        }
        break;
    }
    return Center(
      child: Text(resultado),
    );
  },
```

Flutter (Requisições – Classe Future)

```
Future<CotacaoBitcoin> _criaCotacaoBitcoin() async {
    var url = Uri.parse("https://blockchain.info/ticker");
    http.Response response = await http.get(url);
    Map<String, dynamic> cotacao = json.decode(response.body);

    print("Código de Status: " + response.statusCode.toString());
    print("Cotação: ${cotacao['BRL']['buy']}");

    return CotacaoBitcoin(
      buy: cotacao[CotacaoBitcoin.BRL][CotacaoBitcoin.BUY],
      sell: cotacao[CotacaoBitcoin.BRL][CotacaoBitcoin.SELL],
      last: cotacao[CotacaoBitcoin.BRL][CotacaoBitcoin.LAST],
      symbol: cotacao[CotacaoBitcoin.BRL][CotacaoBitcoin.SYMBOL],
    );
  }
```

Flutter (Requisições – Classe Future)

```
FutureBuilder<CotacaoBitcoin> buildFutureBuider() {
  return FutureBuilder(
    future: _cotacao,
    builder: (context, snapshot) {
      switch (snapshot.connectionState) {
        case ConnectionState.waiting:
          print("conexão waiting");
          return CircularProgressIndicator(color: Color.fromRGBO(247, 147, 26, 1),
          );
        case ConnectionState.done:
          print("conexão done");
          if (snapshot.hasError) {
            return Text("${snapshot.error}");
          } else {
            double buy = snapshot.data!.buy;
            String symbol = snapshot.data!.symbol;
            return Text("\$$symbol ${buy.toString()}");
          }
      }
      return Text("");
    },
  );
}
```

Flutter (Exercícios)

1. Adicionando funcionalidade na App ATM Consultoria:
   - Adicione um Check button com as opções de Real, Euro e Dólar;
     - O botão Check estará configurado como padrão no Real, no entanto, o usuário poderá mudar a seleção da cotação para Euro ou dólar, ou quaisquer combinação dos três.
     - Quando a requisição for feita a cotação a ser exibida deve esta de acordo com a opção que o usuário selecionou.
     - Se nenhuma seleção estiver habilitada deve-se exibir por padrão a cotação do bitcoin em real.