

## PROGRAMAÇÃO VISUAL CTESP/TPSI

### Introdução à Programação Web com ASP.NET + MVC Operações CRUD

Prof. José Brás

F262 – jose.bras-perreira@estsetubal.lps.pt

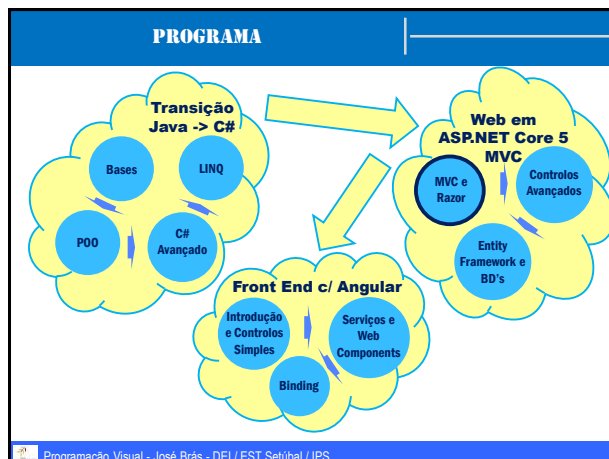
Departamento de Engenharia Informática

Escola Superior de Tecnologia de Setúbal – Instituto Politécnico de Setúbal

2025/26



1



2

#### 5.3 OPERAÇÕES CRUD

#### 5. .NET CORE WEB

##### ■ 5.3 CRUD

- Create / Read / Update / Delete

##### ■ 5.3.1 Preparação do projeto:

- Projeto: ASP.NET Core WebApp (Model-View-Controller)
- Modelo: Foto com título e Código (Guid ?)
  - Criar lista de fotos com uma classe static (classe static?)
- Controlador: FotosController
- Vista: Index numa pasta Fotos da pasta Views

##### ■ 5.3.2 Create

- View Create
- Action Create

##### ■ 5.3.3 Read

- View Details
- Action Details

##### ■ 5.3.4 Update

- View Edit
- Action Edit

##### ■ 5.3.5 Delete

- View Delete
- Action Delete

##### ■ 5.3.6 Vista Parcial p/ Procura

- Partial View
- Action Index

Programação Visual - José Brás - DEI / EST Setúbal / IPS

3

3

#### 5.3.1 MODEL -> FOTO

#### 5. .NET CORE WEB

#### 5.3 - OPERAÇÕES CRUD

public class Foto\_NA

```
{
    public Guid Id { get; set; }
    public String Titulo { get; set; }
    public decimal Preco { get; set; }

    public Foto_NA(String titulo, decimal preco)
    {
        Titulo = titulo;
        Preco = preco;
        Id = Guid.NewGuid();
    }
}
```



1. No Solution Explorer BDR sobre a pasta Models -> Add -> Class -> Foto -> btn Add

2. Adicionar propriedades Id (Guid), Título e Preço

Programação Visual - José Brás - DEI / EST Setúbal / IPS

4

4

#### 5.3.1 GUID

#### 5. .NET CORE WEB

#### 5.3 OPERAÇÕES CRUD

##### ■ 5.3.1 Guid

- Em C# a struct Guid é um *built-in type* do namespace System que representa um identificador único universal.
- O Guid é um valor inteiro de 128-bit integer (16 bytes) que pretende ser único para cada Sistema.
- Há 2 elevado a 128 possíveis combinações (valores)
- Geramos um novo identificador único com o método de classe:

**Guid.NewGuid()**

Programação Visual - José Brás - DEI / EST Setúbal / IPS

5

5

#### 5.3.1 MODEL -> SEED

#### 5. .NET CORE WEB

#### 5.3 - OPERAÇÕES CRUD

1. Criar uma classe static SeedFotos numa pasta Data



```
public static class SeedFotos
{
    public static List<Foto_NA> Seed() => new()
    {
        new Foto_NA("Deserto", 1.1m),
        new Foto_NA("Koala", 1.2m),
        new Foto_NA("Tulips", 1.3m)
    };
}
```

