

LISTA 02

EXERCÍCIO 4 –

```
using System;
using System.Threading;
using System.Text;

class Produto
{
    private string nome;
    private double preco;

    public Produto(string nome, double preco)
    {
        this.nome = nome;
        this.preco = preco;
    }

    public string GetNome()
    {
        return nome;
    }

    public void SetNome(string valor)
    {
        nome = valor;
    }

    public double GetPreco()
    {
        return preco;
    }

    public void SetPreco(double valor)
    {
        preco = valor;
    }
}

class Program
{
    static void Main(string[] args)
    {
        Console.OutputEncoding = Encoding.UTF8;

        try
        {
            Produto[] produtos = new Produto[5];

            for (int i = 0; i < 5; i++)
            {
                Console.Clear();
                Console.WriteLine("+-----+");
                Console.WriteLine("|          Cadastro do Produto " + (i + 1) + "          |");
                Console.WriteLine("+-----+");
            }
        }
    }
}
```

```

        Console.WriteLine("| Nome do produto: ");
        string nome = Console.ReadLine();

        double preco = ObterPrecoValido();

        produtos[i] = new Produto(nome, preco);

        Console.WriteLine("| Produto cadastrado com sucesso!");
        Thread.Sleep(1000);
    }

    double media = CalcularMediaPreco(produtos);
    Produto maisCaro = ObterProdutoMaisCaro(produtos);

    Console.Clear();
    Console.WriteLine("+-----+");
    Console.WriteLine("|                      Resumo dos Produtos                      |");
    Console.WriteLine("+-----+");
    Console.WriteLine("\n|          Produto mais caro: " + maisCaro.GetNome() + " - R$" +
maisCaro.GetPreco().ToString("F2"));
    Console.WriteLine("|          Média de preços: R$" + media.ToString("F2"));
    Console.WriteLine("\n|          Produtos acima da média de preço:");

    foreach (var produto in produtos)
    {
        if (produto.GetPreco() > media)
        {
            Console.WriteLine("|          " + produto.GetNome() + " - R$" + produto.GetPreco().ToString("F2"));
            Thread.Sleep(100);
        }
    }

    Console.WriteLine("+-----+");
}
catch (Exception ex)
{
    Console.WriteLine("Erro: " + ex.Message);
}

Console.WriteLine();
Console.WriteLine("Pressione qualquer tecla para sair...");
Console.ReadKey();
}

private static double ObterPrecoValido()
{
    double preco;
    bool precoValido;

    do
    {
        Console.WriteLine("| Preço do produto: R$");
        string input = Console.ReadLine();

        precoValido = double.TryParse(input, out preco) && preco >= 0;

        if (!precoValido)

```

```

        {
            Console.WriteLine("| Preço inválido. Digite um valor válido.");
            Thread.Sleep(1000);
        }

    } while (!precoValido);

    return preco;
}

private static double CalcularMediaPreco(Produto[] produtos)
{
    double soma = 0;
    foreach (var produto in produtos)
    {
        soma += produto.GetPreco();
    }
    return soma / produtos.Length;
}

private static Produto ObterProdutoMaisCaro(Produto[] produtos)
{
    Produto maisCaro = produtos[0];
    foreach (var produto in produtos)
    {
        if (produto.GetPreco() > maisCaro.GetPreco())
        {
            maisCaro = produto;
        }
    }
    return maisCaro;
}
}

```

EXERCÍCIO 5 –

```

using System;
using System.Numerics;
using System.Threading;

class CalculadoraCientifica
{
    private double baseNum;
    private double expoente;
    private double numero;
    private int numeroInteiro;

    public double GetBaseNum()
    {
        return baseNum;
    }

    public void SetBaseNum(double valor)
    {
        baseNum = valor;
    }
}

