FLUTTER
CHECKBOX E RADIOBUTTON

PROF. ROGÉRIO B. DE ANDRADE

FLUTTER - CHECKBOX E RADIOBUTTON

CHECKBOX.

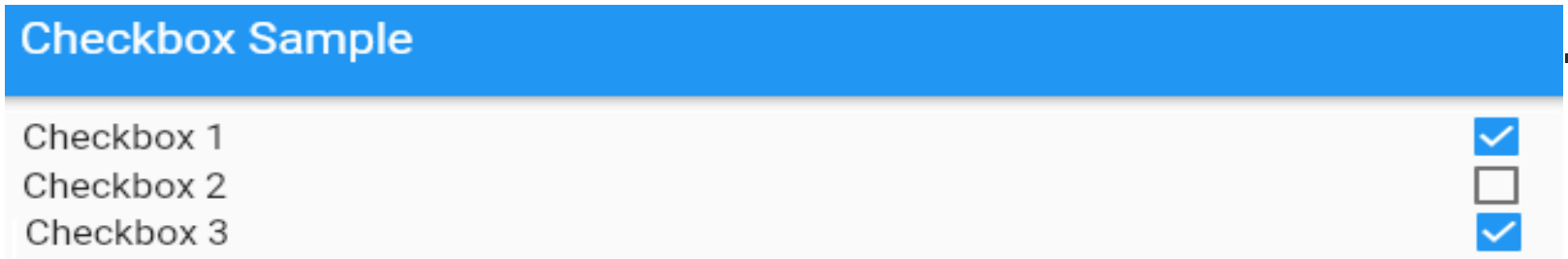
RADIOBUTTON

- CheckBox é utilizado em formulários para promover a escolha múltipla de itens de uma lista.
- RadioButton é utilizado em formulários para promover a escolha exclusiva (única) de um item da lista

CHECKBOX

```dart
class CheckboxSample extends StatefulWidget {
  @override
  CheckboxSampleState createState() => CheckboxSampleState();
}

class CheckboxSampleState extends State<CheckboxSample> {
  Map<String, bool> checked = {
    'Checkbox 1': false,
    'Checkbox 2': false,
    'Checkbox 3': false,
  };
```



- O código ao lado implementa a classe CheckboxSampleState na qual está sendo declarada uma coleção denominda checked que contém em sua estrutura a relação chave valor, inicializada com valores pré-definidos que irão compor a lista de opções do checkBox a ser renderizada na tela.
- O resultado da renderização pode visualizado ao lado.

CHECKBOX

```dart
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: const Text('Checkbox Sample'),
    ),
    body: Container(
      padding: const EdgeInsets.all(16.0),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: _buildCheckboxes(),
      ),
    ),
  );
}
```

Build para renderização da tela com os checkboxes

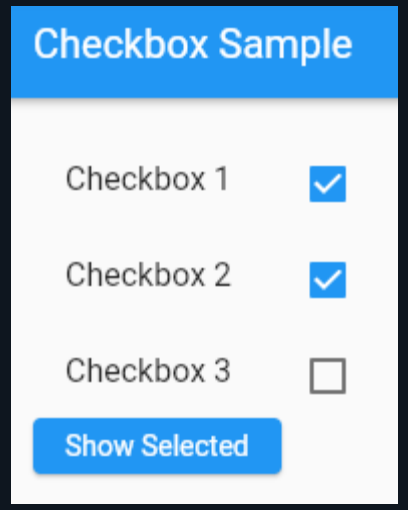
Será renderizada uma lista de checkBoxes

- Os códigos apresentados implementam a renderização do conteúdo do checkbox a partir da coleção Checked, adicionando as informações armazenadas na coleção e implementando o gerenciamento de evento onChange, que reage a interação do usuário, promovendo a alternância de valores de cada checkbox, entre verdadeiro e falso.

```dart
List<Widget> _buildCheckboxes() {
  List<Widget> checkboxes = [];
  // percorre todos os itens armazendos na estrutura
  // Map (chave, valor), obtendo os valores
  checked.forEach((String key, bool value) {
    checkboxes.add(// adiciona um item ao checkbox
      CheckboxListTile(
        title: Text(key),//adiciona título
        value: value,// define valor
        // implementa evento onChage para gerenciar
        // valor do checkbox
        onChanged: ( newValue) {//recebe valor atual
          setState(() {//altera o estado da variável
            newValue!=null? //se diferente de null
              //alterna valor entre verdadeiro e falso
              checked[key]==true? checked[key]=false:
              checked[key] = true
            : checked[key] = false;
          });
        },
      ),
```

