

# Introdução ao Dart

## Computação para Dispositivos Móveis

Prof. Gustavo Custodio  
gustavo.custodio@anhembi.br

## Algumas Datas Importantes

- Primeira entrega - Navegação entre telas (20%)
  - 26/09 e 27/09
- Segunda entrega - Armazenamento offline (20%)
  - 17/10 e 18/10
- Terceira entrega - App completo com consumo de APIs (60%)
  - 05/12 e 06/12



Introducao ao Dart

# Introdução à Linguagem Dart

# Introdução ao Dart



Dart

## Dart: uma Linguagem Familiar

- Linguagem de programação utilizada para desenvolvimento no Flutter.

## Dart: uma Linguagem Familiar

- Linguagem de programação utilizada para desenvolvimento no Flutter.
- Criada pelo Google em 2011.

## Dart: uma Linguagem Familiar

- Linguagem de programação utilizada para desenvolvimento no Flutter.
- Criada pelo Google em 2011.
- Tinha o objetivo de substituir o JavaScript como linguagem da Web.

## Dart: uma Linguagem Familiar

- Linguagem de programação utilizada para desenvolvimento no Flutter.
- Criada pelo Google em 2011.
- Tinha o objetivo de substituir o JavaScript como linguagem da Web.
- Quando foi criada, buscava combinar a natureza dinâmica do JavaScript com o *design* baseado em classes.
  - Lembra bastante o TypeScript.



# Dart: uma Linguagem Familiar



# Dart: uma Linguagem Familiar

- Características da Linguagem Dart:

# Dart: uma Linguagem Familiar

- Características da Linguagem Dart:
  - Variáveis e constantes declaradas usando `var`, `final`, `const`.
  - Possibilidade de definir tipos para variáveis.
  - Classes e construtores.
  - Linguagem *null-safe*.

# Linguagem Dart

- Há um *sandbox* da linguagem Dart disponível em:

# Linguagem Dart

- Há um *sandbox* da linguagem Dart disponível em:
  - <https://dartpad.dev/>.
- Há também a opção do replit:
  - <https://dartpad.dev/>.

# Linguagem Dart

Todos os programas em Dart são iniciados com o método `main` conforme mostrado abaixo:

# Linguagem Dart

Todos os programas em Dart são iniciados com o método main conforme mostrado abaixo:

```
1  void main() {  
2      print("Meu primeiro programa");  
3  }
```

# Variáveis

- Variáveis em Dart são declaradas definindo tipos, assim como a maioria das linguagens de programação:

- `int` umInteiro = 10;



# Variáveis

- Variáveis em Dart são declaradas definindo tipos, assim como a maioria das linguagens de programação:

- `int` umInteiro = 10;

- No entanto, também é possível utilizar a palavra reservada `var` para inferir os tipos de variáveis:

- `var` outroInteiro = 20;

# Variáveis

- Assim como o Java, constantes são definidas usando a palavra **final**.
- E assim como o JavaScript, constantes são definidas usando o **const**.

# Variáveis

- Assim como o Java, constantes são definidas usando a palavra **final**.
- E assim como o JavaScript, constantes são definidas usando o **const**.
- Quando usamos cada um então?

# Variáveis

- A maior diferença entre `final` e `const` é que `const` é determinado durante a compilação e `final` é determinado durante a execução.

# Variáveis

- A maior diferença entre `final` e `const` é que `const` é determinado durante a compilação e `final` é determinado durante a execução.
  - Há certas coisas que você não pode atribuir a uma `const`.
  - Exemplo: `Datetime.now()` não funciona com `const` porque é resolvido em tempo de execução.

# Strings

- Strings em Dart são muito parecidas com Java.

# Strings

- Strings em Dart são muito parecidas com Java.
  - Tipo String é escrito com letra maiúscula.
  - O método contains é utilizado para verificar se uma substring está contida.

# Strings

```
1  void main() {  
2      String nome = "Gustavo";  
3  
4      if (nome.contains("a")) {  
5          print("Seu nome contém a letra a.");  
6      }  
7  }
```



## Strings

- A linguagem não permite concatenação de tipos diferentes.

# Strings

- A linguagem não permite concatenação de tipos diferentes.

```
1  void main() {  
2      String nome = "Michael";  
3      int idade = 20;  
4  
5      print(nome + " tem " + idade + " anos.");  
6  }
```

- Encontraremos um erro ao tentar realizar essa operação.

# Strings

- Variáveis podem ser impressas junto com Strings utilizando o \$:

# Strings

- Variáveis podem ser impressas junto com Strings utilizando o \$:

```
1
2  void main() {
3      String nome = "Michael";
4      int idade = 20;
5
6      print("${nome} tem ${idade} anos.");
7  }
```

## Strings

- Também é possível dispor o conteúdo de uma String em múltiplas linhas delimitando a String por aspas triplas (“““””).

# Strings

- Também é possível dispor o conteúdo de uma String em múltiplas linhas delimitando a String por aspas triplas (“““”).

```
1 String textoMultiLinhas = """
2     Lorem ipsum dolor sit amet, consetetur sadipscing elitr,
3     sed diam nonumy eirmod tempor invidunt ut labore et
4     dolore magna aliquyam erat, sed diam voluptua.
5     At vero eos et accusam et justo duo dolores et ea rebum.
6     Stet clita kasd gubergren, no sea takimata sanctus est
7     Lorem ipsum dolor sit amet.
8     """;
9 print(textoMultiLinhas);
```

## Leitura de Inputs

- No Dart Utilizamos o comando `stdin.readLineSync` para lermos o input de um usuário.

## Leitura de Inputs

- No Dart Utilizamos o comando `stdin.readLineSync` para lermos o input de um usuário.
- Devemos importar o pacote `io` do Dart para conseguirmos realizar essa operação:
  - `import "dart:io";`



# Leitura de Inputs

```
1  import "dart:io";
2
3  void main() {
4      print("Digite seu nome de usuário:");
5      String? usuario = stdin.readLineSync();
6      print("Digite a sua senha:");
7      String? senha = stdin.readLineSync();
8  }
```

## Leitura de Inputs

- Para que serve o ponto de interrogação na frente do tipo?
  - Experimente removê-lo e veja o que acontece.

## Leitura de Inputs

- Para que serve o ponto de interrogação na frente do tipo?
  - Experimente removê-lo e veja o que acontece.
- Dart é uma linguagem *null-safe*.
  - Ou seja, ela não permite que uma variável assuma valores nulos.
  - Em um input, é possível que a entrada do usuário seja nula.
  - Indicamos isso usando a interrogação.

## Leitura de Inputs

- E como lemos inputs de outros tipos (int, double etc)?

## Leitura de Inputs

- E como lemos inputs de outros tipos (int, double etc)?
  - `int.parse(stdin.readLineSync()!);`
  - `double.parse(stdin.readLineSync()!);`
  - Adicionamos o ! para obrigar o compilador a assumir que não haverá valores nulos.

## Exercício

- Escreva um programa que leia três inputs:
  - O primeiro nome;
  - O nome do meio;
  - O sobrenome;
- Mostra na tela a concatenação dos nomes.

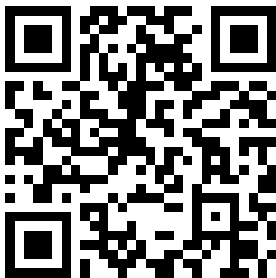
## Referências

 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)