

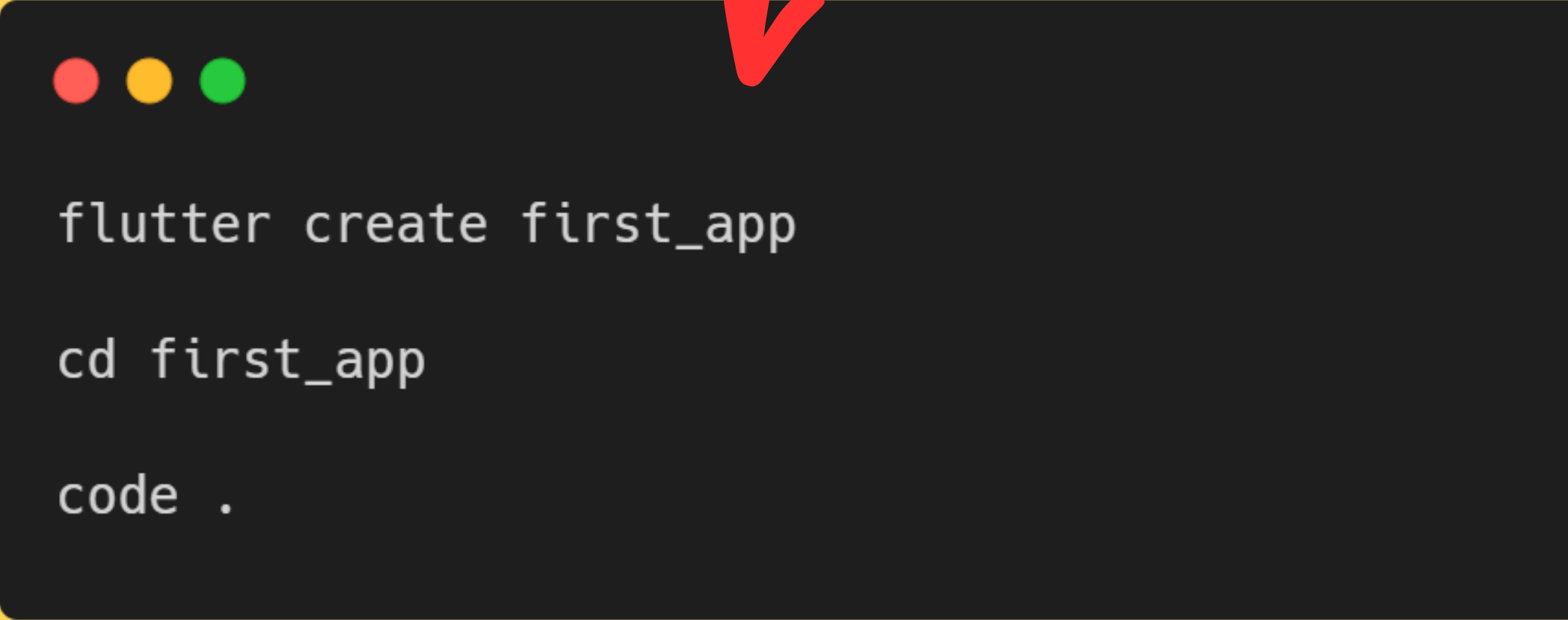


Flutter Widgets



Criando nosso primeiro app

No terminal

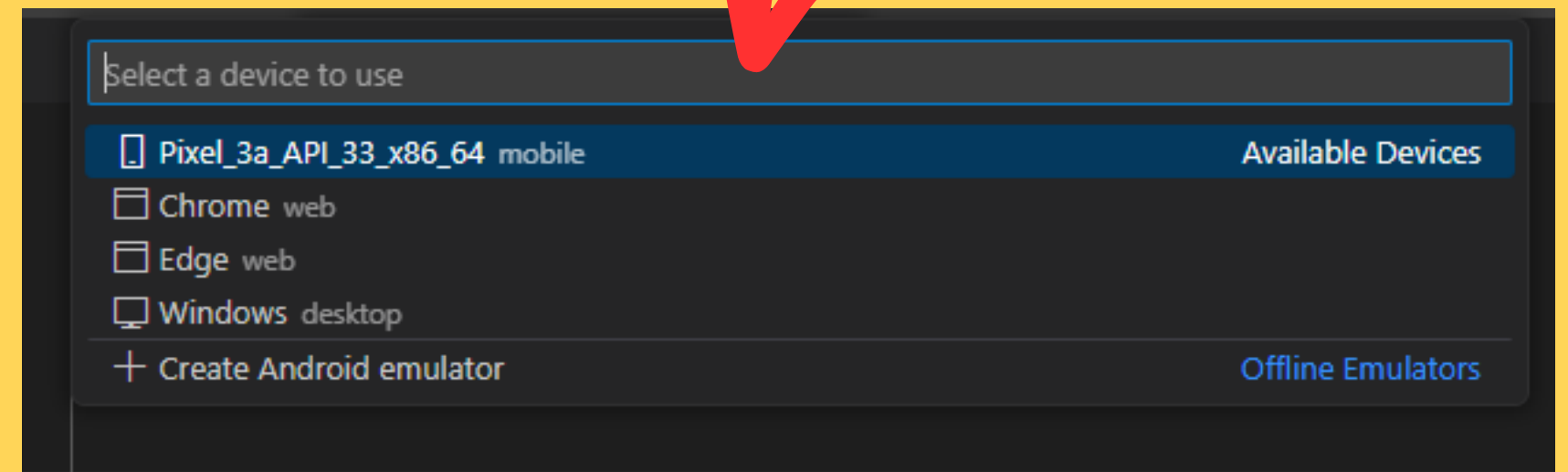
A dark-themed terminal window with three colored window control buttons (red, yellow, green) in the top left corner. A red curved arrow points from the text 'No terminal' to the top right corner of the terminal window.

```
flutter create first_app  
  
cd first_app  
  
code .
```

