

## TEXTFIELD

---

Esse talvez seja o mais comum *Widget* de formulário. Como o próprio nome já sugere, nos fornece um *Widget* “Campo de texto”, onde o usuário poderá inserir informações as quais poderemos obter para realizarmos operações dentro de nossa aplicação, como um cadastro ou atualização de dados.

### *decoration*

---

Essa propriedade nos permite adicionar uma borda decorativa em nosso *Widget TextField*. Esse *Widget* recebe um objeto *InputDecoration*, o qual por sua vez recebe alguns parâmetros como:

- *Border*;
- *HintText*;
- *HintStyle*;
- *Icon*;
- Entre outros.

### *Border*

Como o próprio nome já sugere, nos permite adicionar uma borda em nosso *Widget TextField*. Essa propriedade recebe um objeto *InputBorder*. Normalmente passamos um objeto filho desta classe, como por exemplo, o *OutlineInputBorder* ou *UnderlineInputBorder*, sendo esse o padrão.

```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: Container(  
6     margin: EdgeInsets.only(bottom: 20.0),  
7     child: TextField(  
8       decoration: InputDecoration(  
9         border: OutlineInputBorder(),  
10      ),  
11    ),  
12  ),  
13 )
```

## ***HintText***

Essa propriedade nos permite adicionar um texto de suporte ao preenchimento do campo. Ou seja, um texto temporário informando ao usuário o que deve ser informado naquele campo.

```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: Container(  
6     margin: EdgeInsets.only(bottom: 20.0),  
7     child: TextField(  
8       decoration: InputDecoration(  
9         border: OutlineInputBorder(),  
10        hintText: "Digite seu nome aqui",  
11      ),  
12    ),  
13  ),  
14)
```

## ***HintStyle***

Como já vimos, a propriedade *HintText* é a palavra de suporte para o usuário saber o que deve ser informado em um determinado campo. Já a propriedade *HintStyle* é responsável por estilizar esse texto. Essa propriedade recebe um objeto *TextStyle*, nosso velho conhecido.

```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: Container(  
6     margin: EdgeInsets.only(bottom: 20.0),  
7     child: TextField(  
8       decoration: InputDecoration(  
9         border: OutlineInputBorder(),  
10        hintText: "Digite seu nome aqui",  
11        hintStyle: TextStyle(  
12          fontSize: 12.0,  
13        ),  
14      ),  
15    ),  
16  ),  
17)
```

## ***Icon***

Essa propriedade dispensa apresentações, já que o próprio nome já descreve sua finalidade. Essa propriedade recebe um objeto *Icon*, o qual será adicionado à esquerda do

*Widget TextField.*

```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: Container(  
6     margin: EdgeInsets.only(bottom: 20.0),  
7     child: TextField(  
8       decoration: InputDecoration(  
9         border: OutlineInputBorder(),  
10        hintText: "Digite seu nome aqui",  
11        hintStyle: TextStyle(  
12          fontSize: 12.0,  
13        ),  
14        icon: Icon(Icons.account_circle),  
15      ),  
16    ),  
17  ),  
18)
```

### *enabled*

A propriedade *enabled*, como o próprio nome já sugere, nos permite definir se um *Widget TextField* está ativo ou não, ou seja, se o usuário poderá interagir com ele ou não.

```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: Container(  
6     margin: EdgeInsets.only(bottom: 20.0),  
7     child: TextField(  
8       decoration: InputDecoration(  
9         border: OutlineInputBorder(),  
10        hintText: "Digite seu nome aqui",  
11        hintStyle: TextStyle(  
12          fontSize: 12.0,  
13        ),  
14        icon: Icon(Icons.account_circle),  
15      ),  
16      enabled: false,  
17    ),  
18  ),  
19)
```

### *style*

A propriedade *style* dispensa apresentações, pois já a utilizamos em diversos outros

Widgets. Essa propriedade nos permite alterar o estilo do texto do próprio *TextField*.

```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: Container(  
6     margin: EdgeInsets.only(bottom: 20.0),  
7     child: TextField(  
8       style: TextStyle(  
9         fontSize: 20.0,  
10        fontWeight: FontWeight.bold,  
11      ),  
12    ),  
13  ),  
14)
```

### ***expands***

Como o próprio nome já sugere, essa propriedade nos permite expandir a altura do *Widget TextField*. Como essa propriedade podemos criar caixas de texto mais longas, para que o usuário possa digitar grandes quantidades de texto.

```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: Container(  
6     margin: EdgeInsets.only(bottom: 20.0),  
7     child: TextField(  
8       expands: true,  
9       // Desabilite o número máximo de linhas  
10      maxLines: null,  
11    ),  
12  ),  
13)
```

### ***keyboardType***

A propriedade nos permite definir o tipo do teclado que será exibido para o usuário no momento em que um determinado *TextField* for selecionado. Isso é muito útil como

estratégia para impedir o usuário de fornecer valores incorretos em um determinado campo. Por exemplo, em um campo de valor (Preço), não queremos permitir a adição de letras ou caracteres especiais indesejados.

