



# INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PIAUÍ

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ  
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS  
MÓDULO V - PROGRAMAÇÃO PARA DISPOSITIVOS MÓVEIS  
PROF.: Ely Miranda  
ALUNO: Lucas Gomes de Oliveira (20191ADS0185)

## Exercícios de Dart 02

1. Crie uma função que receba um array e retorne a soma de todos os elementos.

```
void main() {  
    print(  
        somaArray([34, 67, 1, 3, 909])  
    );  
}  
  
int somaArray(List<int> arrayParam) {  
    int valor = 0;  
    for (int i = 0; i < arrayParam.length; i++) {  
        valor += arrayParam[i];  
    }  
    return valor;  
}
```

2. Crie uma função que receba um array e retorna a multiplicação de todos os elementos.

```
void main() {  
    print(  
        multiplicaArray([190, 11, 21, 546])  
    );  
}  
  
int multiplicaArray(List<int> arrayParam) {  
    int valor = 1;  
    for (int i = 0; i < arrayParam.length; i++) {  
        valor *= arrayParam[i];  
    }  
}
```

```
    return valor;
}
```

3. Crie uma função que receba um array e uma das funções com a sintaxe acima e as execute. Faça um exemplo para ambas.

```
void main() {
    printSoma([1, 2, 3], somaArray([1, 2, 3]));
}

int somaArray(List<int> arrayParam) {
    int valor = 0;
    for (int i = 0; i < arrayParam.length; i++) {
        valor += arrayParam[i];
    }
    return valor;
}

void printSoma(List<int> array, int f) {
    print(array);
    print(f);
}
```

4. Crie duas versões de uma função que eleva um número (base) a outro (expoente): a primeira versão é a tradicional, já a segunda deve ser uma arrow function.

```
import "dart:math";

void main() {
    print(funcaoTradicional(3, 2));
    print(funcaoArrowFunction(3, 2));
}

int funcaoTradicional(int x, int y) {
    int r = 1;
    for (int i = 0; i < y; i++) {
        r *= x;
    }
    return r;
}

num funcaoArrowFunction(int x, int y) => pow(x, y);
```

5. Implemente um dos exercícios da disciplina anterior de POO na linguagem DART aplicando ao mesmo tempo: herança, atributos privados, métodos get e/ou set e construtores nomeados. Explique essa questão em vídeo e adicione ao repositório.

```
class Foguete {
    void decolar() {
        print('Iniciando decolagem em 3...2...1...');
    }

    void error() {
        print('Houston, temos um problema!');
    }
}

class NaveEspacial extends Foguete {
    String nome;
    int codigoNave;
    int ano;
    DateTime _dataLancamento;

    NaveEspacial(this.nome, this.codigoNave, this.ano,
this._dataLancamento) {
        nome = nome;
        codigoNave = codigoNave;
        ano = ano;
        _dataLancamento = _dataLancamento;
    }

    NaveEspacial.segundoConstrutor(this.nome, this.codigoNave, this.ano,
this._dataLancamento) {
        print('Iniciando os motores...');
    }

    set setNome(String n) {
        nome = n;
    }

    String get getNome {
        return nome;
    }
}

void main() {
    NaveEspacial viagemEspacial = new NaveEspacial('Voyager I', 82839289,
1977, DateTime(1977, 9, 5));
    NaveEspacial viagemTeste = new NaveEspacial.segundoConstrutor('Voyager
X', 332435, 1999, DateTime(1999, 8, 3));
    viagemEspacial.setNome = 'Saturn VI';

    print(viagemEspacial.getNome);
}
```

```
viagemEspacial.decolar();  
}
```

Link de acesso ao vídeo explicativo: <https://youtu.be/EsaZkh36-fE>