

## EXERCÍCIO 1 –

```
using System;
using System.Threading;

class ContaBancaria
{
    private decimal saldo;
    private string numeroConta = "";
    private string titular = "";

    public decimal Saldo
    {
        get => saldo;
        private set => saldo = value;
    }

    public string Titular
    {
        get => titular;
        set => titular = value;
    }

    public string NumeroConta
    {
        get => numeroConta;
        set => numeroConta = value;
    }

    public void Depositar(decimal valor)
    {
        if (valor > 0)
        {
            saldo += valor;
            Console.WriteLine("+-----+");
            Console.WriteLine("|    Depósito realizado com sucesso!    |");
            Console.WriteLine("+-----+");
        }
        else
        {
            Console.WriteLine("+-----+");
            Console.WriteLine("|          Insira um valor válido!          |");
            Console.WriteLine("+-----+");
        }
    }

    public void Sacar(decimal valor)
    {
        if (valor <= saldo && valor > 0)
```

```

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

class Program
{
    static void Main()
    {
        ContaBancaria contaBancaria = new ContaBancaria();
        int opcao = 0;

        Console.Clear();
        Console.WriteLine("+-----+");
        Console.WriteLine("|      Sistema bancário      |");
        Console.WriteLine("+-----+");
        Console.WriteLine("| Insira o número da conta: \n| ");
        contaBancaria.NumeroConta = Console.ReadLine();
        Console.WriteLine("| Insira o nome do titular: \n| ");
        contaBancaria.Titular = Console.ReadLine();
        Console.WriteLine("\n+-----+");
        Console.WriteLine("|      Conta criada com sucesso!      |");
        Console.WriteLine("+-----+");

        do
        {
            Console.Clear();
            Console.WriteLine("+-----+");
            Console.WriteLine($"| Menu da Conta Nº {contaBancaria.NumeroConta}|");
            Console.WriteLine("| Titular: " + contaBancaria.Titular);
            Console.WriteLine("+-----+");
            Console.WriteLine("| 1 - Depositar      |");
            Console.WriteLine("| 2 - Sacar          |");
            Console.WriteLine("| 3 - Exibir Saldo   |");
            Console.WriteLine("| 0 - Sair           |");
            Console.WriteLine("+-----+");

```

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

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

switch (opcao)
{
    case 1:
        try
        {
            Console.Clear();
            Console.WriteLine("+-----+");
            Console.WriteLine("|          Opção escolhida: Depósito          |");
            Console.WriteLine("+-----+");
            Console.Write("| Valor para depósito: R$");
            decimal valorDeposito = Decimal.Parse(Console.ReadLine());
            contaBancaria.Depositar(valorDeposito);
        }
        catch (FormatException)
        {
            Console.WriteLine("| Erro: valor inválido! Use apenas números.");
        }
        catch (Exception ex)
        {
            Console.WriteLine($"| Erro inesperado: {ex.Message}");
        }
        Thread.Sleep(1500);
        break;

    case 2:
        try
        {
            Console.Clear();
            Console.WriteLine("+-----+");
            Console.WriteLine("|          Opção escolhida: Saque          |");
            Console.WriteLine("+-----+");
            Console.Write("| Valor para saque: R$");
            decimal valorSaque = Decimal.Parse(Console.ReadLine());
            contaBancaria.Sacar(valorSaque);
        }
        catch (FormatException)
        {
            Console.WriteLine("| Erro: valor inválido! Use apenas números.");
        }
        catch (Exception ex)
        {
            Console.WriteLine($"| Erro inesperado: {ex.Message}");
        }
}
```

```

        Thread.Sleep(1500);
        break;

    case 3:
        Console.Clear();
        contaBancaria.VerificarSaldo();
        break;

    case 0:
        Console.Clear();
        Console.WriteLine("+-----+");
        Console.WriteLine("|      Saindo...      |");
        Console.WriteLine("+-----+");
        return;

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

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

} while (true);
}
}

```

## EXERCÍCIO 2 –

```

using System;
using System.Threading;

class Temperatura
{
    private double temperaturaCelsius;

    public double Celsius
    {
        get => temperaturaCelsius;
        set
        {
            if (value < -273.15)
                throw new ArgumentOutOfRangeException("A temperatura não pode ser menor que -273.15°C (zero absoluto).");
            temperaturaCelsius = value;
        }
    }
}

```

```

    }

    public double Fahrenheit
    {
        get => temperaturaCelsius * 1.8 + 32;
    }
}

class Program
{
    static void Main()
    {
        Temperatura temperatura = new Temperatura();

        Console.Clear();
        Console.WriteLine("+-----+");
        Console.WriteLine("|  Insira a temperatura em Celsius  |");
        Console.Write("+-----+\n|  ");
        string entrada = Console.ReadLine();

        if (double.TryParse(entrada, out double tempCelsius))
        {
            try
            {
                temperatura.Celsius = tempCelsius;

                Console.Clear();
                Console.WriteLine("+-----+");
                Console.WriteLine($"|  Temperatura em Fahrenheit:");
{temperatura.Fahrenheit,6:F2}°F          ");
                Console.WriteLine("+-----+");
            }
            catch (ArgumentOutOfRangeException ex)
            {
                Console.WriteLine($"|  Erro: {ex.Message}");
            }
        }
        else
        {
            Console.Clear();
            Console.WriteLine("|  Erro: Valor inválido! Use apenas números.");
        }

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

```

### EXERCÍCIO 3 –

```
using System;
using System.Threading;
using System.Collections.Generic;
using System.Linq;

class Aluno
{
    private List<double> notas = new List<double>();

    public double Media
    {
        get
        {
            if (notas.Count == 0)
                return 0;
            return notas.Average();
        }
    }

    public void AdicionarNota(double nota)
    {
        if (nota >= 0 && nota <= 10)
        {
            notas.Add(nota);
            Console.WriteLine("+-----+");
            Console.WriteLine($"|    Nota {nota} adicionada com sucesso.    |");
            Console.WriteLine("+-----+");
        }
        else
        {
            Console.WriteLine("+-----+");
            Console.WriteLine("| Nota inválida. Use valor entre 0 e 10 |");
            Console.WriteLine("+-----+");
        }
    }

    public void RemoverUltimaNota()
    {
        if (notas.Count > 0)
        {
            double removida = notas[notas.Count - 1];
            notas.RemoveAt(notas.Count - 1);
            Console.WriteLine("+-----+");
            Console.WriteLine($"|    Nota {removida} removida da lista.    |");
            Console.WriteLine("+-----+");
        }
        else
        {
            Console.WriteLine("+-----+");
            Console.WriteLine("|    Nenhuma nota para remover.    |");
            Console.WriteLine("+-----+");
        }
    }
}
```

```

    public void ExibirMedia()
    {
        Console.WriteLine("+-----+");
        Console.WriteLine($"|          Média das notas: {Media:F2}          |");
        Console.WriteLine("+-----+");
    }
}

class Program
{
    static void Main()
    {
        Aluno aluno = new Aluno();
        int opcao = 0;

        do
        {
            Console.Clear();
            Console.WriteLine("+-----+");
            Console.WriteLine("|          Sistema de Notas - Aluno          |");
            Console.WriteLine("+-----+");
            Console.WriteLine("| 1 - Adicionar nota                          |");
            Console.WriteLine("| 2 - Remover última nota                      |");
            Console.WriteLine("| 3 - Exibir média                          |");
            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;
            }

            switch (opcao)
            {
                case 1:
                    try
                    {
                        Console.Clear();
                        Console.WriteLine("+-----+");
                        Console.WriteLine("|          Adicionar Nota          |");
                        Console.WriteLine("+-----+");
                        Console.Write("| Digite a nota (0 a 10): ");
                        double nota = double.Parse(Console.ReadLine());
                        aluno.AdicionarNota(nota);
                    }
                    catch (FormatException)
                    {

```

```

        Console.WriteLine("| Erro: Digite um número válido.");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"| Erro inesperado: {ex.Message}");
    }
    Thread.Sleep(1500);
    break;

case 2:
    Console.Clear();
    aluno.RemoverUltimaNota();
    Thread.Sleep(1500);
    break;

case 3:
    Console.Clear();
    aluno.ExibirMedia();
    Console.WriteLine();
    break;

case 0:
    Console.Clear();
    Console.WriteLine("+-----+");
    Console.WriteLine("|      Saindo...      |");
    Console.WriteLine("+-----+");
    return;

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

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

} while (true);
}
}

```

#### EXERCÍCIO 4 –

```

using System;
using System.Collections.Generic;
using System.Threading;

class Livro

```



```

{
    private string titulo;
    private string isbn;
    private static List<string> IsbnsCadastrados = new List<string>();

    public string Titulo
    {
        get => titulo;
        set
        {
            if (string.IsNullOrEmpty(value))
            {
                Console.WriteLine("+-----+");
                Console.WriteLine("|      Título não pode ser vazio!      |");
                Console.WriteLine("+-----+");
            }
            else
            {
                titulo = value;
                Console.WriteLine("+-----+");
                Console.WriteLine("|      Título definido com sucesso      |");
                Console.WriteLine("+-----+");
            }
        }
    }

    public string ISBN => isbn;

    public Livro(string titulo, string isbn)
    {
        if (!ValidarISBN(isbn))
        {
            Console.WriteLine("+-----+");
            Console.WriteLine("| ISBN inválido! Deve conter 13 números. |");
            Console.WriteLine("+-----+");
            return;
        }

        if (IsbnsCadastrados.Contains(isbn))
        {
            Console.WriteLine("+-----+");
            Console.WriteLine("| ISBN já cadastrado em outro livro! |");
            Console.WriteLine("+-----+");
            return;
        }

        this.titulo = titulo;
        this.isbn = isbn;
        IsbnsCadastrados.Add(isbn);
        Console.WriteLine("+-----+");
        Console.WriteLine("|      Livro criado com sucesso!      |");
        Console.WriteLine("+-----+");
    }
}

```

```

private bool ValidarISBN(string isbn)
{
    return isbn.Length == 13 && long.TryParse(isbn, out _);
}

}

class Program
{
    static List<Livro> livros = new List<Livro>();

    static void Main()
    {
        int opcao = 0;

        do
        {
            Console.Clear();
            Console.WriteLine("+-----+");
            Console.WriteLine("|          Sistema de Livros          |");
            Console.WriteLine("+-----+");
            Console.WriteLine("| 1 - Adicionar livro                |");
            Console.WriteLine("| 2 - Listar livros                  |");
            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;
            }

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

                    Console.Write("| Título: ");
                    string titulo = Console.ReadLine();

                    Console.Write("| ISBN (13 dígitos): ");
                    string isbn = Console.ReadLine();

                    Livro novoLivro = new Livro(titulo, isbn);
                    if (!string.IsNullOrEmpty(novoLivro.Titulo) && novoLivro.ISBN != null)
                    {
                        livros.Add(novoLivro);
                    }
                }
            }
        }
    }
}

```

```

    }

    Thread.Sleep(1500);
    break;

case 2:
    Console.Clear();
    if (livros.Count == 0)
    {
        Console.WriteLine("+-----+");
        Console.WriteLine("|      Nenhum livro cadastrado.      |");
        Console.WriteLine("+-----+");
    }
    else
    {
        Console.WriteLine("+----- Livros -----+");
        for (int i = 0; i < livros.Count; i++)
        {
            Console.WriteLine($"| {i + 1}. Título: {livros[i].Titulo}|");
            Console.WriteLine($"|      ISBN: {livros[i].ISBN}|");
            Console.WriteLine("+-----+");
        }
    }

    break;

case 0:
    Console.Clear();
    Console.WriteLine("+-----+");
    Console.WriteLine("|      Saindo...      |");
    Console.WriteLine("+-----+");
    return;

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

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

} while (true);
}
}

```

## EXERCÍCIO 5 –

```
using System;
using System.Collections.Generic;
using System.Threading;

class Playlist
{
    private List<(string Nome, int Duracao)> musicas = new List<(string, int)>();

    public int DuracaoTotal
    {
        get
        {
            int duracaoTotal = 0;
            foreach (var musica in musicas)
            {
                duracaoTotal += musica.Duracao;
            }
            return duracaoTotal;
        }
    }

    public void AdicionarMusica(string nome, int duracaoSegundos)
    {
        if (musicas.Exists(m => m.Nome == nome))
        {
            Console.WriteLine("+-----+");
            Console.WriteLine("| Música já adicionada na playlist! |");
            Console.WriteLine("+-----+");
            return;
        }

        if (duracaoSegundos <= 0)
        {
            Console.WriteLine("+-----+");
            Console.WriteLine("| Duração inválida! Use um valor maior que zero.");
            Console.WriteLine("+-----+");
            return;
        }

        musicas.Add((nome, duracaoSegundos));
        Console.WriteLine("+-----+");
        Console.WriteLine($"| Música '{nome}' adicionada com sucesso! |");
        Console.WriteLine("+-----+");
    }

    public void RemoverMusica(string nome)
    {
        var musicaRemovida = musicas.Find(m => m.Nome == nome);
        if (musicaRemovida.Nome == null)
        {
            Console.WriteLine("+-----+");
            Console.WriteLine("| Música não encontrada na playlist! |");
            Console.WriteLine("+-----+");
        }
    }
}
```

```

        return;
    }

    musicas.Remove(musicaRemovida);
    Console.WriteLine("+-----+");
    Console.WriteLine($"| Música '{nome}' removida da playlist!");
    Console.WriteLine("+-----+");
}

public void ListarMusicas()
{
    if (musicas.Count == 0)
    {
        Console.WriteLine("+-----+");
        Console.WriteLine("| Playlist vazia! |");
        Console.WriteLine("+-----+");
        return;
    }

    Console.WriteLine("+----- Playlist -----+");
    foreach (var musica in musicas)
    {
        Console.WriteLine($"| {musica.Nome} - {musica.Duracao} segundos");
    }
    Console.WriteLine("+-----+");
}
}

class Program
{
    static void Main()
    {
        Playlist playlist = new Playlist();
        int opcao = 0;

        do
        {
            Console.Clear();
            Console.WriteLine("+-----+");
            Console.WriteLine("| Gerenciador de Playlist |");
            Console.WriteLine("+-----+");
            Console.WriteLine("| 1 - Adicionar música |");
            Console.WriteLine("| 2 - Remover música |");
            Console.WriteLine("| 3 - Listar músicas |");
            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;
    }

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

                Console.Write("| Nome da música: ");
                string nomeMusica = Console.ReadLine();

                Console.Write("| Duração da música (em segundos): ");
                int duracaoMusica = int.Parse(Console.ReadLine());

                playlist.AdicionarMusica(nomeMusica, duracaoMusica);
            }
            catch (FormatException)
            {
                Console.WriteLine("| Erro: Duração deve ser um número inteiro.");
            }
            Thread.Sleep(1500);
            break;

        case 2:
            Console.Clear();
            Console.WriteLine("+-----+");
            Console.WriteLine("|          Remover Música          |");
            Console.WriteLine("+-----+");

            Console.Write("| Nome da música: ");
            string nomeRemover = Console.ReadLine();

            playlist.RemoverMusica(nomeRemover);
            Thread.Sleep(1500);
            break;

        case 3:
            Console.Clear();
            playlist.ListarMusicas();
            Thread.Sleep(1500);
            break;

        case 0:
            Console.Clear();
            Console.WriteLine("+-----+");
            Console.WriteLine("|          Saindo...          |");
            Console.WriteLine("+-----+");
            return;
    }
}

```

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

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

} while (true);
}
```