

LISTA 01

EXERCÍCIO 1 -

```
using System;

class Usuario
{
    private string nome;
    private int idade;
    private int altura;

    public string Nome
    {
        get => nome;
        set => nome = ValidarNome(value) ? value
            : throw new ArgumentException("Nome não pode ser vazio ou conter apenas
espacos.");
    }

    public int Idade
    {
        get => idade;
        set => idade = ValidarIdade(value) ? value
            : throw new ArgumentOutOfRangeException("A idade deve estar em 0 e 130
anos.");
    }

    public int Altura
    {
        get => altura;
        set => altura = value;
    }

    public Usuario(string nome, int idade, int altura)
    {
        Nome = nome;
        Idade = idade;
        Altura = altura;
    }

    private static bool ValidarNome(string nome)
    {
        return !string.IsNullOrEmpty(nome);
    }

    private static bool ValidarIdade(int idade)
    {
        return idade >= 0 && idade <= 130;
    }
}
```

```

public void Apresentar()
{
    Console.Clear();
    Console.WriteLine("|=====|");
    Console.WriteLine($"|    Olá, meu nome é {Nome}, tenho {Idade} anos e minha altura
é {Altura} cm.");
    Console.WriteLine("|=====|");
}
}

class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.Clear();
            Console.WriteLine("|=====|");
            Console.WriteLine("|    Cadastro de Usuário    |");
            Console.WriteLine("|-----|");

            string nome = ObterNomeValido();
            int idade = ObterIdadeValida();
            int altura = ObterAlturaValida();

            Usuario usuario = new Usuario(nome, idade, altura);
            usuario.Apresentar();
        }
        catch (Exception ex)
        {
            Console.WriteLine("|-----|");
            Console.WriteLine($"| Erro: {ex.Message} |");
            Console.WriteLine("|-----|");
        }
        finally
        {
            Console.WriteLine("\n\n|-----|");
            Console.WriteLine("| Pressione qualquer tecla para sair... |");
            Console.WriteLine("|-----|");
            Console.WriteLine("|                Saindo...                |");
            Console.WriteLine("|=====|");

            Console.ReadKey();
        }
    }

    private static string ObterNomeValido()
    {
        string nome;
        do
        {
            Console.Write("| Digite o nome: ");

```

```

        nome = Console.ReadLine()?.Trim() ?? string.Empty;

        if (string.IsNullOrEmpty(nome))
        {
            Console.WriteLine("\n|-----|");
            Console.WriteLine("| Nome inválido. Por favor, digite novamente. |");
            Console.WriteLine("|-----|\n");
        }
    } while (string.IsNullOrEmpty(nome));

    return nome;
}

private static int ObterIdadeValida()
{
    int idade;
    bool idadeValida;
    do
    {
        Console.Write("| Digite a idade: ");
        string input = Console.ReadLine();
        idadeValida = int.TryParse(input, out idade) && idade >= 0 && idade <= 130;

        if (!idadeValida)
        {
            Console.WriteLine("\n|-----|");
            Console.WriteLine("| Idade inválida. Por favor, digite um valor entre 0 e 130. |");
            Console.WriteLine("|-----|\n");
        }
    } while (!idadeValida);

    return idade;
}

private static int ObterAlturaValida()
{
    int altura;
    bool alturaValida;
    do
    {
        Console.Write("| Digite a altura (em cm): ");
        string input = Console.ReadLine();
        alturaValida = int.TryParse(input, out altura) && altura > 0;

        if (!alturaValida)
        {
            Console.WriteLine("\n|-----|");
            Console.WriteLine("| Altura inválida. Por favor, digite um valor válido. |");
        }
    } while (!alturaValida);
}

```

```

        Console.WriteLine("|-----|");
    |\n");
    }
    } while (!alturaValida);

    return altura;
}
}
}

```

EXERCÍCIO 2 –

```

using System;
using System.Threading;

class Calculadora
{
    private double numero1;
    private double numero2;

    public double Numero1
    {
        get => numero1;
        set => numero1 = value;
    }

    public double Numero2
    {
        get => numero2;
        set => numero2 = value;
    }

    public Calculadora(double numero1, double numero2)
    {
        Numero1 = numero1;
        Numero2 = numero2;
    }

    public void Calcular()
    {
        Console.Clear();
        Console.WriteLine("|=====|");
    |\n");

        Console.WriteLine($"| Soma:          {Numero1} + {Numero2} = {Numero1 +
Numero2}");
        Console.WriteLine($"| Subtração:      {Numero1} - {Numero2} = {Numero1 -
Numero2}");
        Console.WriteLine($"| Multiplicação: {Numero1} * {Numero2} = {Numero1 *
Numero2}");

        if (Numero2 != 0)

