



# Flutter

Prof. Rogério B. Andrade

# Introdução

- ◆ Dart criada em 2011 pelo Google
- ◆ Linguagem de alta performance (inicialmente substituir Javascript)
- ◆ É uma linguagem compilada e multiplataforma
- ◆ Server para Mobile, Web e Desktop
- ◆ Compila para a plataforma desejada
- ◆ Dart é conjunto de bibliotecas
- ◆ Dart é orientada a objetos

# Introdução

- ◆ Dart com Hot Reload (salva e executa rapidamente)
- ◆ Just i n Time (compilação para uma linguagem Intermediária, para executar rapidamente a alteração do código, não é compilação para linguagem de máquina).
- ◆ Terminar comando com ponto e virgula
- ◆ A palavra var indica a compilador para definir o tipo de uma variável, do contrário podemos definir diretamente indicando os respectivo tipo



# Introdução

- ◆ Type safe: Analisa o código durante o desenvolvimento para evitar erros na execução do código
- ◆ Variável no Dart precisam ser inicializadas, a não se que digamos que a variável será inicializada posteriormente (null safety)
- ◆ ?variável permite que inicialização ocorra posteriormente ou ainda utilizar o operador late seguido do tipo da variável e depois o nome da variável
- ◆ Por padrão todas as propriedades de uma classe são públicas e Dart não possui a key word private, a indicação de encapsulamento se dá pelo uso de um underline antes do nome da variável

# Introdução

Dart e Flutter trabalham com o conceito de composição

Para criar um projeto Flutter:

- create flutter <nome do aplicativo>
  - \* o flutter vai criar um projeto base
  - \* entrar no diretório (pasta do aplicativo criado)
- cd <nome do aplicativo>



# Introdução

Pastas e arquivos do projeto:

Pubspec.yaml: arquivo principal do projeto, pois, tem as configurações

- Nome do projeto, descrição, versões do sdk e as dependências (pacotes que serão utilizados no projeto). Este arquivo não tem ponto e virgula e a indentação é extremamente importante. Arquivos externos podem ser adicionados na pasta assets e para isto deve retirar o comentário da linha

Pubspec.lock: arquivo compilado de cache

Iml: arquivo de configurações

gitIgnore: informações que não serão sincronizados com outros desenvolvedores ou que fique público (exemplo chaves de apis)

Web: diretório de saída se for gerado o aplicativo para web

# Introdução

Pastas e arquivos do projeto:

Lib: é a pasta principal de trabalho que contém a main. Dart

Ios, android: pastas de configuração



# Introdução

```
lib > main.dart > MeuAplicativo
1  import 'package:flutter/material.dart';
2
3  void main() {
4    runApp(MeuAplicativo());
5  }
6
7  class MeuAplicativo extends StatelessWidget {
8    const MeuAplicativo({Key? key}) : super(key: key);
9
10   @override
11   Widget build(BuildContext context) {
12     return Container(
13       child: null,
14     );
15   }
16 }
17
```

Classe `MeuAplicativo`, que é instanciada pela função `main`, a primeira a ser executada quando o aplicativo é iniciado.



# Introdução

O método Build existe em toda tela, todo componente, ele é responsável por renderizar (desenhar) a widget na tela. Widget são componentes visuais que definem a interface de um aplicativo. No caso abaixo, o retorno será uma tela sem nenhum componente,

```
1 .
@override
Widget build(BuildContext context) {
  return Container(
    child: null,
  );
}
```