

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

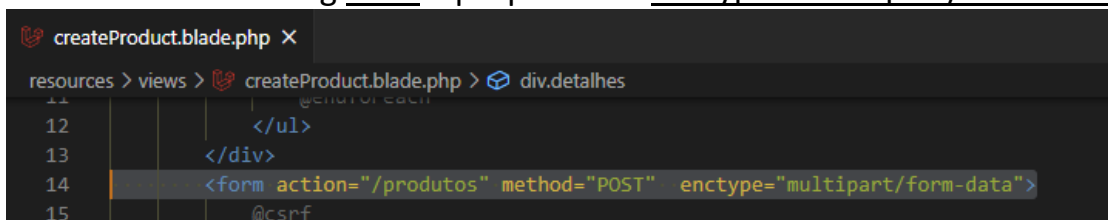
# Laravel – Exemplo (continuação)

### Objetivo:

- Carregar imagens.
- Utilizar validações.
- Utilizar NamedRoutes.
- Criar sistema de autenticação

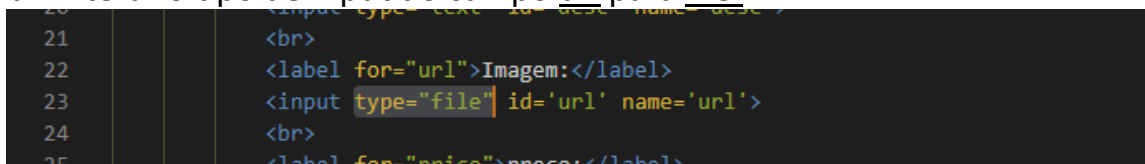
1) Alterar a view “createProduct.blade.php” para aceitar ficheiros.

a. Acrescentar na tag `form` a propriedade `enctype="multipart/form-data"`



```
resources > views > createProduct.blade.php > div.detalhes
11 | @foreach($products as $product)
12 |     </ul>
13 | </div>
14 | <form action="/produtos" method="POST" enctype="multipart/form-data">
15 |     @csrf
```

b. Alterar o tipo de input do campo `url` para `file`:



```
20 | <input type="text" id="desc" name="desc" />
21 | <br>
22 | <label for="url">Imagem:</label>
23 | <input type="file" id="url" name="url" />
24 | <br>
25 | <label for="price">preço:</label>
```

Cofinanciado por:

- 2) Uma vez que os ficheiros enviados para o servidor ficam na pasta storage, antes de continuar é necessário criar um link na pasta publica para a pasta storage, executando:

a. php artisan storage:link

```
TERMINAL  PROBLEMAS  SAÍDA  CONSOLA DE DEBURAÇÃO

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> php artisan storage:link
The [C:\xampp\htdocs\loja_info\public\storage] link has been connected to [C:\xampp\htdocs\loja_info\storage\app/public].
The links have been created.
PS C:\xampp\htdocs\loja_info> 
```

- 3) É agora necessário transferir a imagem da pasta temporária para a sua pasta na storage, no controller, substitui-se a parte onde é pedido o url (request) pelo tratamento da imagem recebida:

a. Acrescentar o parâmetro Request \$request no método.

```
filesystems.php  createProduct.blade.php  ProdutosController.php ●

app > Http > Controllers > ProdutosController.php > ProdutosController > store > [E] $request

28
29     public function store(Request $request)
30     {
```

b. Acrescentar validação server-side para os campos nome e url

```
31         $validateData = $request->validate([
32             'name' => 'required',
33             'url' => 'required|image|mimes:jpeg,png,jpg,gif|max:2048'
34         ]);
```

c. Acrescentar o tratamento de dados

```
35
36         $name = request('name');
37         $desc = request('desc');
38         $price = request('price');
39
40         $url = "";
41         if ($request->has('url'))
42         {
43             $image = $request->file('url');
44
45             $iname = 'prod'.'_'.time();
46             $folder = '/img/produtos/';
47             $fileName = $iname . '.' . $image->getClientOriginalExtension();
48             $filePath = $folder . $fileName;
49
50             $image->storeAs($folder, $fileName, 'public');
51             $url = "/storage/" . $filePath;
52         }
53
54         $produto = new Product();
55
56         $produto->nome = $name;
57         $produto->desc = $desc;
58         $produto->url = $url;
59         $produto->preco = $price;
60
61         $produto->save();
```

