



14° CLASS

NAV BOTTOM BAR && LIST VIEW

ROTEIRO



1

NAV BAR:
O QUE É,
O QUE MELHORA ,
COMO FAZER

2

LIST VIEW:
O QUE É ,
O QUE MELHORA,
COMO FAZER

3

QUIZ - COMO OS
VELHOS TEMPOS.

FAB center docked

Press the button below!

You



Alarm

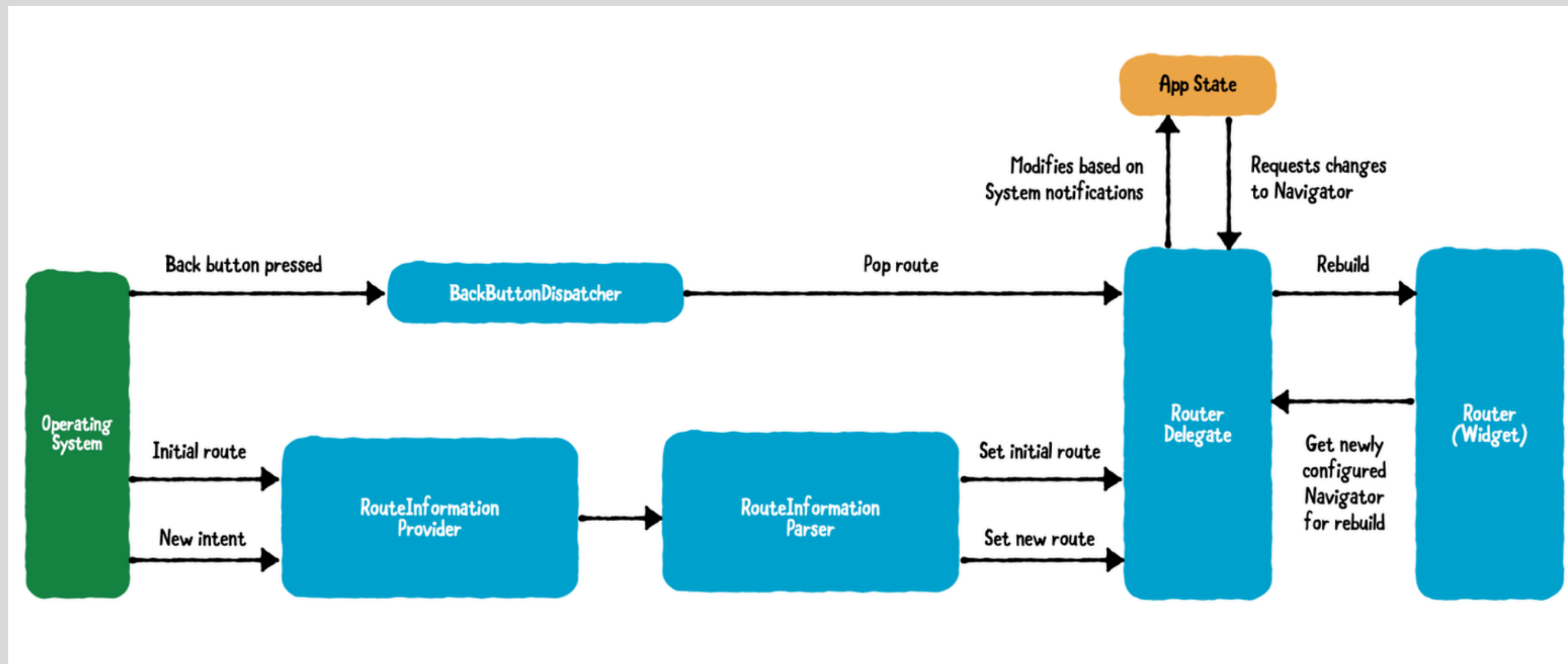
NAV BOTTOM BAR

É A BARRA INFERIOR DE NAVEGAÇÃO
ENTRE TELAS.



