

PPDM COM VINI

# AULA 5

# ROTEIRO



- 1 EXERCICIOS FUNÇÕES.
- 2 TRATAMENTOS DE ENTRADA.
- 3 QUIZ.
- 4 MODELO AVALIAÇÃO 1.

# Exercício 1.

- Uma loja está fazendo promoção e quer calcular o preço final de um produto com base no valor original e na porcentagem de desconto.
- Crie uma função chamada calcularDesconto que receba dois parâmetros:
  - O preço original do produto (double preco)
  - A porcentagem de desconto (double desconto)
- A função deve retornar o valor final do produto.
- No main, peça ao usuário o valor e o desconto, chame a função e mostre o preço final.

# Exercício 2.

- Um sistema de cadastro precisa verificar se o cliente é maior de idade.
- Crie uma função chamada ehMaiorIdade que receba a idade (int idade) e retorne true se for maior ou igual a 18, e false caso contrário.
- No main, pergunte ao usuário a idade e mostre uma mensagem adequada usando o retorno da função.

# Exercício 3.

- Um aplicativo de previsão do tempo precisa converter temperaturas de Celsius para Fahrenheit.
- Enunciado:
- Crie uma função chamada celsiusParaFahrenheit que receba a temperatura em Celsius (double celsius) e retorne a temperatura equivalente em Fahrenheit.
- No main, peça a temperatura ao usuário, chame a função e mostre o resultado.

# Tratamento de entrada/erros.

NULL

VALUE ERROR

TRY  
/CATCH/FINALLY

# vazio x null



# null

```
import 'dart:io';

Run | Debug
void main(){
stdout.write('Digite seu nome: ');
String? nome = stdin.readLineSync();

if (nome == null || nome.trim().isEmpty) {
| print('Nome não pode ficar vazio.');
} else {
| print('Olá, $nome!');
}

}
```

# Try/catch/finally

```
import 'dart:io';

Run | Debug
void main() {
  try {
    stdout.write('Digite um número inteiro: ');
    String? entrada = stdin.readLineSync();

    int numero = int.parse(entrada ?? '');
    print('Você digitou: $numero');

  } catch (e) {
    print(' Erro $e: digite apenas números inteiros.');

  } finally {
    print(' Programa encerrado.');
  }
}
```

# TryValue error

```
import 'dart:io';

Run | Debug
void main() {
    int? numero;

    while (numero == null) {
        stdout.write('Digite um número inteiro: ');
        String? entrada = stdin.readLineSync();

        try {
            numero = int.parse(entrada ?? '');
        } on FormatException {
            print(' Formato inválido. Digite apenas números inteiros.\n');
        }
    }

    print(' Número válido recebido: $numero');
}
```

# Range error.

```
src / dartui / Documents / rangeerror.dart / main.dart
// RangeError
// Usada quando um índice ou valor está fora do intervalo permitido.
import 'dart:io';
Run | Debug
void main() {
    List<int> lista = [1, 2, 3];
    try {
        print(lista[5]); // Erro: índice fora do range
    } on RangeError {
        print("Erro: Índice fora do intervalo.");
    }
}
```

# Exercício 1.

- Exercício 1 – Cadastro de Usuário
- Você está desenvolvendo um sistema de cadastro de clientes para uma pequena loja. Ao solicitar o nome completo, você deve garantir que:
- O nome não seja null (problema na leitura).
- O nome não seja vazio ("") e nem apenas espaços (" ").
- Caso o usuário digite algo inválido, mostre uma mensagem de erro e peça novamente.

# Exercício 2.

Você está criando um sistema de pagamento para uma cantina escolar.

Regras para o valor da compra:

1. O usuário deve informar o valor da compra.
2. Esse valor não pode ser null.
3. O valor deve ser numérico e maior que zero.
4. Caso o usuário digite algo que não seja número, o programa deve tratar o erro com `FormatException`, exibir uma mensagem de erro e pedir novamente até o valor ser válido.

Regras para a forma de pagamento:

1. Após validar o valor, o sistema deve perguntar qual será a forma de pagamento.
2. Devem existir 4 opções, por exemplo:
  - 1 – Dinheiro
  - 2 – Cartão de Débito
  - 3 – Cartão de Crédito
  - 4 – Pix
3. Se o usuário digitar uma opção inválida (qualquer coisa diferente de 1 a 4), o programa deve repetir a pergunta até receber uma opção válida.

# AVALIAÇÃO 1

MODELO.

# Capacidades.

4. DEFINIR OS ELEMENTOS DE ENTRADA,  
PROCESSAMENTO E SAÍDA PARA A  
CODIFICAÇÃO  
DAS FUNCIONALIDADES MOBILE

6. IMPLEMENTAR O CÓDIGO RESPEITANDO  
AS CARACTERÍSTICAS DA LINGUAGEM NA  
PLATAFORMA MOBILE

# Critérios

CAPACIDADE	CRITÉRIO
4. Entrada/ Processo/Saída. ▾	Acertando 60% do Quiz Online.
4. Entrada/ Processo/Saída. ▾	Definindo o tipo correto de variável.
4. Entrada/ Processo/Saída. ▾	<i>Definindo quantidade obrigatória de processos.</i>
4. Entrada/ Processo/Saída. ▾	Definindo saídas obrigatórias.
4. Entrada/ Processo/Saída. ▾	Implementando decisões encadeadas.
4. Entrada/ Processo/Saída. ▾	Implementando tratamentos de erros adicionais.
4. Entrada/ Processo/Saída. ▾	Implementando repetições adicionais.
4. Entrada/ Processo/Saída. ▾	Implementando funções adicionais.
4. Entrada/ Processo/Saída. ▾	Entregando atividades dentro do prazo.

# Critérios

## CAPACIDADE

## CRITÉRIO

6.Respeitar Linguagem ▾

Acertando 60% do Quiz Online.

6.Respeitar Linguagem ▾

Implementando linguagem no ambiente Windows.

6.Respeitar Linguagem ▾

Diferenciando características de outras linguagens.

obrigado

FLUTTER COM VINI.