Primeiro vamos no terminal e executar o comando

**dotnet new mvc**

**Ele é bastante semelhante a criação de API**

**Tem pasta controller, json de configuração**

Para executar o projeto só digitar o comando

dotnet new mvc



namespace MVC.Controllers;

public class HomeController : Controller

{

    private readonly ILogger<HomeController> \_logger;

    public HomeController(ILogger<HomeController> logger)

    {

        \_logger = logger;

    }

    public IActionResult Index()

    {

        return View();

    }

    public IActionResult Privacy()

    {

        return View();

    }

    [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]

    public IActionResult Error()

    {

        return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });

    }

}

É bastante parecido com API com a diferença que herda somente de Controller

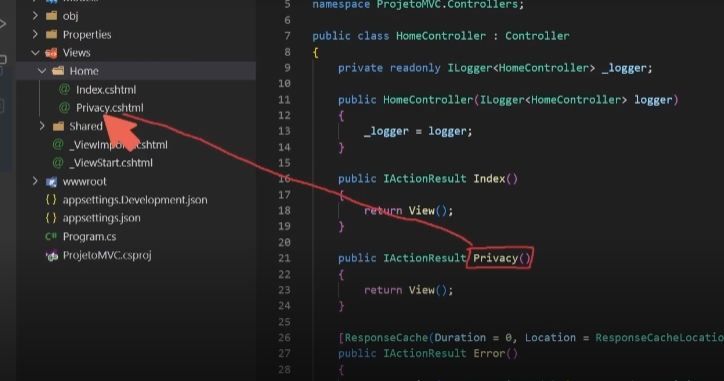
public class HomeController : Controller

Possui inúmeros métodos.

Ao analisar os métodos perceba que ele sempre retorna uma view return View(

Se for analisar a pasta view pode ser encontrada as duas pasta no caso Home, se eu tenho uma HomeCOntroller eu tb vou ter uma pasta Home na minha view.

Ou seja pode ser associado, toda a logica, todas as paginas da minha home controller ele vão ser redirecionados para pasta view da minha home



Se eu tenho um método Privacy o sistema tenta localizar o mesmo nome na views

Exemplo se eu navegar no endereço web

<https://localhost:7043/Home/Privacy>

ele vai fazer a busca da rota e carregar a pagina

/home vai procurar uma controller chamada Home

Localizou , é uma rota valida, então ele vai abrir o arquivo **Homecontroller.cs**

Ai tem /privacy, ele entrou na homecontroller, então ele vai procurar se tem um método chamado Privacy(), então ele vai entrar dentro do método , e o que método faz, ele retorna uma view, então eu vou procurar uma view , e como ele retorna uma view, exatamente com o mesmo nome do método Privacy,

Return view – o sistema vai procurar qual a controller que estou **HomeController ,** então ele vai procurar na pasta View uma pasta chamada Home, ele vai retornar uma view, com mesmo nome do método.

**Configurando o Entity Framework**

**A difença do API, é que agora temos uma tela mvc para fazer a interação**

**dotnet add package Microsoft.EntityFrameworkCore.sqlServer**

**dotnet add package Microsoft.EntityFrameworkCore.Design**

**em program.cs**

**importar**

using Microsoft.EntityFrameworkCore;

using MVC.Context;

**criar migrations**

**dotnet ef migrations add AdicionaTabelaContato**

para aplicar migrations no servidor sql/ criar banco de dados

dotnet ef database update