

# **DCC917A – TÓPICOS ESPECIAIS III: DESENVOLVIMENTO DE APLICATIVOS MÓVEIS**

**AULA 02**

**Carlos Bruno Oliveira Lopes**

*Engenheiro de Computação  
Mestre em Ciência da Computação*

# Desenvolvimento de aplicativos móveis

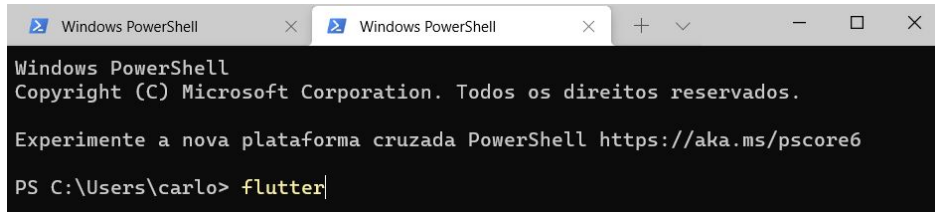
## Flutter SDK

- Baixar, instalar e configurar o Flutter SDK;
  - <https://flutter.io>
  - Vá em “Get started”
    - Selecione em qual sistema operacional você quer instalar:
      - » Windows, macOS, Linux ou Chrome OS.
    - Siga as instruções recomendadas pelo site do Flutter.
      - » Baixar a pasta “.zip”
      - » Definir o local da instalação (Criar um pasta e descompactar a pasta do Flutter nesse local)
      - » Configurar as variáveis de ambiente (Abrir propriedades do Windows, selecionar Variáveis de usuário, selecionar Path, Novo e colar o caminho “...\flutter\bin”)

# Desenvolvimento de aplicativos móveis

## Flutter SDK

- Baixar, instalar e configurar o Flutter SDK;
  - Abra um prompt de comando digite: “flutter”



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

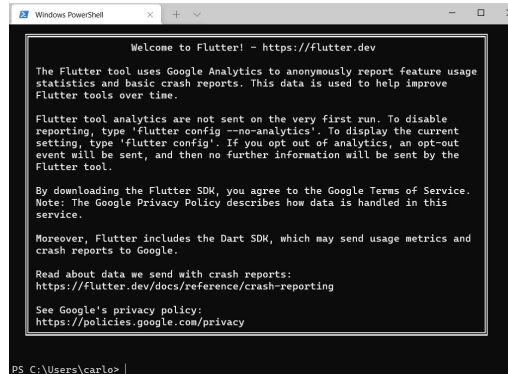
Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

PS C:\Users\carlo> flutter
```

# Desenvolvimento de aplicativos móveis

## Flutter SDK

- Baixar, instalar e configurar o Flutter SDK;
  - Abra um prompt de comando digite: “flutter”



```
Windows PowerShell
Welcome to Flutter! - https://flutter.dev

The Flutter tool uses Google Analytics to anonymously report feature usage
statistics and basic crash reports. This data is used to help improve
Flutter tools over time.

Flutter tool analytics are not sent on the very first run. To disable
reporting, type 'flutter config --no-analytics'. To display the current
setting, type 'flutter config'. If you opt out of analytics, an opt-out
event will be sent, and then no further information will be sent by the
Flutter tool.

By downloading the Flutter SDK, you agree to the Google Terms of Service.
Note: The Google Privacy Policy describes how data is handled in this
service.

Moreover, Flutter includes the Dart SDK, which may send usage metrics and
crash reports to Google.

Read about data we send with crash reports:
https://flutter.dev/docs/reference/crash-reporting

See Google's privacy policy:
https://policies.google.com/privacy

PS C:\Users\carlo>
```

# Desenvolvimento de aplicativos móveis

## Flutter SDK

- Executar o Flutter doctor para verificar se há alguma dependência de plataforma.
- Abra um prompt de comando digite: “flutter doctor”

```
Windows PowerShell
By downloading the Flutter SDK, you agree to the Google Terms of Service.
Note: The Google Privacy Policy describes how data is handled in this
service.

Moreover, Flutter includes the Dart SDK, which may send usage metrics and
crash reports to Google.

Read about data we send with crash reports:
https://flutter.dev/docs/reference/crash-reporting

See Google's privacy policy:
https://policies.google.com/privacy

PS C:\Users\carlo> flutter doctor
Running "flutter pub get" in flutter_tools...
```

```
PS C:\Users\carlo> flutter doctor
Running "flutter pub get" in flutter_tools... 12,1s
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 2.2.3, on Microsoft Windows [vers o 10.0.19043.1083],
    locale pt-BR)
[!] Android toolchain - develop for Android devices (Android SDK version 30.0.3)
    X Android licenses not accepted. To resolve this, run: flutter doctor
      --android-licenses
[✓] Chrome - develop for the web
[!] Android Studio (not installed)
[✓] Connected device (2 available)

! Doctor found issues in 2 categories.
PS C:\Users\carlo>
```

# Desenvolvimento de aplicativos móveis

## Android Studio

- Baixar e instalar
  - <https://developer.android.com/studio>
    - Faça o download da versão mais recente e instale-a;
- Instalação de pacotes necessários para flutter:
  - Abra o Android SDK Manager e instale:
    - A ultima versão do Android SDK, Android SDK Command-line Tools, and Android SDK Build-Tools;
- Execute novamente o comando “flutter doctor” e verifique se o Androide Studio foi localizado.
  - Alguns ajuste podem ser solicitados, tais como:
    - Problemas com licença → use o comando “flutter doctor --android-licenses” e aceite todas as licenças;
    - Problemas de localização do Android → use o comando “flutter config --android-studio-dir <directory>”

# Desenvolvimento de aplicativos móveis

## Android Studio

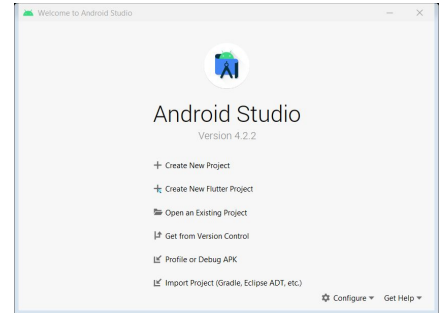
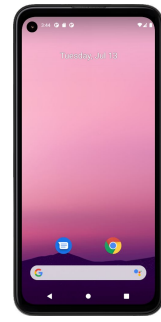
```
PS C:\Users\carlo> flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 2.2.3, on Microsoft Windows [vers o 10.0.19043.1083], locale pt-BR)
[✓] Android toolchain - develop for Android devices (Android SDK version 30.0.3)
[✓] Chrome - develop for the web
[✓] Android Studio (version 4.1.0)
[✓] Connected device (2 available)

