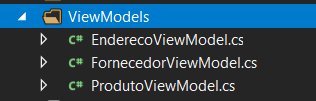
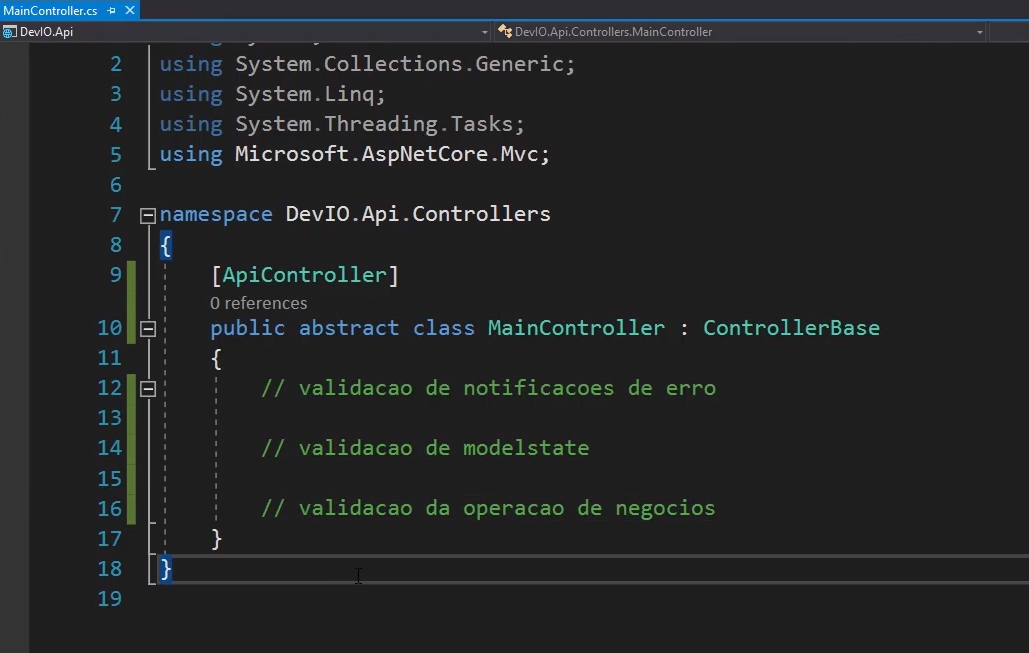


## Implementando DTOs (ViewModels)

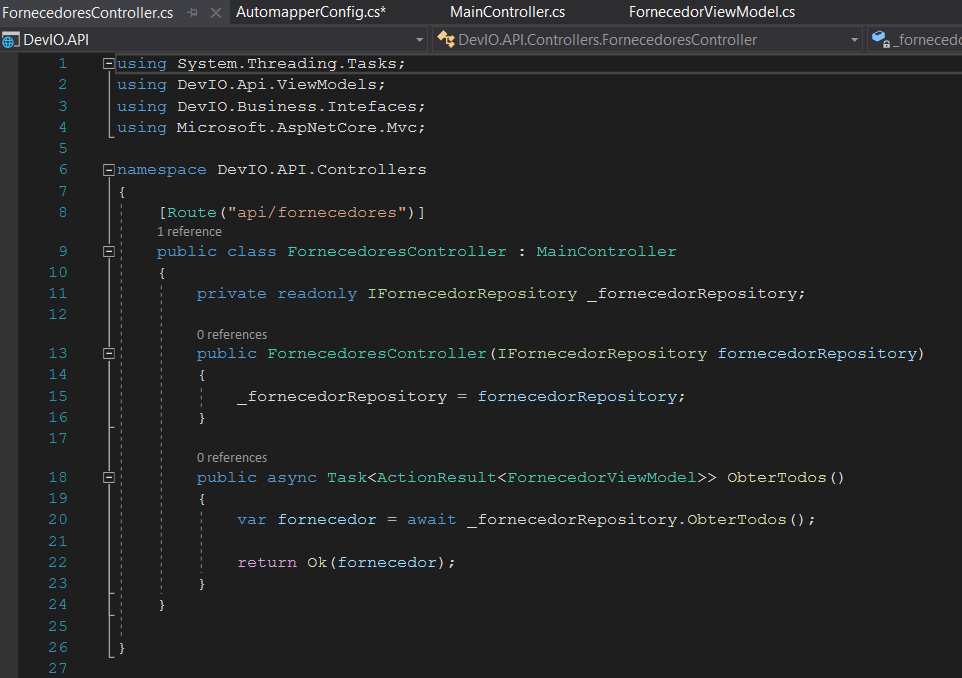
Criar as ViewModels:



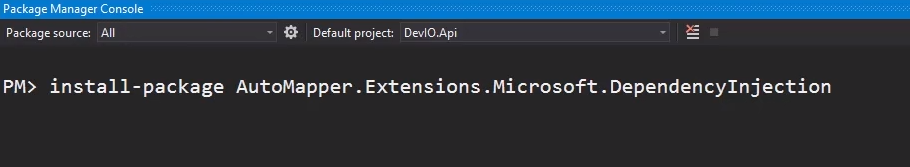
Criar a MainController



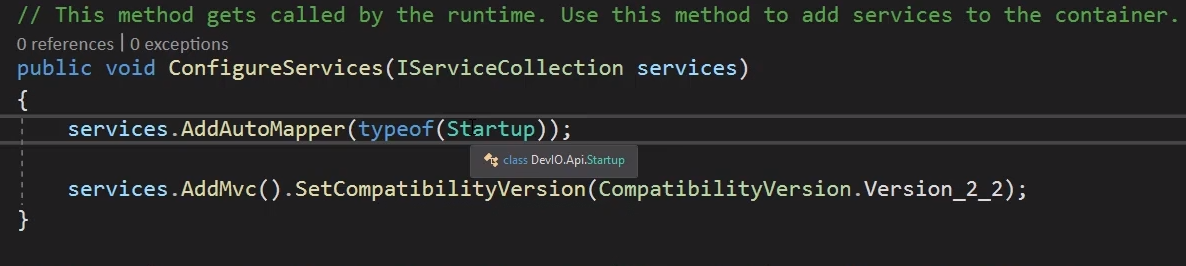
- Criar a classe FornecedorController herdando da Main Controller:



- Instalar o automapper para conversão das entidades em ViewModels.

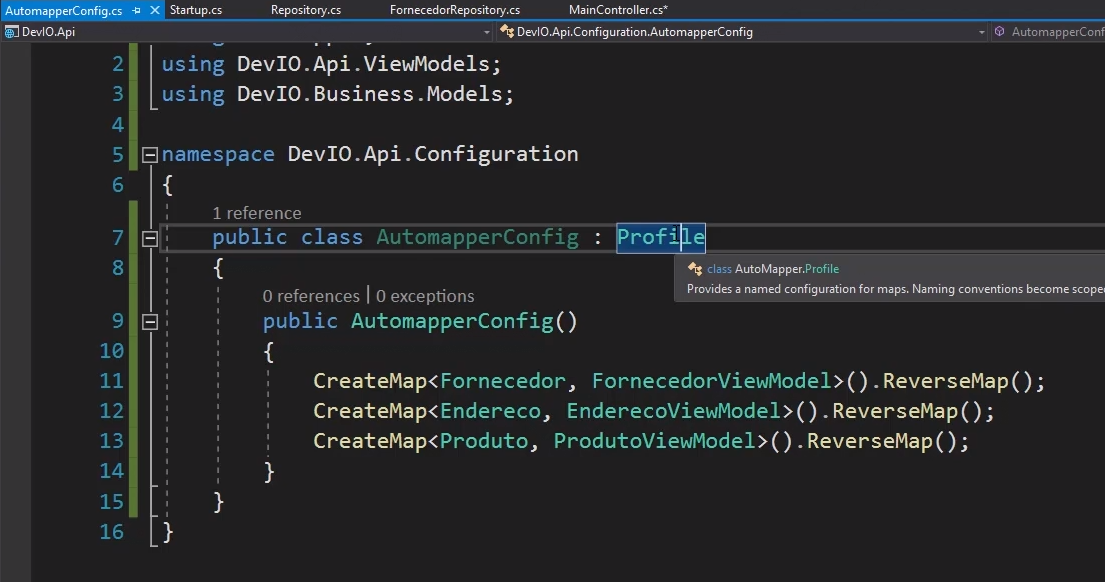


Para utilizar configurar na startup.cs



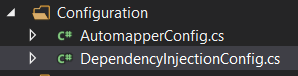
Configurar a Lista de Para, criando a classe AutomapperConfig:

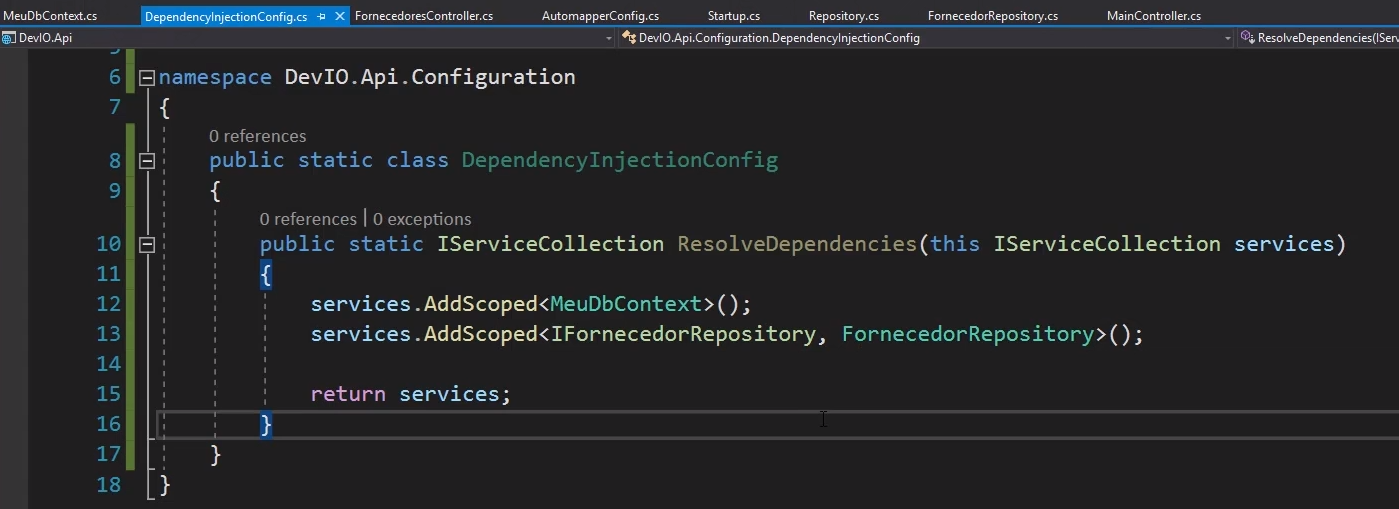




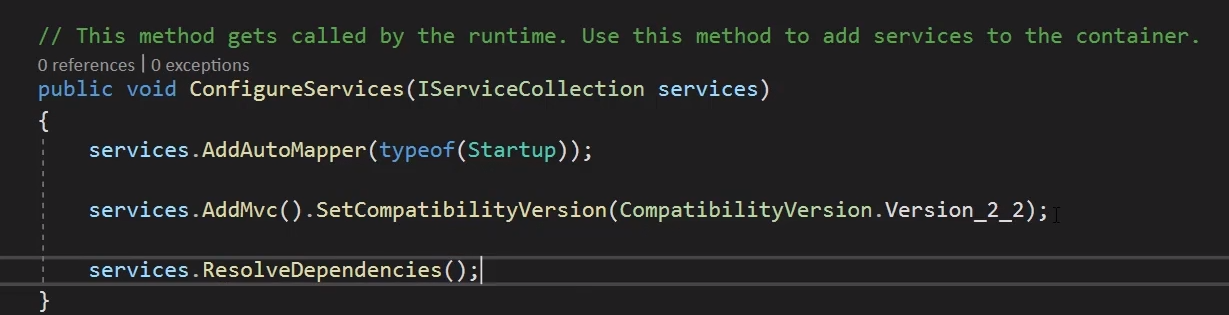
Injetando o automapper como dependencia:

Criar a classe DependencyInjectioConfig:

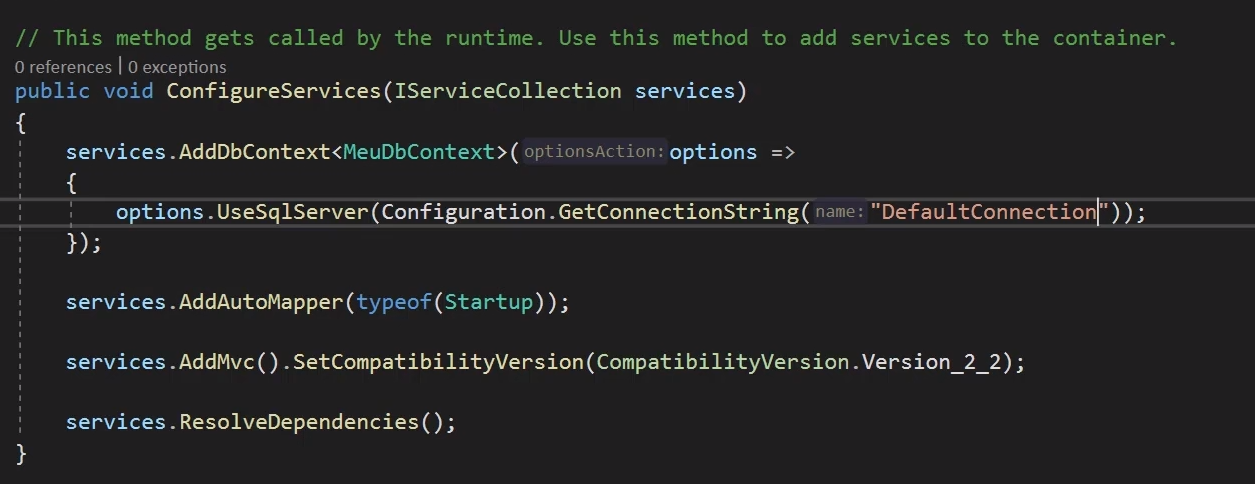




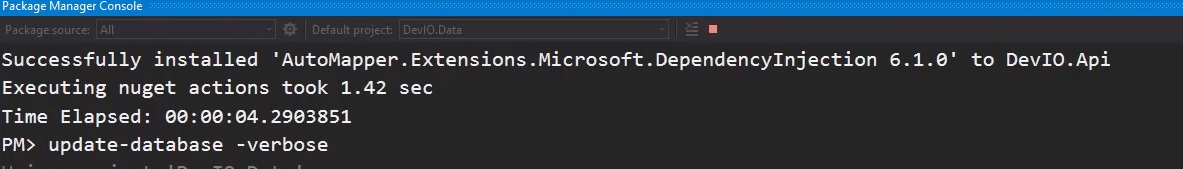
Chamar na Startup:



Configurar o DBContext - EntityFramework:



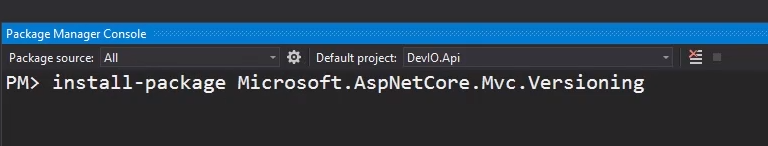
Criar o banco de dados, apontando para o projeto data:



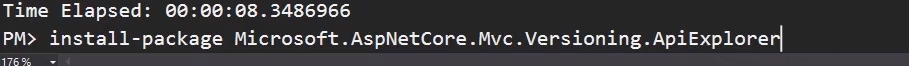
Verionamento de API.

Instalar os seguintes pacotes:

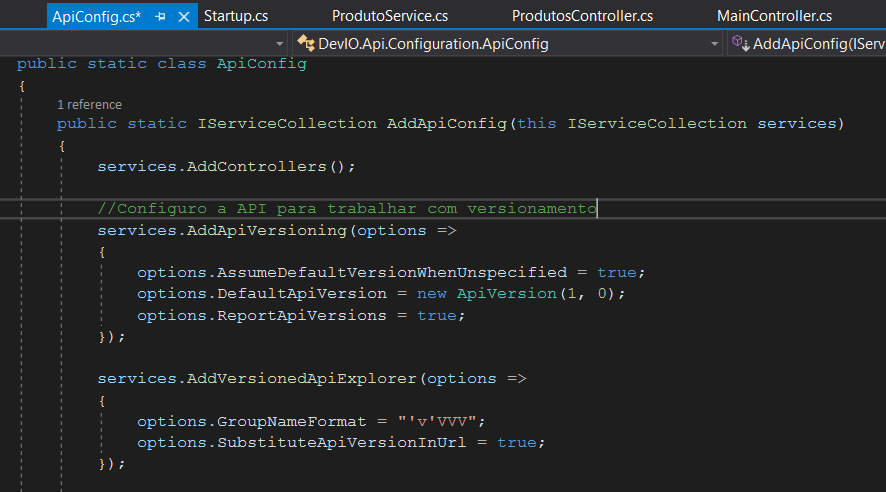
install-package Microsoft.AspNetCore.Mvc.Versioning



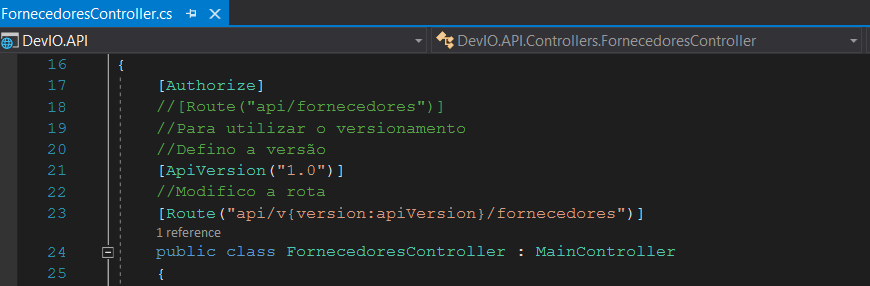
install-package Microsoft.AspNetCore.Mvc.Versioning.ApiExplorer



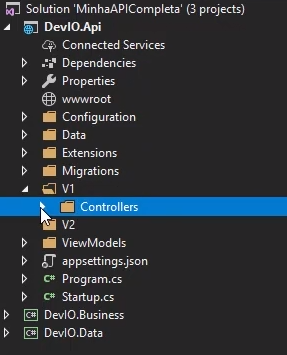
Configurar nossa API para Trabalhar com versionamento:



Para utilizar o versionamento, nas controllers, adicionar o atributo ApiVersion e modificar a o rota para utilização da versão da API.



Criar pastas de versionamentos para as controllers, ex. V1, V2... e copiar a pasta controller para dentro dela.



Ajustar o Namespace, adicionando V1, em todos os arquivos do projeto, desta forma não terei problema de ter a classe com o mesmo nome.

