

Entity Framework (EFC)

Tem duas classes principais

↳ DbContext: serve para estabelecer uma conexão com o banco, recuperar os dados e faz as operações com os dados

↳ DbSet^{<T>}: serve para definir os dados que serão manipulados no banco, no caso, os tabelas do banco

Criando string de conexão com o banco

os tabelas serão criados de forma automática por causa do migrations, mas por isso precisamos configurar a string de conexão

String de conexão → expressão com os parâmetros para conexão ao BD

↳ temos 2 ~~tipos~~ tipos: um com autenticação de SQL server (nome e senha) e o com autenticação de Windows (trusted connection)

↳ Na string consta a instância, nome do DB, detalhes de autenticação e outras configurações de acordo com o ambiente

Primeiro obtemos o nome do servidor e o nome da instância, podemos fazer isso abrindo o Manager do SQL server

Serviços do SQL → SQL server que estiver em execução → M2 → propriedades
Serviços, com isso conseguimos o nome do instância e do servidor local.

Que podemos achar no VS22 em Exibir → Server explorer.

↳ Para criar uma string a partir do VS22, clicamos em conexões de dados → adicionamos informamos o nome do servidor e o instância, daí lá clicamos (M2) no banco → propriedades e iremos obter a string de conexão.

Configurar o EF Core

↳ Baixar pacote NuGet (gerenciador de pacotes)

Microsoft.EntityFrameworkCore.SqlServer
 --> Tools
 --> Design

(Reduzir os pacotes dentro do projeto ou em dependências/pacotes)

↳ Criar uma classe de contexto que herda de DbContext (EFC)
 ↳ Criar um construtor de opções do tipo DbContextOptions

que faz referência ao mapeamento

Carregar as informações necessárias para configurar o DbContext

↳ Definir o nome do DbContext para manter a organização dos arquivos no BD
 ↳ Definir o DbContext (base para os mapeamentos) no EF Core
 ↳ Definir o Mapping de contextos no appsettings.json

Paralelamente ao contexto que criamos, carregar informações de configuração do DbContext e definir quais classes serão mapeadas no banco

↳ Definir o service no startup, as no configurationService
↳ service, AddressContext < origem do contexto > (x = > x . name -
do parâmetro service(configuration, getConnectionString("name -
de string de conexão que definimos no appsettings")));

①. service - criado com tempo de vida request, ou
app, o cada request será criado uma no instância.