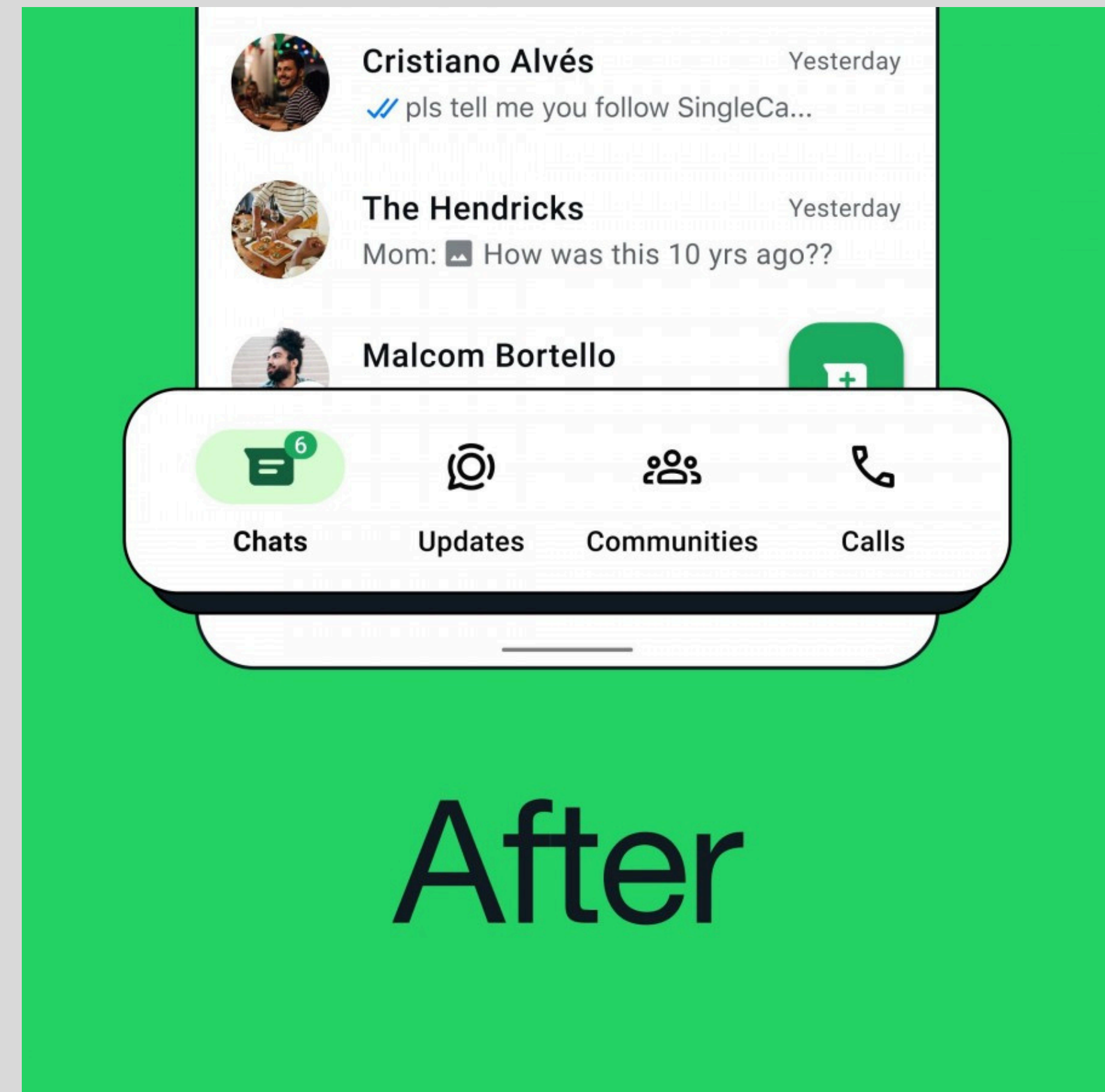
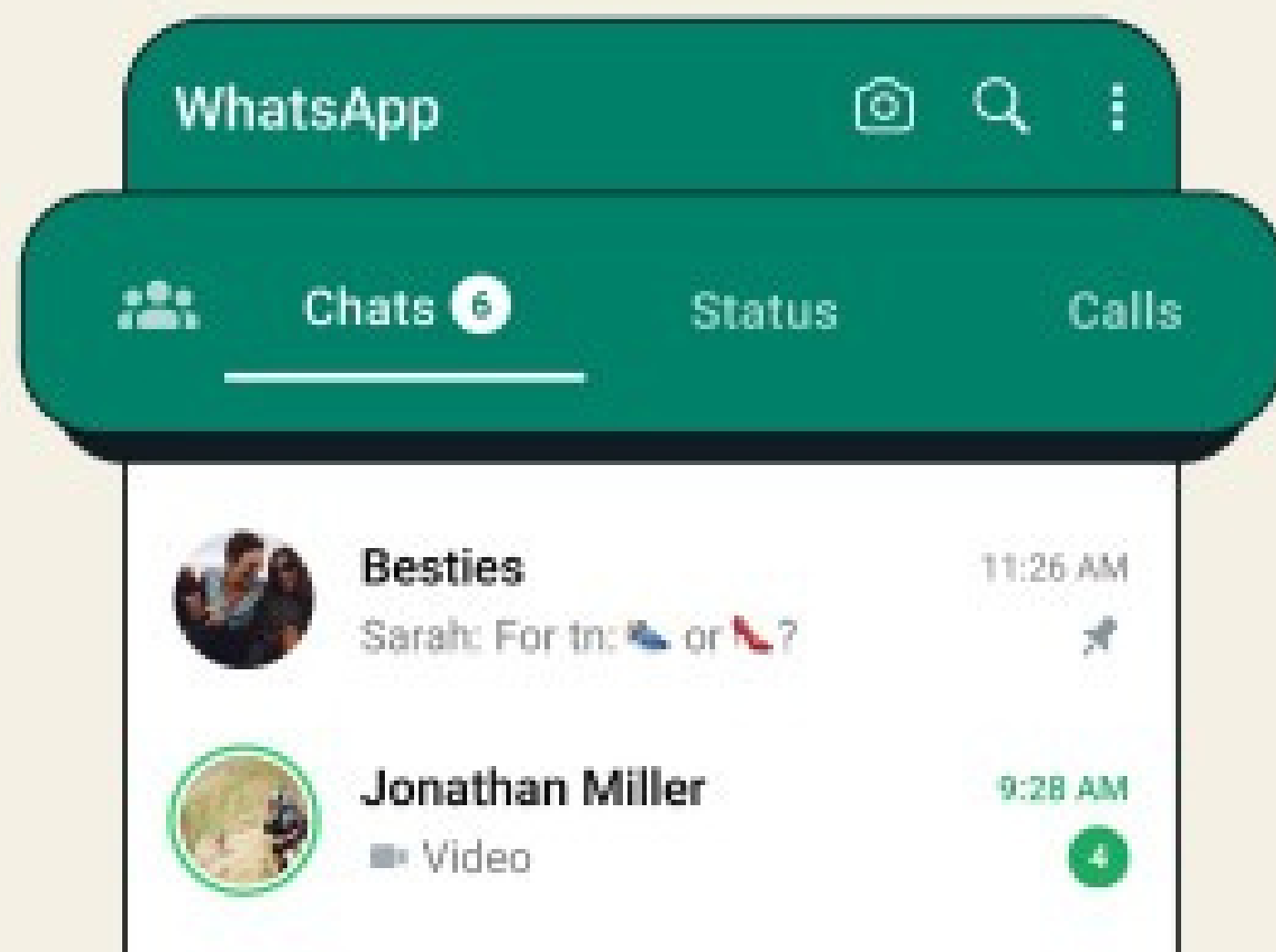
O QUE MELHORA?