```

```
public double GetExpoente()
{
    return expoente;
}

public void SetExpoente(double valor)
{
    expoente = valor;
}

public double GetNumero()
{
    return numero;
}

public void SetNumero(double valor)
{
    numero = valor;
}

public int GetNumeroInteiro()
{
    return numeroInteiro;
}

public void SetNumeroInteiro(int valor)
{
    numeroInteiro = valor;
}

public double Potencia()
{
    return Math.Pow(baseNum, expoente);
}

public double RaizQuadrada()
{
    if (numero < 0)
    {
        throw new ArgumentException("Não é possível calcular a raiz quadrada de número negativo.");
    }
    return Math.Sqrt(numero);
}

public BigInteger Fatorial()
{
    if (numeroInteiro < 0)
    {
        throw new ArgumentException("Não é possível calcular o fatorial de número negativo.");
    }
    BigInteger resultado = 1;
    for (int n = 1; n <= numeroInteiro; n++)
    {
        resultado *= n;
    }
    return resultado;
}
```

```
}

class Program
{
    static void Main(string[] args)
    {
        try
        {
            CalculadoraCientifica calc = new CalculadoraCientifica();

            while (true)
            {
                Console.Clear();
                Console.WriteLine("+-----+");
                Console.WriteLine("|          CALCULADORA CIENTÍFICA BÁSICA          |");
                Console.WriteLine("+-----+");
                Console.WriteLine("| 1 - Calcular Potência                            |");
                Console.WriteLine("| 2 - Calcular Raiz Quadrada                        |");
                Console.WriteLine("| 3 - Calcular Fatorial                            |");
                Console.WriteLine("| 0 - Sair                                          |");
                Console.WriteLine("+-----+");
                Console.Write("Escolha uma opção: ");

                string opcao = Console.ReadLine();
                Console.Clear();

                switch (opcao)
                {
                    case "1":
                        Console.WriteLine("+-----+");
                        Console.WriteLine("|          Cálculo escolhido: Potência          |");
                        Console.WriteLine("+-----+");
                        Console.Write("| Digite a base: ");
                        calc.SetBaseNum(double.Parse(Console.ReadLine()));

                        Console.Write("| Digite o expoente: ");
                        calc.SetExpoente(double.Parse(Console.ReadLine()));

                        Console.WriteLine($"| Resultado: {calc.GetBaseNum()}^{calc.GetExpoente()} = {calc.Potencia()}");
                        Console.WriteLine("+-----+");
                        break;

                    case "2":
                        Console.WriteLine("+-----+");
                        Console.WriteLine("|          Cálculo escolhido: Raiz Quadrada      |");
                        Console.WriteLine("+-----+");
                        Console.Write("| Digite o número: ");
                        calc.SetNumero(double.Parse(Console.ReadLine()));

                        Console.WriteLine($"| Resultado: √{calc.GetNumero()} = {calc.RaizQuadrada()}");
                        Console.WriteLine("+-----+");
                        break;

                    case "3":
                        Console.WriteLine("+-----+");
                        Console.WriteLine("|          Cálculo escolhido: Fatorial          |");
                        Console.WriteLine("+-----+");
                        Console.Write("| Digite um número inteiro: ");
```

```

        calc.SetNumeroInteiro(int.Parse(Console.ReadLine()));

        Console.WriteLine($"| Resultado: {calc.GetNumeroInteiro()}! = {calc.Fatorial()}");
        Console.WriteLine("+-----+");
        break;

    case "0":
        Console.WriteLine("+-----+");
        Console.WriteLine("|          Encerrando o programa...          |");
        Console.WriteLine("+-----+");
        Thread.Sleep(1000);
        return;

    default:
        Console.WriteLine("| Opção inválida.");
        break;
}

Console.WriteLine();
Console.WriteLine("\n\n+-----+");
Console.WriteLine("|   Pressione qualquer tecla para continuar...   |");
Console.WriteLine("+-----+");
Console.ReadKey();
}
}
catch (Exception ex)
{
    Console.WriteLine("| Erro: " + ex.Message);
}

Console.WriteLine();
Console.WriteLine("| Pressione qualquer tecla para sair...");
Console.ReadKey();
}
}

```

EXERCÍCIO 6 -

```

using System;
using System.Text;
using System.Threading;

class ContaBancaria
{
    private string numeroConta;
    private string titular;
    private double saldo;

    public ContaBancaria(string numeroConta, string titular)
    {
        this.numeroConta = numeroConta;
        this.titular = titular;
        this.saldo = 0;
    }

    public string GetNumeroConta()

