

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:

- Editar Registos
- Limitar nr. Registos por utilizador
- Privilégios de Administrador

- 1) Criar campo no Produto para identificar o utilizador que o criou.
 - a. Criar a migration que cria o campo e a relação

```
TERMINAL  PROBLEMAS  SAÍDA  CONSOLE DE DEPURACÃO

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 make:migration add_fk_products_user
Created Migration: 2020_12_03_100450_add_fk_products_user
```

Cofinanciado por:

b. Editar a migration

```
2020_12_03_100450_add_fk_products_user.php X 2014_10_12_000000_create_users_table.php
database > migrations > 2020_12_03_100450_add_fk_products_user.php > AddFkProductsUser > up
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class AddFkProductsUser extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::table('products', function (Blueprint $table)
17         {
18             $table->bigInteger('created_by')->unsigned();
19             $table->foreign('created_by')->references('id')->on('users');
20         });
21     }
22 }
```

c. Executar para criar na base de dados

Php artisan migrate:fresh

NOTA: ao acrescentar **fresh** elimina toda a base de dados e cria de novo

```
TERMINAL  PROBLEMAS  SAÍDA  CONSOLE DE DEPURAÇÃO







PS C:\xampp\htdocs\loja_info> php artisan migrate:fresh
Dropped all tables successfully.
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (39.77ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (36.87ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (40.50ms)
Migrating: 2020_11_10_141106_create_products_table
Migrated: 2020_11_10_141106_create_products_table (18.58ms)
Migrating: 2020_11_20_230300_create_tipo_produto_table
Migrated: 2020_11_20_230300_create_tipo_produto_table (14.41ms)
Migrating: 2020_11_20_230455_add_fk_products_tipo_produto
Migrated: 2020_11_20_230455_add_fk_products_tipo_produto (62.41ms)
Migrating: 2020_12_03_100450_add_fk_products_user
Migrated: 2020_12_03_100450_add_fk_products_user (64.39ms)
PS C:\xampp\htdocs\loja_info> 
```

Cofinanciado por:



- d. Como a base de dados foi eliminada, criar pelo menos 2 **tipos de produto** para teste (ou voltar a adicionar os da ficha anterior).

+ Opções

				id	created_at	updated_at	nome
<input type="checkbox"/>		Edita		Copiar		Apagar	1 NULL NULL Computador Portatil
<input type="checkbox"/>		Edita		Copiar		Apagar	2 NULL NULL Computador Pessoal

- 2) Adicionar ao produto o utilizador que o criou

- a. No **Controller** acrescentar o id do utilizador autenticado no produto

```
ProdutosController.php X 2020_12_03_100450_add_fk_products_user.php 2014_10_12_00000
app > Http > Controllers > ProdutosController.php > ProdutosController > store
67
68     $produto->nome = $name;
69     $produto->desc = $desc;
70     $produto->url = $url;
71     $produto->preco = $price;
72     $produto->tipo_produto_id = $tipo;
73     $produto->created_by = auth()->user()->id;
74
75     $produto->save();
76
77     return redirect('/produtos/create')->with('mssg','Produto Criado');
78 }
```

NOTA: Registrar um utilizador e criar um produto como teste.

- 3) Para editar um produto, acrescentar um botão **Editar** nos produtos (view detalhes)

- a. Acrescentar formulário e botão

```
detalhes.blade.php X ProdutosController.php web.php 2020_12_03_100450_add_fk_products_user.php
resources > views > detalhes.blade.php > div.detalhes
15     @auth
16         <form action="{{ route('products.destroy',$produto->id) }}" method="POST">
17             @csrf
18             @method('DELETE')
19             <button>Eliminar Produto</button>
20         </form>
21
22         <form action="{{ route('products.edit',$produto->id) }}" method="GET">
23             @csrf
24             <button>Editar Produto</button>
25         </form>
26     @endauth
```

