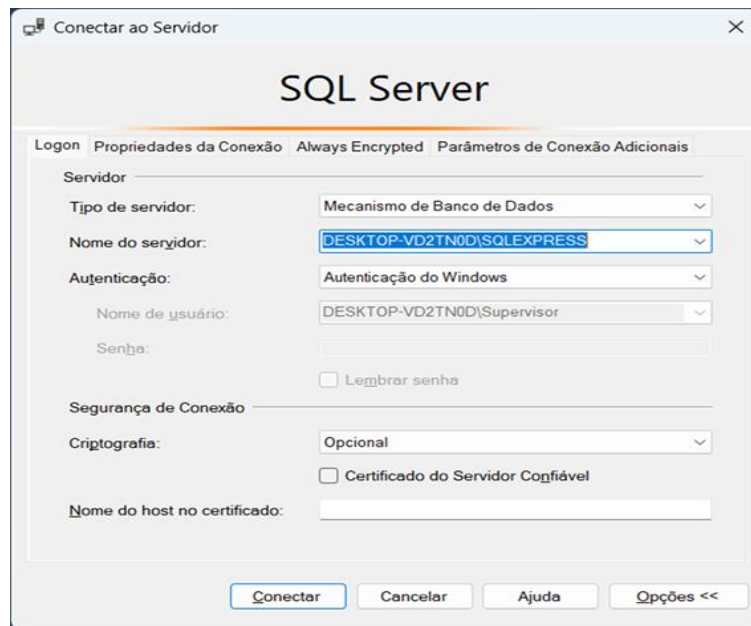
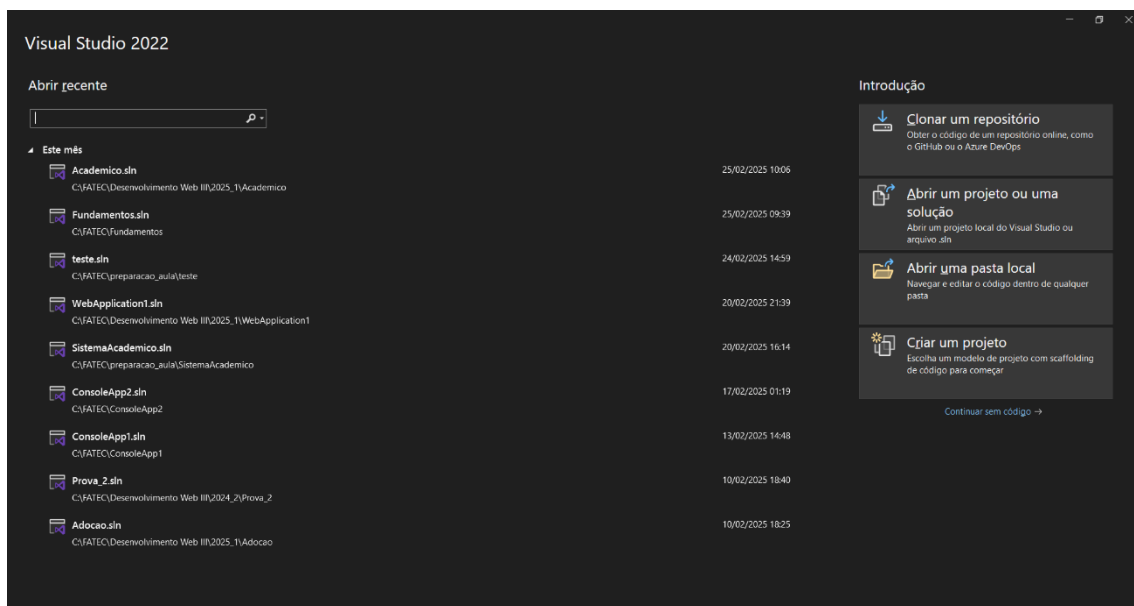


Criando um projeto Asp.Net Core MVC

1. Instalar a parte (caso ainda não tenha instalado) para usar SQLServer
 - a. Microsoft SQLServer Express em <https://www.microsoft.com/pt-br/sql-server/sql-server-downloads>
 - b. SQL Server Management Studio (SSMS) em <https://learn.microsoft.com/pt-br/ssms/download-sql-server-management-studio-ssms>
2. Instalar a parte (caso ainda não tenha instalado) para usar MySQL
 - a. Xampp obter em https://www.apachefriends.org/pt_br/index.html
3. Executar SSMS (Apenas para SQLSERVER)
 - a. Copiar o nome do servidor;
 - b. Abrir a conexão.