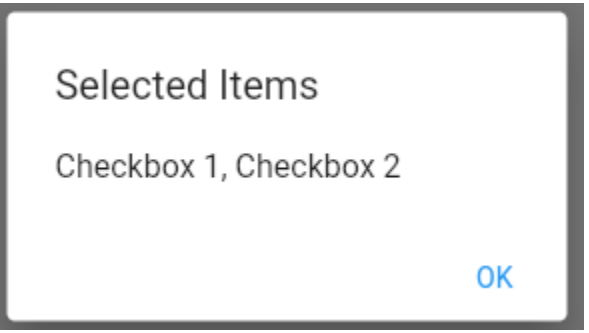
CHECKBOX

```dart
// adiciona ao final do checkBox, botão
checkboxes.add(
  ElevatedButton(
    // se pressionado executar método
    onPressed: () {
      _showSelected();// mostra selecionados
    },
    child: const Text('Show Selected'),
  ),
);
```



Quando o botão "show selected" o método showSelected será executado é irá mostrar cada seleção realizada pelo usuário.

```dart
void _showSelected() {// mostra itens selecionados
  List<String> selected = [];// lista de strings
  // percorre coleção que contém o itens selecionados
  checked.forEach((String key, bool value) {
    if (value) { // se valor válido
      selected.add(key);// adiciona na lista de strings
    }
  });
```



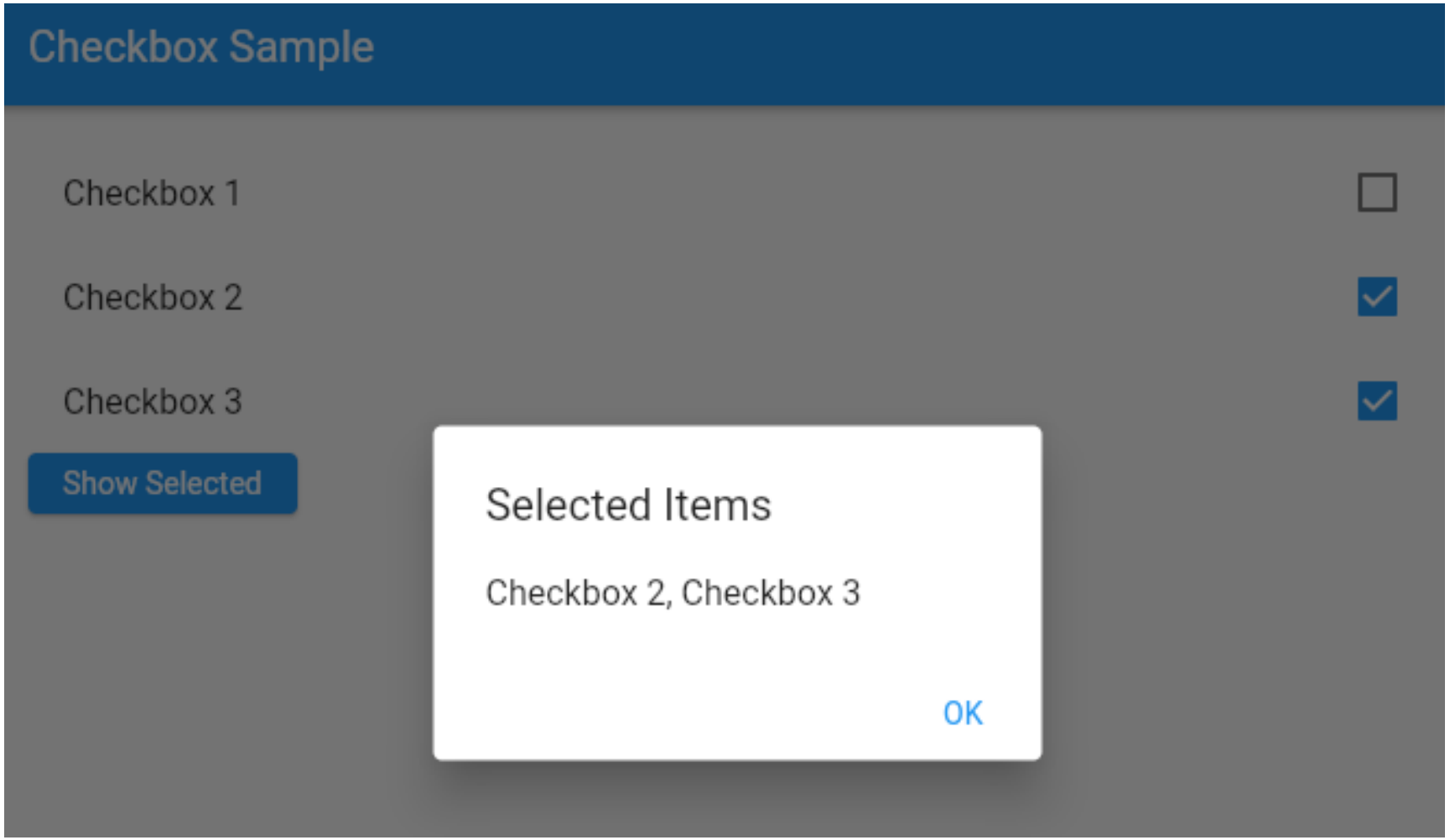
- Resultado da escolha do usuário. Cada escolha é adicionada a lista selected que posteriormente será visualizada em uma caixa de diálogo.

