06

Exibindo resultado da busca por um determinado critério

Transcrição

É possível ainda apresentar a listagem dos documentos de forma ordenada, ou ainda, visualizar somente os 5 primeiros, por exemplo.

Criaremos uma nova classe, denominada "listandoDocumentosOrdem.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.

Aproveitando o exemplo utilizado anteriormente, copiaremos o seguinte trecho:

```
 static void Main(string[] args)
{
    Task T = MainAsync(args);
    Console.WriteLine();
    Console.WriteLine();
    Console.WriteLine("Pressione Enter");
    Console.RealLine();
}
static async Task MainAsync(string[] args)
{
    var conexaobiblioteca = new conectandoMongoDB();

    Console.WriteLine("Listando Documentos Autor = Machado de Assis - Classe");
    var construtor = Builders<Livro>.Filter;
    var condicao = construtor.Eq(x => x.Autor, "Machado de Assis");

    listaLivros = await conexaoBiblioteca.Livros.find(condicoes).ToListAsync();
    foreach (var doc in listaLivros)
    {
        Console.WriteLine(doc.ToJson<Livro>());
    }
    Console.WriteLine("Fim da Lista");
    Console.WriteLine("");
    Console.WriteLine("");
}
```

O critério utilizado será de obras com número de páginas igual ou superior a 100:

```
    Console.WriteLine("Listando Documentos mais de 100 páginas");
    var construtor = Builders<Livro>.Filter;
    var condicao = construtor.Gt(x => x.Paginas, 100);
```

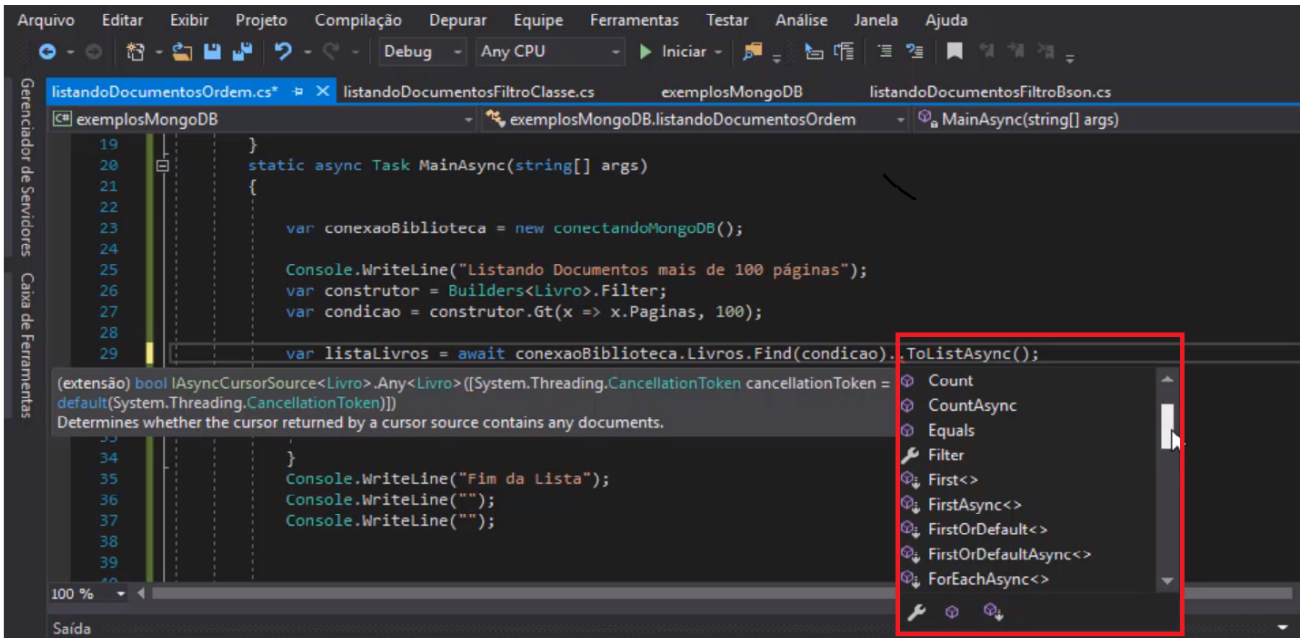
Iremos salvar e em seguida executar o programa, alterando nas "Propriedades", o "Objeto de Inicialização" para "exemplosMongoDB.listandoDocumentosOrdem".

Para acessar as "Propriedades", clique com o botão direito do mouse sobre o nome do projeto, na barra lateral direita da tela.

A seguir clicaremos em "Iniciar" na barra de menu superior.

Surgirá uma caixa de diálogo onde serão exibidos todos os livros com 100 páginas ou mais, indicando que nosso programa está funcionando.

Antes da condição `ToListAsync` temos diversas opções de outros critérios que podemos aplicar.



Utilizaremos o comando `SortBy` , definindo uma condição de ordem, da seguinte forma:

```
var listaLivros = await conexaoBiblioteca.Livros.find(condicao).SortBy(x => x.Titulo).ToListAsy
    foreach (var doc in listaLivros)
```

Será feita a busca pelo número de páginas, conforme estipulamos anteriormente e, além disso, os livros serão listados alfabeticamente pelo título.

Salvaremos e clicaremos em "Iniciar", na barra de menu superior, para executarmos o programa.

Surgirá uma tela onde serão exibidos os títulos, em ordem alfabética.

Limitaremos o número de títulos exibidos, utilizando a função `limit` e definindo, ao lado, a quantidade, no caso, 5:

```
var listaLivros = await conexaoBiblioteca.Livros.find(condicao).SortBy(x => x.Titulo).Limit(5).
```

Salvaremos e executaremos o programa.

```
 static void Main(string[] args)
{
    Task T = MainAsync(args);
    Console.WriteLine();
```

```
        Console.WriteLine();
        Console.WriteLine("Pressione Enter");
        Console.RealLine();
    }
    static async Task MainAsync(string[] args)
    {
        var conexaobiblioteca = new conectandoMongoDB();

        Console.WriteLine("Listando Documentos mais de 100 páginas");
        var construtor = Builders<Livro>.Filter;
        var condicao = construtor.Gt(x => x.Paginas, 100);

    var listaLivros = await conexaoBiblioteca.Livros.find(condicao).SortBy(x => x.Titulo).Limit(5).
        foreach (var doc in listaLivros)
        {
            Console.WriteLine(doc.ToJson<Livro>());
        }
        Console.WriteLine("Fim da Lista");
        Console.WriteLine("");
        Console.WriteLine("");
    }
```

Surgirá uma tela onde serão exibidos os primeiro cinco livros, em ordem alfabética pelo título, que possuem 100 páginas ou mais.