• No issues found!
PS C:\Users\carlo> |
```

# Desenvolvimento de aplicativos móveis

## Android Studio

- Configuração do emulador Android
  - Inicie o Android Studio, vá em configure, selecione AVD Manager e clique em Create Virtual Device.
    - Escolha um dispositivo e selecione Next;
    - Selecione um ou mais imagem de sistema para versão do Android que será emulado e selecione Next. É recomendado a x86 ou x86\_64;
    - Em Emulated Performance, selecione Hardware - GLES 2.0 para habilitar o acelerador de hardware;
    - Verifique se a configuração AVD esta correta, clique em Finish.





# Desenvolvimento de aplicativos móveis

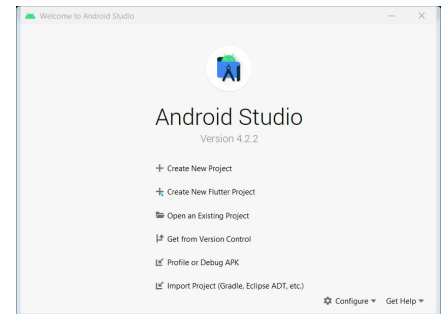
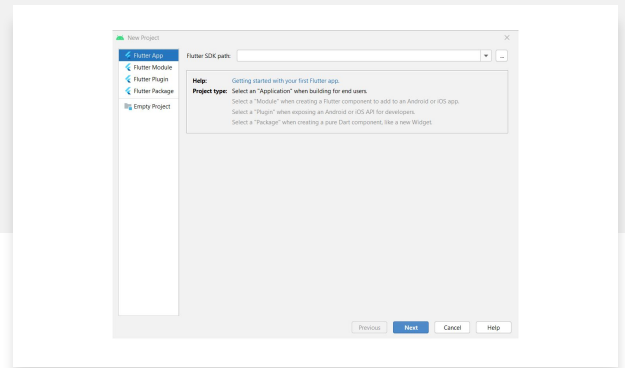
## Android Studio

- Configuração do editor Android e instalação do plugin
  - Inicie o Android Studio, em configure, selecione Plugins. Em Marketplace, digite Flutter e clique em install.
    - Quando a instalação iniciar será solicitado a instalação do Dart também, basta clicar em Yes ou install.
    - Então, reinicie a IDE.

# Desenvolvimento de aplicativos móveis

“Hello, World!”

- Inicie o Android Studio e clique em Create New Flutter Project;
- Um nova janela com quatros tipos de projetos aparecerá:
  1. Flutter Application (Construir um app para usuários)
  2. Flutter Plugin
  3. Flutter Package (Criar um componente puro Dart como uma widget)
  4. Flutter Module (Criar um componente Flutter para Android ou iOS)
    - Selecione Flutter App.
    - Em Flutter SDK Path informe a caminho do SDK do Flutter.
    - Clique Next.

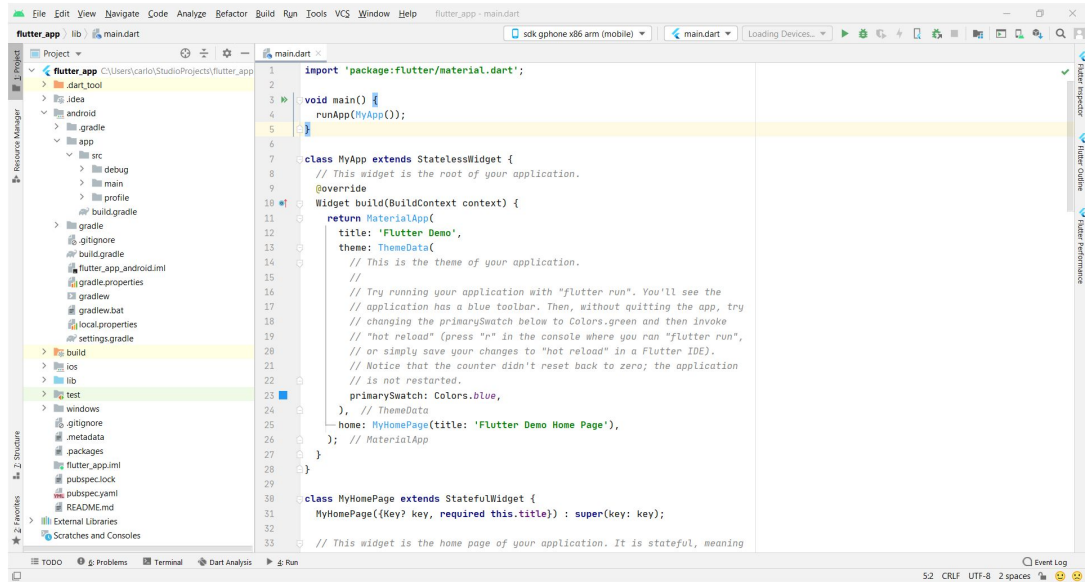


# Desenvolvimento de aplicativos móveis

“Hello World!”

- Na próxima Janela, informe:
  - O nome do Projeto: “flutter\_app”
  - Onde projeto será criado (“Mude para pasta que você usa para programar”)
  - Descrição do projeto (“Pode deixar como esta na configuração padrão”)
  - Organização (domínio da empresa) [“Não precisa mudar”]
  - O resto das configurações não há necessidade de mudar pode deixar na configuração padrão.
- Clique em Finish e projeto será criado.

# Desenvolvimento de aplicativos móveis

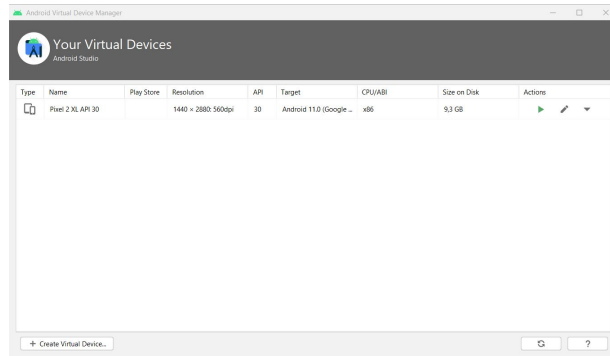


# Desenvolvimento de aplicativos móveis

“Hello World!”

- Após o carregamento do projeto clique no ícone do AVD Manager e inicie o emulador:

Clique aqui

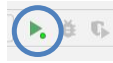


# Desenvolvimento de aplicativos móveis

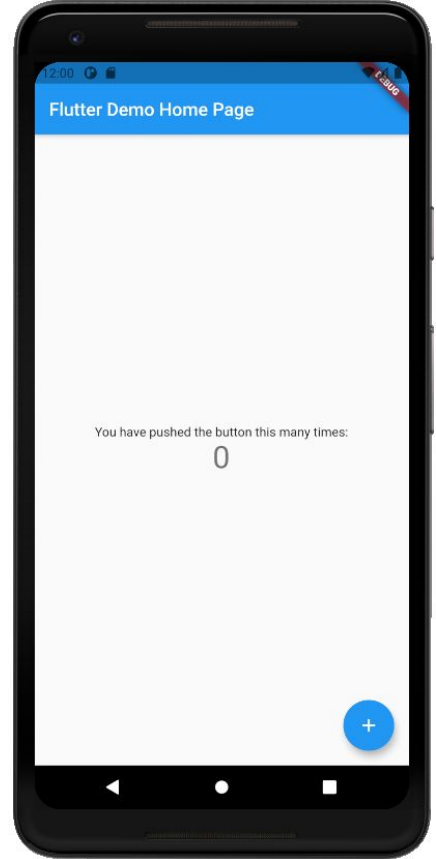
“Hello World!”

- Após a inicialização do emulador, execute o programa de demonstração, clicando no ícone representado por uma setinha em verde na janela do projeto:

Clique aqui



- Quando o programa for carregado no emulador
  - Clique no botão circular e veja o que acontece;



# Desenvolvimento de aplicativos móveis

## “Hello World!”

- Vá em Project → flutter\_app → lib e clique no arquivo “main.dart”
- Nesse arquivo conseguimos ver o **código em dart** que esta sendo executado no emulador.
- Encontre o trecho de código:  
home: MyHomePage(title: 'Flutter Demo Home Page'),
  - Altere para:  
home: MyHomePage(title: 'Olá mundo!'),  
Veja o que acontece no emulador depois que você salvar o arquivo
- Agora encontre o trecho:  
'You have pushed the button this many times:',
  - Altere para:  
'O número de vezes que o botão foi pressionado foi:',

# Desenvolvimento de aplicativos móveis

