

# Exercícios Teóricos – u02: Estruturas Lineares

Catarina F. M. Castro (803531) – AEDs II

## Exercícios resolvidos

1-

- Implemente uma classe lista sequencial sem usar o *ArrayList* e que permita a execução do código abaixo visto na Seção 1 da Unidade 1

```
Console.WriteLine("==== LISTA LINEAR ====");
Lista lista = new Lista(6);
int x1, x2, x3;
lista.InserirInicio(1);
lista.InserirFim(7);
lista.InserirFim(9);
lista.InserirInicio(3);
lista.Inserir(8, 3);
lista.Inserir(4, 2);

lista.Mostrar();

x1 = lista.RemoverInicio();
x2 = lista.RemoverFim();
x3 = lista.Remover(2);

Console.WriteLine(x1 + ", " + x2 + ", " + x3);
lista.Mostrar();
```

```
class Lista {
    // Declaracao de variáveis
    int[] array;
    int n;

    // Construtor sem parâmetros
    Lista () {
        array = new int[6];
        n = 0;
    }

    // Construtor com parâmetros
    Lista (int tamanho){
        array = new int[tamanho];
        n = 0;
    }

    // Método para inserir início
    void InserirInicio(int x) {
        if (n >= array.Length)
            Environment.Exit(0);

        //levar elementos para o fim do array
        for (int i = n; i > 0; i--){
            array[i] = array[i-1];
        }
        array[0] = x;
        n++;
    }
}
```

```

// Método para inserir início
void InserirFim(int x) {
    if (n >= array.Length)
        Environment.Exit(0);
    array[n] = x;
    n++;
}

// Método para inserir elemento no
void Inserir(int x, int pos) {
    if (n >= array.Length || pos < 0 || pos > n)
        Environment.Exit(0);

    //levar elementos para o fim do array
    for (int i = n; i > pos; i--){
        array[i] = array[i-1];
    }
    array[pos] = x;
    n++;
}

// Método para remover início
int RemoverInicio() {
    if (n == 0)
        Environment.Exit(0);
    int resp = array[0];
    n--;
    for (int i = 0; i < n; i++){
        array[i] = array[i+1];
    }
    return resp;
}

// Método para remover o fim
int RemoverFim() {
    if (n == 0)
        Environment.Exit(0);
    return array[--n];
}

// Método remover
int Remover(int pos) {
    if (n == 0 || pos < 0 || pos >= n)
        Environment.Exit(0);
    int resp = array[pos];
    n--;
    for (int i = pos; i < n; i++){
        array[i] = array[i+1];
    }
    return resp;
}

```

```

    }

    // Método Mostrar o array na tela
    void Mostrar() {

        Console.Write("[ ");
        for (int i = 0; i < n; i++){
            Console.Write(array[i] + " ");
        }
        Console.WriteLine(" ]");
    }
}

```

## *Exercício complementar*

### **Exercícios Introdução**

**1-**

```

using System;
using System.Collections;

class Fila {
    ArrayList arrayList;
    public Fila (){
        arrayList = new ArrayList();
    }
    public void Inserir(int valor){
        arrayList.Add(valor);
    }
    public int Remover(){
        int resp = (int)arrayList[0];
        arrayList.RemoveAt(0);
        return resp;
    }
    public void Mostrar(){
        foreach (int i in arrayList){
            Console.WriteLine(i);
        }
    }
}

class ExercicioComplementarU02S01EC01 {
    public static void Main (string[] args) {
        Fila fila = new Fila();

        for(int i = 0; i < 10; i++){
            fila.Inserir(i);
        }
        Console.WriteLine(fila.Remover());
        fila.Inserir(10);
        Console.WriteLine(fila.Remover());
        fila.Inserir(11);
        Console.WriteLine(fila.Remover());
        fila.Inserir(12);
    }
}

```

```

        Console.WriteLine("===== Mostrar");
        fila.Mostrar();
    }
}

```

2-

```

using System;
using System.Collections;

//Faça uma classe Pilha contendo um atributo do tipo ArrayList, os
métodos inserir, remover e mostrar os elementos da sua pilha.
class Pilha {
    ArrayList arrayList;
    public Pilha (){
        arrayList = new ArrayList();
    }
    public void Inserir(int valor){
        arrayList.Add(valor);
    }
    public int Remover(){
        int resp = (int)arrayList[arrayList.Count-1];
        arrayList.RemoveAt(arrayList.Count-1);
        return resp;
    }
    public void Mostrar(){
        for (int i = arrayList.Count - 1; i >= 0; i--){
            Console.WriteLine(arrayList[i]);
        }
    }
}

class ExercicioComplementarU02S01EC02 {
    public static void Main (string[] args) {
        Pilha pilha = new Pilha();

        for(int i = 0; i < 10; i++){
            pilha.Inserir(i);
        }
        Console.WriteLine(pilha.Remover());
        pilha.Inserir(10);
        Console.WriteLine(pilha.Remover());
        pilha.Inserir(11);
        Console.WriteLine(pilha.Remover());
        pilha.Inserir(12);

        Console.WriteLine("===== Mostrar");
        pilha.Mostrar();
    }
}

```

3-

```
using System;
using System.Collections;

class Fila {
    Stack p1, p2;
    public Fila (){
        p1 = new Stack();
        p2 = new Stack();
    }
    public void Inserir(int valor){
        p1.Push(valor);
    }
    public int Remover(){
        while(p1.Count > 1){
            p2.Push(p1.Pop());
        }
        int resp = (int)p1.Pop();
        while(p2.Count > 0){
            p1.Push(p2.Pop());
        }
        return resp;
    }
    public void Mostrar(){
        object[] array = p1.ToArray();
        for (int i = p1.Count - 1; i >= 0; i--){
            Console.WriteLine(array[i]);
        }
    }
}

class ExercicioComplementarU02S01EC03 {
    public static void Main (string[] args) {
        Fila fila = new Fila();

        for(int i = 0; i < 10; i++){
            fila.Inserir(i);
        }
        Console.WriteLine(fila.Remover());
        fila.Inserir(10);
        Console.WriteLine(fila.Remover());
        fila.Inserir(11);
        Console.WriteLine(fila.Remover());
        fila.Inserir(12);

        Console.WriteLine("===== Mostrar");
        fila.Mostrar();
    }
}
```

4-

```
using System;
using System.Collections;

class Pilha {
    Queue f1, f2;
    public Pilha (){
        f1 = new Queue();
        f2 = new Queue();
    }
    public void Inserir(int valor){
        f1.Enqueue(valor);
    }
    public int Remover(){
        while(f1.Count > 1){
            f2.Enqueue(f1.Dequeue());
        }
        int resp = (int)f1.Dequeue();
        while(f2.Count > 0){
            f1.Enqueue(f2.Dequeue());
        }
        return resp;
    }
    public void Mostrar(){
        object[] array = f1.ToArray();
        for (int i = f1.Count - 1; i >= 0; i--){
            Console.WriteLine(array[i]);
        }
    }
}

class ExercicioComplementarU02S01EC04 {
    public static void Main (string[] args) {
        Pilha pilha = new Pilha();

        for(int i = 0; i < 10; i++){
            pilha.Inserir(i);
        }
        Console.WriteLine(pilha.Remover());
        pilha.Inserir(10);
        Console.WriteLine(pilha.Remover());
        pilha.Inserir(11);
        Console.WriteLine(pilha.Remover());
        pilha.Inserir(12);

        Console.WriteLine("===== Mostrar");
        pilha.Mostrar();
    }
}
```

#### Exercícios Lista Linear

```
using System;
```

```
int somar() {
```

```

        int resp = 0;
        for (Celula i = topo; i != null; i = i.prox) {
            resp += i.elemento;
        }
        return resp;
    }

    int maior(){
        int maior = - 1;
        if (primeiro == ultimo) {
            throw new Exception("Erro!");
        }
        else {
            maior = primeiro.prox.elemento;
            Celula i = primeiro.prox.prox;
            while (i != null){
                if (i.elemento > maior) {
                    maior = i.elemento;
                }
                i = i.prox;
            }
        }
        return maior;
    }

    int retornarTerceiroElemento(){
        return (primeiro.prox.prox.prox.elemento);
    }

    int somar() {
        int resp = 0;
        for (Celula i = primeiro.prox; i != null; i = i.prox) {
            resp += i.elemento;
        }
        return resp;
    }

    void inverter () {
        Celula fim = ultimo;
        while (primeiro != fim){
            Celula nova = new Celula (primeiro.prox.elemento);
            nova.prox = fim.prox;
            fim.prox = nova;
            Celula tmp = primeiro.prox;
            primeiro.prox = tmp.prox;
            nova = tmp = tmp.prox = null;
            if (ultimo == fim) {
                ultimo = ultimo.prox;
            }
        }
        fim = null;
    }
}

```

```

int contar() {
    return contar(primeiro.prox);
}

int contar(Celula i){
    int resp;
    if (i == null){
        resp = 0;
    }
    else if (i.elemento % 2 == 0 && i.elemento % 5 == 0){
        resp = 1 + contar(i.prox);
    }
    else {
        resp = contar(i.prox);
    }
    return resp;
}

public class Fila {
    private Celula primeiro;
    public Fila() {
        primeiro = new Celula();
    }

    public void void Inserir(int x) {
        Celula i;
        for (i = primeiro; i.prox != null; i = i.prox){
            i.prox = new Celula(x);
            i = null;
        }
    }

    public int Remover(){
        if (primeiro == ultimo) {
            throw new Exception("Erro ao remover!");
        }
        Celula tmp = primeiro;
        primeiro = primeiro.prox;
        int resp = primeiro.elemento;
        tmp.prox = null;
        tmp = null;
        return resp;
    }
}

class ExercicioComplementarU01S02EC01 {
    public static void Main (string[] args) {

        //Declaração de variáveis
        int num1, num2, soma;

        //Leitura do primeiro número
        Console.WriteLine("Digite um número:");
    }
}

```



```
num1 = int.Parse(Console.ReadLine());

//Leitura do segundo número
Console.WriteLine("Digite outro número:");
num2 = int.Parse(Console.ReadLine());

//Somar
soma = num1 + num2;

//Imprimir resultado na tela
Console.WriteLine("Soma: " + soma);
}
}
```