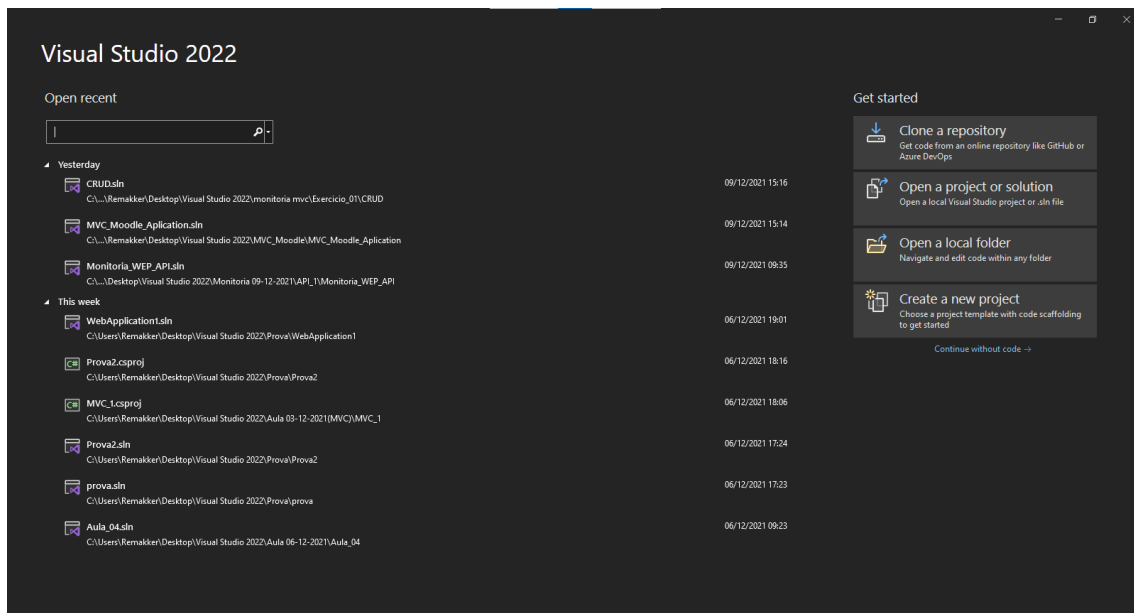


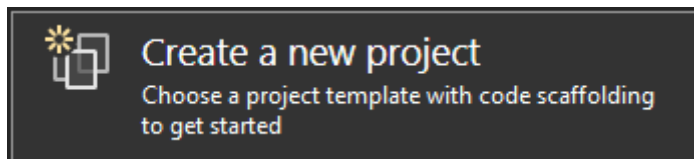
C# - MVC

Alguns adendos, estou usando o Visual Studio 2022

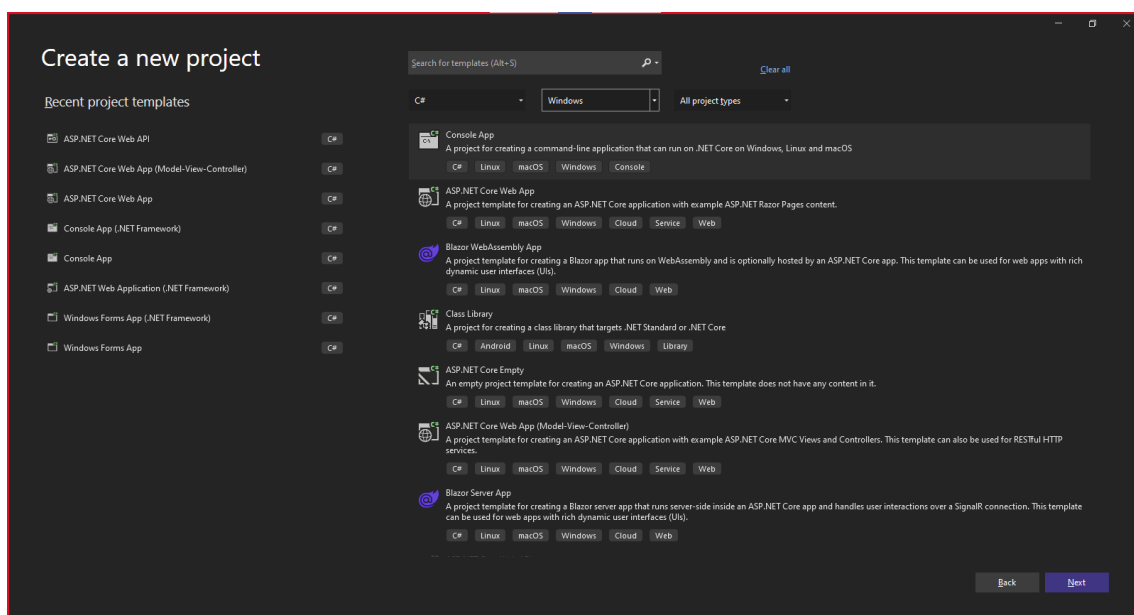
1º Abra o Visual Studio 2022:



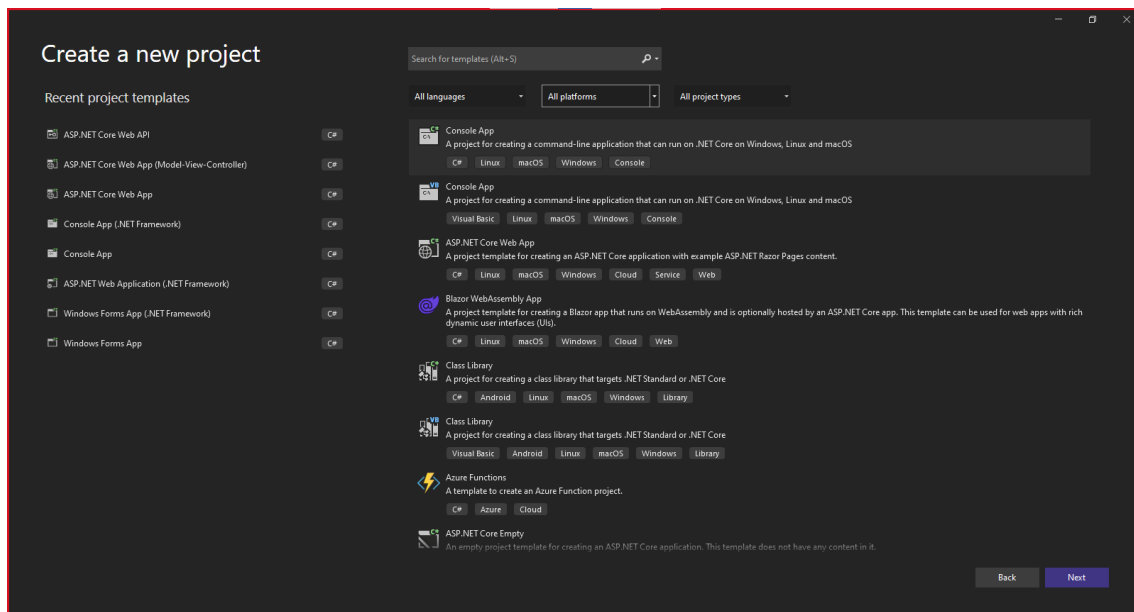
2º Clique em “Create a new Project”:



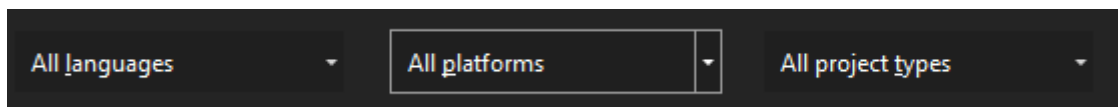
3º Provavelmente sua página estará assim após clicar em “Create a new Project”, porque foi os últimos filtros usados:



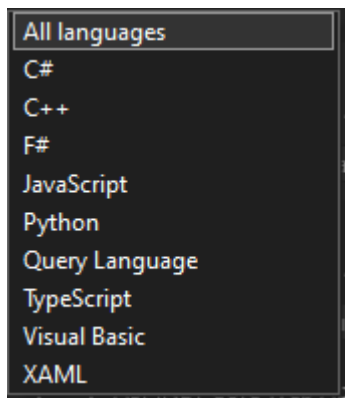
Caso esteja ‘zerado’, quando digo zerado, quero dizer isso:



Note que os filtros, voltaram para o padrão:



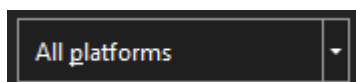
4º Selecione a linguagem:



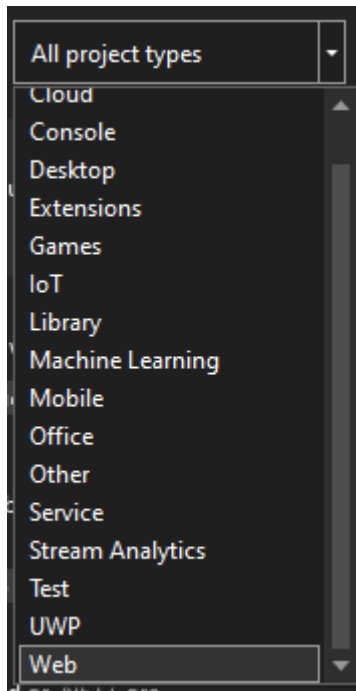
5º No caso, C#:



6º Como o .Net Core, ele é multiplataforma, não interessa a plataforma:

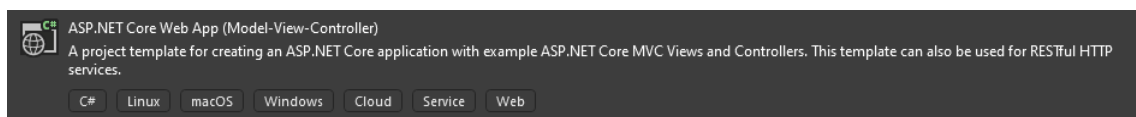


7º Vamos escolher o tipo de projeto:

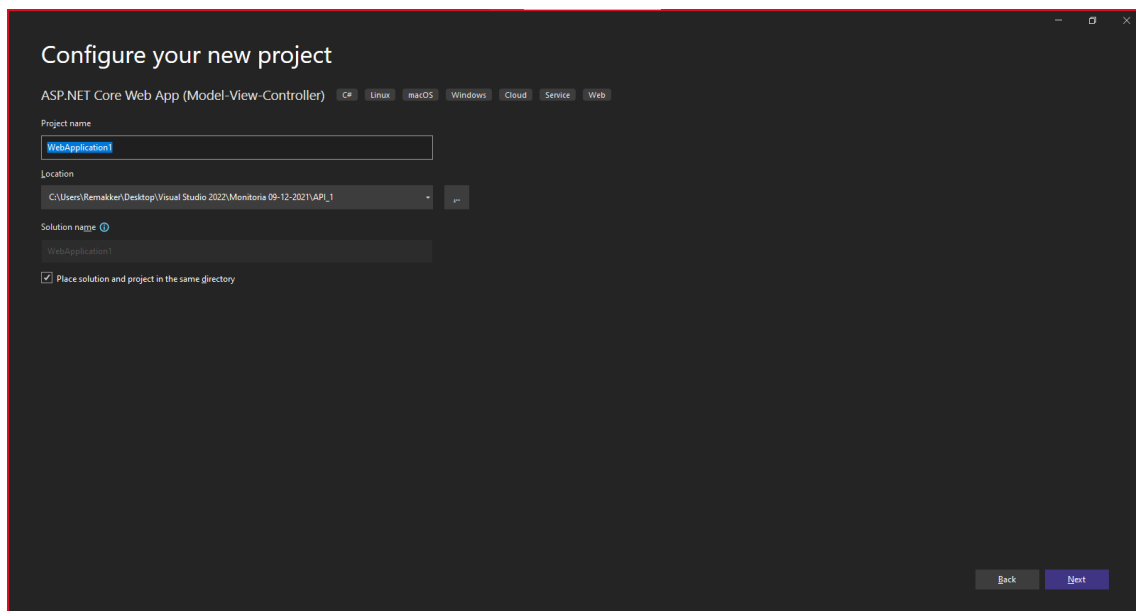


Neste caso Web.

