

01

Relacionamento um para um

Transcrição

Na aula anterior vimos como o Entity trata o relacionamento **muitos para muitos**, usando o conceito da classe `Promocao`. Nessa aula aprenderemos o relacionamento de **um para um**.

A ideia é criarmos uma entidade para os clientes que compram na loja. Além disso, também criaremos uma outra entidade para o endereço de entrega.

Antes de começarmos, iremos mover todo o conteúdo do método `Main()` para um novo método chamado `MuitosParaMuitos()`. Com o método `Main()` vazio, começaremos instanciando um objeto da classe `Cliente`.

```
class Program
{
    static void Main(string[] args)
    {
        var fulano = new Cliente();
        fulano.Nome = "Fulaninho de Tal";
    }

    private static void MuitosParaMuitos()
    {
        // ...
    }

    private static void ExibeEntries(IEnumerable<EntityEntry> entries)
    {
        // ...
    }
}
```

Como a classe `Cliente` ainda não existe, iremos criá-la. Lembrando que além das propriedades `Nome` e `EnderecoDeEntrega`, criaremos a propriedade `Id` para persistirmos no banco de dados.

```
namespace Alura.Loja.Testes.ConsoleApp
{
    public class Cliente
    {
        public int Id { get; set; }
        public string Nome { get; internal set; }
        public Endereco EnderecoDeEntrega { get; set; }
    }
}
```

Criaremos agora a classe `Endereco` com as propriedades `Numero`, `Logradouro`, `Complemento`, `Bairro` e `Cidade`.

```
namespace Alura.Loja.Testes.ConsoleApp
{
    public class Endereco
```

```

    {
        public int Numero { get; internal set; }
        public string Logradouro { get; internal set; }
        public string Complemento { get; internal set; }
        public string Bairro { get; internal set; }
        public string Cidade { get; internal set; }
    }
}

```

Voltaremos para o método `Main()` para popularmos as informações de endereço e associarmos com o cliente. Já instanciaremos o objeto de `LojaContext` para persistirmos.

```

static void Main(string[] args)
{
    var fulano = new Cliente();
    fulano.Nome = "Fulaninho de Tal";
    fulano.EnderecoDeEntrega = new Endereco()
    {
        Numero = 12,
        Logradouro = "Rua dos Inválidos",
        Complemento = "sobrado",
        Bairro = "Centro",
        Cidade = "Cidade"
    };

    using (var contexto = new LojaContext())
    {
        contexto.Clientes.Add(fulano);
        contexto.SaveChanges();
    }
}

```

Dentro da classe `LojaContext` iremos criar a propriedade para o cliente.

```

namespace Alura.Loja.Testes.ConsoleApp
{
    public class LojaContext : DbContext
    {
        public DbSet<Produto> Produtos { get; set; }
        public DbSet<Compras> Compras { get; set; }
        public DbSet<Promocao> Promocoes { get; set; }
        public DbSet<Cliente> Clientes { get; set; }

        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            optionsBuilder.UseSqlServer("Server=(localdb)\\mssqllocaldb;Database=LojaDB;Trusted_Connection=True");
        }
    }
}

```

Fizemos mais uma evolução da aplicação. Agora precisamos sincronizar com o banco de dados.

