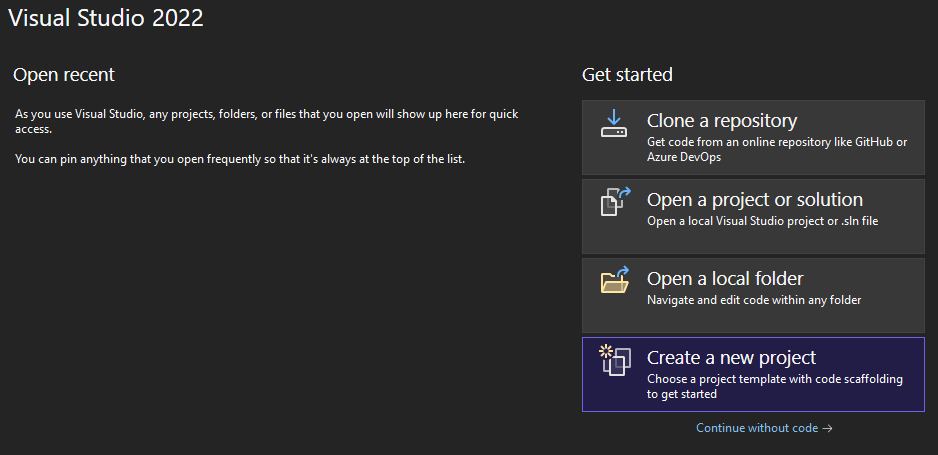
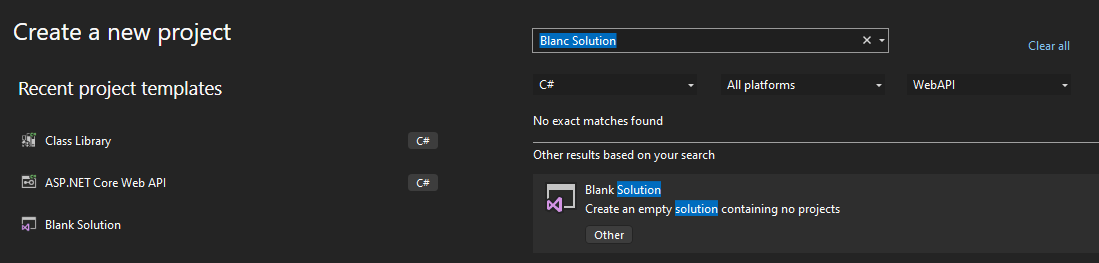
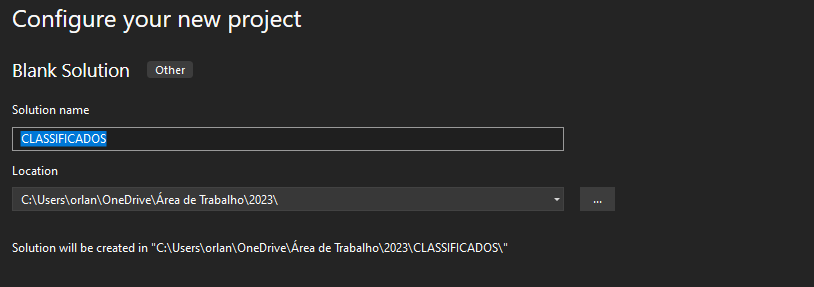
**Criação da App de Classificados**

**1 – Setup**

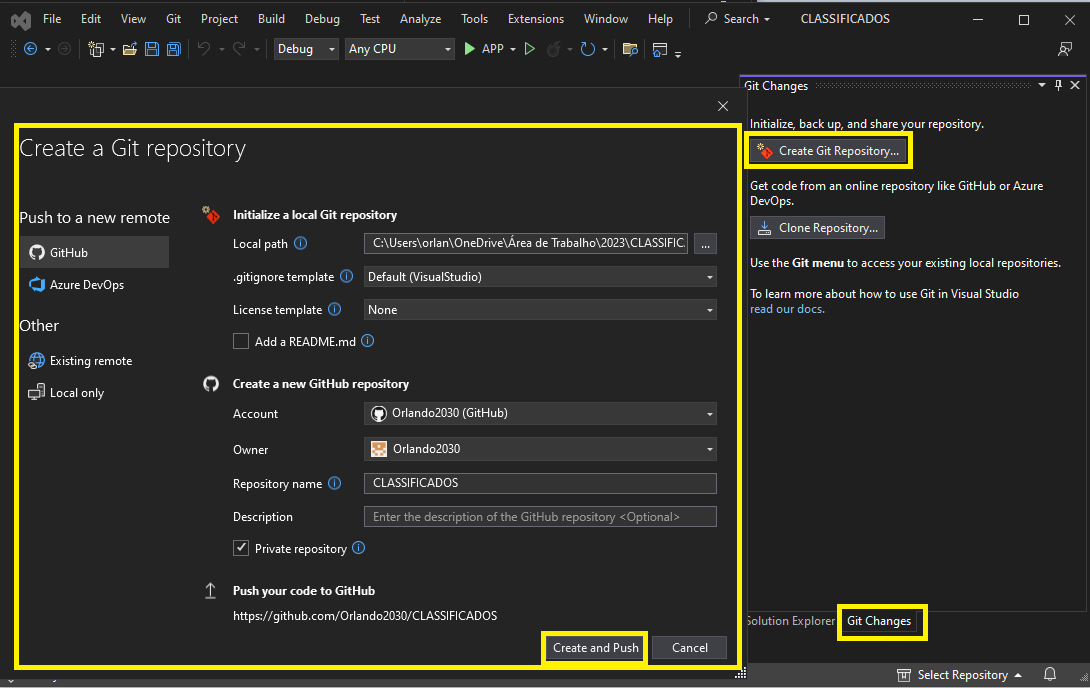
* 1. - Criar uma “Blank Solution”

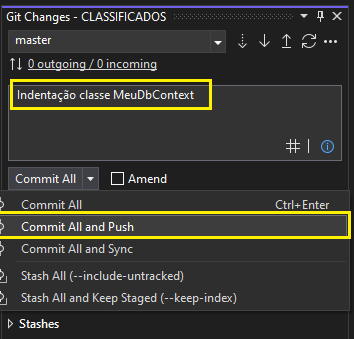


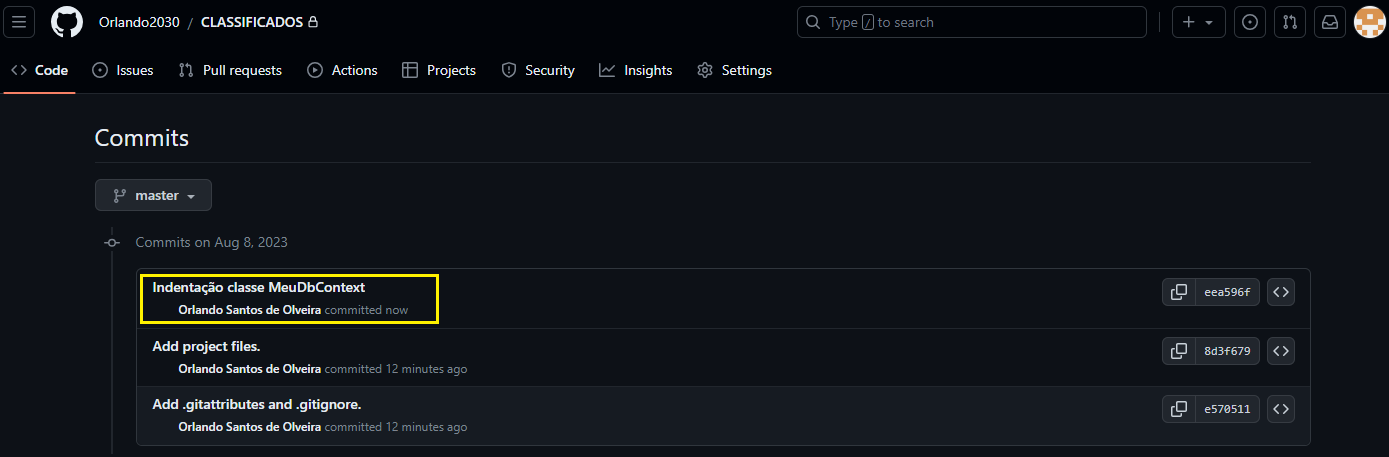




* 1. – Criação de repositorio Git





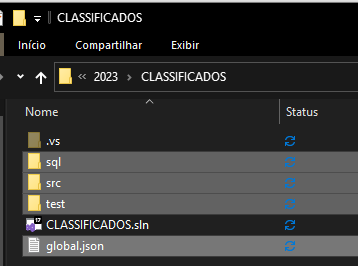


**git push:** Realiza o upload do conteúdo do repositório local para o remoto.

**git sync:** Primeiro executa um git pull que baixa o conteúdo do repositório remoto e atualiza o local.

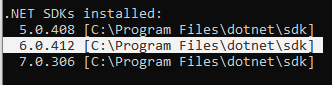
**git commit**: Permite que você crie um commit, ou seja, você consegue guardar o estado do seu repositório naquele momento.

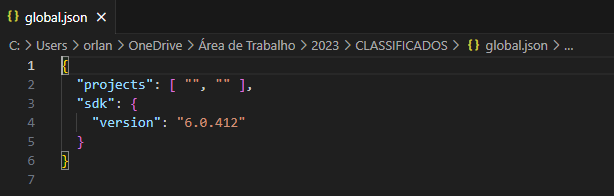
* 1. - Criar as pastas (sql, src, tests) na pasta da Solution.



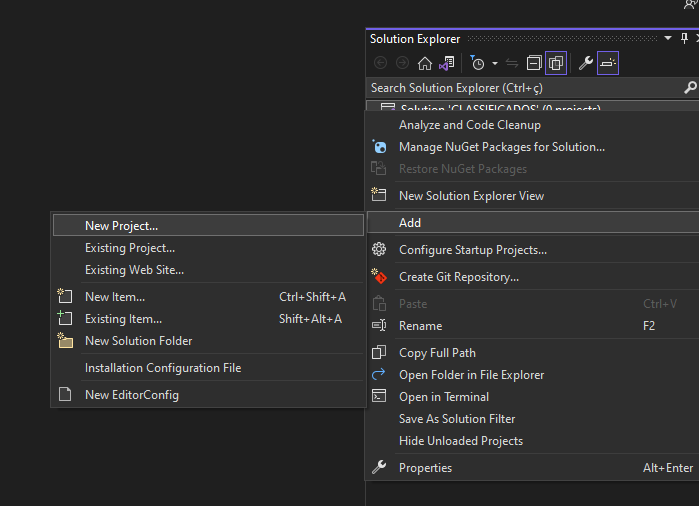
1.3 - Criar o arquivo “global.json” (Define qual versão do seu SDK vi ser utilizado)

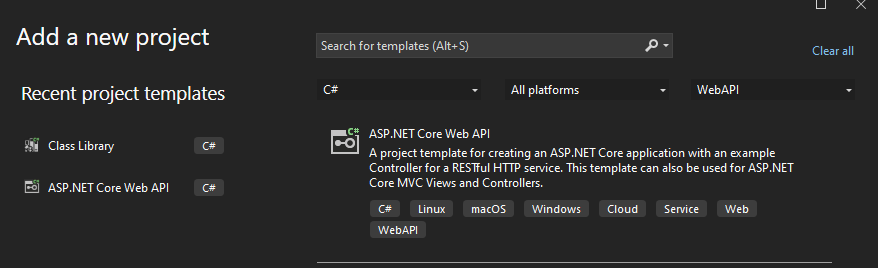
Verificando a versão do SDK instalado na maquina no CMD com o comando **dotnet - - info**

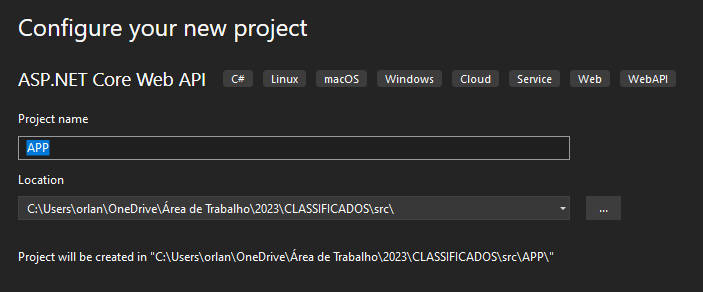


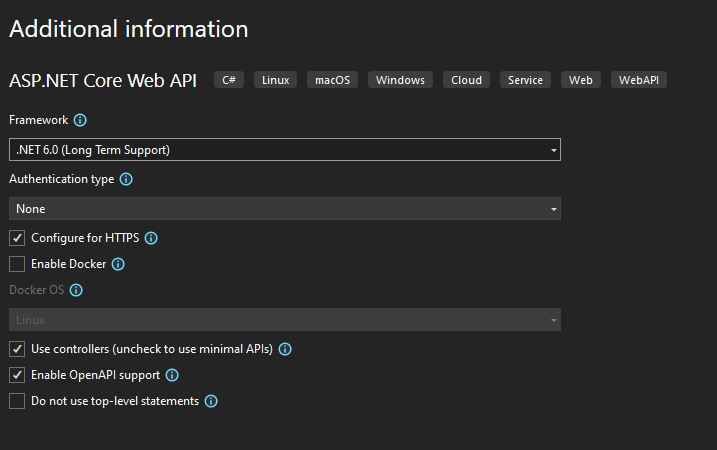


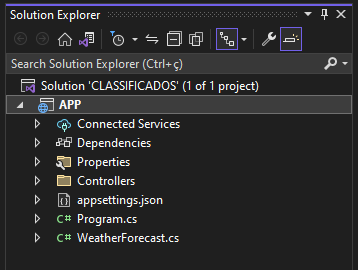
* 1. – (CAMADA DE APRESENTAÇÂO “App”) Add um novo projeto na Solution (ASP.NET Core Web Aplicattion (MVC)) dentro da pasta src



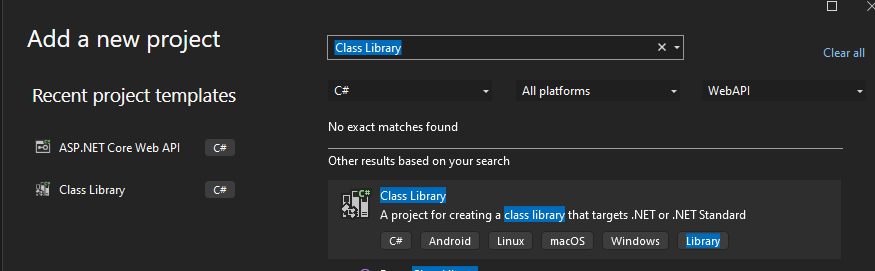


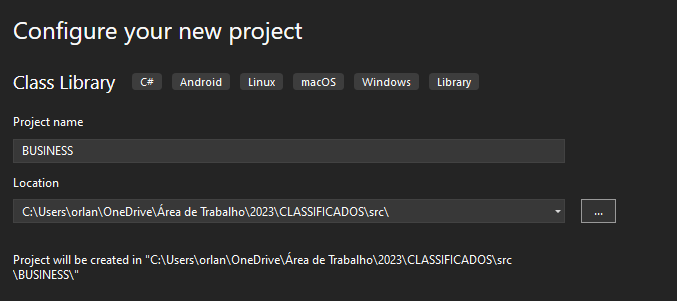


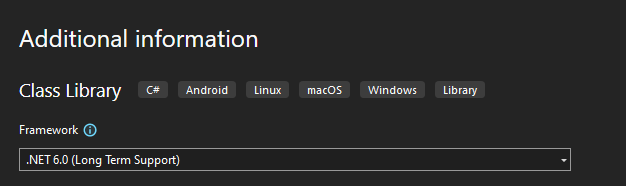


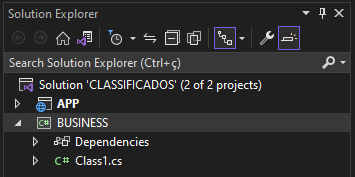


* 1. – (CAMADA DE NEGOCIO(Business) “Business”) Add um novo projeto na Solution (Class Library .Net Core) dentro da pasta src



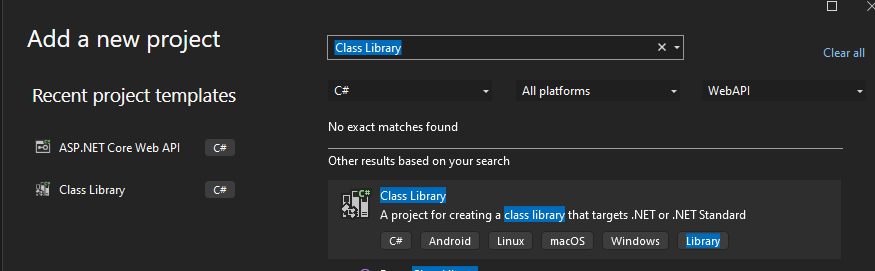


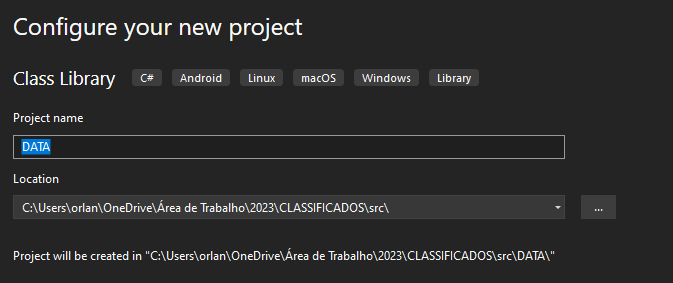


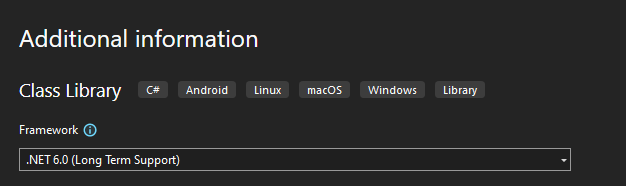


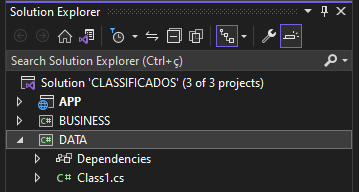
**Pode deletar o Class1.cs**

1.6 – (CAMADA DE DADOS(Data) “Data”)Add um novo projeto na Solution (Class Library .Net Core) dentro da pasta src

****

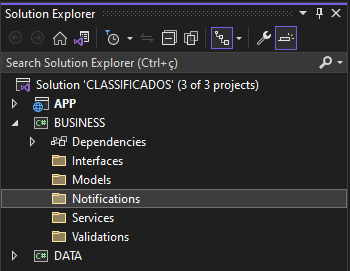
****

****

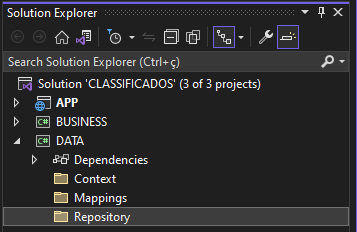


**Pode deletar o Class1.cs**

1.7 – Criar as pastas (Interfaces, Models, Notifications, Services, Validations) dentro da camada (Business)

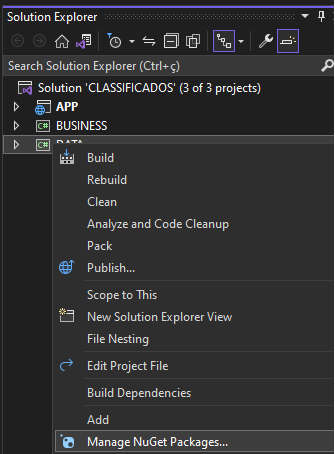


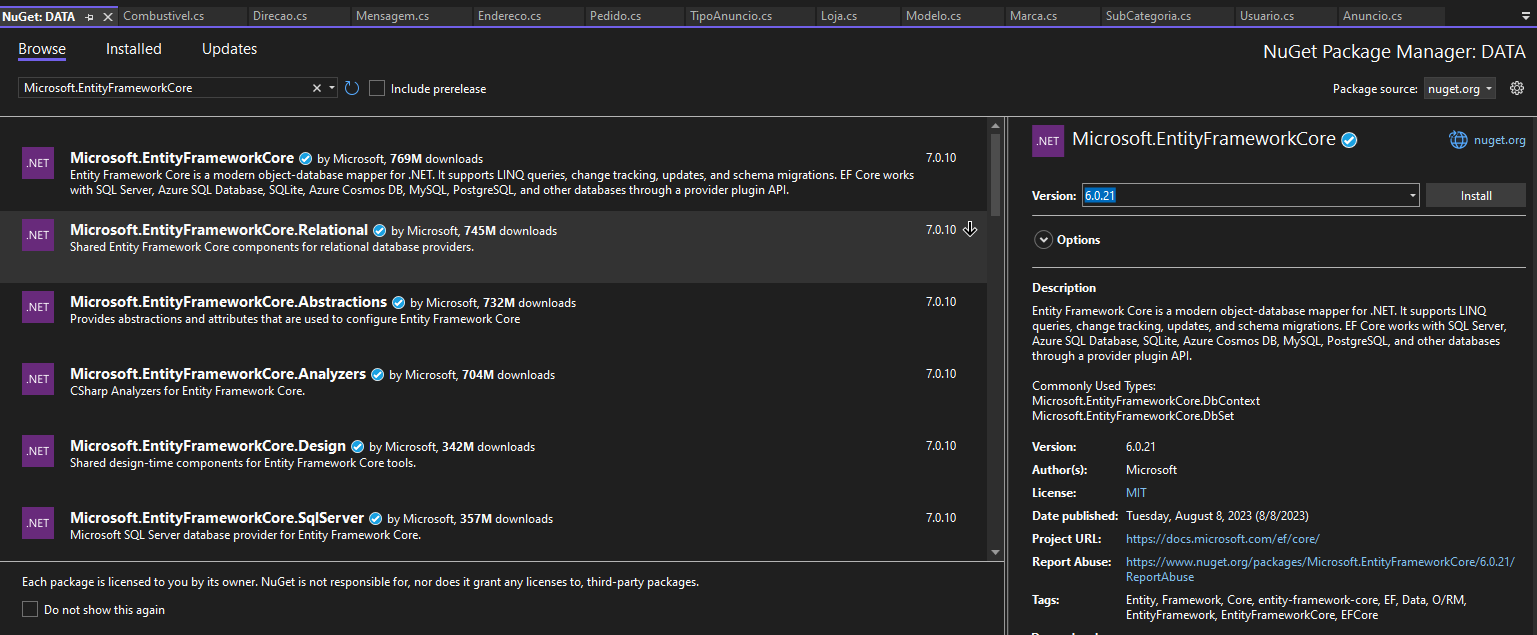
1.8 – Criar as pastas (Context, Mappings, Repository) dentro da camada (Data)

****

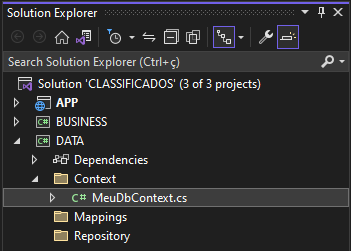
## **– Criação e Mapeando das Entidades**

* 1. – Criação das classes dentro (Business\Models)
  2. - Executar o comando Install-Package Microsoft.EntityFrameworkCore na camada “Date”

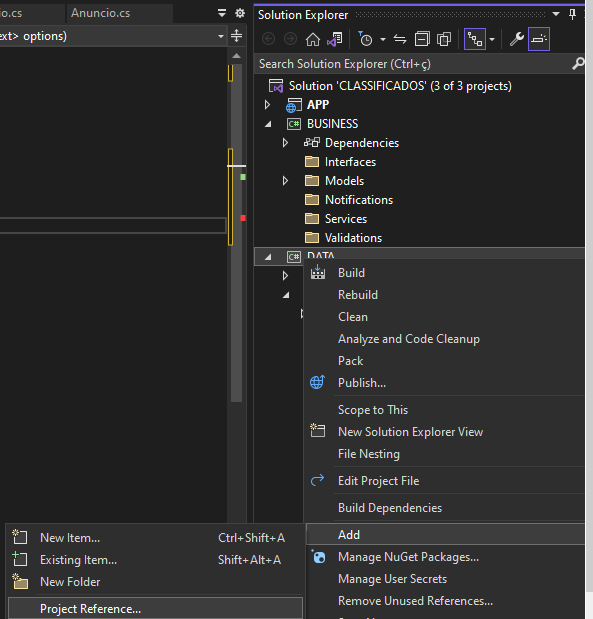


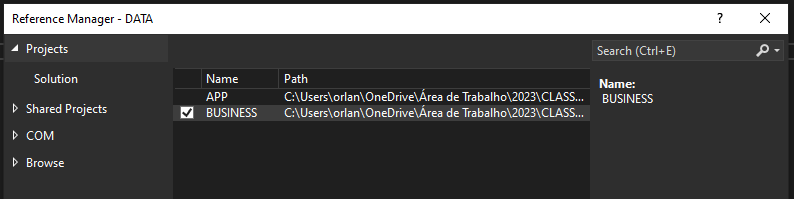


* 1. - Criar a classe “MeuDbContext” na pasta Context na camada Date

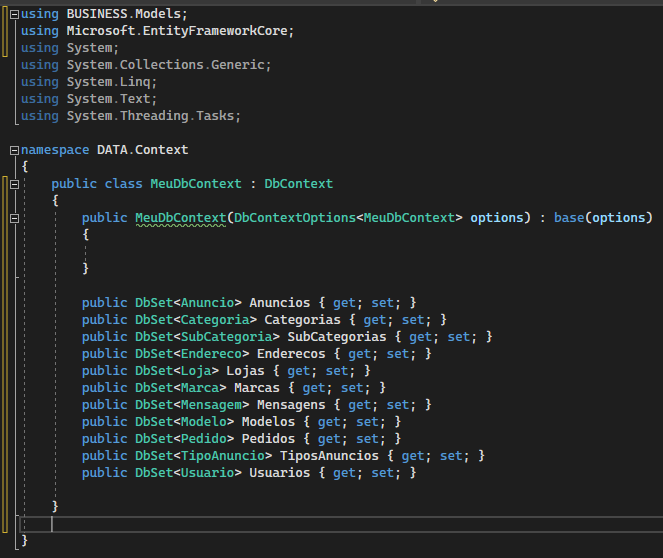


2.4 - Add na camada “DATA” as da camada de “Business”





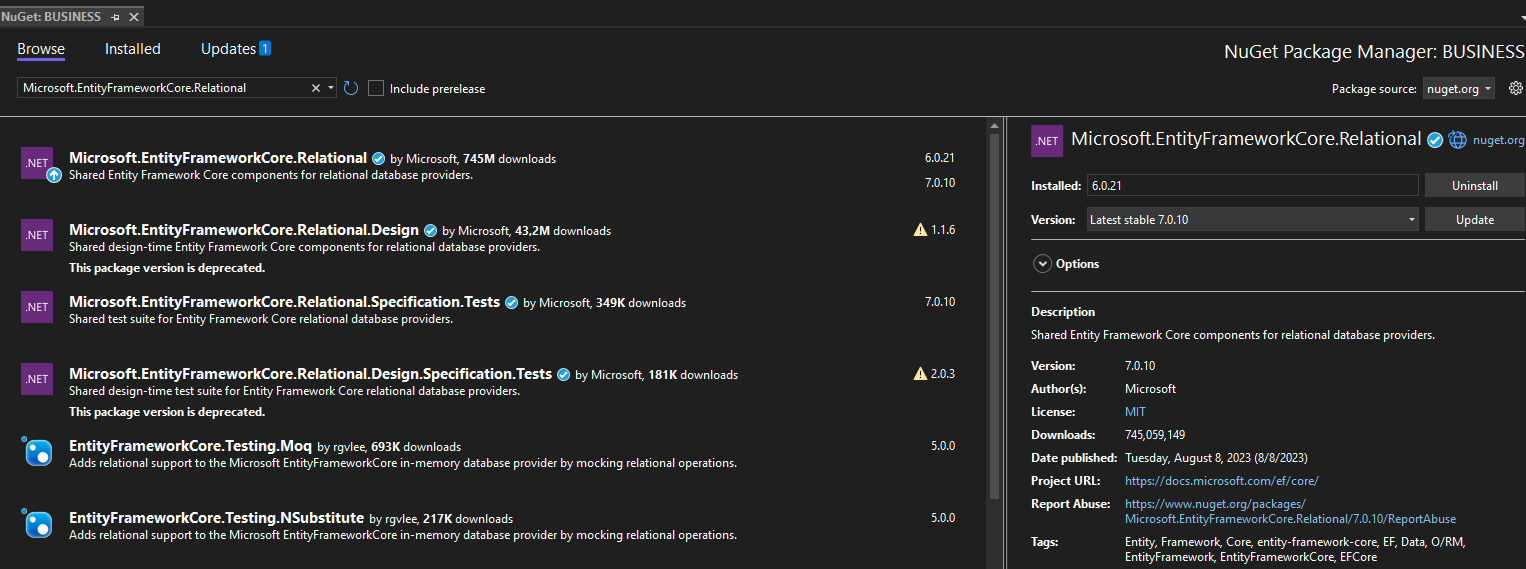
2.5 - Após a instalação é possível relaciona a classe MeuDebContext “: DbContext” e criar um .construtor (Public MeuDbContext(DbContextOptions options) : base(options)).



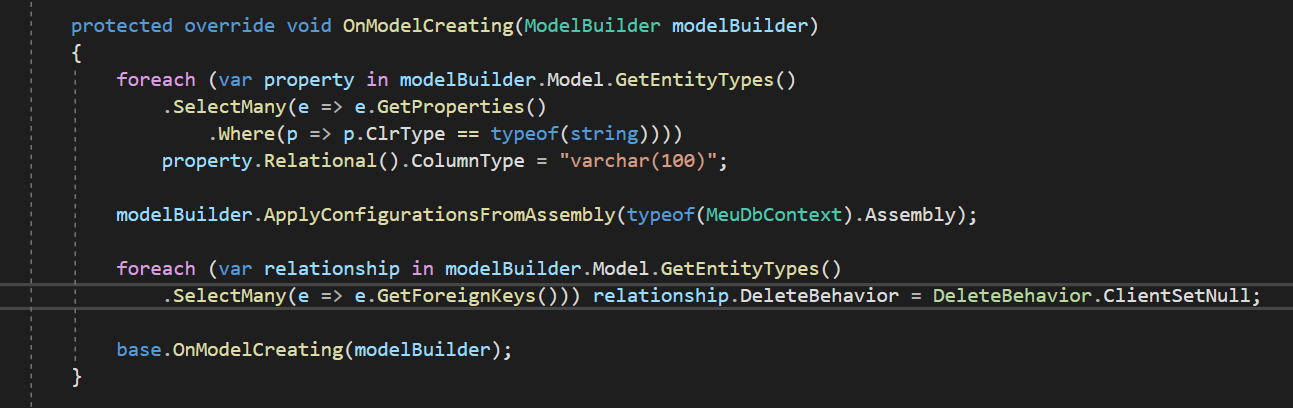
2.6 - Criar a classe “**Model**Mapping” e herda de : IEntityTypeConfiguration<**Model**>.

Dentro da classe é necessário implementar a interfece “public void Configure(EntityTypeBuilder<Produto> builder)” e dentro é feito o mapeamento dos campos e seus relacionamentos

**Obs.: é importante instalar o Microsoft.EntityFrameworkCore.Relational**

****

2.7 – Adicionando na class MeuDbContext



modelBuilder.ApplyConfigurationsFromAssembly(typeof(MeuDbContext).Assembly);

2.8 – Desabilitação do cascade delete (Isso impe que ao excluir, exclua seus filhos juntos)

foreach (var relationship in modelBuilder.Model.GetEntityTypes()

.SelectMany(e => e.GetForeignKeys())) relationship.DeleteBehavior = DeleteBehavior.ClientSetNull;

2.9 – Garantir que as colunas do tipo string esquecidas de ser mapeadas fique varchar(100) e não varchar(MAX)

foreach (var property in modelBuilder.Model.GetEntityTypes()

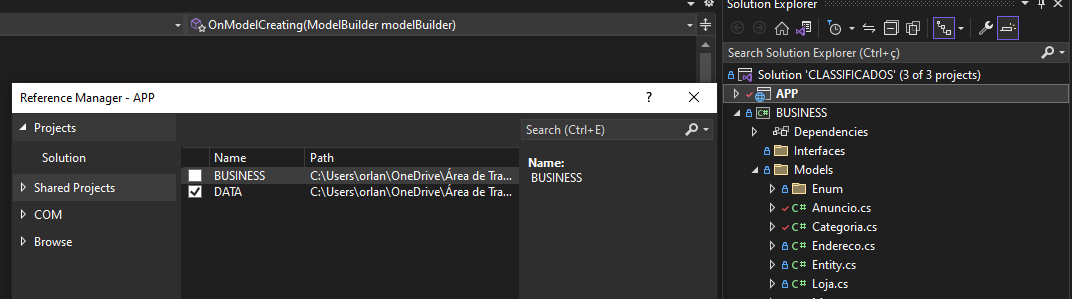
.SelectMany(e => e.GetProperties()

.Where(p => p.ClrType == typeof(string))))

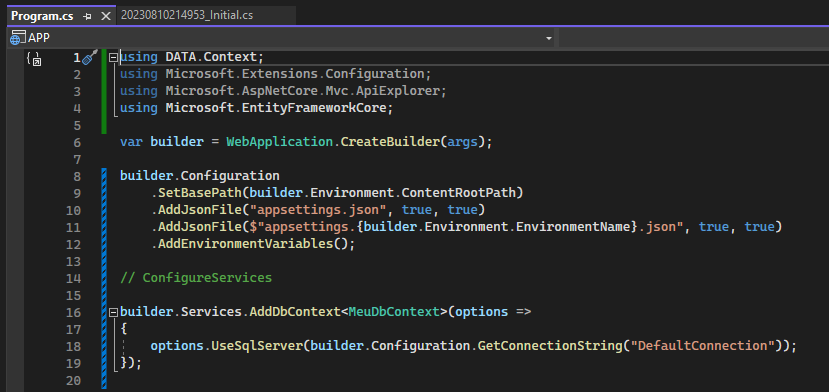
property.Relational().ColumnType="varchar(100)";

2.10 – Configuração do DbContext na aplicação

Ir nas dependencies no App e add a reference Data.



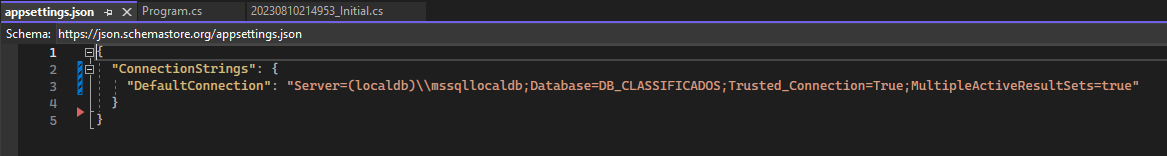
2.11 – Depois ir no Program.cs e add um novo **services.AddDbContext**



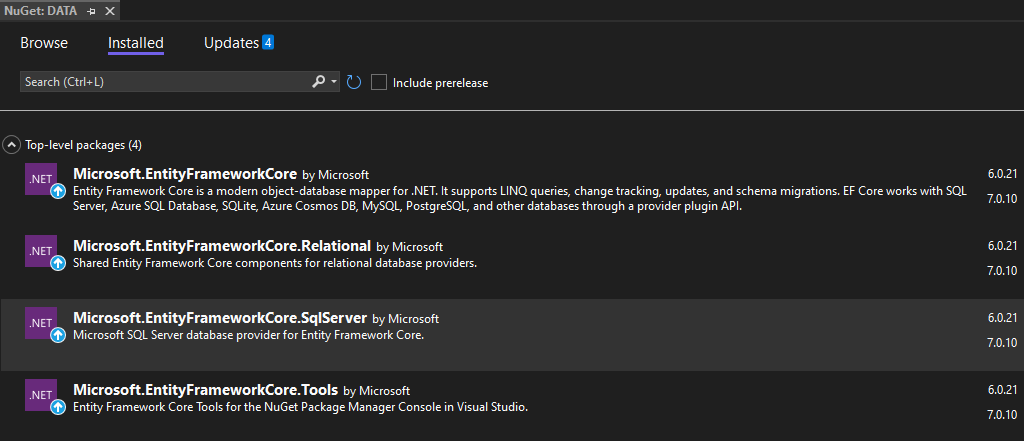
services.AddDbContext<MeuDbContext>(options =>

options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));

2.12 – Alterar o nome do Database para DB\_CLASSIFICADOS no arquivo appsettings.json da camada **App**

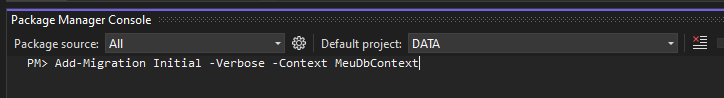


2.13 – É importante intalar na camanda de **Data NuGets**



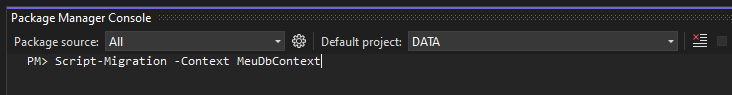
2.14 – Criar uma Migration: (É necessário selecionar a camada (DATA) e só do ser executada em apenas um contexto)

PM> Add-Migration Initial -Verbose -Context MeuDbContext



2.15 – Criação dos scripts das tabelas

PM> Script-Migration -Context MeuDbContext



2.16 – Criação da base da aplicação

PM> Update-Database -Context MeuDbContext

