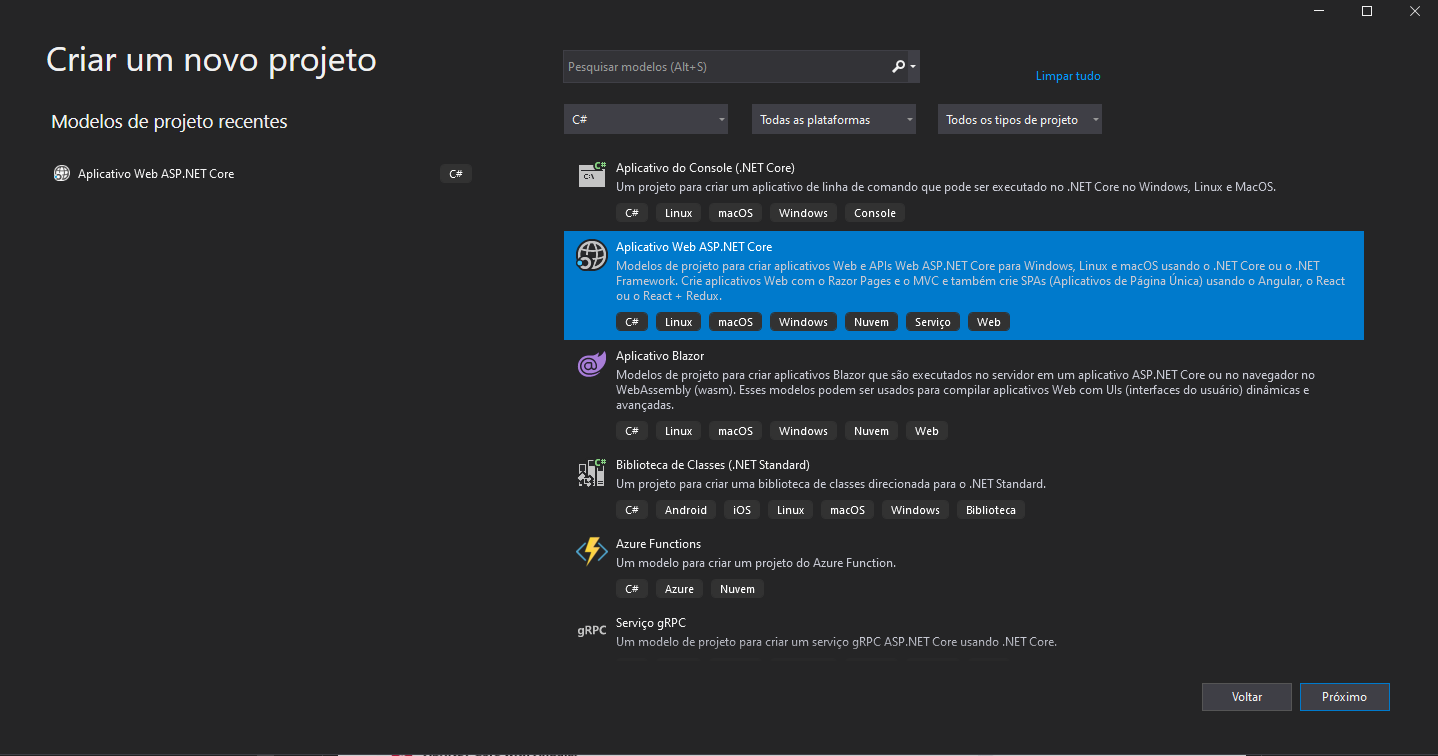
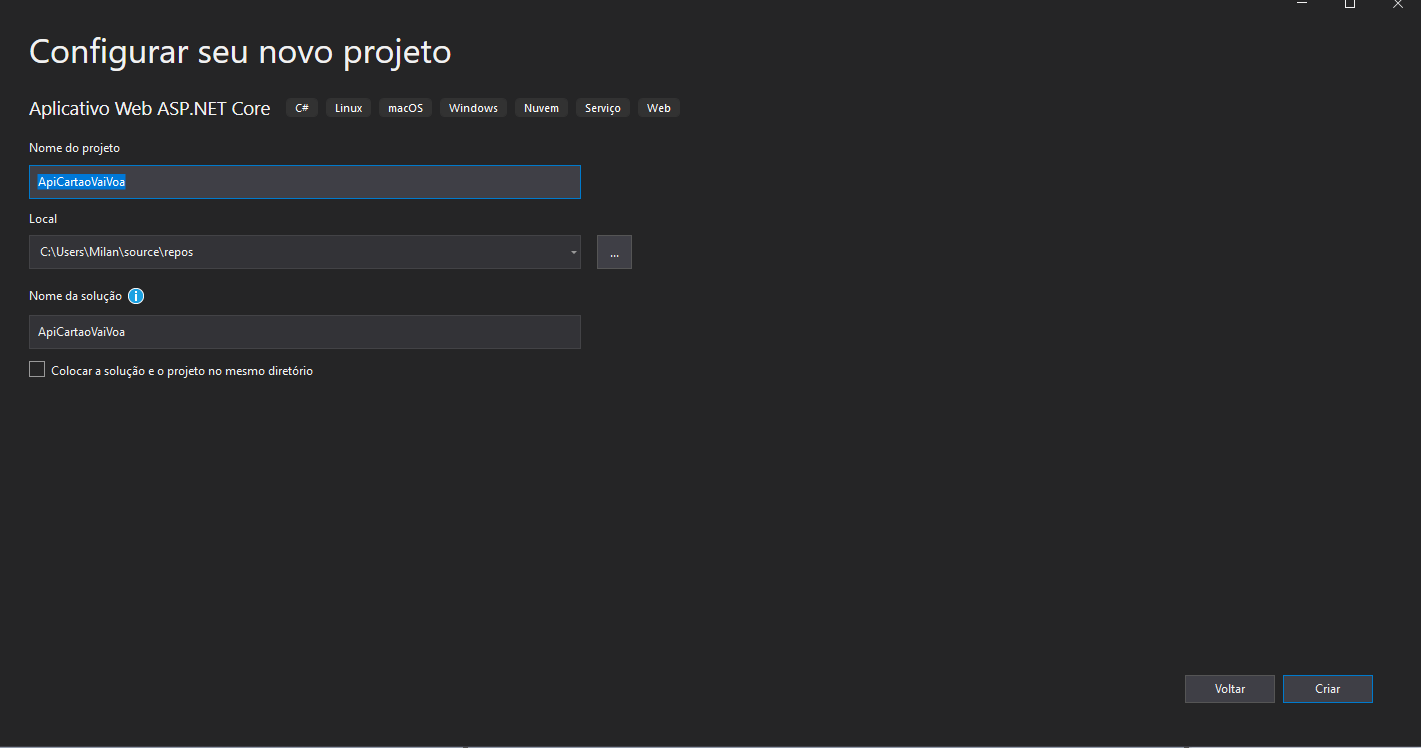
Criação do Projeto

**API C#+ ENTITY FRAMEWORK**

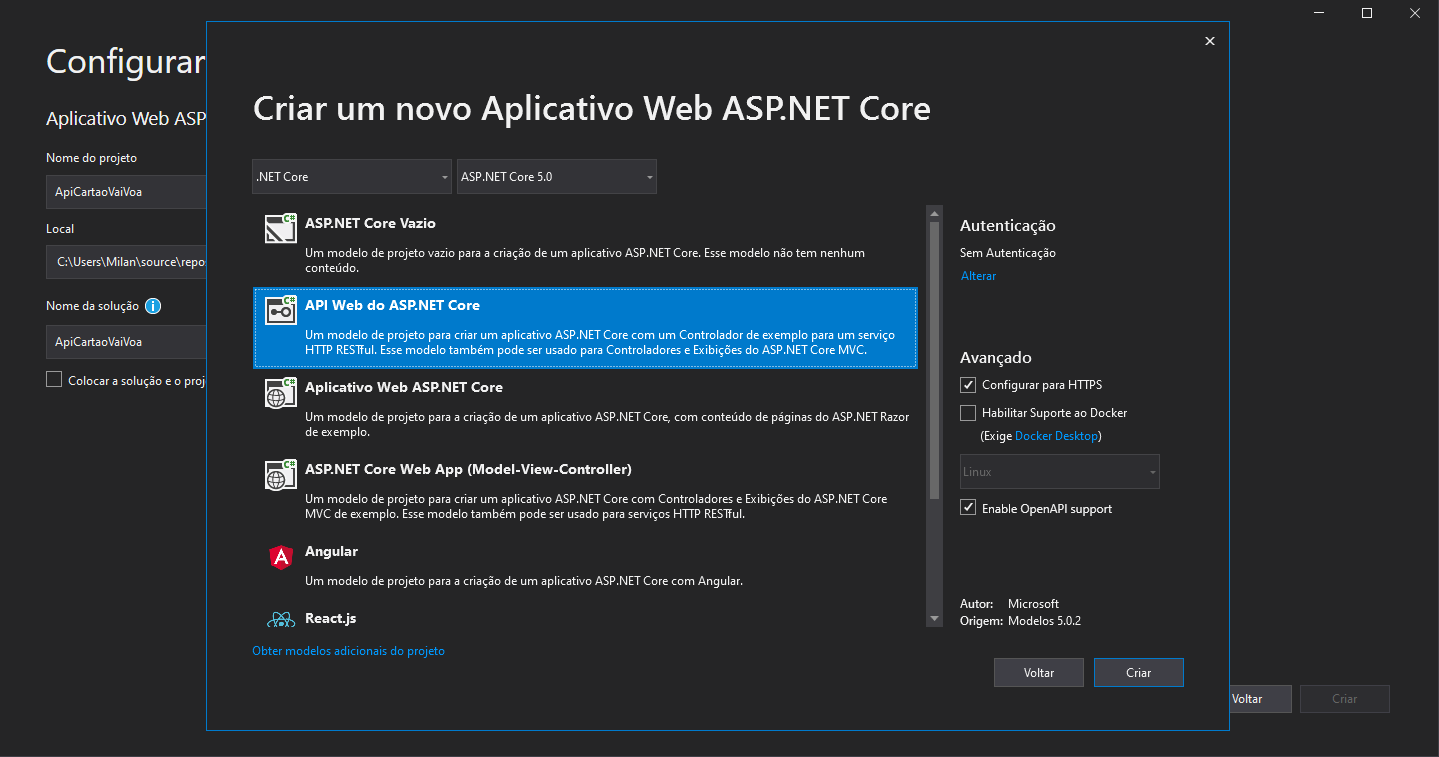
Para começarmos é necessario termos uma IDE instalada no nosso sistema, para a criação da nossa API utilizaremos o Visual Studio 2019.



Na pagina de criação do projeto iremos criar um aplicativo Web ASP.NET Core.