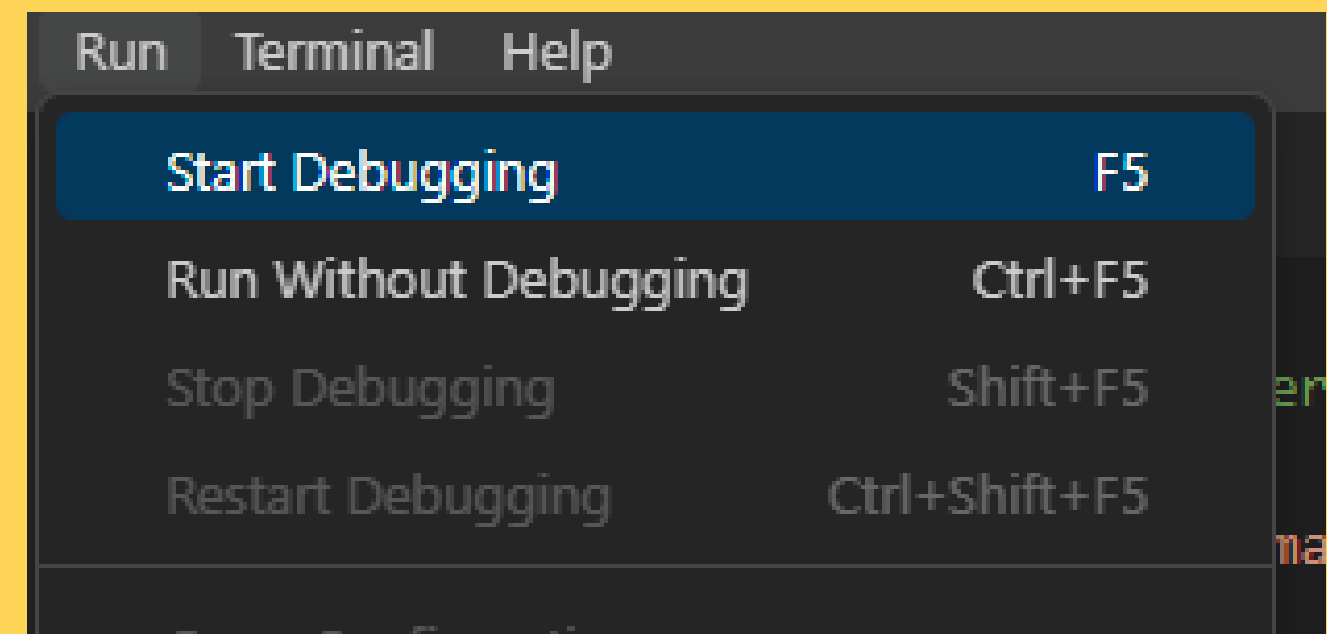
1 - Clique no canto inferior direito onde aparece os dispositivos

{ } Dart Pixel_3a_API_33_x86_64 (android-x64 emulator)

2 - Selecione um dispositivo



3 - Inicie em Debugging



Widgets



- Widgets podem ser entendidos como a representação visual (mas não somente isso) de partes do aplicativo.
- Muitos widgets são colocados juntos para compor a interface do usuário de um aplicativo. Imagine como um quebra-cabeça em que você define as peças.

Everything is a widget

Widgets Flutter estão em toda parte em um aplicativo. Talvez nem tudo seja um widget, mas quase tudo é. Até o app é um widget no Flutter, e é por isso que esse conceito é tão importante. Um widget representa uma parte de uma interface do usuário, mas isso não significa que seja apenas algo que é visível. Pode ser qualquer um dos seguintes:

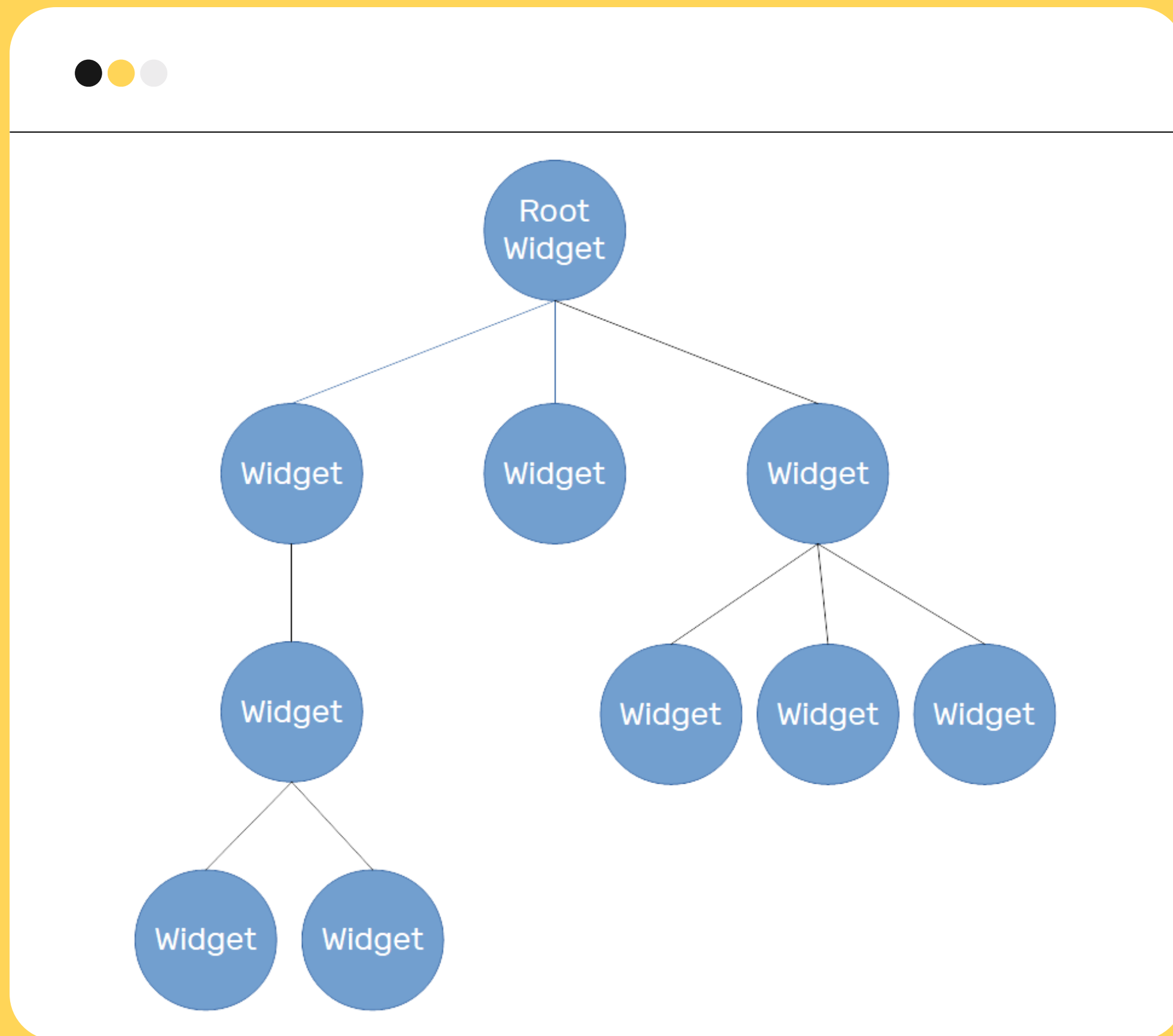
Elementos visuais textos e botões

Elementos de layouts que definem posição, margins e paddings

Elementos de estilo, como cores e temas

Elementos de interação como toques e gestos.

