03

Deletando objetos com o Entity

Transcrição

Vimos como listar os produtos armazenados no banco de dados. Agora, aprenderemos como *remover* produtos da tabela usando o Entity.

Já temos o método `Remover()` na classe `ProdutoDAO` que faz o processo de remoção dos dados persistidos no banco. O método basicamente monta e executa o comando SQL `delete`, passando o **Id** do produto.

Na classe `Program` dentro do método `Main()`, vamos fazer uma chamada para o método `ExcluirProdutos()`. Em seguida implementaremos o método, pegando as instância da classe `LojaContext`.

```
static void Main(string[] args)
{
    //GravarUsandoAdoNet();
    //GravarUsandoEntity();
    RecuperarProdutos();
    ExcluirProdutos();
}

private static void ExcluirProdutos()
{
    using(var repo = new LojaContext())
    {

    }
}
```

Para remover os produtos, antes é preciso pegá-los, visto que não podemos remover um produto que não sabemos. Pegaremos os produtos da mesma forma que fizemos no método `RecuperarProdutos()`.

```
static void Main(string[] args)
{
    //GravarUsandoAdoNet();
    //GravarUsandoEntity();
    RecuperarProdutos();
    ExcluirProdutos();
}

private static void ExcluirProdutos()
{
    using(var repo = new LojaContext())
    {
        IList<Produto> produtos = repo.Produtos.ToList();
    }
}
```

Com a lista `produtos` em mãos, vamos percorrê-la. Dessa forma, passaremos cada item como argumento ao método `Remover()` da propriedade `repo.Produto`.

```
static void Main(string[] args)
{
    //GravarUsandoAdoNet();
    //GravarUsandoEntity();
    RecuperarProdutos();
    ExcluirProdutos();
}

private static void ExcluirProdutos()
{
    using(var repo = new LojaContext())
    {
        IList<Produto> produtos = repo.Produtos.ToList();
        foreach(var item in produtos)
        {
            repo.Produtos.Remove(item);
        }
    }
}
```

Após o encerramento do laço, pediremos para o repositório salvar as alterações feitas.

```
static void Main(string[] args)
{
    //GravarUsandoAdoNet();
    //GravarUsandoEntity();
    RecuperarProdutos();
    ExcluirProdutos();
}

private static void ExcluirProdutos()
{
    using(var repo = new LojaContext())
    {
        IList<Produto> produtos = repo.Produtos.ToList();
        foreach(var item in produtos)
        {
            repo.Produtos.Remove(item);
        }
        repo.SaveChanges();
    }
}
```

Para testar a remoção sem que seja necessário acessar o banco de dados e verificar a remoção do produto, dentro do método `Main()` após a chamada ao `ExcluirProdutos()`, faremos outra chamada ao `RecuperarProdutos()`.

```
static void Main(string[] args)
{
    //GravarUsandoAdoNet();
    //GravarUsandoEntity();
    RecuperarProdutos();
    ExcluirProdutos();
    RecuperarProdutos();
}
```

```
private static void ExcluirProdutos()
{
    using(var repo = new LojaContext())
    {
        IList<Produto> produtos = repo.Produtos.ToList();
        foreach(var item in produtos)
        {
            repo.Produtos.Remove(item);
        }

        repo.SaveChanges();
    }
}
```

Como estamos removendo todos os produtos do banco de dados, o laço do método `RecuperaProdutos()` não será executado, consequentemente não executando o `Console.WriteLine(item.Nome)` , o que não nos indicaria que a remoção foi efetuada com sucesso. Para visualizarmos melhor o teste, vamos adicionar um `Console.WriteLine()` antes do laço.

```
static void Main(string[] args)
{
    //GravarUsandoAdoNet();
    //GravarUsandoEntity();
    RecuperarProdutos();
    ExcluirProdutos();
    RecuperarProdutos();
}

private static void ExcluirProdutos()
{
    using(var repo = new LojaContext())
    {
        IList<Produto> produtos = repo.Produtos.ToList();
        foreach(var item in produtos)
        {
            repo.Produtos.Remove(item);
        }

        repo.SaveChanges();
    }
}


private static void RecuperarProdutos()
{
    using (var repo = new LojaContext())
    {
        IList<Produto> produtos = repo.Produtos.ToList();
        Console.WriteLine("foram encontrados {0} produto(s)", produtos.Count);
        foreach(var item in produtos)
        {
            Console.WriteLine(item.Nome);
        }
    }
}
```

Executando a aplicação, vemos que tudo funcionou como o esperado.