

04

## Listando todos os livros

### Transcrição

Iremos aprender a visualizar o conteúdo do banco de dados do MongoDB. Seja listando todos os livros ou, buscando por um critério específico.

Criaremos uma nova classe, a qual daremos o nome de "listandoDocumentos.cs".

Para criar uma nova classe, clique com o botão direito do mouse sobre o nome do projeto e selecione "Adicionar > Novo Item". Na caixa de diálogo é possível definir o nome do arquivo.

Iremos trabalhar com o mesmo código utilizado anteriormente, isolando somente a conexão com o MongoDB, da seguinte forma:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MongoDB.Bson;
using MongoDB.Driver;

namespace exemploMongoDB
{
    class incluindoMuitosLivros
    {
        static void Main(string[] args)
        {
            Task T = MainAsync(args);
            Console.WriteLine();
            Console.WriteLine("Pressione Enter");
            Console.ReadLine();
        }
        static async Task MainAsync(string[] args)
        {

            var conexaoBiblioteca = new conectandoMongoDB();
            Console.WriteLine("Documento Incluido");

        }
    }
}
```

Incluiremos o novo comando para listar os livros, que será `Console.WriteLine("Listando Documentos")`.

Ao fazer a consulta, o resultado deve aparecer dentro de uma variável no programa. Chamaremos ela de `var listaLivros = await conexaoBiblioteca Livros.Find (new BsonDocument()).ToListAsync()`.

Isso fará a ligação com a classe em que há conexão e com a coleção, além de executar a função `Find` dentro do critério de busca que, neste caso, é `new BsonDocument`, ou seja, nenhum. Utilizando, ainda, `ToListAsync`, porque estamos operando de forma assíncrona.

O próximo passo será escrever o conteúdo da variável, que será `foreach (var doc in listaLivros)` e, dentro, haverá `Console.WriteLine(docToJson())`.

`listaLivros` é uma relação de elementos da classe `Livro`, já que a coleção utiliza elementos da classe.

Para que seja possível imprimir os nomes, incluiremos o comando `ToJson`, para que sejam transformados para este formato.

Para finalizar, incluiremos o comando para que o programa escreva "Fim da Lista".

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MongoDB.Bson;
using MongoDB.Driver;

namespace exemploMongoDB
{
    class IncluindoMuitosLivros
    {
        static void Main(string[] args)
        {
            Task T = MainAsync(args);
            Console.WriteLine();
            Console.WriteLine("Pressione Enter");
            Console.ReadLine();
        }

        static async Task MainAsync(string[] args)
        {

            var conexaoBiblioteca = new ConectandoMongoDB();
            Console.WriteLine("Listando Documentos");

            var listaLivros = await conexaoBiblioteca.Livros.Find(new BsonDocument()).ToListAsync();
            foreach (var doc in listaLivros)
            {
                Console.WriteLine(doc.toJson<Livro>());
            }

            Console.WriteLine("Fim da Lista");
        }
    }
}
```

Feito isso, salvaremos o projeto.

Para executarmos precisaremos antes alterar o objeto de inicialização, clicando com o botão direito do mouse sobre o nome do arquivo, em "Propriedades", definiremos como "exemplosMongoDB.listandoDocumentos". em seguida clicaremos no botão "Iniciar".

Surgirá uma caixa de diálogo onde serão exibidos todos os documentos em formato JSON .