Essa propriedade recebe um Objeto *TextInputType*. Para tal utilizamos uma das constantes desse mesmo objeto. Dentre as opções de teclado temos:

- *datetime*: Otimizado para datas e horas;
- *emailAddress*: Otimizado para endereços de email;
- *multiline*: Otimizado para várias linhas de texto;
- *name*: Otimizado para nomes próprios;
- *number*: Otimizado para números sem precisão;
- *numberWithOptions({signed,decimal})*: Otimizado para números com precisão;
- *phone*: Otimizado para números de telefone;
- *streetAddress*: Otimizado para nomes de ruas e logradouros
- *text*: Otimizado para textos simples;
- *url*: Otimizado para endereços URL;
- *values*: Recebe uma lista de outros *TextInputType*;
- *visiblePassword*: Otimizado para passwords visíveis.

```

1 Scaffold(
2   appBar: AppBar(
3     title: Text("Aplicação Exemplo"),
4   ),
5   body: Container(
6     margin: EdgeInsets.only(bottom: 20.0),
7     child: TextField(
8       keyboardType: TextInputType.number,
9     ),
10  ),
11)

```

### ***obscureText***

Essa propriedade recebe um valor booleano, o qual define se o campo deverá ter seus caracteres ocultos ou não. Esta propriedade é muito utilizada em campos de senha ou que desejamos manter o conteúdo em segredo.

```

1 Scaffold(
2   appBar: AppBar(
3     title: Text("Aplicação Exemplo"),
4   ),
5   body: Container(
6     margin: EdgeInsets.only(bottom: 20.0),
7     child: TextField(
8       obscureText: true,
9     ),
10  ),
11)

```

### ***obscuringCharacter***

Essa propriedade recebe um *caractere*, o qual será utilizado como caractere de ocultação. Se essa propriedade não foi informada, o Flutter usará uma pequena bola preta para tal.

```

1 Scaffold(
2   appBar: AppBar(
3     title: Text("Aplicação Exemplo"),
4   ),
5   body: Container(
6     margin: EdgeInsets.only(bottom: 20.0),
7     child: TextField(
8       obscureText: true,
9       obscuringCharacter: "*",
10    ),
11  ),
12 ),
13 )

```

### *onChanged*

A propriedade *onChanged*, como o próprio nome já sugere, recebe uma função, a qual será executada toda vez que o texto do *TextField* for modificado. Ou seja, sempre que o usuário inserir ou remover um caractere.

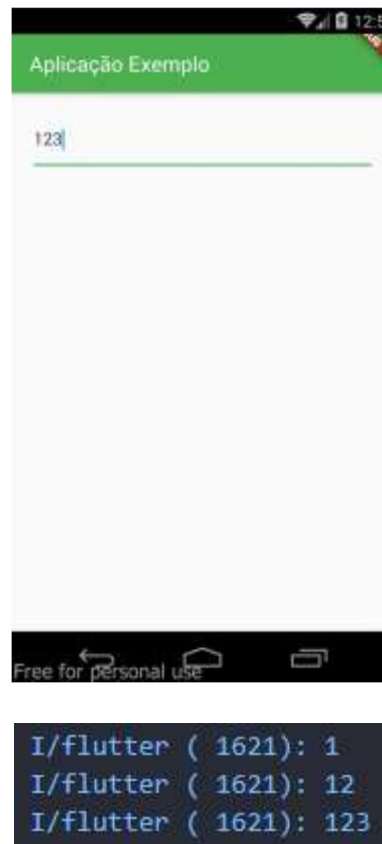
```

1 Scaffold(
2   appBar: AppBar(
3     title: Text("Aplicação Exemplo"),
4   ),
5   body: Container(
6     margin: EdgeInsets.only(bottom: 20.0),
7     child: TextField(
8       onChanged: (String texto) {
9         print(texto);
10      },
11    ),
12  ),
13 )

```



Resultado:



### ***onSubmitted***

A propriedade *onSubmitted* é muito parecida com o *onChanged*, diferenciando-se pelo fato de ser executada quando o botão de submissão é pressionado.

```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: Container(  
6     margin: EdgeInsets.only(bottom: 20.0),  
7     child: TextField(  
8       onSubmitted: (String texto) {  
9         print(texto);  
10      },  
11    ),  
12  ),  
13)
```



Resultado:



```
I/flutter ( 1621): 123
```