Programação Web

Aulas Teóricas – Capítulo 2 – 2.2

1º Semestre - 2023/2024

Departamento de Engenharia Informática e de Sistemas
Instituto Superior de Engenharia de Coimbra/Instituto Politécnico de Coimbra

Programação Web

ASP.NET Core

Departamento de Engenharia Informática e de Sistemas
Instituto Superior de Engenharia de Coimbra/Instituto Politécnico de Coimbra

ASP.NET

- Asp.NET Webforms
  - Utiliza a abordagem *Page controller*, no qual cada página tem o seu próprio controlo, i.e., ficheiro *code-behind* que processa o request
- ASP.MVC
  - Utiliza a abordagem *front controler*, no qual um controlo comum para todas as paginas processa o *request*
- …

Programação Web – 2023/2024 – LEI D+CE+PL – ISEC/IPC @JB

Model-View-Controller

- *MVC* é um *architectural pattern*
  - Desenvolvido nos anos 70 para aplicações *desktop*, inicialmente deignado como *Thing-Model-View-Editor*
  - Amplamente adotado como arquitetura para desenvolvimento de aplicações web
- Existem *frameworks* MVC para várias linguagens
  - Java (ex. Spring Framework)
  - Ruby (Ruby on Rails)
  - .NET (com introdução do ASP.NET MVC em 2007)
  - Várias *frameworks* para clientes Javascript (ex. Angular)

Programação Web – 2023/2024 – LEI/LEC/LECE+PL – ISEC/IPC @JB

isec Engenharia DEIS - Instituto Superior de Engenharia de Coimbra

Do ASP.NET Webforms para MVC

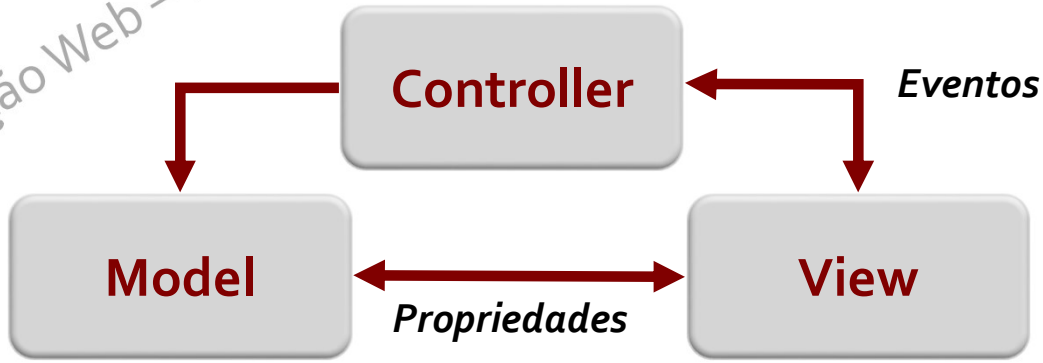
- Remoção do Code-behind para as vistas
- Remoção da *ViewState*
- Remoção do suporte ao *server-side control*
- Adição de *Model Binding*
- Adição de *Routing*
- Adição do *Razor View Engine (a partir do MVC 3)*

Programação Web (2023/2024) - LEI DEIS+PL – ISEC/IPC @ JB

Padrão MVC – *Model View Controller*

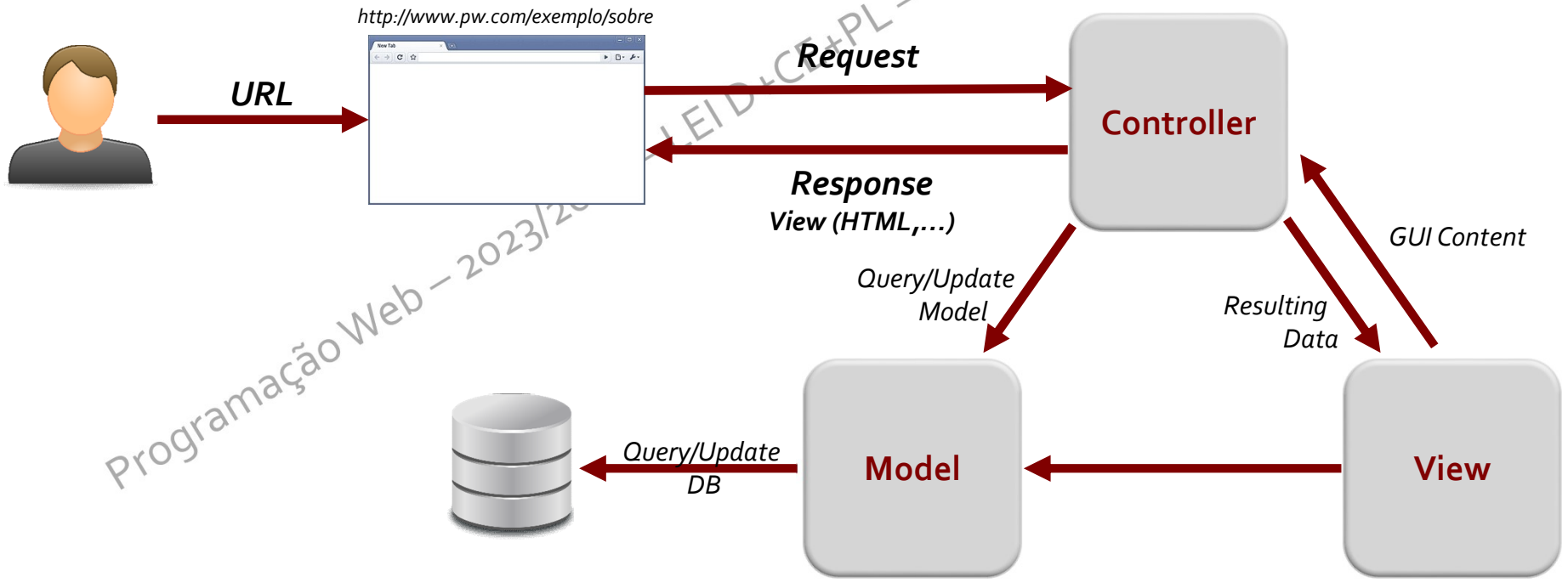
- No ASP.Net Core podem-se criar aplicações com a chamadas Razor Pages ou de acordo com o padrão MVC – iremos dar ênfase a esta 2ª abordagem
  - Na 2ª abordagem utiliza-se o padrão MVC para se fazer a separação do UI de uma aplicação em três aspetos

Programação Web – 2023/2024 – LEI+LEI-PL+CE+RC – ISEC/DEIS/RC @JR



ASP.NET Core MVC

- Fluxo de um pedido em ASP.NET MVC.

