

 01

Introdução

Transcrição

Me chamo Victorino Vila e este é o curso de .Net e MongoDB.

O curso abordará a manipulação de base de dados no MongoDB por meio do Visual Studio .NET, utilizando qualquer uma das linguagens que o framework da Microsoft suporta.

Primeiro vamos entender qual a importância do MongoDB, e por que utiliza-lo ao invés de bancos de dados relacionais. Em tela temos o exemplo de um site de notícias.

Temos o título da notícia e, logo abaixo, os nomes dos jornalistas autores da matéria. Ao lado dos nomes temos a indicação da sucursal, ou seja, o escritório onde a matéria foi elaborada. Em seguida temos o texto em si e, ao final, temos os comentários.

Os comentários podem ou não ter curtidas, respostas, e assim por diante.

Imaginemos que precisamos construir este site utilizando um banco de dados relacional. Sob esta ótica, deveríamos ter pelo menos três tabelas.

Uma tabela de notícias, contendo o ID da notícia, título, data, sucursal e conteúdo. Conectada a uma tabela de jornalistas, porque não necessariamente apenas um jornalista escreveu a matéria.

Por último teríamos uma tabela de comentários, que poderia ter ainda algum relacionamento entre ela mesma, se considerássemos as respostas.

Para construirmos um banco de dados relacional provavelmente teríamos de criar *queries*, utilizando *unions* ou *joins*, empregando as tabelas descritas acima.

De certa maneira, caso houvesse milhares de notícias em nosso banco de dados, isso resultaria em consultas muito complexas, com performance abaixo do esperado e de estrutura rígida.

Digamos que, hipoteticamente, nosso site está no ar e alguém solicite a criação de uma *tag* para esta página, de modo que num mecanismo de busca ela seja mais facilmente encontrada.

A *tag* é um assunto relacionado a um conteúdo. Por exemplo, em uma manchete onde lê-se "Brasil bate Equador, mantém 100% com Tite e ganha 'título' da Eliminatória", a *tag* pode ser esporte, ou Brasil, enfim, há uma série de identificadores, podendo haver uma ou mais *tags*.

Se fosse o caso de utilizarmos o banco de dados relacional deveríamos ter ainda mais uma tabela de *tags* ligada à tabela de notícias. Ou seja, teríamos que alterar a estrutura do nosso banco de dados e isso poderia comprometer as notícias antigas. Isso pode ser problemático caso não seja feita uma boa análise desse site.

Agora vamos estudar a hipótese de criarmos um banco de dados NoSQL, ou seja, cuja representação é feita por documentos, que são na verdade grandes strings, ou caracteres de texto.

Um dos bancos de dados mais utilizados no formato NoSQL é o MongoDB.

Se utilizarmos essa ferramenta para construir, por exemplo, um site de notícias, provavelmente a estrutura seria algo parecido com o exemplo abaixo.

```
{  
    _id:Objectid("342784758374573845734")  
    Título:"Brasil bate equador .....",  
    Data:"31/08/2017 ....",  
    Sucursal: "Porto Alegre",  
    Jornalistas: [  
        Marinho Saldanha,  
        Pedro Ivo  
    ],  
    Conteúdo: "xxxxxxxxx",  
    Comentário: [  
        {  
            Assinante:"Newton42",  
            Data:"30/08/2017",  
            Comentário:"yyyyyyyy"  
        },  
        {  
            Assinante:"Toletão"  
            Data:"30/08/2017"  
            Comentário: "yyyyyyyy"  
        }  
    ]  
}
```

Temos um código JSON onde aparecem, em primeiro lugar o título da matéria, em seguida a data da publicação, a sucursal e, quando chegamos aos jornalistas, temos um *array* de nomes porque podem ser vários profissionais creditados, e logo após temos o conteúdo, que é o texto da notícia em si.

Finalmente, na parte dos comentários temos um vetor com uma outra expressão JSON, onde aparecem o nome do assinante, a data e o comentário do indivíduo.

