Curso Técnico Superior Profissional em: Tecnologias e Programação de Sistemas de Informação

2º Ano/1º Semestre

Unidade Curricular: Aplicações Centradas em Redes
Docente: Michael Silva / Hugo Perdigão

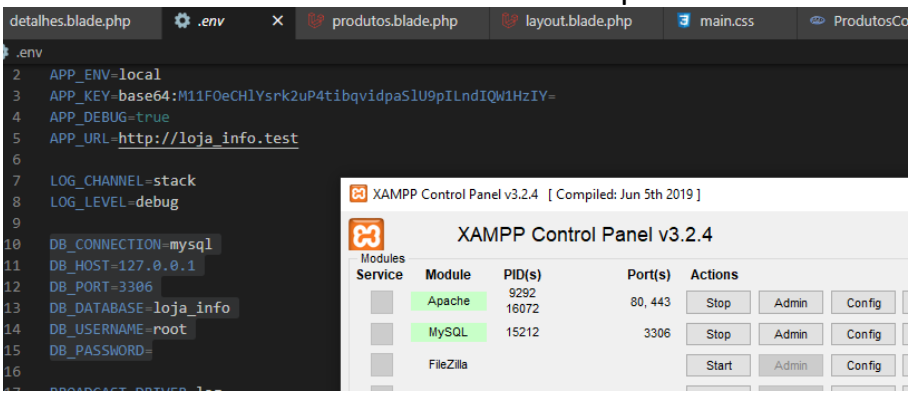
INTRODUÇÃO ÀS LINGUAGENS PARA O DESENVOLVIMENTO DE APLICAÇÕES CENTRADAS EM REDES

Laravel – Exemplo (continuação)

Objetivo:

- **Criar uma base de dados**
- **Criar modelos**
- **Introduzir e eliminar registos**

1) Para aceder à base de dados é necessário ter o acesso configurado no projeto
    a. No ficheiro ".env", verificar host e porta bem como utilizador e password (normalmente em desenvolvimento os valores por defeito estão corretos, basta alterar a base de dados)
    b. É necessário ativar o serviço MySql e se quisermos utilizar o phpMyAdmin será necessário também ativar o apache.



Cofinanciado por:

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA Fundo Social Europeu

2) Criar a base de dados, pode ser feito através do phpMyAdmin ou diretamente no Visual Studio Code através da linha de comandos.
    a. Ligar à base de dados:
        i. Mysql -u root
        ii. Se não conseguir utilizar o mysql como comando pode ser necessário adicionar o mysql ao path
    b. Criar a base de dados
        i. Create database loja_info

```
TERMINAL  PROBLEMAS  SAÍDA  CONSOLE DE DEPURAÇÃO
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\xampp\htdocs\loja_info> mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.4.14-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database loja_info;
Query OK, 1 row affected (0.003 sec)

MariaDB [(none)]>
```

3) Criar uma migration para criação da tabela de produtos.
    a. Num novo terminar, executar:
        i. Php artisan make:migration create_products_table

```
TERMINAL  PROBLEMAS  SAÍDA  CONSOLE DE DEPURAÇÃO
PS C:\xampp\htdocs\loja_info> php artisan make:migration create_products_table
Created Migration: 2020_11_10_141106_create_products_table
```

    b. Editar o ficheiro criado para adicionar as colunas necessárias

```
14      public function up()
15      {
16          Schema::create('products', function (Blueprint $table) {
17              $table->id();
18              $table->timestamps();
19              $table->string('nome');
20              $table->string('desc');
21              $table->string('url');
22              $table->float('preco');
23          });
24      }
25
```

Cofinanciado por:

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA Fundo Social Europeu

4) Para criar a tabela na base de dados executar
    a. Php artisan migrate

```
PS C:\xampp\htdocs\loja_info> php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated:  2014_10_12_000000_create_users_table (52.85ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated:  2014_10_12_100000_create_password_resets_table (32.08ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated:  2019_08_19_000000_create_failed_jobs_table (28.98ms)
Migrating: 2020_11_10_141106_create_products_table
Migrated:  2020_11_10_141106_create_products_table (12.81ms)
PS C:\xampp\htdocs\loja_info>
```