```
void main() {  
  
  runApp(MyApp()  
  );  
}  
  
class MyApp extends  
StatelessWidget {  
  @override  
  Widget build(BuildContext  
  context) {  
    return MaterialApp(  
      title: 'Flutter Demo',  
      theme: ThemeData(  
        primarySwatch:  
        Colors.blue,  
      ),  
      home: MyHomePage(title:  
        'Olá mundo!'),  
    );  
  }  
}
```

- Todo aplicativo Flutter é iniciado a partir de um método **main()**.
  - Esse método chama o método **runApp( )**
    - Ele recebe como passagem de parâmetro um widget de nível superior
      - Widget que esta no top da hierarquia e que irá conter todos os outros widgets
      - **MyApp**, classe que é um widget stateless (não renderiza de forma automática quando há mudanças de estados)
  - **MyApp**
    - **build()**, retorna uma widget com ou sem filhos. Ela é uma instância de **MaterialApp** (estilo UI do Google - framework);
      - title, argumento nomeado do construtor de MaterialApp
      - theme, define um tema para um aplicativo



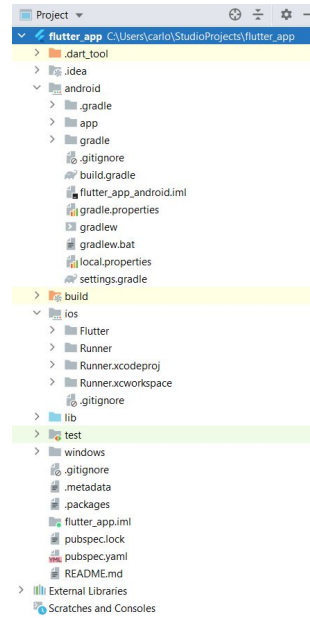
# Desenvolvimento de aplicativos móveis