EVITA CONSTRUIR
VÁRIOS BOTÕES EM
TODAS AS TELAS.



USABILIDADE

Before



After

COMO USAR?

1

EM UM ARQUIVO
SEPARADO CRIE UMA
CLASSE STATEFUL,
PARA A NAVEGAÇÃO.

```
class NavFooter extends StatefulWidget {  
  const NavFooter({super.key});  
  
  @override  
  State<NavFooter> createState() => _NavFooterState();  
}  
  
class _NavFooterState extends State<NavFooter> {
```

ANTES DO SEGUNDO
OVERRIDE CRIE UMA
VARIÁVEL QUE SERÁ
O INDEX DE CADA
TELA.

```
class NavFooter extends StatefulWidget {  
  const NavFooter({super.key});  
  
  @override  
  State<NavFooter> createState() => _NavFooterState();  
}
```

```
class _NavFooterState extends State<NavFooter> {
```

```
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Flutter Demo',  
      theme: ThemeData(  
        primaryColor: Colors.blue,  
      ),  
      home: NavFooter(),  
    );  
  }  
}
```

```
class NavFooter extends StatefulWidget {  
  const NavFooter({super.key});  
  
  @override  
  State<NavFooter> createState() => _NavFooterState();  
}
```

```
class _NavFooterState extends State<NavFooter> {  
  int currentIndex = 0;
```

```
  void onTapIndex(int index) {
```


CRIE UMA FUNÇÃO
QUE ALTERA O INDEX
TODA VEZ QUE
APERTAR EM UMA
TELA DIFERENTE

```
class NavFooter extends StatefulWidget {  
  const NavFooter({super.key});  
  
  @override  
  State<NavFooter> createState() => _NavFooterState();  
}  
  
class _NavFooterState extends State<NavFooter> {  
  int currentIndex = 0;  
  
  void onTapIndex(int index){  
    setState(() {  
      currentIndex = index;  
    });  
  }  
}
```



```
class NavFooter extends StatefulWidget {  
  const NavFooter({super.key});  
  
  @override  
  State<NavFooter> createState() => _NavFooterState();  
}
```

```
class _NavFooterState extends State<NavFooter> {  
  int currentIndex = 0;  
  
  void onTapIndex(int index){  
    setState(() {  
      currentIndex = index;  
    });  
  }  
}
```

```
List<Widget> screens = [  
  Page1(),  
  Page2(),  
  Page3(),  
];
```

```
@override
```

4

**CRIE UMA LISTA DE
TELAS.**

OBS:

**VAI DAR ERRO , POIS
AINDA NÃO CRIAMOS
AS OUTRAS TELAS.**

5

NO RETURN, CRIE O
M.APP E O SCAFFOLD

```
@override  
Widget build(BuildContext context) {  
  return MaterialApp(  
    home: Scaffold(  
      children: [
```

6

NO BODY, INDICAMOS QUE VAMOS TER AS
TELAS DA LISTA ,POREM SERÁ MOSTRADA A
TELA QUE ESTÁ NO INDEX DESEJADO.

```
@override  
Widget build(BuildContext context) {  
  return MaterialApp(  
    home: Scaffold(  
      body: screens.elementAt(currentIndex),
```

