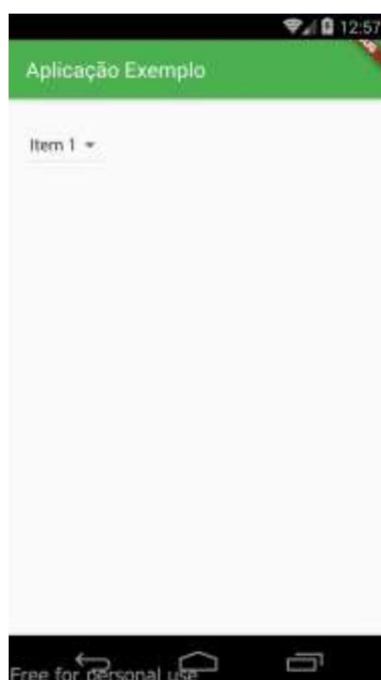


DROPDOWNBUTTON

O *Widget DropdownButton* nos permite adicionar um botão suspenso em nossa tela. Esse *Widget* permite adição de diversas opções de seleção. Esses elementos são objetos *DropdownMenuItem*. Para tal utilizamos a propriedade *items*;

```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: Container(  
6     margin: EdgeInsets.all(20.0),  
7     child: DropdownButton<String>(  
8       value: "Item 1",  
9       items: [  
10        DropdownMenuItem(  
11          value: "Item 1",  
12          child: Text("Item 1"),  
13        ),  
14        DropdownMenuItem(  
15          value: "Item 2",  
16          child: Text("Item 2"),  
17        ),  
18        DropdownMenuItem(  
19          value: "Item 3",  
20          child: Text("Item 3"),  
21        )  
22      ],  
23      onChanged: () {},  
24    ),  
25  ),  
26)
```

Resultado:



Esse Widget possui algumas propriedades importantes que devem ser conhecidas por qualquer desenvolvedor Flutter. Vejamos quais são elas.

Propriedade `dropdownColor`

A propriedade `dropdownColor`, como o próprio já sugere, nos permite definir uma cor de fundo para o *Dropdown*.

A screenshot of a code editor with a dark background and light-colored text. The code is written in Dart and defines a Flutter widget. It starts with a `Scaffold` widget containing an `AppBar` with the title "Aplicação Exemplo". The main body is a `Container` with a margin and a `DropdownButton<String>` child. The dropdown button is configured with a value of "Item 1", a list of items (commented as `DropdownMenuItem`), and the `dropdownColor` property set to `Colors.greenAccent`.

```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: Container(  
6     margin: EdgeInsets.all(20.0),  
7     child: DropdownButton<String>(  
8       value: "Item 1",  
9       items: [  
10         // DropdownMenuItem  
11       ],  
12       dropdownColor: Colors.greenAccent,  
13     ),  
14   ),  
15 )
```

Propriedade *hint*

A propriedade *hint* nos permite definir um texto alternativo, o qual será exibido sempre que nenhum item estiver selecionado.



```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: Container(  
6     margin: EdgeInsets.all(20.0),  
7     child: DropdownButton<String>(  
8       value: "Item 1",  
9       items: [  
10        // DropdownMenuItem  
11      ],  
12     hint: Text("Selecione um item"),  
13   ),  
14 ),  
15)
```

Propriedade *style*

Essa propriedade, como o próprio nome já sugere, nos permite estilizar o texto presente nas opções de seleção.

Essa propriedade deve receber um objeto *TextStyle*. Dentre as possíveis customizações encontramos:

- *color*;
- *decoration*;
- *fontFamily*;
- *fontSize*;
- *fontStyle*;

- *fontWeight*.

```
1 Scaffold(  
2   appBar: AppBar(  
3     title: Text("Aplicação Exemplo"),  
4   ),  
5   body: Container(  
6     margin: EdgeInsets.all(20.0),  
7     child: DropdownButton<String>(  
8       value: "Item 1",  
9       items: [  
10        // DropdownMenuItem  
11      ],  
12     hint: Text("Selecione um item"),  
13   ),  
14   style: TextStyle(  
15     color: Colors.green,  
16     fontSize: 16.0,  
17   ),  
18 ),  
19 )
```

Propriedade underline

A propriedade *underline*, como o próprio nome já sugere, nos permite adicionar uma linha horizontal abaixo do *Widget*. Para isso, utilizamos um outro *Widget* “box”. Para nosso exemplo, utilizamos o *Widget Container*.

```

1 Scaffold(
2   appBar: AppBar(
3     title: Text("Aplicação Exemplo"),
4   ),
5   body: Container(
6     margin: EdgeInsets.all(20.0),
7     child: DropdownButton<String>(
8       value: "Item 1",
9       items: [
10         // DropdownMenuItem
11       ],
12       hint: Text("Selecione um item"),
13     ),
14     style: TextStyle(
15       color: Colors.green,
16       fontSize: 16.0,
17     ),
18     isExpanded: true,
19     underline: Container(
20       height: 2,
21       color: Colors.green,
22     ),
23   ),
24 )

```