CHECKBOX

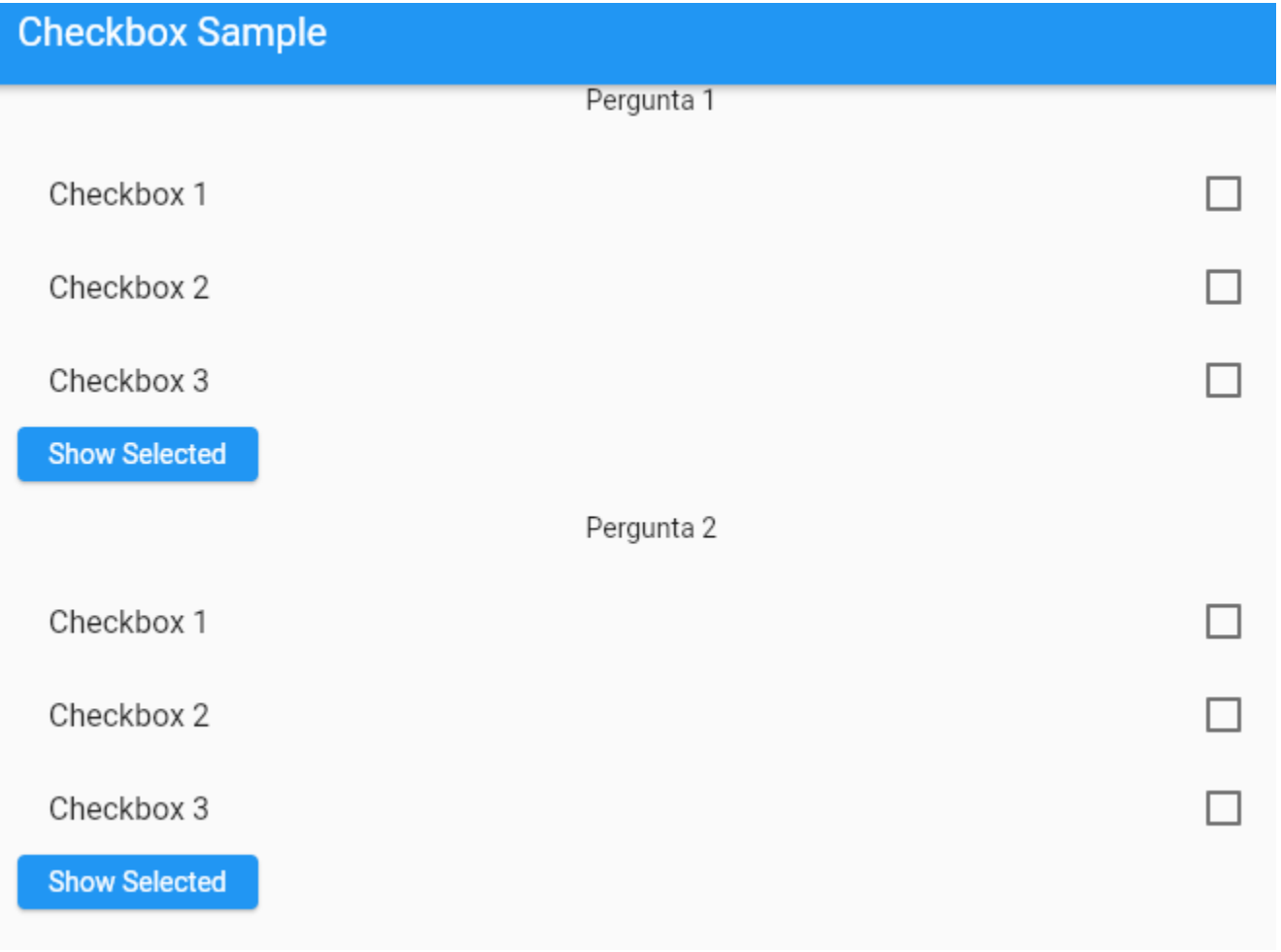
```
// mostra caixa de diálogo com as opções selecionadas
showDialog(
  context: context,
  builder: (BuildContext context) {
    return AlertDialog(
      // título da lista
      title: const Text('Selected Items'),
      // conteúdo da lista
      //* a cada elemento adicina uma vírgula
      content: Text(selected.join(', ')),
      actions: [ // cria botão na caixa de diálogo
        TextButton(
          onPressed: () {//fecha caixa de diálogo
            Navigator.of(context).pop();
          },
          child: const Text('OK'),
        ),
      ],
    );
  },
);
```