DEPOIS DO BODY, MAS AINDA DENTRO DO
SCAFFOLD, COLOQUE A ESTRUTURA DO
BOTTOM NAVIGATIONBAR.

```
@override
Widget build(BuildContext context) {
  return MaterialApp(
    home: Scaffold(
      body: screens.elementAt(currentIndex),

      bottomNavigationBar: BottomNavigationBar(items: <BottomNavigationBarItem> [
        BottomNavigationBarItem(label: "page1" ,icon: Icon(Icons.pages),),
        BottomNavigationBarItem(label: "page2" ,icon: Icon(Icons.pages),),
        BottomNavigationBarItem(label: "page3" ,icon: Icon(Icons.pages),),
      ], // <BottomNavigationBarItem>[]
      onTap: onTapIndex,
      currentIndex: currentIndex,

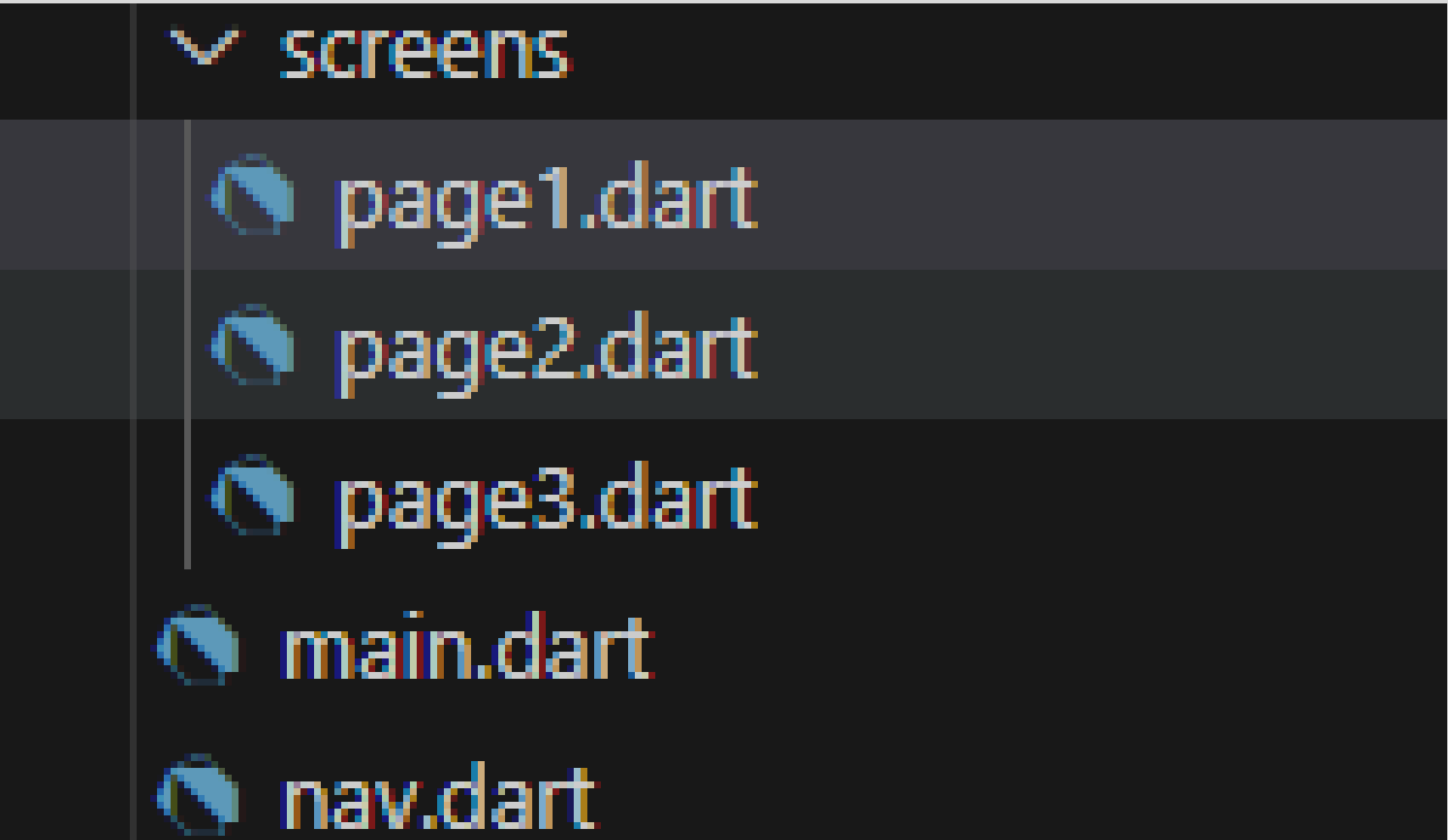
    ), // BottomNavigationBar

  ), // Scaffold
); // MaterialApp
```

