

02

Introdução

Transcrição

Eu sou Marcelo Oliveira, e este é o curso de C# Collections Parte 2.

Caso você não tenha assistido à [Parte 1](https://cursos.alura.com.br/course/csharp-collections) (<https://cursos.alura.com.br/course/csharp-collections>) do curso, é aconselhável que assista, pois lá estão os fundamentos das coleções.

Veremos um resumo dos tópicos que serão abordados adiante.

Em primeiro lugar, teremos as coleções ordenadas, que são aquelas em que conforme inserimos ou removemos objetos, se ordenam automaticamente.

O primeiro tipo de coleção a ser estudado é o *sorted list*, que é, na verdade, um dicionário.

O nome "coleção" se dá pelo fato de que, internamente, ele trabalha com listas, uma para chaves e outra para valores.

Veremos também o *sorted dictionary*, que armazena as chaves em uma árvore binária.

Em seguida, teremos o *sorted set*, um tipo de coleção-conjunto, que armazena seus valores em uma árvore binária.

Estudaremos as arrays multidimensionais, que, diferentemente das unidimensionais, permite trabalharmos com linhas e colunas. Isto facilita a organização das informações.

Após isso, estudaremos a matriz denteada, ou *jagged array*, ou ainda "array de arrays".

Mais adiante, veremos como é possível consultarmos nossas coleções por meio de uma tecnologia chamada "Linq", que é a consulta integrada a linguagem.

Ele permite que façamos consultas em coleções de origem, utilizando filtros, ordenado os resultados, ou ainda transformando as informações oriundas das coleções.

Veremos também, como podemos utilizar o "Linq" para obter faixas de valores e de elementos dentro de uma sequência.

Em seguida, estudaremos métodos para trabalharmos operações de conjuntos dentro do "Linq". Para, por exemplo, fazer uniões, intercessões, concatenações, entre outras operações.

Aprenderemos como converter uma coleção em outra, por exemplo, como transformar uma lista de strings em lista de objetos. Veremos problemas práticos e métodos de resolvê-los.

Por fim, estudaremos o funcionamento do Laço Foreach internamente, tanto para coleções quanto para arrays, e os mecanismos do .NET Framework para proteger essa coleção que está sendo varrida, contra alterações indevidas.