5) Verificar se esta foi criada através do phpMyAdmin

	#	Nome	Tipo	Agrupamento (Collation)	Atrib
☐	1	id	bigint(20)		UNSIGI
☐	2	created_at	timestamp		
☐	3	updated_at	timestamp		
☐	4	nome	varchar(255)	utf8mb4_unicode_ci	
☐	5	desc	varchar(255)	utf8mb4_unicode_ci	
☐	6	url	varchar(255)	utf8mb4_unicode_ci	
☐	7	preco	double(8,2)		

6) Inserir a informação que anteriormente estava no array, na base de dados (através do phpMyAdmin).

Opções

		id	created_at	updated_at	nome	desc	url	preco
☐	Edita Copiar Apagar	1	NULL	NULL	PC Gamer	PC Gamer Top	/img/produtos/1.jpg	1000.00
☐	Edita Copiar Apagar	2	NULL	NULL	PC Workstation	PC para trabalho	/img/produtos/2.jpg	600.00
☐	Edita Copiar Apagar	3	NULL	NULL	PC Bonzinho	PC Bonzinho, desenrasca	/img/produtos/3.jpg	400.00
☐	Edita Copiar Apagar	4	NULL	NULL	PC Rasca	PC mesmo muito fraco	/img/produtos/4.jpg	250.00

7) Criar um Model para interagir com a tabela criada

```
PS C:\xampp\htdocs\loja_info> php artisan make:model Product
Model created successfully.
PS C:\xampp\htdocs\loja_info>
```

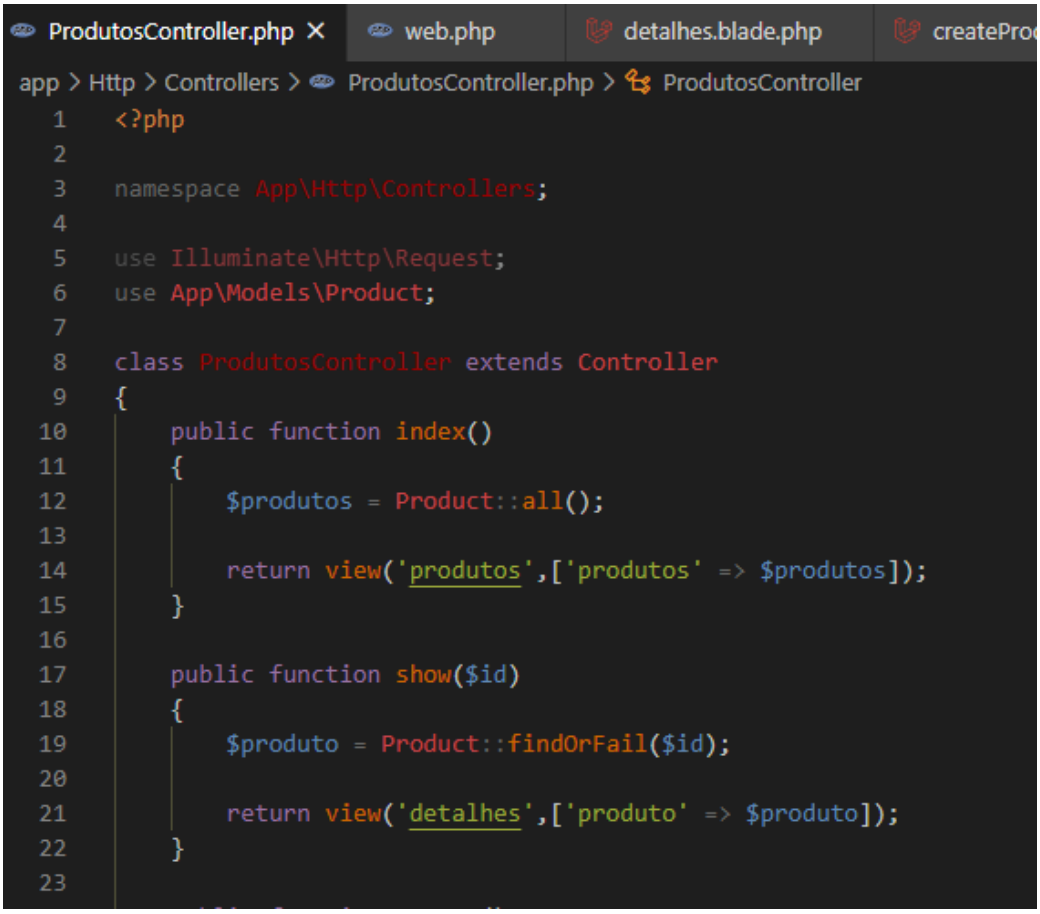
NOTA: <u>não é necessário efetuar qualquer alteração ao Modelo criado, mas para fins informativos este encontra-se na pasta /App/Moddels/.</u>

