



Flutter Avançado - Aula 3

Exercícios 8 a ?

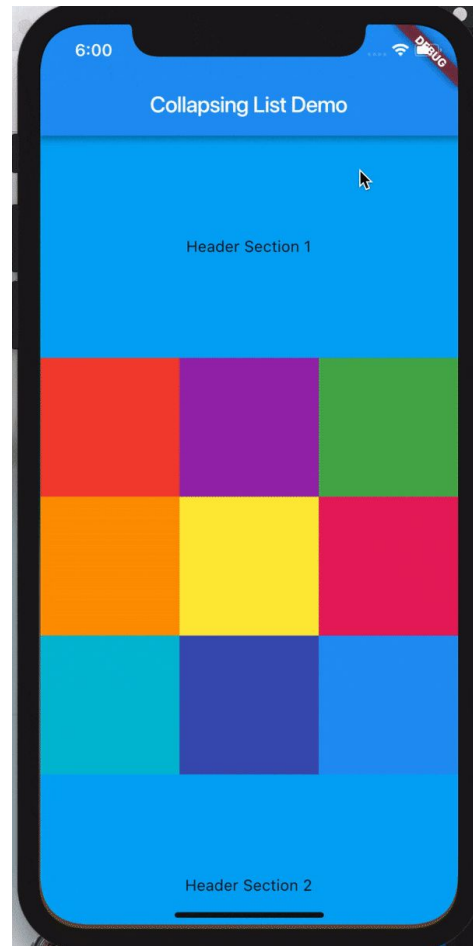


- FlutLab.io

[Ver mais](#)



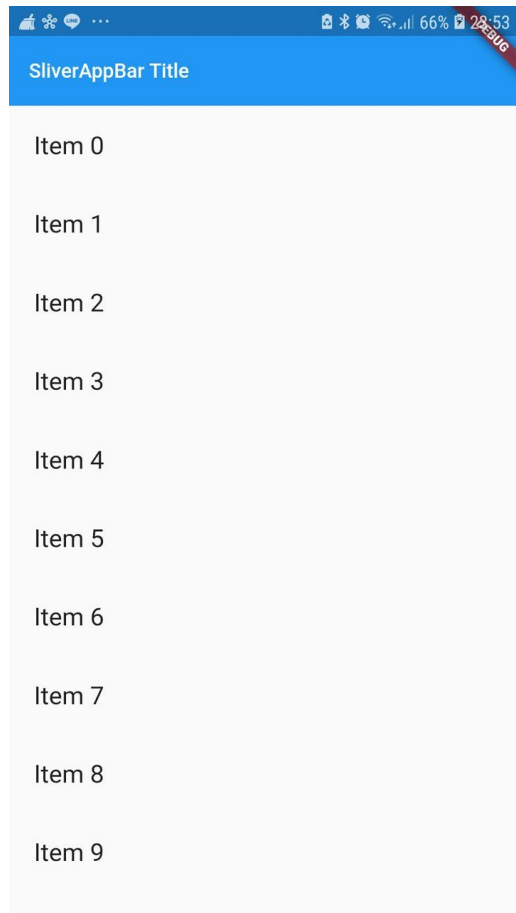
- Componente de *scroll* (*ScrollArea*)
- *Widgets* que nós já vimos, como *ListView* e *GridView* utilizam *Slivers*
- Com *slivers*, temos maior controle sobre o comportamento de *scroll*
- Além de maior controle,





- Uma lista, utilizando *Slivers*
- Para cada lista, deve ser utilizado um *delegate*
- Deve ser sempre utilizado em conjunto a uma *CustomScrollView*

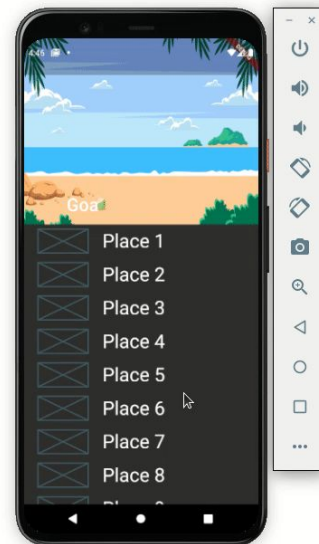
```
SliverList(  
  delegate: SliverChildBuilderDelegate((context, index) {  
    return Container(  
      height: 50,  
      child: Text("${index}"),  
    );  
  }, childCount: 20),  
)
```





- Semelhante a *AppBar*, se expande e colapsa conforme é efetuado o *scroll* em tela
- Deve ser sempre utilizado em conjunto a uma *CustomScrollView*

```
SliverAppBar(  
  pinned: true,  
  snap: true,  
  floating: true,  
  expandedHeight: 160,  
  flexibleSpace: FlexibleSpaceBar(  
    title: Text("Meu teste"),  
    background: FlutterLogo(),  
  ),  
),
```



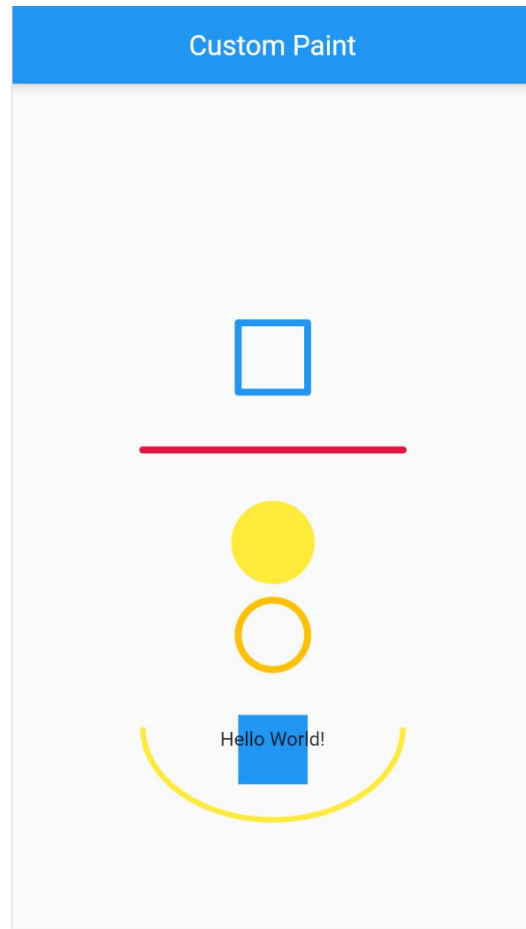


- **pinned:** Determina se a barra ficará sempre visível em tela
- **snap:** Adiciona suavidade ao esconder/exibir a AppBar
- **floating:** Determina que a barra ficará visível assim que o usuário começar a efetuar o *scroll* para cima





- Componente que permite a criação de desenhos
- O *widget* disponibilizará um *canvas* (uma tela) para desenho
- Deveremos implementar um *delegate*, chamado de *painter*





paint

Esse método é chamado sempre que o componente deverá ser repintado. Receberemos por parâmetro o tamanho disponível e a tela em que iremos pintar.

```
void paint(Canvas canvas, Size size) {  
    ...  
}
```



shouldRepaint

Esse método determinará se nossa imagem deve ser pintada novamente ou não. Recebemos por parâmetro o nosso antigo *delegate*, e retornamos um valor *boolean*, determinando se a imagem deve ser refeita ou não.

```
bool shouldRepaint(SunflowerPainter oldDelegate) {  
  
}
```