8

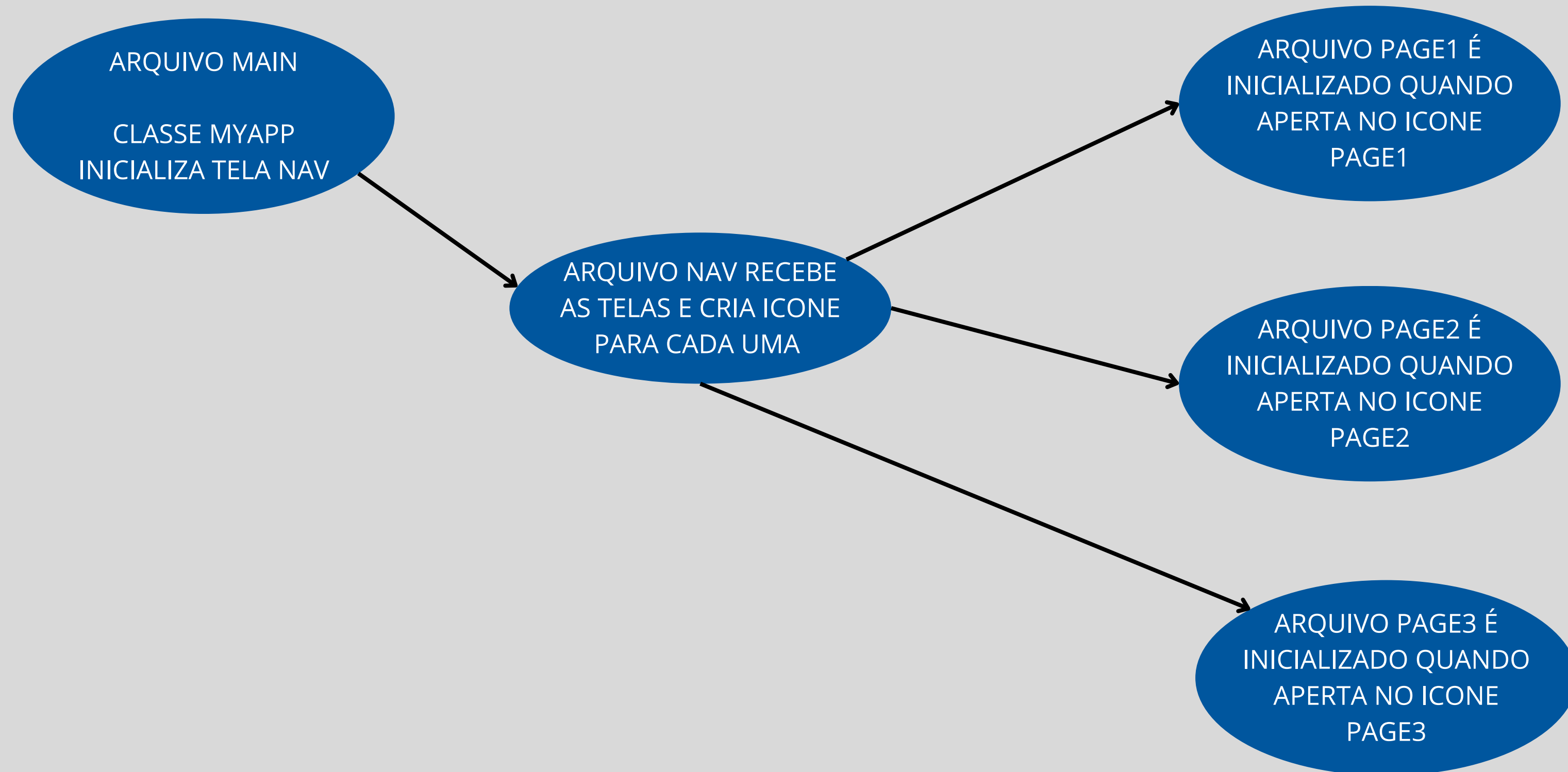
**CRIE SUA CLASSE CENTRALIZADORA , QUE
CHAMA A NAVBAR, E DEPOIS CRIE AS 3
TELAS.**

```
main.dart > MyApp
1 import 'package:flutter/material.dart';
2 import 'package:nav_bar_example/nav.dart';
3
Run | Debug | Profile
4 void main() {
5   runApp(const MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   const MyApp({super.key});
10
11   @override
12   Widget build(BuildContext context) {
13     return MaterialApp(home: NavFooter());
14   }
15 }
```



```
▼ screens
  ◉ page1.dart
  ◉ page2.dart
  ◉ page3.dart
  ◉ main.dart
  ◉ nav.dart
```