```

```
{
    return numeroConta;
}

public void SetNumeroConta(string valor)
{
    numeroConta = valor;
}

public string GetTitular()
{
    return titular;
}

public void SetTitular(string valor)
{
    titular = valor;
}

public double GetSaldo()
{
    return saldo;
}

public void Depositar(double valor)
{
    if (valor > 0)
    {
        saldo += valor;
        Console.WriteLine("+-----+");
        Console.WriteLine("|      Depósito realizado com sucesso!      |");
        Console.WriteLine("+-----+");
    }
    else
    {
        Console.WriteLine("+-----+");
        Console.WriteLine("|      Valor inválido para depósito.      |");
        Console.WriteLine("+-----+");
    }
}

public void Sacar(double valor)
{
    if (valor > 0 && valor <= saldo)
    {
        saldo -= valor;
        Console.WriteLine("+-----+");
        Console.WriteLine("|      Saque realizado com sucesso!      |");
        Console.WriteLine("+-----+");
    }
    else
    {
        Console.WriteLine("+-----+");
        Console.WriteLine("|      Valor inválido ou saldo insuficiente.      |");
        Console.WriteLine("+-----+");
    }
}
```

```

public void ExibirSaldo()
{
    Console.WriteLine($"| Saldo atual: R${saldo:F2}");
    Console.WriteLine("+-----+");
}
}

```

```

class Program
{
    static void Main(string[] args)
    {
        Console.OutputEncoding = Encoding.UTF8;
        Console.InputEncoding = Encoding.UTF8;

        Console.Clear();
        Console.WriteLine("+-----+");
        Console.WriteLine("|          Cadastro de Conta Bancária          |");
        Console.WriteLine("+-----+");

        Console.Write("| Número da conta: ");
        string numero = Console.ReadLine();

        Console.Write("| Nome do titular: ");
        string nome = Console.ReadLine();

        ContaBancaria conta = new ContaBancaria(numero, nome);

        int opcao;
        do
        {
            Console.Clear();
            Console.WriteLine("+-----+");
            Console.WriteLine("|          Menu da Conta          |");
            Console.WriteLine("+-----+");
            Console.WriteLine("| 1 - Depositar");
            Console.WriteLine("| 2 - Sacar");
            Console.WriteLine("| 3 - Exibir Saldo");
            Console.WriteLine("| 0 - Sair");
            Console.WriteLine("+-----+");
            Console.Write("| Escolha uma opção: ");

            string entrada = Console.ReadLine();
            bool valida = int.TryParse(entrada, out opcao);

            if (!valida)
            {
                Console.Clear();
                Console.WriteLine("| Opção inválida!");
                Thread.Sleep(1000);
                continue;
            }

            Console.Clear();
            switch (opcao)
            {
                case 1:
                    Console.Clear();
                    Console.WriteLine("+-----+");

```



```

        Console.WriteLine("|          Opção escolhida: depósito          |");
        Console.WriteLine("+-----+");
        Console.Write("| Valor para depósito: R$");
        double valorDeposito = double.Parse(Console.ReadLine());
        conta.Depositar(valorDeposito);
        Thread.Sleep(1000);
        break;

    case 2:
        Console.Clear();
        Console.WriteLine("+-----+");
        Console.WriteLine("|          Opção escolhida: saque          |");
        Console.WriteLine("+-----+");
        Console.Write("| Valor para saque: R$");
        double valorSaque = double.Parse(Console.ReadLine());
        conta.Sacar(valorSaque);
        Thread.Sleep(1000);
        break;

    case 3:
        Console.Clear();
        Console.WriteLine("+-----+");
        Console.WriteLine("|          Opção escolhida: Verificar Saldo          |");
        Console.WriteLine("+-----+");
        Console.WriteLine($"| Titular: {conta.GetTitular()}");
        Console.WriteLine($"| Número da conta: {conta.GetNumeroConta()}");
        Console.WriteLine("+-----+");
        conta.ExibirSaldo();
        break;

    case 0:
        Console.Clear();
        Console.WriteLine("+-----+");
        Console.WriteLine("|          Encerrando o programa...          |");
        Console.WriteLine("+-----+");
        return;

    default:
        Console.Clear();
        Console.WriteLine("| Opção inválida.");
        break;
}

Console.WriteLine();
Console.WriteLine("\n\n+-----+");
Console.WriteLine("| Pressione qualquer tecla para continuar... |");
Console.WriteLine("+-----+");
Console.ReadKey();

} while (true);
}
}

```