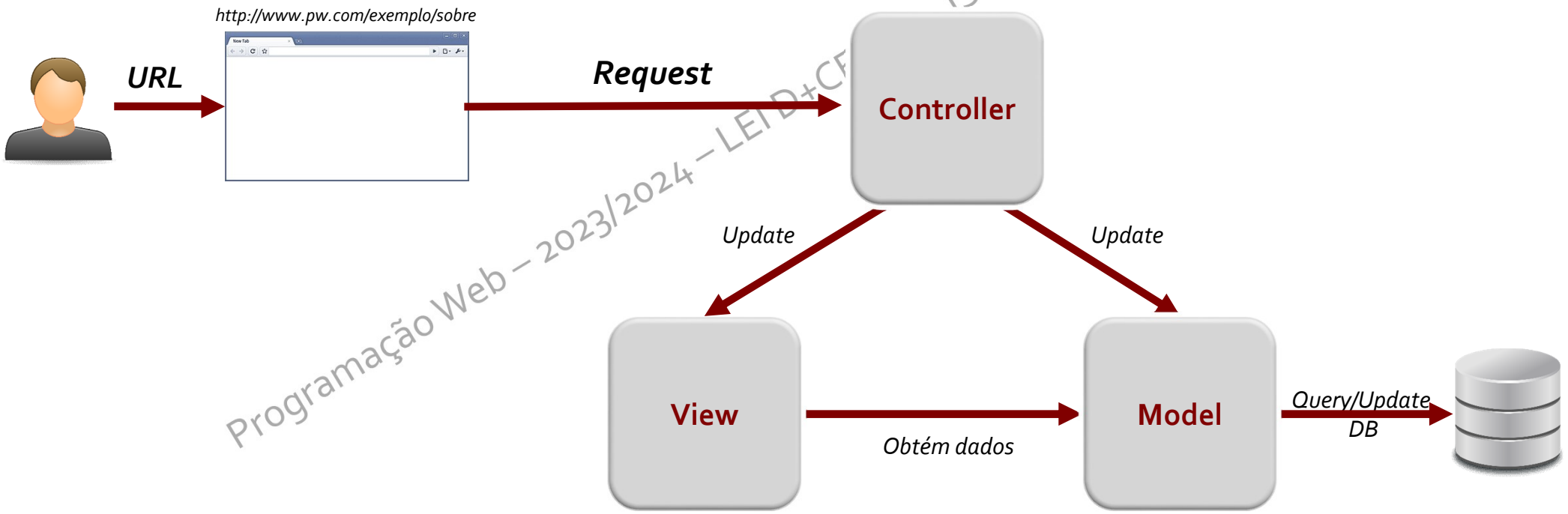


Programação Web – 2023/2024 – LEI D-CE+PL – ISEC/IPC @JB

isec Engenharia DEIS - Instituto Superior de Engenharia de Coimbra

ASP.NET Core MVC

- Fluxo de um pedido em ASP.NET MVC.

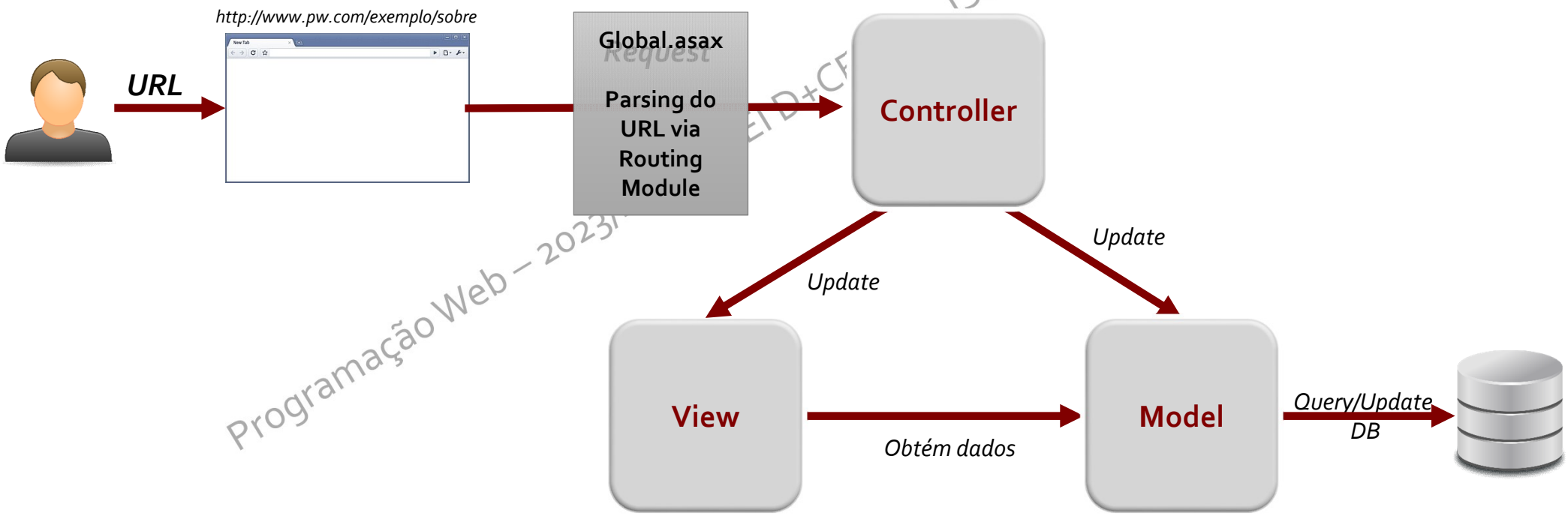


Programação Web – 2023/2024 – LEI D+CF ISEC/IPC @JB

isec Engenharia DEIS - Instituto Superior de Engenharia de Coimbra

ASP.NET Core MVC

- Fluxo de um pedido em ASP.NET MVC.



Programação Web – 2023/24 ISEC/IPC @JB

isec Engenharia DEIS - Instituto Superior de Engenharia de Coimbra

Model-View-Controller

- Permite o desenvolvimento de aplicações mais flexíveis;
- Separação de responsabilidades entre:
  - a **vista**
  - o **modelo**
  - o **controlador**

Programação Web – 2023/2024 – LEI D+CE PL – ISEC/IPC @ 23

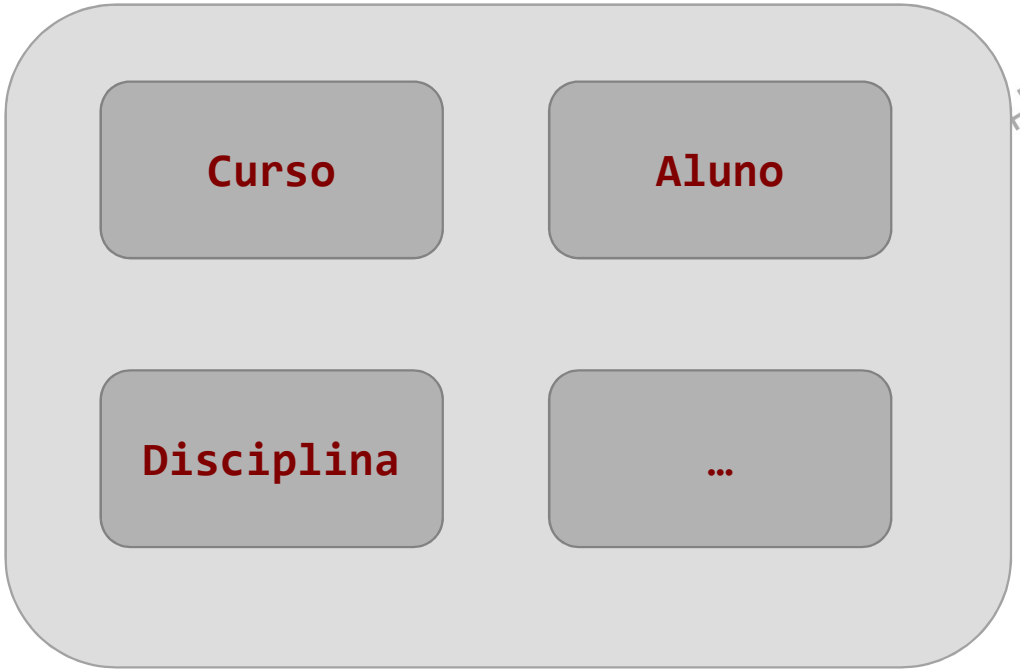
ASP.NET Core MVC - *Model*

- Domínio dos dados e comportamento da aplicação, independente do UI
- Tipicamente representados por ***P**lain **O**ld **C**LR **O**bjects* (**POCOs**)
- O modelo de classes tem validação *built in* e, depende da *framework* Javascript do lado cliente utilizada (como por exemplo a knockout.js)

Programação Web – 2023/2024 – LEI D+CR+PL – ISEC/IPC @ls

ASP.NET Core MVC - *Model*

- Model



PL – ISEC/IPC @JB

```
namespace MVC1.Models
{
    public class Aluno
    {
        public int Id { get; set; }
        public string Nome { get; set; }
        public string Mail { get; set; }
    }
}
```

isec Engenharia DEIS - Instituto Superior de Engenharia de Coimbra

ASP.NET Core MVC – *View*

- Interface do utilizador para apresentação dos dados, isto é, representa o UI da aplicação;
- O HTML que se apresenta ao utilizador;

```
@{
    ViewBag.Title = “Aluno";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h2>Aluno</h2>
```

Programação Web (2023/2024) – LEI D+CE+PL – ISEC/IPC @JB

ASP.NET Core MVC - *Controller*

- Traduz as ações do utilizador em operações apropriadas;
- **"Cérebro" da operação**, responsável por tratar o pedido HTTP
- *Controllers* (assim como os modelos e vistas) devem ser leves e deixar os outros componentes manter separação de responsabilidades/interesses.

Programação Web – 2023/2024 – LEI D+CE+PL – ISEC/IPC @ 2

ASP.NET Core

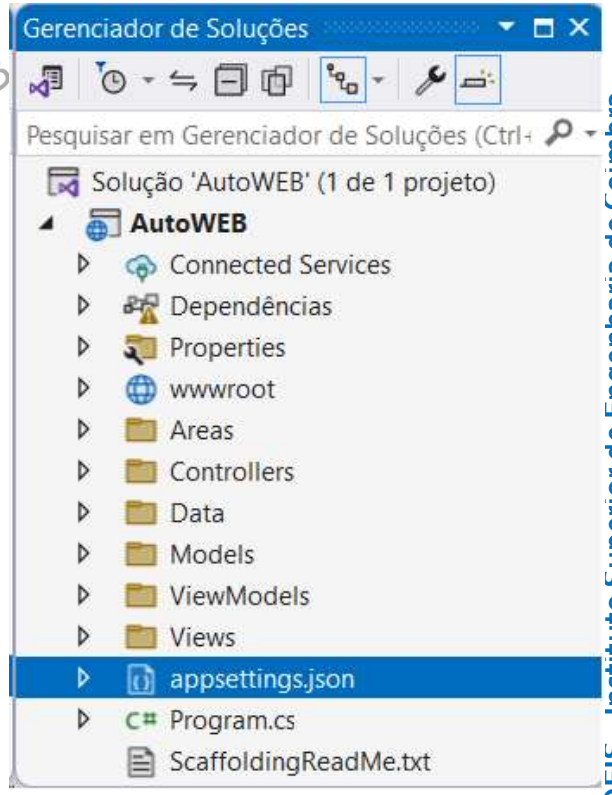
- Convenções de nomes
  - Simplificam a leitura de código entre varios elementos de uma equipa de desenvolvimento;
  - Evita que os programadores tenham de configurar e especificar elementos que podem ser inferidos por convençao:
- Exemplo:
  - Nome para a estrutura de directórios para as vistas (permite a omissão do caminho concreto quando se quer referenciar uma vista numa class controller)

Programação Web – 2023/2024 – LEILD+CE+PL – ISEC/IPC @JB

isec Engenharia DEIS - Instituto Superior de Engenharia de Coimbra

Estrutura ASP.N Core

- Estrutura de diretórios
  - *wwwroot* – Ficheiros CSS, imagens, entre outros elementos do site
  - *Data* – pode-se usar para as Migrations e o Contexto da Base de Dados
  - *Models – Classes de domínio referentes às entidades*
  - *Controllers* - Controladores
  - *Views* – Classes das views da aplicação (Razor)

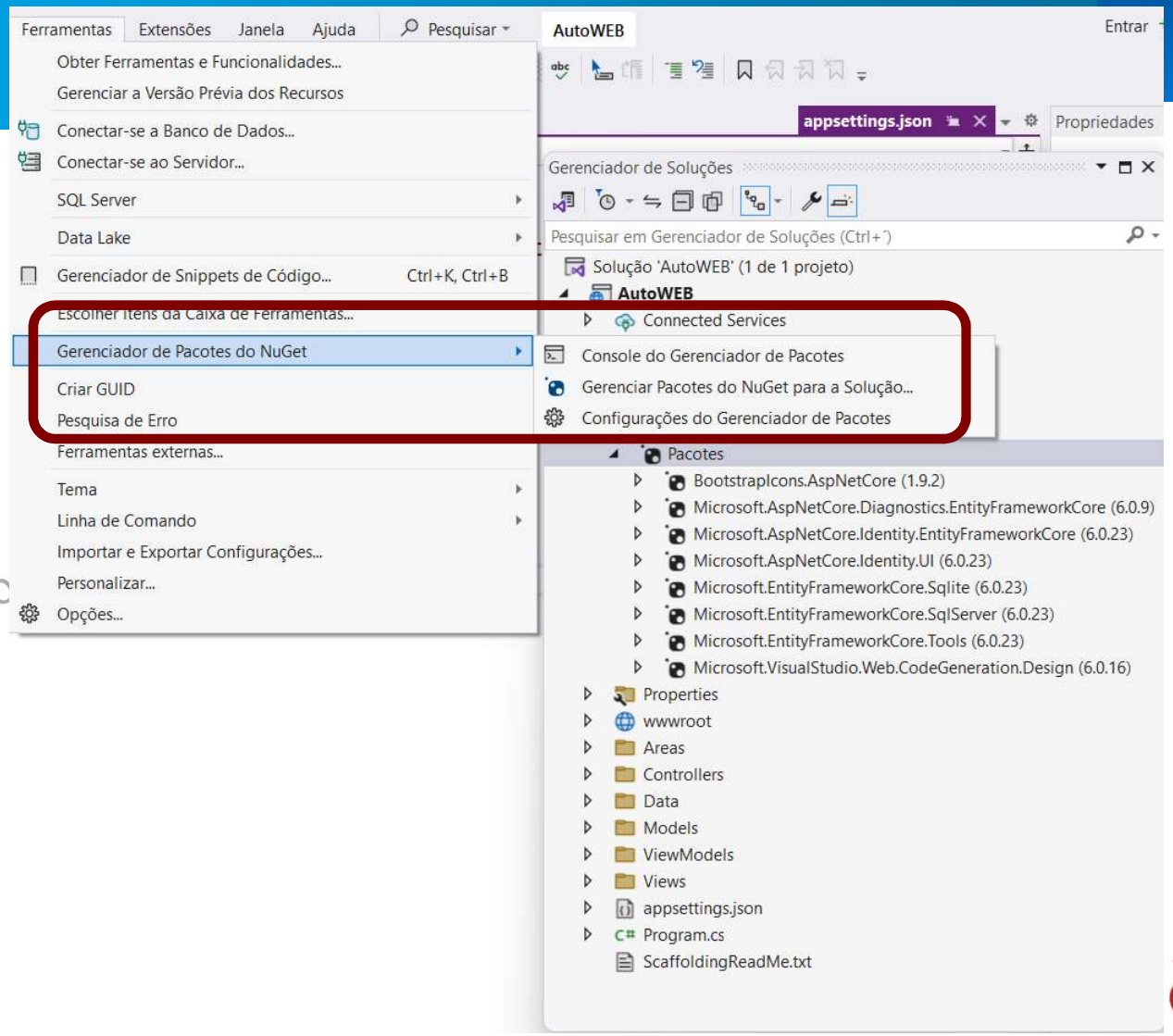


Programação Web – 2023/2024 – LEI+CE+PL – ISEC/IPC @JB

isec Engenharia DEIS - Instituto Superior de Engenharia de Coimbra

Pacotes

- Utilizado pelo *NuGet package manager*
  - Permite gerir as dependências da aplicação



Programação Web – 2023/20

isec Engenharia DEIS - Instituto Superior de Engenharia de Coimbra

Outros ficheiros de Configuração

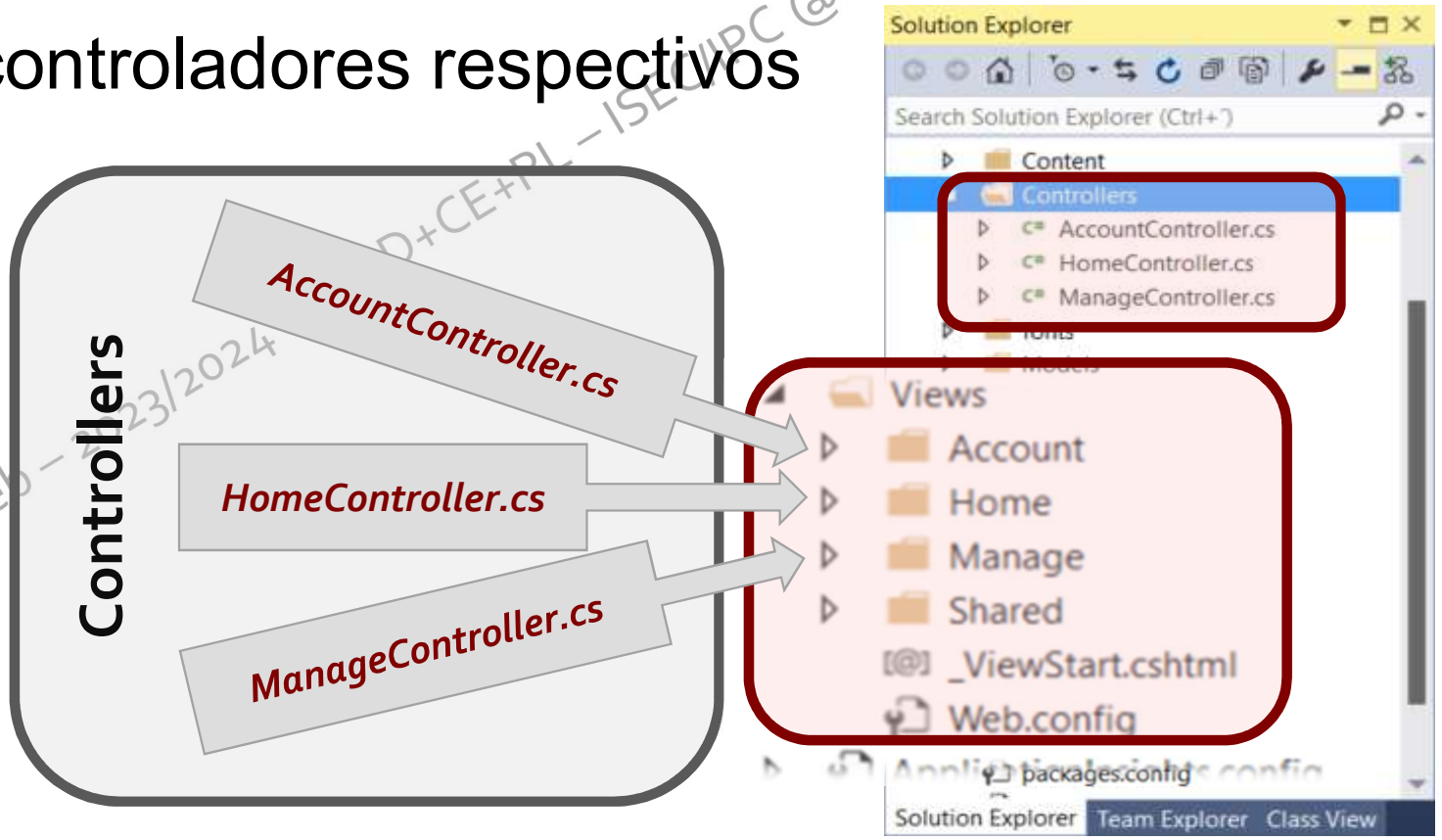
- Program.cs
  - Utilizado na configuração e inicialização da aplicação
  - Nota: No .Net 6, só existe este ficheiro para as configurações

Programação Web – 2023/2024 – LEI D+CF+PL – ISEC/IPC @JB

Views

- Os ficheiros das Views são colocados em Pastas com o nome dos controladores respectivos

Convenções!

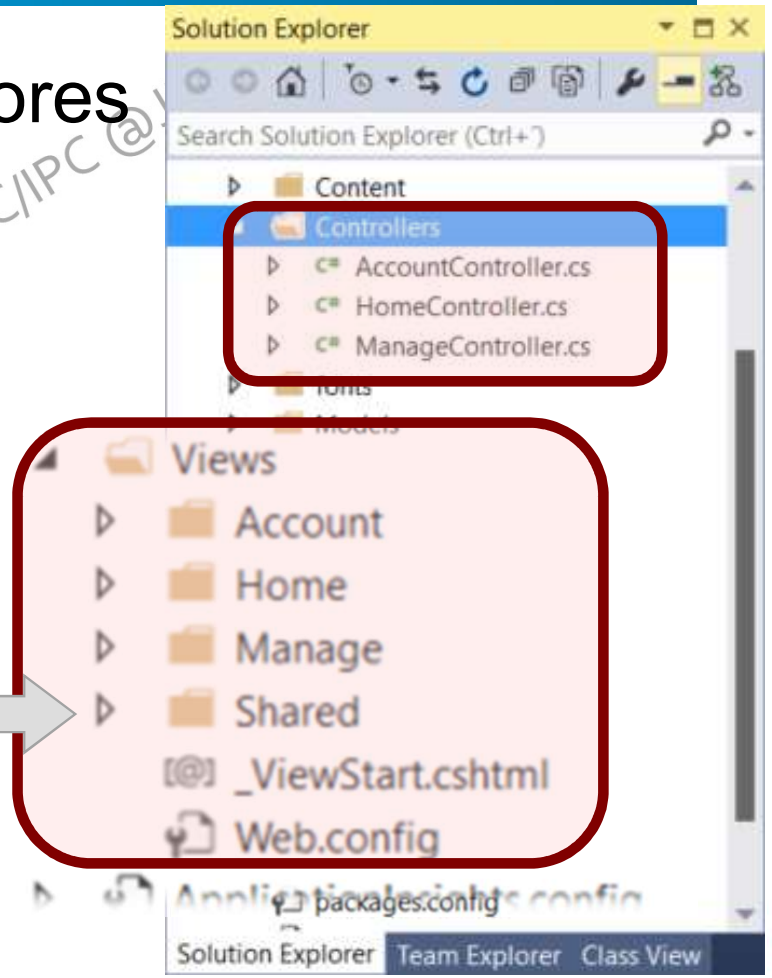


Programação Web – 2023/2024 D+CE+PL – ISEC/UPC @JP

isec Engenharia DEIS - Instituto Superior de Engenharia de Coimbra

Views (2)

- Pastas com o nome dos controladores



Shared
Vista utilizada por todos os controladores

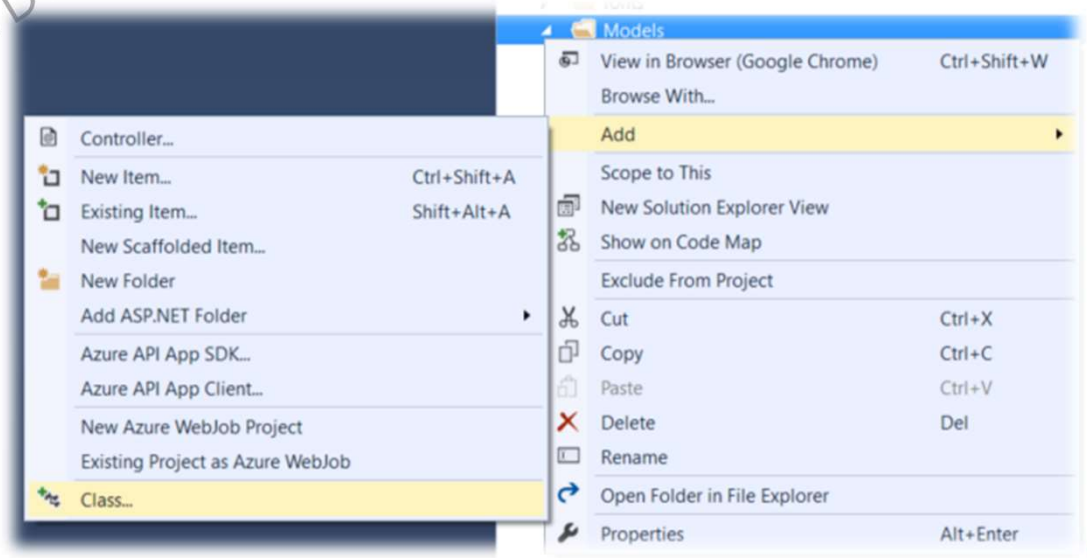
Pr – 2023/2024 – LEI D+CE+PL – ISEC/IPC @

isec Engenharia DEIS - Instituto Superior de Engenharia de Coimbra

Criação de um Model

- **Criação de Modelo**
  - Classe Aluno

```
namespace AutoWeb.Models
{
    public class Aluno
    {
        public int Id { get; set; }
        public string Nome { get; set; }
        public string Mail { get; set; }
    }
}
```

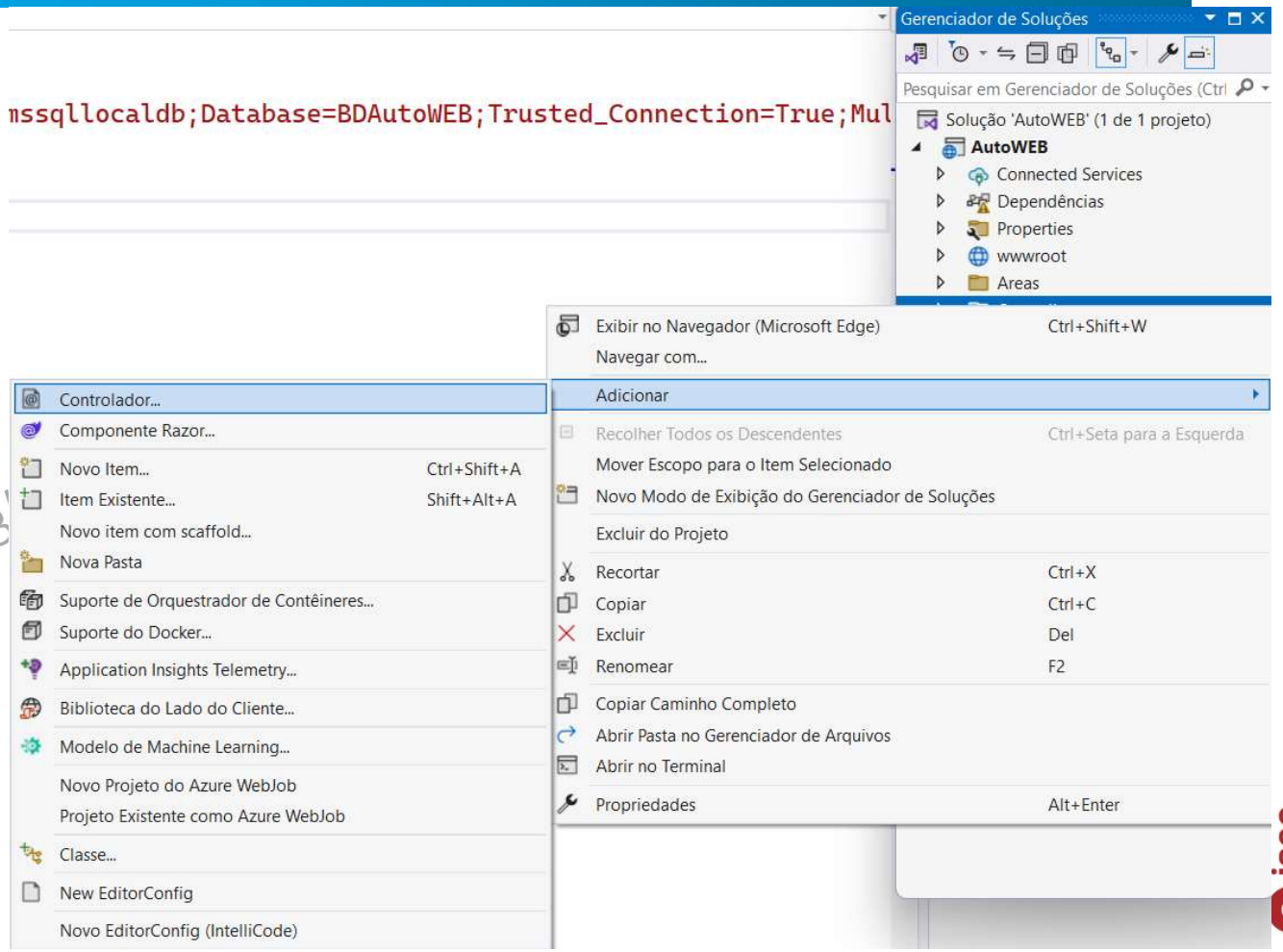


Programação Web – 2023/2024 – LEI D+CE+PL – ISEC/IPC @JB

isec Engenharia DEIS - Instituto Superior de Engenharia de Coimbra

Criação de um Controler

- Adicionar um controlador - 1

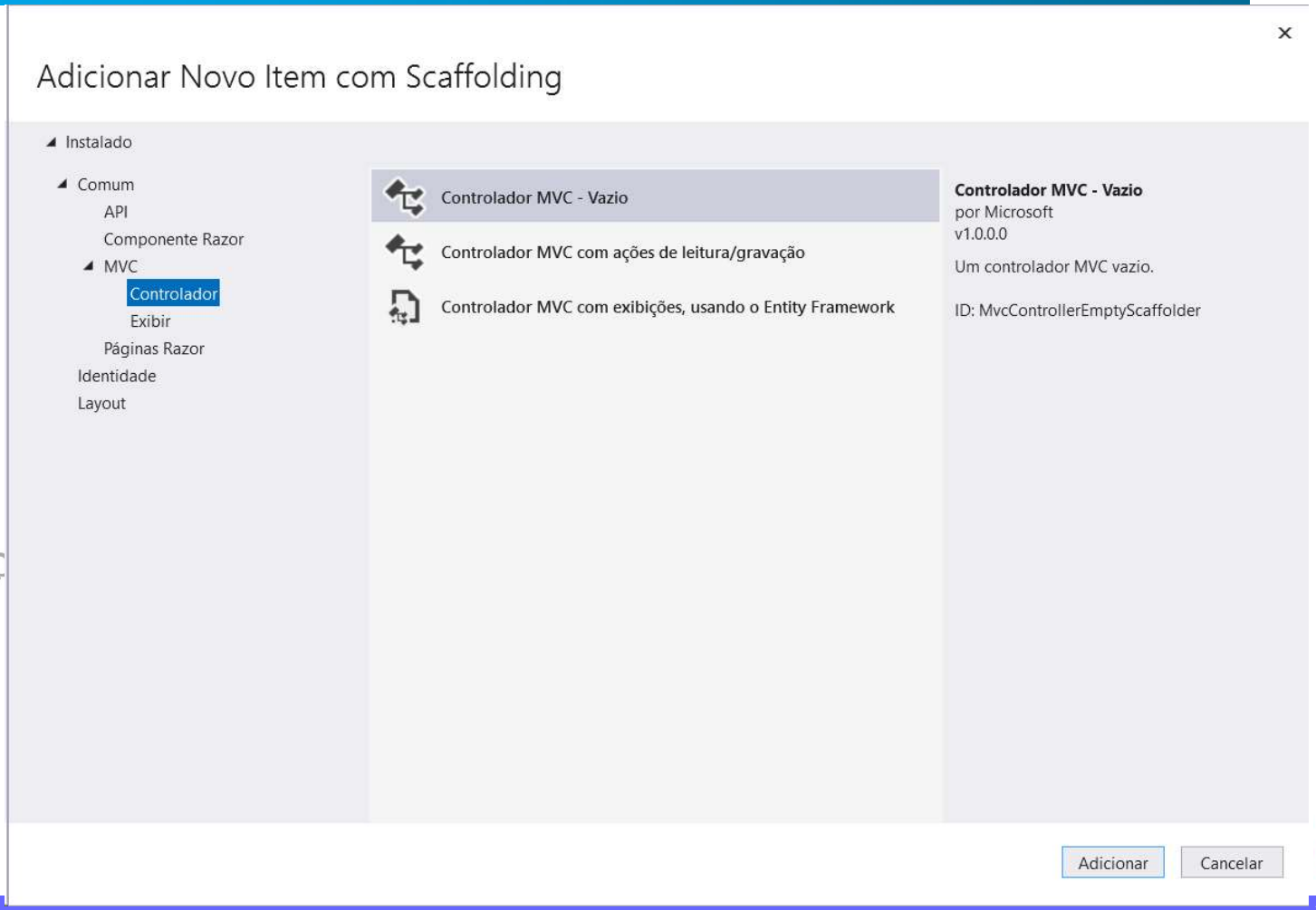


Programação Web – 2023

isec Engenharia DEIS - Instituto Superior de Engenharia de Coimbra

Criação de um Controler

- Adicionar um controlador - 2

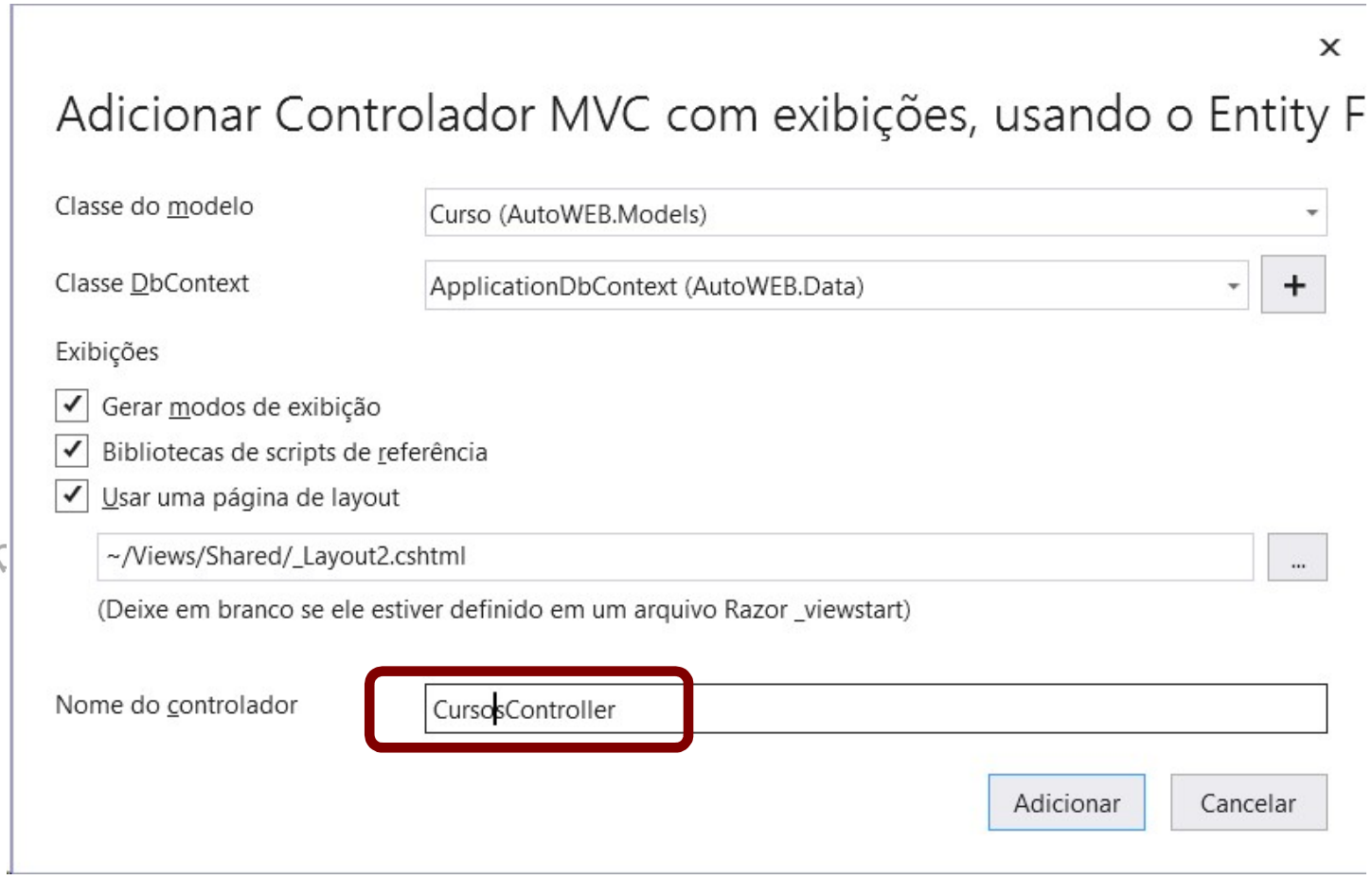


Programação Web

isec Engenharia DEIS - Instituto Superior de Engenharia de Coimbra

Criação de um Controler

- Adicionar um controlador - 3



Programação Web

isec Engenharia DEIS - Instituto Superior de Engenharia de Coimbra

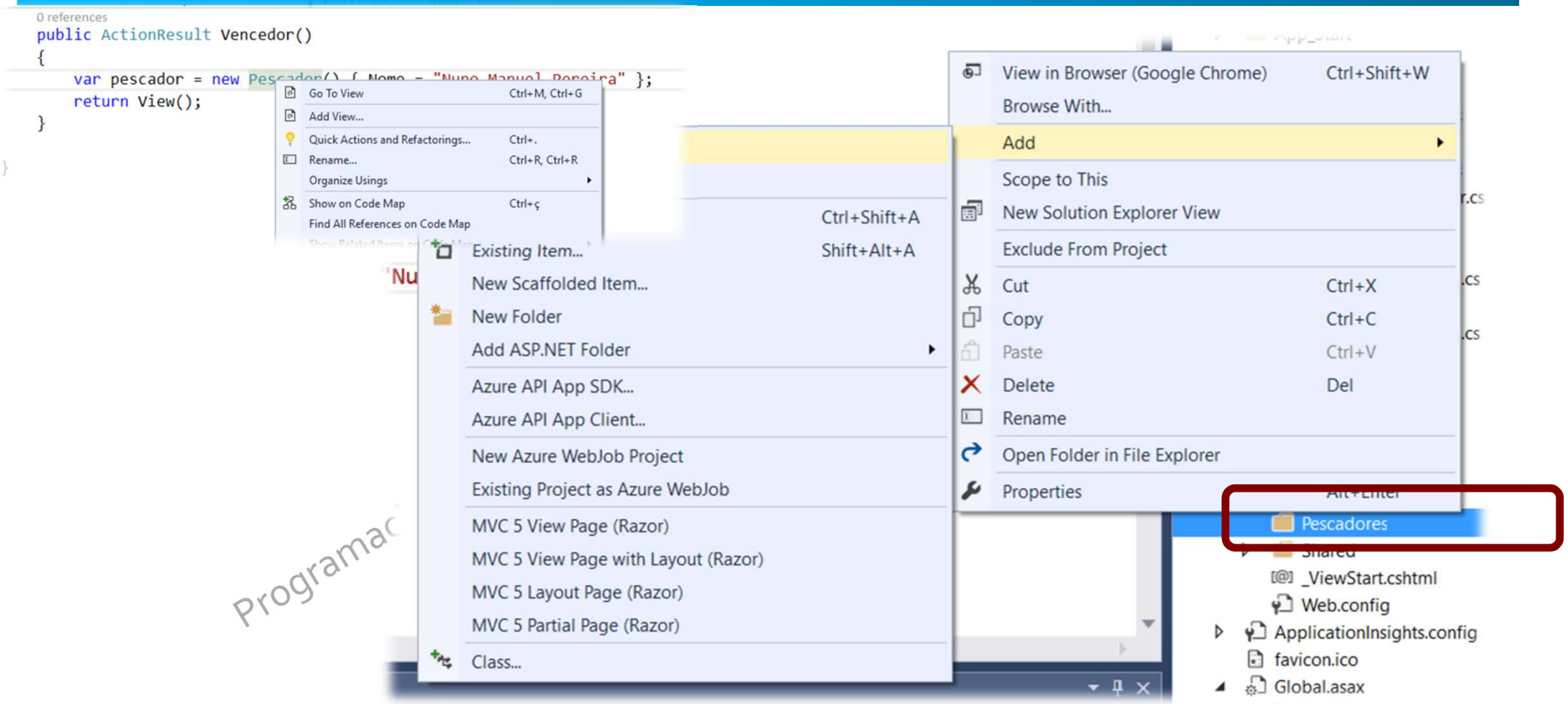
Criação do *Controller*

```
using AutoWEB.Models;
using AutoWEB.ViewModels;
using Microsoft.AspNetCore.Authorization;

namespace AutoWEB.Controllers
{
    1 referência
    public class CursosController : Controller
    {
        private readonly ApplicationDbContext _context;

        0 referências
        public CursosController(ApplicationDbContext context)
        {
            _context = context;
        }

        // GET: Cursos
        3 referências
        public async Task<IActionResult> Index(bool? disponivel)
        {
            ViewData["ListaDeCategorias"] = new SelectList(_context.Cate

            if (disponivel != null)
            {
                if (disponivel == true)
```

Programação Web – 2023/

isec Engenharia DEIS - Instituto Superior de Engenharia de Coimbra

Criação de uma View



Programac

isec Engenharia DEIS - Instituto Superior de Engenharia de Coimbra

Action Methods – Class Controller

- Todos os métodos públicos da classe *Controller* são designados como ***Action methods***

CE+PL – ISEC/IPC @ja

```
// POST: Cursos/Create
// To protect from overposting attacks, enable the specific properties you want to bind to.
// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
[Authorize(Roles = "Admin")]
0 referências
public async Task<IActionResult> Create([Bind("Id,Nome,Descricao,DescricaoResumida,Requisitos,IdadeMinima,Disponivel,Preco,EmDestaque,CategoriaId")] Curso curso)
{
    ModelState.Remove(nameof(curso.categoria));

    if (ModelState.IsValid)
    {
        _context.Add(curso);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    ViewData["ListaDeCategorias"] = new SelectList(_context.Categorias.OrderBy(c => c.Disponivel).ToList(), "Id", "Nome", curso.CategoriaId);

    return View(curso);
}
```

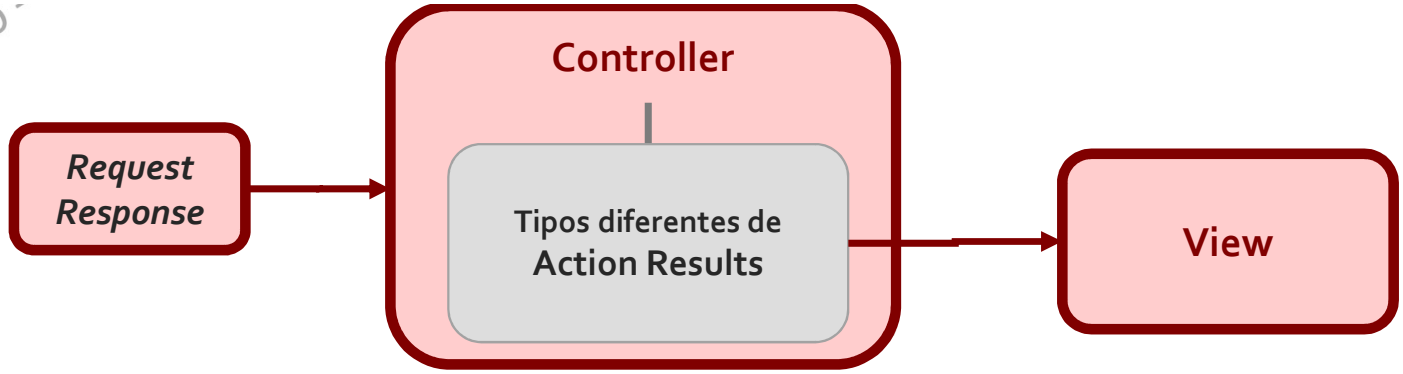
- Devem ser públicos
- Não podem ser *overloaded*
- *Não podem ser estáticos*

isec Engenharia DEIS - Instituto Superior de Engenharia de Coimbra

Action Methods - ActionResult

```
// POST: Cursos/Create
// To protect from overposting attacks, enable the specific properties you want to bind to.
// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
[Authorize(Roles = "Admin")]
0 referências
public async Task<IActionResult> Create([Bind("Id,Nome,Descricao,DescricaoResumida,Requisitos,IdadeMinima,Disponivel,Preco,EmDestaque,CategoriaId")] Curso curso)
{
    ModelState.Remove(nameof(curso.categoria));

    if (ModelState.IsValid)
    {
        _context.Add(curso);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    ViewData["ListaDeCategorias"] = new SelectList(_context.Categorias.OrderBy(c => c.Disponivel).ToList(), "Id", "Nome", curso.CategoriaId);

    return View(curso);
}
```

Retorna uma ***ActionResult*** *usando o método* ***View()*** *definido na* ***class Controller***



Action Results

Action Result	Helper Method
ViewResult	View()
PartialViewResults	PartialView()
ContentResult	Content()
RedirectResult	Redirect()
RedirectToRouteResult	RedirectToAction()
JsonResult	Json()
FileResult	File()
HttpNotFountResult	HttpNotFound()
EmpyResult	

https://msdn.microsoft.com/en-us/library/system.web.mvc.actionresult(v=vs.118).aspx

GET vs POST vs PUT vs DELETE

- **GET**: Método genérico para qualquer pedido que **obtém** dados do servidor;

```
[HttpGet]
public ActionResult Edit(int id)
{

}
```

```
@Html.ActionLink("Editar", "Edit", new { id = item.Id })
```

ISEC/IPC

GET vs POST vs PUT vs DELETE

- **POST**: Método genérico para qualquer pedido que **envia** dados ao servidor;

```
@using (Html.BeginForm("Edit", "Exemplo", FormMethod.Post))
```

```
 [HttpPost]
 public ActionResult Edit(Disciplina disciplina) {
...
}
```

PL – ISEC/IPC @ 23/2024

Action Parameters – Parameter Binding

- As *actions* podem ter parâmetros de entrada (tipos de dados primitivos ou complexos), provenientes de:
  - *URL*
  - *Query String*
  - *Formulário de dados*
- Os parâmetros podem ser do tipo *Nullable*.

o Web – 2023/2024 – LEI D+CE+PL – ISEC/IPC @JB