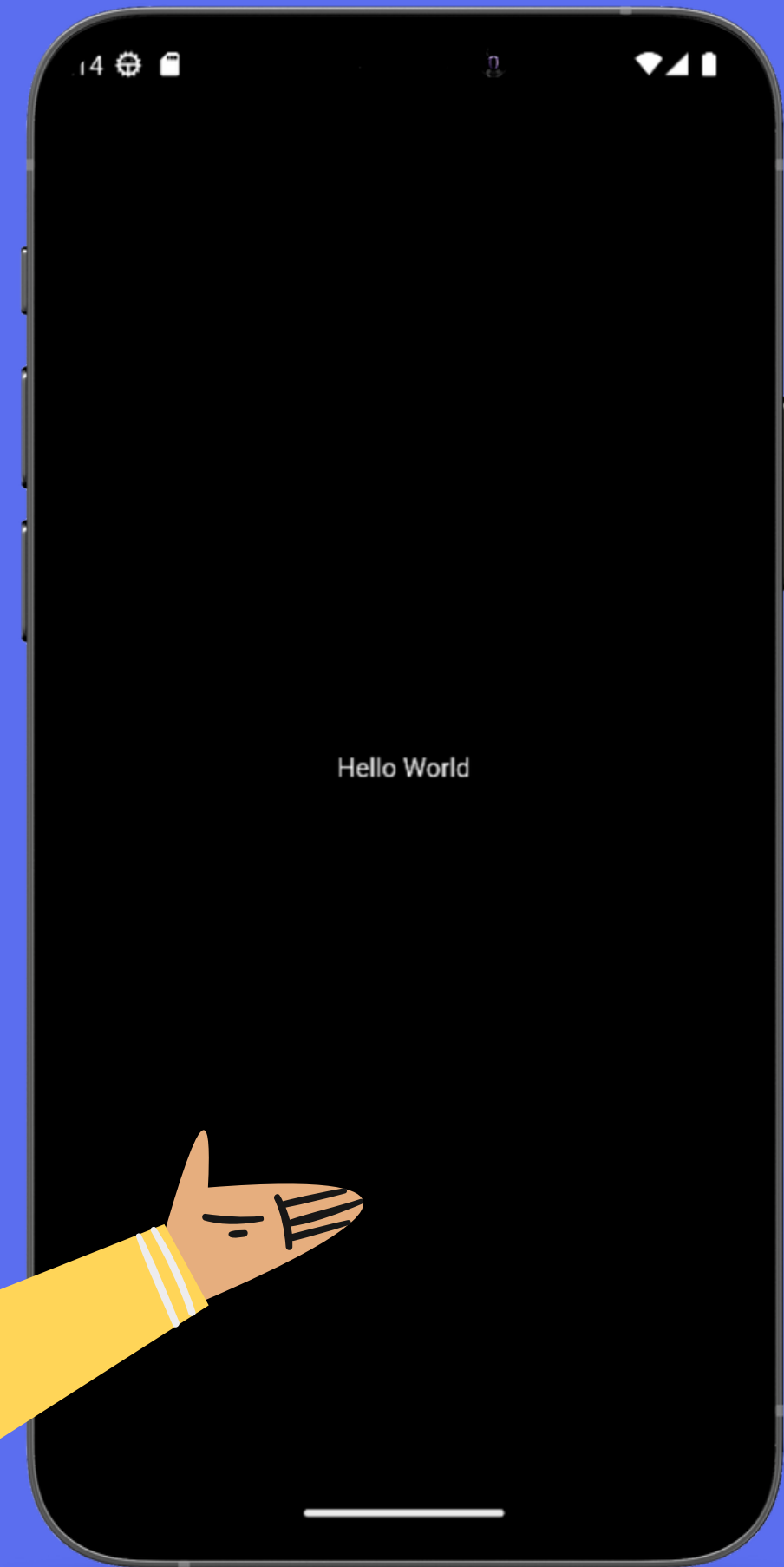
The widget tree



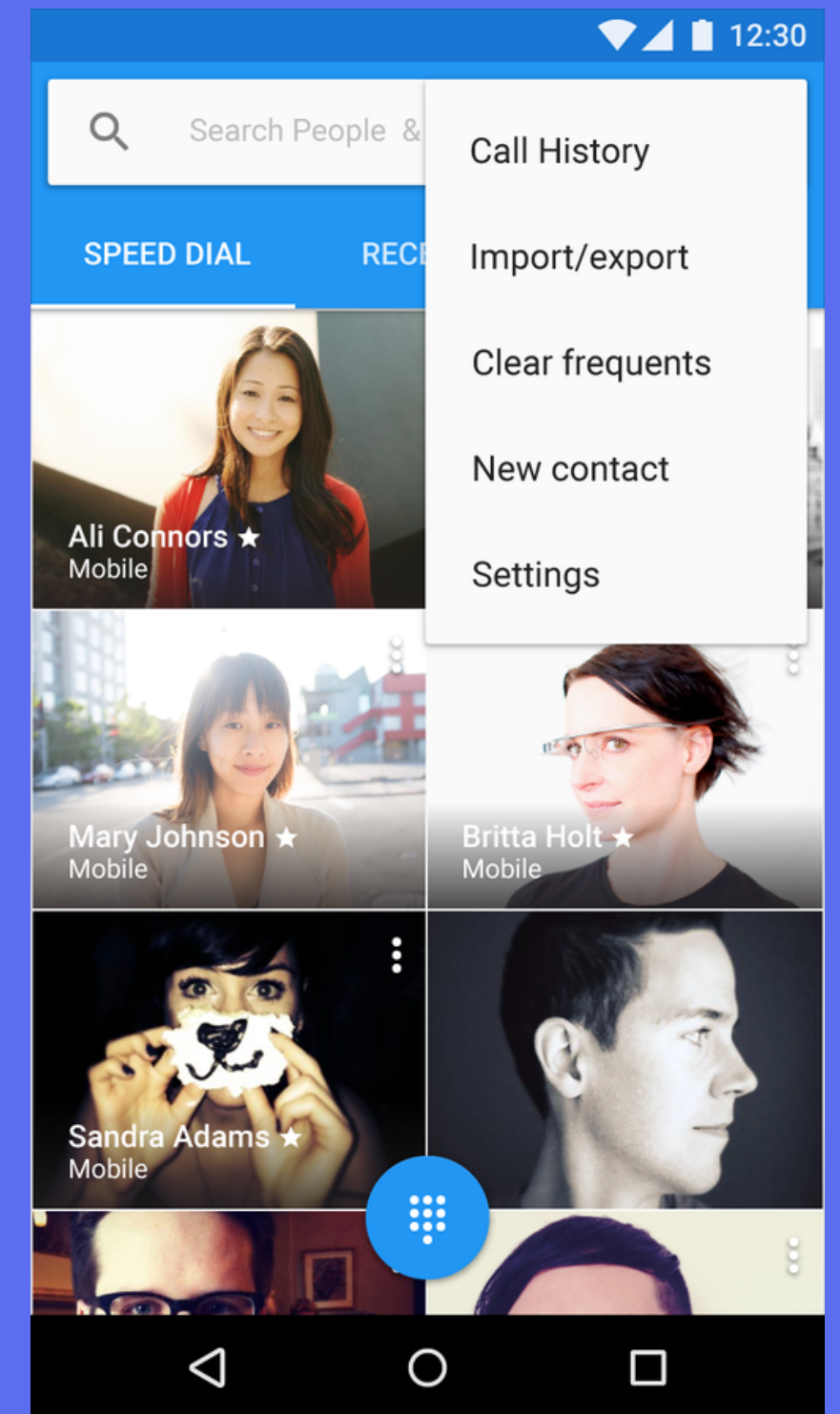
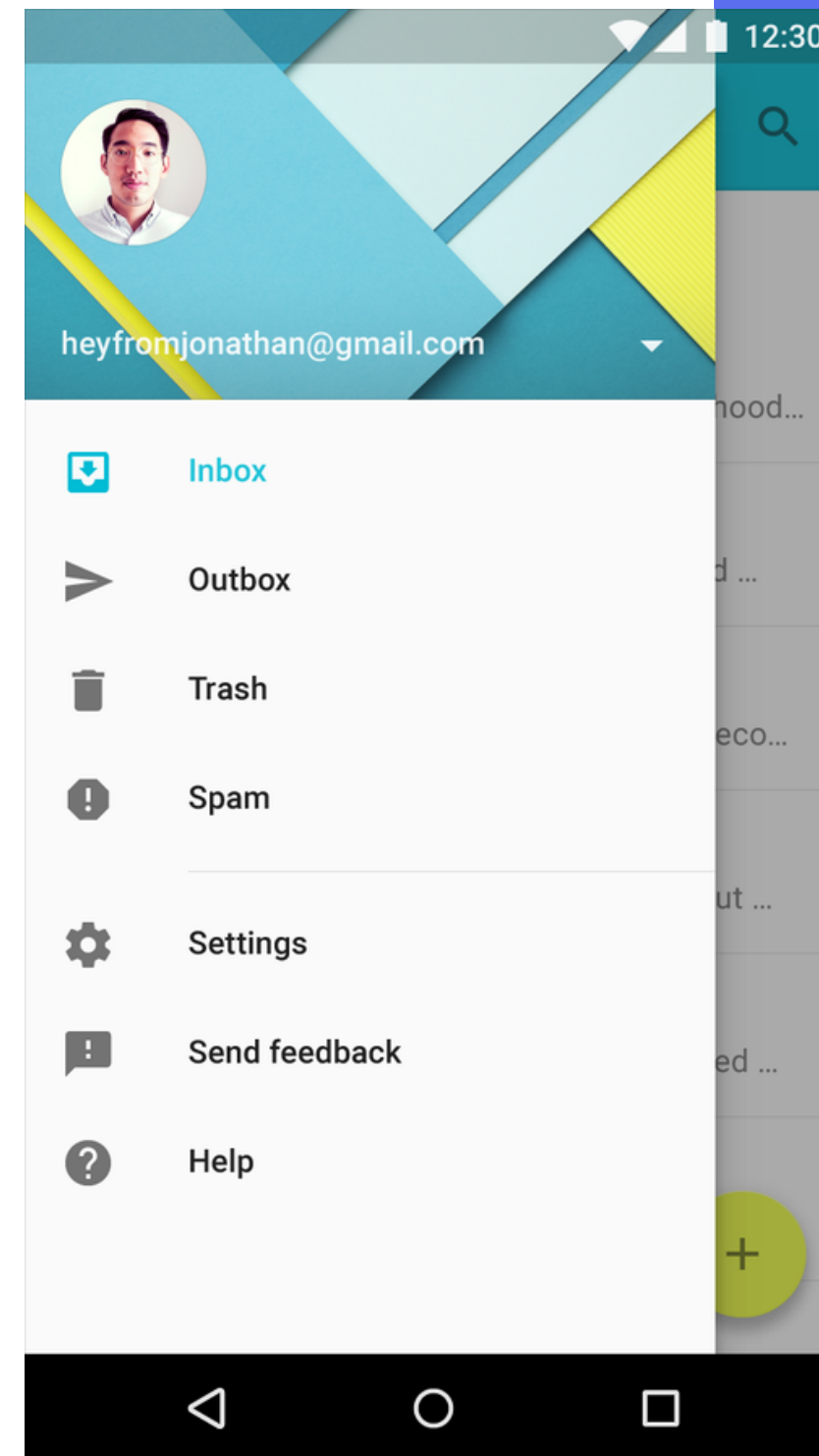


```
import 'package:flutter/material.dart';

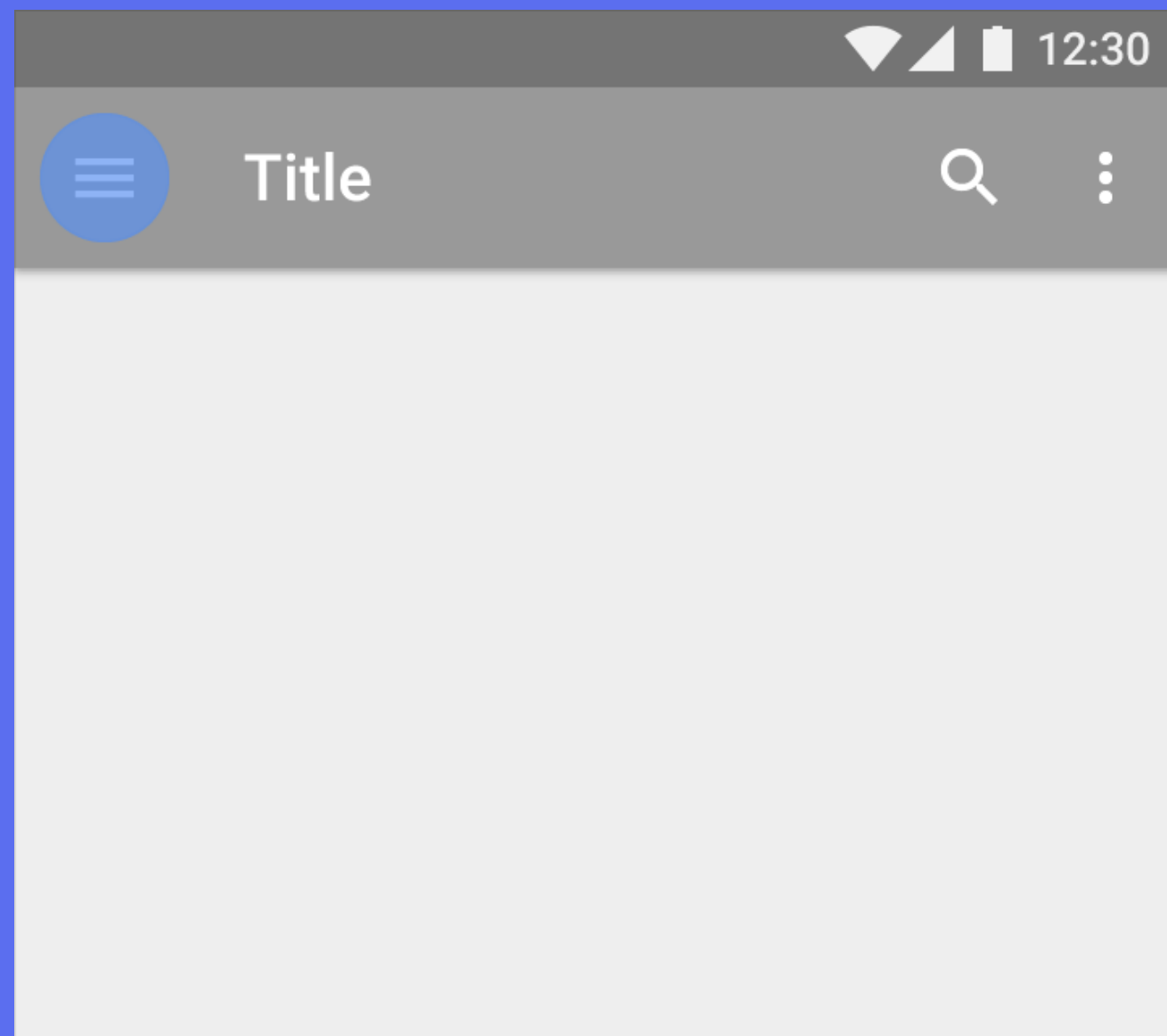
void main() {
  runApp(
    const Center(
      child: Text(
        'Hello, world!',
        textDirection: TextDirection.ltr,
      ),
    ),
  );
};
```