```
class MyHomePage extends StatefulWidget {  
  MyHomePage({Key? key, required  
    this.title}) : super(key: key);  
  
  final String title;  
  
  @override  
  _MyHomePageState createState() =>  
    _MyHomePageState();  
}  
  
class _MyHomePageState extends  
  State<MyHomePage> {  
  int _counter = 0;  
  
  void _incrementCounter() {  
    setState(() {  
      _counter++;  
    });  
  }  
}
```

- Todo aplicativo Flutter é iniciado a partir de um método **main()**.
  - MyHomePage, (classe) widget stateful (renderização automática) [filha de MaterialApp]
  - Usa duas classes:
    - Classe principal, StatefulWidget
    - Classe de estado associada a ela, State
      - » Classe widget, pois possui um método build () que retorna um única widget, Scaffold;

# Desenvolvimento de aplicativos móveis

Estrutura principal de diretórios:



# Desenvolvimento de aplicativos móveis

Estrutura principal de diretórios:

- **android**. Contém código e recursos específicos do Android, tais como ícones de aplicação, código Java, configuração e os recursos transientes do Gradle.
- **ios**. Contém código de projeto específico do iOS.
- **lib**. Nesse diretório é onde o código da aplicação residirá. O arquivo `main.dart` fica nessa pasta.
- **test**. Diretório com os arquivos do Dart para execução de testes em seu aplicativo.
  - O Flutter fornece um utilitário Widget Tester que pode fazer uso desses testes para confirmar a funcionalidade dos widgets.

# Desenvolvimento de aplicativos móveis

Arquivos no diretório raiz do projeto:

- **.gitignore**. Arquivo que o Git usa para saber que arquivos, se houver algum, devem ser ignorados no controle de versões.
  - O uso do Git é totalmente opcional na criação de aplicativos Flutter, mas esse arquivo é gerado mesmo assim.
- **.metadata**. Dados que o Android Studio usa para rastrear o projeto.
- **.packages**. Usada para o Flutter vem gerenciar dependências dentro do projeto (gerenciador de pacotes).
  - O gerenciador chama-se Pub, e é esse arquivo que ele usa para rastrear dependências.
- **\*.iml**. Esse arquivo recebe o nome de seu projeto e é o arquivo de configuração de projeto do Android Studio.
- **pubspec.lock** e **pubspec.yaml**. **pubspec.yaml** é usado para descrever seu projeto para o Pub, inclusive suas dependências. O arquivo **pubspec.lock** é usado pelo Pub internamente.
  - **pubspec.yaml**, e é um dos poucos que será preciso editar em algum momento.
- **README.md**. Arquivo readme pode ser usado como quisermos. Normalmente, esse arquivo Markdown é o que sites como o GitHub usam para exibir informações sobre o projeto quando navegamos para um repositório em que ele está na raiz.

# Desenvolvimento de aplicativos móveis

## Modificando o programa base

– Faça as seguintes modificações:

- home: MyHomePage(title: 'Frases do dia'),
- children: <Widget>[  
    Text(  
    'Pressione o botão para gerar uma frase:',  
    )  
],
- primarySwatch: Colors.lightGreen,

# Desenvolvimento de aplicativos móveis

## Modificando o programa base

- Modificando a classe `_MyHomePageState`:
  - Substitua as seguintes linha de código:
    - `int _counter = 0;`
    - `_counter++;`
    - `'$_counter',`
  - por:
    - `int _numeroAleatorio = 0;`
    - `_numeroAleatorio = new Random().nextInt(4); //0,1,2,3`
    - `'$_numeroAleatorio',`

# Desenvolvimento de aplicativos móveis

## Modificando o programa base

### – Modificando a classe `_MyHomePageState`:

- Adicione embaixo de `int _numeroAleatorio = 0` as seguintes linha de código:

- `List _frases = [`

- `'O importante não é vencer todos os dias, mas jornada que nós leva a vitória ou a derrota.'`

- `'Mas vale um passáro na mão do que dois voando.'`

- `'O medo de perder tira a vontade de ganhar.'`

- `'Perder para a razão, sempre é ganhar'`

- `];`

# Desenvolvimento de aplicativos móveis

## Modificando o programa base

- Modificando a classe `_MyHomePageState`:
  - Substitua as seguintes linha de código:
    - `'$_numeroAleatorio';`
  - por:
    - `_frases[_numeroAleatorio],`