

OAT (25 Pontos)

1) Conversor de Temperatura:

Objetivo: Implementar um programa em Dart que converta temperaturas entre Fahrenheit e Celsius utilizando valores mockados.

Requisitos:

- O programa deve definir uma temperatura em Fahrenheit (ex: 68°F) e convertê-la para Celsius.
- O programa deve definir uma temperatura em Celsius (ex: 20°C) e convertê-la para Fahrenheit.
- Exibir os resultados das conversões.

R=

```
void main() {  
  List<double> grausFare = [68,20];  
  List<double> grausCelsios = [20,68];  
  
  for (double fahrenheit in grausFare){  
    double fahrenheitCelsios = (5 * (fahrenheit - 32))/9;  
    print("${fahrenheit.toStringAsFixed(2)}° Farenhheit é o que  
Corresponde a ${fahrenheitCelsios.toStringAsFixed(2)} Celsios.");  
  };  
  
  for (double celsios in grausCelsios){  
    double celsiosFahrenheit = (9/5) * celsios + 32;  
    print("${celsios.toStringAsFixed(2)}° Celsios é o que Corresponde  
a ${celsiosFahrenheit.toStringAsFixed(2)} Farenheit.");  
  };  
}
```

2) Calculadora de Média de Notas

Objetivo: Desenvolver um algoritmo em Dart que calcule a média das notas de um estudante utilizando valores mockados e determine se ele está aprovado ou reprovado.

Requisitos:

- Definir um array de notas (ex: [7.5, 9.0, 6.8, 8.2]).
- Calcular a média das notas.
- Determinar e exibir se o aluno está aprovado (média >= 6.0) ou reprovado (média < 6.0).

R=

```
void main() {  
  List <double> notas = [7.5, 9.0, 6.8, 8.2];  
  double soma= 0;  
  for(double nota in notas){  
    soma += nota;  
  }  
  double media =(soma)/notas.length;  
  
  if(media>=6.0){  
    print("A media foi ${media.toStringAsFixed(2)}, O aluno está  
provado.");  
  }
```

```

    }else{
        print("A media foi ${media.toStringAsFixed(2)}, O aluno está
Reprovado.");
    }
}

```

3) Verificador de Números Primos

Objetivo: Escrever um programa em Dart que verifique se um número mockado é primo.

Requisitos:

- Definir um número (ex: 29).
- Verificar e exibir se o número é primo.
- Se não for primo, listar e exibir seus divisores.

R=

```

void main() {
    List<int> numeros = [29];

    for (int numero in numeros) {
        bool ePrimo = true;

        if (numero <= 1) {
            ePrimo = false;
        } else {
            for (int i = 2; i * i <= numero; i++) {
                if (numero % i == 0) {
                    ePrimo = false;
                    break;
                }
            }
        }

        if (ePrimo) {
            print("O número $numero é primo");
        } else {
            print("O número $numero não é primo.");
            print("Divisores: ${listarDivisores(numero)}");
        }
    }
}

List<int> listarDivisores(int numero) {
    List<int> divisores = [];
    for (int i = 1; i <= numero; i++) {
        if (numero % i == 0) {
            divisores.add(i);
        }
    }
    return divisores;
}

```

4) Gerador de Fibonacci

Objetivo: Implementar um programa em Dart que gere e exiba a sequência de Fibonacci até um certo número n definido de forma mockada.

Requisitos:

- Definir um número n (ex: 10).
- Gerar e exibir a sequência de Fibonacci até o n-ésimo termo.

R=

```
void main() {  
  int n = 10;  
  List<int> sequencia = [];  
  
  for (int i = 0; i < n; i++) {  
    if (i == 0) {  
      sequencia.add(0);  
    } else if (i == 1) {  
      sequencia.add(1);  
    } else {  
      int proximoTermo = sequencia[i - 1] + sequencia[i - 2];  
      sequencia.add(proximoTermo);  
    }  
  }  
  
  print("Sequência de Fibonacci com $n termos:");  
  print(sequencia.join(', '));  
}
```

5) Cálculo de Desconto de Produto

Objetivo: Criar um programa em Dart que calcule o preço final de um produto após a aplicação de um desconto, usando valores mockados.

Requisitos:

- Definir um preço inicial para um produto (ex: R\$ 120,00).
- Definir um percentual de desconto (ex: 15%).
- Calcular e exibir o preço final após a aplicação do desconto.

R=

```
void main() {  
  double produto = 120.00;  
  double desconto =(produto*15/100);  
  
  double total = produto - desconto;  
  print("Total: $total");  
}
```