



Flutter



LUANA DIAS

Principais Widgets para Input



No desenvolvimento de aplicativos móveis com Flutter, os widgets de input desempenham um papel essencial ao permitir que os usuários interajam e forneçam informações aos aplicativos. Esses widgets possibilitam a entrada de texto, seleção de datas, escolha de opções e interações específicas, tornando a experiência do usuário mais fluida e intuitiva. Ao utilizar os widgets de input do Flutter, os desenvolvedores têm acesso a uma ampla gama de ferramentas poderosas para capturar e processar dados de forma eficiente. Esses widgets não apenas simplificam a implementação de elementos interativos, mas também oferecem controle detalhado sobre a aparência e o comportamento dos componentes de entrada. Neste eBook, exploraremos alguns dos principais widgets de input do Flutter e demonstraremos como utilizá-los em situações práticas.



Capítulo 01

TextFormField



O widget TextFormField é usado para capturar texto do usuário de maneira fácil e eficiente. Ele fornece um campo de texto que pode ser usado para inserir informações como nome, senha, e-mail, etc. Vamos criar um exemplo simples de um campo de entrada de e-mail:

Exemplo de uso:




Vamos criar um exemplo simples de um campo de entrada de e-mail:

```
TextFormField(  
  decoration: InputDecoration(  
    labelText: 'E-mail',  
    hintText: 'Digite seu e-mail',  
    prefixIcon: Icon(Icons.email),  
    border: OutlineInputBorder(),  
  ),  
  keyboardType: TextInputType.emailAddress,  
  validator: (value) {  
    if (value.isEmpty) {  
      return 'Por favor, insira seu e-mail';  
    }  
    return null;  
  },  
  onSave: (value) {  
    // Lógica para salvar o e-mail  
  },  
)
```

Neste exemplo, utilizamos o `TextFormField` para capturar e validar um endereço de e-mail. O `keyboardType` especifica o tipo de teclado a ser exibido no dispositivo para facilitar a entrada de e-mails.

Capítulo 02

Slider



O Slider é usado para permitir ao usuário selecionar um valor de um intervalo contínuo. Vamos ver como usar um Slider para selecionar um valor numérico:

Exemplo de uso:



O Slider é usado para permitir ao usuário selecionar um valor de um intervalo contínuo. Vamos ver como usar um Slider para selecionar um valor numérico:

```
slider.dart

double _currentSliderValue = 50;

Slider(
  value: _currentSliderValue,
  min: 0,
  max: 100,
  divisions: 5,
  label: _currentSliderValue.round().toString(),
  onChanged: (double value) {
    setState(() {
      _currentSliderValue = value;
    });
  },
)
```

Este exemplo cria um Slider que permite selecionar um valor de 0 a 100 com incrementos visíveis. O onChanged é usado para atualizar o estado do valor selecionado.

Capítulo 3

Checkbox

O widget Checkbox é usado para permitir que o usuário selecione uma ou mais opções em uma lista de seleção múltipla. Ele é ideal para situações em que é necessário permitir escolhas múltiplas por meio de caixas de seleção.

Exemplo de uso:



```
checkBox.dart

bool _isChecked = false;

CheckboxListTile(
  title: Text('Aceitar Termos e Condições'),
  value: _isChecked,
  onChanged: (bool? value) {
    setState(() {
      _isChecked = value!;
    });
  },
)
```

Neste exemplo, um `CheckboxListTile` exibe uma opção para o usuário aceitar os termos e condições. O estado `_isChecked` é atualizado conforme o usuário marca ou desmarca a caixa de seleção.

Capítulo 4

Radio Button

O widget Radio é utilizado quando o usuário deve fazer uma única escolha em um conjunto de opções mutuamente exclusivas. Cada botão de rádio representa uma escolha única dentro do grupo.

Exemplo de uso:



```
RadioButton.dart

enum Gender { male, female }


Gender? _character = Gender.masculino;

Column(
  children: <Widget>[
    ListTile(
      title: Text('Masculino'),
      value: Gender.male,
      groupValue: _selectedGender,
      onChanged: (Gender? value) {
        setState(() {
          _character = value;
        });
      },
    ),
    RadioListTile<Gender>(
      title: Text('Feminino'),
      value: Gender.female,
      groupValue: _character,
      onChanged: (Gender? value) {
        setState(() {
          _character = value;
        });
      },
    ),
  ],
)
```

Neste exemplo, dois `RadioListTile` são utilizados para permitir que o usuário selecione seu gênero. A propriedade `groupValue` indica qual opção está selecionada dentro do grupo.

Capítulo 5

DropDownButton



O DropDownButton exibe uma lista suspensa de opções onde o usuário pode selecionar uma única escolha a partir de uma lista de itens. É uma forma eficaz de economizar espaço na interface do usuário ao exibir opções de seleção.

Exemplo de uso:



```
dropDownButton.dart

String _selectedCity;

DropdownButtonFormField<String>({
  value: _selectedCity,
  items: ['Nova York', 'Los Angeles', 'Chicago', 'Miami']
    .map((city) {
      return DropdownMenuItem<String>({
        value: city,
        child: Text(city),
      });
    })
    .toList(),
  onChanged: (String value) {
    setState(() {
      _selectedCity = value;
    });
  },
  decoration: InputDecoration(
    labelText: 'Selecione uma Cidade',
    border: OutlineInputBorder(),
  ),
})
```

Neste exemplo, um `DropdownButtonFormField` exibe uma lista suspensa de cidades e permite que o usuário selecione uma delas.

Conclusão

Estes são mais alguns widgets de input essenciais no Flutter que podem ser utilizados para criar interfaces ricas e interativas em seus aplicativos móveis. Cada um desses widgets oferece uma maneira intuitiva de capturar diferentes tipos de entrada do usuário, desde seleção de opções até entrada de texto e escolha de datas. Experimente esses widgets em seus projetos para melhorar a experiência do usuário e tornar seus aplicativos mais dinâmicos e envolventes.