8°Selecione o template “ASP.NET Core Web App(Model-View-Controller)”:



9°Escolha o nome desejado:



10°Ao escolher o diretório desejado, digitar o nome do projeto e clicar em next, a próxima pagina será:

Additional information

ASP.NET Core Web App (Model-View-Controller) C# Linux macOS Windows Cloud Service Web

Framework ⓘ
.NET 6.0 (Long-term support)

Authentication type ⓘ
None

☒ Configure for HTTPS ⓘ
☐ Enable Docker ⓘ

Docker OS ⓘ
Linux

Back Create

11°No tipo de autenticação, escolha:

Authentication type ⓘ

None

None
Individual Accounts
Microsoft identity platform
Windows

Windows.

12°Sua pagina deve estar assim:

Additional information

ASP.NET Core Web App (Model-View-Controller) C# Linux macOS Windows Cloud Service Web

Framework ⓘ
.NET 6.0 (Long-term support)

Authentication type ⓘ
Windows

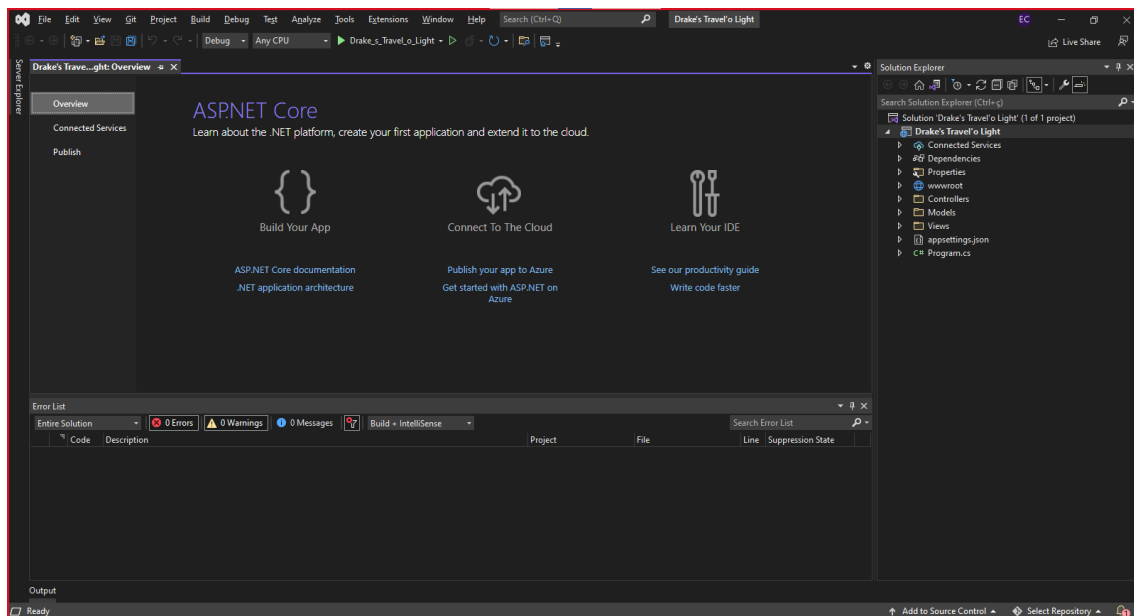
☒ Configure for HTTPS ⓘ
☐ Enable Docker ⓘ

Docker OS ⓘ
Linux

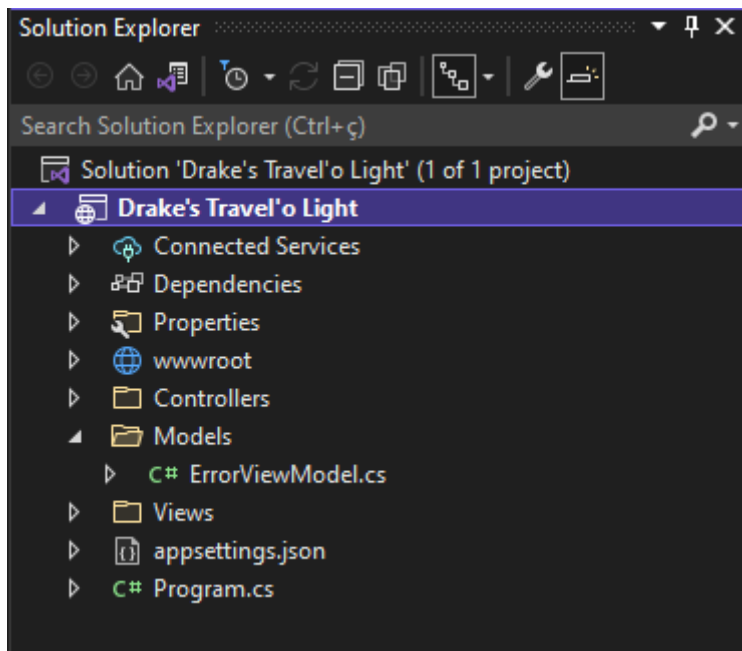
Back Create

E Clique em “Create”.

13°Ao criar o projeto, ele deve se parecer com isso:



14°Abra a pasta Model:

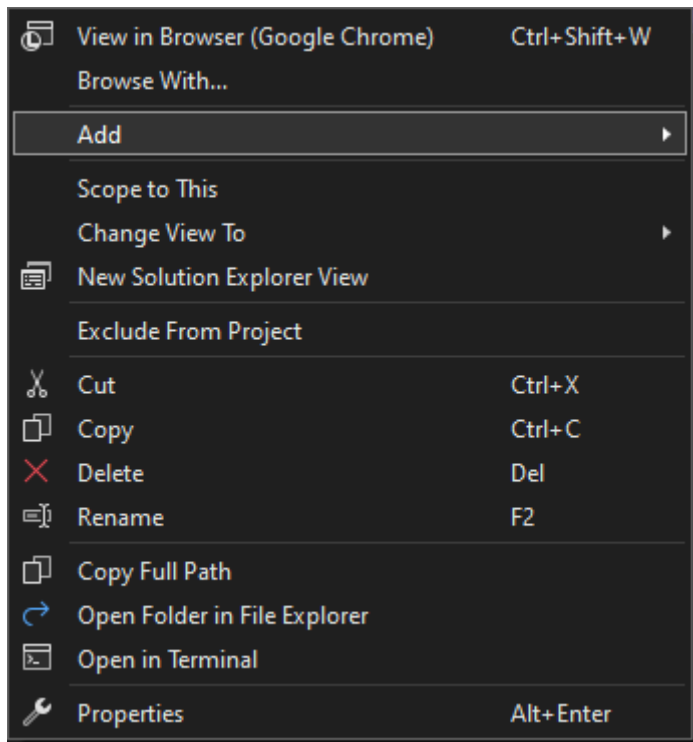


- O que devo colocar na “Models” e por onde começar meu projeto ?
Bom, em “Models” é em teoria onde vai ficar a sua tabela no db. O que quero dizer com isso é **[Exemplo: se você está criando uma Pizzaria, você tem que ter um formulário de cadastro de clientes, ou pelo menos um formulário de entrega, que seria mais ou menos assim:**
Formulario de Cadastro:
Nome: [Nome hipotético] Sheronline
Endereço:[Endereço hipotético] Rua dos limões
N° da Casa:[N°da casa hipotético] 456
Bairro: [Bairro hipotético] Bairro Camarão

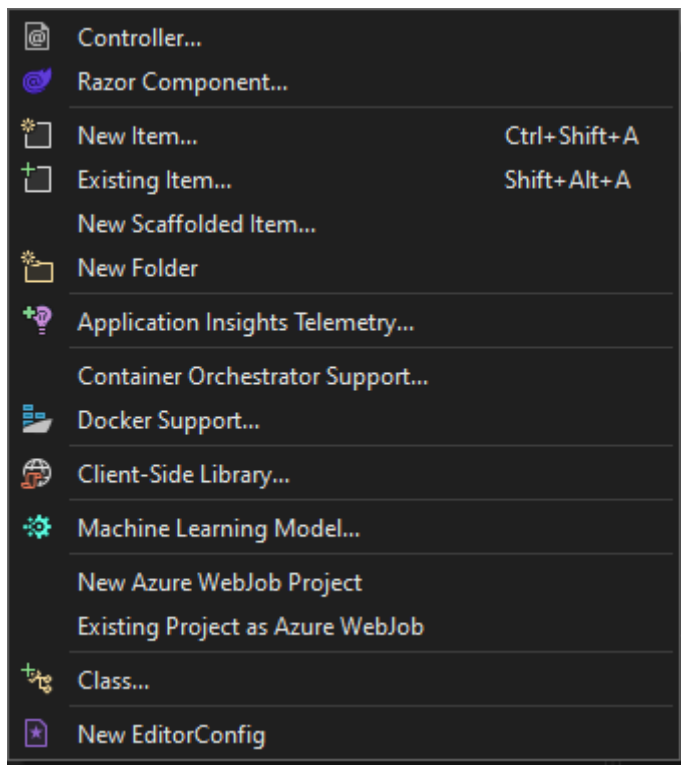
Show esse seria seu formulário, tendo isso em mente o que vc precisa, quais os dados que você quer pegar do usuário e guardar ? (Nome, Endereço, N° e Bairro) **Fim do Exemplo**].

Então isso ai é basicamente uma classe que você tem que fazer, ou seja, o seu modelo de negocio é Pessoa ou endereço, meio que não interessa o nome, mas um nome mais auto descritivo ajuda caso alguém vá ver esse modelo, e esse modelo vai virar uma tabela no banco de dados. Voltado ao nosso projeto.

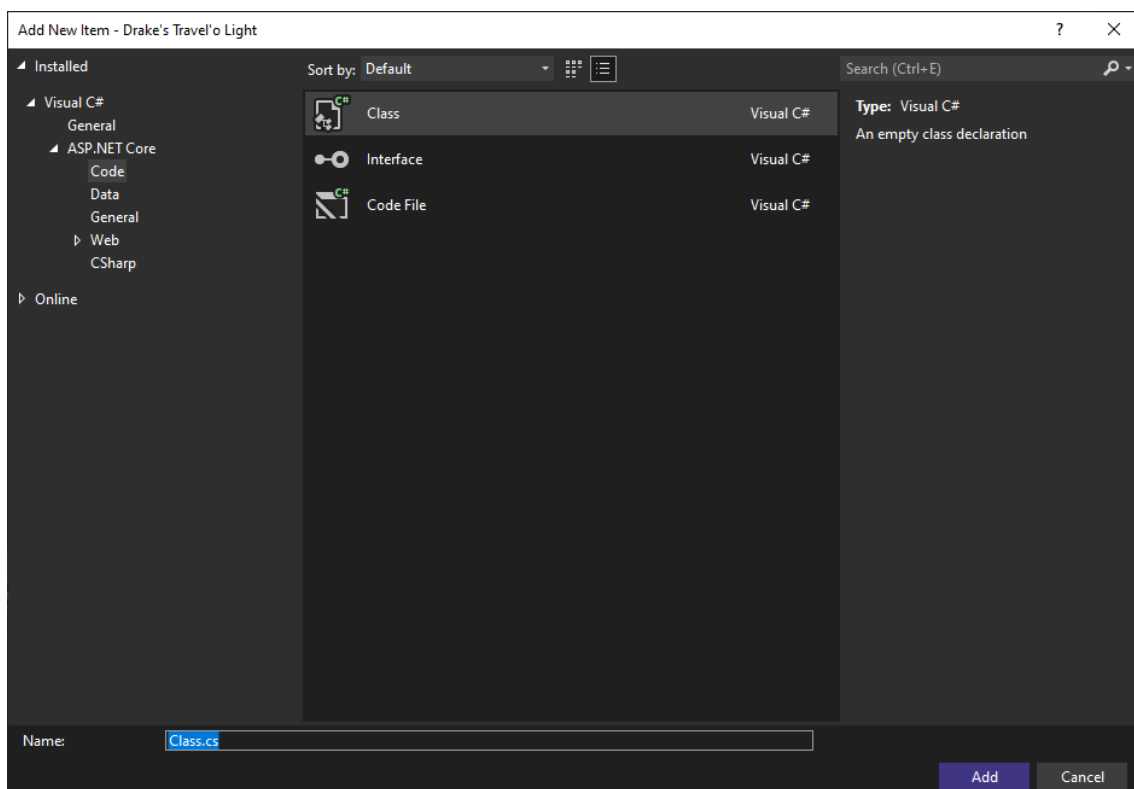
15°Clique com o botão direito do mouse ou touchpad, caso use notebook, na pasta "Models" e Clique em add:



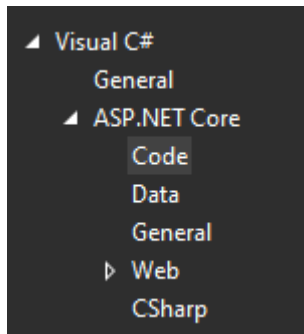
16°Em Add, clique em "Class":



17° Em Class, vai aparecer essa pagina:



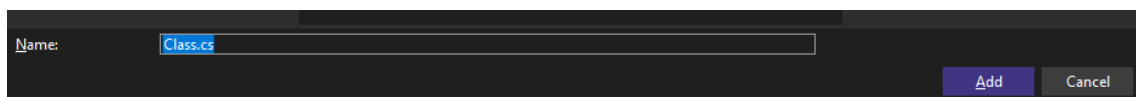
18° Seleção na parte da direita:



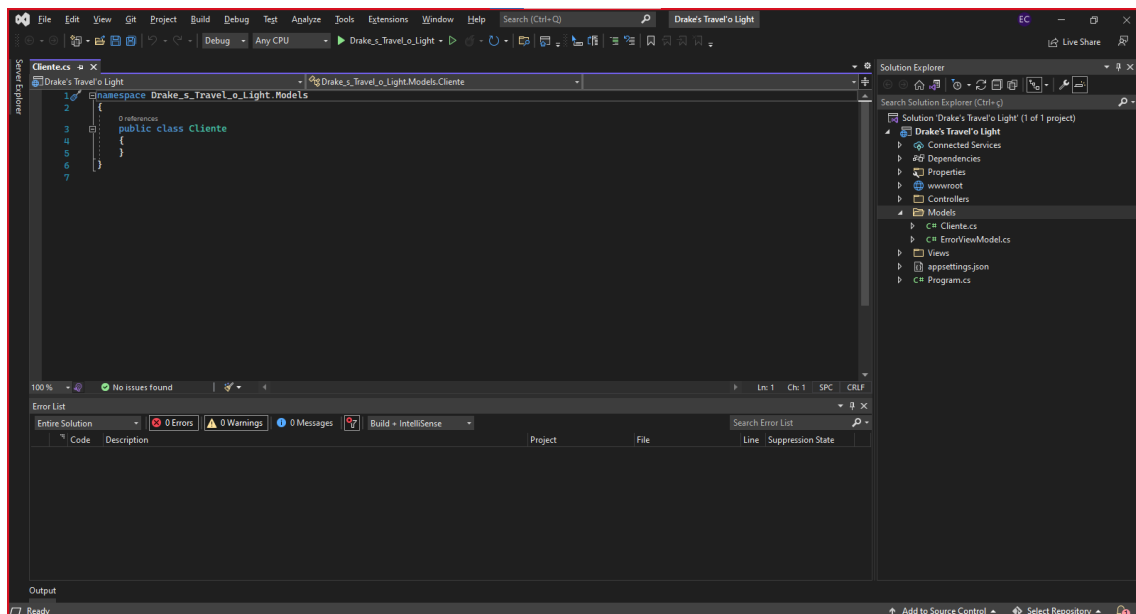
19°Selecione o template “Class”:



20°Na parte de baixo, selecione o Nome, lembrando que esse nome, será o nome de uma classe dentro da Pasta Models(Modelo), me arrisco dizer que você pode falar que seu projeto C#-MVC é baseado em {fala os nomes das classes que há dentro dessa pasta}, mas não leva isso a ferro e a fogo.



21°Ao selecionar o template e escolher o nome clique em add, e é suposto aparecer a sua classe na tela, mais ou menos assim:



22°Assim como toda classe, você deve adicionar os atributos e métodos dela:


```

0 references
public string nome { get;set; }
public string nome2;

0 references
public string getName()
{
    return nome2;
}

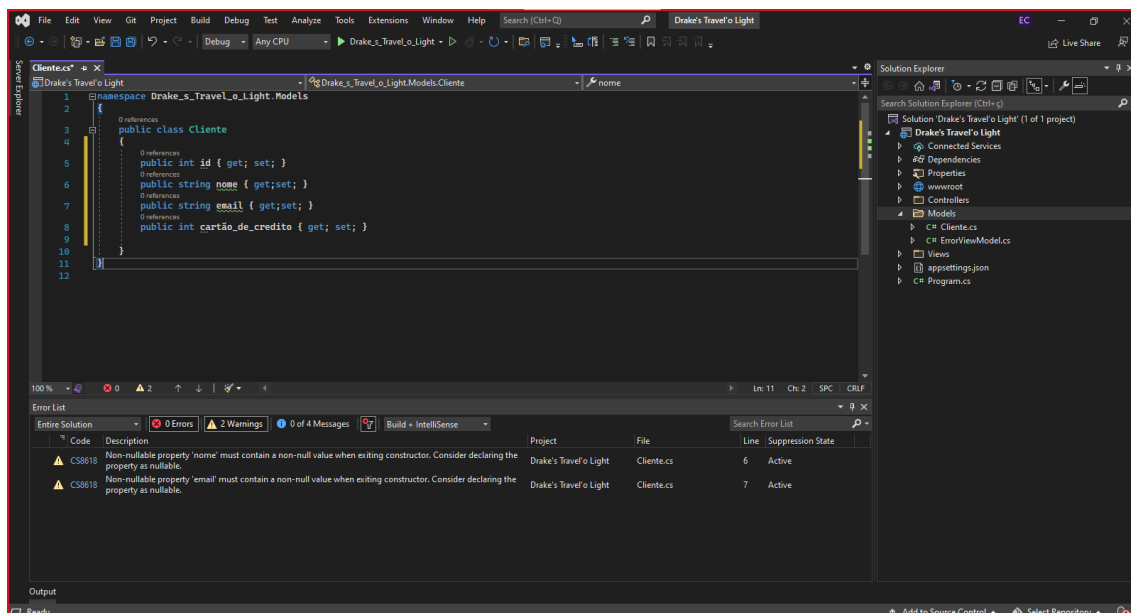
0 references
public void setName(string x) //poderia ser tabem string nome)
{
    nome2 = x;
    //this.nome2 = x;
}

```

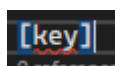
Isso aqui são os Getters and Setters, que em C# nós temos o shortcut(atalho) de usar a forma reduzida, que está na linha 1.

O que tem na linha 1 é a mesma cosia que os dois métodos abaixo, só que reduzido, o get; = getName() e o set; = setName(), ou seja, se eu apagar o método getName e setName, e usar na frente do nome2 {get; set;} ele vai fazer a mesma cosia.

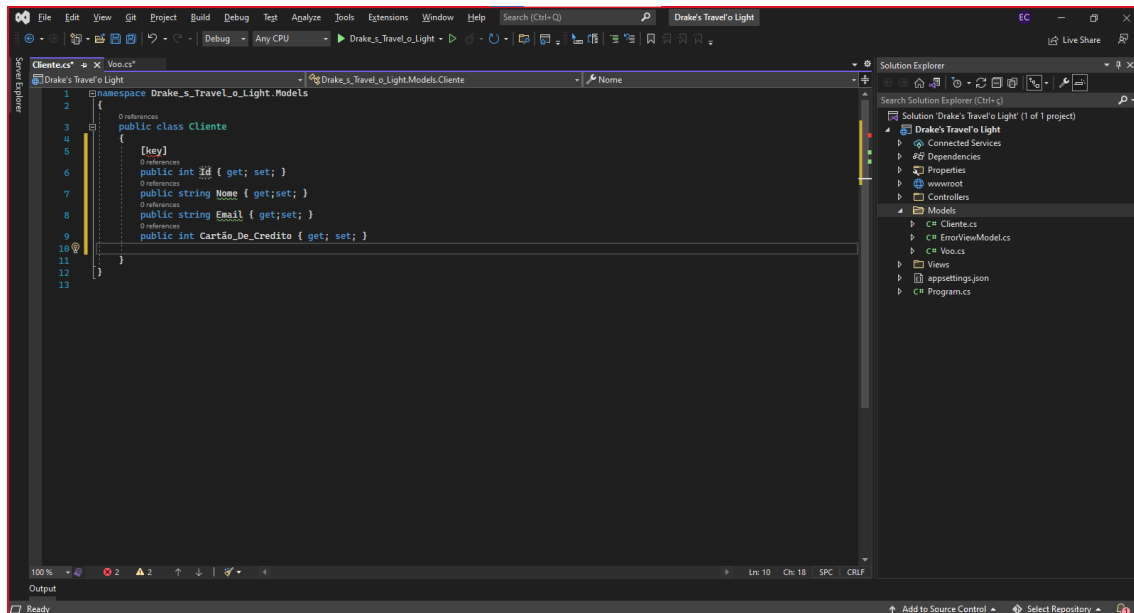
23°Com os atributos e métodos definidos, repita isso com todos os modelos e classes quequiser:



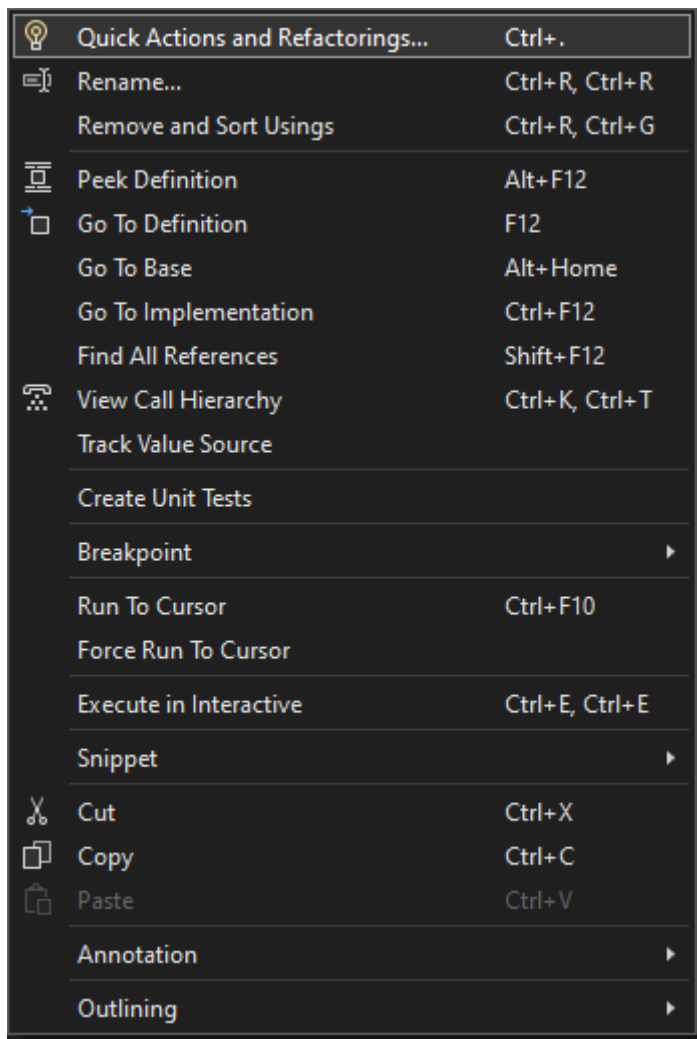
24°Como eu quero que o id de cliente seja a chave primaria no banco de dados, eu tenho que especificar aqui, pq essa classe ela pode ser compreendida como uma forma de bolo que vc coloca a massa e sai no formado, neste caso ela serve como modelo pra tabela, que irá ter id, nome, email, cartão de credito.



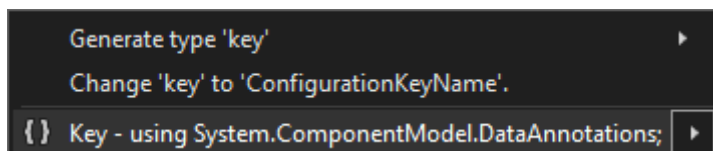
Eu adicionei a palavra key em cima de id pra especificar que quando for criada a tabela, o id seja criado como chave primaria.



25° Mas como podem ver está com erro, e para eliminar esse erro, clicamos com botão direito em cima:



“Quick Actions and Refactorings...”

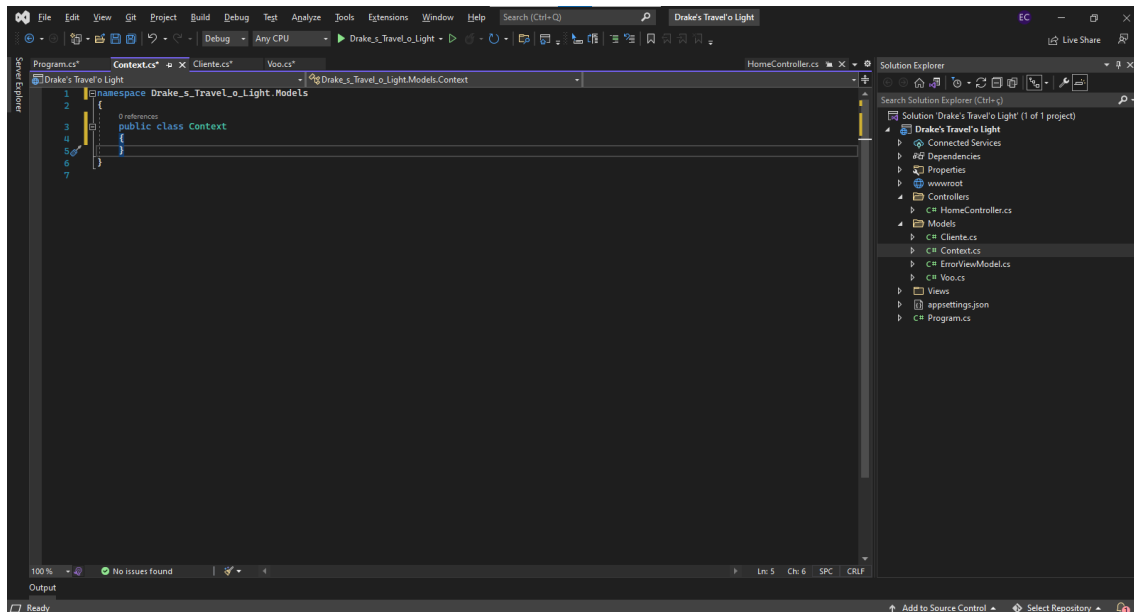


“Key – using System.ComponetModel.DataAnnotations;” Basicamente isso aqui faz a importação do Data Annotations, que não é bom ficar traduzindo isso, mas leva que é um modulo/ uma importação necessária pra usar [Key].

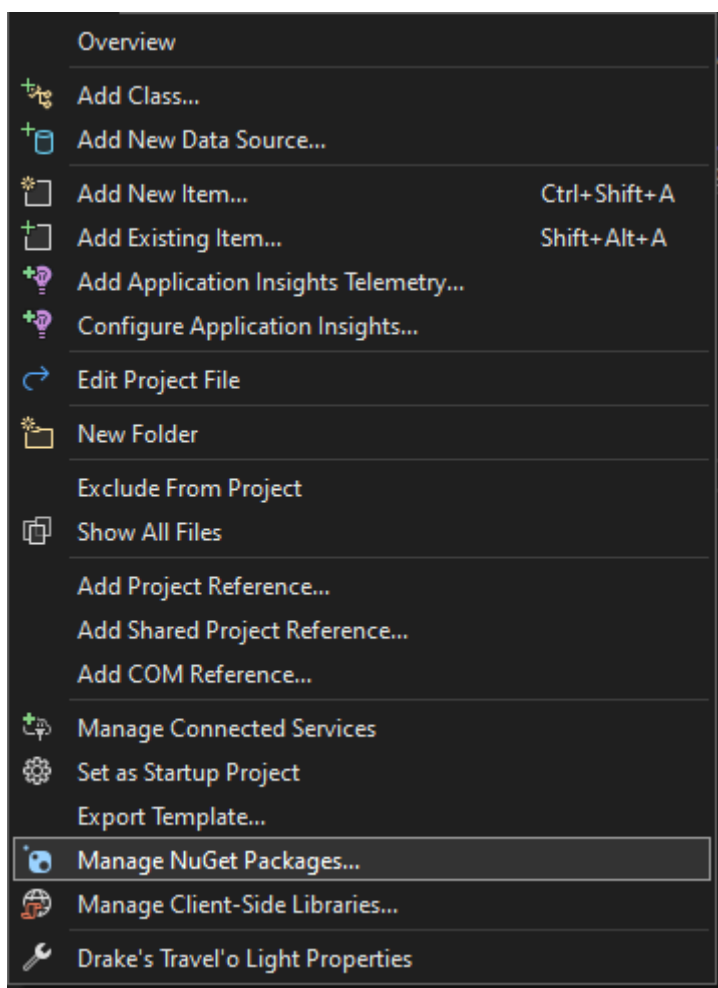
Ao clicar isso aparecerá no topo da classe:

```
using System.ComponentModel.DataAnnotations;
```

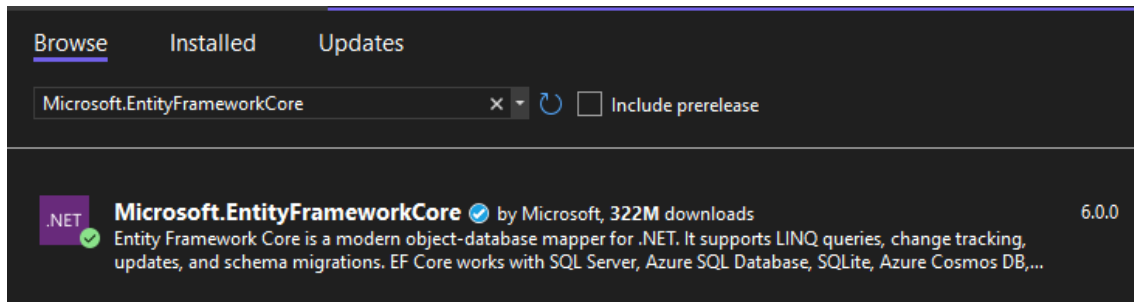
26°Ao terminar de adicionar os atributos e tipos de dados vá para a pasta e crie(Passo 15° - 20°) uma nova classe chamada Context:



27° Antes de prosseguir temos que instalar alguns pacotes, vá em “Project” depois em “Manage NuGet Packages”:

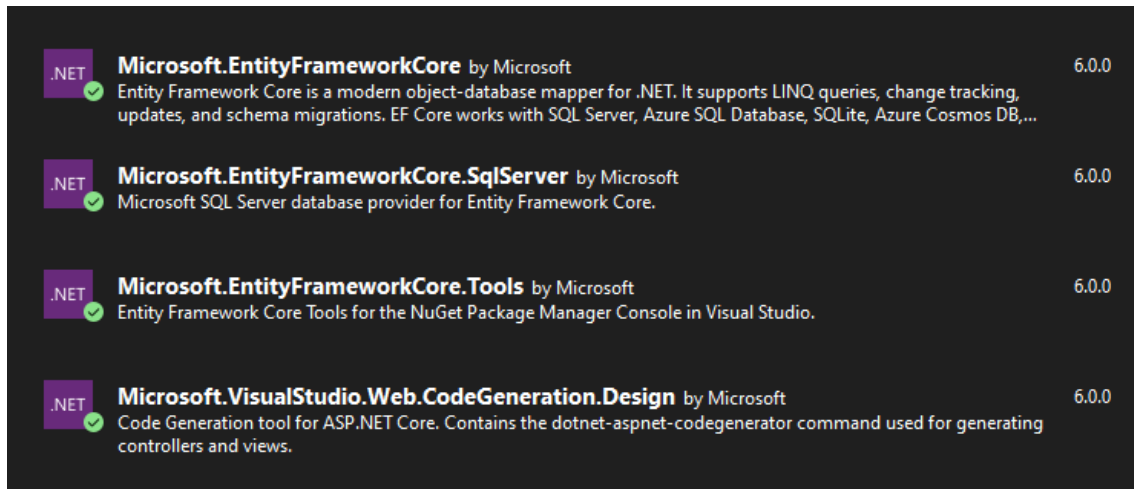


28°Vá em “Browse” e escreva Microsoft.EntityFrameworkCore:



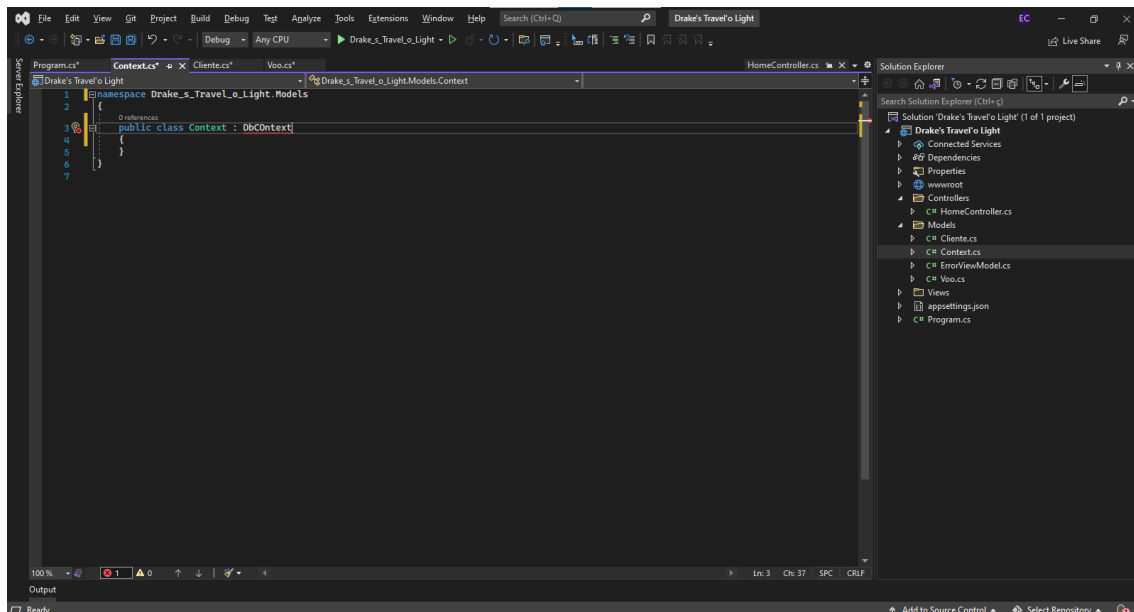
E Instale.

29°Você deverá ter esses pacotes:

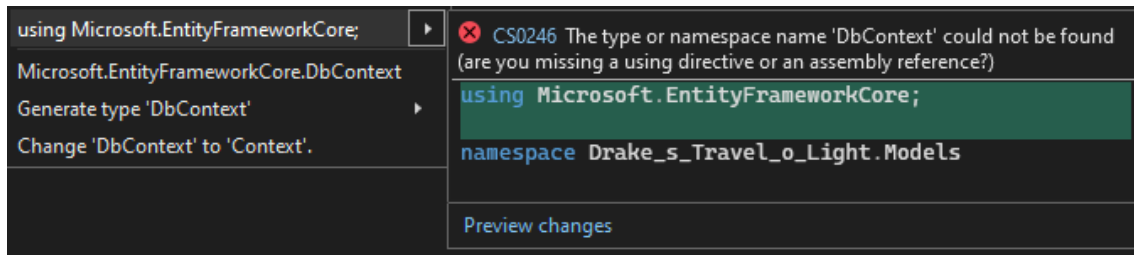


30°Pronto agora já podemos seguir na Classe Context.

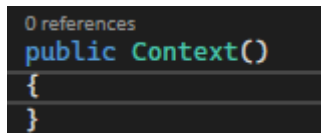
31°Herde os atributos e métodos da Classe “DbContext”:



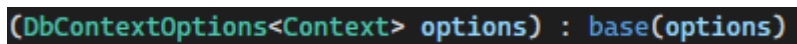
32°Vai estar dando erro, resolva importando o EntityFramework.Core:



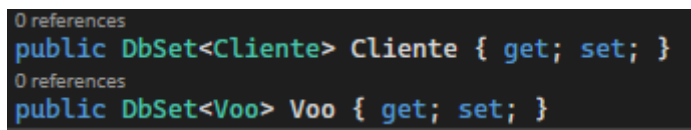
33°Crie o método Construtor, que lembrando tem o mesmo nome da classe:



34°Esse método construtor é criado quando instanciamos ela em outra classe, ou seja, ela ao iniciar vai ter que ter alguma coisa ao iniciar, mas esse 'construtor' não vai ficar sem parâmetros, e vamos passar eles:

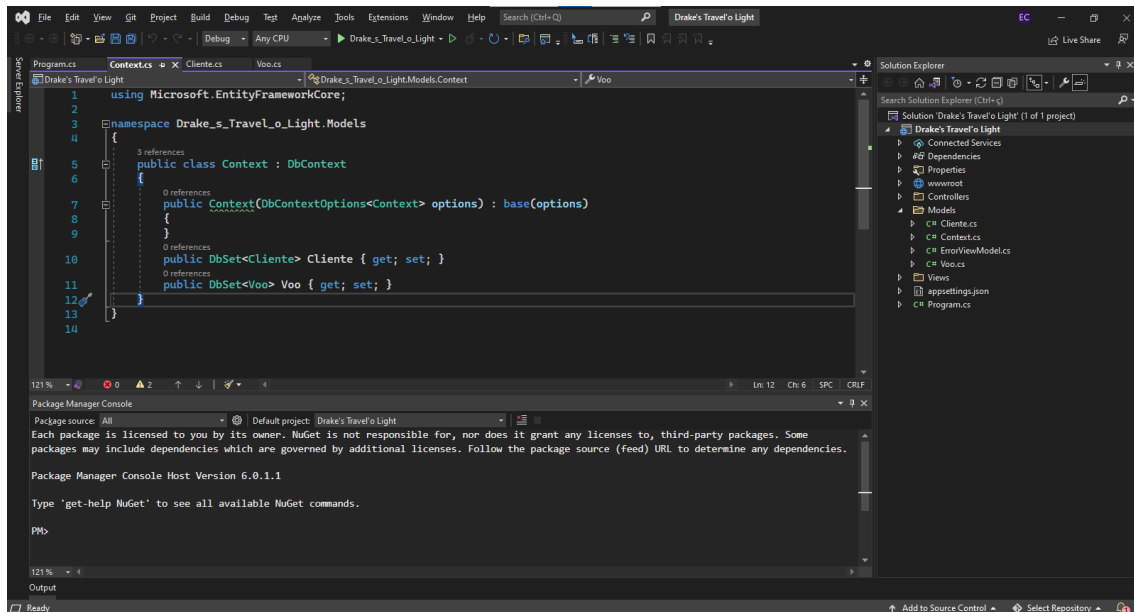


35°Logo abaixo vamos 'Settar' as tabelas que nós criamos dentro da pasta 'Models':



Basicamente aqui eu usei um get e set padrão, nós passamos a classe DbSet que mapeia as entidades, que no meu caso é Voo e Cliente, de forma bem simplória e até mesmo errônea, mas para o entendimento, o DbContext é uma espécie de instancia do Banco de dados na aplicação, mesmo não sendo, acredito que pensar dessa forma ajuda e entender essa classe.

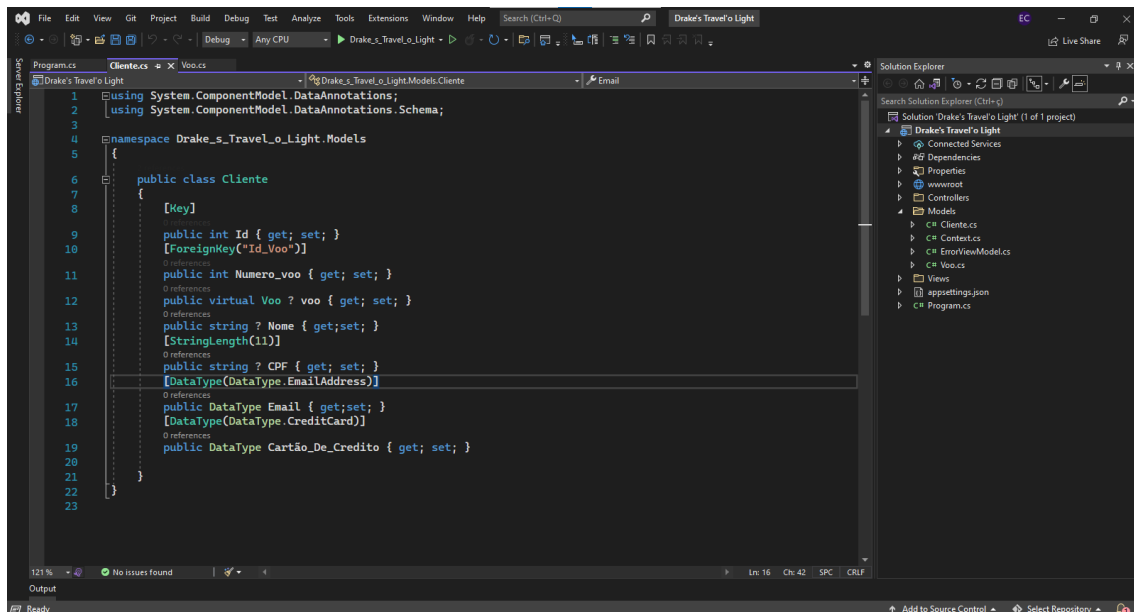
36°Sua pagina deverá estar mais ou menos assim:



Para mais informações sobre DbContext e DbSet acesse o link:

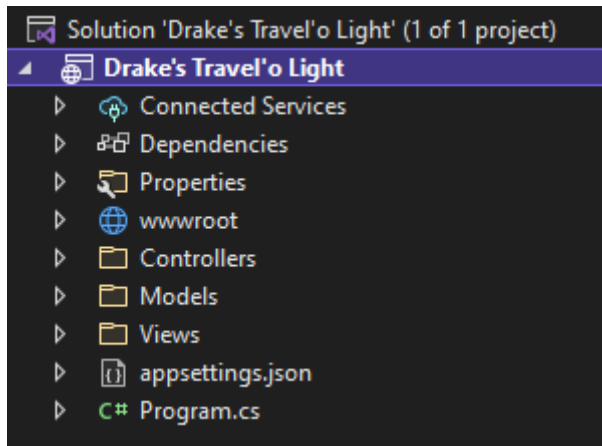
https://www.youtube.com/watch?v=CYzc1h36C3Q&ab_channel=JoseCarlosMacoratti

37° Lembrando que em alguma das suas classes, é recomendado ter uma *primary key*:

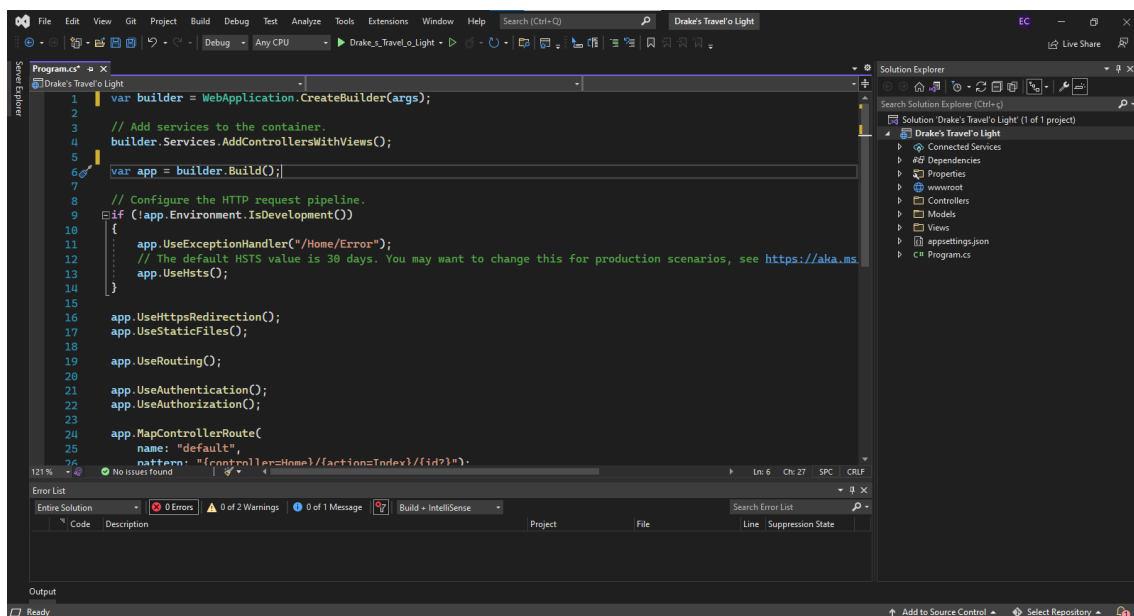


Segue a minha como referência.

38°Vá em 'program.cs' :



Sua página vai estar assim:



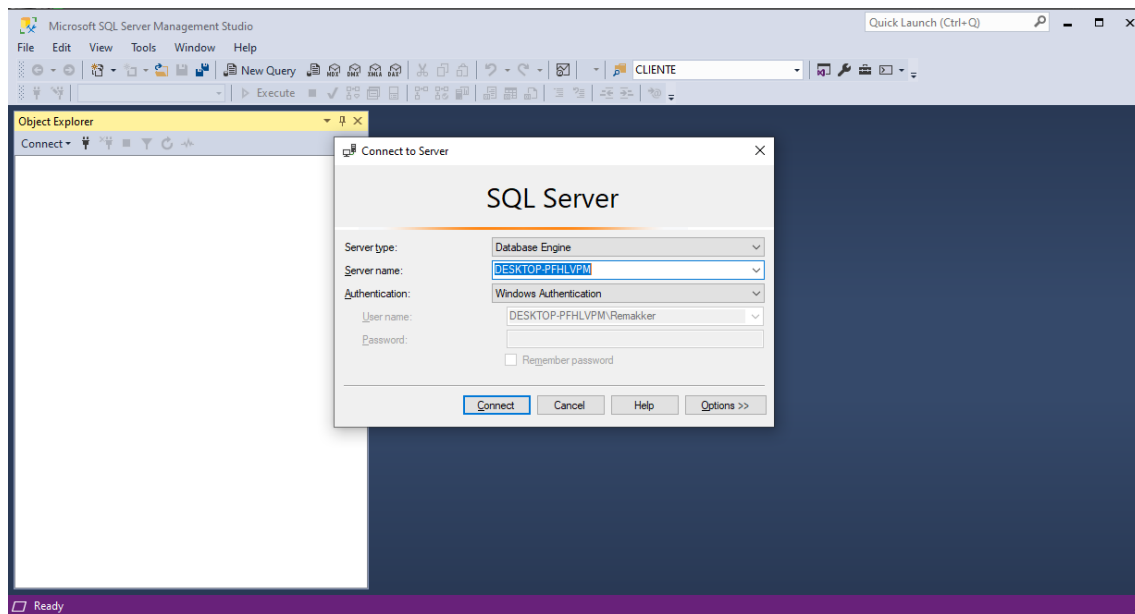
39°Na linha 5 digite:

```
builder.Services.AddDbContext<Context>(options => options.UseSqlServer(@""));
```

Basicamente, está passando os atributos e métodos do contexto usando uma expressão lambda para o db. Em poucas palavras é a conexão com o Servidor.

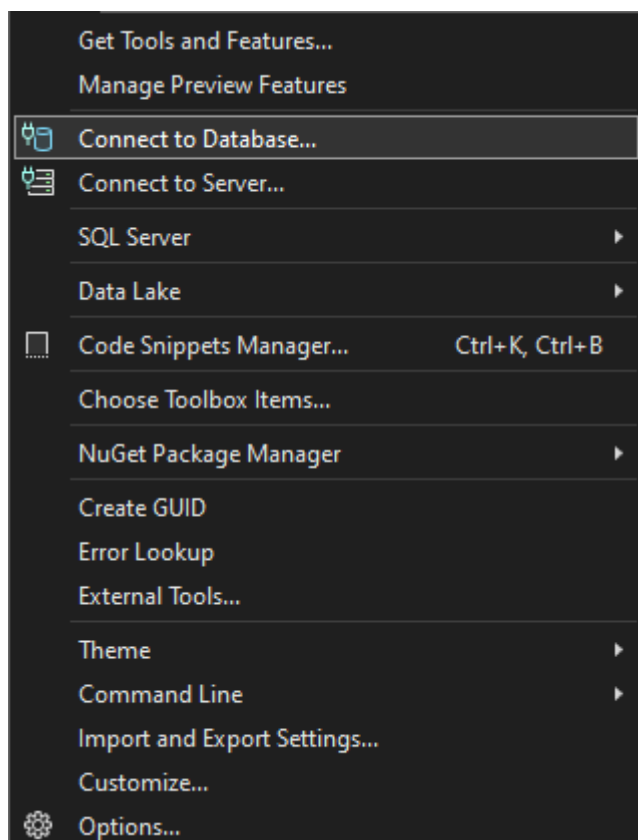
Mas como podemos ver ainda está com erro, vamos resolver isso, mas antes precisamos pegar o caminho até o servidor, ou como é chamado a 'String de conexão'.

38°Abra msSQL(Microsoft SQL Server Management Studio):

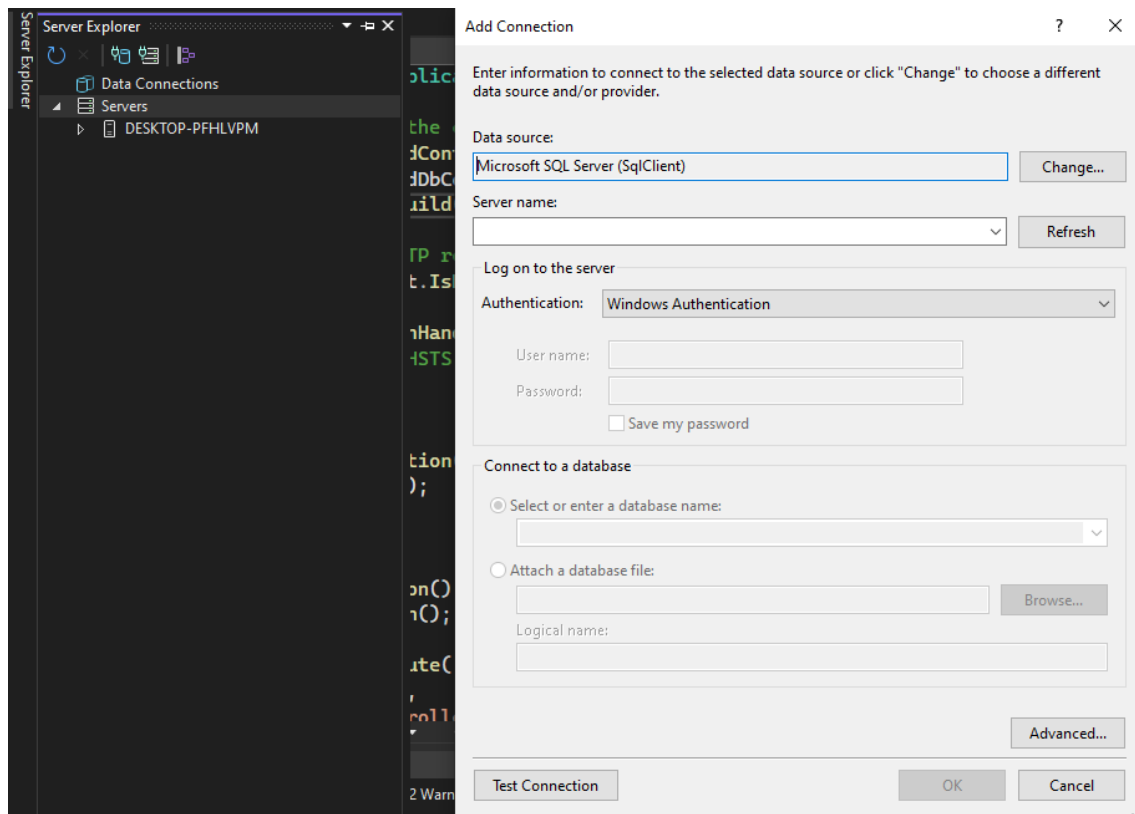


De cara nós já temos o nome do servidor que é o que precisamos para nós conectar à ele usando o Visual Studio 2022, **copie o nome do servidor**.

39°Volte ao VS 2022 e vá em 'Tools' > 'Connect to Database':



Irá abrir essas duas janelas.



E em '*Server name*' você cola o nome do Servidor que apareceu na tela inicial de msSQL e vai selecionar qualquer database, o que nós precisamos aqui é somente a string de conexão:

Add Connection ? X

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft SQL Server (SqlClient) Change...

Server name:
DESKTOP-PFHLVPM Refresh

Log on to the server

Authentication: Windows Authentication

User name:

Password:

☐ Save my password

Connect to a database

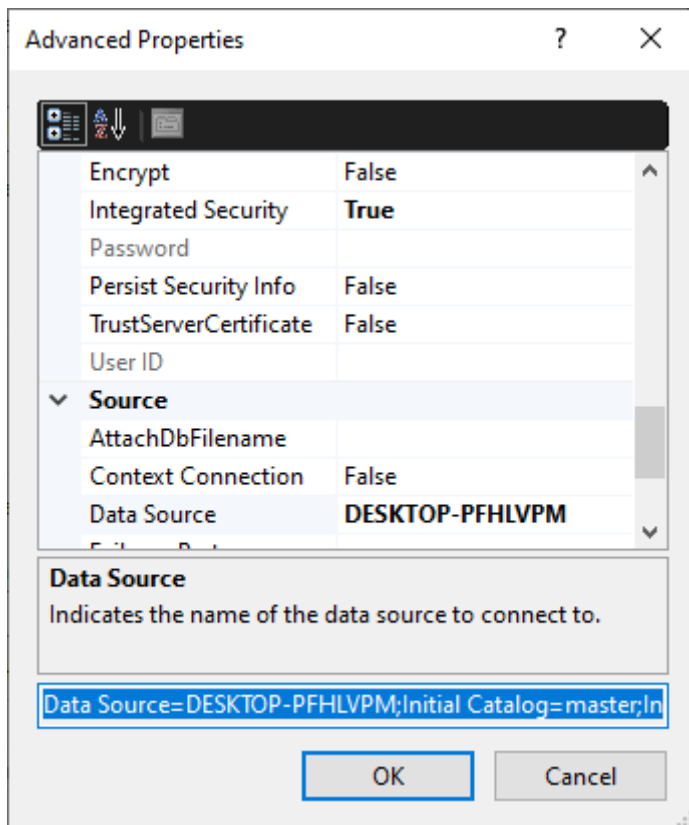
☒ Select or enter a database name:
master

☐ Attach a database file:
 Browse...
Logical name:

Advanced...

Test Connection OK Cancel

40° Após inserir o '*Server name*' e selecionar qualquer '*database*', clique em '*Advanced*' e copie a string de conexão:

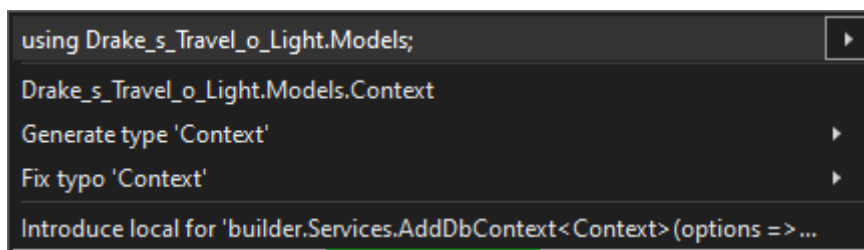


Ao copiar, clique em 'OK' depois em 'Cancel', voltando ao 'program.cs', no final da linha 5 dentro das "" do final coloque a string de conexão:

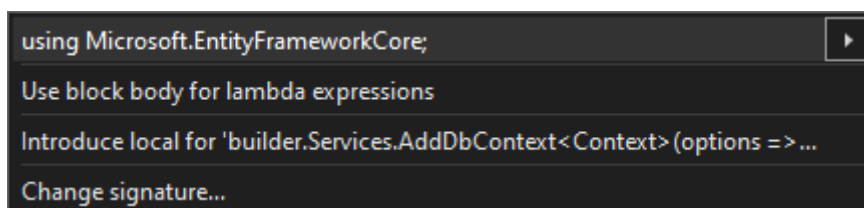
```
(@"Data Source=DESKTOP-PFHLVPM;Initial Catalog=master;Integrated Security=True"
```

41°Na frente de 'Inicia Catalog=' apague o que estiver escrito até o ; e digite o nome do servidor que você vai querer criar para a sua aplicação MVC.

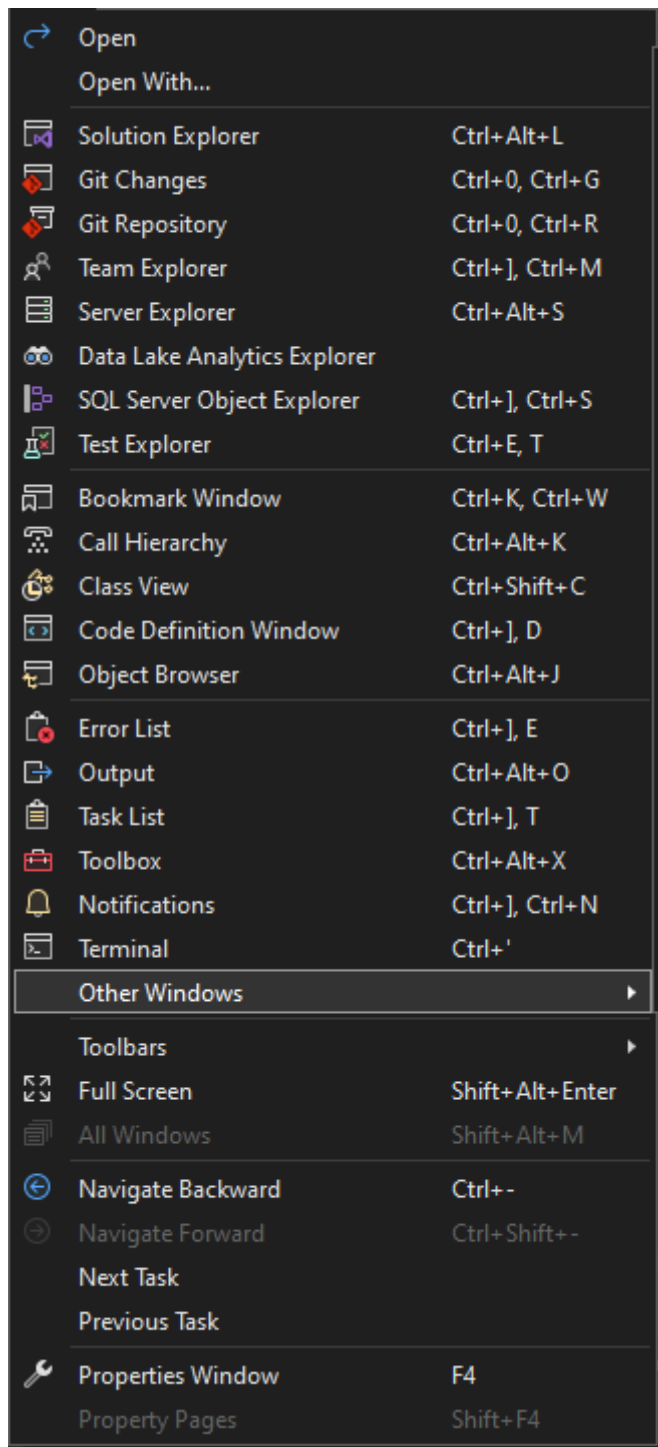
42°Mas a linha 5 ainda tem erros, agora clique com o botão direito em 'Context' e vá em 'Using + [Nome do seu projeto].Models;':

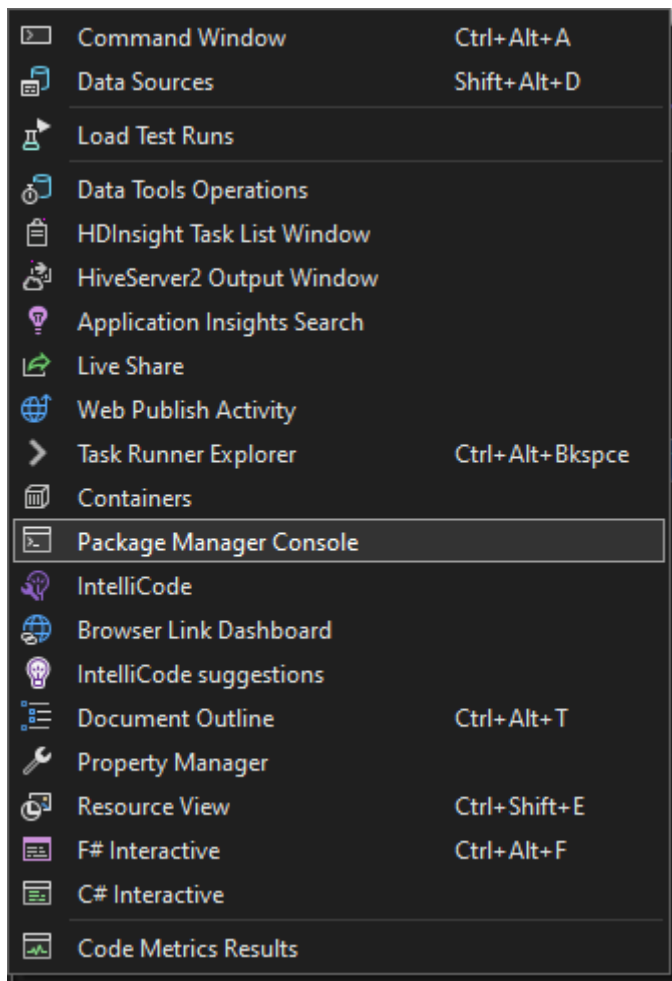


43°Agora vá em 'UseSqlServer' e faça o mesmo, mas dessa vez ira clicar em 'Using Microsoft.EntityFrameworkCore.Core;':

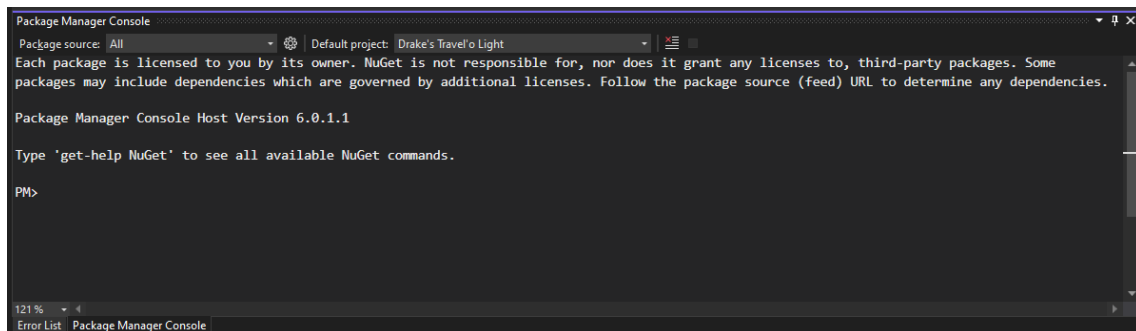


43° Agora vá em 'View > Other Windows > Package Manager Console':

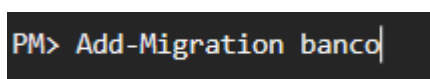




44°Na parte de baixo do VS irá abrir isso:

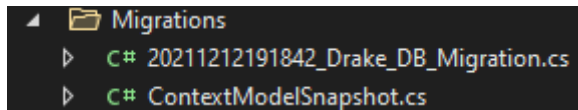
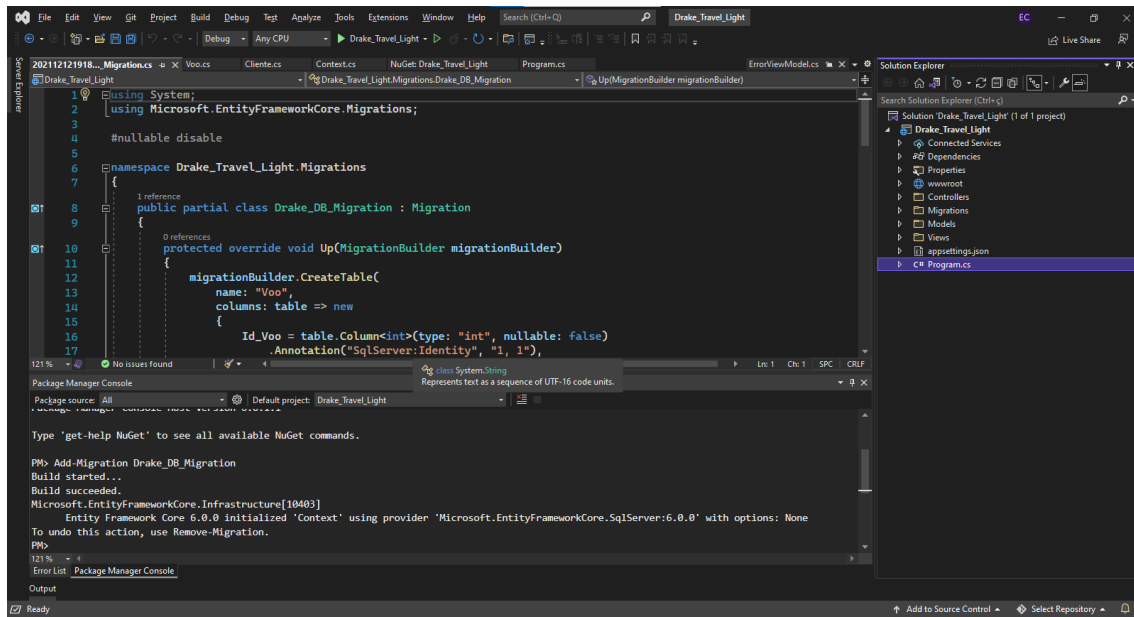


45°Digite 'Add-Migration + [Nome da sua migração]' não coloque o '+' pelo amor de deus, em geral vc está passando os parâmetros da sua classe que está na pasta Models para o Db, e nessa migração ele irá criar o banco de dados, com os atributos que você passou, ou seja, se você criou uma classe cliente assim '*public string nome*', vai ter um campo na sua tabela cliente com um varchar, pq no banco de dados não existe o tipo 'String' necessariamente com esse nome, lá é VARCHAR e em C# é string.

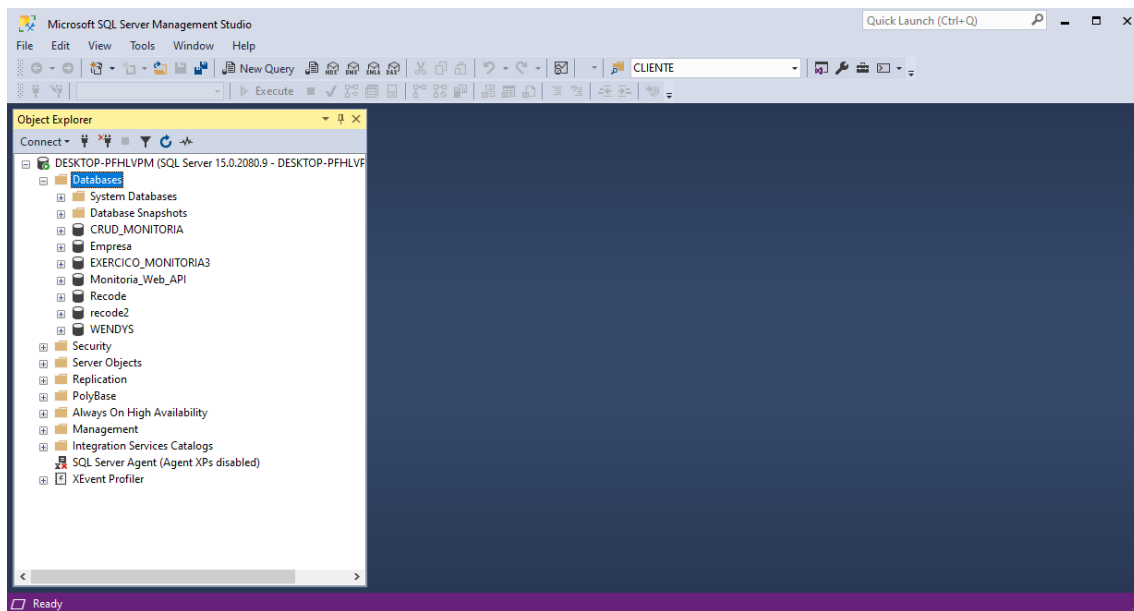


46°Execute.

47° Quando terminar, vai ter aparecido uma pasta com a sua migração:



48° Se você abrir o msSQL, vai ver que ainda não tem nada lá:



49° A tabela só foi criada na aplicação, mas como já temos o caminho até o db podemos dar um 'update-database':

```

Package Manager Console
Package source: All | Default project: Drake_Travel_Light
PM> Update-database
Build started...
Build succeeded.
Microsoft.EntityFrameworkCore.Infrastructure[10403]
    Entity Framework Core 6.0.0 initialized 'Context' using provider 'Microsoft.EntityFrameworkCore.SqlServer:6.0.0' with options: None
Microsoft.EntityFrameworkCore.Database.Command[20101]
    Executed DbCommand (474ms) [Parameters=[], CommandType='Text', CommandTimeout='60']
    CREATE DATABASE [Drake_Travel_DB];
Microsoft.EntityFrameworkCore.Database.Command[20101]
    Executed DbCommand (188ms) [Parameters=[], CommandType='Text', CommandTimeout='60']
    IF SERVERPROPERTY('EngineEdition') <> 5
    BEGIN
        ALTER DATABASE [Drake_Travel_DB] SET READ_COMMITTED_SNAPSHOT ON;
    END;
Microsoft.EntityFrameworkCore.Database.Command[20101]
    Executed DbCommand (9ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
    SELECT 1
Microsoft.EntityFrameworkCore.Database.Command[20101]
    Executed DbCommand (15ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
    CREATE TABLE [__EFMigrationsHistory] (
        [MigrationId] nvarchar(150) NOT NULL,
        [ProductVersion] nvarchar(32) NOT NULL,
        CONSTRAINT [PK__EFMigrationsHistory] PRIMARY KEY ([MigrationId])
    );
Microsoft.EntityFrameworkCore.Database.Command[20101]
    Executed DbCommand (0ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
    SELECT 1

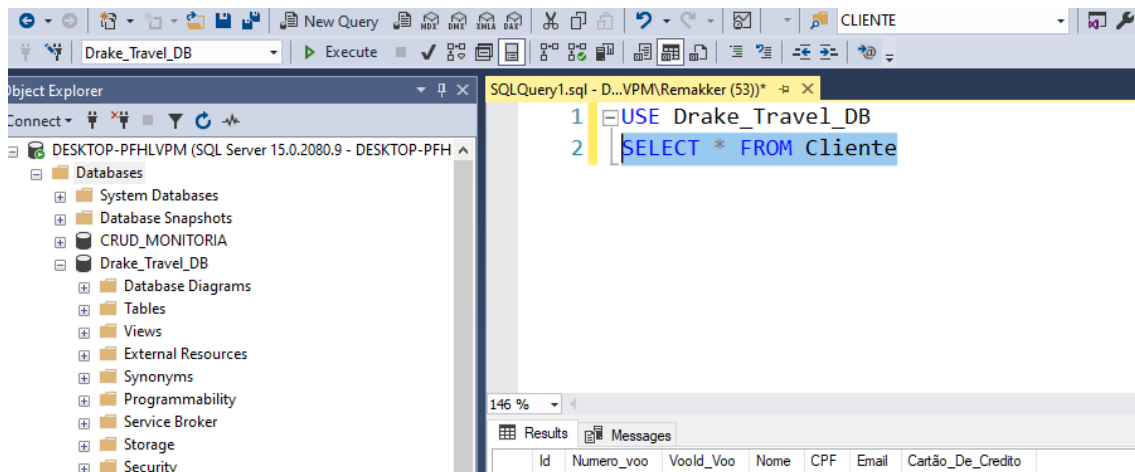
```

```

Package Manager Console
Package source: All | Default project: Drake_Travel_Light
    [Destino] nvarchar(max) NULL,
    [Assento] nvarchar(max) NULL,
    [Data_ida] datetime2 NOT NULL,
    [Data_volta] datetime2 NOT NULL,
    CONSTRAINT [PK_Voo] PRIMARY KEY ([Id_Voo])
    );
Microsoft.EntityFrameworkCore.Database.Command[20101]
    Executed DbCommand (6ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
    CREATE TABLE [Cliente] (
        [Id] int NOT NULL IDENTITY,
        [Numero_voo] int NOT NULL,
        [VooId_Voo] int NULL,
        [Nome] nvarchar(max) NULL,
        [CPF] nvarchar(11) NULL,
        [Email] int NOT NULL,
        [Cartão_De_Credito] int NOT NULL,
        CONSTRAINT [PK_Cliente] PRIMARY KEY ([Id]),
        CONSTRAINT [FK_Cliente_Voo_VooId_Voo] FOREIGN KEY ([VooId_Voo]) REFERENCES [Voo] ([Id_Voo])
    );
Microsoft.EntityFrameworkCore.Database.Command[20101]
    Executed DbCommand (4ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
    CREATE INDEX [IX_Cliente_VooId_Voo] ON [Cliente] ([VooId_Voo]);
Microsoft.EntityFrameworkCore.Database.Command[20101]
    Executed DbCommand (4ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
    INSERT INTO [__EFMigrationsHistory] ([MigrationId], [ProductVersion])
    VALUES (N'20211212191842_Drake_DB_Migration', N'6.0.0');
Done.

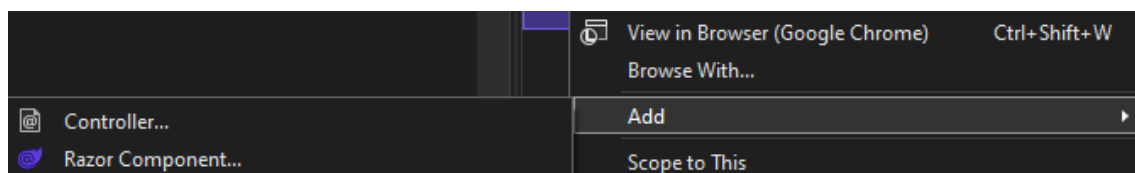
```

50°Quando terminar, vá no Server e confira se a tabela foi criada lá, na duvida dá um Select, pra aparecer tudo:

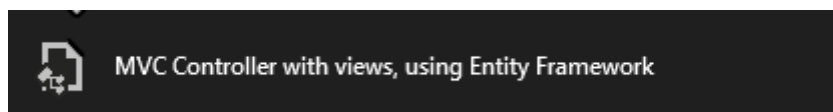


Como podem ver a tabela foi Criada, mas ainda falta criar os Controles.

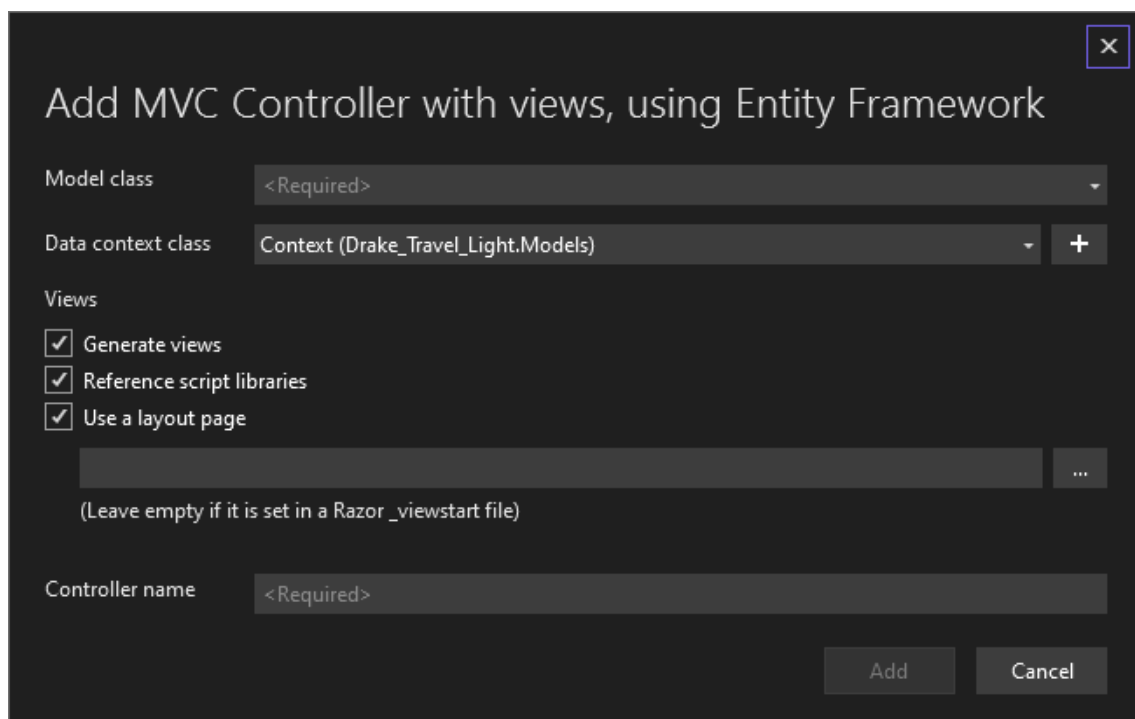
51°Vá em 'Controllers' clique com o botão direito "Add > Controllers":



52°Escolha esta opção e clique em Add:

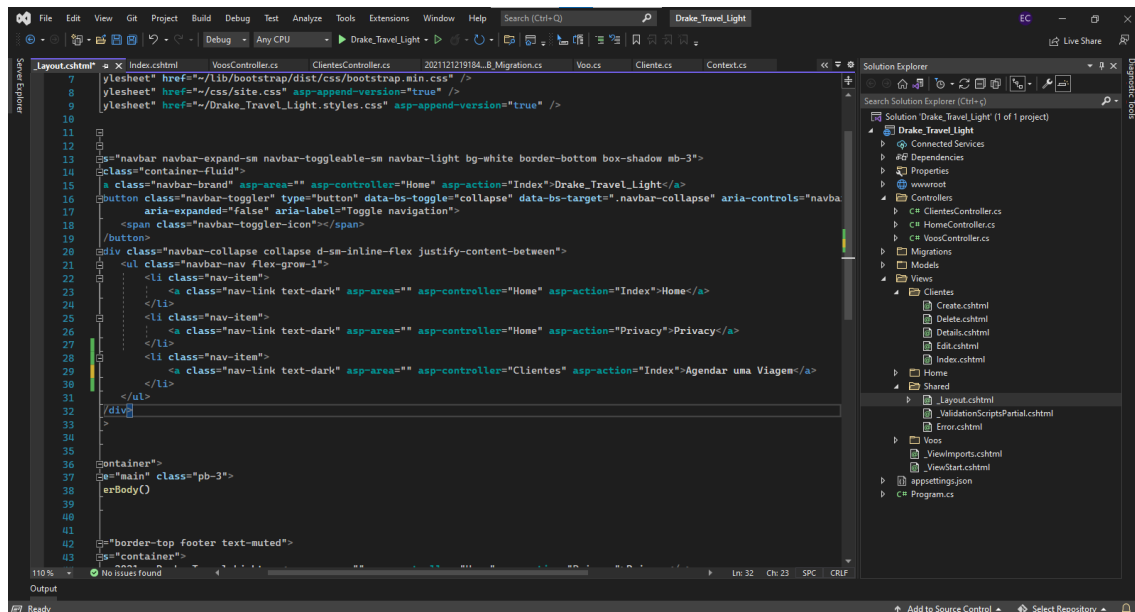


53°Escolha a classe que você quer fazer o CRUD e a classe contexto, se quiser mudar o nome, mude, mas eu não recomendo:



E clique em add.

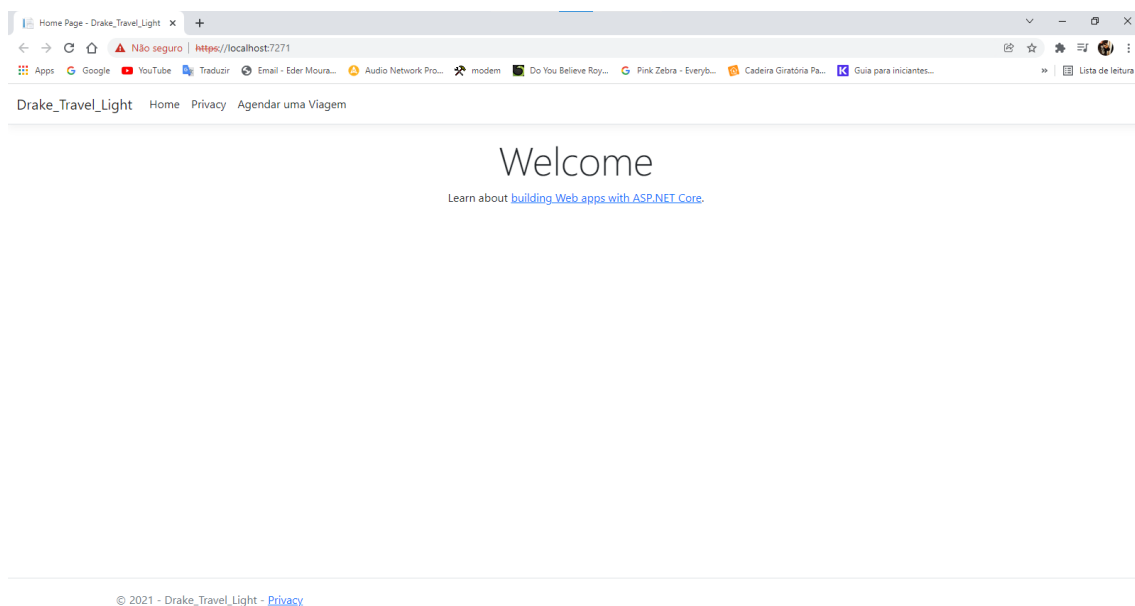
54°Vá em “Views > Shared > Layout” Copie da linha 25 – 27 e cole logo abaixo da 27, ou seja na linha 28.

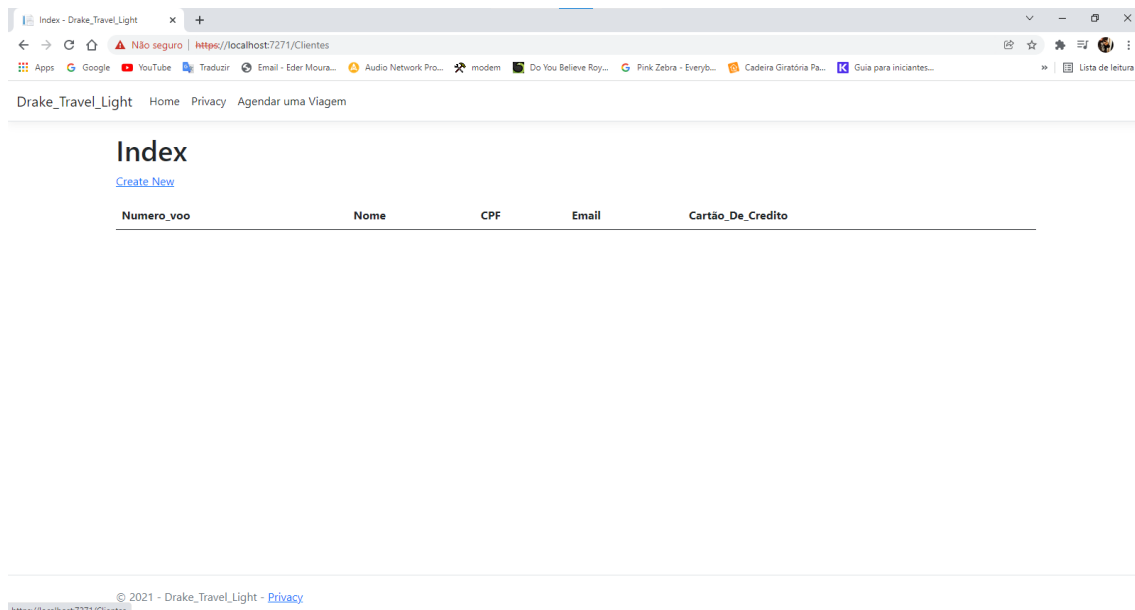


```
7 <!-->
8 <!-->
9 <!-->
10
11
12
13 <div class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">
14 <div class="container-fluid">
15 <div class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">Drake_Travel_Light</div>
16 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse" aria-controls="navbar-collapse"
17 <span class="navbar-toggler-icon"></span>
18 </button>
19 <div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
20 <ul class="navbar-nav flex-grow-1">
21 <li class="nav-item">
22 <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>
23 </li>
24 <li class="nav-item">
25 <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>
26 </li>
27 <li class="nav-item">
28 <a class="nav-link text-dark" asp-area="" asp-controller="Clientes" asp-action="Index">Agendar uma Viagem</a>
29 </li>
30 </ul>
31 </div>
32 </div>
33
34
35
36 <div class="main">
37 <div class="main">
38 <div class="main">
39 <div class="main">
40 <div class="main">
41 <div class="main">
42 <div class="main">
43 <div class="main">
```

Em “asp-controllers” coloque o nome do controle que vc criou, e em “asp-action” vc coloca o nome da pagina que você quer que o usuário vá quando clicar no botão, para achar as paginas que os controles fazem referencia vá em ‘View’ e lá vai estar com os nomes das classes e dentro delas vai ter as paginas, basta fazer a ligação com a tag ancora <a> no html.

55°Execute pra ter certeza que tudo está funcionando corretamente.





E seja feliz.

Fim.

Considerações finais

Bom eu fiz o melhor que eu consegui, espero que esse paper de “só” 27 páginas te ajude, se você seguiu isso tudo e não consegui, me desculpe. Eu não sigo com o front por que, bom esse é o seu trabalho, faça mais páginas, deixe ele bonito e tire um 10!.

Bom, até mais, se vemos na próxima!.

--Eder Moura (12/12/2021)