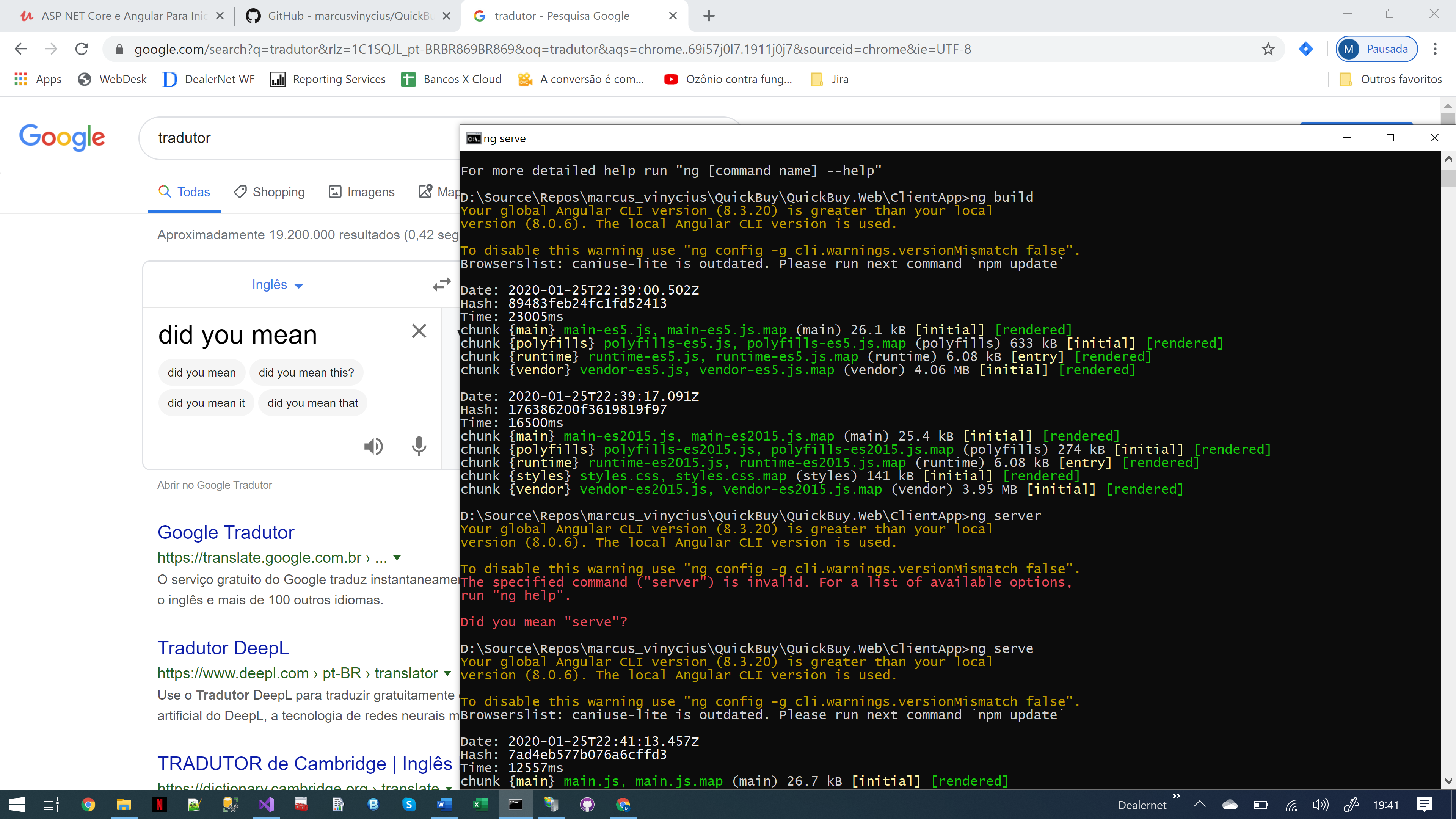
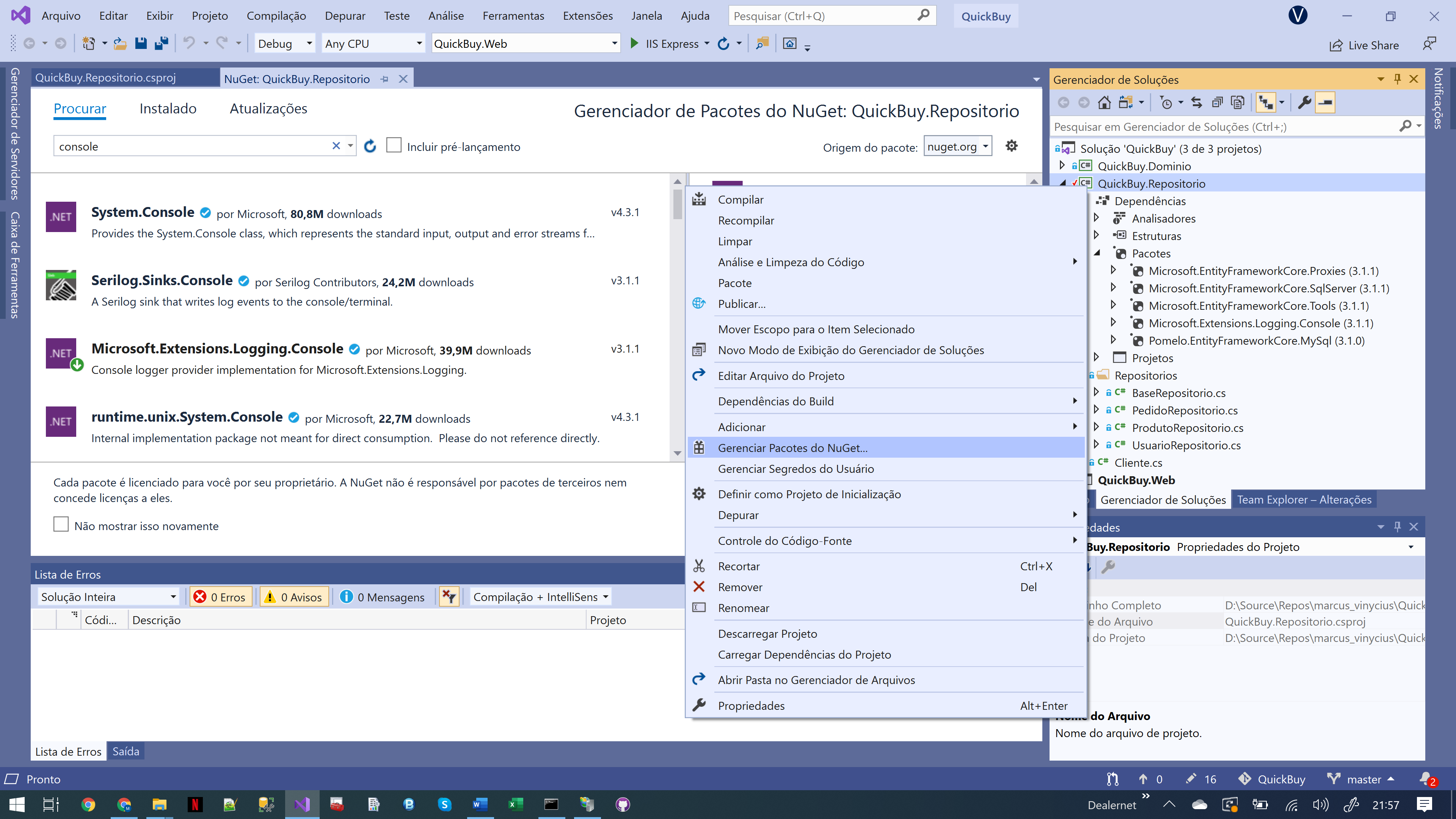
1. Iniciando o Angular
   1. Prompt como ADM
   2. Comando “ng” versão do angular Global e versão do angular Local
   3. Pasta ClientApp
      1. “D:\Source\Repos\marcus\_vinycius\QuickBuy\QuickBuy.Web\ClientApp\”
   4. Comando para fazer o build na solução:
      1. “ng build”
   5. Disponibilizar o APP fora do visual studio
      1. “ng serve”
   6. Ctrl + C -> S , para Sair
   7. Comando para fazer o Build na solução utilizando os arquivos de configuração que estão na no app ClienteApp (agular.json, package.json, tsconfig.json)
      1. “npm start”



1. Instalacao dos pacotes ”Nuget”
2. Teclas de Atalho:
   1. Ctrl + R + G : Apaga as referencias não utilizadas em uma classe
   2. Ctrl + . : Add referencias em uma classe
   3. F12 : Abre a classe do objeto q esta com o cursor em cima.
3. Iniciar o angular Utilizar o comando abaixo na pasta do projeto
   1. Comando: “npm start”
   2. D:\Source\Repos\marcus\_vinycius\QuickBuy\QuickBuy.Web\ClientApp
4. Aula 16: Rodar o projeto com Angular
5. Aula 36: Mapeamento 1 para muitos “IMPORTANTE”
6. Aula 42: Add Migration
   1. Exibir -Outros – Controle de Gerenciador de Pacotes
   2. Projeto Padrao: QuickBuy.Repositorio
      1. PM> Add-Migration PrimeiraVersaoBase -> Criar a estrutura do Migration
      2. PM> Update-Database -Verbose -> Cria o banco de dados e as tabelas, a propriedade “-Verbose” exibe os comandos do BD na tela.
      3. No arquivo Startup.cs eu informei qual a string de conexão será utilizada
      4. O nome do banco vai pegar da string de conexão declarada no arquivo “config.json” no projeto “QuickBuy.Web”.
      5. O arquivo “config.json” no projeto “QuickBuy.Web” tem q está com a propriedade “Copiar para diretório de Saída” -> “Copiar Sempre”
      6. PM> Add-Migration CargaFormaPagamento -> adiciona a estrutura no Migration.
      7. PM> Update-Database -> Insere valores na tabela FormaPagamento no banco de dados
7. Aula : Rodar o projeto com Core

Parei na aula 40.



* 1. Pomelo.EntityFrameworkCore.MySQL: Pacote para o banco de dados MySQL;
  2. Microsoft.EntityFrameworkCore.SqlServer: Pacote para o banco de dados SQLServer;
  3. Microsoft.EntityFrameworkCore.Proxies: Este pacote vai permitir usar os recursos do Lazy Loading no EF Core;
     1. Lazy Loading ou 'Carregamento preguiçoso' é um recurso onde as entidades relacionadas no modelo de entidades são carregadas nas consultas sob demanda, ou seja, somente quando forem realmente necessárias e isso ocorre quando usamos um First, Find, Single, Find, ToList, ToArray ou quando iteramos usando o foreach ou for/next.
  4. Microsoft.Extensions.Logging.Console: Para poder logar as consultas SQL geradas pelo EF Core no Console;
  5. Microsoft.EntityFrameworkCore.Tools: Este pacote vai permitir usar uma interface de linha de comandos (CLI) para poder realizar tarefas com o EF Core;

1. Comandos .Net
   1. Ctrl + .. : add uma referencia a classe
   2. Ctrl + r + g : Excluir as referencias que não estão sendo utilizadas na classe