Flutter

Curso Flutter de Verão

Objetivo da aula de hoje

- O Flutter
- Tipos de design
- Componentes
- Grupo de componentes
- Posicionamento de grupos

Flutter

Criado pela Google em 2015

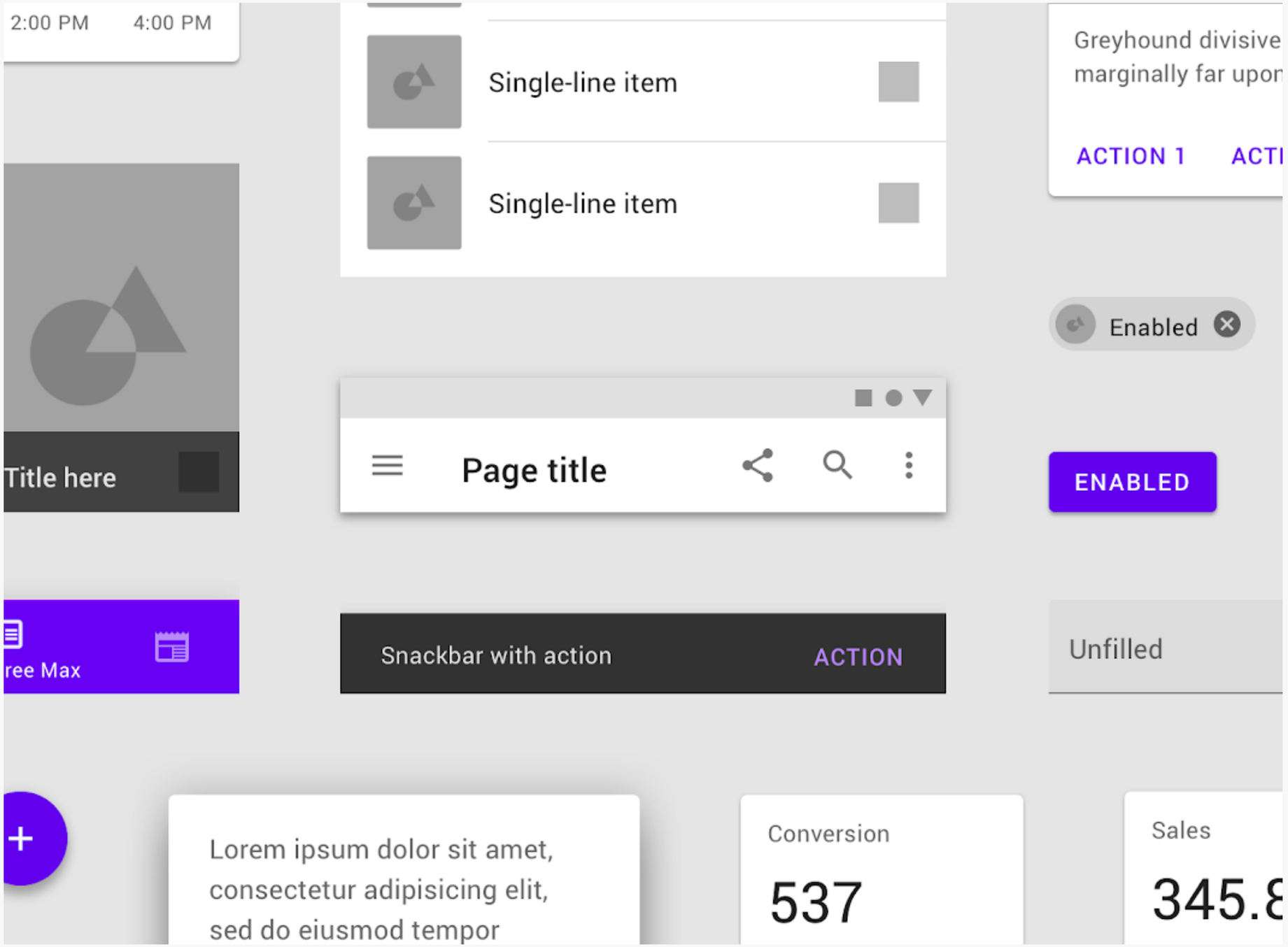
Possui modelo de componentes

Suporte a designs Material e Cupertino

Material Design

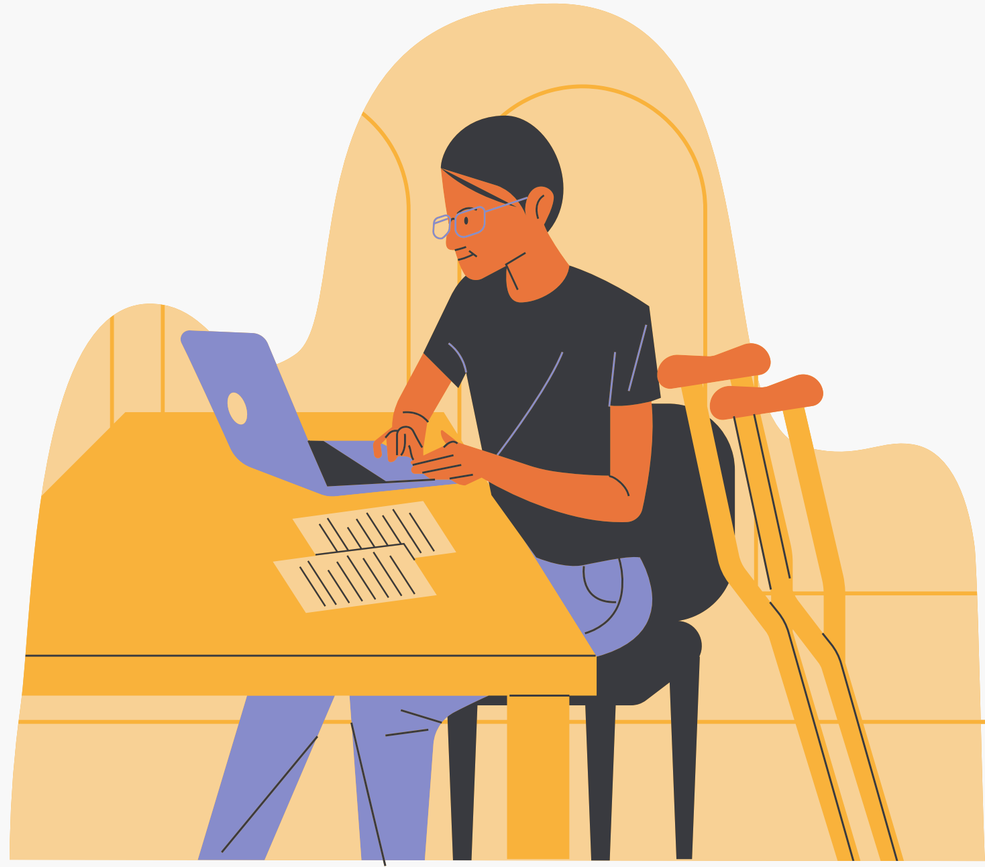
sistema de design criado e mantido por designers e desenvolvedores do Google

inclui orientações detalhadas de UX e implementações de componentes de interface do usuário para Android, Flutter e Web

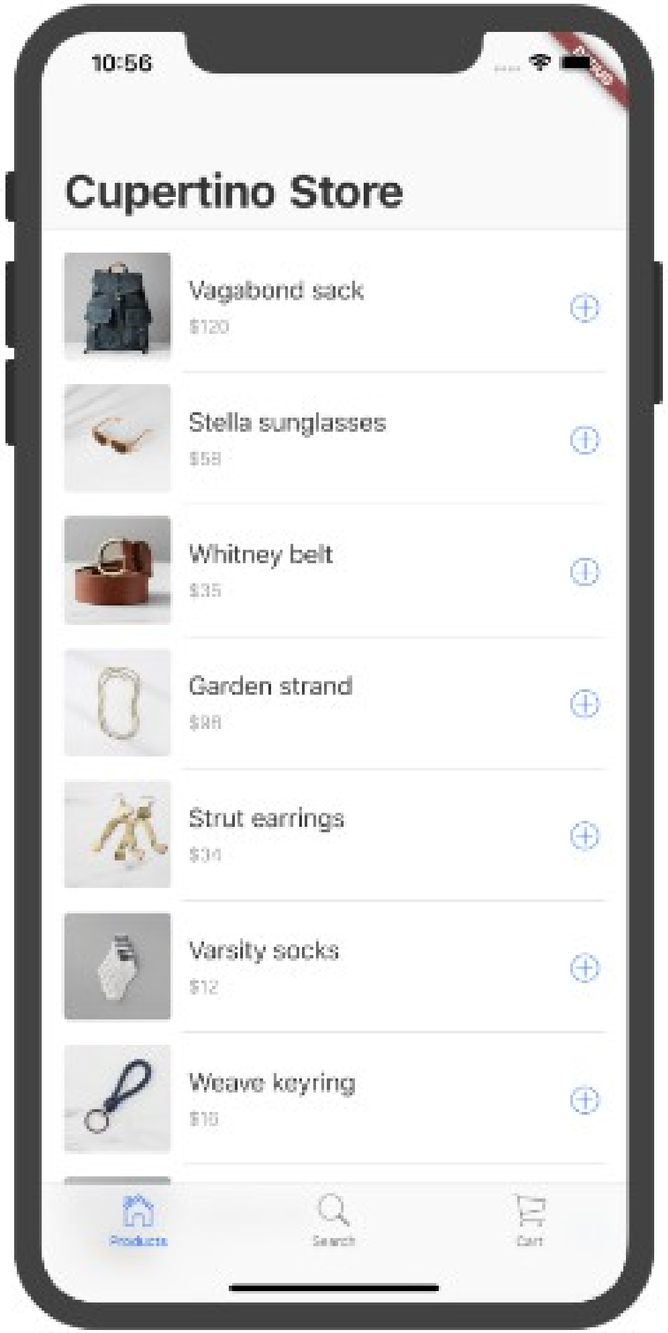

2:00 PM
4:00 PM
Single-line item
Single-line item
Greyhound divisive
marginally far upon
ACTION 1
ACTI
Enabled
Title here
Page title
ENABLED
ree Max
Snackbar with action
ACTION
Unfilled
Lorem ipsum dolor sit amet,
consectetur adipisicing elit,
sed do eiusmod tempor
Conversion
537
Sales
345.8

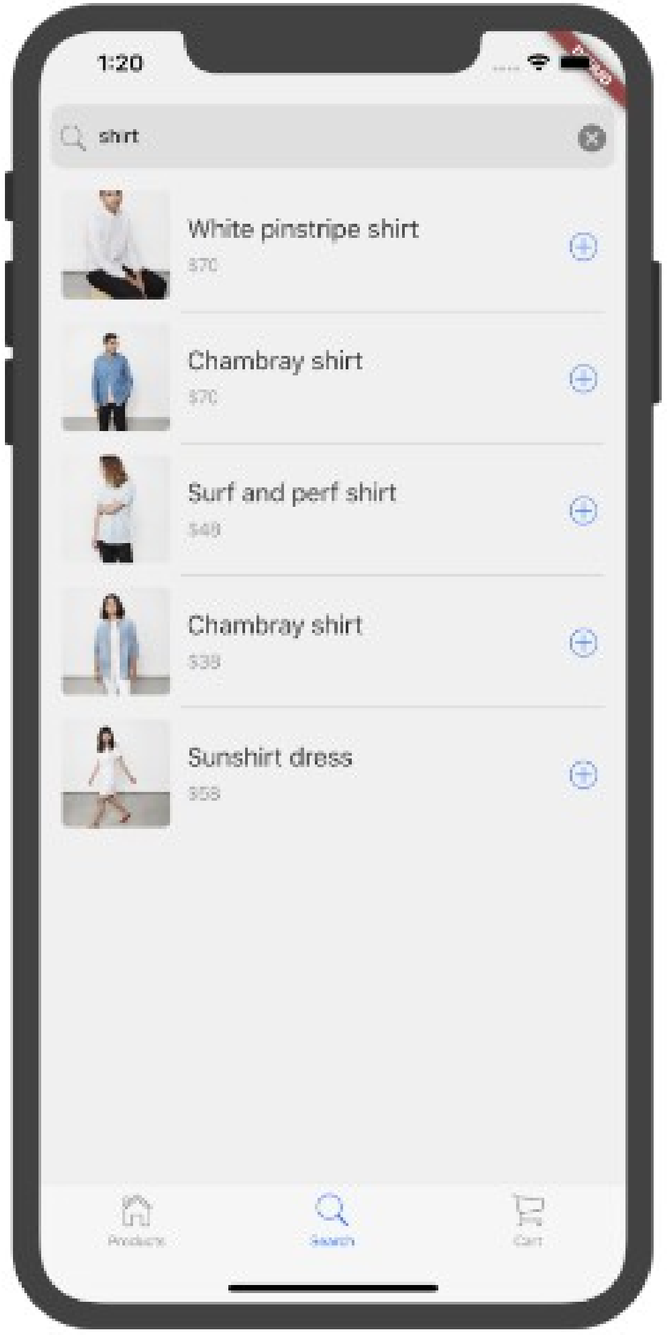
Cupertino Design

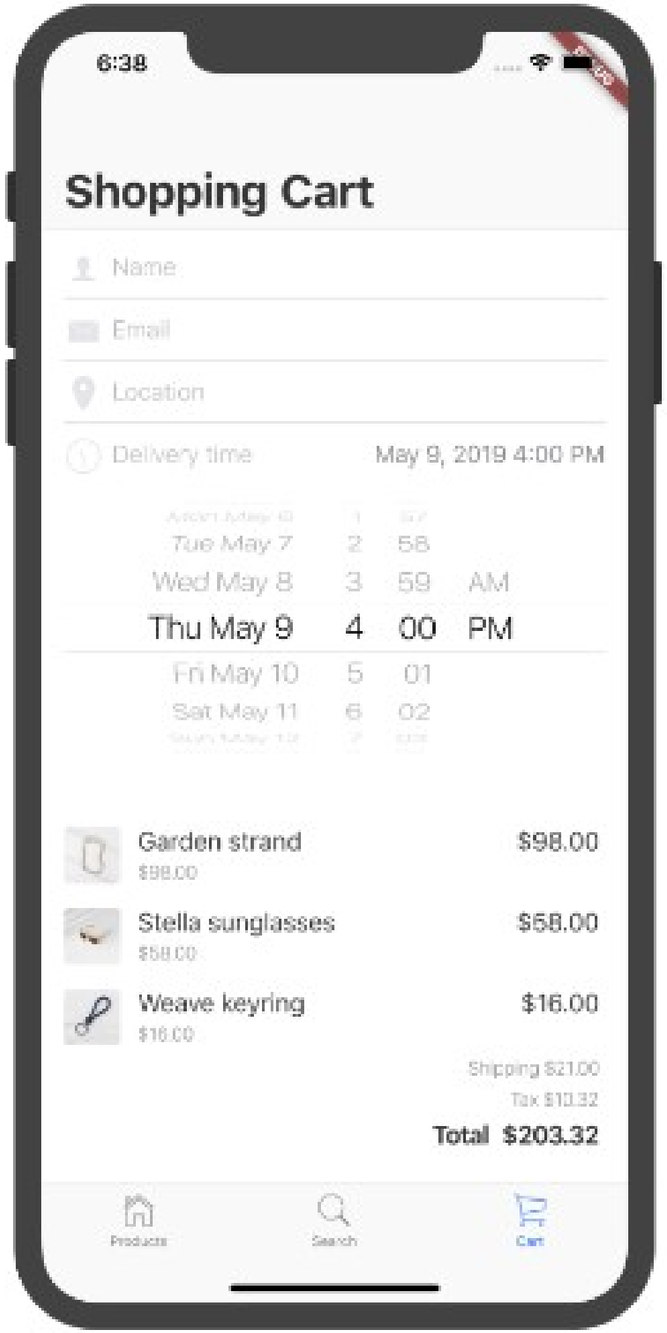
desenvolvido pela Apple e baseado nas Diretrizes de Interface Humana do iOS



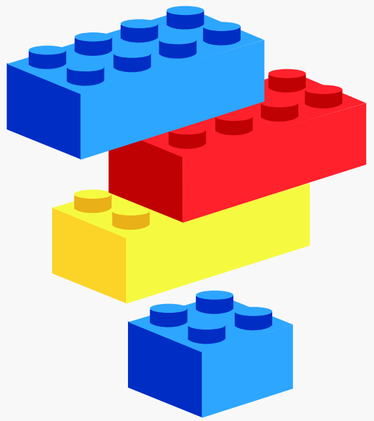
criado exclusivamente para ser utilizado em dispositivos iOS


10:56
Cupertino Store
Vagabond sack
$120
Stella sunglasses
$58
Whitney belt
$35
Garden strand
$98
Strut earrings
$34
Varsity socks
$12
Weave keyring
$16
Products
Search
Cart


1:20
shirt
White pinstripe shirt
$70
Chambray shirt
$70
Surf and perf shirt
$48
Chambray shirt
$39
Sunshirt dress
$58
Products
Search
Cart


6:38
Shopping Cart
Name
Email
Location
Delivery time
May 9, 2019 4:00 PM
Tue May 7 2 58
Wed May 8 3 59 AM
Thu May 9 4 00 PM
Fri May 10 5 01
Sat May 11 6 02
Garden strand
$98.00
$98.00
Stella sunglasses
$58.00
$58.00
Weave keyring
$16.00
$16.00
Shipping $21.00
Tax $10.32
Total $203.32
Products
Search
Cart

Componentes



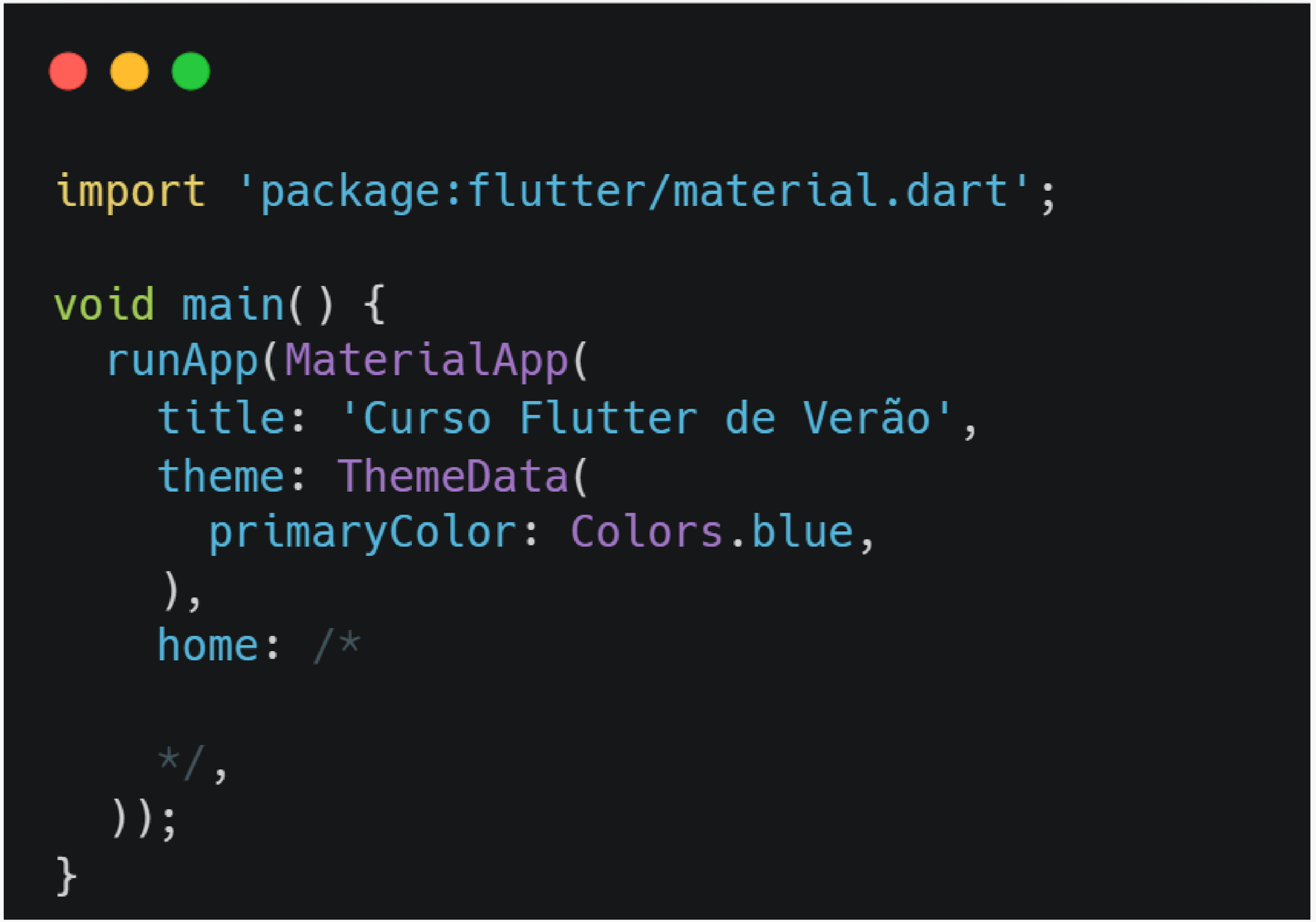
Constroem a interface do usuário por meio de composição

Podem ser compostos por um ou mais componentes

Descrevem como deve ser a visualização de tela, considerando sua configuração e seu estado atual

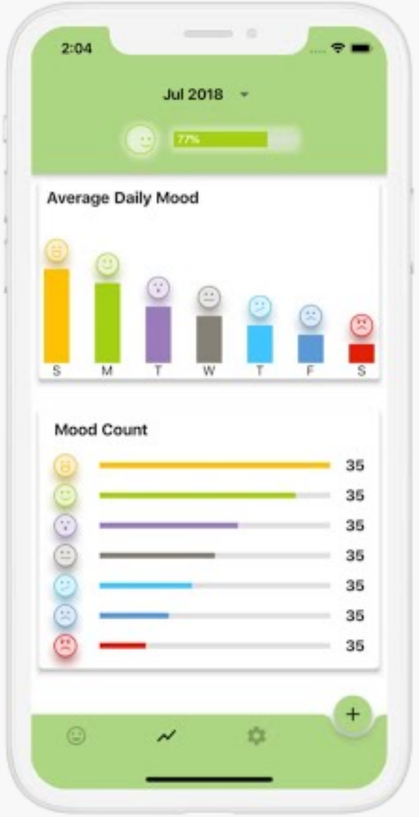

void main() {
}

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

void main() {
  runApp(/*

  )*/);
}
```


import 'package:flutter/material.dart';
void main() {
runApp(MaterialApp(
title: 'Curso Flutter de Verão',
theme: ThemeData(
primaryColor: Colors.blue,
),
home: /*
*/,
));
}

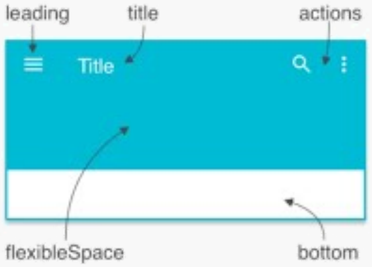

import 'package:flutter/material.dart';
void main() {
runApp(MaterialApp(
title: 'Curso Flutter de Verão',
theme: ThemeData(
primaryColor: Colors.blue,
),
home: Scaffold(
body: /* ... */,
),
));
}

Scaffold



UI / UX

AppBar



Text

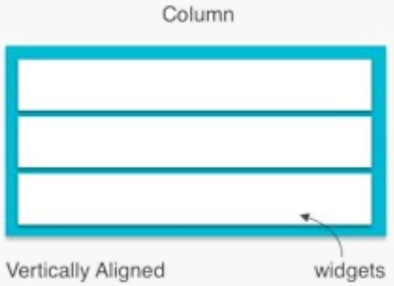
onPanUpdate:
DragUpdateDetails(Offset(0.3, 0.0))

RichText

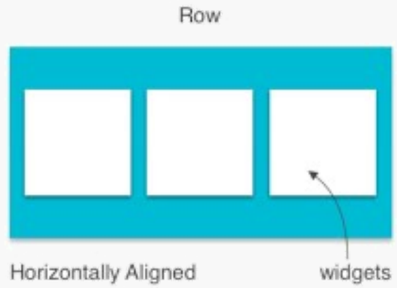
Flutter World for **Mobile**

SafeArea

Column



Row



Container

Button

Barista
Default

Barista
StadiumBorder

Barista
UnderlineInputBorder

Barista
OutlineInputBorder

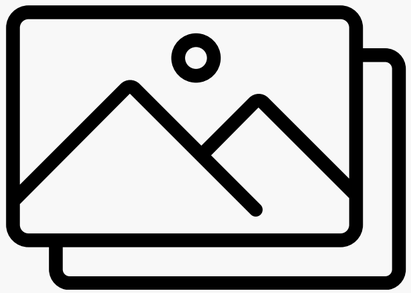
Textos

s the ground or stays
iverse is vast, and you
s also beautiful. You a
nething bigger than yo
t of something that ma
most of your time. Tak
e a blog post. Make a