Cofinanciado por:

b. Criar Route

```
web.php x detalhes.blade.php ProdutosController.php 2020_12_03_100450_add_fk_products_user.php 2020_11_20_230300_
routes > web.php > ...
9  })->name('root');
10
11 Route::get('/produtos', [ProdutosController::class, 'index'])->name('products.index');
12 Route::get('/produtos/tipo/{id}', [ProdutosController::class, 'produtosPorTipo'])->name('products.by.tipo');
13 Route::get('/produtos/create', [ProdutosController::class, 'create'])->name('products.create')->middleware('auth');
14 Route::get('/produtos/edit/{id}', [ProdutosController::class, 'edit'])->name('products.edit')->middleware('auth');
15 Route::post('/produtos', [ProdutosController::class, 'store'])->name('products.store')->middleware('auth');
```

c. Criar o método no controller

```
ProdutosController.php x web.php detalhes.blade.php 2020_12_03_100450_add_fk_products
app > Http > Controllers > ProdutosController.php > ProdutosController
36
37 public function create()
38 {
39     $tipos = tipo_produto::all();
40
41     return view('createProduct', [ 'tipos' => $tipos]);
42 }
43
44 public function edit($id)
45 {
46     $tipos = tipo_produto::all();
47     $produto = Product::findOrFail($id);
48
49     return view('createProduct', [ 'produto' => $produto, 'tipos' => $tipos]);
50 }
```

Cofinanciado por:



- 4) No formulário, verificar se o produto existe, e alterar algumas especificações de acordo com isso.
- O título existente para especificar Criar ou Editar consoante exista ou não produto.

```
createProduct.blade.php X web.php detalhes.blade.php ProdutosController.php 202
resources > views > createProduct.blade.php > h1
1 @extends('layouts.app')
2
3 @section('content')
4     <h1>Loja de Informática -
5     @if (isset($produto))
6         Editar Produto
7     @else
8         Criar Produto
9     @endif
10 </h1>
11 <div class="detalhes">
12     <p class="message"> {{ session('mssg') }} </p>
```

- Alterar a **action** do formulário para **store** ou **update** e especificar o método **PUT** no caso do **update**.

```
18 </div>
19 </div>
20 <form method="POST" enctype="multipart/form-data"
21 @if (isset($produto))
22     action="{{ route('products.update',$produto->id) }}"
23 @else
24     action="{{ route('products.store') }}"
25 @endif
26 >
27 @csrf
28 @if (isset($produto))
29     @method('PUT')
30 @endif
31 <label for="name">Nome do Produto:</label>
```

- Alterar os inputs para receber o valor correspondente caso exista produto

 - Nome do produto:

```
22 <label for="name">Nome do Produto:</label>
23 <input type="text" id='name' name='name'
24 @if (isset($produto))
25     value="{{ $produto->nome }}"
26 @endif
27 >
28 <br>
29 <label for="desc">Descrição do Produto:</label>
```

Cofinanciado por:

ii. Descrição do produto:

```
28 <br>
29 <label for="desc">Descrição do Produto:</label>
30 <input type="text" id='desc' name='desc'
31 @if (isset($produto))
32 value="{{ $produto->desc }}"
33 @endif
34 >
35 <br>
```

iii. Preço do Produto:

```
47 <br>
48 <label for="price">preço:</label>
49 <input type="number" id='price' name='price'
50 @if (isset($produto))
51 value="{{ $produto->preco }}"
52 @endif
53 >
54 <br>
55 <label for="tipoProduto">Tipo de Produto:</label>
```

iv. Imagem

1. Adicionar um input (hidden) para registar se o campo file foi ou não alterado;
2. Adicionar js ao evento “onchange” para alterar o estado do input;
3. Adicionar a mensagem “não alterar para manter foto”

```
43 >
44 <br>
45 <input type="hidden" id="changed" name="changed" value="false">
46 <label for="url">Imagem:</label>
47 <input type="file" id='url' name='url'
48 onchange="document.getElementById('changed').value='true';">
49 @if (isset($produto))
50 (não alterar para manter foto)
51 @endif
52 <br>
```

- v. Tipo de produto – aqui é necessário adicionar **selected** na opção correspondente ao tipo atribuído ao produto (quando existe).

```
56 <select name="tipoProduto" id="tipoProduto">
57 @foreach ($tipos as $tipo)
58 <option value="{{ $tipo->id }}"
59 @if (isset($produto) && $produto->tipo_produto_id == $tipo->id)
60 selected="selected"
61 @endif
62 >{{ $tipo->nome }}</option>
63 @endforeach
64 </select>
```

Cofinanciado por:

- vi. Alterar o valor do botão para mostrar a mensagem correspondente

```
68         <br>
69         <input type="submit"
70             @if(isset($produto))
71                 value="Editar Produto"
72             @else
73                 value="Criar Produto"
74             @endif>
75     </form>
76     <a href="/produtos">voltar aos produtos</a>
```

5) Criar a Route para o Update

```
createProduct.blade.php  web.php  detalhes.blade.php  ProdutosController.php  EditProductRequest.php  2020_12_03
routes > web.php > ...
14 Route::get('/produtos/edit/{id}', [ProdutosController::class, 'edit'])->name('products.edit')->middleware('auth');
15 Route::post('/produtos', [ProdutosController::class, 'store'])->name('products.store')->middleware('auth');
16 Route::get('/produtos/{id}', [ProdutosController::class, 'show'])->name('products.show');
17 Route::put('/produtos/{id}', [ProdutosController::class, 'update'])->name('products.update')->middleware('auth');
18 Route::delete('/produtos/{id}', [ProdutosController::class, 'destroy'])->name('products.destroy')->middleware('auth');
19
```

6) Criar um novo request para envio dos dados do formulário editar

```
TERMINAL  PROBLEMAS  3  SAIDA  CONSOLE DE DEPURACÃO

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 make:request EditProductRequest
Request created successfully.
PS C:\xampp\htdocs\loja_info>
```

a. Autorizar o request

```
EditProductRequest.php  createProduct.blade.php  web.php  detalhes.blade.php
app > Http > Requests > EditProductRequest.php > EditProductRequest > authorize
14 public function authorize()
15 {
16     return true;
17 }
```

Cofinanciado por:



- b. Adiciona as Regras (igual ao NewProduct request mas a imagem deixa de ser obrigatória)

```
23     */
24     public function rules()
25     {
26         return [
27             'name' => 'required',
28             'url' => 'image|mimes:jpeg,png,jpg,gif|max:2048'
29         ];
30     }
31
32
```

- c. Adiciona as mensagens Personalizadas

```
32
33     public function messages()
34     {
35         return [
36             'name.required' => 'O Nome do Produto é Obrigatório.',
37             'url.image' => 'A Imagem deve ser uma imagem.',
38             'url.mimes' => 'A Imagem pode ser jpeg,png,jpg,gif.',
39             'url.max' => 'A Imagem Não pode exceder 2MB.',
40         ];
41     }
42 }
```

7) Criar o método Update no controller

```
ProdutosController.php • createProduct.blade.php • web.php • detalhes.blade.php
app > Http > Controllers > ProdutosController.php > ProdutosController
86
87     return redirect('/produtos/create')->with('mssg','Produto Criado');
88 }
89
90     public function update($id, EditProductRequest $request)
91     {
92         $name = request('name');
93         $desc = request('desc');
94         $price = request('price');
95         $tipo = request('tipoProduto');
96
97         $changed = bool request('changed');
98
99         $produto = Product::findOrFail($id);
100
```

Cofinanciado por:




```

100
101     if($changed == 'true')
102     {
103         $url = "";
104         if ($request->has('url'))
105         {
106             $image = $request->file('url');
107
108             $iname = 'prod'.'_'.time();
109             $folder = '/img/produtos/';
110             $fileName = $iname.'.'.$image->getClientOriginalExtension();
111             $filePath = $folder.$fileName;
112
113             $image->storeAs($folder, $fileName, 'public');
114             $url = "/storage/".$filePath;
115         }
116
117         $produto->url = $url;
118     }
119
120
121     $produto->nome = $name;
122     $produto->desc = $desc;
123     $produto->preco = $price;
124     $produto->tipo_produto_id = $tipo;
125
126     $produto->save();
127
128     return redirect("/produtos/$id")->with('mssg','Produto Criado');
129 }
130

```

8) Para contabilizar os Produtos criados pelo utilizador, adicionar no Model User a relação com os produtos

```

User.php  x  ProdutosController.php  tipo_produto.php  createProduct.blade.php  web.php
app > Models > User.php > User
40     protected $casts = [
41         'email_verified_at' => 'datetime',
42     ];
43
44     public function products()
45     {
46         return $this->hasMany('App\Models\Product', 'created_by', 'id');
47     }
48 }

