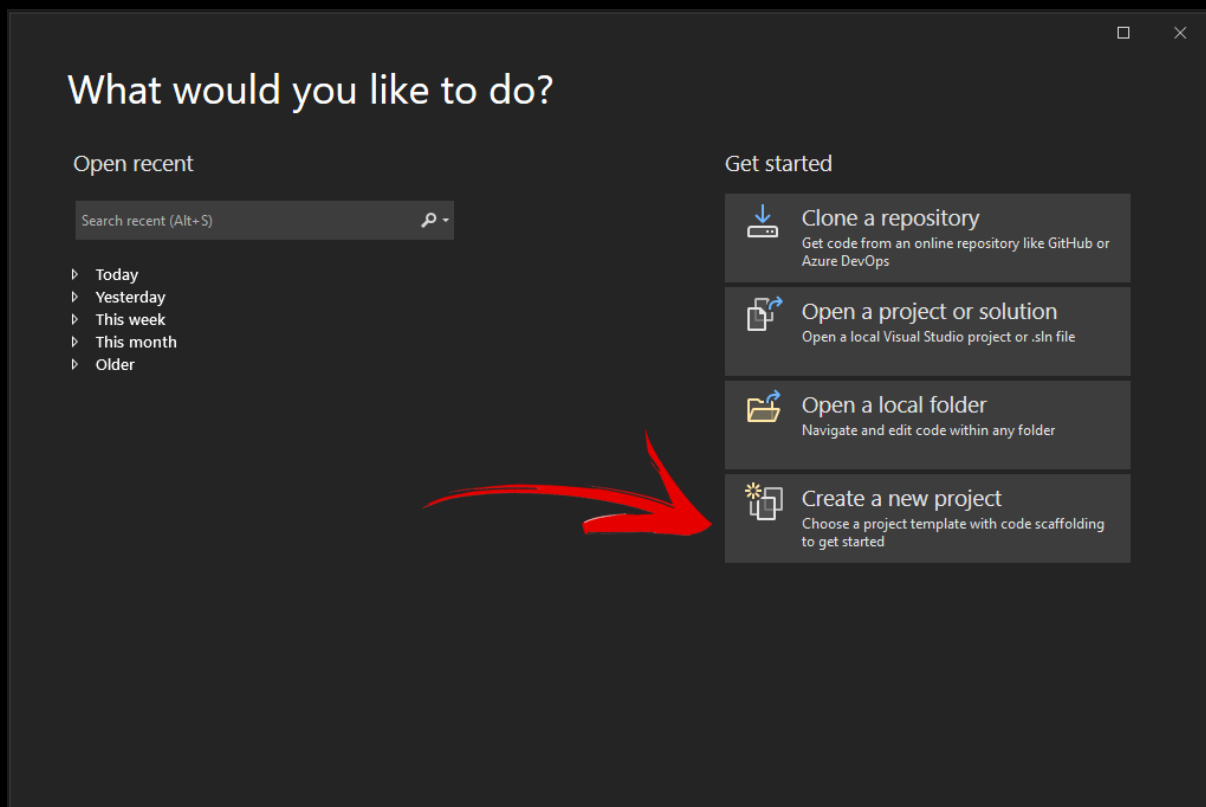


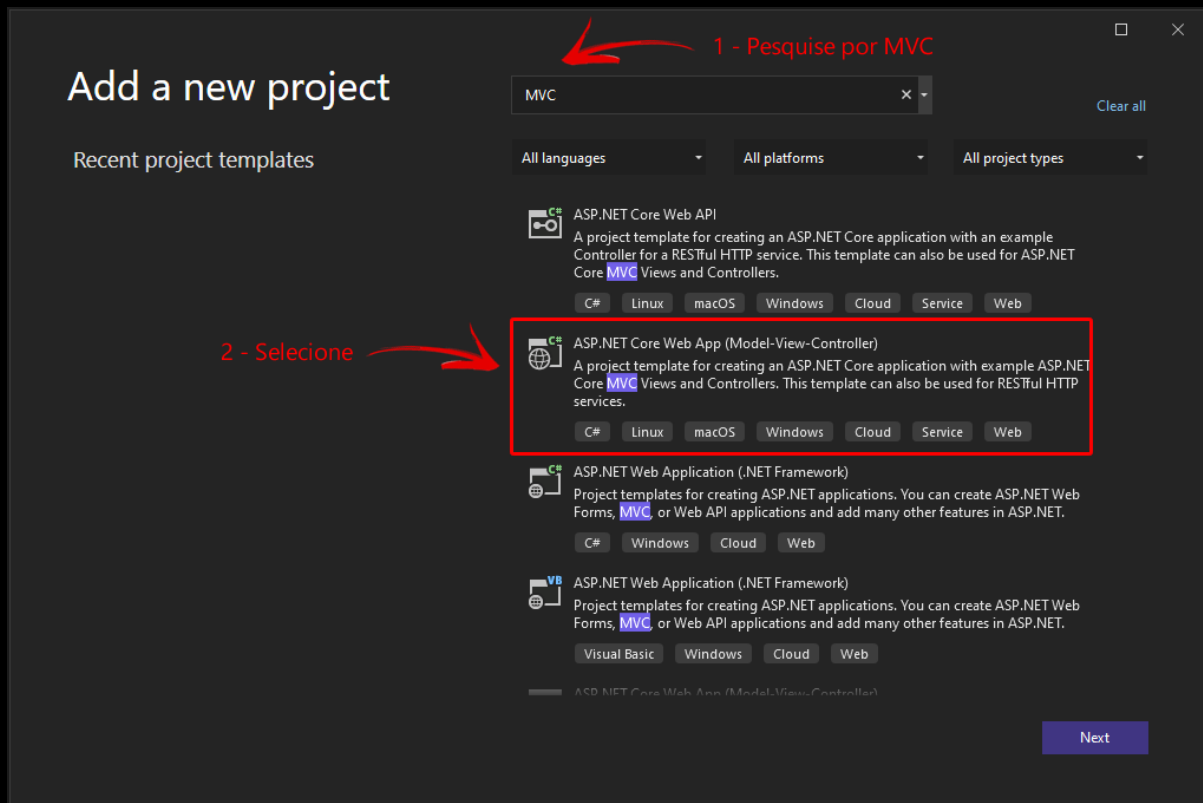
Roteiro VS2022 e .NET 6

Utilize o roteiro abaixo para reproduzir no Visual Studio 2022 com ASP.NET 6 caso esteja utilizando esta versão, isto lhe dará suporte para os processos executados nas próximas aulas deste módulo.

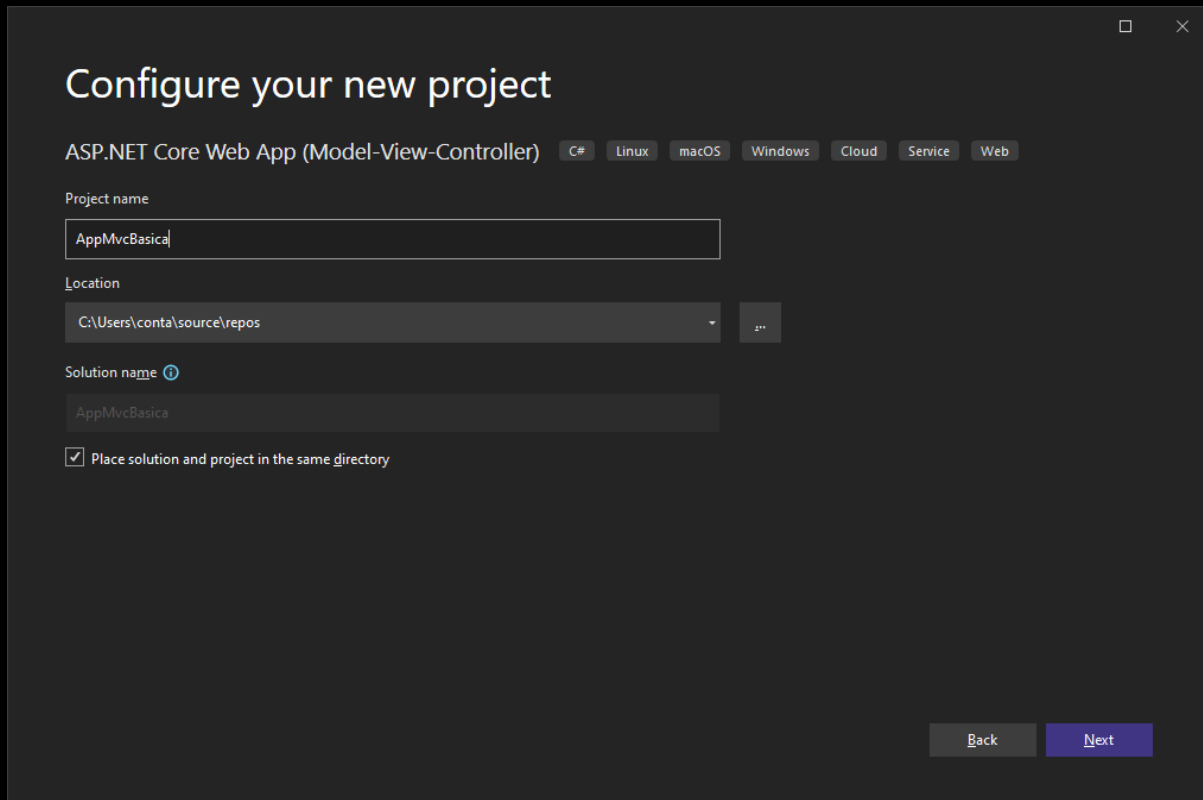
Na janela inicial do Visual Studio 2022, crie um novo projeto



Pesquise pelo template MVC e escolha o modelo conforme a imagem

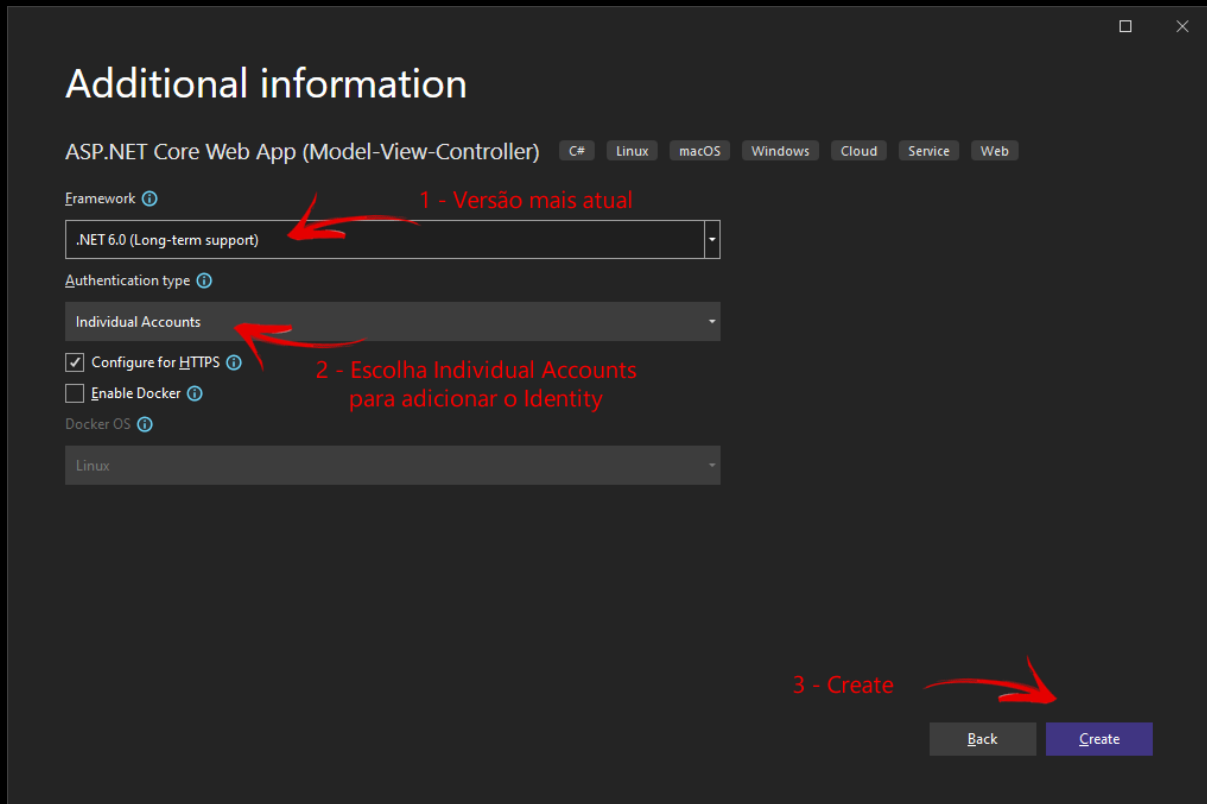


Defina o nome do projeto



Defina a versão do .NET a ser utilizada (neste caso a 6.0)

Não esqueça de adicionar o suporte ao Identity com o Individual Accounts



Additional information

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

Framework ⓘ

.NET 6.0 (Long-term support) 1 - Versão mais atual

Authentication type ⓘ

Individual Accounts 2 - Escolha Individual Accounts para adicionar o Identity

☒ Configure for HTTPS ⓘ

☐ Enable Docker ⓘ

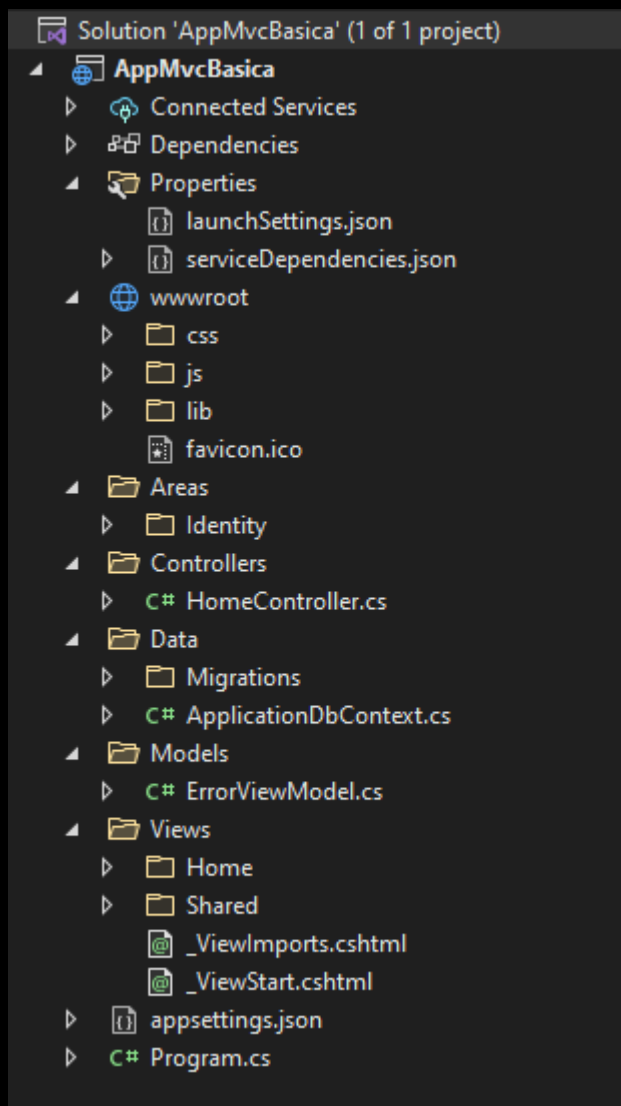
Docker OS ⓘ

Linux

3 - Create

Back Create

O resultado do projeto criado é esta estrutura de pastas:



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 C# (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"
    />
  </ItemGroup>
</Project>
```

```

    <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>
```

Analizando a classe Program.cs ela já vem preparada para todas as necessidades, daqui para frente basta adicionar / customizar conforme as novas implementações.

```

using AppMvcBasica.Data;
using Microsoft.AspNetCore.Identity;
using Microsoft.EntityFrameworkCore;

var builder = WebApplication.CreateBuilder(args);

// Guardando a connection string do arquivo appSettings.json
var connectionString =
builder.Configuration.GetConnectionString("DefaultConnection");

// Adicionando o suporte ao acesso ao DB do Identity via EF
builder.Services.AddDbContext<ApplicationDbContext>(options =>
    options.UseSqlServer(connectionString));

// Adicionando a tela de erro de banco de dados (para desenvolvimento)
builder.Services.AddDatabaseDeveloperPageExceptionFilter();

// Adicionando o Identity
builder.Services.AddDefaultIdentity<IdentityUser>(options =>
options.SignIn.RequireConfirmedAccount = true)
    .AddEntityFrameworkStores<ApplicationDbContext>();

// Adicionando o MVC
builder.Services.AddControllersWithViews();

// Gerando a APP
var app = builder.Build();

// Configuração conforme os ambientes
if (app.Environment.IsDevelopment())
{

```

```
        app.UseMigrationsEndPoint();
    }
    else
    {
        app.UseExceptionHandler("/Home/Error");
        app.UseHsts();
    }

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

    app.UseRouting();

    app.UseAuthentication();
    app.UseAuthorization();

    app.MapControllerRoute("default", "{controller=Home}/{action=Index}/{id?}");
    app.MapRazorPages();

    app.Run();
```