

UNOESC VIDEIRA

24/08/2023

ALUNO: JOÃO VITOR BORGES GODINHO

DISCIPLINA: PROGRAMAÇÃO II

PROFESSOR: MAURÍCIO GONZATTO

Atividade:

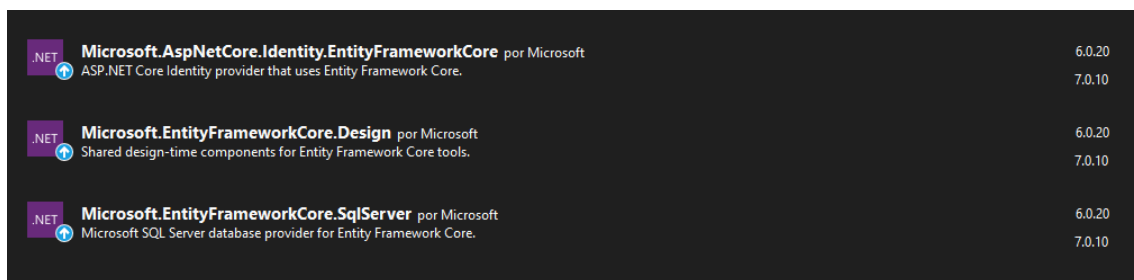
- 1- Crie um projeto ASP.NET MVS (template) com nome PjNomeSobrenome
- 2- Configure no projeto os NuGet Packages do EF Core
- 3- Especifique uma classe Model para Psicologo e defina um DbContext herdando do Identity
- 4- Configure no appsettings.json uma string de conexão com o SQL Server
- 5- Defina os parâmetros de configuração dos serviços da aplicação no program.cs
- 5- Crie a migration (dotnet ef migrations add Initial) e atualiza o banco (dotnet ef database update)

Roteiro:

Criar um projeto com o Visual Studio > Utilizar o modelo de projeto "Aplicativo Web do ASP.NET Core (MVC)" > Renomear o projeto > Selecionar a versão do .net e desativar o HTTPS

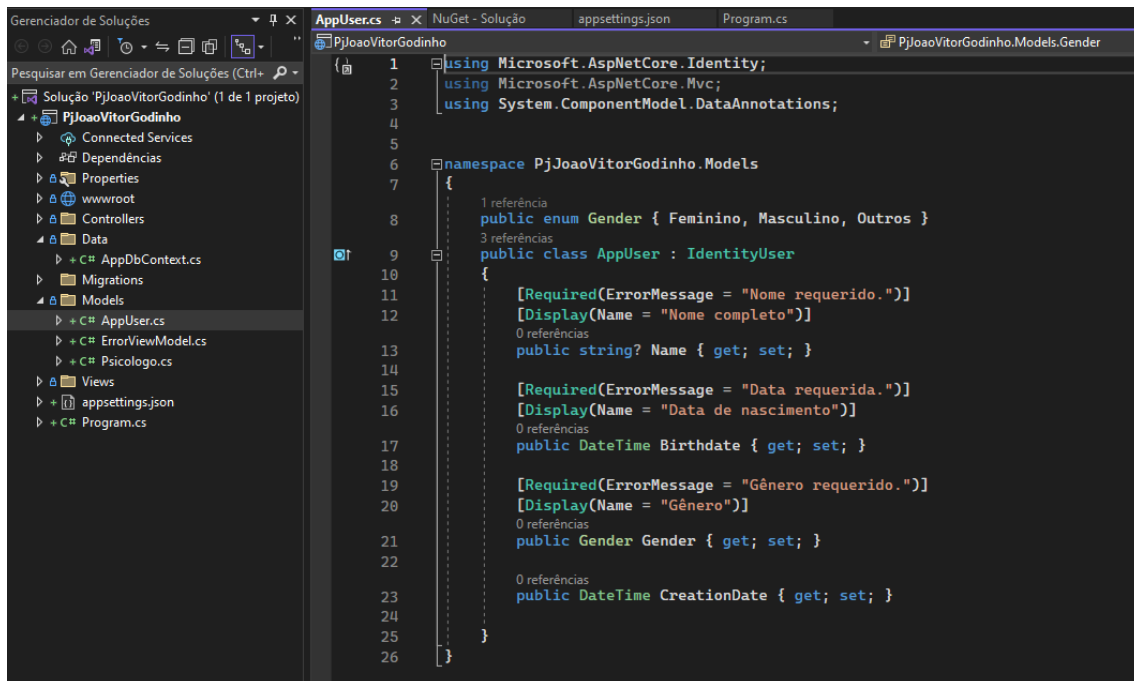
Abrir o gerenciador dos pacotes NuGet e instalar os pacotes:

- > Microsoft.AspNetCore.Identity.EntityFrameworkCore
- > Microsoft.EntityFrameworkCore.Design
- > Microsoft.EntityFrameworkCore.SqlServer



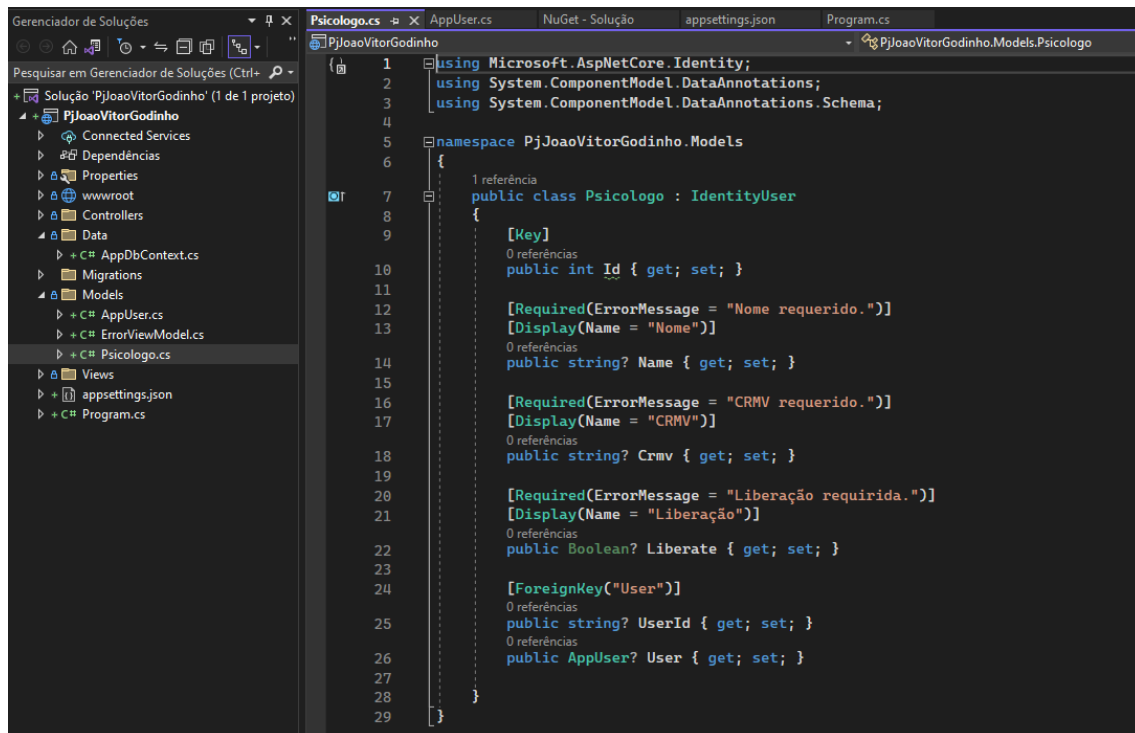
Na pasta Models criar dois novos itens de Classe C#, "AppUser" e "Psicologo";

> Em AppUser definimos os objetos que todos os tipos de cadastros usarão (Um perfil de usuário geral)



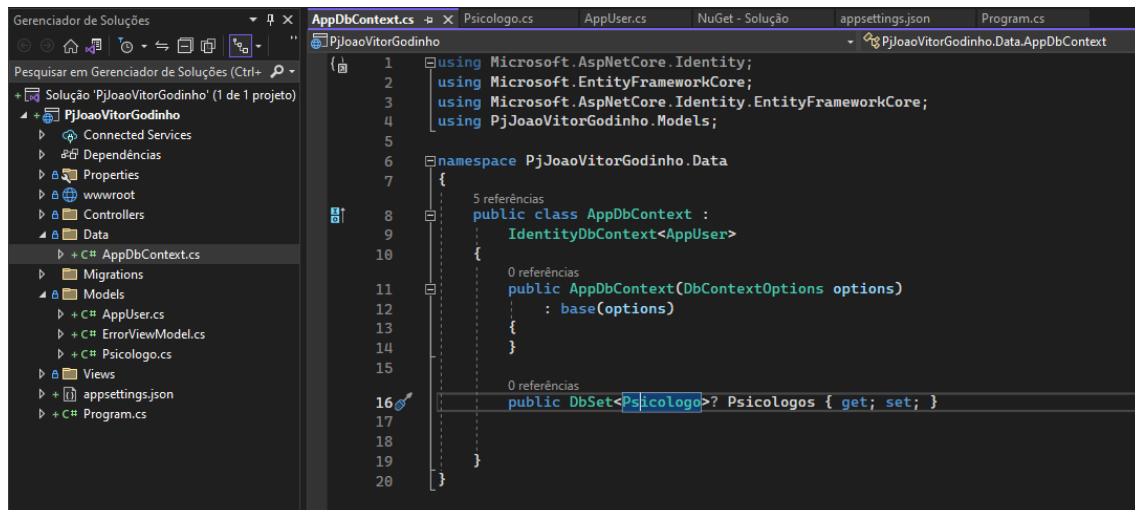
> Na classe Psicologo, definimos os objetos específicos para a classe (Id, Name, Crmv, Liberate) e herdamos os objetos de AppUser;

> Definir chave primária, chave estrangeira, mensagens de erro e requisições.



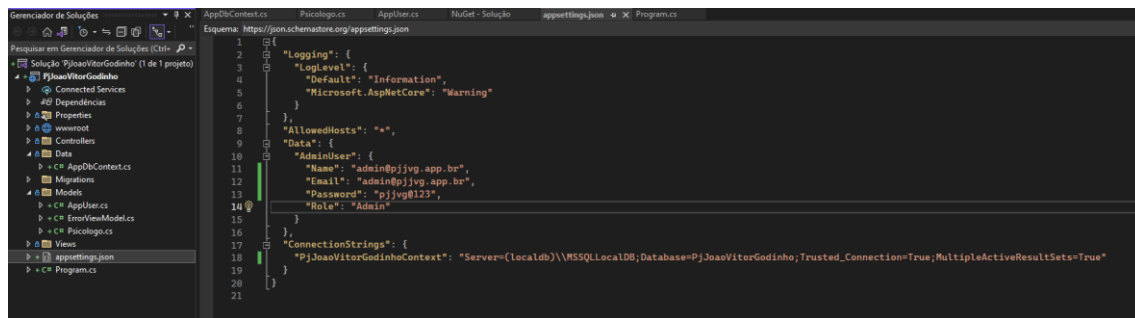
```
1 using Microsoft.AspNetCore.Identity;
2 using System.ComponentModel.DataAnnotations;
3 using System.ComponentModel.DataAnnotations.Schema;
4
5 namespace PjJoaoVitorGodinho.Models
6 {
7     public class Psicologo : IdentityUser
8     {
9         [Key]
10         public int Id { get; set; }
11
12         [Required(ErrorMessage = "Nome requerido.")]
13         [Display(Name = "Nome")]
14         public string? Name { get; set; }
15
16         [Required(ErrorMessage = "CRMV requerido.")]
17         [Display(Name = "CRMV")]
18         public string? Crmv { get; set; }
19
20         [Required(ErrorMessage = "Liberação requerida.")]
21         [Display(Name = "Liberação")]
22         public Boolean? Liberate { get; set; }
23
24         [ForeignKey("User")]
25         public string? UserId { get; set; }
26         public AppUser? User { get; set; }
27     }
28 }
29
```

Criamos uma pasta Data e dentro dela, uma classe e definimos o DbContext herdando do Identity



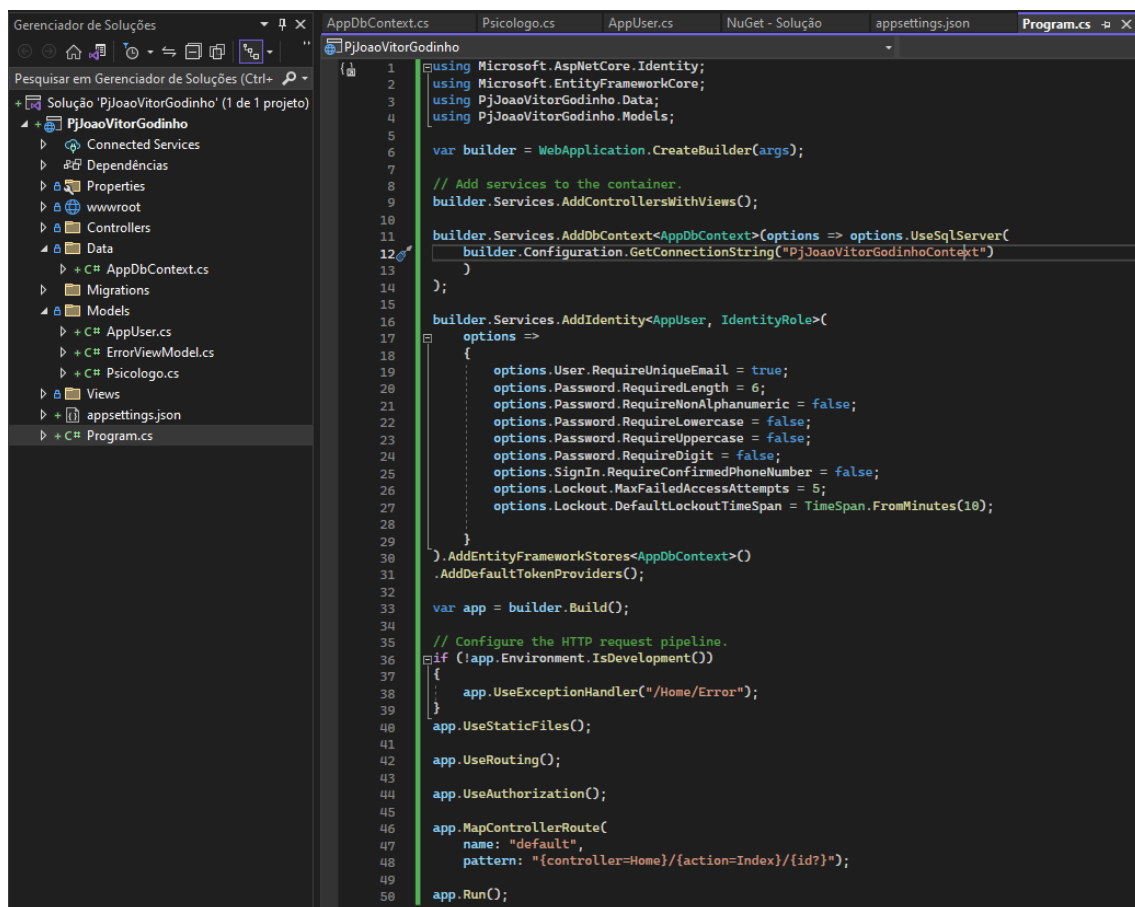
```
1 using Microsoft.AspNetCore.Identity;
2 using Microsoft.EntityFrameworkCore;
3 using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
4 using PjJoaoVitorGodinho.Models;
5
6 namespace PjJoaoVitorGodinho.Data
7 {
8     public class AppDbContext : IdentityDbContext<AppUser>
9     {
10         public AppDbContext(DbContextOptions options)
11             : base(options)
12         {
13         }
14
15         public DbSet<Psicologo>? Psicologos { get; set; }
16     }
17 }
18
19
20
```

Em appsettings.json criamos um usuário para o Admin e setamos uma string para a conexão



```
1 {
2   "Logging": {
3     "LogLevel": {
4       "Default": "Information",
5       "Microsoft.AspNetCore": "Warning"
6     }
7   },
8   "AllowedHosts": "*",
9   "Data": {
10    "AdminUser": {
11      "Name": "admin@pjjvg.app.br",
12      "Email": "admin@pjjvg.app.br",
13      "Password": "pjjvg@123",
14      "Role": "Admin"
15    },
16  },
17  "ConnectionStrings": {
18    "PjJoaoVitorGodinhoContext": "Server=(localdb)\MSSQLLocalDB;Database=PjJoaoVitorGodinho;Trusted_Connection=True;MultipleActiveResultSets=True"
19  }
20 }
21
```

Em Program.cs adicionamos as obrigatiedades e requisições para criação de usuário e senha



```
1 using Microsoft.AspNetCore.Identity;
2 using Microsoft.EntityFrameworkCore;
3 using PjJoaoVitorGodinho.Data;
4 using PjJoaoVitorGodinho.Models;
5
6 var builder = WebApplication.CreateBuilder(args);
7
8 // Add services to the container.
9 builder.Services.AddControllersWithViews();
10
11 builder.Services.AddDbContext<AppDbContext>(options => options.UseSqlServer(
12     builder.Configuration.GetConnectionString("PjJoaoVitorGodinhoContext")
13 ));
14
15 builder.Services.AddIdentity<AppUser, IdentityRole>(
16     options =>
17     {
18         options.User.RequireUniqueEmail = true;
19         options.Password.RequiredLength = 6;
20         options.Password.RequireNonAlphanumeric = false;
21         options.Password.RequireLowercase = false;
22         options.Password.RequireUppercase = false;
23         options.Password.RequireDigit = false;
24         options.SignIn.RequireConfirmedPhoneNumber = false;
25         options.Lockout.MaxFailedAccessAttempts = 5;
26         options.Lockout.DefaultLockoutTimeSpan = TimeSpan.FromMinutes(10);
27     }
28 );
29 builder.Services.AddEntityFrameworkStores<AppDbContext>()
30     .AddDefaultTokenProviders();
31
32 var app = builder.Build();
33
34 // Configure the HTTP request pipeline.
35 if (!app.Environment.IsDevelopment())
36 {
37     app.UseExceptionHandler("/Home/Error");
38 }
39 app.UseStaticFiles();
40
41 app.UseRouting();
42
43 app.UseAuthorization();
44
45 app.MapControllerRoute(
46     name: "default",
47     pattern: "{controller=Home}/{action=Index}/{id?}");
48
49 app.Run();
50
```

Dentro da pasta do projeto instalamos o EF utilizando "dotnet tool install --global dotnet-ef"

> criar a Migration utilizando "dotnet ef migrations add Initial"

> Ainda dentro da pasta, atualizamos o banco de dados utilizando "dotnet ef database update"