Cofinanciado por:

MADEIRA 14-20 REGIÃO AUTÓNOMA DA MADEIRA PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu

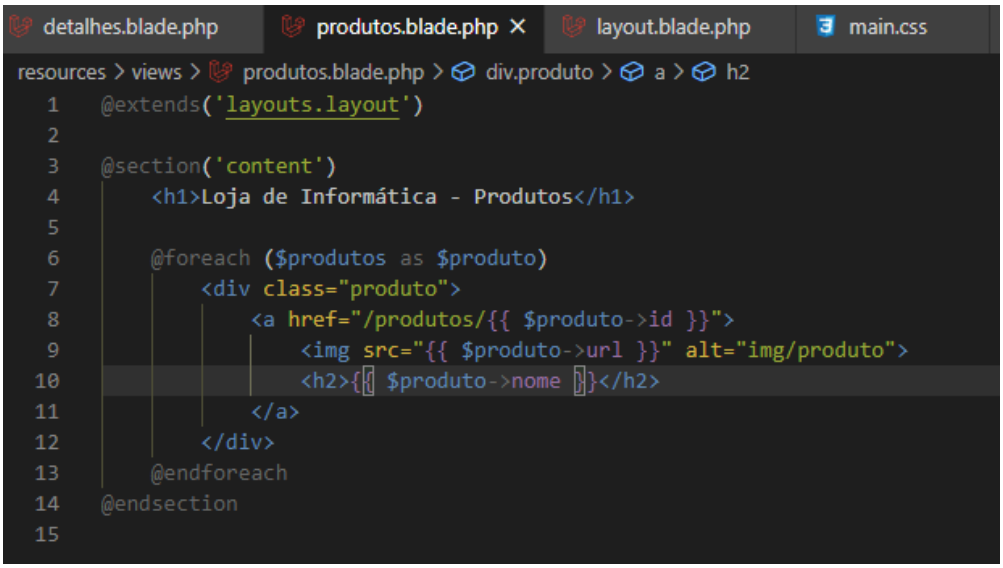
8) Alterar o Controller para utilizar o Modelo criado em vez do array:
    a. Adicionar o namespace App\Model\Product
    b. Eliminar o array;
    c. Utilizar o método Product::all() para retornar todos os registo da DB.
    d. Utilizar o método Product::findOrFail($id) para retornar o registo correspondente ao id passado.

    NOTA: O método findOrFail faz o mesmo que o método find, mas em caso de não existir nenhum registo associado, não gera um erro mas sim, o aviso Page Not Found.