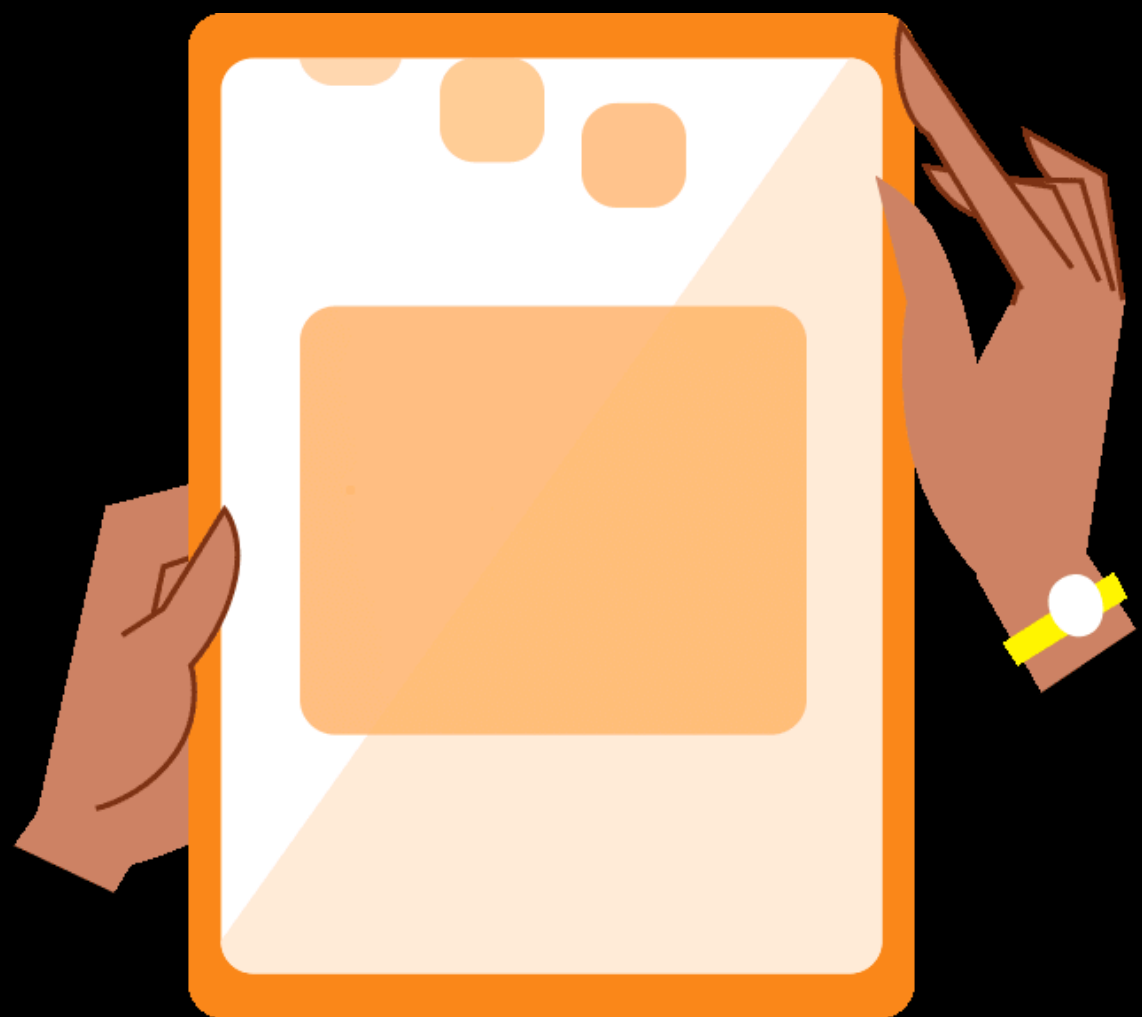
FUNCIONAMENTO





RESULTADO

NAV BOTTOM BAR



LIST VIEW

ROLAGEM DE TELA

É A POSSIBILIDADE DE TER UMA TELA COM
CONTEÚDOS INFINITOS. SEM LIMITE DE
PIXELS.

IMAGINE UMA TELA QUE PRECISE DE 4
CONTAINERS COM:
WIDTH: 500
HEIGHT: 500



CÓDIGO

```
import 'package:flutter/material.dart';

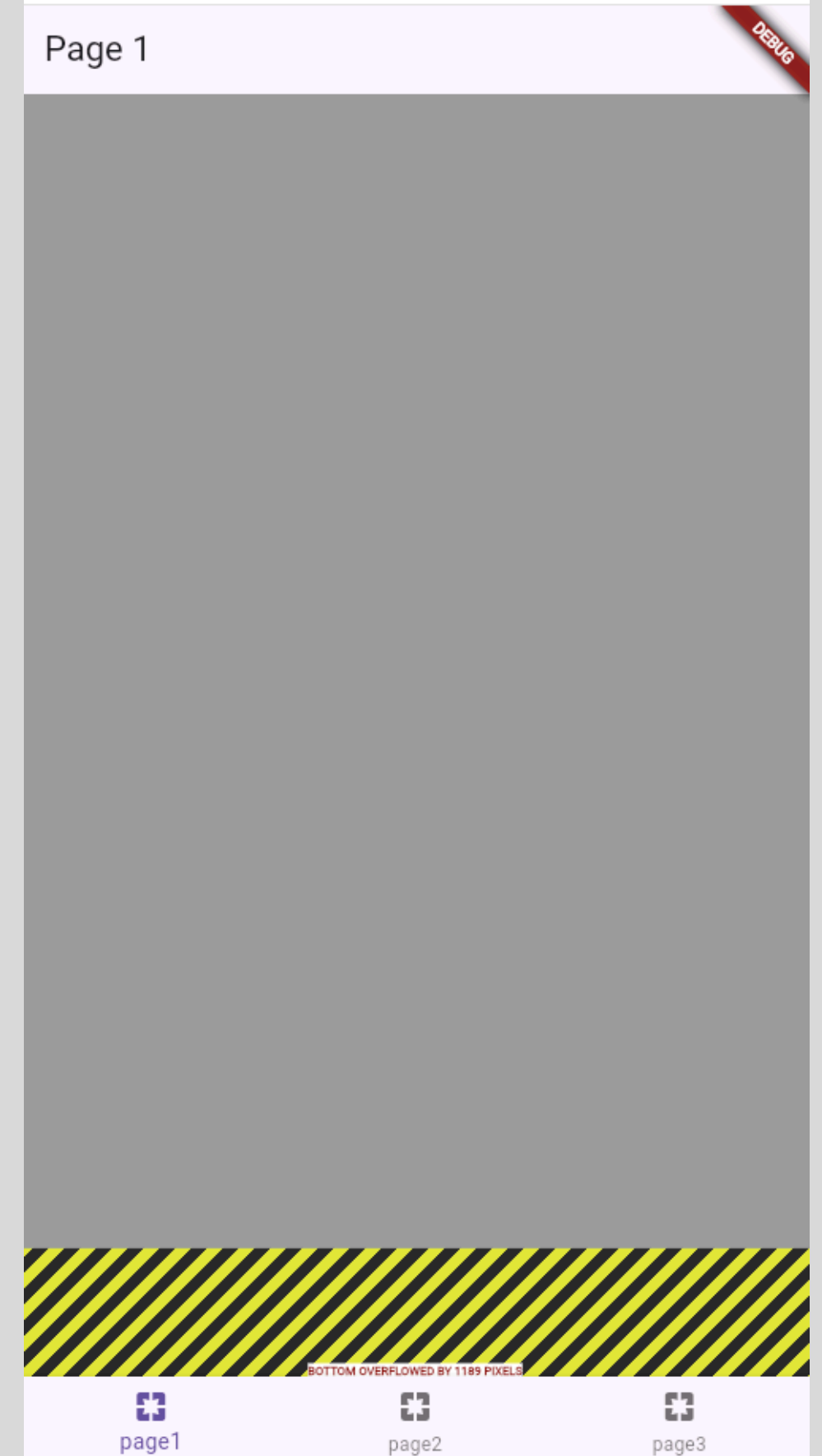
class Page1 extends StatelessWidget {
  const Page1({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: Text("Page 1")),
        body:
          ListView(
            children: [
              Column(
                children: [
                  Container(width: 500, height: 500, color: Colors.grey),
                  Container(width: 500, height: 500, color: const Color.fromARGB(255, 236, 30, 30)),

                  Container(width: 500, height: 500, color: const Color.fromARGB(255, 14, 212, 57)),

                  Container(width: 500, height: 500, color: const Color.fromARGB(255, 192, 22, 22)),
                ],
              ), // Column
            ],
          ) // ListView
      ), // Scaffold
    ); // MaterialApp
  }
}
```

