

## Alterando um documento

### Transcrição

Aprenderemos a alterar o conteúdo de um documento.

Em primeiro lugar é preciso localizar o documento para, em seguida, realizar a alteração.

O primeiro passo é criar um novo item, ao qual daremos o nome de "alterandoDocumentos.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 listar o livro "Guerra dos Tronos", alterando o parâmetro para `Console.WriteLine("Listar e alterar o livro Guerra dos Tronos")`, e o critério para `var condicao = construtor.Eq(x => x.Titulo, "Guerra dos Tronos")`.

A exibição não será limitada, nem ordenada.

```
Console.WriteLine("Listar e alterar o livro Guerra dos Tronos");  
var construtor = Builders<Livro>.Filter;  
var condicao = construtor.Eq(x => x.Titulo, Guerra dos Tronos);
```

Salvaremos e executaremos 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.

Clicando no botão "Iniciar", na barra de menu superior, surgirá uma tela onde será exibido o título definido.

Para alterarmos os dados deste título, faremos algumas alterações no programa.

Na variável `doc` está o conteúdo do documento buscado pelo MongoDB, portanto iremos alterá-lo. Utilizaremos ainda um comando `await`, para que as demais informações não sejam alteradas.

A condição ( `condicao` ) é alterarmos somente os dados do livro "Guerra dos Tronos", com as informações contidas em `doc`.

```
Console.WriteLine(doc.ToJson<Livro>())  
doc.Ano = 2000;  
doc.Paginas = 900;  
await conexaoBiblioteca.Livros.ReplaceOneAsync(condicao, doc);
```

Além de alterar o referido item, também iremos listá-lo. Para isso basta repetirmos o comando, abaixo do que já criamos, alterando a função para apenas listar o livro.

```

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

    Console.WriteLine("Listar e alterar o livro Guerra dos Tronos");
    var construtor = Builders<Livro>.Filter;
    var condicao = construtor.Eq(x => x.Titulo, Guerra dos Tronos);

    var listaLivros = await conexaoBiblioteca.Livros.find(condicao).SortBy(x => x.Titulo).Limit(5).
    foreach (var doc in listaLivros)
    {
        Console.WriteLine(doc.ToJson<Livro>())
        doc.Ano = 2000;
        doc.Paginass = 900;
        await conexaoBiblioteca.Livros.ReplaceOneAsync(condicao, doc);
    }
    Console.WriteLine("Fim da Lista");
    Console.WriteLine("");
    Console.WriteLine("");

    Console.WriteLine("Listar o livro Guerra dos Tronos");
    construtor = Builders<Livro>.Filter;
    condicao = construtor.Eq(x => x.Titulo, Guerra dos Tronos);

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

```

Salvaremos o programa e o executaremos, alterando nas "Propriedades", o "Objeto de Inicialização" para "exemplosMongoDB.alterandoDocumento".

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

Para executar, clicaremos em "Iniciar", na barra de menu superior.

Surgirá uma tela onde, em primeiro lugar temos o livro "Guerra dos Tronos" com o ano 1999, e 856 páginas. Em seguida, ele foi listado novamente, agora com o ano de 2000, e 900 páginas.

Abrindo o Robomongo, veremos que os dados também foram alterados na exibição.