```

Cofinanciado por:



9) No controller antes de disponibilizar o formulário, verificar se já atingiu o Limite.

```
ProdutosController.php • User.php • tipo_produto.php • createProduct.blade.php • web.php • de
app > Http > Controllers > ProdutosController.php > ProdutosController > create
33     return view('detalhes',['produto' => $produto]);
34 }
35
36 public function create()
37 {
38     $tipos = tipo_produto::all();
39
40     $user = auth()->user();
41     $userProducts = $user->products;
42     if($userProducts->count() == 10)
43     {
44         return redirect('/home')->with('mssg','Não pode criar mais produtos');
45     }
46
47     return view('createProduct',[ 'tipos' => $tipos]);
48 }
49
```

NOTA: Para não inserir diretamente no código o número máximo, pode ser criada uma variável de ambiente com o esse valor no ficheiro **.env**:

```
.env × createProduct.blade.php home.blade.php ProdutosController.php
.env
48
49 MAX_PRODUCTS = 10
50
```

E utilizada em vez do valor no controller

```
42     if($userProducts->count() >= env('MAX_PRODUCTS'))
43     {
44         return redirect('/home')->with('mssg','Não pode cr
45     }
```

10) Para dar acesso de administrador é necessário adicionar uma coluna que dá essa permissão.

a. Criar a migration para adicionar a coluna:

```
TERMINAL PROBLEMAS 1 SAÍDA CONSOLE DE DEPURAÇÃO
Request created successfully.
PS C:\xampp\htdocs\loja_info> php artisan make:migration Add_IsAdmin_to_Users_table
Created Migration: 2020_12_04_104445_add__is_admin_to__users_table
PS C:\xampp\htdocs\loja_info>
```

Cofinanciado por:



b. Adicionar a coluna no método UP

```
2020_12_04_104445_add_is_admin_to_users_table.php X User.php web.php detailh
database > migrations > 2020_12_04_104445_add_is_admin_to_users_table.php > AddIsAdminToUsersTable

9      /**
10     * Run the migrations.
11     *
12     * @return void
13     */
14     public function up()
15     {
16         Schema::table('users', function (Blueprint $table) {
17             $table->boolean('IsAdmin')->default(false);
18         });
19     }
20
```

c. Executar a migration

```
PS C:\xampp\htdocs\loja_info> php artisan migrate
Migrating: 2020_12_04_104445_add_is_admin_to_users_table
Migrated: 2020_12_04_104445_add_is_admin_to_users_table (3.28ms)
PS C:\xampp\htdocs\loja_info>
```

d. Registrar um novo utilizador e dar permissões de administrador apenas ao administrador (alterar diretamente na BD).

id	name	email	email_verified_at	password	remember_token	created_at	updated_at	IsAdmin
1	Hugo Perdigão	hugopq@gmail.com	NULL	\$2y\$10\$4R6yAUMlbs4ActYj3wii0.CHPMIS9XXF.MyyRx2v7Vk...	NULL	2020-12-03 10:27:01	2020-12-03 10:27:01	0
2	Admin	admin@admin.com	NULL	\$2y\$10\$UFdw1NjRlwlynKt4rPwDDOiR9bfdjdz72i5TpD5n/...	NULL	2020-12-04 10:52:42	2020-12-04 10:52:42	1

e. Criar a condição para mostrar os botões (se é o utilizador que criou o produto ou se é administrador)

```
detalhes.blade.php X 2020_12_04_104445_add_is_admin_to_users_table.php User.php web.php
resources > views > detalhes.blade.php > div.detalhes

14
15     @auth
16     <?php if ($produto->created_by == auth()->user()->id || auth()->user()->IsAdmin)
17     <form action="{{ route('products.destroy',$produto->id) }}" method="POST">
18         @csrf
19         @method('DELETE')
20         <button>Eliminar Produto</button>
21     </form>
22
23     <form action="{{ route('products.edit',$produto->id) }}" method="GET">
24         @csrf
25         <button>Editar Produto</button>
26     </form>
27     @endif
```

Cofinanciado por:



- f. Criar um terceiro utilizador e fazer login com os três para verificar se os botões aparecem apenas no utilizadores corretos.

Bom Trabalho!

Cofinanciado por:

