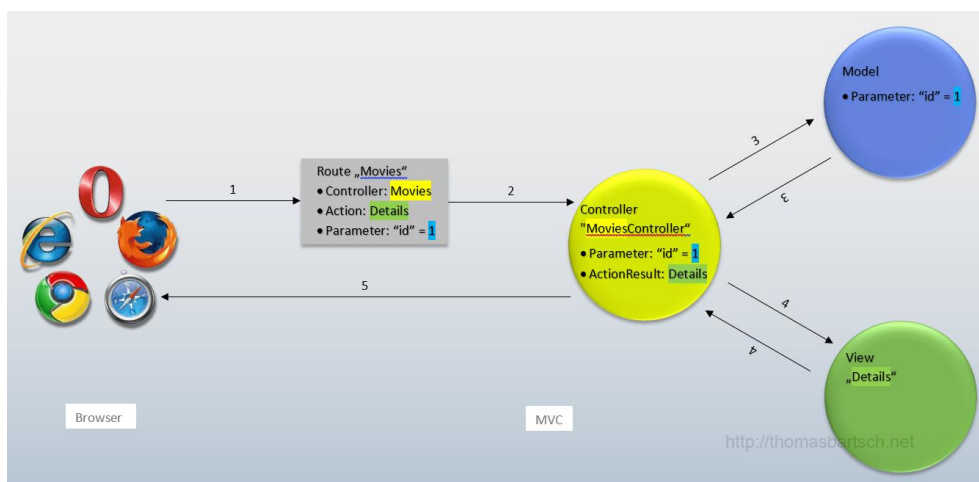
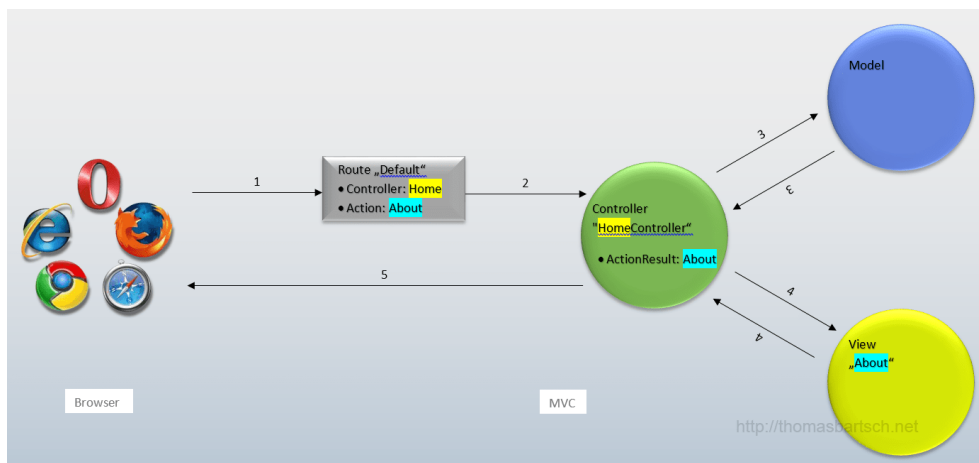
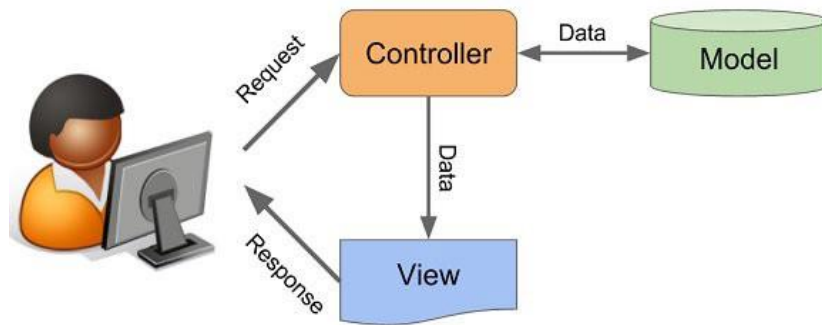


Objetivo:

- Criar um projeto Web ASP.NET Core MVC
- Configurar a aplicação
- Adicionar Controller

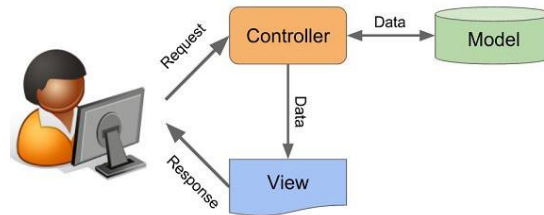


Criar aplicação ASP.NET Core.....	3
Adicionar um Controller.....	3

Criar aplicação ASP.NET Core

1. Criar o projeto ASP.NET Core baseado no Template MVC (01-ASP.NET Core- Conceitos fundamentais) ou Empty (02-ASP.NET Core-Projeto de raiz)

Adicionar um Controller

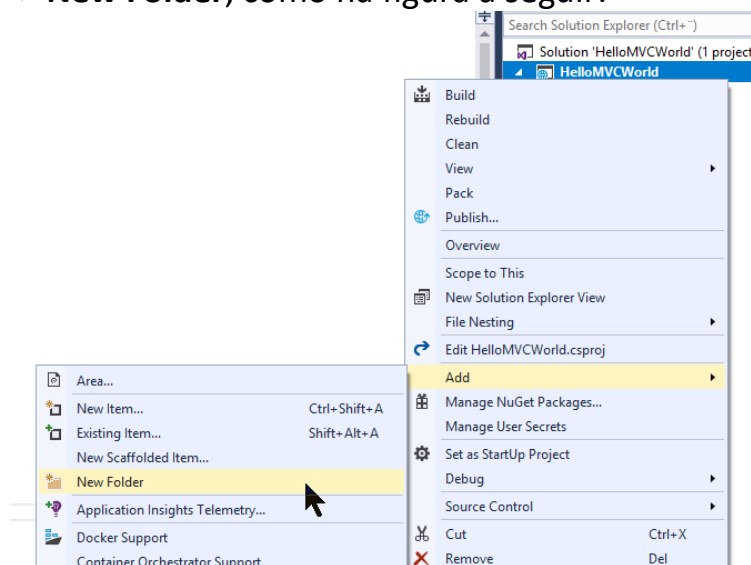


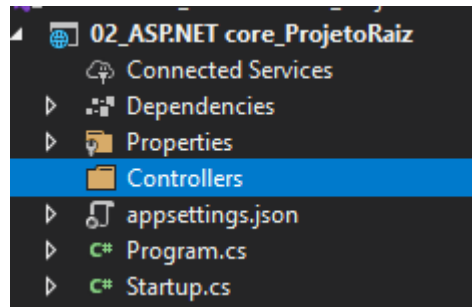
Controller (controlador)

- Faz a mediação da entrada (requisição do utilizador, browser), convertendo-a em comandos para o **Model** ou **View**.
- Recebe pedidos, executa operações no **Model** e retorna Views para o Browser

2. Criar a pasta *Controllers*

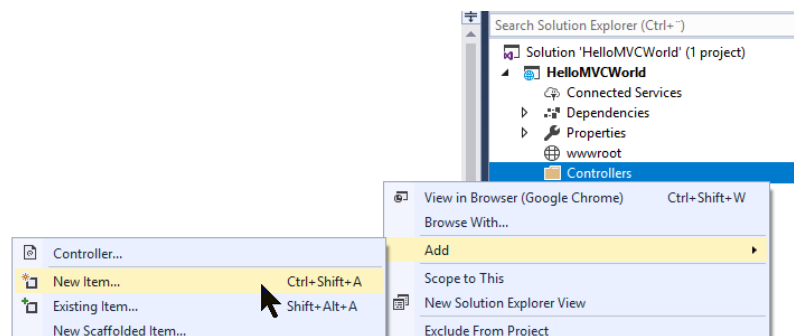
Em Solution Explorer, sobre o nome do projeto, fazer Tecla direita do rato e seleccionar **Add -> New Folder**, como na figura a seguir:





3. Adicionar a classe de uma novo *Controller*

Em Solution Explorer, sobre o nome da pasta “*Controllers*”, fazer Tecla direita do rato e seleccionar **Add** -> **New Item** ou **Controller**, como na figura a seguir:



Selecionar a opção “**MVC Controller - Empty**”

Este primeiro *controller* terá o nome de **HomeController**

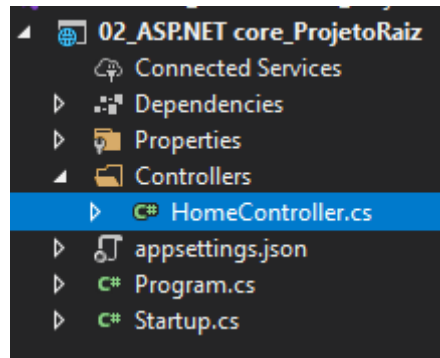
Nomes dos **Controllers**

- Os nomes dos *controllers* deve ter o como sufixo a expressão “**Controller**”.
- Sintaxe: Nome**Controller**

Exemplo:

HomeController

ClientesController



HomeController.cs

```
using Microsoft.AspNetCore.Mvc;

namespace _02_ASP.NET_core_ProjetoRaiz.Controllers
{
    public class HomeController : Controller
    {
        public IActionResult Index()
        {
            return View();
        }
    }
}
```

O controlador *HomeController.cs* define uma classe típica de C# que herda a classe *Controller* e um método *Index* (ação) e retorna uma *View*.

Para aceder via URL: Home/Index

<https://localhost:44304/Home/Index>

ou, como é que está definido como *default*

<https://localhost:44304>

Controller

```
public class NomeController : Controller
{
    public IActionResult Metodo()
    {
        return View();
    }
}
```

URL

<https://www.site.pt/Nome/Metodo>

Exemplo:

<https://www.site.pt/Clientes/Adicionar>

Uma vez que ainda não temos a View correspondente, vamos alterar o código do método Index() para retornar outro conteúdo diretamente:

```
public IActionResult Index()
{
    return Content($"Teste de Controller no dia {DateTime.Now}");
    // return View();
}
```

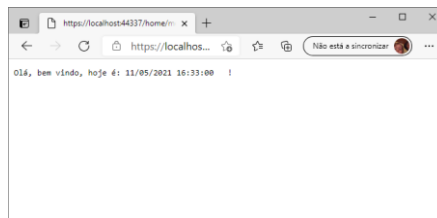
Executar a aplicação



Adicionar o método mensagem():

<https://localhost:44304/Home/Mensagem>

```
public string Mensagem()
{
    return $"Olá, bem vindo, hoje é: {DateTime.Now}  !";
}
```



Adicionar o método CriaHtml():

<https://localhost:44304/Home/CriaHtml>

```
public ContentResult CriaHtml()
{
    ContentResult content = new ContentResult();

    content.ContentType = "text/html";
    content.Content = "<html><body><h1>Bem Vindo!</h1><p>" + DateTime.Now + "</p></body></html>";

    return content;
}
```

