

## IF-ELSE E TRY-CATCH

## EXERCÍCIO 1 –

```
using System;
using System.Threading;

class Calculadora
{
    static void Main()
    {
        while (true)
        {
            Console.Clear();
            Console.WriteLine("+-----+");
            Console.WriteLine("|          CALCULADORA          |");
            Console.WriteLine("+-----+");

            double numero1 = LerNumero("| Digite o primeiro número: ");
            double numero2 = LerNumero("| Digite o segundo número : ");
            string operacao = LerOperacao("| Digite a operação (+, -, *, /): ");

            try
            {
                double resultado = Calcular(numero1, numero2, operacao);

                Console.WriteLine("+-----+");
                Console.WriteLine($"|   Resultado: {resultado}   |");
                Console.WriteLine("+-----+");
            }
            catch (DivideByZeroException ex)
            {
                MostrarErro(ex.Message);
            }
            catch (ArgumentException ex)
            {
                MostrarErro(ex.Message);
            }

            Console.WriteLine("\n| Deseja fazer outra operação? (s/n): ");
            string continuar = Console.ReadLine().ToLower();
            if (continuar != "s")
            {
                Console.WriteLine("\n+-----+");
                Console.WriteLine("|          Saindo...          |");
                Console.WriteLine("+-----+");
                Thread.Sleep(1000);
                break;
            }
        }
    }
}
```

```

static double LerNumero(string mensagem)
{
    double numero;
    while (true)
    {
        Console.Write(mensagem);
        if (double.TryParse(Console.ReadLine(), out numero))
        {
            return numero;
        }
        else
        {
            MostrarErro("Você digitou um número inválido.      ");
        }
    }
}

static string LerOperacao(string mensagem)
{
    while (true)
    {
        Console.Write(mensagem);
        string op = Console.ReadLine();
        if (op == "+" || op == "-" || op == "*" || op == "/")
        {
            return op;
        }
        else
        {
            MostrarErro("Operação inválida. Use apenas +, -, *, /.");
        }
    }
}

static double Calcular(double n1, double n2, string operacao)
{
    return operacao switch
    {
        "+" => n1 + n2,
        "-" => n1 - n2,
        "*" => n1 * n2,
        "/" => n2 != 0 ? n1 / n2 : throw new DivideByZeroException("Não é possível dividir por
zero.      "),
        _ => throw new ArgumentException("Operação inválida.")
    };
}

static void MostrarErro(string mensagem)
{
    Console.WriteLine("+-----+\\n");
    Console.WriteLine("\\n+-----+");
    Console.WriteLine("|                      ERRO!                      |");
    Console.WriteLine("+-----+");
    Console.WriteLine($"| {mensagem} |");
}

```

```
        Console.WriteLine("+-----+\n");
        Thread.Sleep(1500);
    }
}
```

---

---

## EXERCÍCIO 2 –

```
using System;
using System.IO;
using System.Threading;

class LeitorDeArquivo
{
    static void Main()
    {
        while (true)
        {
            Console.Clear();
            MostrarCabecalho();

            string caminho = SolicitarCaminho();

            TentarLerArquivo(caminho);

            if (!DesejaContinuar())
            {
                MostrarSaida();
                break;
            }
        }
    }

    static void MostrarCabecalho()
    {
        Console.WriteLine("+-----+");
        Console.WriteLine("|          LEITOR DE ARQUIVOS          |");
        Console.WriteLine("+-----+");
    }

    static string SolicitarCaminho()
    {
        Console.Write("| Digite o caminho do arquivo: ");
        return Console.ReadLine();
    }

    static void TentarLerArquivo(string caminho)
    {
        try
        {
            string conteudo = LerArquivo(caminho);
            MostrarConteudo(conteudo);
        }
    }
}
```

```

        catch (FileNotFoundException)
        {
            MostrarErro("          Arquivo não encontrado.          ");
        }
        catch (IOException)
        {
            MostrarErro("Erro ao ler o arquivo.          ");
        }
    }

    static string LerArquivo(string caminho)
    {
        return File.ReadAllText(caminho);
    }

    static void MostrarConteudo(string conteudo)
    {
        Console.WriteLine("+-----+");
        Console.WriteLine("| Conteúdo do arquivo:          |");
        Console.WriteLine("+-----+");
        Console.WriteLine(conteudo);
        Console.WriteLine("+-----+");
    }

    static bool DesejaContinuar()
    {
        Console.Write("\n| Deseja tentar ler outro arquivo? (s/n): ");
        string resposta = Console.ReadLine().ToLower();
        return resposta == "s";
    }

    static void MostrarErro(string mensagem)
    {
        Console.WriteLine("\n+-----+");
        Console.WriteLine("|          ERRO!          |");
        Console.WriteLine("+-----+");
        Console.WriteLine($"| {mensagem} |");
        Console.WriteLine("+-----+");
        Thread.Sleep(1500);
    }

    static void MostrarSaida()
    {
        Console.WriteLine("\n+-----+");
        Console.WriteLine("|          Saindo...          |");
        Console.WriteLine("+-----+");
        Thread.Sleep(1000);
    }
}