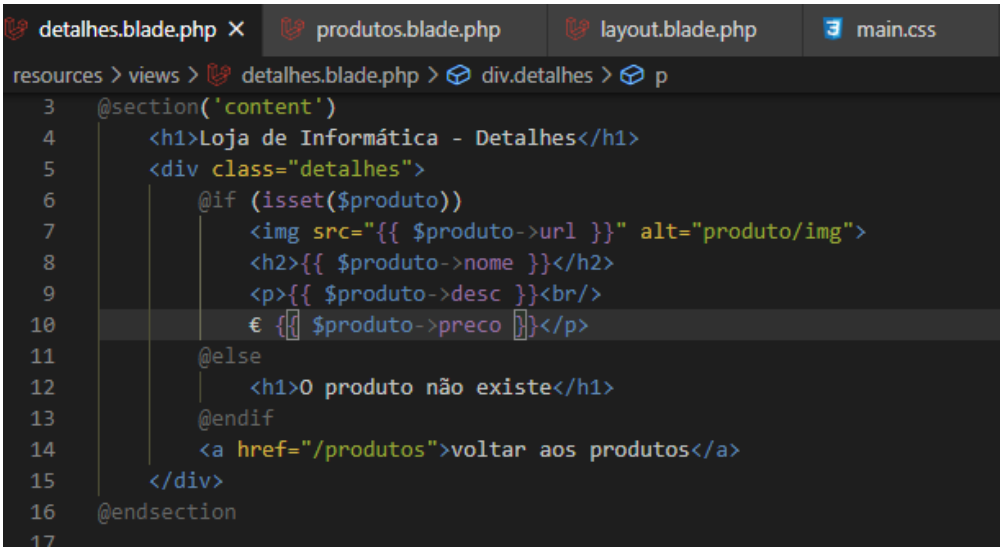


Cofinanciado por:
MADEIRA 14-20
REGIÃO AUTÓNOMA DA MADEIRA
PORTUGAL 2020
UNIÃO EUROPEIA
Fundo Social Europeu

9) Alterar os templates para funcionarem com o objeto retornado em vez do array
   a. products.blade.php



   b. detalhes.blade.php



Cofinanciado por:

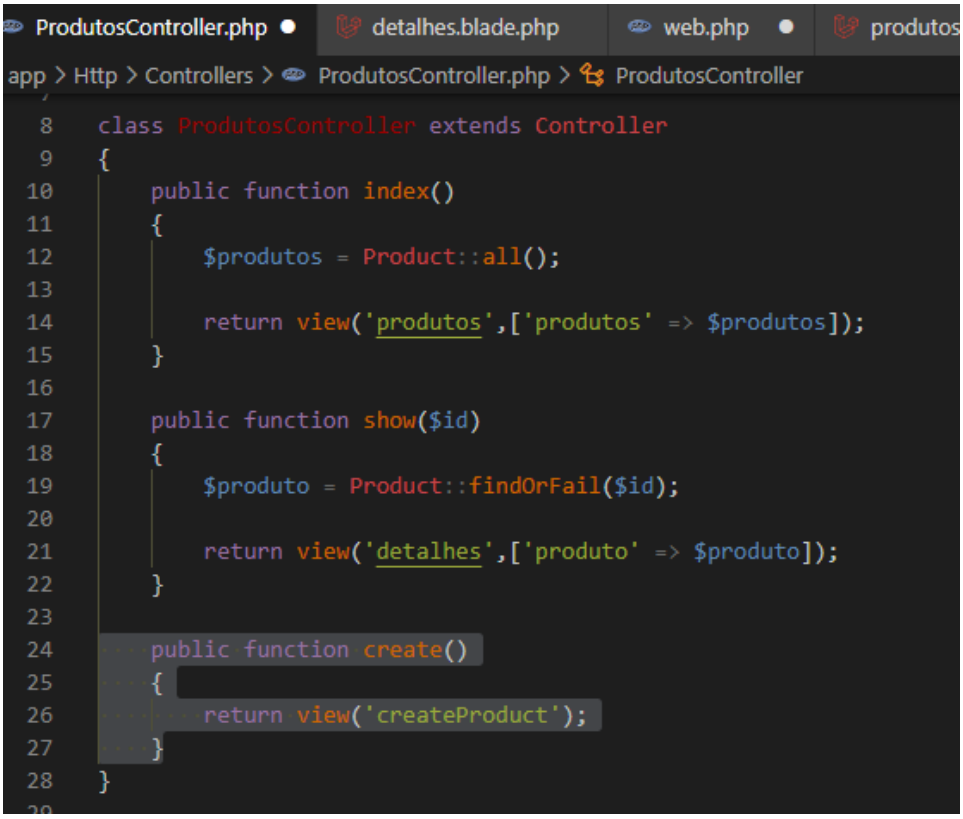
MADEIRA 14-20 REGIÃO AUTÓNOMA DA MADEIRA PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu

10) Criar um formulário para inserir novos produtos:

a. Criar uma nova Route (**apenas a linha selecionada**):

NOTA: A route para o create tem que ficar antes da do show senão create será interpretado como um wildcard e enviado para essa.

b. Criar o método "create" no Controller:



Cofinanciado por:

MADEIRA 14-20 REGIÃO AUTÓNOMA DA MADEIRA PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu

c. Criar a vewi “createProduct”:
   i. Criar uma view “createProduct.blade.php”
   ii. Criar o formulário na view

   NOTA: a linha 6 exibe uma mensagem no casso de ser enviada alguma coisa depois de criado o produto (ver no ponto 10.e).

ProdutosController.php | createProduct.blade.php | detalhes.blade.php

resources > views > createProduct.blade.php > ...

```
1   @extends('layouts.layout')
2
3   @section('content')
4       <h1>Loja de Informática - Criar Produto</h1>
5       <div class="detalhes">
6           <p class="message"> {{ session('mssg') }} </p>
7           <form action="/produtos" method="POST">
8               @csrf
9               <label for="name">Nome do Produto:</label>
10              <input type="text" id='name' name='name'>
11              <br>
12              <label for="desc">Descrição do Produto:</label>
13              <input type="text" id='desc' name='desc'>
14              <br>
15              <label for="url">Imagem:</label>
16              <input type="text" id='url' name='url'>
17              <br>
18              <label for="price">preço:</label>
19              <input type="text" id='price' name='price'>
20              <br>
21              <input type="submit" value="Criar Produto">
22          </form>
23          <a href="/produtos">voltar aos produtos</a>
24      </div>
25  @endsection
26
```

Cofinanciado por:

MADEIRA 14-20 | REGIÃO AUTÓNOMA DA MADEIRA | PORTUGAL 2020 | UNIÃO EUROPEIA Fundo Social Europeu

d. Criar uma Route para onde enviar o formulário. Na propriedade action foi especificada a Route "/produtos" que embora já exista vai ser tratada de forma diferente, uma vez que em vez de GET é utilizado POST.

web.php | ProdutosController.php | createProduct.blade.php | detalhes.blade.pl

routes > web.php > ...

```php
<?php

use App\Http\Controllers\ProdutosController;
use Illuminate\Support\Facades\Route;

Route::get('/', function ()
{
    return view('welcome');
});

Route::get('/produtos', [ProdutosController::class,'index']);
Route::get('/produtos/create', [ProdutosController::class,'create']);
Route::post('/produtos', [ProdutosController::class,'store']);
Route::get('/produtos/{id}', [ProdutosController::class,'show']);

```

e. Por fim criar o método "store" no Controller, chamado para resolver esta Route.

web.php | ProdutosController.php | createProduct.blade.php | detalhes.blade.php

app > Http > Controllers > ProdutosController.php > ProdutosController

```php
    }

    public function create()
    {
        return view('createProduct');
    }

    public function store()
    {
        $name = request('name');
        $desc = request('desc');
        $url = request('url');
        $price = request('price');

        $produto = new Product();

        $produto->nome = $name;
        $produto->desc = $desc;
        $produto->url = $url;
        $produto->preco = $price;

        $produto->save();

        return redirect('/produtos/create')->with('mssg','Produto Criado');
    }
}

```

11) Para eliminar produtos, criar um botão na vista dos detalhes .

Cofinanciado por:

MADEIRA 14-20 | REGIÃO AUTÓNOMA DA MADEIRA | PORTUGAL 2020 | UNIÃO EUROPEIA Fundo Social Europeu

a. Na view “detalhes.blade.php” criar um formulário e o botão.
NOTA: uma vez que não existe metodo DELETE fo formulário é necessário utilizar POST e incluir e diretiva @method(‘DELETE’) dentro do formulário.

detalhes.blade.php | createProduct.blade.php | web.php | ProdutosControll

resources > views > detalhes.blade.php > div.detalhes

```
@extends('layouts.layout')

@section('content')
    <h1>Loja de Informática - Detalhes</h1>
    <div class="detalhes">
        @if (isset($produto))
            <img src="{{ $produto->url }}" alt="produto/img">
            <h2>{{ $produto->nome }}</h2>
            <p>{{ $produto->desc }}<br/>
            € {{ $produto->preco }}</p>
        @else
            <h1>O produto não existe</h1>
        @endif
        <form action="/produtos/{{ $produto->id }}" method="POST">
            @csrf
            @method('DELETE')
            <button>Eliminar Produto</button>
        </form>
        <a href="/produtos">voltar aos produtos</a>
    </div>
@endsection

```

b. Acrescentar a nova Route.
NOTA: apesar de ter a mesma assinatura da route anterior, o tipo de pedido é diferente get vs delete.

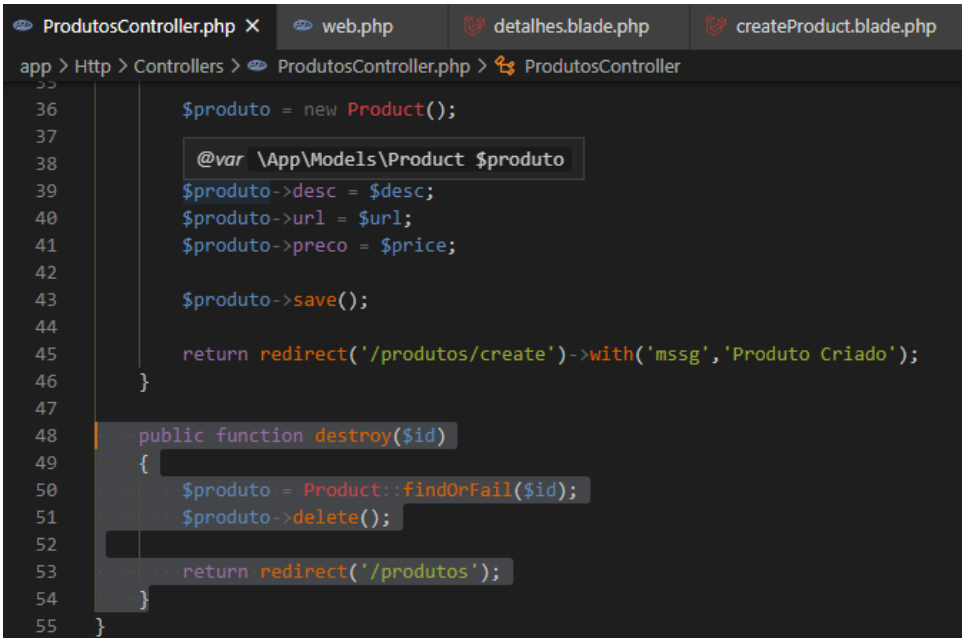
web.php | ProdutosController.php | detalhes.blade.php | createProduct.blade.php

routes > web.php > ...

```
<?php

use App\Http\Controllers\ProdutosController;
use Illuminate\Support\Facades\Route;

Route::get('/', function ()
{
    return view('welcome');
});

Route::get('/produtos', [ProdutosController::class,'index']);
Route::get('/produtos/create', [ProdutosController::class,'create']);
Route::post('/produtos', [ProdutosController::class,'store']);
Route::get('/produtos/{id}', [ProdutosController::class,'show']);
Route::delete('/produtos/{id}', [ProdutosController::class,'destroy']);

```

Cofinanciado por:
MADEIRA 14-20 | REGIÃO AUTÓNOMA DA MADEIRA | PORTUGAL 2020 | UNIÃO EUROPEIA Fundo Social Europeu

c. Por fim criamos o método “destroy” no Controller



Bom Trabalho!

Cofinanciado por:

MADEIRA 14-20 REGIÃO AUTÓNOMA DA MADEIRA PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu