

# **Apresentação da Disciplina**

## **Computação para Dispositivos Móveis**

Prof. Gustavo Custodio  
gustavo.custodio@anhembi.br

# Bem-vindos

- Um pouco sobre mim:
  - Graduação em Ciência da Computação;
  - Mestrado em Ciência da Computação pela UFABC;
  - Atualmente cursando o doutorado.

# Bem-vindos

- Um pouco sobre vocês:
  - Curso?
  - Semestre?
  - Experiência com Mobile?

# Cr terios de Avalia  o

- **Avalia  o N1:**
  - Entrega Parcial 1 (50%)
  - Entrega Parcial 2 (50%)

# Critérios de Avaliação

- **Avaliação N1:**
  - Entrega Parcial 1 (50%)
  - Entrega Parcial 2 (50%)
- **Avaliação N2:**
  - Apresentação e relatório (50%)
  - Avaliação individual (50%)
  - Não temos APS.

## Critérios de Avaliação

$$\text{Média final} = 0.4 \times N1 + 0.6 \times N2$$

# Critérios de Avaliação

- **Substitutiva:**

- Caso alguém não esteja presente na apresentação ou
- alguém não tenha obtido nota o suficiente.
- A pessoa terá que adicionar uma funcionalidade na aplicação e apresentar no dia da sub.

# Conteúdo

- Introdução ao Flutter;
- Linguagem Dart;
- Manipulação de widgets;
- Navegação;
- Persistência;
- Consumo de APIs.





Apresentação da Disciplina

# Aplicações Móveis



**android**



**android**



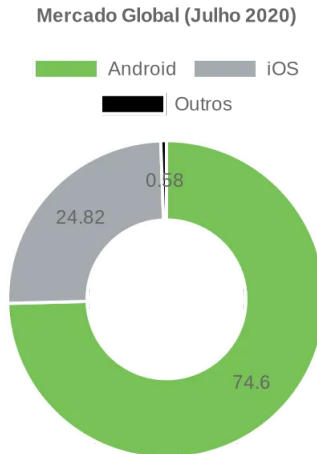
# Aplicações Móveis

- Mercado global praticamente dividido em iOS e Android.
  - Fonte: *StatCounter*

# Aplicações Móveis

- Mercado global praticamente dividido em iOS e Android.

– Fonte: *StatCounter*



# Aplicações Móveis

- E no Brasil?
  - Fonte: *StatCounter*

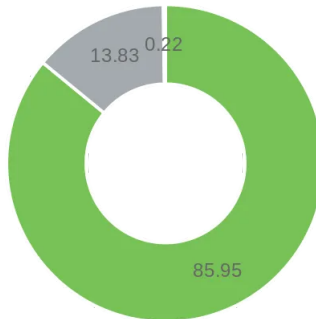
# Aplicações Móveis

- E no Brasil?

– Fonte: *StatCounter*

Mercado Brasil (Julho 2020)

Android iOS  
Outros



# Publicando Apps

- **Android:**
  - 25 USD para comprar a conta de desenvolvedor;
  - Sem processo de revisão para a publicação;
  - Pode ser desenvolvido em Windows, Mac ou Linux.



# Publicando Apps

- **Android:**

- 25 USD para comprar a conta de desenvolvedor;
- Sem processo de revisão para a publicação;
- Pode ser desenvolvido em Windows, Mac ou Linux.

- **iOS:**

- 100 USD por ano para manter a conta de desenvolvedor.
- Processo de revisão rigoroso.
- Só pode ser desenvolvido em um Mac.



Apresentação da Disciplina

# Aplicações Nativas

# Aplicações Nativas

- **Android**

- Originalmente em **Java**.
- Hoje permite o uso de **Kotlin**.
- Desenvolvimento é feito dentro do **Android Studio**.

# Aplicações Nativas

- **Android**

- Originalmente em **Java**.
- Hoje permite o uso de **Kotlin**.
- Desenvolvimento é feito dentro do **Android Studio**.

- **iOS**

- Originalmente programado em **ObjectiveC**.
- Hoje permite o uso de **Swift**.
- Desenvolvimento acontece na IDE **XCode**.

