

Relacionamento um para muitos

Transcrição

Na aula anterior aprendemos como o Entity sincroniza estrutura das classes entidade, com a estrutura das tabelas no banco de dados. Isso era necessário porque precisávamos evoluir a aplicação.

Para uma loja prosperar, é necessário que os clientes comprem. Nessa aula, modelaremos o conceito de compra, em uma nova classe e relacionaremos com o produto. Dessa forma veremos como o Entity trabalhará com relacionamentos.

Na classe `Program` dentro do método `Main()`, simularemos uma compra de um produto. Começaremos criando um novo produto e popularemos:

```
class Program
{
    static void Main(string[] args)
    {
        //compra de 6 pães franceses
        var paoFrances = new Produto();
        paoFrances.Nome = "Pão Francês";
        paoFrances.PrecoUnitario = 0.40;
        paoFrances.Unidade = "Unidade";
        paoFrances.Categoria = "Padaria";
    }
}
```

Mas como simularemos uma compra? Instanciaremos a classe `Compra`, populando os atributos `Quantidade`, `Produto` e `Preco`. O valor de `Preco` é a multiplicação entre o `PrecoUnitario` do produto e a `Quantidade` da compra.

```
class Program
{
    static void Main(string[] args)
    {
        //compra de 6 pães franceses
        var paoFrances = new Produto();
        paoFrances.Nome = "Pão Francês";
        paoFrances.PrecoUnitario = 0.40;
        paoFrances.Unidade = "Unidade";
        paoFrances.Categoria = "Padaria";

        var compra = new Compra();
        compra.Quantidade = 6;
        compra.Produto = paoFrances;
        compra.Preco = paoFrances.PrecoUnitario * compra.Quantidade;
    }
}
```

Como a classe e as propriedades ainda não foram criadas, podemos usar o atalho "Ctrl + . (ponto)" do Visual Studio, e pedir a criação. Podemos persistir as informações da compra no banco de dados, porém ainda não possuímos a tabela que representa a classe `Compra`. Para sincronizar a classe com o banco, usaremos novamente **Migrations**.

