



Flutter - Aula 2

Exercícios 6 a 8



- Flutter Gallery; [Ver mais](#)
- Flutter Fire; [Ver mais](#)
- Flutter Browser; [PlayStore](#) [Github](#)



- Utilize para efetuar validações de vários campos
- Validators

```
final formKey = GlobalKey<FormState>();  
  
...  
  
Form(  
    key: formKey,  
    child:  
)
```

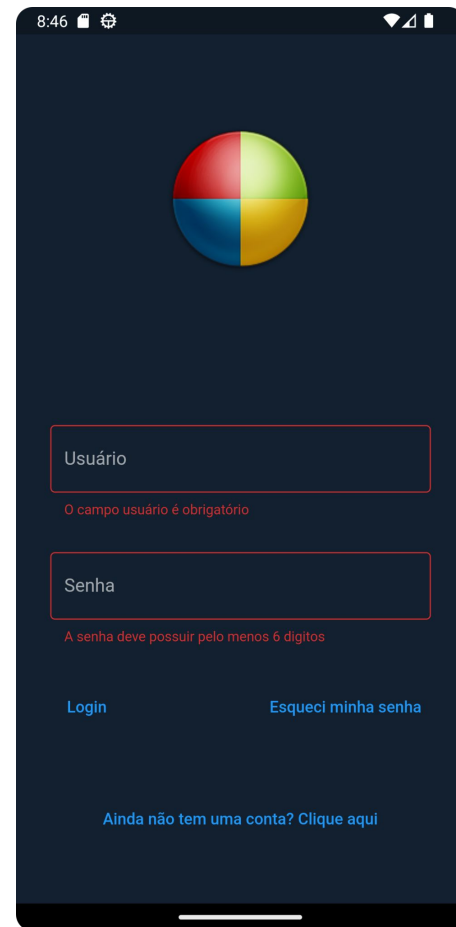
```
final formKey = GlobalKey<FormState>();  
  
...  
  
Form(  
    key: formKey,  
    child:  
)
```



Vamos criar uma página de login, com um campo de login, um campo de senha e um botão de login.

Regras:

- O campo de Login é obrigatório, se não estiver preenchido, você deve avisar o usuário;
- O campo senha precisa ter no mínimo 6 caracteres;
- O campo senha pode ter no máximo 12 caracteres;





- Uma lista de itens com scroll
- A lista pode ser fixa ou dinâmica
- Semelhante a Column e Row
- Você poderá controlar o sentido de exibição dos itens

```
return Scaffold(  
  appBar: AppBar(title: Text("Lista estática")),  
  body: ListView(  
    children: [  
      Text("Item 1"),  
      Text("Item 2"),  
      Text("Item 3"),  
      Text("Item 4"),  
      Text("Item 5"),  
    ],  
  ), // ListView  
); // Scaffold
```

```
appBar: AppBar(title: Text("Lista dinâmica")),  
body: ListView.builder(  
  itemCount: vrItemList.length,  
  itemBuilder: (BuildContext context, int index) {  
  
    return Row(  
      children: [  
        Text((index + 1).toString()),  
        SizedBox(width: 30,),  
        Text(vrItemList[index])  
      ],  
    ); // Row  
  
  }); // ListView.builder, Scaffold
```



- Um componente padrão para exibir itens
- Além de exibir os itens, eles podem ser clicáveis

One-line ListTile



One-line with leading widget

One-line with trailing widget



One-line with both widgets



One-line dense ListTile



Two-line ListTile
Here is a second line

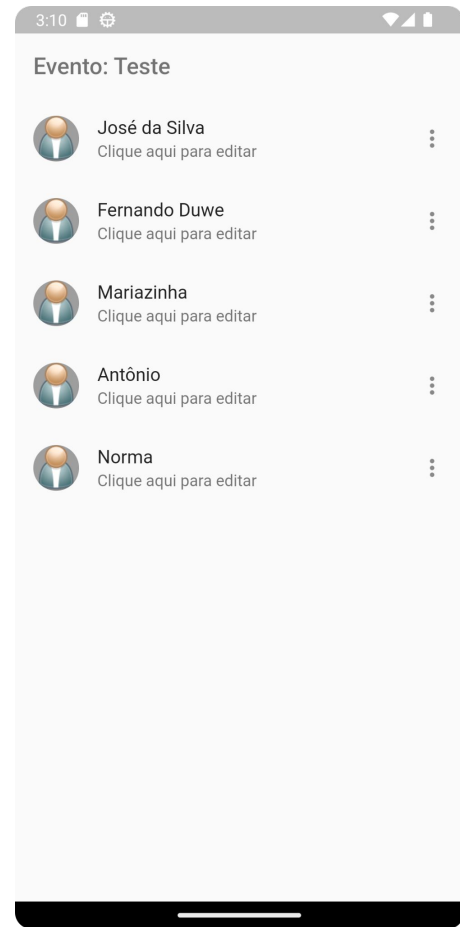


Three-line ListTile
A sufficiently long subtitle
warrants three lines.



Vamos criar uma tela que liste os convidados de uma festa.

- A lista de convidados deve ser lida a partir de uma lista;
- Ao clicar em cada item, escreva no console (print), qual foi o convidado;
- O layout da tela fica a seu critério, divirta-se!





- ListTile com opção de expansão, para exibir itens filhos;

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: Text("Expansion Tile")),
    body: ExpansionTile(
      title: Text("Meu teste"),
      subtitle: Text("Clique aqui para ver algumas opções"),
      children: [
        Text("Oi, eu sou um texto"),
        ElevatedButton(onPressed: () => print(""), child: Text("Clique em mim"))
      ],
    ), // ExpansionTile
  ); // Scaffold
}
```




```
class _CheckboxScreenState extends State<CheckboxScreen> {  
  bool marcado = true;  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(title: Text("Checkbox")),  
      body: Center(  
        child: Checkbox(  
          value: this.marcado,  
          onChanged: (prChecked) {  
            setState(() {  
              this.marcado = prChecked!;  
            });  
          },  
        ), // Checkbox  
      ), // Center  
    ); // Scaffold  
  }  
}
```

- Um ListTile com uma checkbox integrada

```
class _CheckboxListTileScreenState extends State<CheckboxListTileScreen> {  
  List<TaskModel> tasks = [  
    TaskModel("Roçar a grama", false),  
    TaskModel("Tomar café", false),  
    TaskModel("Ir para a Proway", true)  
  ];  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      body: ListView.builder(  
        itemCount: tasks.length,  
        itemBuilder: (BuildContext context, int index) {  
          return CheckboxListTile(  
            value: this.tasks[index].check,  
            onChanged: (value) {  
              this.tasks[index].check = value!  
            },  
            title: Text(this.tasks[index].title,  
              style: (this.tasks[index].check ? TextStyle(color: Colors.grey) : null)), // Text  
          ); // CheckboxListTile  
        }) // ListView.builder  
      ); // Scaffold  
    )  
  }  
}
```



Vamos criar uma lista de tarefas. Essa lista deve possuir as seguintes características:

- A lista precisa ser cadastravel pelo usuário;
- Cada item deve ser exibido com um check, para que seja marcado como concluído;

