10

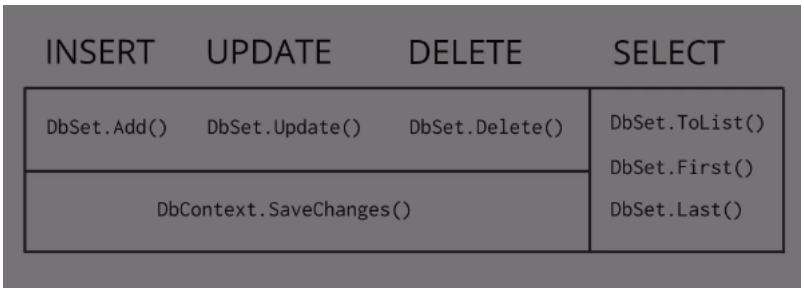
Encerramento - o que aprendemos

Transcrição

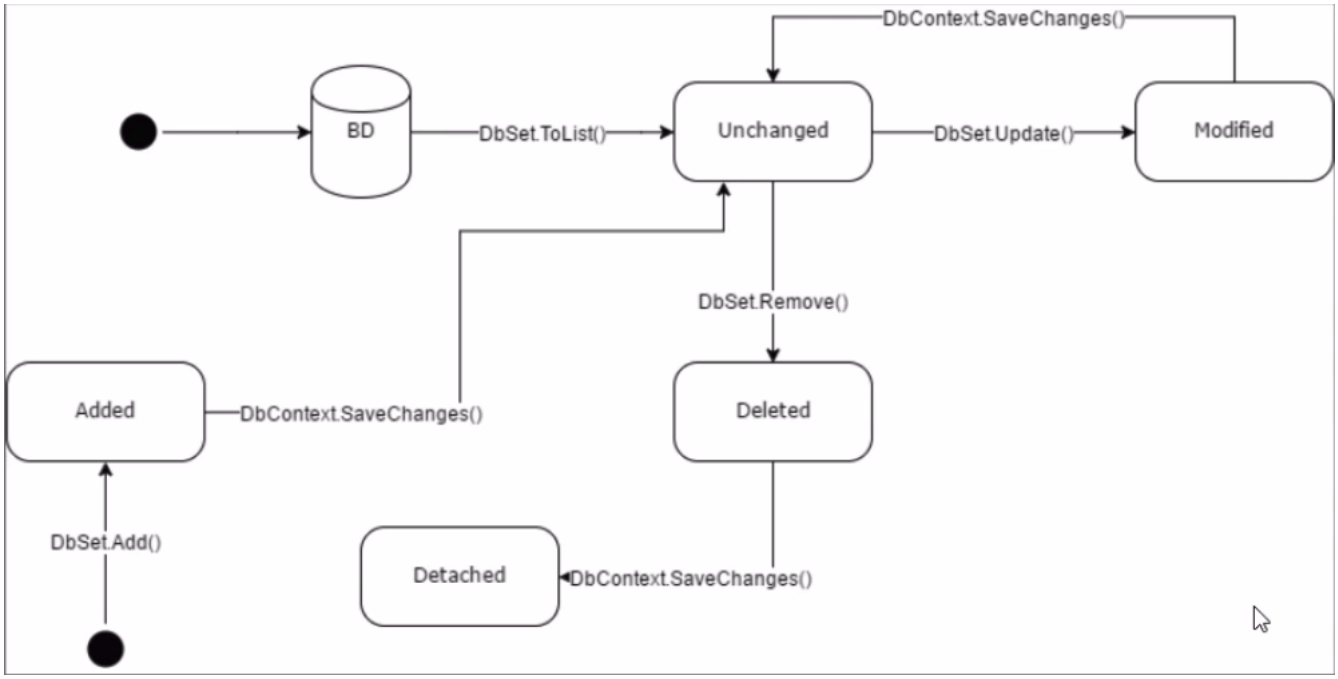
Finalizamos o curso de Entity Framework Core! Faremos um resumo sobre o que aprendemos durante o curso.

Vimos que para trabalhar com o modelo de negócios e persisti-lo no banco de dados era algo extremamente trabalhoso. O programador era responsável por gerenciar e manter o código que se comunicava com o banco. Qualquer mudança na lógica de negócio, gerava uma série de impactos nas classes de acesso aos dados.