```dart
Text(
  'Olá, mundo!',
  style: const TextStyle(
    color: Colors.black,
    fontSize: 24,
    fontWeight: FontWeight.bold,
    fontStyle: FontStyle.italic,
    decoration: TextDecoration.underline,
  ),
);
```

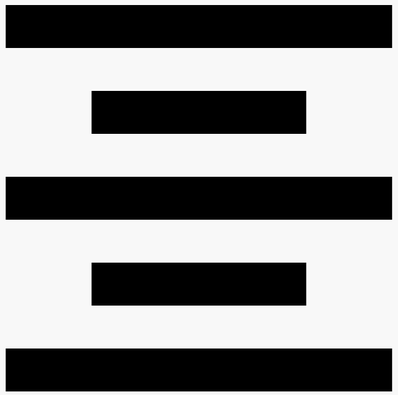
Ícones

https://fonts.google.com/icons



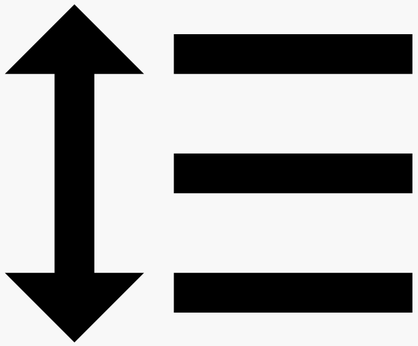
```
Icon(
  Icons.add,
  color: Colors.red,
  size: 42,
);
```

Alinhamento



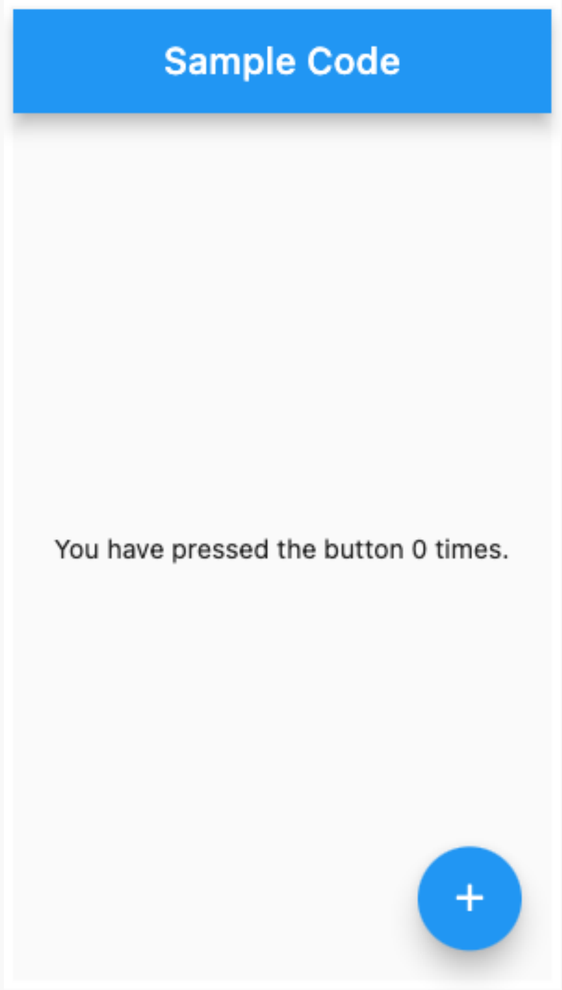
```
Alignment(
  align: Alignment.topLeft,
  // [top|center|bottom][Left|Center|Right]
  child: /* ... */,
);

Center(
  child: /* ... */,
);
```

Margens



```dart
Padding(
  padding: EdgeInsets.all(10),
  // [all|only|symmetric]
  // only: [top,bottom,left,right]
  // symmetric: [vertical,horizontal]
  child: /* ... */,
),
```

Tela básica




```
Scaffold(
  appBar: AppBar(title: Text(/* ... */)),
  body: /* ... */,
  floatingActionButton: FloatingActionButton(
    child: /* ... */,
  ),
);
```

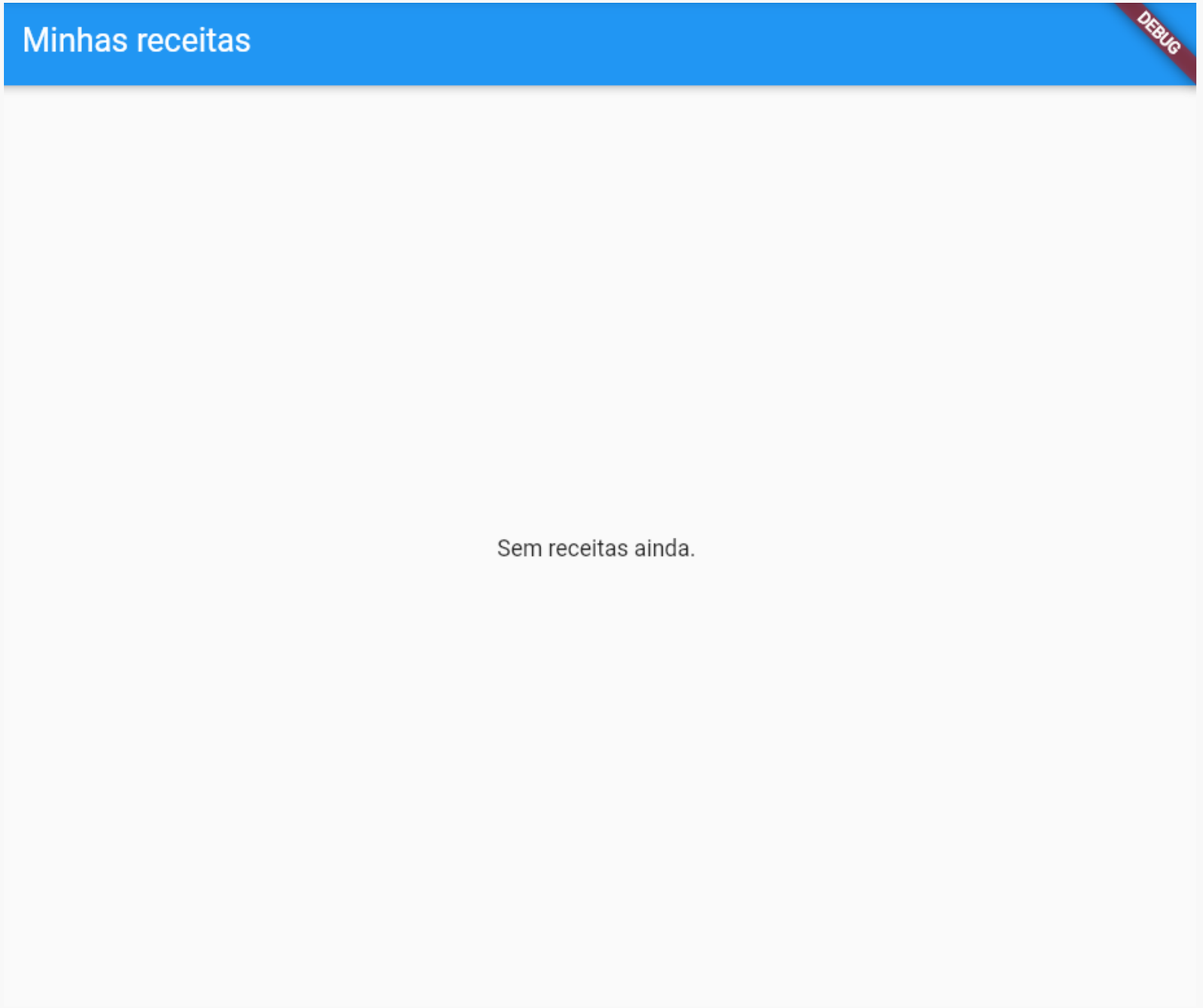
Exercícios

Você foi contratado para desenvolver um aplicativo de receitas para uma editora. O aplicativo deve exibir diferentes categorias de receitas em uma tela inicial. Cada receita deve ser exibida contendo seu nome.

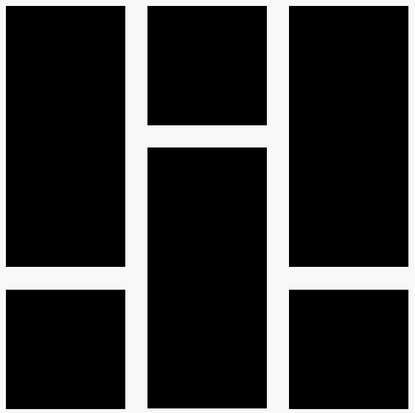
1) Para iniciar o desenvolvimento do aplicativo, você precisa criar a sua estrutura básica. Exiba uma versão mínima contendo o texto "Sem receitas ainda." centralizado na tela.

Futuramente, você deve adicionar botões de ações no canto superior direito da tela, como ajuda e sair do aplicativo. Adicione também uma barra de tarefas superior contendo, no momento, apenas o título "Minhas receitas".

Verifique se o layout é exibido corretamente na tela.

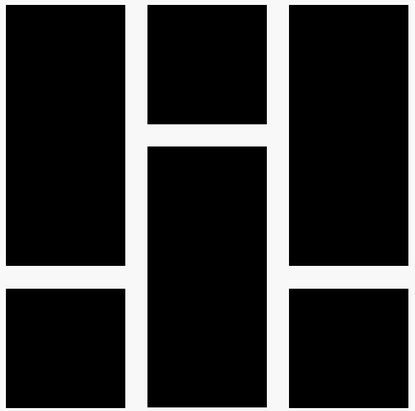

Minhas receitas
DEBUG
Sem receitas ainda.

Listas de widgets



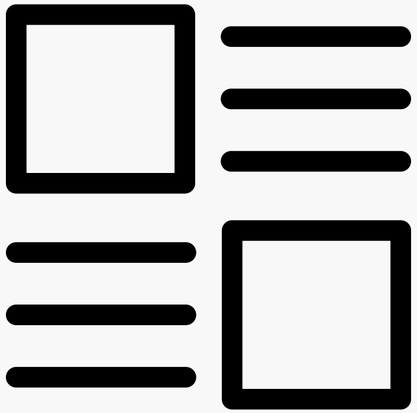
```
Flex(
  direction: Axis.horizontal,
  // [horizontal|vertical]
  children: [
    /* ... */,
    /* ... */,
    /* ... */,
  ],
);
```

Listas de widgets