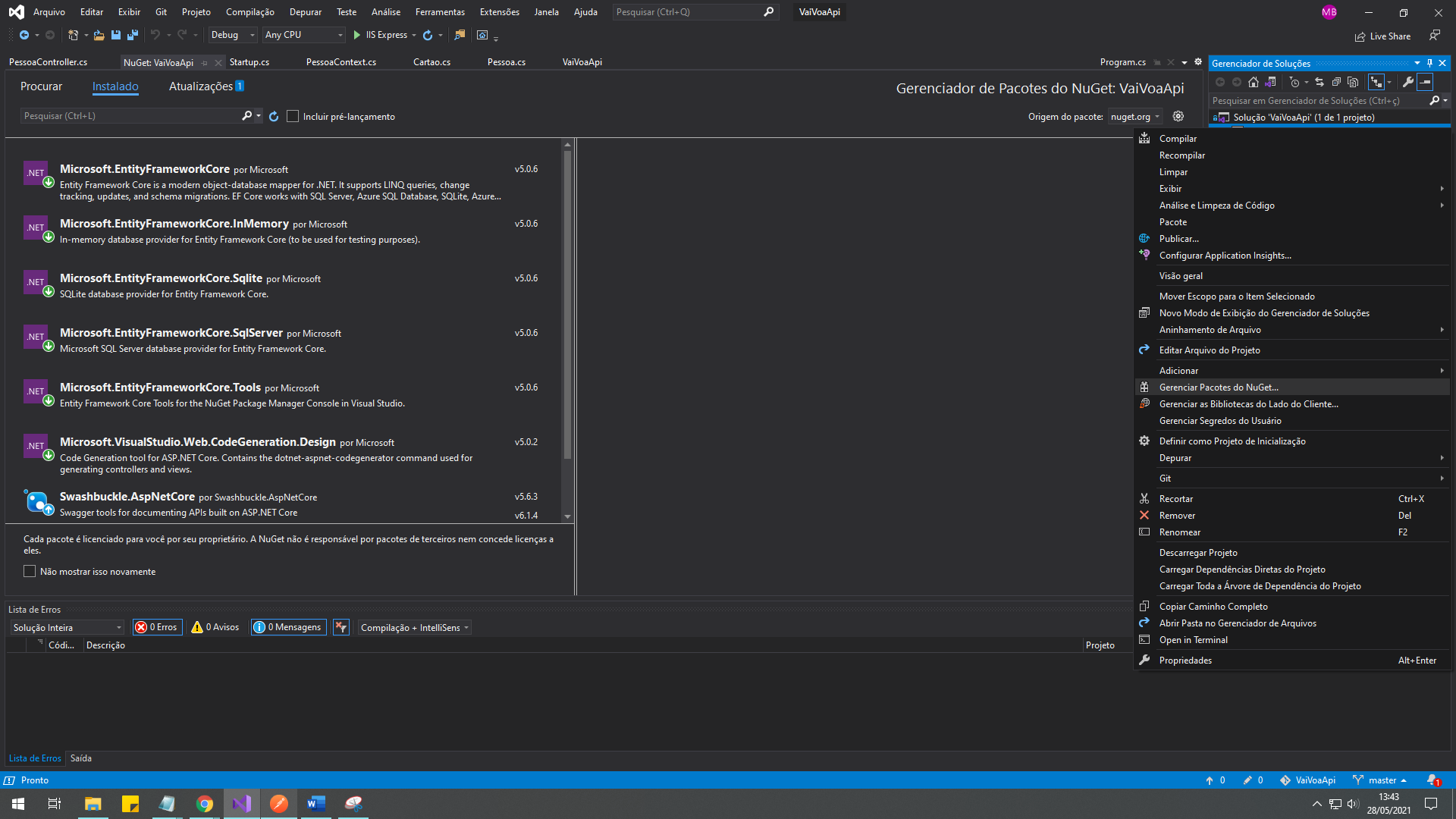


Na tela seguinte nomearemos o nosso projeto e também podemos selecionar o local onde a pasta contendo o projeto sera salvo, para posterior recuperação do mesmo.



O Visual Studio tem varios modelos pré estruturados de projetos para serem utilizados em novos projetos, para aceleração do nosso projeto utilizaremos um dos modelos que sera o Api Web ASP.NET Core que traz o modelo de API que é oque nos desenvolveremos.

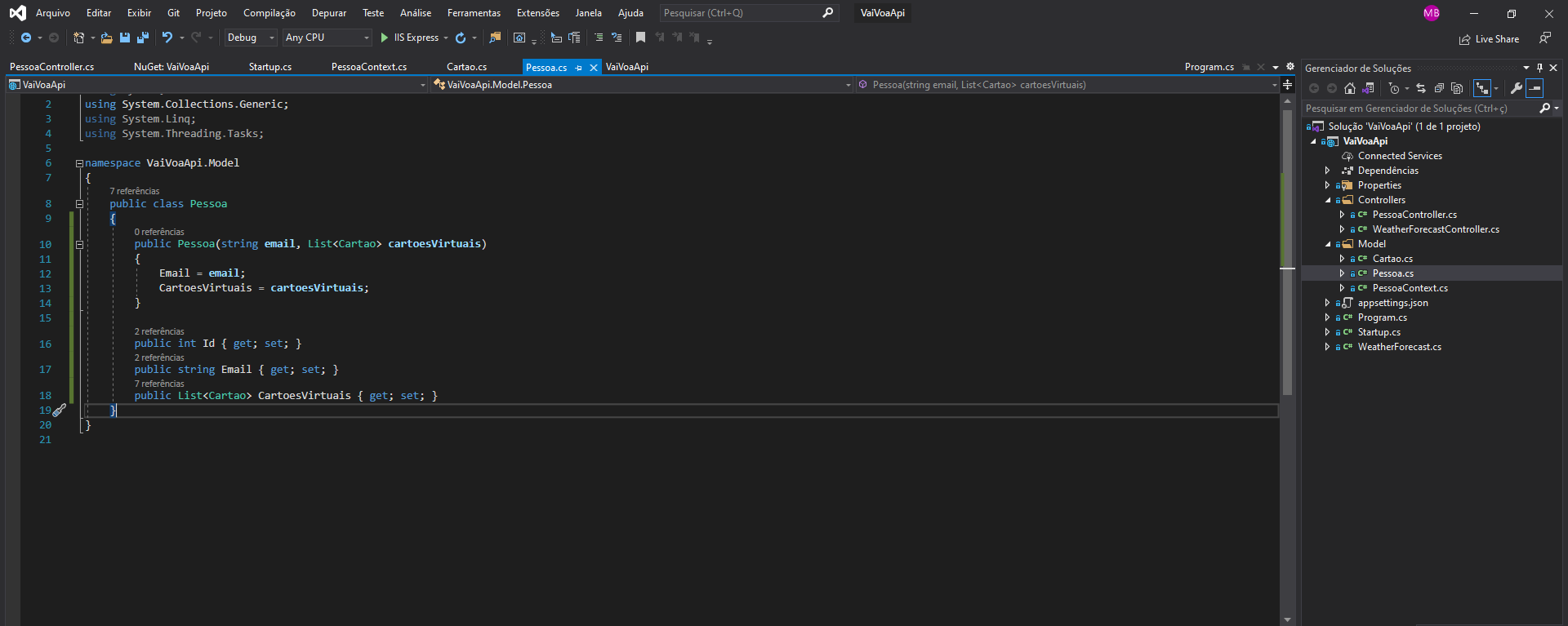
EntityFramework



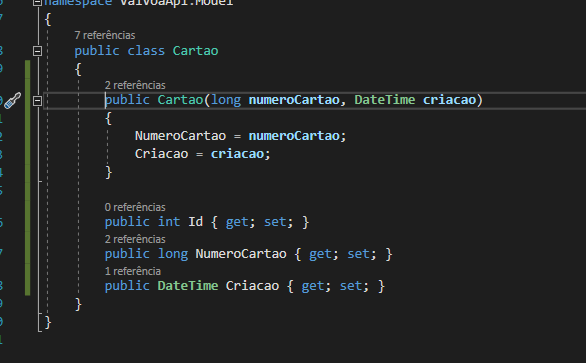
Para continuarmos o desenvolvimento precisamos intalar alguns pacotes do NuGet para termos acessos a bibliotecas que possuem funções que utilizaremos, as Bibliotecas são as seguintes Microsoft.EntityFrameworkCore, Microsoft.EntityFrameworkCore.InMemory, Microsoft.EntityFramework.Tools, no nosso projeto para não termos a necessidade de um Banco de Dados utilizaremos a biblioteca InMemory que guardara os dados da nossa API em memoria em tempo de execução, para apenas exemplo instalei também a Microsoft.EntityFrameworkCore.SqlServer para mostrar que o Entity possui bibliotecas pré estabelecidas para Bancos de Dados não sendo apenas o SqlServer mas também PostGressSQl,Mongo Db entre outros.

Organização do Projeto MVC

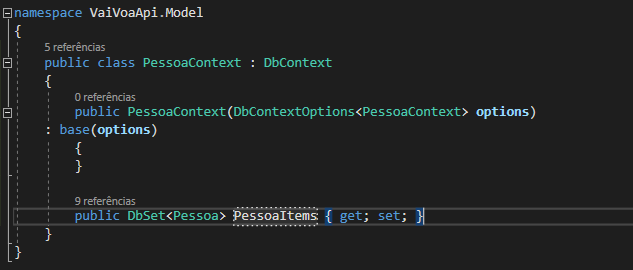
Para desenvolvimento utilizaremos o padrão MVC, Model,View,Controller, por ser uma API não teremos a pasta View, a model tem como função modelar nossas classes exemplificando, precisaremos de uma classe Pessoa que possui varios cartões então modelaremos ela com seus tipos de dados respectivo.



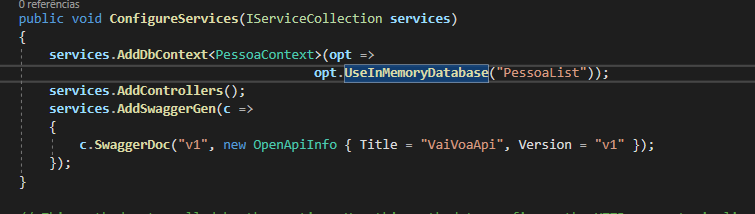
Após a criação da classe pessoa precisamos criar a classe que possui o cartão que estara dentro da nossa Lista de Cartões da pessoa.



Após a criação dos modelos precisamos fazer a criação de um Contexto, contextos são criados durante o processo de ativação para objetos que são configurados para exigir determinados serviços automáticos, como sincronização, transações, ativação Just-in-time, segurança e assim por diante.

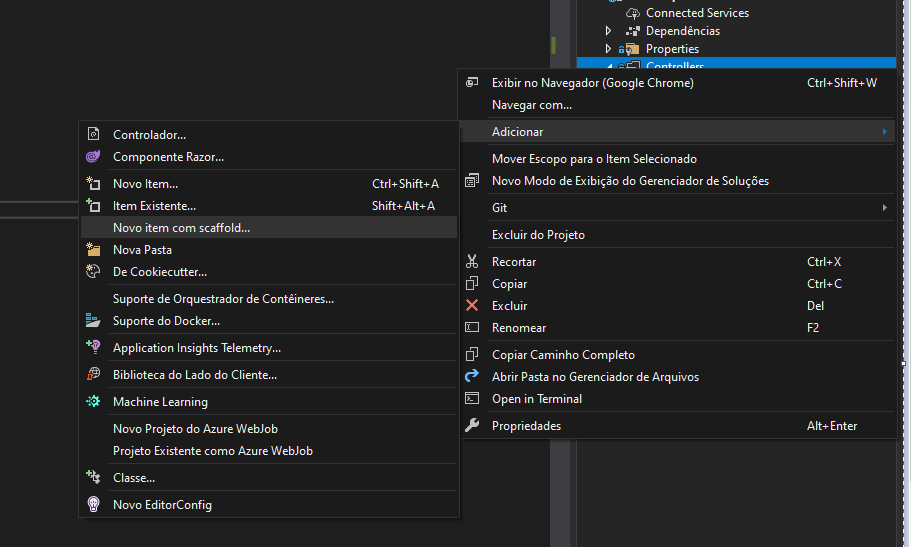


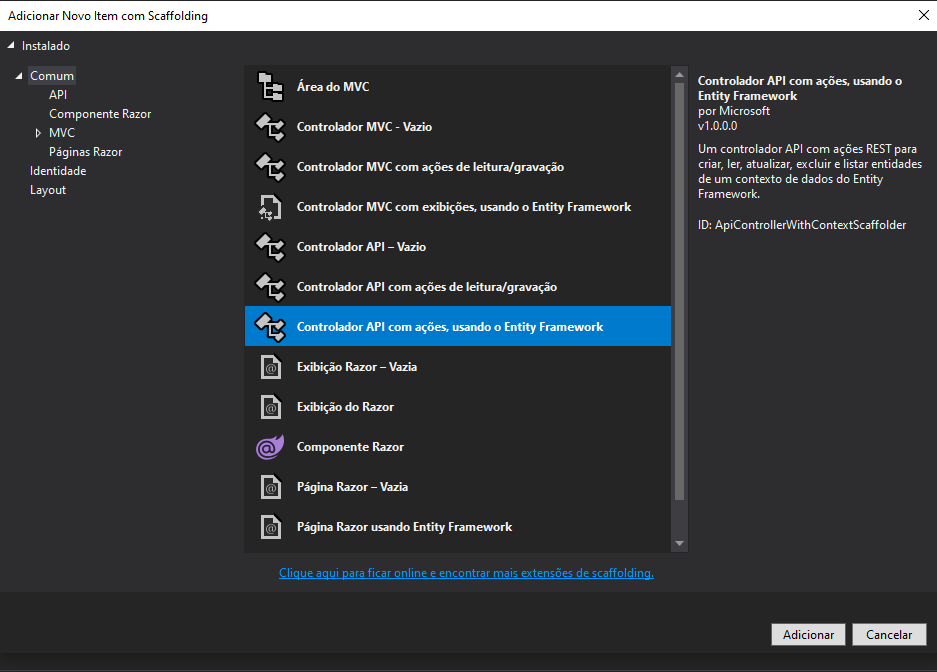
Após a criação de toda nossa pasta de Modelo temos nossa mini Base de Dados pronta pra ser utilizada, porém para utilizarmos o Contexto que criamos precisamos definilo como um serviço na classe startup do projeto.

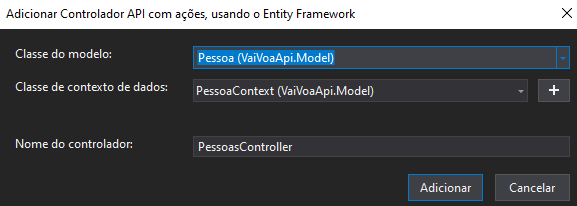


Após isso feito criaremos agora a controlladora que controlara os objetos criando dando as ações da API,PUT,POST,GET,DELETE etc.

Como modelamos nossa base de dados em classes e definimos um contexto podemos utilizar uma das função do entity framework que criara nossa controladora automaticamente sendo ela o scaffold.







Após este passo a passo feito temos nossa controladora com os principais metodos Básicos da Api Gerado e dando uma boa base para construção de novas funções para iniciantes. Observe bem a classe controladora gerada pelo Entity e vamos para mão na massa.

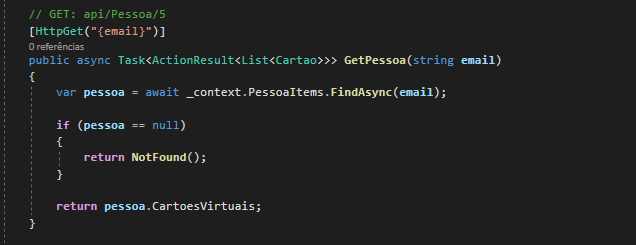
Faremos 2 funções novas para aprendizado.

A primeira função receberá o email da pessoa e retornará um objeto de resposta com o número do cartão de crédito.



A função apresentada gera o novo cartão gerando de metodo totalmente aleatorio os doze digitos do mesmo e após a geração do cartão verifica se a pessoa ja existe em nossa base de dados em Memoria, caso ja tenha sido cadastrada anteriormente é adicionada o novo Cartão Virtual a sua lista de cartões ou caso não tenha registro a nova pessoa é registrada em nossa base de dados contendo o seu Novo cartão.

A segunda função ira listar, em ordem de criação, todos os cartões de crédito virtuais de um solicitante, veja nosso solicitante possui uma lista de cartões que foi adicionada em ordem de criação por isso não precisamos criar nenhum metodo em um primeiro momento para saber a ordem, pois a lista ja esta em ordem.



E com isso finalizamos o processo de criação de nossa API c# Entity Framework.