



# INTRODUÇÃO A LINGUAGEM DE PROGRAMAÇÃO DART

Prof. Me. Hélio Esperidião

# A linguagem

- Dart é uma linguagem de programação apresentada pelo Google, em 2011, com o objetivo de substituir o JavaScript como principal linguagem embutida nos navegadores.

# Algumas características da linguagem são:



A sintaxe é C-like, portanto, se você programa em Java, C#, PHP ou Javascript, você não terá dificuldades em aprender a linguagem;



Segue o paradigma orientado a objetos;



Todos os objetos herdam da classe Object;



Fortemente tipada, mas não é necessário colocar um tipo, pois Dart consegue inferir os tipos;

# Variáveis

Tipo	Descrição	Exemplo
int	Valores inteiros	1, 2, -1233123
double	Números de ponto flutuante	0.5, -333.234
String	Texto de comprimento arbitrário. Aspas simples ou duplas podem ser usadas para criar uma string.	Texto' ou "Text"
bool	Um booleano é um valor lógico que pode ser verdadeiro ou falso.	true ou false
List	Uma coleção de objetos.	[1, 2, 3]
Map	Um mapa é um objeto que associa chaves e valores.	{ 'key-1': 'value-1', 'key-2': 'value-2' }

# Operadores relacionais

Operação	Descrição	Exemplo
==	Igual	k == 0
!=	Diferente	x != 3
>	Maior que	w > 7
<	Menor que	a < 22
>=	Maior ou Igual a	b >= 80
<=	Menor ou Igual a	c <= -3

# messageBox(String msg)

```
messageBox(String msg) {  
    showDialog(  
        context: context,  
        builder: (context) {  
            return AlertDialog(  
                content: Text(msg),  
            );  
        },  
    );  
}
```

# Formulário

```
componenteTxtNota1() {  
  TextField txtNota1 = TextField(  
    controller: txtN1Controller,  
    //controller: myController,  
    decoration: InputDecoration(hintText: 'Nota 1'));  
  return txtNota1;  
}  
componenteTxtNota2() {  
  TextField txtNota2 = TextField(  
    controller: txtN2Controller,  
    decoration: const InputDecoration(hintText: 'Nota 2'));  
  return txtNota2;  
}  
componenteLblResposta() {  
  Text lblResposta = Text("Média: $resposta");  
  return lblResposta;  
}
```

```
double calcularMedia(double n1, double n2) {  
    double x = n1;  
    double y = n2;  
    double r = (x + y) / 2;  
  
    return r;  
}
```



```
componenteFormulario() {  
    return Center(  
        child: Column(children: <Widget>[  
            componenteTxtNota1(),  
            componenteTxtNota2(),  
            componenteBtnCalcular(),  
            componenteLblResposta(),  
        ]));  
}
```

```
componenteBtnCalcular() {  
  TextButton btn1 = TextButton(  
    onPressed: () {  
      double x = double.parse(txtN1Controller.text);  
      double y = double.parse(txtN2Controller.text);  
      double r = calcularMedia(x, y);  
      if (r < 6.0) {  
        messageBox("Reprovado");  
      } else {  
        messageBox("Aprovado");  
      }  
      setState(() {  
        resposta = r.toString();  
      });  
    },  
    child: const Text("Calcular 1"),  
  );  
  return btn1;  
}
```

# componenteBtnCalcular()

```
componenteBtnTabuada() {  
  TextButton btn1 = TextButton(  
    onPressed: () {  
      double x = double.parse(txtN1Controller.text);  
      String r = "";  
      for (var i = 0; i <= 10; i++) {  
        double temp = i*x;  
        r = "$r$x x $i=$temp\n";  
      }  
      setState(() {  
        resposta = r.toString();  
      });  
    },  
    child: const Text("Calcular 1"),  
  );  
  return btn1;  
}
```

