

Flutter

Ambiente de Desenvolvimento

Prof. Bruno Monteiro

brunomonteiro@ufersa.edu.br

V: 04/07/2023

Em resumo...

- **Flutter** é um *framework*.
- **Dart** é a linguagem adotada.
- Pertence ao Google.
- Cross-Platform (iOS e Android).
- Código-fonte único, mas permite configurações específicas.
- O código-fonte em Dart é compilado para código nativo.
- Disponibiliza biblioteca de componentes de interface com usuário (*widgets*).
- Plataformas: Mobile (estável), Web (recente), Desktop (muito recente)

Linguagem Dart

- Código Dart é compilado, pelo Flutter SDK, para código nativo (Android/iOS).
- Tipada e Orientada a Objetos.
- Sintaxe “similar” a Java
 - <https://codelabs.developers.google.com/codelabs/from-java-to-dart?hl=pt-br>

Estrutura de Widgets

- Widget = Componente de User Interface.
- Componentes são hierarquizados.
- O Flutter “desenha” os componentes visuais nativos, por meio de um canvas na plataforma alvo (Android/iOS).
- Suporta “imperative style” e “declarative style” (mais usado).

Ambiente de Desenvolvimento

1. Java JDK

Download: <https://www.oracle.com/java/technologies/downloads/>

- Flutter 2.0 pedia JDK 8 ou mais;
- Flutter 3.0 pede JDK 11 ou mais (vou instalar 17 pois é LTS)

Configuração:

1. No cmd, teste: `java -version` e `javac -version`
 - se não funcionar: adicionar no path do sistema operacional: `...\jdk-17\bin`

Ambiente de Desenvolvimento

2. Git

Download: <https://git-scm.com/downloads>

Configuração:

1. Instalação padrão

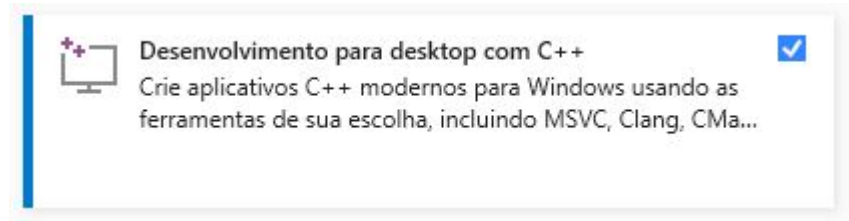
Ambiente de Desenvolvimento

3. Visual Studio Community

Download: <https://visualstudio.microsoft.com/downloads/>

Configuração:

1. Instalar pacote: “Desenvolvimento para desktop com C++”



Ambiente de Desenvolvimento

4. Android Studio

Download: <https://developer.android.com/studio>

- OBS: requer mais de 15 GB de espaço, dependendo das APIs Android instaladas.

Configuração:

1. Instalar Android Virtual Device.
2. Em SDK Manager (SDK Platforms): instale pelo menos a versão do seu aparelho.
3. Em SDK Manager (SDK Tools): instale “Android SDK Command-line tools (latest)”.
4. Em AVD Manager (Virtual Device Manager): crie um dispositivo.
5. Pode ser necessário instalar: Intel x86 Emulator Accelerator (HAXM).
6. Instalar plugin do Flutter (e Dart) no Android Studio.
7. No smartphone Android: em “Modo de desenvolvedor” → Depuração USB (habilitado)

Ambiente de Desenvolvimento

5. Flutter SDK

Download: <https://flutter.dev/docs/get-started/install>

Configuração:

1. O path não pode ter espaços nem ficar em “Arquivos de Programas”.
2. Adicionar no path do sistema operacional: ...\\flutter\\bin
3. No cmd, teste: flutter --version
4. No cmd, execute (e aceite todas com y): flutter doctor --android-licenses
5. No cmd, execute: flutter doctor
 - Se erro “Unable to locate android sdk”
 - Adicione no path do windows: “...\\android-sdks\\platforms” e “...\\android-sdks\\platform-tools”
 - Se erro “Android Studio (not installed)”
 - Execute: flutter config --android-studio-dir="C:\\Program Files\\Android\\Android Studio"

Ambiente de Desenvolvimento (Flutter)

6.1. “Hello World”

1. New Flutter Project → Generator: Flutter
2. Informe o “Flutter SDK path”: pasta do Flutter.
3. Preencha o formulário do wizard e crie o novo projeto.
4. Selecione onde o projeto será executado:
 - Emulador (criado no Device Manager) ou
 - Smartphone (com depuração USB habilitada)
5. Menu: Run → Run “main.dart”

Ambiente de Desenvolvimento (Dart)

6.2. “Hello World”

1. New Flutter Project → Generator: Dart
2. Informe o “Dart SDK path”: ...\\flutter\\bin\\cache\\dart-sdk
3. Preencha o formulário do wizard e crie o novo projeto
4. Coloque os códigos-fontes .dart na pasta lib
5. Botão direito no arquivo com main(): Run arquivo.dart
6. Se desejar compilar: dart compile exe arquivo.dart
 - Coloque no final do código-fonte para segurar tela: `stdin.readLineSync()`

Flutter: Pastas do projeto

- .dart_tool : arquivos compilados (não tocar)
- .idea : metadados do intellij (não tocar)
- android : arquivos relacionados à aplicação Android (raramente usado)
- build : pacote compilado.
- ios : arquivos relacionados à aplicação iOS (raramente usado)
- lib : código-fonte do projeto (IMPORTANTE)
- test : criação de casos de teste (opcional)
- arquivos relacionados à aplicação Web (raramente usado)

Flutter: Arquivos do projeto

- .gitignore : o que ignorar no commit para o repositório
- .metadata : metadados do Flutter (não tocar)
- .packages : dependências
- <projeto>.iml : metadados do intellij (não tocar)
- pubspec.lock : gerado a partir do pubspec.yaml (não tocar)
- pubspec.yaml : metadados do projeto (IMPORTANTE)
- README.md : “leia-me” que será exibido no repositório