```

### EXERCÍCIO 3 –

```
using System;
using System.Threading;

class SimuladorDeBancoDeDados
{
    static void Main()
    {
        while (true)
        {
            Console.Clear();
            MostrarCabecalho();

            ConectarBancoDeDados();

            if (!DesejaContinuar())
            {
                MostrarSaida();
                break;
            }
        }
    }

    static void MostrarCabecalho()
    {
        Console.WriteLine("+-----+");
        Console.WriteLine("|          SIMULADOR DE CONEXÃO DB          |");
        Console.WriteLine("+-----+");
    }

    static void ConectarBancoDeDados()
    {
        bool conexaoAberta = false;
        try
        {
            AbrirConexao();
            conexaoAberta = true;
            Console.WriteLine("|      Conexão estabelecida com sucesso!      |");
            Console.WriteLine("+-----+");
        }
        catch (Exception ex)
        {
            MostrarErro(ex.Message);
        }
        finally
        {
            if (conexaoAberta)
            {
                FecharConexao();
            }
            else
            {
                Console.WriteLine("| Tentando fechar conexão (mesmo com erro) |");
                Console.WriteLine("+-----+");
            }
        }
    }
}
```

```

    }
    Thread.Sleep(1000);
}

static void AbrirConexao()
{
    Console.WriteLine("| Tentando conectar ao banco de dados... |");
    Console.WriteLine("+-----+");
    Thread.Sleep(1000);

    Random aleatorio = new Random();
    bool sucesso = aleatorio.Next(0, 2) == 0;

    if (!sucesso)
    {
        throw new Exception(" Falha ao conectar ao banco de dados. ");
    }
}

static void FecharConexao()
{
    Console.WriteLine("| Fechando a conexão com o banco de dados. |");
    Console.WriteLine("+-----+");
}

static bool DesejaContinuar()
{
    Console.WriteLine("\n| Deseja tentar outra conexão? (s/n): ");
    string resposta = Console.ReadLine().ToLower();
    return resposta == "s";
}

static void MostrarErro(string mensagem)
{
    Console.WriteLine("\n+-----+");
    Console.WriteLine("| ERRO! |");
    Console.WriteLine("+-----+");
    Console.WriteLine($"| {mensagem} |");
    Console.WriteLine("+-----+");
    Thread.Sleep(1500);
}

static void MostrarSaida()
{
    Console.WriteLine("\n+-----+");
    Console.WriteLine("| Saindo... |");
    Console.WriteLine("+-----+");
    Thread.Sleep(1000);
}
}

```

## EXERCÍCIO 4 –

```
using System;
using System.Threading;

// Exceção personalizada
class IdadeInvalidaException : Exception
{
    public IdadeInvalidaException(string mensagem) : base(mensagem) { }
}

class ProgramaValidacaoIdade
{
    static void Main()
    {
        while (true)
        {
            Console.Clear();
            MostrarCabecalho();

            try
            {
                int idade = LerIdade();
                MostrarMensagem($"Idade registrada: {idade} anos.");
            }
            catch (IdadeInvalidaException ex)
            {
                MostrarErro(ex.Message);
            }
            catch (FormatException)
            {
                MostrarErro("Formato inválido. Digite apenas números.");
            }

            if (!DesejaContinuar())
            {
                MostrarSaida();
                break;
            }
        }
    }

    static void MostrarCabecalho()
    {
        Console.WriteLine("+-----+");
        Console.WriteLine("|          VALIDAÇÃO DE IDADE          |");
        Console.WriteLine("+-----+");
    }

    static int LerIdade()
    {
        Console.Write("| Digite sua idade: ");
        string entrada = Console.ReadLine();

        if (!int.TryParse(entrada, out int idade))
    }
```

```
{
    throw new FormatException();
}

if (idade < 0 || idade > 120)
{
    throw new IdadeInvalidaException("Idade inválida. Informe entre 0 e 120.");
}

return idade;
}

static bool DesejaContinuar()
{
    Console.Write("\n| Deseja validar outra idade? (s/n): ");
    string resposta = Console.ReadLine().ToLower();
    return resposta == "s";
}

static void MostrarMensagem(string mensagem)
{
    Console.WriteLine("\n+-----+");
    Console.WriteLine($"| {mensagem.PadRight(40)}|");
    Console.WriteLine("+-----+");
    Thread.Sleep(1500);
}

static void MostrarErro(string mensagem)
{
    Console.WriteLine("\n+-----+");
    Console.WriteLine($"|                ERRO!                |");
    Console.WriteLine("+-----+");
    Console.WriteLine($"| {mensagem.PadRight(40)}|");
    Console.WriteLine("+-----+");
    Thread.Sleep(1500);
}

static void MostrarSaida()
{
    Console.WriteLine("\n+-----+");
    Console.WriteLine($"|                Saindo...                |");
    Console.WriteLine("+-----+");
    Thread.Sleep(1000);
}
}
```

---

---

---



## EXERCÍCIO 5 –

```
using System;
using System.Linq;
using System.Threading;

class ProgramaCalculoMedia
{
    static void Main()
    {
        while (true)
        {
            Console.Clear();
            MostrarCabecalho();

            try
            {
                // Solicitar e processar a lista de números
                string entrada = LerNumeros();
                double media = CalcularMedia(entrada);
                MostrarMensagem($"A média dos números é: {media:F2}");
            }
            catch (ArgumentNullException)
            {
                MostrarErro("Entrada vazia. Por favor, digite uma lista de números.");
            }
            catch (FormatException)
            {
                MostrarErro("Formato inválido. Certifique-se de digitar apenas números separados por vírgula.");
            }
            catch (OverflowException)
            {
                MostrarErro("Número muito grande ou muito pequeno. Tente novamente.");
            }

            if (!DesejaContinuar())
            {
                MostrarSaida();
                break;
            }
        }
    }

    static void MostrarCabecalho()
    {
        Console.WriteLine("+-----+");
        Console.WriteLine("|          CÁLCULO DE MÉDIA DE NÚMEROS          |");
        Console.WriteLine("+-----+");
    }

    static string LerNumeros()
    {
        Console.Write("| Digite uma lista de números separados por vírgula: ");
        string entrada = Console.ReadLine();
    }
}
```

```

        if (string.IsNullOrEmpty(entrada))
        {
            throw new ArgumentNullException();
        }

        return entrada;
    }

    static double CalcularMedia(string entrada)
    {
        string[] numerosString = entrada.Split(',');

        double soma = 0;
        int contador = 0;

        foreach (var numero in numerosString)
        {
            if (double.TryParse(numero.Trim(), out double numeroConvertido))
            {
                soma += numeroConvertido;
                contador++;
            }
            else
            {
                throw new FormatException();
            }
        }

        if (contador == 0)
        {
            throw new ArgumentNullException();
        }

        return soma / contador;
    }

    static bool DesejaContinuar()
    {
        Console.WriteLine("\n| Deseja calcular outra média? (s/n): ");
        string resposta = Console.ReadLine().ToLower();
        return resposta == "s";
    }

    static void MostrarMensagem(string mensagem)
    {
        Console.WriteLine("\n+-----+");
        Console.WriteLine($"| {mensagem} |");
        Console.WriteLine("+-----+");
        Thread.Sleep(1500);
    }

    static void MostrarErro(string mensagem)
    {

```

```
    Console.WriteLine("\n+-----+");
    Console.WriteLine("|                ERRO!                |");
    Console.WriteLine("+-----+");
    Console.WriteLine($"| {mensagem} |");
    Console.WriteLine("+-----+");
    Thread.Sleep(1500);
}
```

```
static void MostrarSaida()
```

```
{
    Console.WriteLine("\n+-----+");
    Console.WriteLine("|                Saindo...                |");
    Console.WriteLine("+-----+");
    Thread.Sleep(1000);
}
```

```
}
```