

# CURSO LARAVEL 5.3

## EspecializaTI (Carlos Ferreira)

<https://www.youtube.com/watch?v=iPTxznidNHo&list=PLVSNL1PHDWvR3PeLXz6nvBkDhv1IQk4wP>

Resumo do curso feito por Roberto Pinheiro

### Aulas 4 e 5 - Rotas

#### routes/web.php

```
<?php
```

```
/* Aula 04 */
```

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

```
Route::get('/', function () {  
    return redirect() -> route('rota.nomeada');  
});
```

```
// Route::get('/empresa', function(){  
//     return "Pg empresa";  
// });
```

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

```
Route::get('/contato', function(){  
    return "Contato";  
});
```

```
Route::match(['get','post'],'/match',function(){  
    return "Route Match";  
});
```

```
Route::any('/any', function(){  
    return "Route any";  
});
```

```

Route::get('/nome/nome2/nome7', function(){
    return "Rota grande";
})->name('rota.nomeada');

/* ----- */

/* Aula 05 */

Route::get('/categoria/{idCat}', function($idCat){
    return "Posts da categoria $idCat";
});

Route::get('/categoria/{idCat}/nome-fixa/{prm2}', function($idCat, $prm2){
    return "Posts da categoria $idCat - $prm2";
});

// Route::get('/categoria2/{idCat?}', function($idCat=null){
//     return "Posts da categoria $idCat";
// });

Route::get('/categoria2/{idCat?}', function($idCat=1){
    return "Posts da categoria $idCat";
});

Route::group(['prefix' => 'painel', function(){
    Route::get('/grupo1', function(){
        return "Grupo 1";
    });
    Route::get('/grupo2', function(){
        return "Grupo 2";
    });
});

Route::group(['prefix' => 'painel', 'middleware' => 'auth'], function(){
    Route::get('/', function(){
        return "Dashboard";
    });
    Route::get('/financeiro', function(){
        return "Financeiro Pannel";
    });
    Route::get('/users', function(){
        return "Controle de users";
    });
});

Route::get('/login', function(){
    return "#frm login";
});

/* ----- */

```

## Aula 6 - Controllers

No terminal do Laravel, crie o controller `SiteController` entrando com o comando:

`php artisan make:controller SiteController`

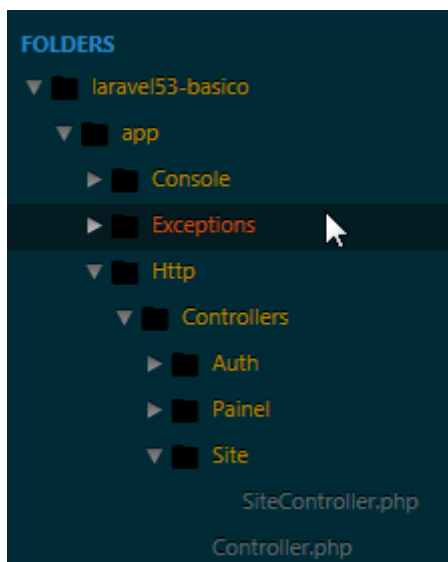
```
C:\laragon\www\primeiro
λ cd ..

C:\laragon\www
λ cd laravel53-basico

C:\laragon\www\laravel53-basico
λ php artisan make:controller SiteController
Controller created successfully.

C:\laragon\www\laravel53-basico
λ |
```

Dentro da pasta “Controller” crie duas novas pastas: “Site” e “Painel”. Mova o arquivo `SiteController` para a pasta Site.



## **web.php**

```
<?php
```

```
/* Aula 06 */
```

```
Route::group(['namespace' => 'Site'], function(){  
    Route::get('/', 'SiteController@index');  
    Route::get('/contato', 'SiteController@contato');  
    Route::get('/categoria/{id}', 'SiteController@categoria');  
    Route::get('/categoria2/{id?}', 'SiteController@categoriaOp');  
});
```

## **SiteController.php**

```
<?php
```

```
namespace App\Http\Controllers\Site;
```

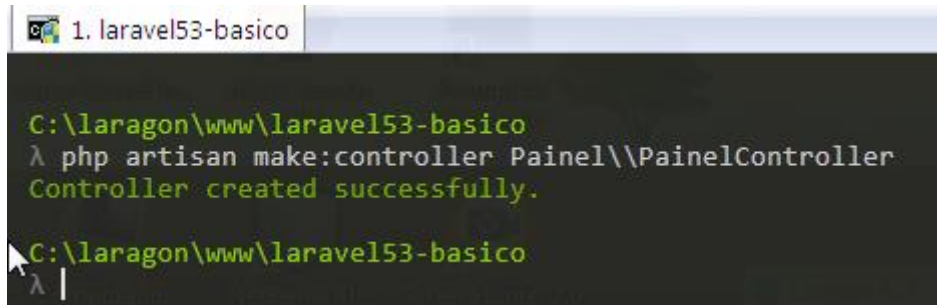
```
use Illuminate\Http\Request;
```

```
use App\Http\Controllers\Controller;
```

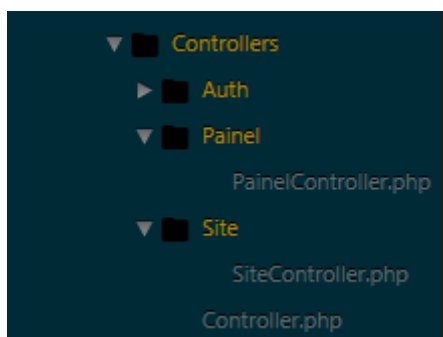
```
class SiteController extends Controller  
{  
    public function index(){  
        return 'Home Page do Site';  
    }  
  
    public function contato(){  
        return 'Pg Contato';  
    }  
  
    public function categoria($id){  
        return "Listagem dos posts da categoria $id";  
    }  
  
    public function categoriaOp($id = 1){  
        return "Listagem dos posts da categoria $id";  
    }  
}
```

No terminal do Laravel, crie o controller `PainelController` dentro da pasta “Painel” entrando com o comando:

`php artisan make:controller Painel\\PainelController`



```
1. laravel53-basico  
C:\laragon\www\laravel53-basico  
λ php artisan make:controller Painel\\PainelController  
Controller created successfully.  
C:\laragon\www\laravel53-basico  
λ
```

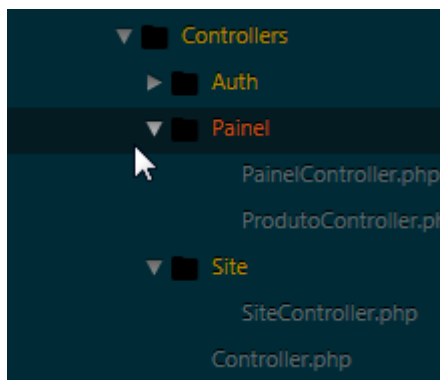


## Aula 7 – Controllers Resource

No terminal do Laravel, crie o controller `ProdutoController` dentro da pasta “Painel” entrando com o comando:

```
php artisan make:controller Painel\\ProdutoController --resource
```

```
C:\laragon\www\laravel53-basico
λ php artisan make:controller Painel\\ProdutoController --resource
Controller created successfully.
```



### web.php

```
<?php
```

```
Route::resource('/painel/produtos','Painel\ProdutoController');
```

### ProdutoController.php

```
<?php
```

```
namespace App\Http\Controllers\Painel;
```

```
use Illuminate\Http\Request;
```

```
use App\Http\Controllers\Controller;
```

```
class ProdutoController extends Controller
```

```
{
```

```
    /**
```

```
     * Display a listing of the resource.
```

```
     *
```

```
     * @return \Illuminate\Http\Response
```

```
     */
```

```

public function index()
{
    return "Listagem dos produtos";
}

/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    //
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    //
}

/**
 * Display the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    //
}

/**
 * Show the form for editing the specified resource.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit($id)
{
    //
}

```

```

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, $id)
{
    //
}

/**
 * Remove the specified resource from storage.
 *
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    //
}
}

```



## Aula 8 – Views

Na pasta View crie duas pastas: painel e site. E dentro da pasta site criar uma pasta chamada: home. Dentro desta pasta crie um arquivo chamado teste.blade.php.

### SiteController.php

```
<?php

namespace App\Http\Controllers\Site;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;

class SiteController extends Controller
{

    public function __construct(){
        // $this -> middleware('auth');
        // $this -> middleware('auth') -> only(['contato', 'categoria']);
        // $this -> middleware('auth') -> except(['contato', 'index']);
    }

    public function index(){

        $teste = 123;
        $teste2 = 321;
        $teste3 = 132;

        return view('site.home.teste', compact('teste', 'teste2', 'teste3'));
    }

    public function contato(){
        return 'Pg Contato';
    }

    public function categoria($id){
        return "Listagem dos posts da categoria $id";
    }

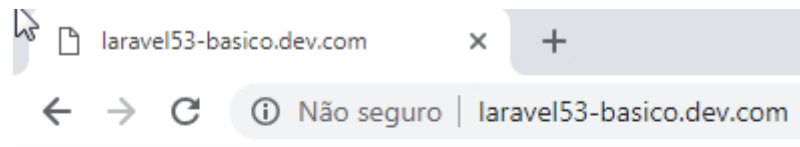
    public function categoriaOp($id = 1){
        return "Listagem dos posts da categoria $id";
    }

}
```

teste.blade.php

<h1>Home Page do Site</h1>

{{ \$teste }} - {{ \$teste2 }} - {{ \$teste3 }}



# Home Page do Site

123 - 321 - 132

## Aula 9 – Sistema de Template Blade

### SiteController.php

```
<?php

namespace App\Http\Controllers\Site;

use Illuminate\Http\Request;

use App\Http\Controllers\Controller;

class SiteController extends Controller
{

    public function __construct(){
        // $this->middleware('auth');
        // $this->middleware('auth')->only(['contato', 'categoria']);
        // $this->middleware('auth')->except(['contato', 'index']);
    }

    public function index(){

        $title = 'Título Teste';
        $xss = '<script>alert("Ataque XSS")</script>';
        $teste = 123;
        $teste2 = 321;
        $teste3 = 132;

        return view('site.home.index', compact('teste', 'teste2', 'teste3', 'title', 'xss'));
    }

    public function contato(){
        return view('site.contact.index');
    }

    public function categoria($id){
        return "Listagem dos posts da categoria $id";
    }

    public function categoriaOp($id = 1){
        return "Listagem dos posts da categoria $id";
    }

}
```

### **template.blade.php**

```
<!DOCTYPE html>
<html>
<head>
    <title>{{ $title ?? 'Curso de Laravel 5.3 - EspecializaTI' }}</title>
</head>
<body>
    @yield('content')
</body>
</html>
```

### **index.blade.php (home)**

```
@extends('site.templates.template1')
```

```
@section('content')
```

```
    <h1>Home Page do Site</h1>
```

```
    {{ $var1 ?? 'Não existe' }}
```

```
    <br />
```

```
    {{ $teste2 ?? 'Não existe' }}
```

```
    <br />
```

```
    {{ $xss }}
```

```
    <br />
```

```
    {!! $xss !!}
```

```
@endsection
```

Na pasta View crie uma pasta chamada: contact. E dentro desta pasta crie uma pasta um arquivo chamado index.blade.php.

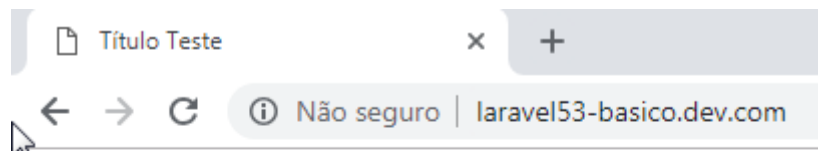
### **index.blade.php (contact)**

```
@extends('site.templates.template1')
```

```
@section('content')
```

```
    <h1>Página de Contato</h1>
```

```
@endsection
```



# Home Page do Site

Não existe

321

<script>alert("Ataque XSS")</script>

## Aula 10 – Sistema de Template Blade (Final)

Em views, na pasta site, cria uma pasta chamada includes e dentro desta pasta um arquivo chamado sidebar.blade.php

### SiteController.php

```
<?php

namespace App\Http\Controllers\Site;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;

class SiteController extends Controller
{

    public function index(){

        $title = 'Título Teste';
        $xss = '<script>alert("Ataque XSS")</script>';
        $var1 = '123';
        // $arrayData = [1,2,3,4,5,6,7,8,9];
        $arrayData = [];
        $teste = 123;
        $teste2 = 321;
        $teste3 = 132;

        return view('site.home.index', compact('teste', 'teste2', 'teste3','title', 'xss', 'var1',
'arrayData'));
    }

    public function contato(){
        return view('site.contact.index');
    }

    public function categoria($id){
        return "Listagem dos posts da categoria $id";
    }

    public function categoriaOp($id = 1){
        return "Listagem dos posts da categoria $id";
    }
}
```

## index.blade.php (home)

```
@extends('site.templates.template1')
```

```
@section('content')
```

```
<h1>Home Page do Site</h1>
```

```
{{ $var1 ?? 'Não existe' }}
```

```
<br />
```

```
{{ $teste2 ?? 'Não existe' }}
```

```
<br />
```

```
{{ $xss }}
```

```
<br />
```

```
<!-- {!! $xss !!} -->
```

```
@if($var1=='1234')
```

```
<p>É igual</p>
```

```
@else
```

```
<p>É diferente</p>
```

```
@endif
```

```
@unless($var1=='1234')
```

```
<p>Não é igual...unless</p>
```

```
@endunless
```

```
@for($i = 0; $i < 10; $i++)
```

```
<p>For: {!! $i }</p>
```

```
@endfor
```

```
{{--
```

```
@if(count($arrayData) > 0)
```

```
@foreach($arrayData as $array)
```

```
<p>Foreach: {!! $array }</p>
```

```
@endforeach
```

```
@else
```

```
<p>Não existem itens para serem impressos</p>
```

```
@endif
```

```
--}}
```

```
@forelse($arrayData as $array)
```

```
<p>Forelse: {!! $array }</p>
```

```
@empty
```

```
<p>Não existem itens para serem impressos</p>
```

```
@endforelse
```

@php

@endphp

**@include('site.includes.sidebar', compact('var1'))**

@endsection

**sidebar.blade.php**

#sidebar - teste

{{ \$var1 ?? '' }}



## Aula 11 – Blade @stacks

### template1.blade.php

```
<!DOCTYPE html>
<html>
<head>
    <title>{{ $title ?? 'Curso de Laravel 5.3 - EspecializaTI' }}</title>
</head>
<body>
    @yield('content')
    @stack('scripts')
</body>
</html>
```

### index.blade.php (home)

```
@extends('site.templates.template1')
```

```
@section('content')
```

```
    <h1>Home Page do Site</h1>
```

```
    {{ $var1 ?? 'Não existe' }}
    <br />
    {{ $teste2 ?? 'Não existe' }}
    <br />
    {{ $xss }}
    <br />
    <!-- {!! $xss !!} -->
```

```
@if($var1=='1234')
    <p>É igual</p>
@else
    <p>É diferente</p>
@endif
```

```
@unless($var1=='1234')
    <p>Não é igual...unless</p>
@endunless
```

```
@for($i = 0; $i < 10; $i++)
    <p>For: {{ $i }}</p>
```

@endfor

{{--

@if(count(\$arrayData) > 0)

    @foreach(\$arrayData as \$array)

        <p>Foreach: {{\$array}}</p>

    @endforeach

@else

    <p>Não existem itens para serem impressos</p>

@endif

--}}

@forelse(\$arrayData as \$array)

    <p>Forelse: {{\$array}}</p>

@empty

    <p>Não existem itens para serem impressos</p>

@endforelse

@php

@endphp

@include('site.includes.sidebar', compact('var1'))

**@push('scripts')**

    <!-- Latest compiled and minified CSS -->

    <link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"

integrity="sha384-

BVYiISiFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"

crossorigin="anonymous">

**@endpush**

@endsection

# Home Page do Site

123

321

<script>alert("Ataque XSS")</script>

É diferente

Não é igual...unless

For: 0

For: 1

For: 2

For: 3

For: 4

For: 5

For: 6

For: 7

For: 8

For: 9

Não existem itens para serem impressos

#sidebar - teste 123

## Aula 12 – Migrations

### Criando o banco de dados

No terminal:

**mysql -u root -p**

```
C:\laragon\www\laravel53-basico
λ mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 12
Server version: 10.2.3-MariaDB-log mariadb.org binary distribution

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| curso-laravel53 |
| information_schema |
| mysql |
| performance_schema |
| primeiro |
| segundo |
| test |
+-----+
7 rows in set (0.00 sec)

MariaDB [(none)]> |
```

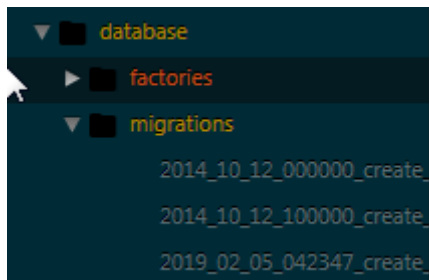
Crie um novo banco de dados chamado “curso-laravel53”. Se preferir, você pode usar o phpmyadmin para criar esse banco de dados.

Crie uma nova Migration com o nome produtos

**php artisan make:migration create\_produtos\_table**

```
1. laravel53-basico

C:\laragon\www\laravel53-basico
λ php artisan make:migration create_produtos_table
Created Migration: 2019_02_05_042347_create_produtos_table
|
```



## Renomeando a tabela

Renomeie a tabela de “produtos” para “products”.

### 2019\_02\_05\_042437\_create\_products\_table.php

```
<?php
```

```
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
```

```
class CreateProductsTable extends Migration
{
    public function up()
    {
        Schema::create('products', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name', 150);
            $table->integer('number');
            $table->boolean('active');
            $table->string('image',200);
            $table->enum('category',['eletronicos','moveis','limpeza','banho']);
            $table->text('description');
            $table->timestamps();
        });
    }
    public function down()
    {
        Schema::dropIfExists('products');
    }
}
```

No terminal entre com o comando:

**composer dump-autoload**

```

C:\laragon\www\laravel53-basico
λ composer dump-autoload
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: beyondcode/laravel-dump-server
Discovered Package: fideloper/proxy
Discovered Package: laravel/nexmo-notification-channel
Discovered Package: laravel/slack-notification-channel
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.

C:\laragon\www\laravel53-basico
λ |

```

Para criar as tabelas, entre no terminal com o comando:

**php artisan migrate**

```

C:\laragon\www\laravel53-basico
λ php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table
Migrating: 2019_02_05_042347_create_products_table
Migrated: 2019_02_05_042347_create_products_table

C:\laragon\www\laravel53-basico
λ |

```

Laragon

curso-laravel53	64,0 KiB
migrations	16,0 KiB
password_resets	16,0 KiB
products	16,0 KiB
users	16,0 KiB

Colunas: Adicionar Remover Mover para cima Mover para baixo

#	Nome	Tipo de dados	Tamanho/It...	Unsign...	Permitir...	Zerofill	Padrão	Comentário	Colaço
1	id	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT		
2	name	VARCHAR	150	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nenhum padrão		utf8mb4_unicode_ci
3	number	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nenhum padrão		
4	active	TINYINT	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nenhum padrão		
5	image	VARCHAR	200	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nenhum padrão		utf8mb4_unicode_ci
6	category	ENUM	'eletronicos',...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nenhum padrão		utf8mb4_unicode_ci
7	description	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nenhum padrão		utf8mb4_unicode_ci
8	created_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
9	updated_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		

## Aula 13 – Migration Refresh

### Atualizando dados da tabela

Vamos tornar o preenchimento do campo image não obrigatório.

**2019\_02\_05\_042437\_create\_products\_table.php**

```
<?php
```

```
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;
```

```
class CreateProductsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('products', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name', 150);
            $table->integer('number');
            $table->boolean('active');
            $table->string('image',200)->nullable();
            $table->enum('category',['eletronicos','moveis','limpeza','banho']);
            $table->text('description');
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('products');
    }
}
```

Para atualizar, no terminal, entre com o seguinte comando:

**php artisan migrate:refresh**

As tabelas serão deletadas e recriadas com a nova estrutura.

```
1. laravel53-basico
C:\laragon\www\laravel53-basico
λ php artisan migrate:refresh
Rolling back: 2019_02_05_042347_create_products_table
Rolled back: 2019_02_05_042347_create_products_table
Rolling back: 2014_10_12_100000_create_password_resets_table
Rolled back: 2014_10_12_100000_create_password_resets_table
Rolling back: 2014_10_12_000000_create_users_table
Rolled back: 2014_10_12_000000_create_users_table
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table
Migrating: 2019_02_05_042347_create_products_table
Migrated: 2019_02_05_042347_create_products_table

C:\laragon\www\laravel53-basico
λ |
```

Colunas: <span>Adicionar</span> <span>Remover</span> <span>Mover para cima</span> <span>Mover para baixo</span>									
#	Nome	Tipo de dados	Tamanho/It...	Unsign...	Permitir...	Zerofill	Padrão	Comentário	Colaço
1	id	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT		
2	name	VARCHAR	150	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nenhum padrão		utf8mb4_unicode_ci
3	number	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nenhum padrão		
4	active	TINYINT	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nenhum padrão		
5	image	VARCHAR	200	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci
6	category	ENUM	'eletronicos',...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nenhum padrão		utf8mb4_unicode_ci
7	description	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Nenhum padrão		utf8mb4_unicode_ci
8	created_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		
9	updated_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		

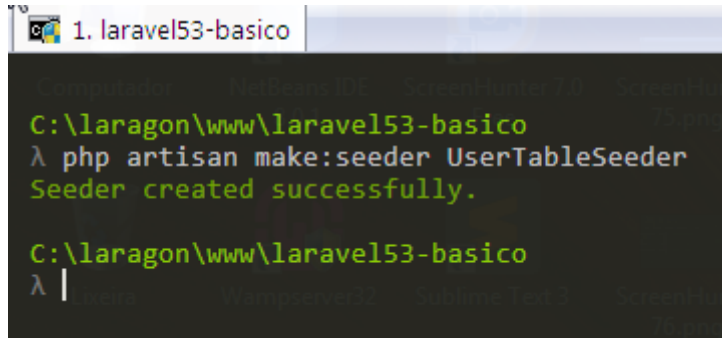


## Aula 14 – Seeders

Para utilizar o sistema de autenticação (login) é necessário ter pelo menos 1 usuário cadastrado.

No terminal entre com o seguinte comando:

**php artisan make:seeder UserTableSeeder**

A screenshot of a terminal window with a dark background. The title bar at the top shows a file explorer icon and the text '1. laravel53-basico'. The terminal content shows the command prompt 'C:\laragon\www\laravel53-basico' followed by the command 'php artisan make:seeder UserTableSeeder'. The output of the command is 'Seeder created successfully.' followed by a new command prompt 'C:\laragon\www\laravel53-basico' and a cursor.

### UserTableSeeder.php

```
<?php
```

```
use Illuminate\Database\Seeder;
```

```
use App\User;
```

```
class UserTableSeeder extends Seeder
```

```
{
```

```
    public function run()
```

```
    {
```

```
        User::create([
```

```
            'name' => 'Carlos Ferreira',
```

```
            'email' => 'carlos@especializati.com.br',
```

```
            'password' => bcrypt('123456')
```

```
        ]);
```

```
    }
```

```
}
```

## DatabaseSeeder.php

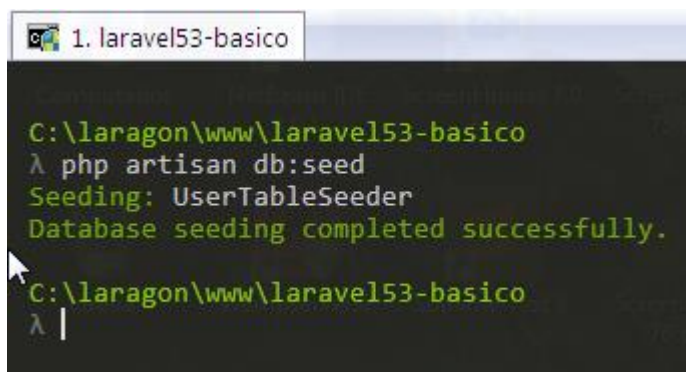
```
<?php
```

```
use Illuminate\Database\Seeder;
```

```
class DatabaseSeeder extends Seeder
{
    public function run()
    {
        $this->call(UserTableSeeder::class);
    }
}
```

No terminal entre com o seguinte comando:

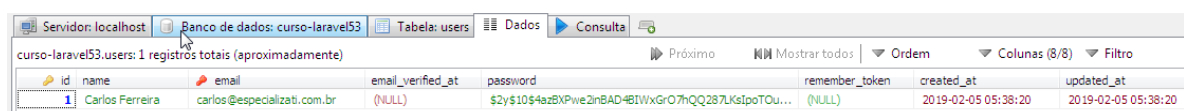
**php artisan db:seed**



```
1. laravel53-basico

C:\laragon\www\laravel53-basico
λ php artisan db:seed
Seeding: UserTableSeeder
Database seeding completed successfully.

C:\laragon\www\laravel53-basico
λ |
```



