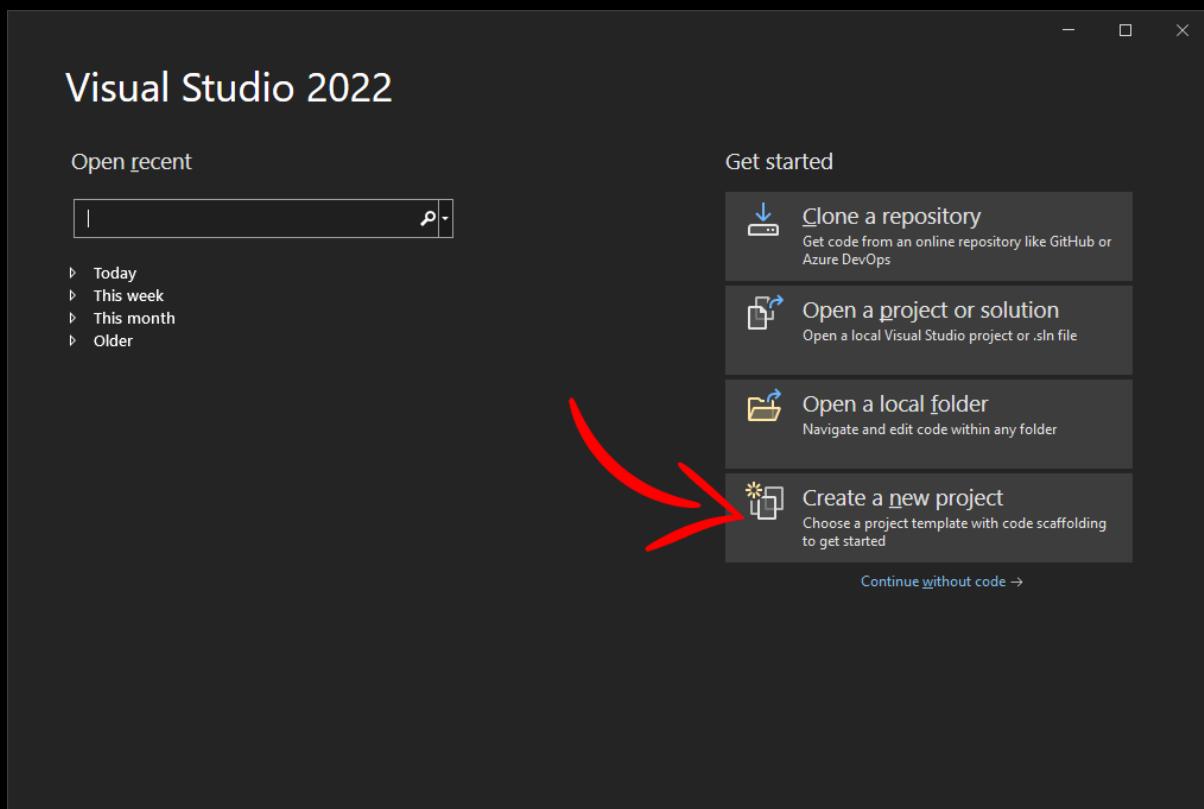


Atualização .NET 6

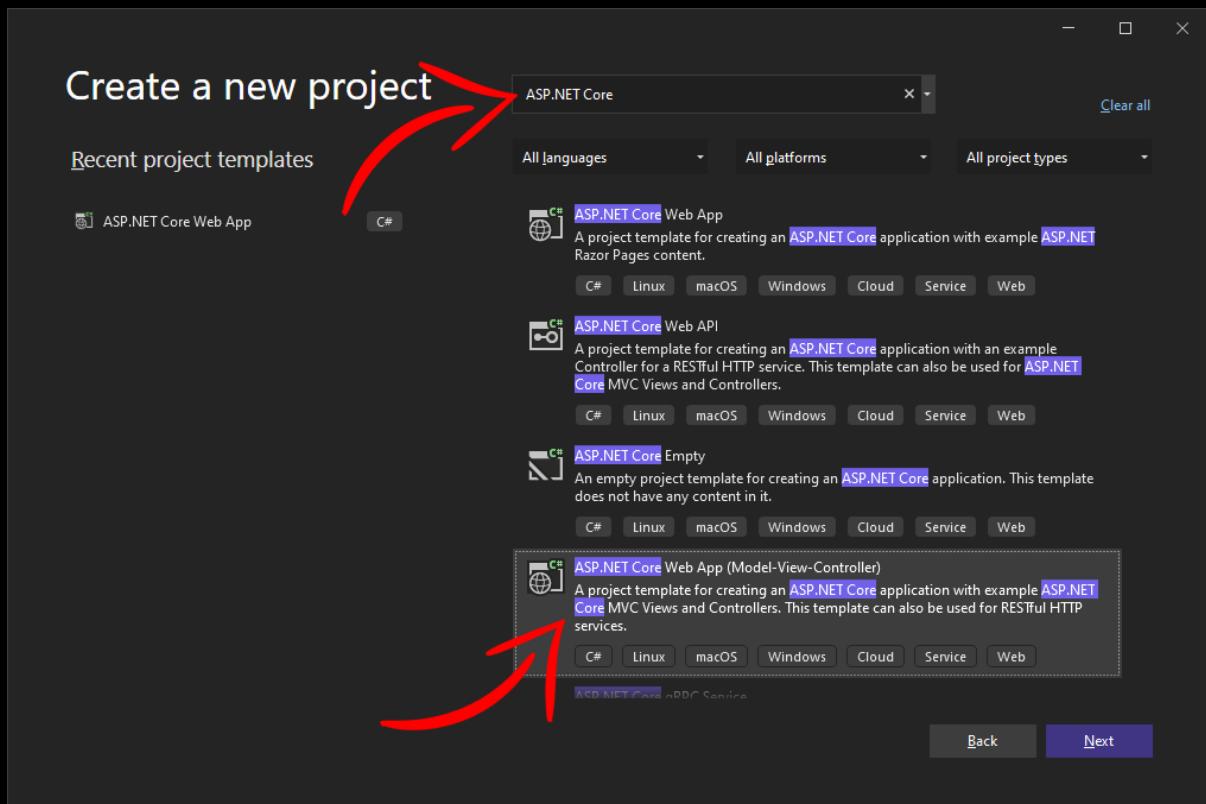
O .NET 6 trouxe algumas mudanças na maneira de criar e configurar a App MVC, siga as instruções com atenção:

Utilizando Visual Studio 2019 ou 2022 siga os passos:

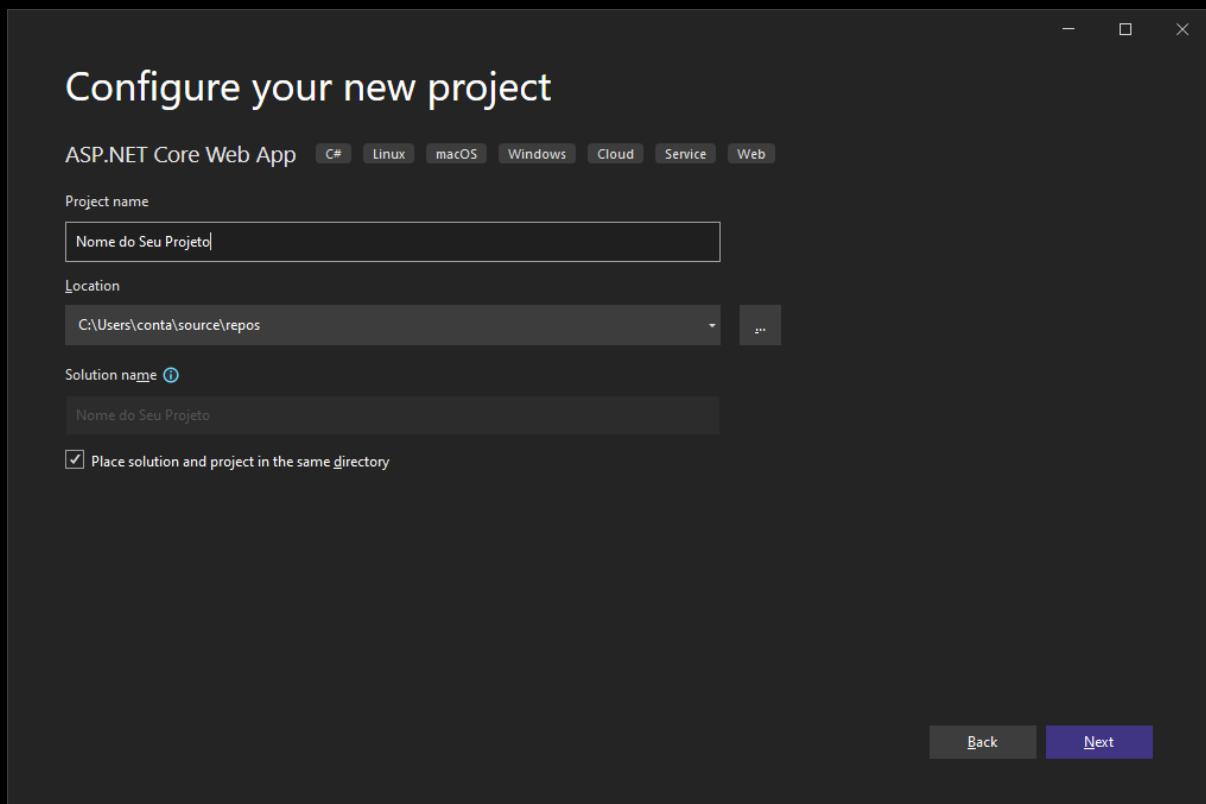
Criação do projeto:



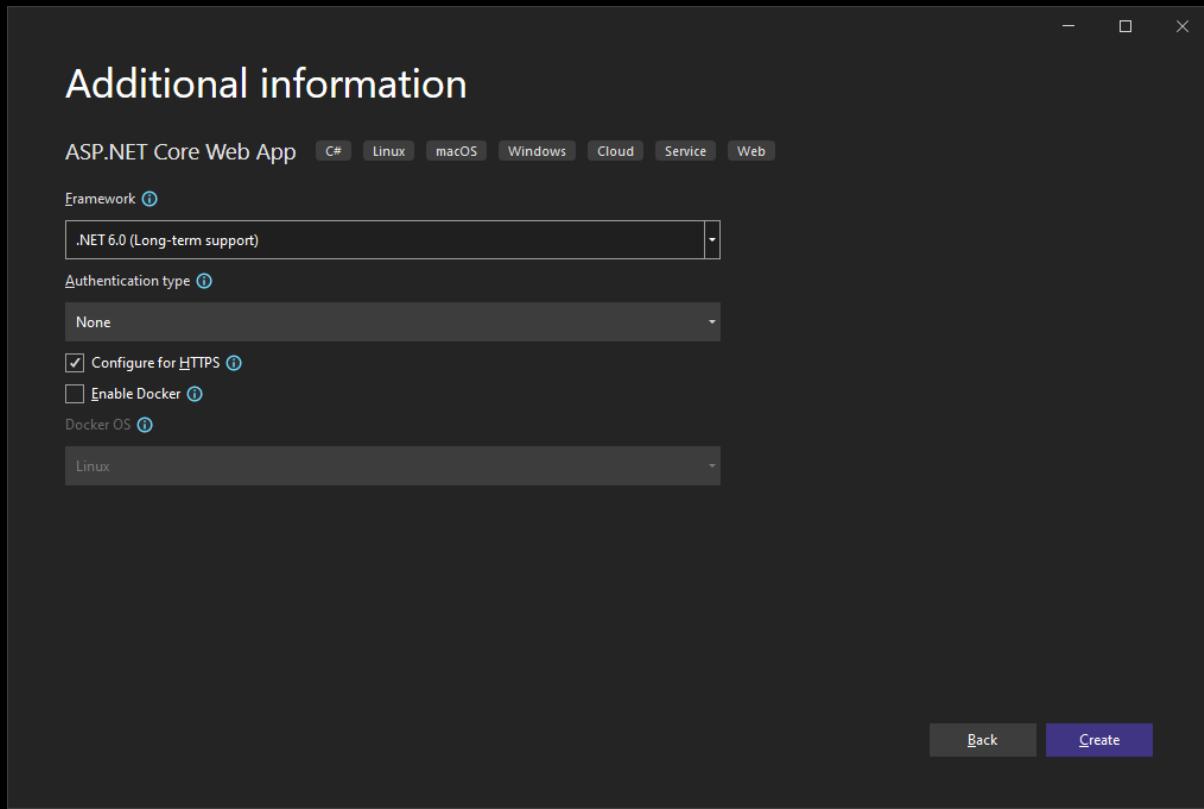
Procure por ASP.NET Core para facilitar a pesquisa do template correto, na sequencia selecione a opção ASP.NET Core Web App (Model-View-Controller)



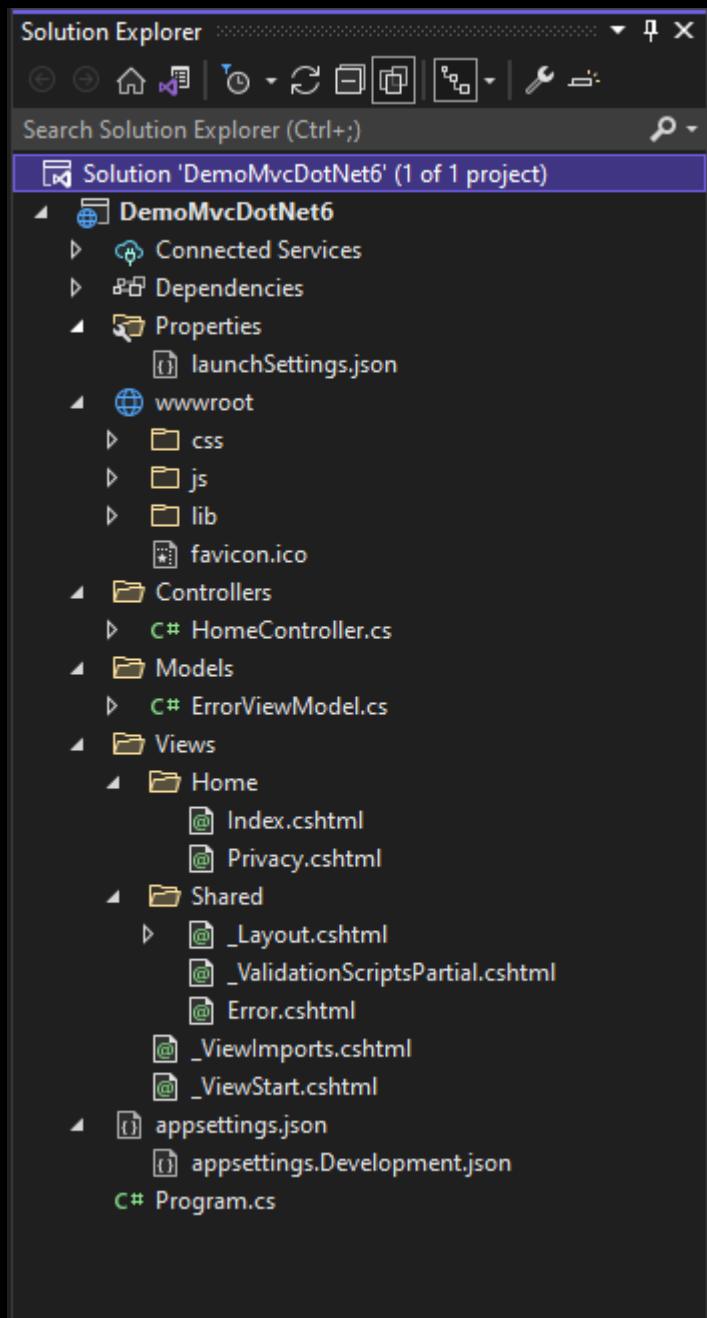
Defina o nome do seu projeto:



Escolha a versão do .NET, por razões de estudos sempre utilize a última versão, caso queira adicionar suporte ao Identity marque o combo Authentication Type como **Individual Accounts**



O projeto será criado, observe a estrutura de projeto MVC do ASP.NET 6



Mudanças no MVC no ASP.NET 6

Por uma questão de compatibilização com o padrão de código escrito em C# até o momento é recomendável desativar o suporte a validação de Nullable Types do CSPROJ (clique 2x no projeto para abrir):

```
<Project Sdk="Microsoft.NET.Sdk.Web">  
  <PropertyGroup>
```

```
<TargetFramework>net6.0</TargetFramework>
<Nullable>enable</Nullable>
<ImplicitUsings>enable</ImplicitUsings>
<UserSecretsId>aspnet-AppMvcBasica-E371D4CD-D91B-418D-B969-
E35920E572C4</UserSecretsId>
</PropertyGroup>

<ItemGroup>
    <PackageReference
Include="Microsoft.AspNetCore.Diagnostics.EntityFrameworkCore" Version="6.0.3"
/>
    <PackageReference
Include="Microsoft.AspNetCore.Identity.EntityFrameworkCore" Version="6.0.3" />
    <PackageReference Include="Microsoft.AspNetCore.Identity.UI" Version="6.0.3"
/>
    <PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer"
Version="6.0.3" />
    <PackageReference Include="Microsoft.EntityFrameworkCore.Tools"
Version="6.0.3" />
</ItemGroup>

</Project>
```

É importante remover esta chave em todos os projetos que forem criados caso não queira tratar cada propriedade do seu código como Nullable

```
<Nullable>enable</Nullable>
```

A única grande mudança é a saída da classe Startup.cs, toda sua responsabilidade foi levada para a classe Program.cs (que já existia desde sempre). Não é mais necessário ter uma classe startup, mas caso você queira manter a sua num projeto ASP.NET 6 ainda é possível, siga este tutorial [no Youtube](#)

Preste atenção aos comentários na classe Program.cs abaixo, é importante entender algumas responsabilidades:

```
// O builder é responsável por fornecer os métodos de controle
// dos serviços e demais funcionalidades na configuração da App
var builder = WebApplication.CreateBuilder(args);

// Daqui para baixo é conteúdo que vinha dentro do método ConfigureServices()
// na antiga Startup.cs
// Nesta área adicionamos serviços ao pipeline

// Essa é a nova forma de adicionar o MVC ao projeto
// Não se usa mais services.AddMvc();
builder.Services.AddControllersWithViews();
```

```
// Essa linha precisa sempre ficar por último na configuração dos serviços
var app = builder.Build();

// Daqui para baixo é conteúdo que vinha dentro do método Configure() na antiga
Startup.cs
// Aqui se configura comportamentos do request dentro do pipeline
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

// Note a ligeira mudança na declaração da rota padrão
// No caso de precisar mapear mais de uma rota duplique o código abaixo
app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");

// Essa linha precisa sempre ficar por último na configuração do request
pipeline
app.Run();
```

Entendendo estas sutis mudanças você não terá problemas em estudar utilizando o ASP.NET 6 em seu projeto.