# Problema

- Necessário manter dois apps.
  - 2 equipes programando em linguagens diferentes.
  - 2 *codebases*.
  - 2 ciclos de *releases* diferentes.
  - 2 vezes o número de *bugs*.



## Solução: *Cross-Platform*

- Hoje, soluções que são compiladas em um app nativo:

## Solução: *Cross-Platform*

- Hoje, soluções que são compiladas em um app nativo:
  - Xamarin (C#/F#)
  - React Native (JavaScript)
  - Flutter (Dart)

## Solução: *Cross-Platform*

- Hoje, soluções que são compiladas em um app nativo:
  - Xamarin (C#/F#)
  - React Native (JavaScript)
  - Flutter (Dart)
- Embora a performance não seja a mesma de apps realmente nativos, a diferença é **imperceptível** na maioria dos casos.



## Solução: *Cross-Platform*

- Hoje, soluções que são compiladas em um app nativo:
  - Xamarin (C#/F#)
  - React Native (JavaScript)
  - Flutter (Dart)
- Embora a performance não seja a mesma de apps realmente nativos, a diferença é **imperceptível** na maioria dos casos.
- Utilizaremos o **Flutter**.



Apresentação da Disciplina

# Introdução ao Flutter

## O que é Flutter?



## O que é Flutter?

- Flutter é um kit de desenvolvimento de interfaces de usuário (UI).

## O que é Flutter?

- Flutter é um kit de desenvolvimento de interfaces de usuário (UI).
  - Criado pelo Google em 2015.
  - Utiliza a linguagem de programação **Dart**.
  - Utilizado para desenvolvimento de aplicações para **múltiplas plataformas** (incluindo Android e iOS).

## Vantagens

- Redução de tempo de desenvolvimento.
- Utilização da linguagem Dart, que une os melhores elementos de Java e JavaScript.
- Performance próxima de apps nativos.
- Leve quando comparado com outras soluções *Cross-Platform* (Ionic).

## Desvantagens

- Tamanho dos apps.
- Muitos widgets aninhados.

## Desvantagens

- Tamanho dos apps.
- Muitos widgets aninhados.

```
@override
Widget build(BuildContext context) {
  return Padding(
    child: Column(
      children: <Widget>[
        Row(
          children: <Widget>[
            Text('Here we go...'),
            Icon(Icons.format_indent_increase),
            Padding(
              child: Column(
                children: <Widget>[
                  Icon(Icons.photo),
                  Center(
                    child: Text('I can see my house from here!'),
```





Apresentação da Disciplina

Instalação

# Instalação

- O Windows é o sistema operacional mais utilizado, então vamos realizar a instalação nele.

# Instalação

- O Windows é o sistema operacional mais utilizado, então vamos realizar a instalação nele.
- O processo de instalação em outros Sistemas Operacionais está disponível em:
  - <https://docs.flutter.dev/get-started/install>

# Instalação

- Baixe o arquivo e extraia o arquivo:
  - flutter\_windows\_3.0.5-stable.zip

# Instalação

- Baixe o arquivo e extraia o arquivo:
  - `flutter_windows_3.0.5-stable.zip`
- Adicione o flutter nas variáveis de ambiente.

# Instalação

- O Flutter oferece um recurso para validar a instalação.

# Instalação

- O Flutter oferece um recurso para validar a instalação.
- Digite na linha de comando:
  - `flutter doctor`.
  - Isso mostrará problemas encontrados com a instalação do Flutter.

## Instalação

- Se algum problema for encontrado, você verá uma tela como essa:



# Instalação

- Se algum problema for encontrado, você verá uma tela como essa:

```
;\Flutter\flutter>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
✓] Flutter (Channel stable, 2.5.3, on Microsoft Windows [Version 10.0.19043.1348], locale en-US)
✗] Android toolchain - develop for Android devices
    X Unable to locate Android SDK.
      Install Android Studio from: https://developer.android.com/studio/index.html
      On first launch it will assist you in installing the Android SDK components.
      (or visit https://flutter.dev/docs/get-started/install/windows#android-setup for detailed instructions).
      If the Android SDK has been installed to a custom location, please use
      `flutter config --android-sdk` to update to that location.

✓] Chrome - develop for the web
✓] Android Studio (not installed)
✓] IntelliJ IDEA Community Edition (version 2021.1)
✓] VS Code (version 1.62.3)
✓] Connected device (2 available)

Doctor found issues in 2 categories.
```

