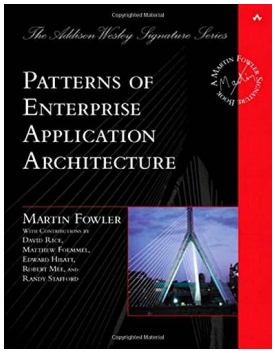
Nivelamento: Entity Framework

http://educandoweb.com.br

Prof. Dr. Nelio Alves

Problema

Por muitos anos, uma grande dificuldade de se criar sistemas orientados a objetos foi a comunicação com o banco de dados relacional.



Martin Fowler: ~30% do esforço de se fazer um sistema

Exemplo simples

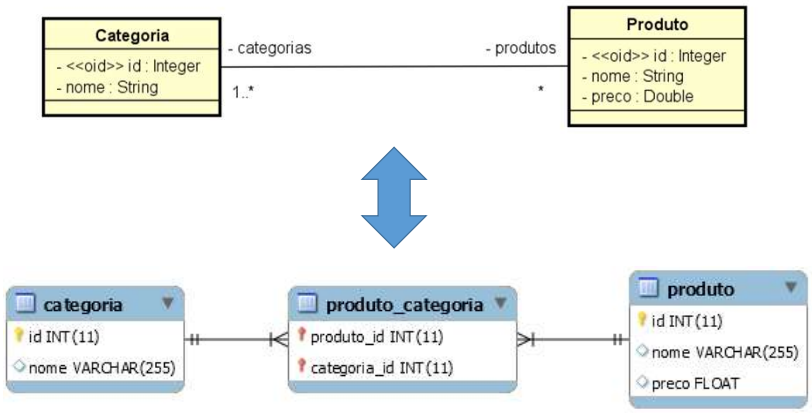
```
Client client = null;
using (connection)
{
    using (var command = new SqlCommand("SELECT * FROM Clients WHERE Id = @id;", connection))
    {
        command.Parameters.Add(new SqlParameter("@id", id));
        connection.Open();
        using (var reader = command.ExecuteReader())
        {
            if (reader.Read())
            {
                client = new Client();
                client.Id = reader.GetString(0);
                client.Name = reader.GetString(1);
                client.Email = reader.GetString(2);
                client.Phone = reader.GetString(3);
            }
        }
    }
}
return client;
```


Outras questões que devem ser tratadas:

- Contexto de persistência (monitorar alterações nos objetos que estão atrelados a uma conexão em um dado momento)
  - Alterações
  - Transação
  - Concorrência
- Mapa de identidade (cache de objetos já carregados)
- Carregamento tardio (lazy loading)
- Etc.

Solução: Mapeamento Objeto-Relacional

ORM (Object-Relational Mapping): Permite programar em nível de objetos e comunicar de forma transparente com um banco de dados relacional



Entity Framework

https://docs.microsoft.com/en-us/ef/

Entity Framework Core

EF Core is a lightweight, extensible, and cross-platform version of Entity Framework.

Entity Framework 6

EF 6 is a tried and tested data access technology with many years of features and stabilization.

Providers

https://docs.microsoft.com/en-us/ef/core/providers/index

Principais classes

- **DbContext:** um objeto DbContext encapsula uma sessão com o banco de dados para um determinado modelo de dados (representado por DbSet's).
  - É usado para consultar e salvar entidades no banco de dados
  - Define quais entidades farão parte do modelo de dados do sistema
  - Pode definir várias configurações
  - É uma combinação dos padrões Unity of Work e Repository
    - **Unity of work:** "mantém uma lista de objetos afetados por uma transação e coordena a escrita de mudanças e trata possíveis problemas de concorrência" - Martin Fowler.
    - **Repository:** define um objeto capaz de realizar operações de acesso a dados (consultar, salvar, atualizar, deletar) para uma entidade.
- **DbSet<TEntity>**: representa a coleção de entidades de um dado tipo em um contexto. Tipicamente corresponde a uma tabela do banco de dados.

Processo geral para se executar operações