Material Design



O que é Material Design?



Material Design é um sistema de design criado e suportado por designers e desenvolvedores do Google. O Material.io inclui orientações detalhadas de UX e implementações de componentes de interface do usuário para Android, Flutter e Web.

Material.IO



MaterialApp Widget



- É um dos widgets mais poderosos de Flutter. Geralmente, começamos nossos aplicativos com o MaterialApp
- O MaterialApp envolve vários widgets que são comumente necessários para aplicar o Material Design no seu aplicativo.

MaterialApp Widget

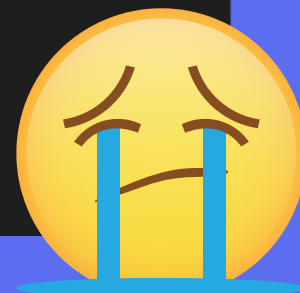


- Ao agrupar seu aplicativo dentro do MaterialApp, você está instruindo seu aplicativo a usar o Material Design do Android, que é um sistema de design criado pelo Google



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

void main() {
  runApp(
    MaterialApp(
      home: Center(
        child: Text(
          'Hello World',
        ),
      ),
    ),
  );
}
```



MaterialApp Docs



Scaffold

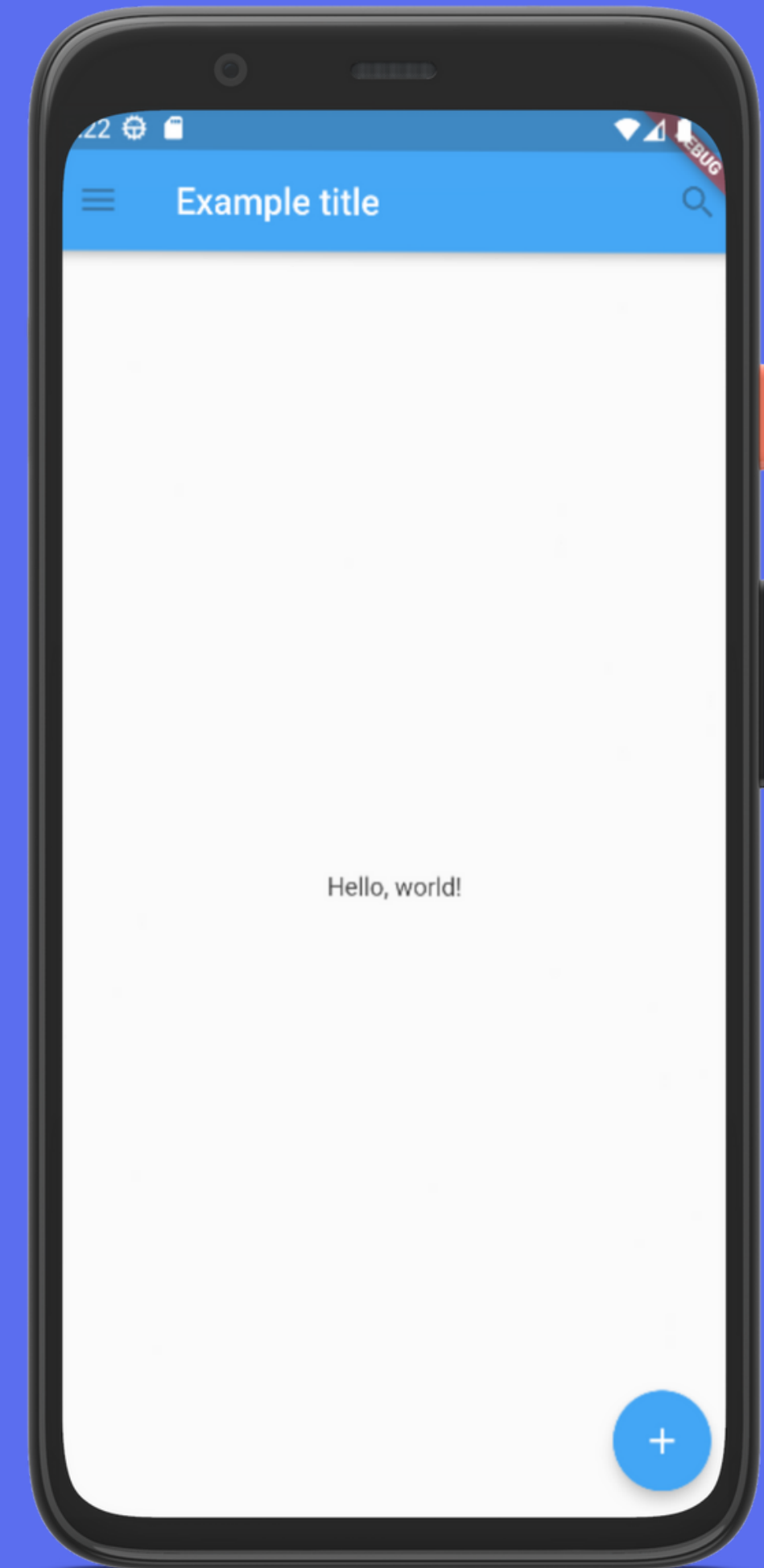
Widget de estrutura
de layout para o
Material Design



Scaffold Docs



```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      leading: const IconButton(
        icon: Icon(Icons.menu),
        tooltip: 'Navigation menu',
        onPressed: null,
      ),
      title: const Text('Example title'),
      actions: const [
        IconButton(
          icon: Icon(Icons.search),
          tooltip: 'Search',
          onPressed: null,
        ),
      ],
    ),
    // body is the majority of the screen.
    body: ..,
    floatingActionButton: const FloatingActionButton(
      tooltip: 'Add', // used by assistive technologies
      onPressed: null,
      child: Icon(Icons.add),
    ),
  );
}
```



Criando Widgets






Stateless Widget



1 - Defina um nome para seu widget.

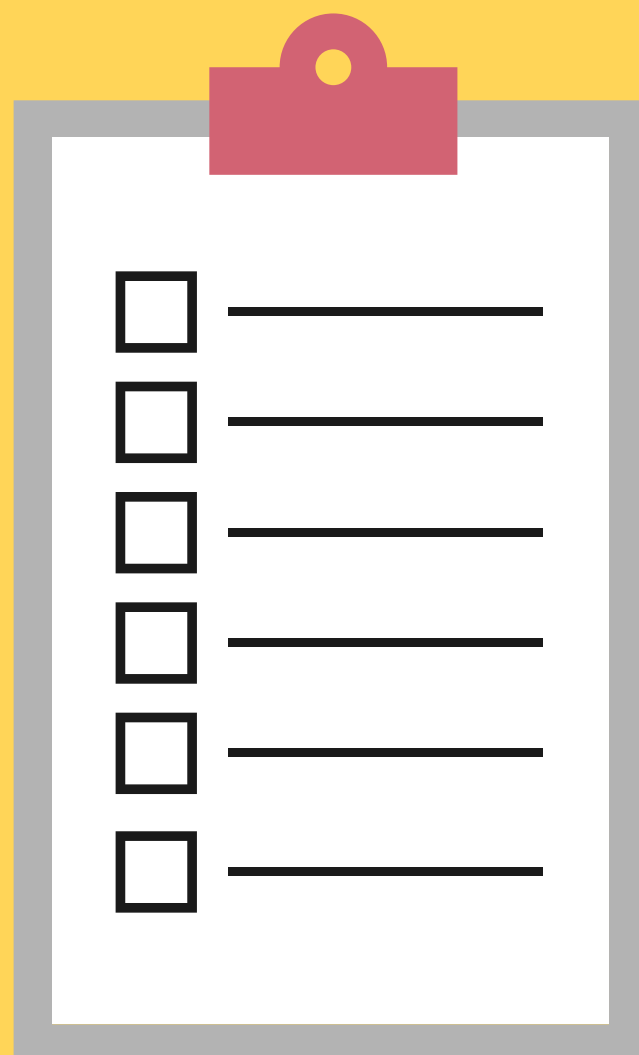
2 - herde sua classe de StatelessWidget.

3 - Implemente o método build() e retorne um Widget.



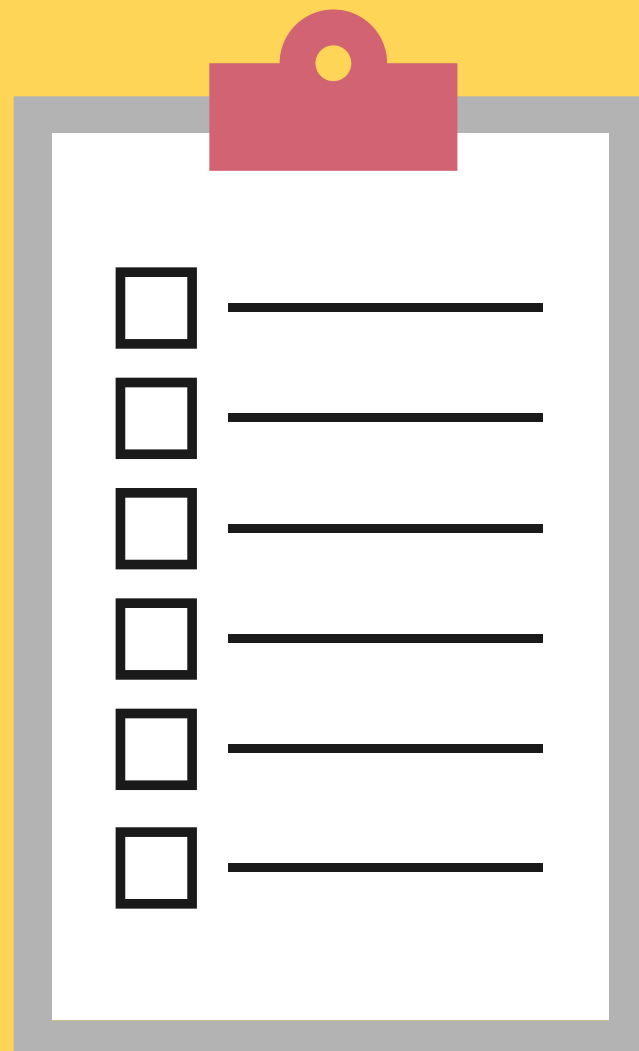
```
class MyStatelessWidget extends StatelessWidget {  
  
  @override  
  Widget build(BuildContext context) {  
    return Text('Hello!');  
  }  
}
```

O que aprendemos?



- **Widgets:** como o flutter utiliza widgets para compor interfaces.
- **MaterialApp:** utilização do Widget para iniciar apps com Material Design.
- **Scaffold:** utilização e configuração do widget scaffold para criação de layout inicial.

O que aprendemos?



- **Stateless Widgets:** Como criar seus próprios widgets (sem estado) a partir da composição de outros Widgets



**Obrigado pela
presença!**