# Instalação

- Se algum problema for encontrado, você verá uma tela como essa:

```
;\Flutter\flutter>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
✓] Flutter (Channel stable, 2.5.3, on Microsoft Windows [Version 10.0.19043.1348], locale en-US)
✗] Android toolchain - develop for Android devices
    ✗ Unable to locate Android SDK.
      Install Android Studio from: https://developer.android.com/studio/index.html
      On first launch it will assist you in installing the Android SDK components.
      (or visit https://flutter.dev/docs/get-started/install/windows#android-setup for detailed instructions).
      If the Android SDK has been installed to a custom location, please use
      `flutter config --android-sdk` to update to that location.

✓] Chrome - develop for the web
✓] Android Studio (not installed)
✓] IntelliJ IDEA Community Edition (version 2021.1)
✓] VS Code (version 1.62.3)
✓] Connected device (2 available)

Doctor found issues in 2 categories.
```

- Neste exemplo, foram encontrados dois problemas.

# Instalação

- É necessário a instalação do Android Studio e do Visual Studio para utilizar o Flutter.
  - Não vamos utilizá-los por enquanto, só precisamos instalá-los.

# Instalação

- É necessário a instalação do Android Studio e do Visual Studio para utilizar o Flutter.
  - Não vamos utilizá-los por enquanto, só precisamos instalá-los.
- Após completar a instalação, é necessário concordar com algumas licenças:
  - `flutter doctor --android-licenses`

# Instalação

- Durante a instalação do Visual Studio, marque a opção:
  - SDK do Windows 10;
  - C++ CMake tools;
  - MSVC v142 - VS C++ build tools.

# Instalação

- Durante a instalação do Visual Studio, marque a opção:
  - SDK do Windows 10;
  - C++ CMake tools;
  - MSVC v142 - VS C++ build tools.
- É possível que um problema seja encontrado ao concordar com as licenças do Android Studio.
  - Neste caso, abra o Android Studio e clique em Ferramentas → SDK Manager.
  - Procure pela opção *SDK Command-line Tools* e marque ela.

## Instalação

- Rode o flutter doctor novamente.

# Instalação

- Rode o flutter doctor novamente.
- Se nenhum problema for encontrado, você verá uma tela como essa:

```
Doctor summary (to see all details, run flutter doctor -v):  
[✓] Flutter (Channel stable, 3.0.5, on Manjaro Linux 5.15.59-1-MANJARO, locale  
    en_US.utf8)  
[✓] Android toolchain - develop for Android devices (Android SDK version 33.0.0)  
[✓] Chrome - develop for the web  
[✓] Linux toolchain - develop for Linux desktop  
[✓] Android Studio (version 2021.2)  
[✓] Connected device (2 available)  
[✓] HTTP Host Availability  
  
• No issues found!
```





Apresentação da Disciplina

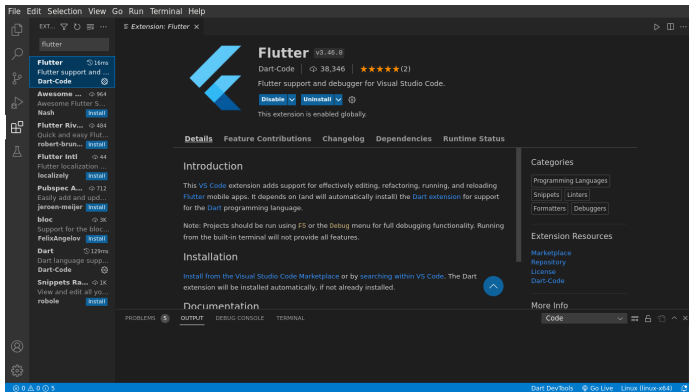
# Primeiro App

## VSCode

- Utilizaremos o VSCode para criar nosso primeiro app.
- Procure pela extensão Flutter (Dart-code).

# VSCode

- Utilizaremos o VSCode para criar nosso primeiro app.
- Procure pela extensão Flutter (Dart-code).



## Primeiro App

- Para criar o primeiro app, abra uma aplicação de linha de comando:
  - escolha a pasta que deseja salvar seu projeto.

# Primeiro App

- Para criar o primeiro app, abra uma aplicação de linha de comando:
  - escolha a pasta que deseja salvar seu projeto.
  - Em seguida, use o comando:
    - `flutter create meu-primeiro-app`
    - Isso criará um aplicativo base no diretório escolhido.

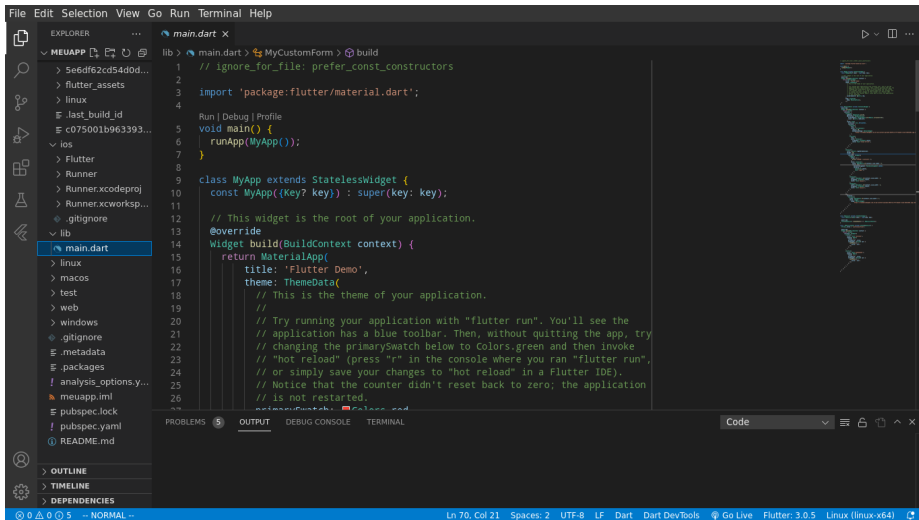
## Primeiro App

- Abra a pasta do projeto usando o VSCode.
- Procure o arquivo `main.dart` dentro da pasta `lib`.

## Primeiro App

- Abra a pasta do projeto usando o VSCode.
- Procure o arquivo `main.dart` dentro da pasta `lib`.
- Esse é o arquivo principal de nosso projeto e será o que modificaremos por enquanto.

# Primeiro App





## Primeiro App

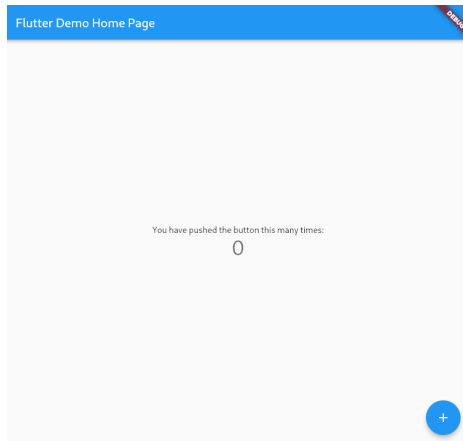
- Aperte F5 no VSCode para rodar o projeto.
  - Você também pode utilizar o comando `flutter run` dentro da pasta de algum projeto.

## Primeiro App

- Aperte F5 no VSCode para rodar o projeto.
  - Você também pode utilizar o comando `flutter run` dentro da pasta de algum projeto.
- Por padrão, um app que conta quantas vezes você clicou em um botão é criado.

# Primeiro App

- Aperte F5 no VSCode para rodar o projeto.
  - Você também pode utilizar o comando `flutter run` dentro da pasta de algum projeto.
- Por padrão, um app que conta quantas vezes você clicou em um botão é criado.



## Primeiro App

- Vamos substituir todo o código gerado automaticamente no `main.dart`.

## Primeiro App

- Vamos substituir todo o código gerado automaticamente no `main.dart`.
- Vamos começar escrevendo um *Hello World*.

# Primeiro App

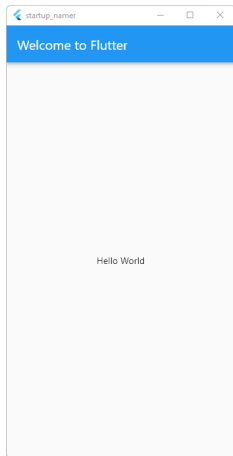
```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6 class MyApp extends StatelessWidget {
7   const MyApp({super.key});
8   @override
9   Widget build(BuildContext context) {
10    return MaterialApp(
11      title: 'Welcome to Flutter',
12      home: Scaffold(
13        appBar: AppBar(
14          title: const Text('Welcome to Flutter'),
15        ),
16        body: const Center(
17          child: Text('Hello World'),
18        ),
19      ),
20    );
21  }
22 }
```

# Primeiro App

- Resultado:

# Primeiro App

- Resultado:





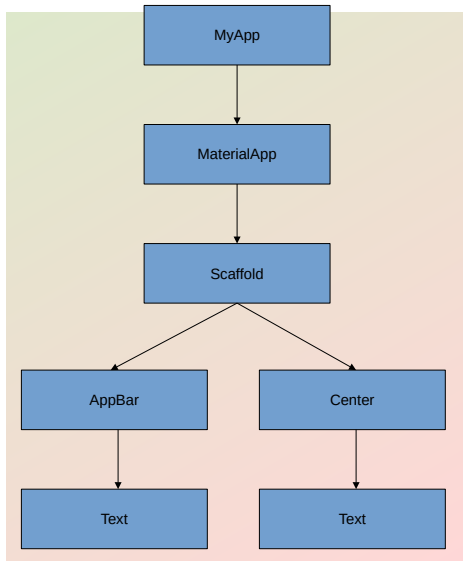
## Entendendo o App

- Tudo em Flutter é um Widget!

# Entendendo o App

- Tudo em Flutter é um Widget!
- `runApp(const MyApp())`
  - Essa linha inicializa o *framework* Flutter e instancia o MyApp.
  - A classe MyApp é um widget utilizado para preparar todos os dados globais e iniciar a árvore de widgets.
  - Os widgets que formam nosso app podem ser representados por uma árvore.

# Entendendo o App



## Entendendo o App

- A classe principal herda de *Stateless Widget*, ou seja, um Widget que não possui estado.
  - Utilizamos esse tipo de Widget quando não temos *inputs* do usuário e, dessa forma, não é necessário alterar algum estado.

## Entendendo o App

- A classe principal herda de *Stateless Widget*, ou seja, um Widget que não possui estado.
  - Utilizamos esse tipo de Widget quando não temos *inputs* do usuário e, dessa forma, não é necessário alterar algum estado.
  - Para os outros casos, temos um `StatefulWidget`.

## Entendendo o App

- O AppBar representa a barra azul que vemos no aplicativo.
  - É possível alterar a cor dessa barra.

## Exercício

- Tente mudar a cor da AppBar utilizada.
- Adicione mais widgets Text em linhas diferentes.

## Referências



Documentação do flutter.

<https://docs.flutter.dev/>.

Acessado 15/08/2022.



Faq do flutter.

<https://docs.flutter.dev/resources/faq#what-is-flutter>.

Acessado 15/08/2022.



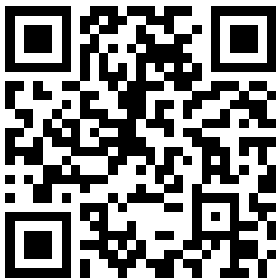
Simone Alessandria and Brian Kayfitz.

*Flutter Cookbook: Over 100 proven techniques and solutions for app development with Flutter 2.2 and Dart.*

Packt Publishing Ltd, 2021.



## Conteúdo



<https://gustavotcustodio.github.io/dispomoveis.html>

Obrigado

[gustavo.custodio@anhembi.br](mailto:gustavo.custodio@anhembi.br)