```
Column(
  children: [ /* ... */ ],
);

Row(
  children: [ /* ... */ ],
);
```

Posicio namento

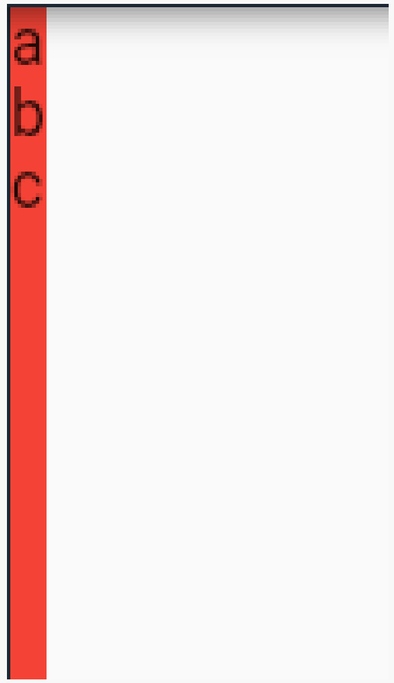


```
Column(
  mainAxisAlignment: MainAxisAlignment.start,
  // [start|end|center|space[Between|Around|Evenly]]
  mainAxisSize: MainAxisSize.max,
  // [max|min]
  crossAxisAlignment: CrossAxisAlignment.center,
  // [start|end|center|stretch|baseline]
  children: [
    /* ... */,
    /* ... */,
    /* ... */,
  ],
);
```

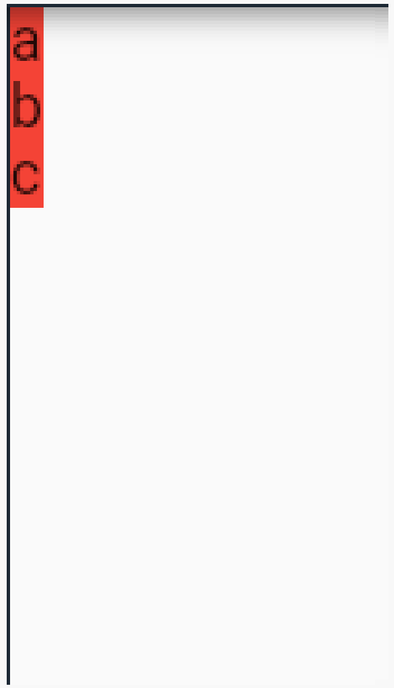
MainAxisSize

Column

MainAxisSize.max



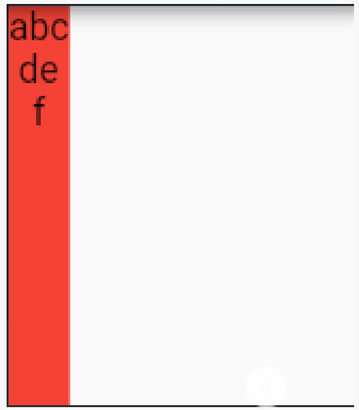
MainAxisSize.min



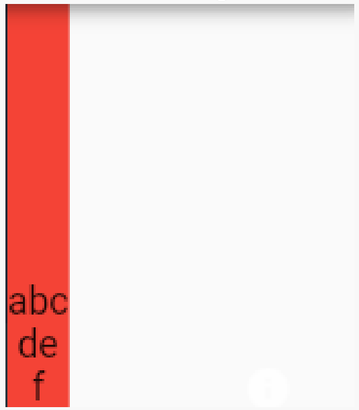
MainAxisAlignment

Column

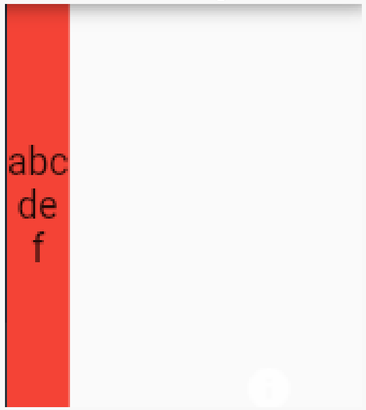
MainAxisAlignment.start



MainAxisAlignment.end



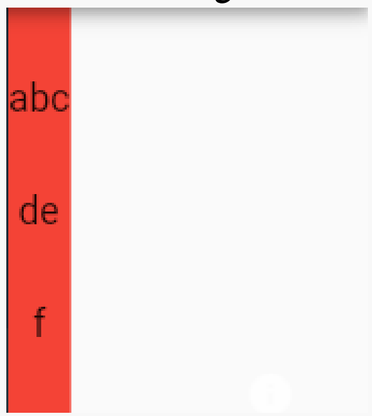
MainAxisAlignment.center



MainAxisAlignment.spaceAround

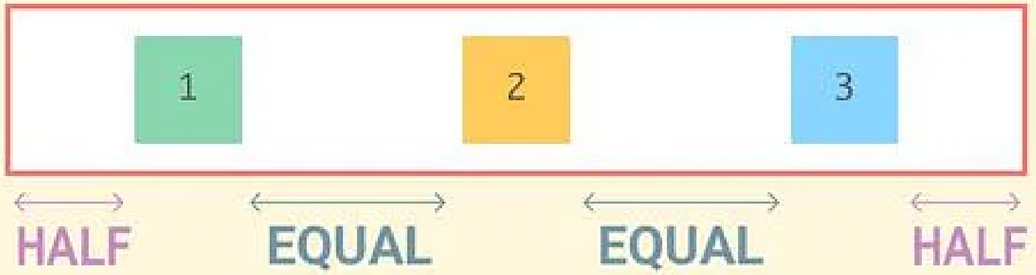
MainAxisAlignment.spaceBetween

MainAxisAlignment.spaceEvenly

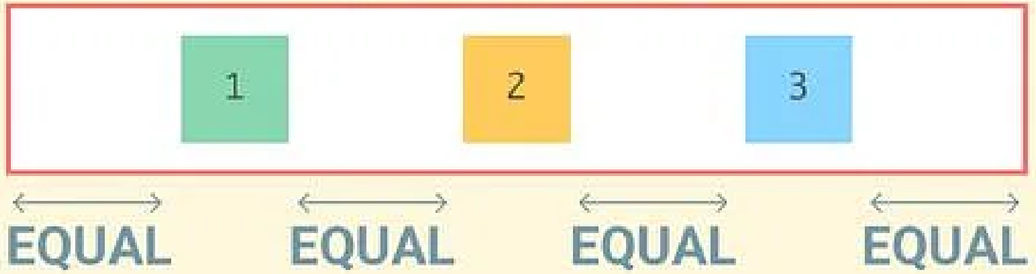


space-around vs space-evenly

space-around



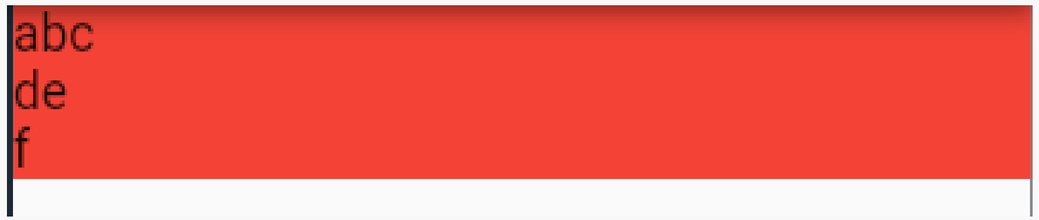
space-evenly



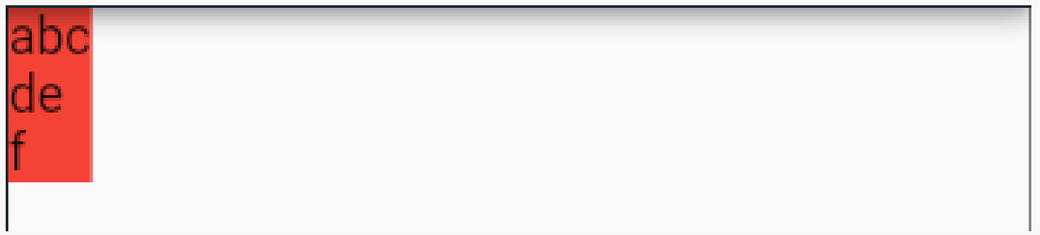
CrossAxisAlignment

Column

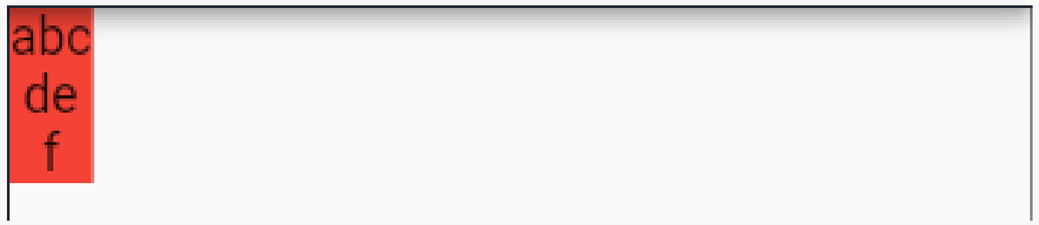
CrossAxisAlignment.stretch



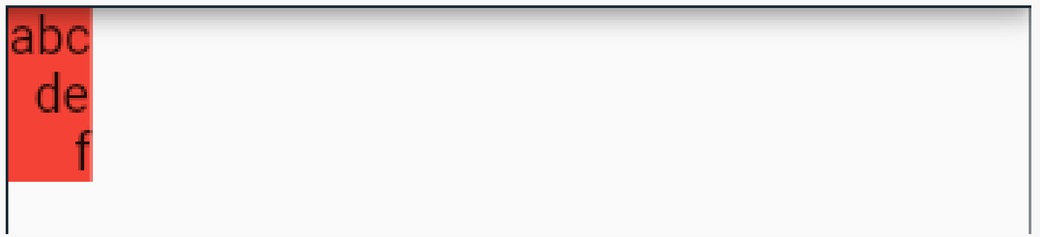
CrossAxisAlignment.start



CrossAxisAlignment.center



CrossAxisAlignment.end

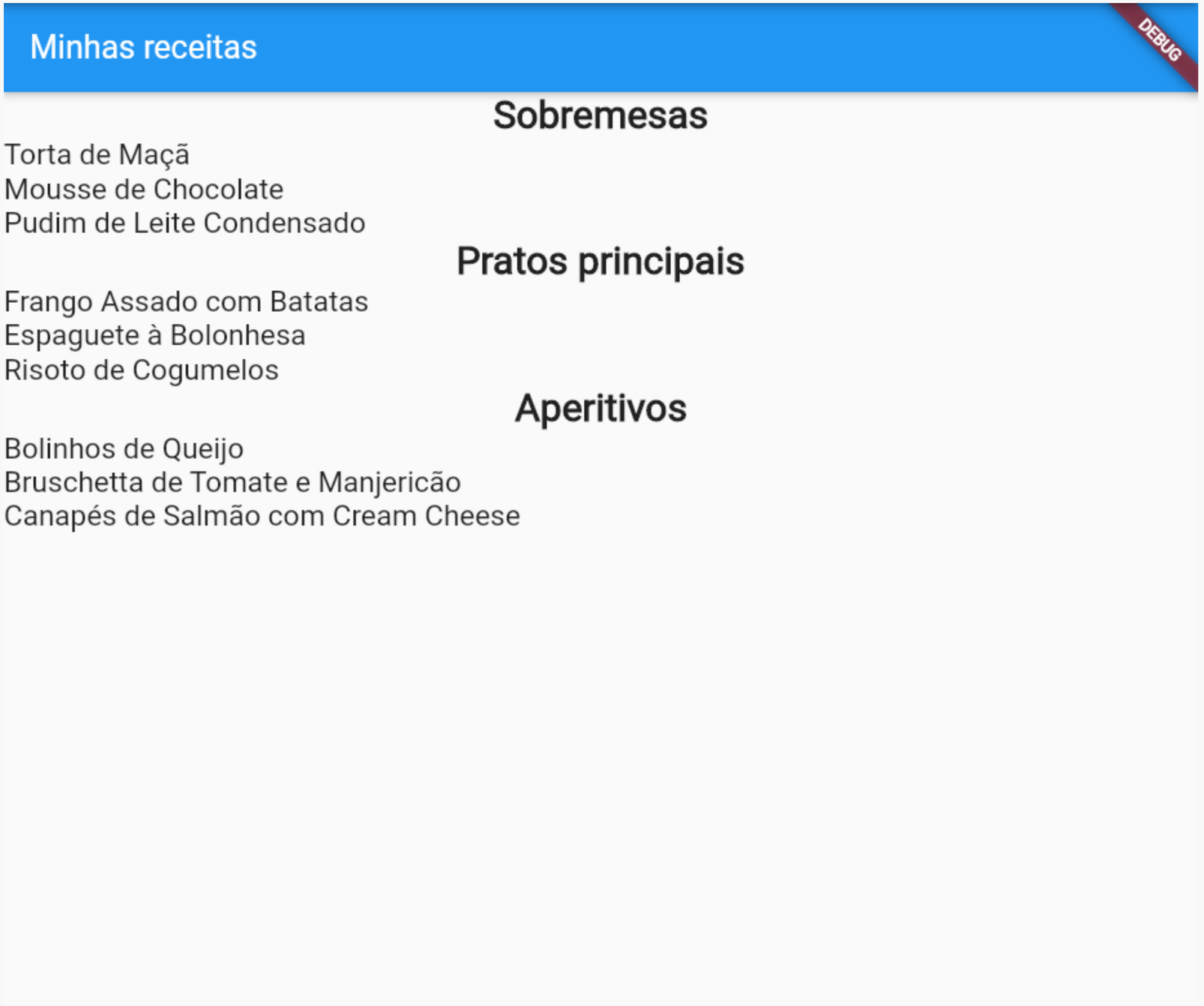


Exercícios

2) O conteúdo desta tela deve conter textos representando diferentes categorias de receitas, como "Sobremesas", "Pratos principais" e "Aperitivos". Adicione estes itens verticalmente ao longo da tela. Estes textos deverão ser formatados como títulos, portanto formate-os com negrito e com fonte 24. Por fim, centralize cada texto na tela.

Embaixo de cada título, deverão conter três textos com o nome das receitas. Estes textos deverão ser formatados como subtítulo, portanto formate-os com fonte 18. Os textos não deverão ficar centralizados na tela.

Verifique se o layout é exibido corretamente na tela.


Minhas receitas
DEBUG
Sobremesas
Torta de Maçã
Mousse de Chocolate
Pudim de Leite Condensado
Pratos principais
Frango Assado com Batatas
Espaguete à Bolonhesa
Risoto de Cogumelos
Aperitivos
Bolinhos de Queijo
Bruschetta de Tomate e Manjericão
Canapés de Salmão com Cream Cheese

3) Seu chefe pediu para que houvesse suporte para ler os dados de um site. Para iniciar a implementação, você deverá ler os itens de um mapa, de chaves String e valores List<String>.

Crie, dentro da função main mas antes da chamada da função runApp, um mapa chamado dados com a mesma estrutura à direita.

Verifique se o layout é exibido corretamente na tela.

```
final Map<String, List<String>> dados = {
  'Sobremesas': [
    'Torta de Maçã',
    'Mousse de Chocolate',
    'Pudim de Leite Condensado',
  ],
  'Pratos principais': [
    'Frango Assado com Batatas',
    'Espaguete à Bolonhesa',
    'Risoto de Cogumelos',
  ],
  'Aperitivos': [
    'Bolinhos de Queijo',
    'Bruschetta de Tomate e Manjericão',
    'Canapés de Salmão com Cream Cheese',
  ],
};
```

4) Seu chefe pediu em seguida que houvesse suporte para filtros, onde o usuário poderia escolher qual categoria ele gostaria de ver apenas. Para iniciar a implementação, além de ler os itens de um mapa, você deverá ler um int de 1 a 3, que pode ser nulo. Caso este número seja nulo, todas as categorias deverão ser exibidas. Caso contrário, apenas a categoria indicada pelo seu valor deverá ser exibida (se 1, apenas sobremesas; se 2, apenas pratos principais; se 3, apenas aperitivos).

Crie, dentro da função main mas antes da chamada da função runApp, um inteiro nulável chamado categoria com um valor arbitrário.

```
final int? categoria /* = null, 1, 2 ou 3 */;
```

Verifique se o layout é exibido corretamente na tela.

5) Para finalizar, seu chefe pediu que houvesse suporte para buscas por texto, onde o usuário poderia inserir um texto I e serem exibidas apenas as receitas cujos nomes contém I como substring. Para iniciar a implementação, além de ler os itens de um mapa e o filtro de um inteiro, você deverá ler uma String qualquer. Caso seu valor seja vazio, todas as receitas deverão ser exibidas. Caso contrário, apenas as receitas cujos nomes contém seu valor deverão ser exibidas. Caso nenhuma receita se encaixe nessa condição, exiba o texto "Nenhuma receita encontrada" centralizado na tela.

Crie, dentro da função main mas antes da chamada da função runApp, um texto chamado busca com um valor arbitrário.

```
final String busca = 'queijo';
```

Verifique se o layout é exibido corretamente na tela.

Flutter

Curso Flutter de Verão