- Implementação da caixa de diálogo que mostra as informações selecionadas pelo usuário.
- O método builder, configura as propriedades da caixa de diálogo e adiciona um botão, além de definir o conteúdo da caixa (contentes) com o conteúdo selecionado, separado por vírgula (Join (‘. ‘).

CHECKBOX


Checkbox Sample
Checkbox 1
Checkbox 2
Checkbox 3
Show Selected
Selected Items
Checkbox 2, Checkbox 3
OK

CHECKBOX



Atividade: Altere o código exemplo para que seja possível implementar o layout ao lado. Escolha uma temática e crie um questionário, no qual sejam apresentadas duas perguntas, com resposta de múltipla escolha (independentes).

Código (base) disponível em:

https://docs.google.com/document/d/1OEJA9dqnUIWZN3efW5RcQj-Kn7PO959wIhrNxoXcuDc/edit?usp=sharing

RADIOBUTTON

RADIOBUTTON

- RadioButton é um componente que permite a escolha exclusiva de uma lista.

RADIOBUTTON

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
        ...
         home:  RadioButtonSample(),
      );
    }
  }
```

- Classe principal que instância a classe RadioButtonSample

RADIOBUTTON

```
class RadioButtonSample extends StatefulWidget {
  @override
  RadioButtonSampleState createState() => RadioButtonSampleState();
}
```

- Classe que instancia classe de state para o radio button, onde será realizado o método build para inserir efetivamente o radioButton.

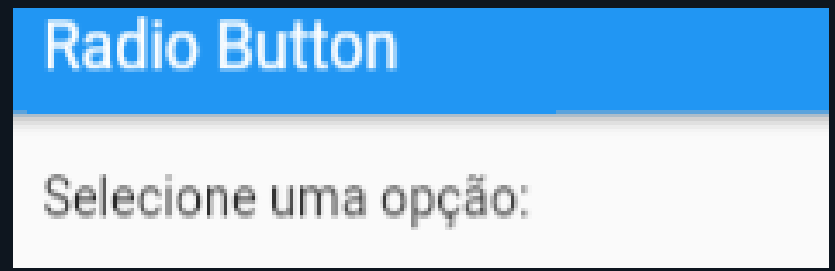
RADIOBUTTON

```
class RadioButtonSampleState extends State<RadioButtonSample> {
  String _selectedOption = ""; // armazena opção selecionada
  @override
  Widget build(BuildContext context) {
    //build da classe
    return Scaffold(
      ...
      body: Container(
        ...
        children: _buildRadioButtons(),// adiciona radio
```

- Build instanciando _buildRadioButton()

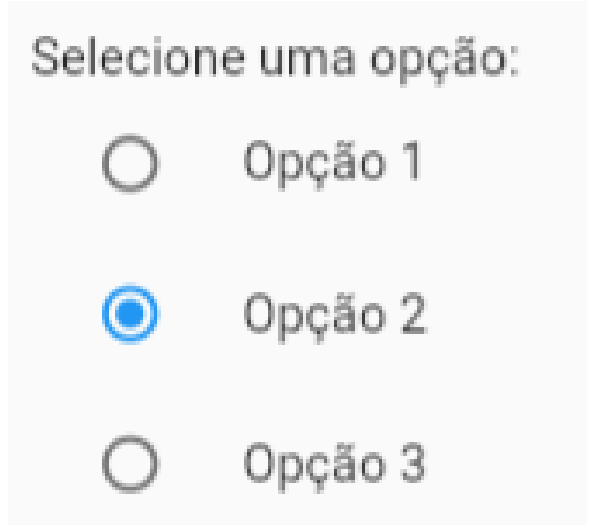
RADIOBUTTON

```
// cria lista de opções
List<Widget> _buildRadioButtons() {
  List<Widget> radioButtons = []; //lista vazia
  // texto das opções da lista
  List<String> options = ['Opção 1', 'Opção 2', 'Opção 3'];
  radioButtons.add(
    // adiciona radio e configura
    const Text(
      'Selecione uma opção:',
      style: TextStyle(fontSize: 16.0),
    ),
  );
```



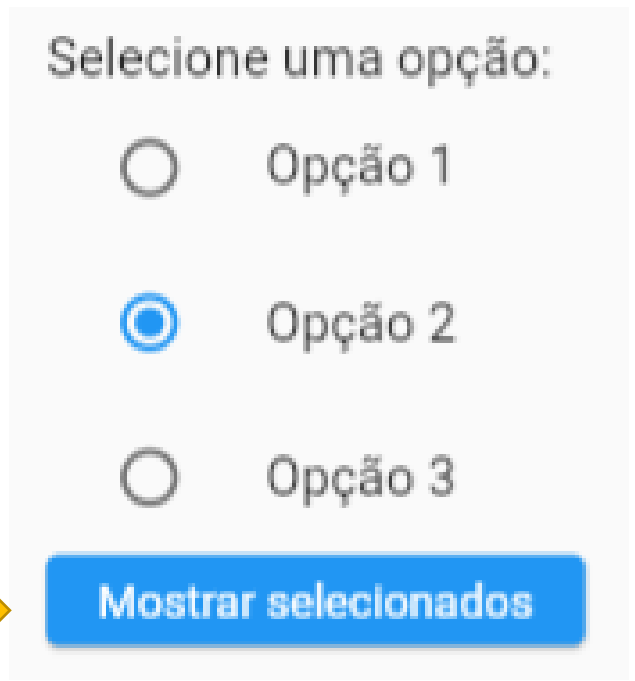
RADIOBUTTON

```
// percorre lista e adicona ao radio
for (var option in options){
  // adiciona texto e radio
  radioButtons.add(
    ListTile(
      title: Text(option),//oopção
      leading: Radio(// radio
        value: option,//valor associado quando selecionado
        groupValue: _selectedOption,//armazena selecionado
        onChanged: (newValue) {// evento mudança, ocorre
          setState(() {// atualiza state
            newValue != null// é válido
                ? _selectedOption = newValue// armazena selecionado
                : _selectedOption = "";
          });
        },
      ),
    ));
}
```



RADIOBUTTON

```
//adiciona  botão para mostrar selecionados
radioButtons.add(
  ElevatedButton(
    onPressed: () {
      _showSelected();// mostra selecionados
    },
    child: const Text('Mostrar selecionados'),
  ),
);
```



RADIOBUTTON

```dart
// janela de diálogo para mostrar selecionados
void _showSelected() {
  showDialog(
    context: context,
    builder: (BuildContext context) {
      return AlertDialog(//janela
        title: const Text('Opção selecionada'),
        content: Text(_selectedOption),
        actions: [
          TextButton(// botão do diálogo
            onPressed: () {
              Navigator.of(context).pop();
            },
            child: const Text('OK'),
          ),
        ],
      );
    },
  );
}
```