A vantagem de utilizar o MongoDB é a possibilidade de termos, dentro de uma mesma coleção, documentos com JSONs diferentes, portanto, se o usuário quisesse que o site fosse controlado através de tags de assuntos, poderíamos fazer isso simplesmente criando uma nova estrutura JSON que incluisse um array onde apareceriam os tags de identificação de determinada notícia. O passado permaneceria sem o objeto *tag* e o futuro passaria a contar com isso. Portanto não é necessário re-desenhar o banco nem alterar a sua estrutura.

Entretanto, é importante notar que o MongoDB pode não ser aplicável em determinadas situações.

Seguindo o exemplo com o qual estamos trabalhando, podemos perceber que as *tags* são comandos strings livres, então não temos muito controle sobre o tipo de *tag* que vamos utilizar. Se for por exemplo "esportes", no plural, e alguém escrever "esporte" por um erro de digitação mas querendo falar do mesmo assunto, o MongoDB entenderá como coisas diferentes, por serem strings distintas.

No MongoDB não há integridade referencial entre os elementos do documento de uma única coleção, ou entre elementos das coleções.

Entretanto, em sites de redes sociais por exemplo, ou qualquer tipo de site que seja dinâmico e a estrutura de dados está sempre se modificando, os bancos de dados NoSQL e por exemplo o MongoDB são uma das melhores opções a serem

utilizadas. Por isso é necessário sabermos identificar em quais casos o MongoDB vai ou não ser o melhor banco de dados para a sua aplicação, em vez de bancos de dados relacionais.

Um exemplo de site que faz uso do MongoDB é o [Cartola FC \(http://globoesporte.globo.com/cartola-fc/\)](http://globoesporte.globo.com/cartola-fc/). Trata-se de uma rede social onde o usuário compra e vende jogadores de futebol, e utiliza os seus *scores* no campeonato brasileiro para criar uma disputa de pontos entre os usuários. Por ser uma estrutura bastante complexa com inter-relacionamento muito forte, o MongoDB foi a melhor solução.

Outra rede social que utiliza o MongoDB é o Foursquare, que trabalha com curtidas dos usuários em restaurantes, bares, enfim, diversos estabelecimentos sobre os quais os membros podem opinar publicamente.

O MongoDB hoje já é considerado o quinto maior banco de dados do mundo, ainda distante da Oracle, SQL Server e DB2, por exemplo, mas em constante crescente nos últimos anos. Isso se dá por conta de sua flexibilidade e das necessidades de modelagem às quais o MongoDB vêm se adaptando muito mais facilmente.

O foco deste curso é estudarmos como é possível manipular o MongoDB dentro do Visual Studio .NET. Utilizaremos em nossos exemplos a linguagem C#.

Faremos um grande projeto no qual serão criados diversos programas, cada um evidenciando qual é a interface do C# com o MongoDB, feito a partir de um modelo de console, onde iremos colocar na tela somente a resposta, e depois faremos duas aplicações práticas utilizando ASP.NET MVC.

O primeiro exemplo é uma aplicação de blog com registro de usuários, com o acesso vinculado ao seu e-mail, possibilitando a visualização do conteúdo e permitindo que sejam deixados comentários e a busca de publicações pela tag relacionada.

A próxima aplicação que faremos utilizará mapas, georreferenciamento. Ela nos permitirá pesquisar aeroportos a partir da entrada de uma cidade e do raio desejado. A partir disso ela mostrará automaticamente um mapa onde o ponto verde representa a localização do usuário, e os pontos em vermelho são os aeroportos localizados dentro do raio definido por ele.

Vamos estudar ainda como instalar o MongoDB num ambiente Windows e aprender como ele pode se tornar um serviço deste OS na nossa máquina. Além disso, vamos trabalhar com uma aplicação que se chama Robomongo, ou Robo 3T, que permite visualizar a estrutura do MongoDB de forma mais fácil do que simplesmente na interface padrão em linhas de comando, apesar de ela continuar existindo.

Finalmente, vamos utilizar o Visual Studio 2017 e aprender como criar em nossos projetos a referência e o acesso ao MongoDB.