# Repetições

```
componenteBtnTabuadaWhile() {
  TextButton btn1 = TextButton(
    onPressed: () {
      double x = double.parse(txtN1Controller.text);
      String r = "";
      var i = 0;
      while(i<=10) {
        double temp = i*x;
        r = "$r$x x $i=$temp\n";
        i++;
      }
      setState(() {
        resposta = r.toString();
      });
    },
    child: const Text("Calcular 1"),
  );
  return btn1;
}
```

```
componenteBtnTabuadaDoWhile() {  
  TextButton btn1 = TextButton(  
    onPressed: () {  
      double x = double.parse(txtN1Controller.text);  
      String r = ""; var i = 0;  
      do{  
        double temp = i*x;  
        r = "$r$x x $i=$temp\n";  
        i++;  
      }while(i<=10);  
      setState(() {  
        resposta = r.toString();  
      });  
    },  
    child: const Text("Calcular 1"),  
  );  
  return btn1;  
}
```

# Do While

Programação  
orientada a  
objetos

Objetos

Classes

Atributos

# Objetos

- Um objeto é uma implementação de Tipos Abstratos de Dados (TAD's) que encapsula dados e operações. Todo objeto tem um estado, um comportamento e uma identidade.

# Classes

- Classe é uma abstração que define um tipo de objeto e o que objetos desse determinado tipo tem dentro dela (atributos) e também que tipo de ação ele é capaz de executar (métodos).



# Atributos

- São características que os objetos possuem. Ao definir um atributo sempre é passado nome e o que irá guardar, ou seja, qual tipo de dado que ele vai armazenar, exemplo: cor do olho, altura.

# Construtores

- Método que inicializa o estado de um objeto.

# Retangulo.dart

```
class Retangulo {  
  late double base;  
  late double altura;  
  Retangulo.parametros(this.base, this.altura);  
  Retangulo.parametroSimples(double lado){  
    base=lado;  
    altura=lado;  
  }  
  Retangulo(){  
    base=0;  
    altura=0;  
  }  
}
```

## Encapsulamento

No Java e C# você usa a palavra reservada `private`

No Dart basta colocar um *underline* (`_`) no início do nome de um atributo, método ou classe para torná-lo privado.

# Encapsulamento

```
class Retangulo {
  late double _base;
  late double _altura;
  double get altura {
    return _altura;
  }
  set altura(double altura){
    _altura = altura;
  }
  double get base {
    return _base;
  }
  set base(double base){
    _base = base;
  }
  Retangulo.parametros(this._base, this._altura);
  Retangulo.parametroSimples(double lado) {
    _base = lado;
    _altura = lado;
  }
  Retangulo() {
    _base = 0;
    _altura = 0;
  }
}
```

# Métodos

```
// import 'dart:math';  
double area() {  
    return _base*_altura;  
}  
double perimetro() {  
    return 2*_base + 2*_altura;  
}  
double diagonal() {  
    return sqrt(_base*_base + _altura*_altura );  
}
```

```
componenteBtnAreaRetangulo() {  
  TextButton btn1 = TextButton(  
    onPressed: () {  
      double b = double.parse(txtN1Controller.text);  
      double h = double.parse(txtN2Controller.text);  
      Retangulo ret1 = new Retangulo();  
      ret1.altura=h;  
      ret1.base=b;  
      double a = ret1.area();  
      String r =a.toString();  
      setState(() {  
        resposta = r.toString();  
      });  
    },  
    child: const Text("Calcular 1"),  
  );  
  return btn1;  
}
```

# Importe e instancie

# Possibilidade de nomear construtores

```
class Usuario {  
    late String nome;  
    late String email;  
    late String Senha;  
  
    Usuario.comParametros(String nome, String email, String Senha) {  
        this.nome = nome;  
        this.email = email;  
        this.Senha = Senha;  
    }  
  
    Usuario.semSenha(String nome, String email) {  
        this.nome = nome;  
        this.email = email;  
    }  
  
    Usuario() {  
        nome = "";  
        email = "";  
        Senha = "";  
    }  
}
```



```
class Usuario {  
    late String _nome;  
    late String _email;  
    late String _Senha;  
    String get Senha => _Senha;  
    set Senha(String value) {  
        _Senha = value;  
    }  
  
    String get email => _email;  
    set email(String value) {  
        _email = value;  
    }  
  
    String get nome => _nome;  
    set nome(String value) {  
        _nome = value;  
    }  
}
```

# Encapsulamento

# Métodos

```
import 'dart:math';

class Point {
  double x = 0;
  double y = 0;

  Point(this.x, this.y);

  double distanceTo(Point other) {
    var dx = x - other.x;
    var dy = y - other.y;
    return sqrt(dx * dx + dy * dy);
  }
}
```