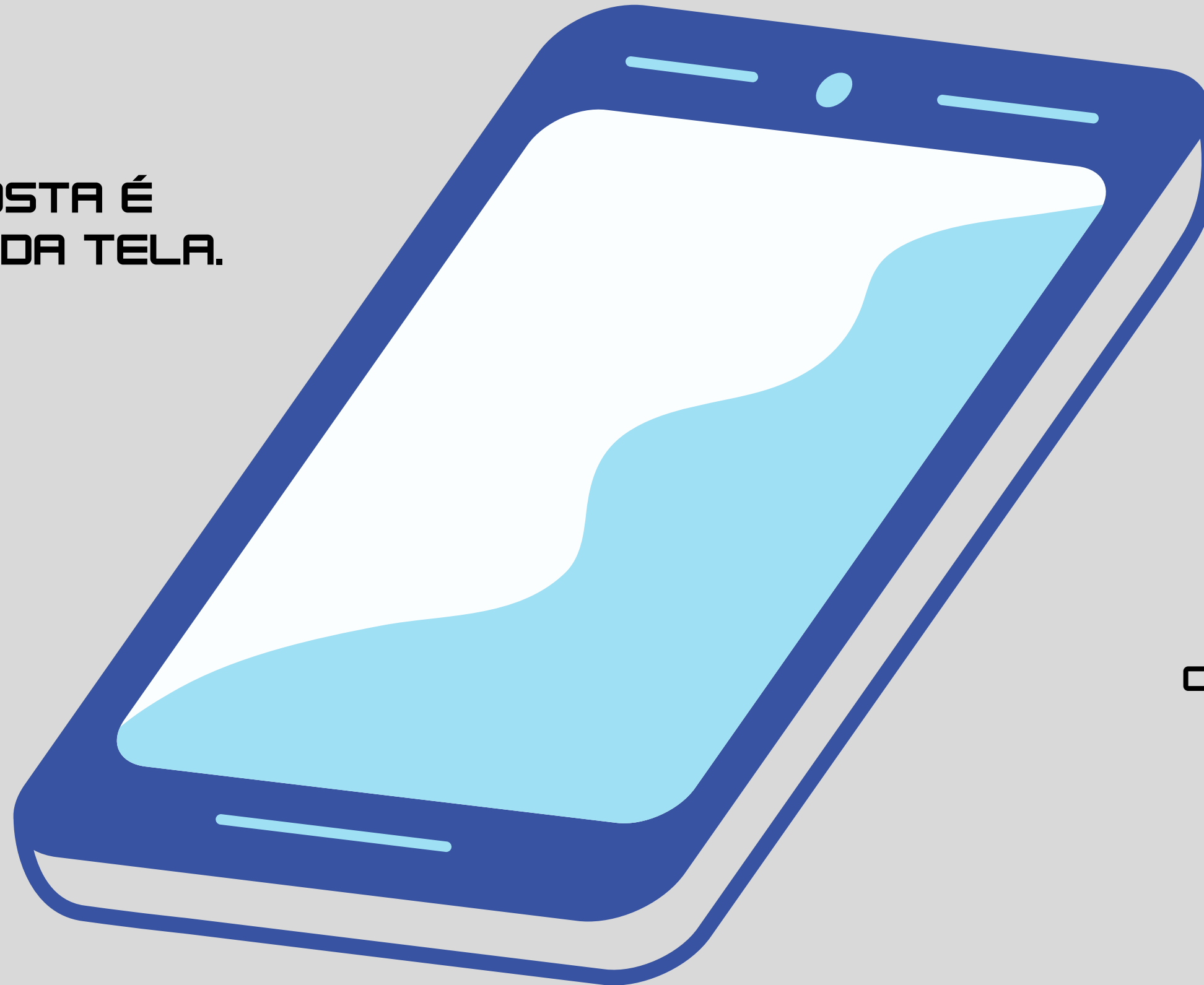
RESULTADO



**VEJA QUE DEU ERRO,
POR QUE SERÁ?**



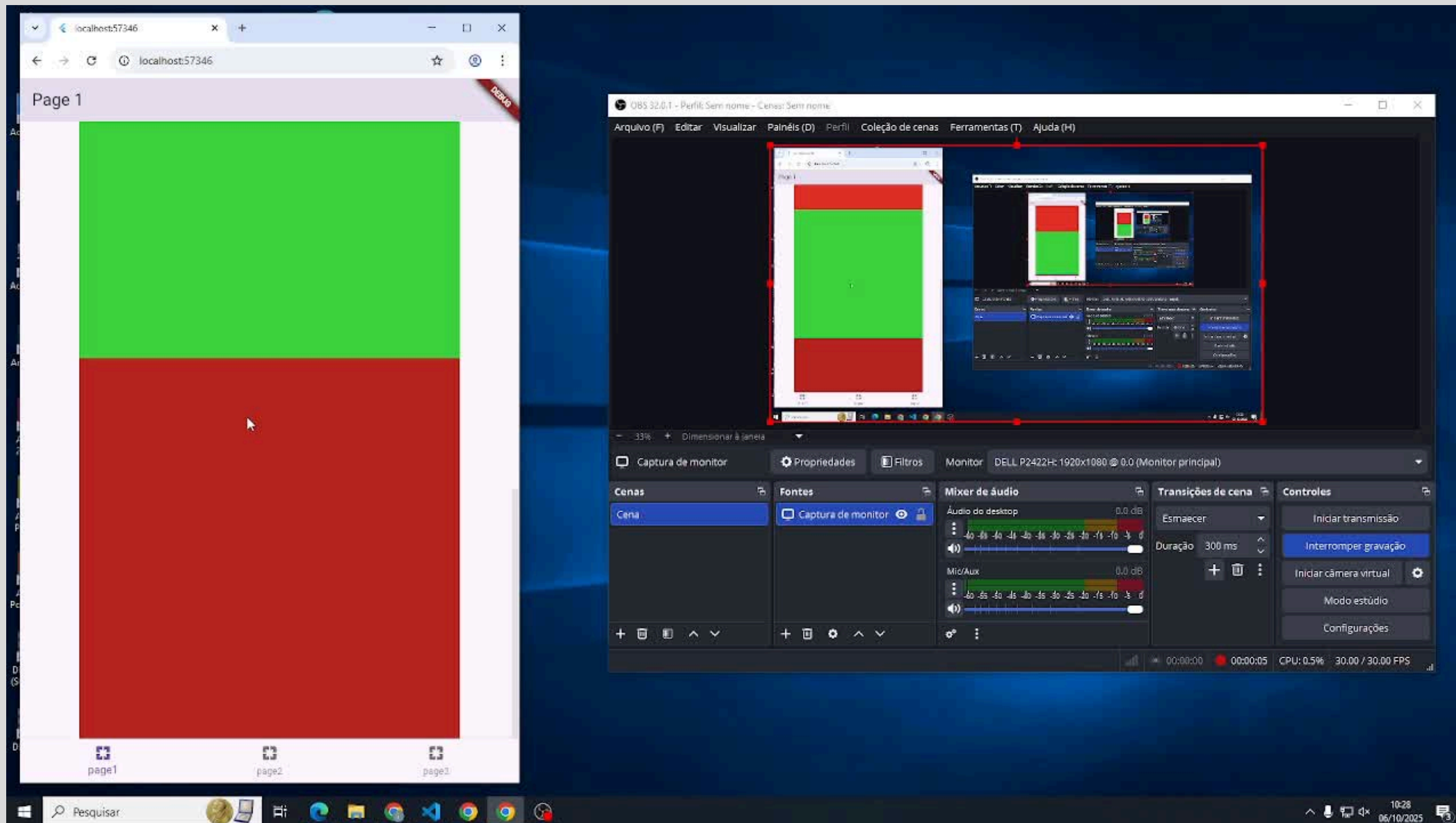
**A RESPOSTA É
DIMENSÃO DA TELA.**



**O TAMANHO DOS 4
CONTAINERS É MAIOR QUE A
DIMENSÃO DA TELA DO
CELULAR.**

PARA RESOLVERMOS
ISSO ,

USAMOS O LISTVIEW.



OBRIGADO!

FLUTTER WITH VINI;