Cofinanciado por:

d. Acrescentar na view (createProduct.blade.php) a listagem de erros

```
1 @extends('layouts.app')
2
3 @section('content')
4     <h1>Loja de Informática - Criar Produto</h1>
5     <div class="detalhes">
6         <p class="message"> {{ session('mssg') }} </p>
7         <div class="error">
8             <ul>
9                 @foreach ($errors->all() as $error)
10                     <li>{{ $error }}</li>
11                 @endforeach
12             </ul>
13         </div>
14         <form action="{{ route('products.store') }}" method="POST" enctype="mu
15             @csrf
```

4) A utilização de NamedRoutes permite dar um nome a uma Route para utilizar em qualquer parte da aplicação.

a. Dar nome as routes existentes

```
1 <?php
2
3 use App\Http\Controllers\ProdutosController;
4 use Illuminate\Support\Facades\Route;
5
6 Route::get('/', function ()
7 {
8     return view('welcome');
9 })->name('home');
10
11 Route::get('/produtos', [ProdutosController::class, 'index'])->name('products.index');
12 Route::get('/produtos/create', [ProdutosController::class, 'create'])->name('products.create');
13 Route::post('/produtos', [ProdutosController::class, 'store'])->name('products.store');
14 Route::get('/produtos/{id}', [ProdutosController::class, 'show'])->name('products.show');
15 Route::delete('/produtos/{id}', [ProdutosController::class, 'destroy'])->name('products.destroy');
```

b. Nas views ir buscar as Routes em vez de utilizar urls diretos

i. welcome.blade.php

Cofinanciado por:



```
welcome.blade.php X produtos.blade.php filesystems.php createProduct.blade.php ● Produ

resources > views > welcome.blade.php > div.intro > a
1 @extends('layouts.layout')
2
3 @section('content')
4     <h1>Loja de Informática</h1>
5     <div class="intro">
6         
7
8         <a href="{{ route('products.index') }}">Ver Produtos</a>
9     </div>
```

## ii. produtos.blade.php

```
welcome.blade.php produtos.blade.php X filesystems.php createProduct.blade.php ●

resources > views > produtos.blade.php > div.produto > a
1 @extends('layouts.layout')
2
3 @section('content')
4     <h1>Loja de Informática - Produtos</h1>
5
6     @foreach ($produtos as $produto)
7         <div class="produto">
8             <a href="{{ route('products.show', $produto->id) }}">
9                 
10                 <h2>{{ $produto->nome }}</h2>
11             </a>
12         </div>
13     @endforeach
14 @endsection
```

## iii. detalhes.blade.php

```
detalhes.blade.php X web.php welcome.blade.php produtos.blade.php filesystems.php

resources > views > detalhes.blade.php > div.detalhes > form
1 @extends('layouts.layout')
2
3 @section('content')
4     <h1>Loja de Informática - Detalhes</h1>
5     <div class="detalhes">
6         @if (isset($produto))
7             
8             <h2>{{ $produto->nome }}</h2>
9             <p>{{ $produto->desc }}<br/>
10             € {{ $produto->preco }}</p>
11         @else
12             <h1>O produto não existe</h1>
13         @endif
14         <form action="{{ route('products.destroy', $produto->id) }}" method="POST">
15             @csrf
16             @method('DELETE')
17             <button>Eliminar Produto</button>
18         </form>
19         <a href="/produtos">voltar aos produtos</a>
20     </div>
21 @endsection
22
```

Cofinanciado por:

#### iv. createProduct.blade.php

```
createProduct.blade.php X layout.blade.php main.css detalhes.blade.php web.php welc
resources > views > createProduct.blade.php > div.detalhes > form
1  @extends('layouts.app')
2
3  @section('content')
4      <h1>Loja de Informática - Criar Produto</h1>
5      <div class="detalhes">
6          <p class="message"> {{ session('mssg') }} </p>
7          <div class="error">
8              <ul>
9                  @foreach ($errors->all() as $error)
10                     <li>{{ $error }}</li>
11                 @endforeach
12             </ul>
13         </div>
14         <form action="{{ route('products.store') }}" method="POST" enctype="multipart/form-data">
15             @csrf
16             <label for="name">Nome do Produto:</label>
17             <input type="text" id='name' name='name'>
18             <br>
19             <label for="desc">Descrição do Produto:</label>
20             <input type="text" id='desc' name='desc'>
21             <br>
22             <label for="url">Imagem:</label>
23             <input type="file" id='url' name='url'>
24             <br>
25             <label for="price">preço:</label>
26             <input type="number" id='price' name='price'>
27             <br>
28             <input type="submit" value="Criar Produto">
29         </form>
30         <a href="/produtos">voltar aos produtos</a>
31     </div>
32 @endsection
```

Cofinanciado por:



5) Para instalar o sistema de autenticação pré-definido:

a. Instalar a UI utilizando o composer:

composer require "laravel/ui"

```
TERMINAL  PROBLEMAS  SAÍDA  CONSOLE DE DEPURAÇÃO

PS C:\xampp\htdocs\loja_info> composer require "laravel/ui"
Using version ^3.1 for laravel/ui
./composer.json has been updated
Running composer update laravel/ui
```

b. Instalar os componentes na aplicação:

php artisan ui vue --auth

```
PS C:\xampp\htdocs\loja_info> php artisan ui vue --auth
Vue scaffolding installed successfully.
Please run "npm install && npm run dev" to compile your fresh scaffolding.
Authentication scaffolding generated successfully.
PS C:\xampp\htdocs\loja_info>
```

c. seguir as instruções e correr

npm install

```
PS C:\xampp\htdocs\loja_info> npm install
npm WARN deprecated popper.js@1.16.1: You can find the new Popper v2 at @popperjs/core
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated chokidar@2.1.8: Chokidar 2 will break on node v14+. Upgrade to chokidar 3
npm WARN deprecated fsevents@1.2.13: fsevents 1 will break on node v14+ and could be u
```

npm run dev

```
PS C:\xampp\htdocs\loja_info> npm run dev

> @ dev C:\xampp\htdocs\loja_info
> npm run development

> @ development C:\xampp\htdocs\loja_info
> cross-env NODE_ENV=development node_modules/webpack/bin/webpack.js --progress --config=node_
```

após a mensagem final, a UI estará instalada:

```
TERMINAL  PROBLEMAS  1  SAÍDA  CONSOLE DE DEPURAÇÃO

DONE Compiled successfully in 17417ms

    Asset      Size  Chunks             Chunk Names
 /css/app.css  178 KiB  /js/app [emitted]  /js/app
  /js/app.js   1.41 MiB  /js/app [emitted]  /js/app
Installing shortcut: "C:\Users\hugop\AppData\Roaming\Microsoft\Windows\Start Men
t\snoretoast-x64.exe" Snore.DesktopToasts.0.6.0
PS C:\xampp\htdocs\loja_info>
```

Cofinanciado por:



6) A UI instalada criou uma série novos templates, e nova logica de utilização. Também foi criado um novo layout que pode ser aplicado às páginas existentes:

- a. Nas views criadas substituir
  - i. welcome.blade.php
  - ii. produtos.blade.php
  - iii. detalhes.blade.php
  - iv. createProduct.blade.php

```
resources > views > detalhes.blade.php > ...  
1  @extends('layouts.app')  
2
```

7) Pode acontecer algumas página não ficarem bem configuradas pois o novo layout não inclui a CSS criada.

- a. Incluir a CSS no novo layout:

```
app.blade.php X layout.blade.php main.css detalhes.blade.php web.php  
resources > views > layouts > app.blade.php > html > head  
14  
15 <!-- Fonts -->  
16 <link rel="dns-prefetch" href="//fonts.gstatic.com">  
17 <link href="https://fonts.googleapis.com/css?family=Nunito" rel="stylesheet">  
18  
19 <!-- Styles -->  
20 <link href="{{ asset('css/app.css') }}" rel="stylesheet">  
21 <link href="/css/main.css" rel="stylesheet">  
22 </head>  
23 <body>
```

- b. Substituir o título (Laravel) pelo nome da loja

```
23 <body>  
24 <div id="app">  
25 <nav class="navbar navbar-expand-md navbar-light bg-white shadow-sm">  
26 <div class="container">  
27 <a class="navbar-brand" href="{{ route('root') }}">  
28 Loja dos Computadores  
29 </a>  
30 <button class="navbar-toggler" type="button" data-toggle="coll  
31 <span class="navbar-toggler-icon"></span>  
32 </button>  
33  
34 <div class="collapse navbar-collapse" id="navbarSupportedConte
```

Cofinanciado por:



8) A partir deste momento é possível fazer autenticação no site, mas qualquer pessoa pode aceder a todas as páginas.

- a. Proteger as Route products.create, products.store e products.destroy para apenas utilizadores registados poderem aceder. Acrescentar nas routes protegidas a execução do método middleware('auth').

```
web.php x createProduct.blade.php .env layout.blade.php main.css detalhes.blade.php welcome.t
routes > web.php > ...
1 <?php
2
3 use App\Http\Controllers\ProdutosController;
4 use Illuminate\Support\Facades\Route;
5
6 Route::get('/', function ()
7 {
8     return view('welcome');
9 })->name('root');
10
11 Route::get('/produtos', [ProdutosController::class,'index'])->name('products.index');
12 Route::get('/produtos/create', [ProdutosController::class,'create'])->name('products.create')->middleware('auth');
13 Route::post('/produtos', [ProdutosController::class,'store'])->name('products.store')->middleware('auth');
14 Route::get('/produtos/{id}', [ProdutosController::class,'show'])->name('products.show');
15 Route::delete('/produtos/{id}', [ProdutosController::class,'destroy'])->name('products.destroy')->middleware('auth');
16
17 Auth::routes();
18
19 Route::get('/home', [App\Http\Controllers\HomeController::class, 'index'])->name('home');
20
```

9) Acrescentar o acesso à página de criação de produtos no dashboard ([home.blade.php](#)).

```
home.blade.php x web.php createProduct.blade.php .env layout.blade.
resources > views > home.blade.php > div.container > div.row.justify-content-center > div.col-md-8 >
1 @extends('layouts.app')
2
3 @section('content')
4 <div class="container">
5     <div class="row justify-content-center">
6         <div class="col-md-8">
7             <div class="card">
8                 <div class="card-header">{{ __('Dashboard') }}</div>
9
10                <div class="card-body">
11                    @if (session('status'))
12                        <div class="alert alert-success" role="alert">
13                            {{ session('status') }}
14                        </div>
15                    @endif
16
17                    <a href="{{ route('products.create') }}">Criar Produto</a>
18                </div>
19            </div>
20        </div>
21    </div>
22 </div>
23 @endsection
```

Cofinanciado por:



- 10) Fazer com que o botão eliminar só apareça para utilizadores registados, fazendo uma validação básica e colocando o botão dentro de um bloco @auth.

```
app.blade.php | detalhes.blade.php | home.blade.php | web.php | createProduct.blade.php
resources > views > detalhes.blade.php > div.detalhes
1  @extends('layouts.app')
2
3  @section('content')
4      <h1>Loja de Informática - Detalhes</h1>
5      <div class="detalhes">
6          @if (isset($produto))
7              
8              <h2>{{ $produto->nome }}</h2>
9              <p>{{ $produto->desc }}<br/>
10             € {{ $produto->preco }}</p>
11          @else
12              <h1>O produto não existe</h1>
13          @endif
14
15          @auth
16              <form action="{{ route('products.destroy',$produto->id) }}" method="POST">
17                  @csrf
18                  @method('DELETE')
19                  <button>Eliminar Produto</button>
20              </form>
21          @endauth
22          <a href="/produtos">voltar aos produtos</a>
23      </div>
24  @endsection
```

- 11) Acrescentar um botão para criar produto na dropdown do utilizador no layout (app.blade.php)

```
app.blade.php X | detalhes.blade.php | home.blade.php | web.php | createProduct.blade.php | .env | layout.blade.php
#app > nav.navbar.navbar-expand-md.navbar-light.bg-white.shadow-sm > div.container > div#navbarSupportedContent.collapse.navbar-collapse > ul.navbar-nav
48  @endif
49
50  @if (Route::has('register'))
51      <li class="nav-item">
52          <a class="nav-link" href="{{ route('register') }}">{{ __('Register') }}</a>
53      </li>
54  @endif
55  @else
56      <li class="nav-item dropdown">
57          <a id="navbarDropdown" class="nav-link dropdown-toggle" href="#" role="button" data-toggle="dropdown">
58              {{ Auth::user()->name }}
59          </a>
60
61          <div class="dropdown-menu dropdown-menu-right" aria-labelledby="navbarDropdown">
62
63              <a class="dropdown-item" href="{{ route('products.create') }}">
64                  Criar Produto
65              </a>
66
67              <a class="dropdown-item" href="{{ route('logout') }}">
68                  onclick="event.preventDefault();
69                  document.getElementById('logout-form').submit();"
70                  {{ __('Logout') }}
71              </a>
72          </div>
73      </li>
74  @endif
```

- 12) Testar todas as funcionalidades do Site

Bom Trabalho!

Cofinanciado por:

