

# Flutter

Aula 05

Prof. Dr. Rodrigo Plotze

[rodrigoplotze@gmail.com](mailto:rodrigoplotze@gmail.com)



- Build Widgets e Build Method
- Stateless Widgets
- Codificação de Widgets em arquivos separados
- Assets e o arquivo pubspecs.yaml
- Barra de Rolagem com SingleChildScrollView
- Adicionar imagens com Image.asset
- Espaçamento com o widget Padding
- Decoração do widget Container com BoxDecoration
- Atividade Prática

# **APP**

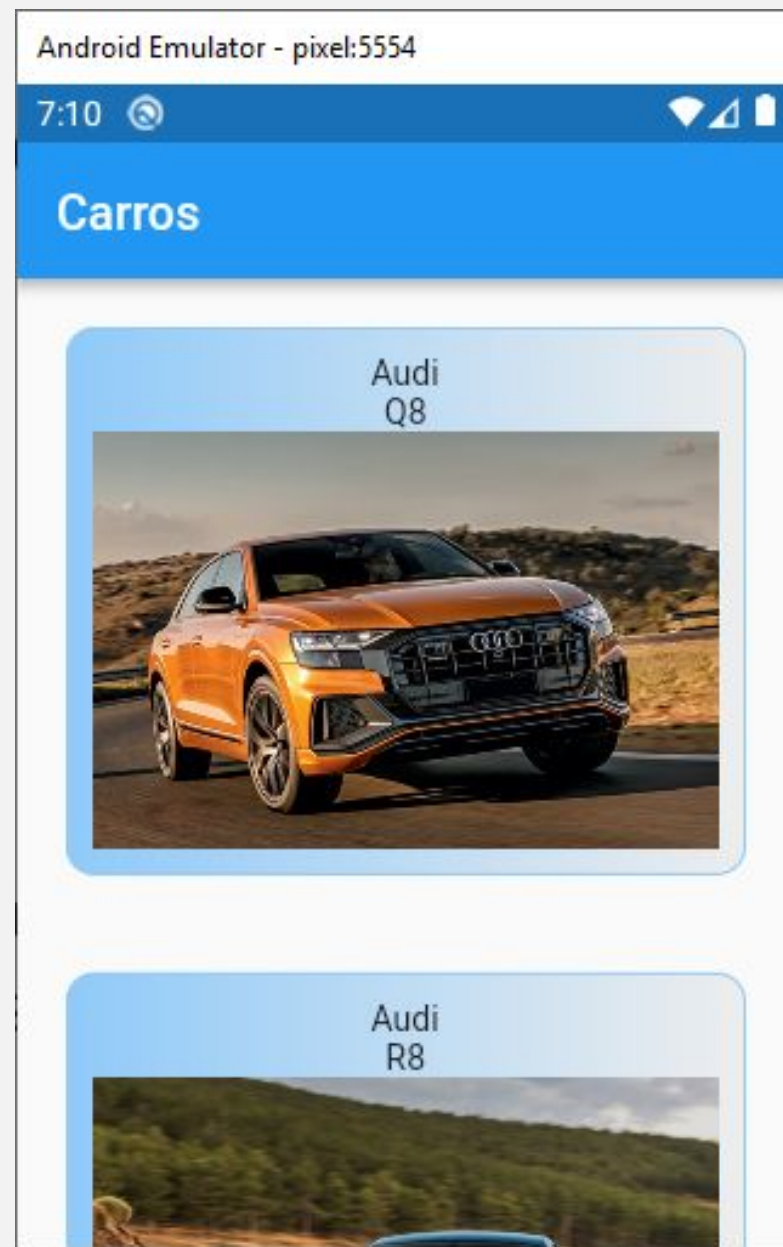
## **LISTAGEM DE VEÍCULOS**



# App para Listagem de Veículos



- Objetivo
  - Construir um App para listagem de marcas, modelos e fotos de veículos.






- Flutter não renderiza automaticamente os widgets da App.
- Para isso é necessário escrever um método denominado ***build***

```
class MyApp extends StatelessWidget {  
  // This widget is the root of your application.  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Flutter Demo',  
      home: Center(child: Text('Hello World')));  
  }  
}
```



- O método ***build*** possui um parâmetro denominado ***BuildContext*** que indica a localização na ***Widget Tree***.

```
class MyApp extends StatelessWidget {  
  // This widget is the root of the application.  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Flutter Demo',  
      home: Center(child: Text('Hello World')) );  
  }  
}
```





- As interfaces gráficas das App são formadas por uma série de Widgets que não são ***“inteligentes”*** ou que não ***“interagem com o usuário”***.
- Em outras palavras, existem widgets que são utilizados para apresentar alguma coisa e não interagem com o usuário.
- Estes widgets são denominados ***Stateless Widgets***.

```
class EmptyWidget extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return <Insert Some Widgets Here>;  
  }  
}
```



- A aparência de um Stateless Widget é gerada a partir do método ***build***, em que o resultado é renderizado pelo Flutter.
- O método pode utilizar valores das variáveis que fazem parte da classe (***Member Variables***) ou utilizar recursos herdados da árvores de Widgets.





- Exemplo de uso variáveis no widget

```
@override
Widget build(BuildContext context) {
  return Center(
    child: Column(children: <Widget>[
      Text(make),
      Text(model),
      Image.network(imageSrc)
    ]));
```

- Exemplo de uso de recursos

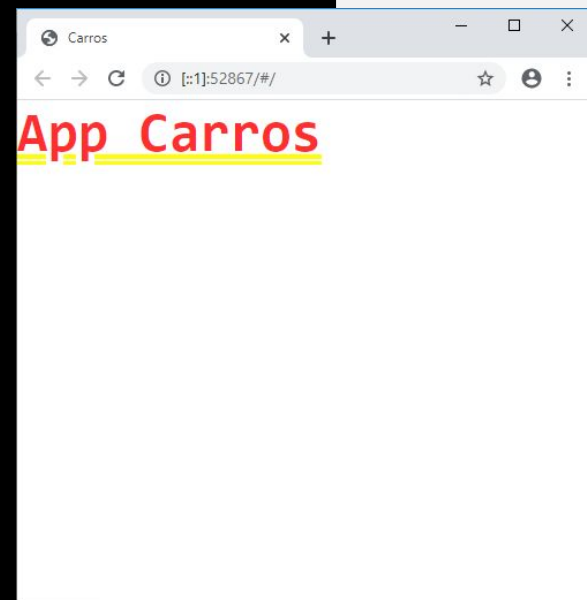
```
@override
Widget build(BuildContext context) {
  return Center(
    child: Column(children: <Widget>[
      Text("Hello", style: Theme.of(context).textTheme.display1),
      Text("There", style: Theme.of(context).textTheme.display1) ]));
}
```



- Crie um novo Flutter Project denominado ***app\_carros*** e adicione o seguinte código:

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

void main() => runApp(App());
class App extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Carros',
      theme: new ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: Text("App Carros"),
    );
  }
}
```





- Na pasta **lib** adicione um novo arquivo denominado **widget\_carro.dart**

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

class CarroWidget extends StatelessWidget {

  //atributos
  final String marca;
  final String modelo;
  final String foto;

  // construtor
  CarroWidget(this.marca, this.modelo, this.foto): super();

  @override
  Widget build(BuildContext context) {
    return new Center(
      child:
        Column(children: <Widget>[
          Text(marca),
          Text(modelo),
          Image.asset(foto),
        ],),
    );
  }
}
```



main.dart  
tela\_listacarros.dart  
widget\_carro.dart

Na pasta **lib** adicione um novo arquivo denominado **tela\_listacarros.dart**

**Key** é um parâmetro opcional utilizado para preservar o estado na widget tree.

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

class ListaCarros extends StatelessWidget {

  final String titulo;

  ListaCarros({Key key, this.titulo}): super(key:key);

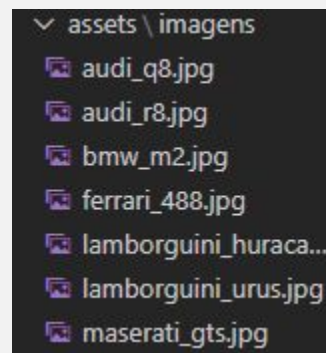
  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      appBar: new AppBar(title: new Text(this.titulo)),
      body: new Column(children: <Widget>[
        new CarroWidget("Audi", "Q8", "assets/imagens/audi_q8.jpg"),
        new CarroWidget("Audi", "R8", "assets/imagens/audi_r8.jpg"),
        new CarroWidget("BMW", "M2", "assets/imagens/bmw_m2.jpg"),
        new CarroWidget("Ferrari", "488", "assets/imagens/ferrari_488.jpg"),
        new CarroWidget("Lamborghini", "Huracan", "assets/imagens/lamborghini_huracan.jpg"),
        new CarroWidget("Lamborghini", "Urus", "assets/imagens/lamborghini_urus.jpg"),
        new CarroWidget("Maserati", "GTS", "assets/imagens/maserati_gts.jpg"),
      ],),
    );
  }
}
```



Adicione a estrutura do projeto uma pasta denominada ***assets***

Flutter apps can include both code and *assets* (sometimes called resources). An asset is a file that is bundled and deployed with your app, and is accessible at runtime. Common types of assets include static data (for example, JSON files), configuration files, icons, and images (JPEG, WebP, GIF, animated WebP/GIF, PNG, BMP, and WBMP).

Adicione na pasta ***assets*** uma subpasta denominada ***imagens*** e cole as imagens de carros disponíveis.




Indique o caminho da pasta de imagens no arquivo de dependências ***pubspecs.yaml***

```
# The following section is specific to Flutter.
flutter:

  # The following line ensures that the Material Icons font is
  # included with your application, so that you can use the icons in
  # the material Icons class.
  uses-material-design: true

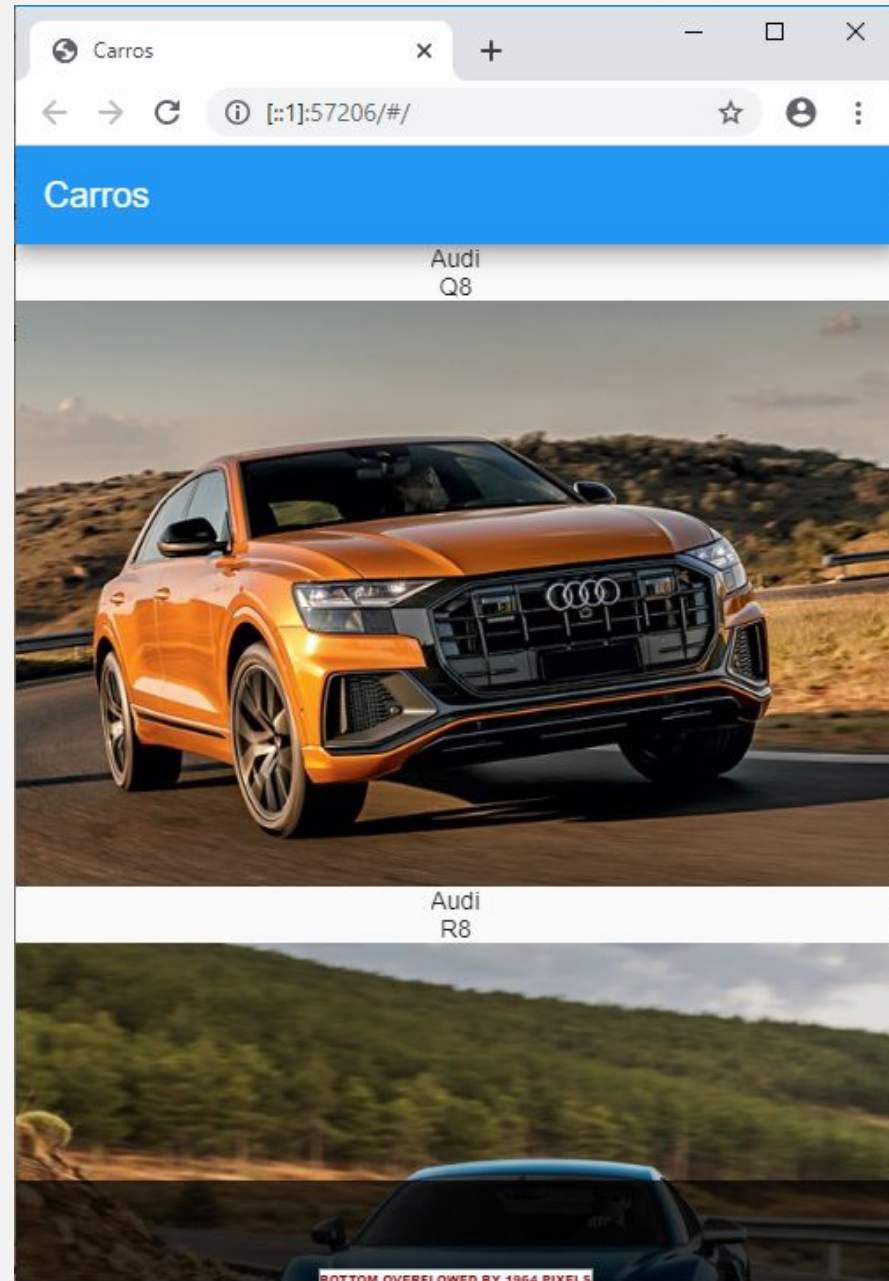
  assets:
    - assets/imagens/
```





Execute o App para visualizar  
o resultado até este  
momento:

```
flutter run -d chrome
```





- Para incluir barras de rolagem na App, adicione no widget ***ListaCarros*** o widget ***SingleChildScrollView***:

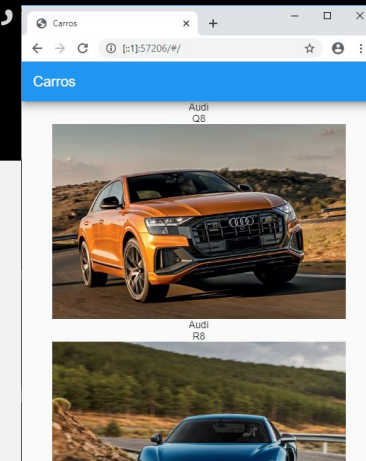


```
body:
  new SingleChildScrollView(
    child:
      new Column(children: <Widget>[
        new CarroWidget("Audi", "Q8", "assets/imagens/audi_q8.jpg"),
        new CarroWidget("Audi", "R8", "assets/imagens/audi_r8.jpg"),
        new CarroWidget("BMW", "M2", "assets/imagens/bmw_m2.jpg"),
        new CarroWidget("Ferrari", "488", "assets/imagens/ferrari_488.jpg"),
        new CarroWidget("Lamborghini", "Huracan", "assets/imagens/lamborghini_huracan.jpg"),
        new CarroWidget("Lamborghini", "Urus", "assets/imagens/lamborghini_urus.jpg"),
        new CarroWidget("Maserati", "GTS", "assets/imagens/maserati_gts.jpg"),
      ]),
  ),
);
```



- Para ajustar as dimensões das imagens altere o widget ***CarroWidget***:

```
// Definição da largura e altura da imagem  
Image.asset(foto, height: 100, width: 100,)  
  
// O tamanho da imagem aumenta se o fator de escala é menor que 1  
Image.asset(foto, scale: 2,)  
  
// O tamanho da imagem é ajustado a partir da largura  
Image.asset(foto, scale: 1.5, fit: BoxFit.fitWidth,)
```







- Para adicionar espaçamento entre os carros é possível utilizar os recursos do widget ***Padding***:

An immutable set of offsets in each of the four cardinal directions.

The EdgeInsets class specifies offsets in terms of visual edges, left, top, right, and bottom.

```
@override
Widget build(BuildContext context) {
  return new Padding(
    padding: EdgeInsets.all(20.0),
    child: Center(
      child:
        Column(children: <Widget>[
          Text(marca),
          Text(modelo),
          Image.asset(foto, scale: 1.5, fit: BoxFit.fitWidth,),
        ],),
    ),
  );
}
```

- Opções

```
EdgeInsets.symmetric(vertical: 8.0)
EdgeInsets.only(left: 40.0)
```



- Para adicionar bordas é possível explorar o widget ***BoxDecoration***

```
return new Padding(  
  padding: EdgeInsets.all(20.0),  
  child:  
    Container(  
      decoration: BoxDecoration(border: Border.all()),  
      padding: EdgeInsets.all(10.0),  
      child:  
        Center(  
          child:  
            Column(children: <Widget>[  
              Text(marca),  
              Text(modelo),  
              Image.asset(foto, scale: 1.5, fit: BoxFit.fitWidth,),  
            ],),  
        ),  
    ),  
);
```



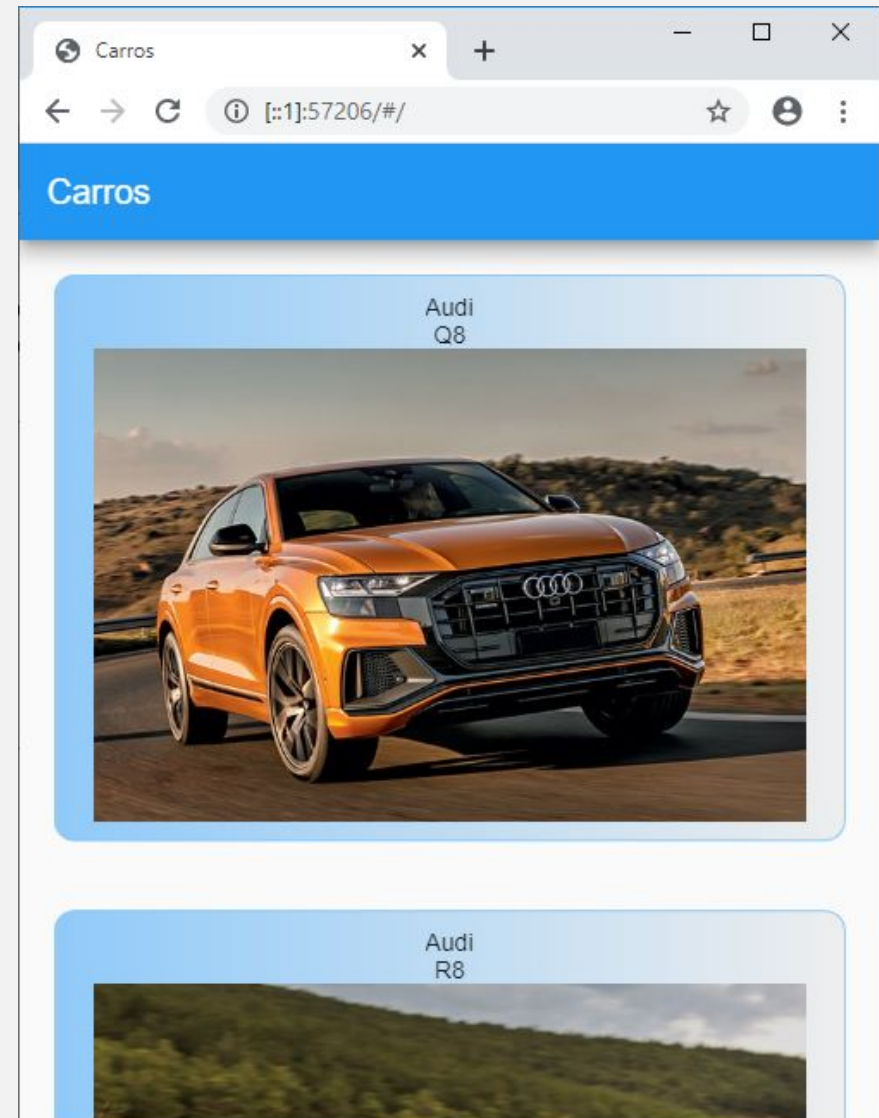
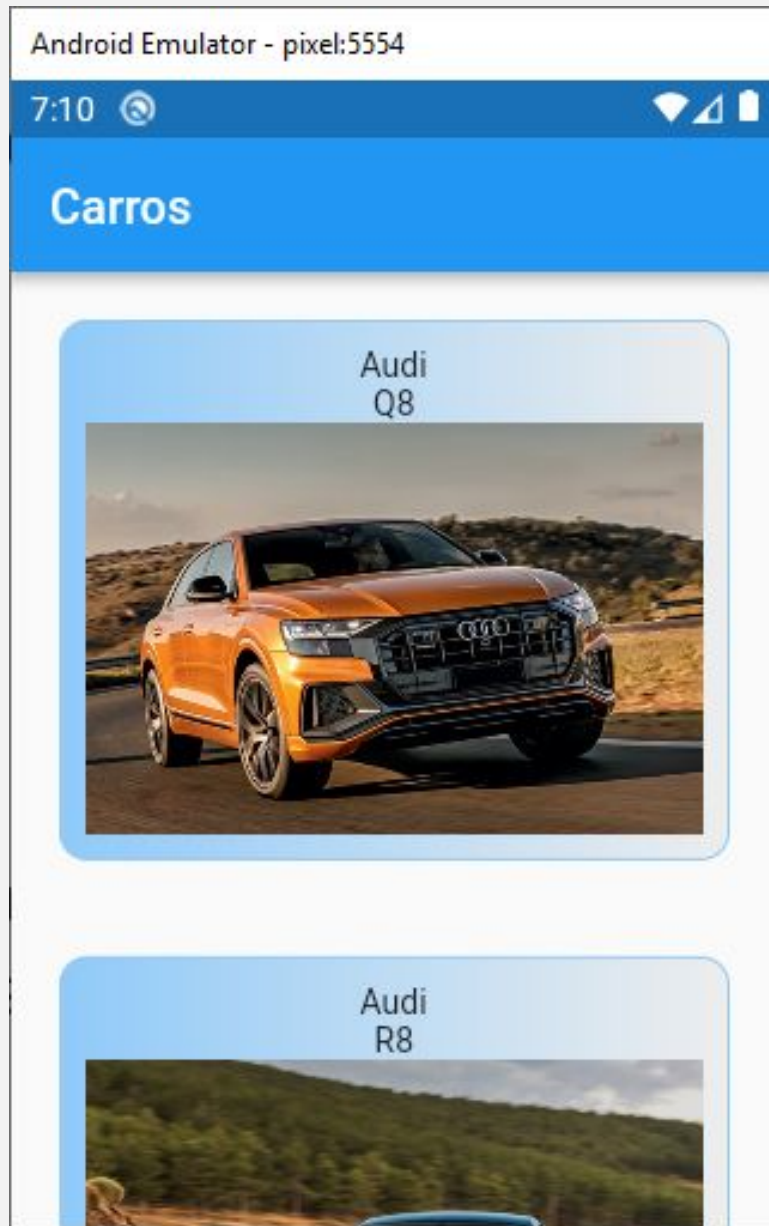
- Você pode explorar outras propriedades do ***BoxDecoration***:

```
decoration: BoxDecoration(  
  border: Border.all(color: Colors.blue[200], width: 1),  
  borderRadius: BorderRadius.circular(12),  
  
  gradient: new LinearGradient(  
    colors: [Colors.blue[200], Colors.grey[200]],  
    begin: Alignment.centerLeft,  
    end: Alignment.centerRight  
  ),  
),
```



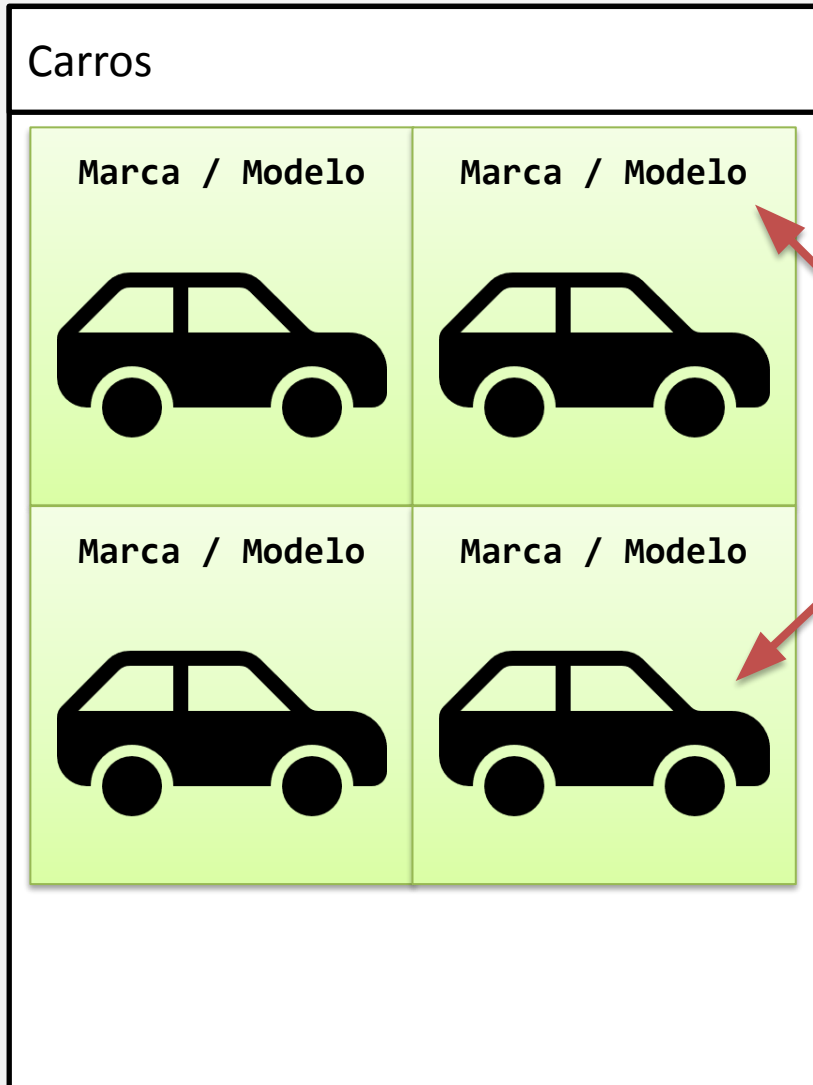
Veja mais em:

<https://medium.com/jlouage/flutter-boxdecoration-cheat-sheet-72cedaa1ba20>



# **ATIVIDADE PRÁTICA**

# Altere o App Listagem de Veículos



- a) Exibir a marca e o modelo do veículo em uma única linha.
- b) Exibir dois veículos em cada linha.

**FIM**