Em seguida, Começamos o uso do Entity após a instalação dos pacotes, passando a substituir todo o trabalho, permitindo forcarmos na lógica de negócio. Aprendemos como o Entity gerencia os comandos SQL, trabalhando com a DML (`INSERT` , `SELECT` , `UPDATE` e `DELETE`) para manipular os dados.



Para gerenciar quais objetos e quais comandos SQL precisavam ser emitidos para o banco de dados, o Entity usava um recurso chamado *Change Tracker*, onde era armazenado os estados de cada objeto.



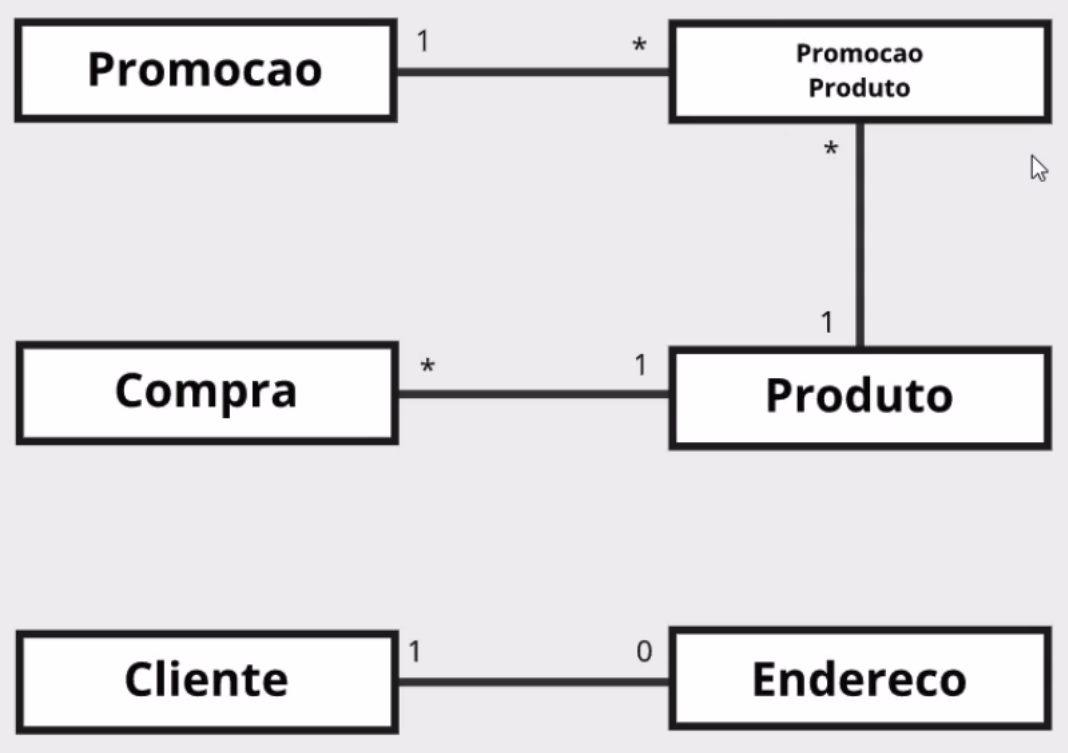
Enquanto a aplicação evoluía, era necessário sincronizá-la com o banco de dados, onde Entity também nos auxilia com a DDL (`CREATE` , `DROP` , `ALTER` , `RENAME`). Utilizamos o conceito de migrações (*Migrations*), onde as alterações no negócio são aplicadas ao banco. As migrações eram feitas por meio de classes, que tinha os métodos `Up()` para subir e `Down()` para descer de versão.

Instalamos no NuGet, um pacote para utilizarmos diversos comandos que trabalham com as migrações.

- Add-Migration
- Remove-Migration
- Update-Database

Todas as alterações no banco de dados eram armazenadas na tabela `__EFMigrationHistory`. A tabela indicava quais migrações tinham sido aplicadas no banco de dados.

Após entender como o Entity funciona, aprendemos como ele trabalha e gerencia os relacionamentos entre classe. Os relacionamentos eram **um para um**, **um para muitos** e **muitos para muitos**. Porém o Entity Core não gerencia sozinho as tabelas `JOIN`, por isso contornamos criando uma classe específica.



Para recuperarmos dados relacionados, usamos métodos específicos:

- `Include()`
- `ThenInclude()`
- `Load()`

Esse foi o curso de Entity Framework Core. Te vejo nos próximos cursos!