4. Abrir Visual Studio



5. Criar um projeto Aplicativo Web do ASP.Net Core(Model-View-Controller) C#

c. Microsoft.EntityFrameworkCore.SqlServer 9.02

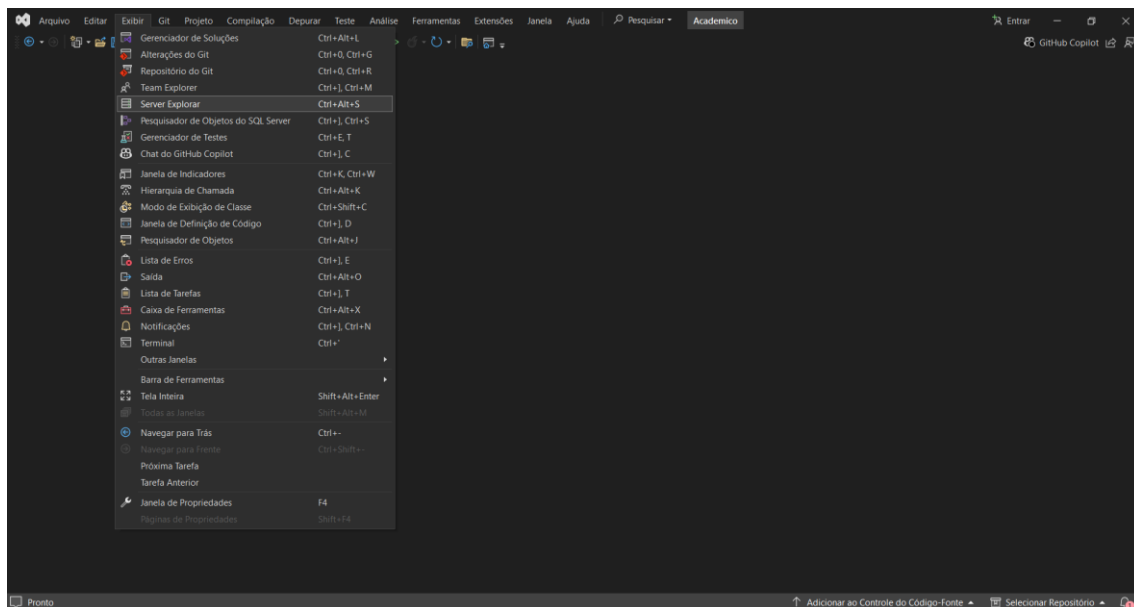
Para o .net 8 e MySql

d. Microsoft.EntityFrameworkCore 8.013

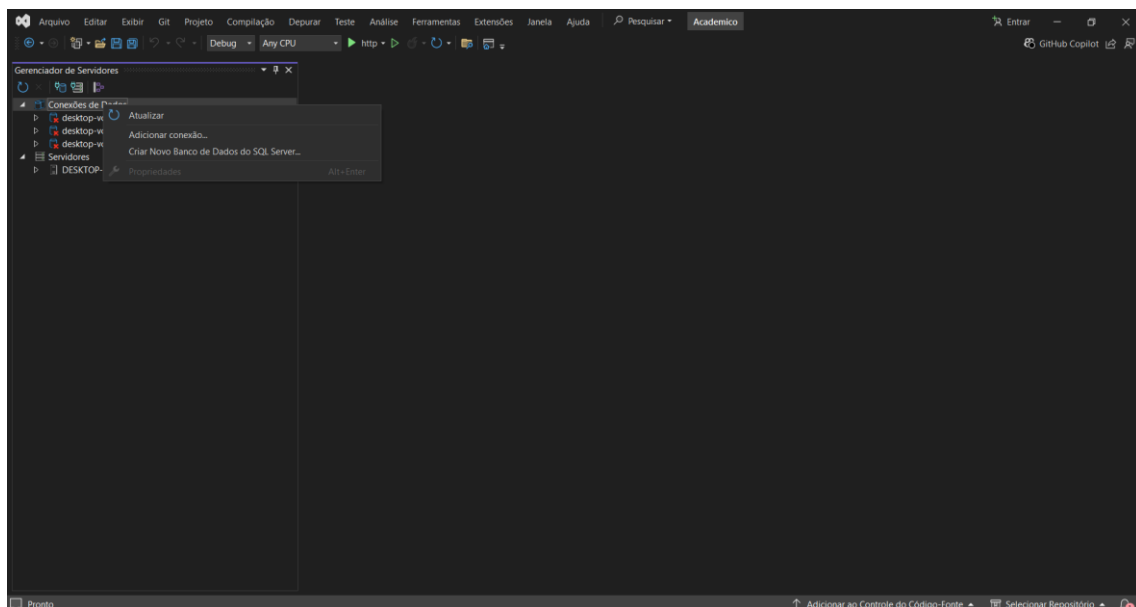
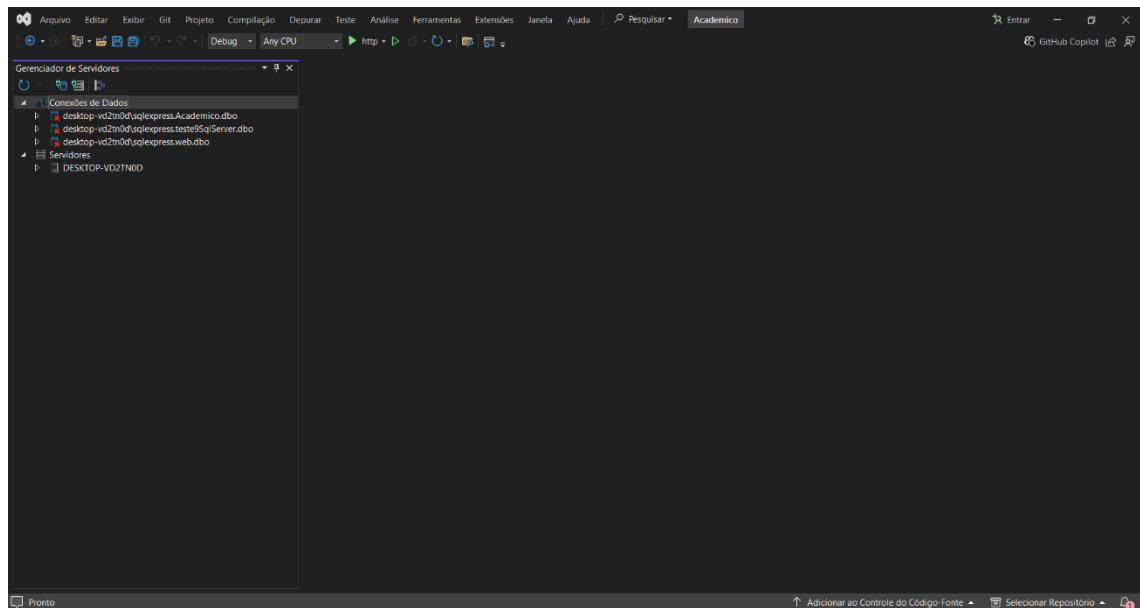
e. Microsoft.EntityFrameworkCore.tools 8.013

f. Pomelo.entityframeworkcore.mysql 8.0.3

7. Clicar no menu Exibir/Server Explorar (Apenas para SQLSERVER)



Abrirá o Gerenciador de Servidores. Clicar com o botão direito do mouse em Conexões de Dados e escolher a opção Criar Novo Banco de dados no SqlServer .



- a. Colar o nome do servidor copiado anteriormente no SSMS em Nome do Servidor;
- b. Em Encriptografar escolher Optional(False);
- c. Marcar Certificado de Servidor Confiável.
- d. Colocar o nome do Banco de dados;
- e. Ok

The image shows a screenshot of the 'Criar Novo Banco de Dados do SQL Server' (Create New SQL Server Database) dialog box. The dialog has a title bar with a question mark and a close button. The main text reads: 'Digite as informações para se conectar ao SQL Server e especifique o nome de um banco de dados para ser criado...'. Below this, there is a section for 'Nome do servidor:' with a dropdown menu and an 'Atualizar' button. A section titled 'Fazer login no servidor' contains two radio buttons: 'Usar Autenticação do Windows' (selected) and 'Usar Autenticação do SQL Server'. Below these are fields for 'Nome de usuário:' and 'Senha:'. The 'Encriptografar:' section has a dropdown menu set to 'Mandatory (True)' and two checkboxes: 'Certificado de Servidor Confiável' (checked) and 'Salvar minha senha'. At the bottom, there is a field for 'Nome do novo banco de dados:' and two buttons: 'OK' and 'Cancelar'.

Criar Novo Banco de Dados do SQL Server ? X

Digite as informações para se conectar ao SQL Server e especifique o nome de um banco de dados para ser criado...

Nome do servidor:

Atualizar

Fazer login no servidor

☒ Usar Autenticação do Windows

☐ Usar Autenticação do SQL Server

Nome de usuário:

Senha:

Encriptografar: Mandatory (True)

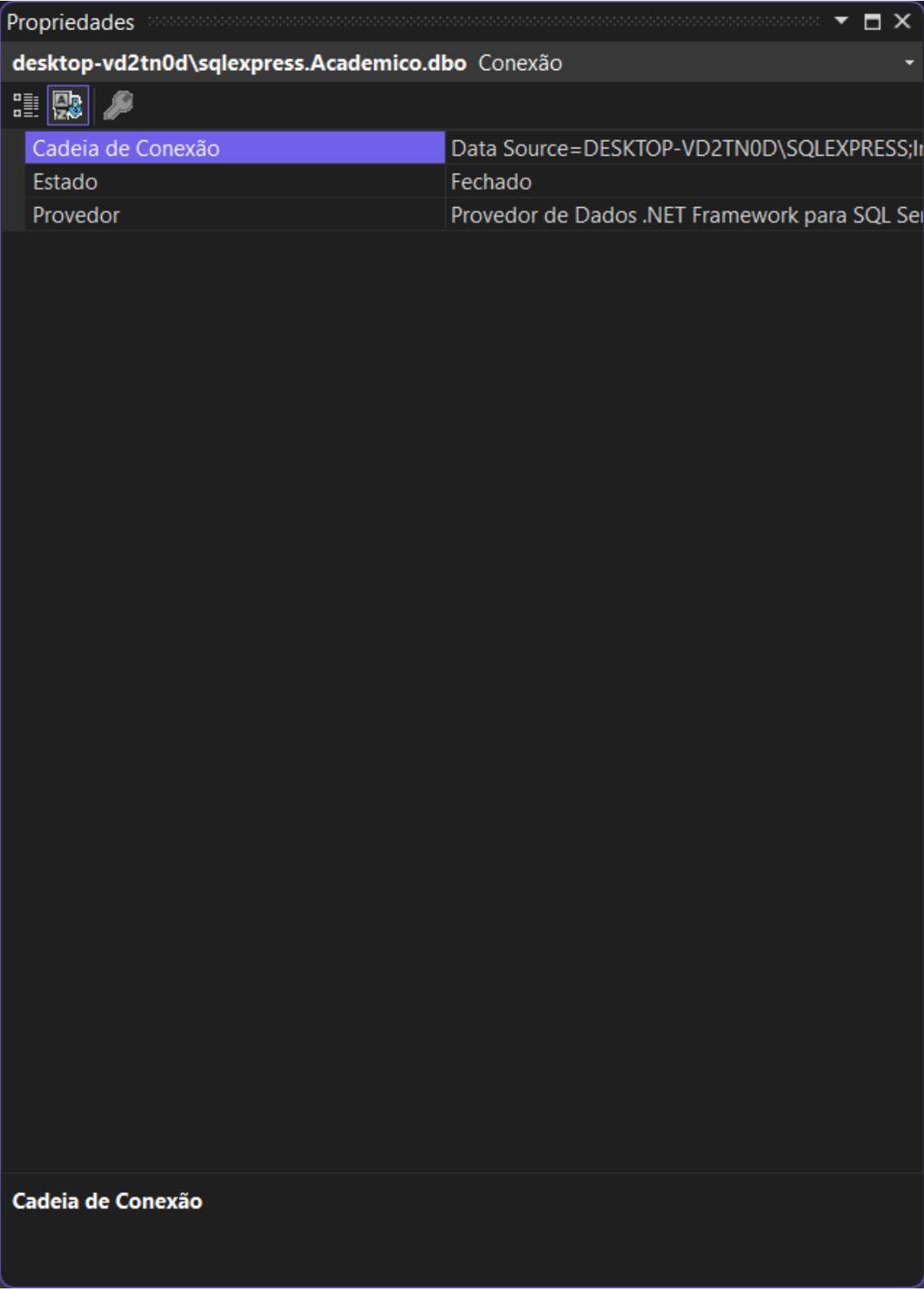
☒ Certificado de Servidor Confiável

☐ Salvar minha senha

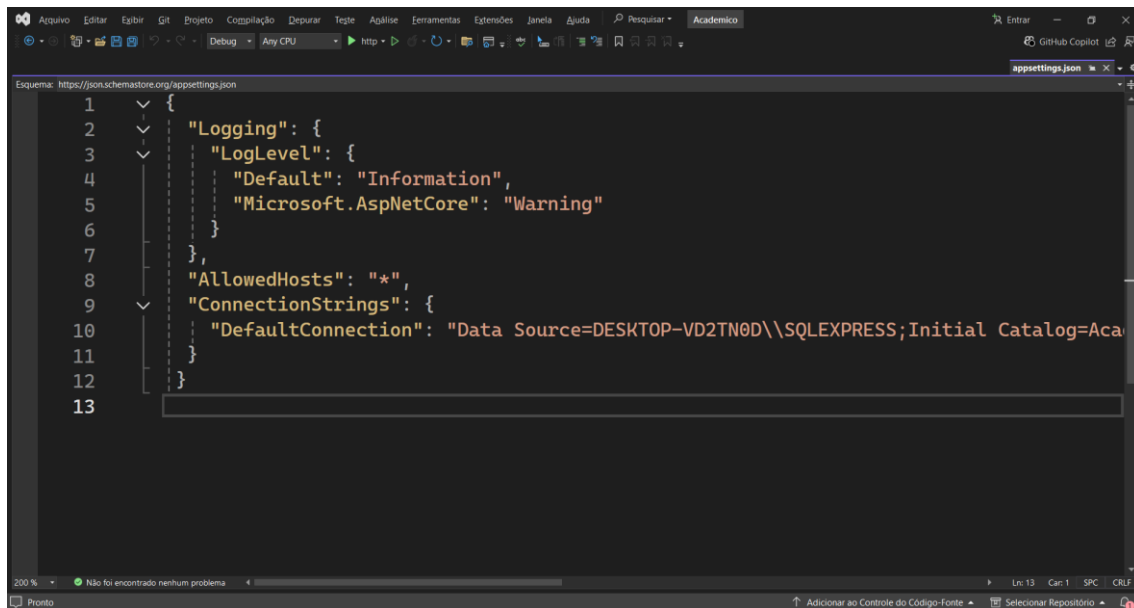
Nome do novo banco de dados:

OK Cancelar

8. Selecionar a conexão criada no item anterior e ir no menu Exibir/Janela de Propriedades (Apenas para SQLSERVER)
 - a. Copiar a Cadeia de Conexão



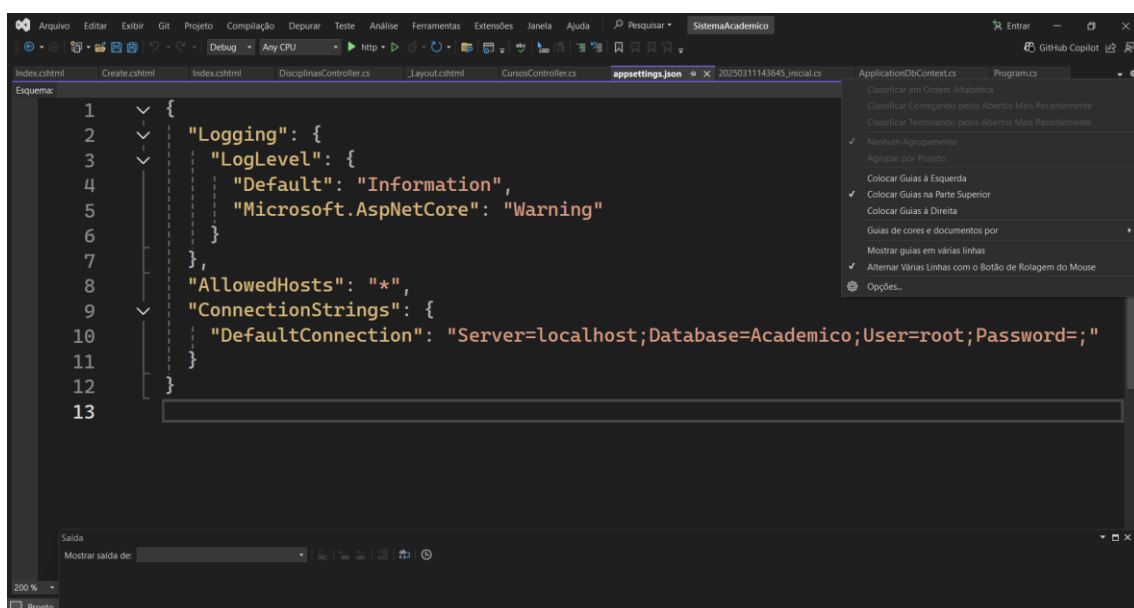
9. Editar o arquivo do projeto Appsettings.json e acrescentar a string de conexão, colando a Cadeia de Conexão copiada no item anterior em DefaultConnection, caso o banco de dados for SQLSERVER.



The screenshot shows the Visual Studio IDE with the 'appsettings.json' file open. The file contains a JSON configuration for logging and connection strings. The 'DefaultConnection' property is set to a SQL Server connection string.

```
1 {
2   "Logging": {
3     "LogLevel": {
4       "Default": "Information",
5       "Microsoft.AspNetCore": "Warning"
6     }
7   },
8   "AllowedHosts": "*",
9   "ConnectionStrings": {
10    "DefaultConnection": "Data Source=DESKTOP-VD2TN0D\\SQLEXPRESS;Initial Catalog=Aca
11  }
12 }
13
```

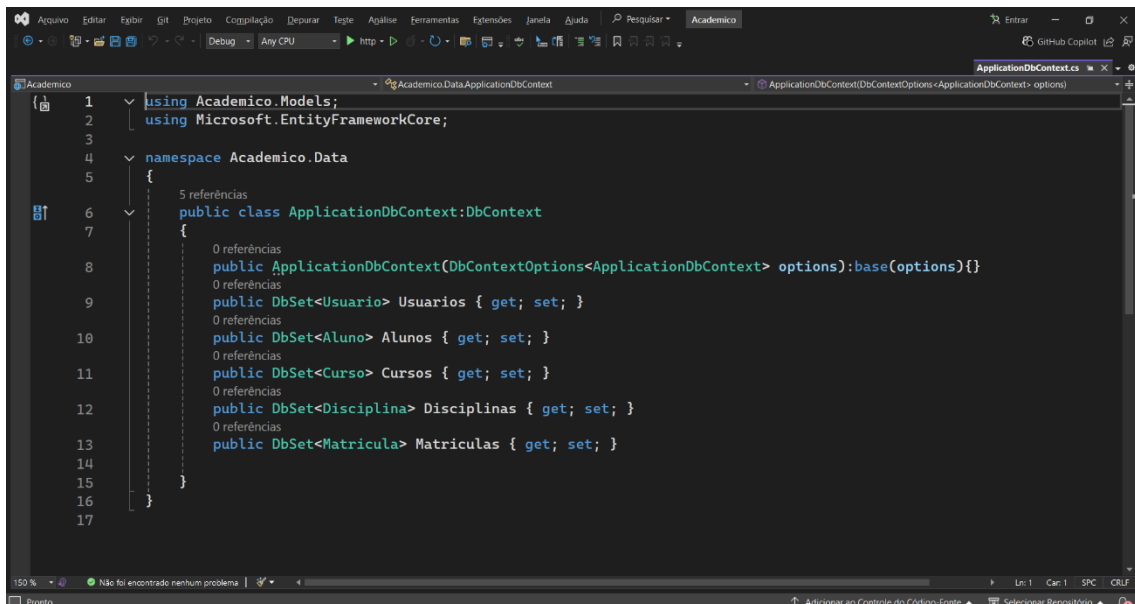
Se for MySQL fazer como abaixo.



The screenshot shows the Visual Studio IDE with the 'appsettings.json' file open. The file contains a JSON configuration for logging and connection strings. The 'DefaultConnection' property is set to a MySQL connection string. A context menu is visible on the right side of the editor.

```
1 {
2   "Logging": {
3     "LogLevel": {
4       "Default": "Information",
5       "Microsoft.AspNetCore": "Warning"
6     }
7   },
8   "AllowedHosts": "*",
9   "ConnectionStrings": {
10    "DefaultConnection": "Server=localhost;Database=Academico;User=root;Password=;"
11  }
12 }
13
```

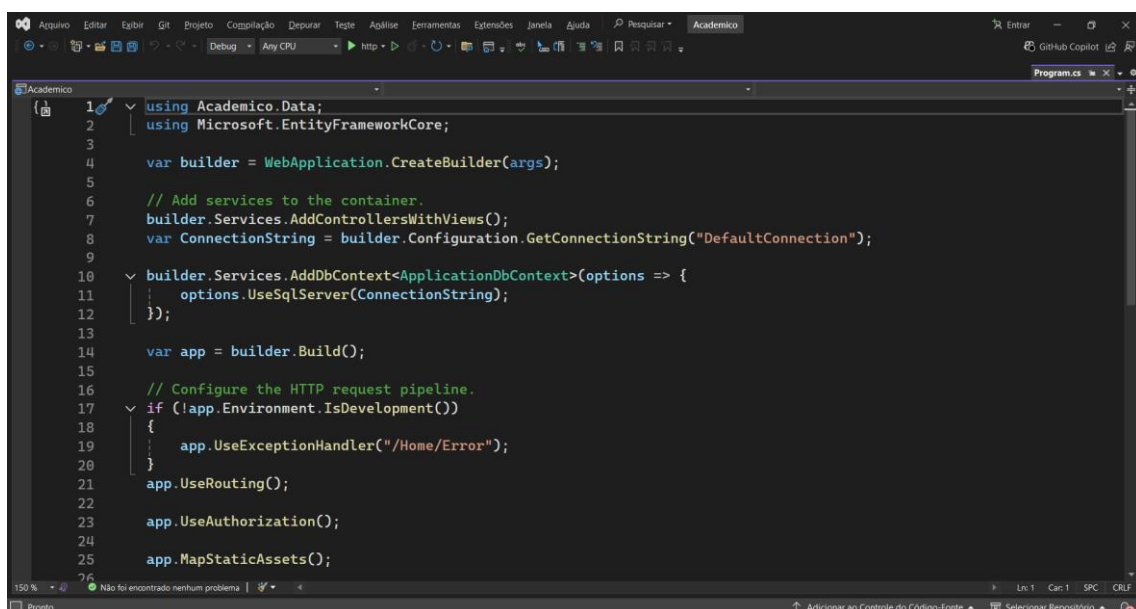
10. Criar uma pasta chamada Data e criar uma classe ApplicationDbContext.cs que herda de DbContext.



The screenshot shows the Visual Studio IDE with the file 'Academico.Data.ApplicationDbContext.cs' open. The code defines a class 'ApplicationDbContext' that inherits from 'DbContext' within the 'Academico.Data' namespace. It includes several DbSet properties for 'Usuarios', 'Alunos', 'Cursos', 'Disciplinas', and 'Matriculas'. The IDE interface shows the 'Debug' configuration and a status bar at the bottom indicating 'Pronto'.

```
1 using Academico.Models;
2 using Microsoft.EntityFrameworkCore;
3
4 namespace Academico.Data
5 {
6     public class ApplicationDbContext:DbContext
7     {
8         public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options):base(options){}
9         public DbSet<Usuario> Usuarios { get; set; }
10        public DbSet<Aluno> Alunos { get; set; }
11        public DbSet<Curso> Cursos { get; set; }
12        public DbSet<Disciplina> Disciplinas { get; set; }
13        public DbSet<Matricula> Matriculas { get; set; }
14    }
15 }
16
17
```

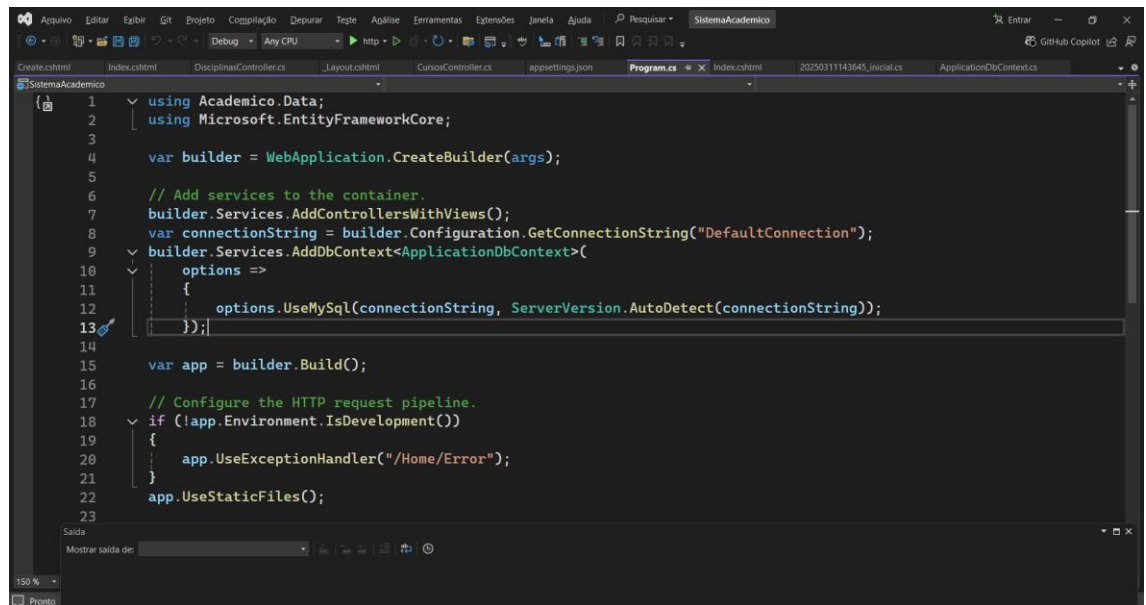
11. Editar o arquivo Program.cs do projeto e fazer as alterações para registrar a conexão com o banco de dados utilizando a classe criada no item anterior. Linhas de 8 a 12 da imagem abaixo (Apenas para SQLSERVER).



The screenshot shows the Visual Studio IDE with the 'Program.cs' file open. The code configures the web application, including adding services, getting the connection string, and using the 'ApplicationDbContext' class. The IDE interface shows the 'Debug' configuration and a status bar at the bottom indicating 'Pronto'.

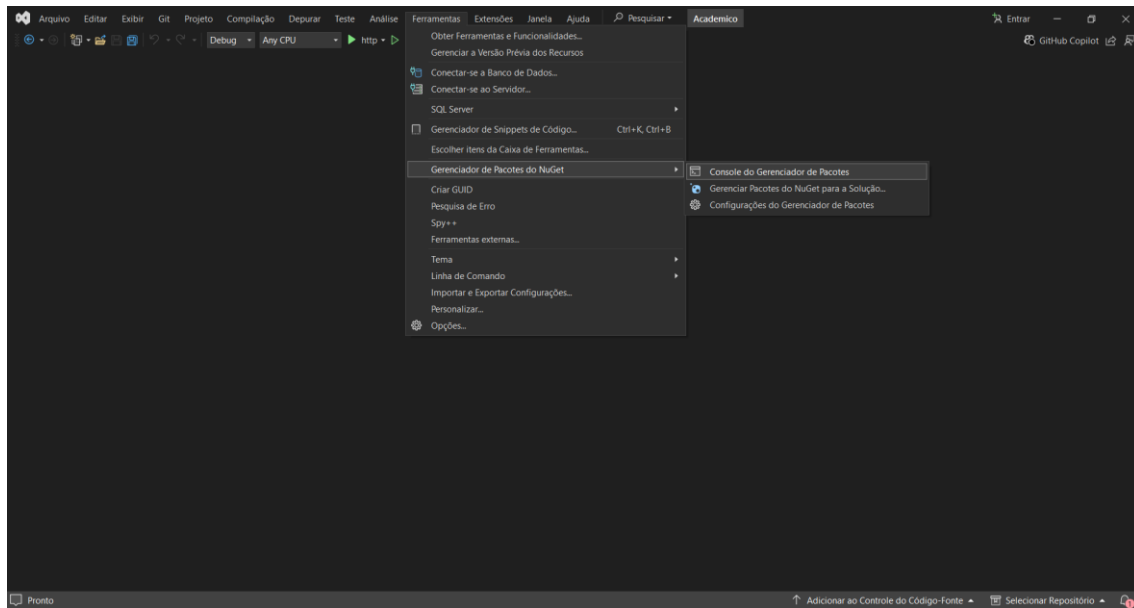
```
1 using Academico.Data;
2 using Microsoft.EntityFrameworkCore;
3
4 var builder = WebApplication.CreateBuilder(args);
5
6 // Add services to the container.
7 builder.Services.AddControllersWithViews();
8 var ConnectionString = builder.Configuration.GetConnectionString("DefaultConnection");
9
10 builder.Services.AddDbContext<ApplicationDbContext>(options => {
11     options.UseSqlServer(ConnectionString);
12 });
13
14 var app = builder.Build();
15
16 // Configure the HTTP request pipeline.
17 if (!app.Environment.IsDevelopment())
18 {
19     app.UseExceptionHandler("/Home/Error");
20 }
21 app.UseRouting();
22
23 app.UseAuthorization();
24
25 app.MapStaticAssets();
26
```

Para o MySQL o program.cs fica conforme a figura abaixo. Linhas de 8 a 13.



```
1 using Academico.Data;
2 using Microsoft.EntityFrameworkCore;
3
4 var builder = WebApplication.CreateBuilder(args);
5
6 // Add services to the container.
7 builder.Services.AddControllersWithViews();
8 var connectionString = builder.Configuration.GetConnectionString("DefaultConnection");
9 builder.Services.AddDbContext<ApplicationDbContext>(<
10     options =>
11     {
12         options.UseMySQL(connectionString, ServerVersion.AutoDetect(connectionString));
13     });
14
15 var app = builder.Build();
16
17 // Configure the HTTP request pipeline.
18 if (!app.Environment.IsDevelopment())
19 {
20     app.UseExceptionHandler("/Home/Error");
21 }
22 app.UseStaticFiles();
23
```

12. Criar as classes correspondentes as tabelas do banco de dados na pasta Models, fazendo os relacionamentos entre elas (1x1, 1xn, nxn).
13. Para criar as tabelas no banco de dados, conforme o mapeamento feito nas classe
 - a. Abrir Console do Gerenciador de Pacote;



- b. Digitar para criar a Migration: Add-Migration Initial (o nome da migration pode ser qualquer um);
- c. Após criada a migration criar as tabelas no BD: Update-Database

Obs.: para que os passos sejam executados corretamente, não pode haver erros de compilação e todos os passos anteriores devem estar corretos.