

# Lógica de Programação

Professor: Emerson Camargo

# Agenda

- Listas
- Exercícios

## Listas

A classe C# List representa uma coleção de objetos fortemente tipados que podem ser acessados por índice.

List<T> fornece funcionalidade para criar uma lista de objetos, localizar itens de lista, classificar lista, pesquisar lista e manipular itens de lista.

## Como criar uma lista em c#?

Ele pode estar vazio ou receber um valor Integer como um argumento que define o tamanho inicial da lista, também conhecido como capacidade.

Se não houver um inteiro passado no construtor, o tamanho da lista será dinâmico e aumentará toda vez que um item for adicionado ao array. Você também pode passar uma coleção inicial de elementos ao inicializar um objeto.

## Como criar uma lista em c#?

**// List com capacidade padrão**

```
List<Int16> list = new List<Int16>();
```

**// List com capacidade= 5**

```
List<string> authors = new List<string>(5);
```

```
string[] animals = { "Cow", "Camel", "Elephant" };
```

```
List<string> animalsList = new List<string>(animals);
```

## Como adicionar itens a uma lista C#?

O método Add adiciona um elemento a uma List.

```
List<int> numberList = new List<int>();  
numberList.Add(32);  
numberList.Add(21);  
numberList.Add(45);
```

## Como ler itens de uma lista C #?

Lista é uma coleção de itens. Podemos usar um loop foreach para percorrer seus itens.

```
foreach (string a in autores)  
Console.WriteLine(a);
```

## Como ler itens de uma lista C #?

Para recuperar um item em uma posição específica na Lista, podemos usar o índice da coleção.

```
WriteLine(authors[2]);
```



## Como usar as propriedades da lista C#

Para recuperar um item em uma posição específica na Lista, podemos usar o índice da coleção.

*As propriedades da classe de lista incluem o seguinte:*

**Capacidade** – Número de elementos que `List<T>` pode conter. A capacidade padrão de um `List<T>` é 0.

**Contagem** – Número de elementos em `List<T>`.

## Como usar as propriedades da lista C#

```
using static System.Console;  
ArrayList authors = new ArrayList();  
authors.Add("Mahesh Chand");  
authors.Add("Praveen Kumar");  
authors.Add("Raj Kumar");  
authors.Add("Dinesh Beniwal");  
authors.Add("David McCarter");  
WriteLine("Count: " + authors.Count);  
WriteLine("Capacity: " + authors.Capacity);  
ReadKey();
```

## Como inserir elementos em uma posição em uma lista C#?

O método InsertRange pode inserir uma coleção na posição especificada.

```
authors.Insert(3, "Bill Author");  
string[] newAuthors = { "New Author1",  
"New Author2", "New Author3" };  
authors.InsertRange(2, newAuthors);
```

## Como remover itens de uma lista C#?

A classe List fornece métodos Remove que podem ser usados para remover um item ou um intervalo de itens.

O método Remove remove a primeira ocorrência do item fornecido na Lista.

```
authors.Remove("New Author1");
```

## Como remover itens de uma lista C#?

O método RemoveAt remove um item na posição especificada.

```
authors.RemoveAt(3);
```

## Como remover itens de uma lista C#?

O método `RemoveRange` remove uma lista de itens do índice inicial para o número de itens.

```
authors.RemoveRange(3, 2);
```

## Como remover itens de uma lista C#?

O método Clear remove todos os itens de um List.

```
authors.Clear();
```

## Como encontrar um elemento em uma lista C#?

O método `IndexOf` encontra um item em uma lista.

O método `IndexOf` retorna -1 se não houver itens encontrados na Lista.

```
int idx = authors.IndexOf("Naveen Sharma");  
if (idx > 0)  
    Console.WriteLine($"Item index in List is: {idx}");  
else  
    Console.WriteLine("Item not found");
```



## Como classificar elementos de uma lista C #?

O método Sort de List<T> classifica todos os itens da lista usando o algoritmo QuickSort.

```
authors.Sort();
```

## Como classificar elementos de uma lista C #?

O método Sort de List<T> classifica todos os itens da lista usando o algoritmo QuickSort.

```
authors.Sort();
```

## Como reverter elementos de uma lista C #?

O método Reverse de List<T> inverte a ordem de todos os itens na lista.

```
authors.Reverse();
```

## Exercício 1

*Faça um programa que crie uma lista de nomes, e peça ao usuário digitar 10 nomes, exiba a lista ao final, e, exiba o primeiro e o último nome da lista.*

**Sistema**  
**Fiep**



**nosso i é de indústria.**