2. Criar um método static Seed que retorne uma Lista de Fotos

Programação Visual - José Brás - DEI / EST Setúbal / IPS

6

6

### 5.3.1 CLASSE STATIC

#### 5.3.1 Classe static

- ❑ **Não pode ser instanciada:**
  - Acedemos aos membros com o nome da classe **NomeDaClasse.NomeDoMembro;**
- ❑ **Só tem membros static:**
  - Todos os membros são de classe.
- ❑ **Não tem construtores de instância:**
  - Mas pode ter um construtor static, chamado uma única vez quando a classe é carregada.
- ❑ **É sealed:**
  - O que significa que não pode ser herdada.
- ❑ **Carregada automaticamente:**
  - O .NET runtime carrega a classe estática na memória quando é referida pela primeira vez.
- ❑ **Uso comum:**
  - Ideal para classes que fornecem funcionalidades mas não mantêm um estado único por instância, como cálculos matemáticos, manipulação de strings ou logging.

Programação Visual - José Brás - DEI / EST Setúbal / IPS

7

### 5.3.1 CONTROLADOR FOTOSCONTROLLER

#### 5.3 - OPERAÇÕES CRUD

- 1. BDR sobre a pasta Controllers -> Add -> Controller**
- 2. Escolher MVC Controller - Empty**
- 3. Premir Add e ...**
- 4. Escolher MVC Controller - Empty**
- 5. Nome: FotosController.cs**
- 6. Premir Add e ...**

Programação Visual - José Brás - DEI / EST Setúbal / IPS

8

### 5.3.1 CONTROLADOR FOTOSCONTROLLER

#### 5.3 - OPERAÇÕES CRUD

```
using A18_Slides_Operacoes_CRUD.Data;
using A18_Slides_Operacoes_CRUD.Models;
using Microsoft.AspNetCore.Mvc;

namespace A18_Slides_Operacoes_CRUD.Controllers
{
    public class FotosController : Controller
    {
        private static List<Foto_NA> fotos = SeedFotos.Seed();

        public IActionResult Index()
        {
            return View(fotos);
        }
    }
}
```

- 1. Inserir os usings requeridos**
- 2. Criar um campo (atributo) static fotos e Inicializá-lo usando o método Seed da classe SeedFotos**
- 3. Passar o campo (atributo) fotos para a view da ação Index como argumento**

Programação Visual - José Brás - DEI / EST Setúbal / IPS

9

### 5.3.2 VIEW INDEX C/ CRUD

#### 5.3 - OPERAÇÕES CRUD

- 1. BDR sobre Views -> Add -> new Folder -> Fotos**
- 2. BDR sobre Fotos -> Add -> View -> Razor View -> Add**
- 3. View name = Index Template = List Model Classe = Foto\_NA Use layout page -> prima [...]**
- 4. Navegue Views -> Shared -> Selecione \_Layout.cshtml -> OK -> Add -> wait :-)**

Programação Visual - José Brás - DEI / EST Setúbal / IPS

10

### 5.3.1 VIEW INDEX C/ CRUD

#### 5.3 - OPERAÇÕES CRUD

localhost:5197/Fotos

A18\_Slides\_Operacoes\_CRUD Home Privacy

**Create**

**Read**

Id	Título	Preço	
c9bc0cc7-2fce-47b4-827f-96adcc9645ba	Deserto	1.10	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
0e3fe16d-9d83-4b23-a5e1-255fd1792a94	Koala	1.20	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
d198715d-2674-44e2-99c2-e2b17666ba7f	Tulips	1.30	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>

© 2023 - A18\_Slides\_Operacoes\_CRUD - Privacy

**Update**

**Delete**

- 1. Run -> já temos a lista com os links para as operações CRUD**

Programação Visual - José Brás - DEI / EST Setúbal / IPS

11

### 5.3.2 VIEW CREATE

#### 5.3 - OPERAÇÕES CRUD

- 1. Expandir a pasta Fotos**
- 2. BDR sobre a pasta Fotos -> Add -> View -> Razor View -> Add**
- 3. View name = Create Template = Create Model Classe = Foto\_NA Use layout page -> prima [...]**
- 4. Navegue Views -> Shared -> Selecione \_Layout.cshtml -> OK -> Add -> wait :-)**

Programação Visual - José Brás - DEI / EST Setúbal / IPS

12

### 5.3.2 VIEW -> CREATE

#### 5. .NET CORE WEB

##### 5.3 - OPERAÇÕES CRUD

1. Apagamos o código que mostra o Id da Fotografia

2. E abrimos o ficheiro FotosController.cs para adicionar 2 (duas) ações Create

Programação Visual - José Brás - DEI / EST Setúbal / IPS

13

### 5.3.2 ACTION -> CREATE

#### 5. .NET CORE WEB

##### 5.3 - OPERAÇÕES CRUD

```
public class FotosController : Controller
{
    private static List<Foto_NA> fotos = SeedFotos.Seed();
    public IActionResult Index()
    {
        return View(fotos);
    }

    public IActionResult Create()
    {
        return View();
    }

    [HttpPost]
    public IActionResult Create(Foto_NA foto)
    {
        fotos.Add(foto);
        return View("Index", fotos);
    }
}
```

1. Uma action Create do tipo HttpGet para mostrar a view onde poderemos inserir os dados do objeto a criar

2. Uma segunda action Create do tipo HttpPost usada para passar os dados da view para o controlador

Depois de criado o objeto somos enviados para a lista

Programação Visual - José Brás - DEI / EST Setúbal / IPS

14

### 5.3.2 ACTION CREATE

#### 5. .NET CORE WEB

##### 5.3 - OPERAÇÕES CRUD

1. Run -> na barra de navegação do browser [...]/localhost/Fotos

2. Premimos o link Create New ... que invoca a action HttpGet Create a qual no envia para a view Create

3. Preenchemos os campos

4. Premimos o btn Create ... e ...

Programação Visual - José Brás - DEI / EST Setúbal / IPS

15

### 5.3.2 ACTION CREATE

#### 5. .NET CORE WEB

##### 5.3 - OPERAÇÕES CRUD

Kabuum!

Falta um construtor sem parâmetros no nosso modelo

Programação Visual - José Brás - DEI / EST Setúbal / IPS

16

### 5.3.1 (BACK TO) MODEL -> FOTO

#### 5. .NET CORE WEB

##### 5.3 - OPERAÇÕES CRUD

```
public static class Foto_NA
{
    public Guid Id { get; set; }
    public String Titulo { get; set; }
    public decimal Preco { get; set; }

    public Foto_NA(String titulo, decimal preco)
    {
        Titulo = titulo;
        Preco = preco;
        Id = Guid.NewGuid();
    }

    public Foto_NA() : this("Sem Titulo", 0.0m)
    {
    }
}
```

4. Abrir o ficheiro Foto\_NA.cs

5. Adicionar um construtor sem parâmetros

Quando criamos o modelo, temos de criar (s e m p r e) um construtor sem parametros

Programação Visual - José Brás - DEI / EST Setúbal / IPS

17

### 5.3.2 ACTION CREATE

#### 5. .NET CORE WEB

##### 5.3 - OPERAÇÕES CRUD

6. (and again) ... Run -> na barra de navegação do browser [...]/localhost/Fotos

7. Premimos o link Create New ... que invoca a action HttpGet Create a qual nos envia para a view Create

8. Preenchemos os campos

9. Premimos o btn Create ...

Nice! O objeto está na lista

Programação Visual - José Brás - DEI / EST Setúbal / IPS

18

### 5.3.1 VIEW CREATE – NÃO ACEITA DECIMAL

#### 5. .NET CORE WEB

##### 5.3 – OPERAÇÕES CRUD

**Solução?**

Como o português usa ',' para separar as casas decimais das inteiros, o Parse do C# pode não converter a string recebida num double ou decimal

```
app.Use(async (context, next) =>
{
    var currentThreadCulture =
        (CultureInfo)Thread.CurrentThread.CurrentCulture.Clone();
    currentThreadCulture.NumberFormat =
        NumberFormatInfo.InvariantInfo;

    Thread.CurrentThread.CurrentCulture = currentThreadCulture;
    Thread.CurrentThread.CurrentUICulture = currentThreadCulture;

    //await next();
});
```

Adicionar ao ficheiro Program.cs, antes do App.Run() para redefinir a CurrentCulture (\*)

\* Thanks to David Anastácio ;-)  
https://stackoverflow.com/questions/64704061/asp-net-core-decimal-input-with-period-turns-into-a-whole-number

Programação Visual - José Brás - DEI / EST Setúbal / IPS

19

### 5.3.3 VIEW DETAILS (READ)

#### 5. .NET CORE WEB

##### 5.3 – OPERAÇÕES CRUD

1. Expandir a pasta **Fotos** das views
2. BDR sobre a pasta **Fotos** -> Add -> View -> Razor View -> Add
3. View name = **Details**  
Template = **Details**  
Model Classe = **Foto\_NA**  
Use layout page: -> prima [...]
4. Navegue Views -> Shared -> Selecione **\_Layout.cshtml** -> OK -> Add -> wait ;)

Programação Visual - José Brás - DEI / EST Setúbal / IPS

20

### 5.3.3 VIEW DETAILS (READ)

#### 5. .NET CORE WEB

##### 5.3 – OPERAÇÕES CRUD

Para navegarmos do link da lista (view Index, Controller Fotos) para um vista de detalhes com os dados do elemento selecionado ...

6. Abrimos o ficheiro Index.cshtml do controlador Fotos ... e

7. Substituímos a linha correspondente ao comando Details pela linha abaixo

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

Programação Visual - José Brás - DEI / EST Setúbal / IPS

21

### 5.3.3 ACTION DETAILS (READ)

#### 5. .NET CORE WEB

##### 5.3 – OPERAÇÕES CRUD

8. Criamos uma action **Details** do tipo **HttpGet** para mostrar a view onde editar os dados do objeto que vem da lista

```
public class FotosController : Controller
{
    //[[ restante Código omitido]

    public IActionResult Details(Guid id)
    {
        var foto = fotos.FirstOrDefault(c => c.Id == id);
        if (foto is null) return View("Index", fotos);
        return View(foto);
    }
}
```

Programação Visual - José Brás - DEI / EST Setúbal / IPS

22

### 5.3.3 VIEW DETAILS (READ)

#### 5. .NET CORE WEB

##### 5.3 – OPERAÇÕES CRUD

9. Run -> na barra de navegação do browser [...]localhost/Fotos

10. Premimos o link **Details** ... que invoca a action **HttpGet Details** a qual no envia para a view **Details**

11. Premimos o link **Back to List**

Programação Visual - José Brás - DEI / EST Setúbal / IPS

23

### 5.3.4 VIEW EDIT (UPDATE)

#### 5. .NET CORE WEB

##### 5.3 – OPERAÇÕES CRUD

1. Expandir a pasta **Fotos**
2. BDR sobre a pasta **Fotos** -> Add -> View -> Razor View -> Add
3. View name = **Edit**  
Template = **Edit**  
Model Classe = **Foto\_NA**  
Use layout page: -> prima [...]
4. Navegue Views -> Shared -> Selecione **\_Layout.cshtml** -> OK -> Add -> wait ;)

Programação Visual - José Brás - DEI / EST Setúbal / IPS

24

### 5.3.4 VIEW EDIT (UPDATE)

#### 5. .NET CORE WEB

##### 5.3 - OPERAÇÕES CRUD

```

1  @model A18_Slides_Operacoes_CRUD.Models.Foto
2  ViewData["Titulo"] = "Edit";
3  Layout = "~/Views/Shared/_Layout.cshtml";
4
5  @addTagHelper
6
7  @using (Html.BeginForm())
8  {
9      <div class="form-group">
10         <div class="form-group">
11             <input type="text" value="@Model.Titulo" class="form-control"/>
12             <input type="text" value="@Model.Preco" class="form-control"/>
13         </div>
14         <div class="form-group">
15             <input type="text" value="@Model.Id" class="form-control" disabled="" />
16             <span asp-validation-for="Id" class="text-danger"></span>
17         </div>
18     </div>
19
20     <div class="form-group">
21         <input type="button" value="Salvar" class="btn btn-primary"/>
22         <input type="button" value="Cancelar" class="btn btn-secondary"/>
23     </div>
24 </div>
25
26 @section Scripts {
27     <script src="~/lib/jquery/dist/jquery.min.js"></script>
28     <script src="~/lib/jquery-validation/dist/jquery.validate.min.js"></script>
29     <script src="~/lib/jquery-validation-unobtrusive/jquery.validate.unobtrusive.min.js"></script>
30 }

```

**5. Na View Edit Desativamos o campo do Id com o atributo disabled do HTML**

25

### 5.3.4 VIEW EDIT (UPDATE)

#### 5. .NET CORE WEB

##### 5.3 - OPERAÇÕES CRUD

Para podermos navegar do link da lista (view Index, Controller Fotos) para um vista de edição com os dados do elemento selecionado ...

6. Abrimos o ficheiro Index.cshtml do controlador Fotos ... e

7. Substituímos a linha correspondente ao comando Edit pela linha abaixo

```

<td>
    @Html.ActionLink("Edit", "Edit", new { /* id=item.PrimaryKey */ }) |
    @Html.ActionLink("Details", "Details", new { /* id=item.PrimaryKey */ }) |
    @Html.ActionLink("Delete", "Delete", new { /* id=item.PrimaryKey */ })
</td>

```

26

### 5.3.4 ACTION EDIT (UPDATE)

#### 5. .NET CORE WEB

##### 5.3 - OPERAÇÕES CRUD

```

public class FotosController : Controller
{
    //[[ restante Código omitido]]
    public IActionResult Edit(Guid id)
    {
        var foto = fotos.FirstOrDefault(c => c.Id == id);
        if (foto is null) return View("Index", fotos);
        return View(foto);
    }

    [HttpPost]
    public IActionResult Edit(Guid id, Foto_NA foto)
    {
        var f = fotos.FirstOrDefault(c => c.Id == id);
        if (f is null) return View("Index");
        f.Titulo = foto.Titulo;
        f.Preco = foto.Preco;
        return View("Index", fotos);
    }
}

```

**8. Uma action Edit do tipo HttpGet para mostrar a view onde editar os dados do objeto que vem da lista**

**9. Uma action Edit do tipo HttpPost usada para passar os dados da view para o controlador**

Depois de atualizado o objeto somos enviados para a lista

27

### 5.3.4 ACTION EDIT (UPDATE)

#### 5. .NET CORE WEB

##### 5.3 - OPERAÇÕES CRUD

10. Run -> na barra de navegação do browser [...]/localhost/Fotos

11. Premimos o link Edit... que invoca a action HttpGet Edit a qual nos envia para a view Edit com os dados do objeto a editar

12. Preenchemos os dados a atualizar

13. Premimos o btn Save ... e ...

14. Voltamos para a lista com dados atualizados

28

### 5.3.5 VIEW DELETE

#### 5. .NET CORE WEB

##### 5.3 - OPERAÇÕES CRUD

1. Expandir a pasta Fotos

2. BDR sobre a pasta Fotos -> Add -> View -> Razor View -> Add

3. View name = Delete  
Template = Delete  
Model classe = Foto\_NA  
Use layout page: -> prima [...]

4. Navegue Views -> Shared -> Selecione \_Layout.cshtml -> OK -> Add -> wait :-)

29

### 5.3.5 VIEW DELETE

#### 5. .NET CORE WEB

##### 5.3 - OPERAÇÕES CRUD

Para podermos navegar do link da lista (view Index, Controller Fotos) para um vista Delete com os dados do elemento selecionado ...

5. Abrimos o ficheiro Index.cshtml do controlador Fotos ... e

6. Substituímos a linha correspondente ao comando Delete pela linha

```

@Html.ActionLink("Delete", "Delete", new { id = item.Id })

```

30

### 5.3.5 ACTION DELETE

#### 5. .NET CORE WEB

##### 5.3 - OPERAÇÕES CRUD

```
public class FotosController : Controller
{
    //[[ restante Código omitido]]
    public IActionResult Delete(Guid id)
    {
        var foto = fotos.FirstOrDefault(c => c.Id == id);
        if (foto is null) return View("Index");
        return View(foto);
    }

    [HttpPost, ActionName("Delete")]
    public IActionResult DeleteConfirmed(Guid id)
    {
        fotos = fotos.Where(c => c.Id != id).ToList();
        return View("Index", fotos);
    }
}
```

7. Uma action **Delete** do tipo **HttpGet** para mostrar a view **Delete** com o objeto que vem da lista

8. Uma action **DeleteConfirmed** do tipo **HttpPost**, **ActionName("Delete")** usada para remover o elemento

Retorna a lista SEM o elemento ;)

Programação Visual - José Brás - DEI / EST Setúbal / IPS

31

### 5.3.5 ACTION DELETE

#### 5. .NET CORE WEB

##### 5.3 - OPERAÇÕES CRUD

9. Run -> na barra de navegação do browser [...]\localhost\Fotos

10. Premimos o link **Delete** -> action **HttpGet Delete** que nos envia para a view **Delete** com os dados do objeto a apagar

11. Confirmamos com o botão **Delete** -> action **DeleteConfirmed**

12. Voltamos para a lista sem o elemento removido

Programação Visual - José Brás - DEI / EST Setúbal / IPS

32

### 5.3.6 PARTIAL VIEW

#### 5. .NET CORE WEB

##### 5.3 - OPERAÇÕES CRUD

1. Expandir a pasta **Shared**

2. BDR sobre a pasta **Shared** -> Add -> View -> **Razor View Empty** -> Add

3. view name: **FotoSearchByTitle.cshtml**

4. Add -> wait -> volá

Programação Visual - José Brás - DEI / EST Setúbal / IPS

33

### 5.3.6 PARTIAL VIEW

#### 5. .NET CORE WEB

##### 5.3 - OPERAÇÕES CRUD

```
@*Criação do formulário*@
<div class="row">
    <div class="col-md-4">
        <form asp-action="Index">
            <div class="form-group">
                <label class="control-label"></label>
                <input name="tituloFoto" class="form-control" />
            </div>
            <div class="form-group">
                <input type="submit" value="Search" class="btn btn-primary" />
            </div>
        </form>
    </div>
</div>
```

5. Substituir o código da **partial view FotoSearchByTitle** pelo de um formulário com um parâmetro **titulo**

Programação Visual - José Brás - DEI / EST Setúbal / IPS

34

### 5.3.6 PARTIAL VIEW

#### 5. .NET CORE WEB

##### 5.3 - OPERAÇÕES CRUD

```
@model A18_Slides_Operacoes_CRUD.Models.Foto_NA
@{
    ViewData["Title"] = "Create";
    Layout = "~/Views/Shared/_Layout.cshtml";
}

<h1>Create</h1>

<partial name="_FotoSearchByTitle" />

<h1>Index</h1>

<partial name="_FotoSearchByTitle" />

<p>
    <a asp-action="Create">Create New</a>
</p>
```

6. Inserir a **partial view FotoSearchByTitle** na view **Create** usando o **Tag Helper partial**

7. Inserir a **partial view FotoSearchByTitle** na view **Index** usando o **Tag Helper partial**

Programação Visual - José Brás - DEI / EST Setúbal / IPS

35

### 5.3.5 ACTION INDEX (FOR SEARCH)

#### 5. .NET CORE WEB

##### 5.3 - OPERAÇÕES CRUD

```
public class FotosController : Controller
{
    //[[ restante Código omitido]]
    // Partial view to search in the Index view
    [HttpPost]
    public IActionResult Index(string tituloFoto)
    {
        if (string.IsNullOrEmpty(tituloFoto)) return View(fotos);

        var filteredFotos = fotos.Where(c =>
            c.Titulo.ToLower().Contains(tituloFoto.ToLower()),
            StringComparison.InvariantCulture).ToList();

        return View(filteredFotos);
    }
}
```

8. Uma action **Index** do tipo **HttpPost** para fazer o request da foto com **tituloFoto**

9. Se o argumento estiver vazio retorna a lista com **todas as fotos**

10. Caso contrário retorna todas as fotos com um **Titulo** que contenha **tituloFoto**

Programação Visual - José Brás - DEI / EST Setúbal / IPS

36

5.3.6 PARTIAL VIEW

5. .NET CORE WEB  
5.3 - OPERAÇÕES CRUD

HomeFotografiasPrivacy

Index

Search

Create New

Id	Titulo	Preco	
48eb6f5f-6ac3-4971-b765-f072cf1e093	Deserto	1.10	<a href="#">Edit</a> <a href="#">Details</a> <a href="#">Delete</a>
2306fbfb-8793-431a-8817-b14c882b28fc	Koala	1.20	<a href="#">Edit</a> <a href="#">Details</a> <a href="#">Delete</a>
13ae5f5f-a42d-4954-8e04-dc459d1b95c5	Tulips	1.30	<a href="#">Edit</a> <a href="#">Details</a> <a href="#">Delete</a>

© 2023 - A18\_Slides\_Operacoes\_CRUD - Privacy

11. Run & Try & Voliá

Programação Visual - José Brás - DEI / EST Setúbal / IPS

37