```

```

        Console.WriteLine($"| Divisão:          {Numero1} / {Numero2} = {Numero1 /
Numero2}");
    else
        Console.WriteLine($"| Divisão:          Não é possível dividir por zero.");

    Console.WriteLine($"|=====
");
}
}

class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.Clear();
            Console.WriteLine($"|=====|");
            Console.WriteLine($"|          Calculadora Simples          |");
            Console.WriteLine($"|-----|");

            double numero1 = ObterNumero("primeiro");
            double numero2 = ObterNumero("segundo");

            Calculadora calc = new Calculadora(numero1, numero2);
            calc.Calcular();
        }
        catch (Exception ex)
        {
            Console.WriteLine($"|-----|");
            Console.WriteLine($"| Erro: {ex.Message} |");
            Console.WriteLine($"|-----|");
        }
        finally
        {
            Console.WriteLine($"|\\n\\n|-----|");
            Console.WriteLine($"| Pressione qualquer tecla para sair... |");
            Console.WriteLine($"|-----|");
            Console.ReadKey();
            Console.WriteLine($"|\\n|-----|");
            Console.WriteLine($"|          Saindo...          |");
            Console.WriteLine($"|=====|");
        }
    }
}

private static double ObterNumero(string ordem)
{
    double numero;
    bool valido;
    do
    {
        Console.Write($"| Digite o {ordem} número: ");
        string input = Console.ReadLine();
        valido = double.TryParse(input, out numero);
    }
}

```

```

        if (!valido)
        {
            Console.WriteLine("\n|-----|");
|");
            Console.WriteLine("| Valor inválido. Por favor, digite um número. |");
            Console.WriteLine("|-----|");
|\\n");
            Thread.Sleep(1000);
            Console.Clear();
        }
    } while (!valido);

    return numero;
}
}

```

EXERCÍCIO 3 -

```

using System;
using System.Threading;

class AvaliadorNota
{
    private double nota;

    public double Nota
    {
        get => nota;
        set => nota = ValidarNota(value) ? value
            : throw new ArgumentOutOfRangeException("A nota deve estar entre 0 e 10.");
    }

    public AvaliadorNota(double nota)
    {
        Nota = nota;
    }

    public void Classificar()
    {
        Console.Clear();
        Console.WriteLine($"|=====|");

        if (nota >= 9 && nota <= 10)
            Console.WriteLine($"|      Nota: {nota} - Excelente      |");
        else if (nota >= 7)
            Console.WriteLine($"|      Nota: {nota} - Bom      |");
        else if (nota >= 5)
            Console.WriteLine($"|      Nota: {nota} - Regular      |");
        else
            Console.WriteLine($"|      Nota: {nota} - Insuficiente      |");
        Console.WriteLine($"|=====|");
    }
}

```

```

private static bool ValidarNota(double valor)
{
    return valor >= 0 && valor <= 10;
}

}

class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.Clear();
            Console.WriteLine("|=====|");
            Console.WriteLine("|          Avaliação de Nota          |");
            Console.WriteLine("|-----|");

            double nota = ObterNotaValida();

            AvaliadorNota avaliador = new AvaliadorNota(nota);
            avaliador.Classificar();
        }
        catch (Exception ex)
        {
            Console.WriteLine("|-----|");
            Console.WriteLine($"| Erro: {ex.Message} |");
            Console.WriteLine("|-----|");
        }
        finally
        {
            Console.WriteLine("\n\n|-----|");
            Console.WriteLine("| Pressione qualquer tecla para sair... |");
            Console.WriteLine("|-----|");
            Console.ReadKey();
            Console.WriteLine("\n|-----|");
            Console.WriteLine("|          Saindo...          |");
            Console.WriteLine("|=====|");
        }
    }
}

private static double ObterNotaValida()
{
    double nota;
    bool valido;
    do
    {
        Console.Write("| Digite uma nota entre 0 e 10: ");
        string input = Console.ReadLine();
        valido = double.TryParse(input, out nota) && nota >= 0 && nota <= 10;

        if (!valido)

```

```
        {
            Console.WriteLine("\n|-----|");
            Console.WriteLine("|          Nota inválida. Por favor, digite entre 0 e
10.          |");
            Console.WriteLine("|-----|");
            Thread.Sleep(1000);
            Console.Clear();
        }
    } while (!valido);

    return nota;
}
```