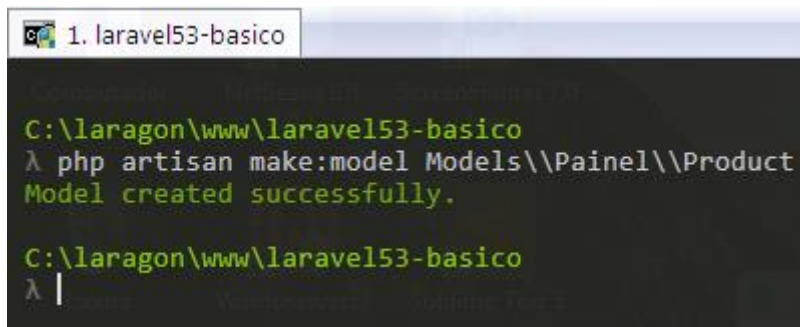
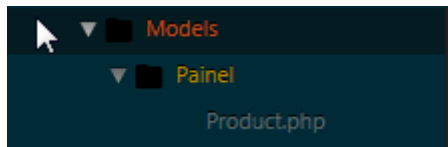
curso-laravel53.users: 1 registros totais (aproximadamente)

id	name	email	email_verified_at	password	remember_token	created_at	updated_at
1	Carlos Ferreira	carlos@especializati.com.br	(NULL)	\$2y\$10\$4azbXPwe2inBAD#BIWxGrO7hQQ287UKsIpTOu...	(NULL)	2019-02-05 05:38:20	2019-02-05 05:38:20

## Aula 15 – Models

No terminal do Laravel, crie o model **Product** dentro da pasta “Models” e dentro de uma pasta chamada “Painel”, entrando com o comando:

```
php artisan make:model Models\\Painel\\Product
```

A screenshot of a terminal window titled "1. laravel53-basico". The prompt is "C:\laragon\www\laravel53-basico". The command entered is "php artisan make:model Models\\Painel\\Product". The output is "Model created successfully." followed by a new prompt "C:\laragon\www\laravel53-basico" and a cursor.

### Product.php

```
<?php

namespace App\Models\Painel;

use Illuminate\Database\Eloquent\Model;

class Product extends Model
{
    //
}
```

Eloquent é uma classe para manipulação de banco de dados.

```
php artisan make:model Models\\Painel\\Product -m
```

Com esse comando é criado o model e a migration.

## ProductController.php

```
<?php

namespace App\Http\Controllers\Painel;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Models\Painel\Product;

class ProdutoController extends Controller
{
    public function index(Product $product)
    {
        // $product = new Product;
        $products = $product->all();

        // return "Listagem dos produtos";
        return view('painel.products.index', compact('products'));
    }
    public function create()
    {
        //
    }
    public function store(Request $request)
    {
        //
    }
    public function show($id)
    {
        //
    }
    public function edit($id)
    {
        //
    }
    public function update(Request $request, $id)
    {
        //
    }
    public function destroy($id)
    {
        //
    }
}
```

## index.blade.php (products)

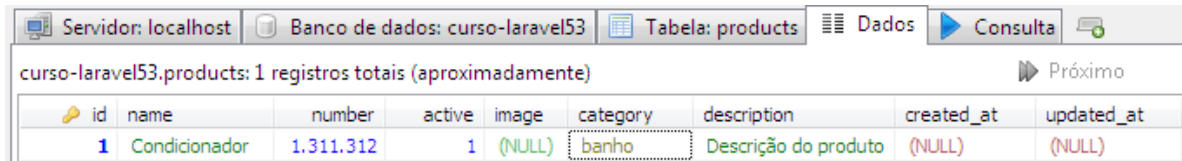
<h1>Listagem dos produtos</h1>

```
<table>
  <tr>
    <th>Nome</th>
    <th>Descrição</th>
  </tr>

  @foreach($products as $product)
    <tr>
      <td>{{$product->name}}</td>
      <td>{{$product->description}}</td>
    </tr>
  @endforeach

</table>
```

Na tabela **products** inserir manualmente o seguinte registro:



The screenshot shows a database management interface. At the top, there are tabs for 'Servidor: localhost', 'Banco de dados: curso-laravel53', and 'Tabela: products'. Below these, it says 'curso-laravel53.products: 1 registros totais (aproximadamente)'. A table with 10 columns is displayed: id, name, number, active, image, category, description, created\_at, and updated\_at. The first row contains the values: 1, Condicionador, 1.311.312, 1, (NULL), banho, Descrição do produto, (NULL), and (NULL). Below the table, there is a browser window showing the URL 'laravel53-basico.dev.com/painel/' and a search bar with the text 'laravel53-basico.dev.com/painel/produtos'.

id	name	number	active	image	category	description	created_at	updated_at
1	Condicionador	1.311.312	1	(NULL)	banho	Descrição do produto	(NULL)	(NULL)

## Listagem dos produtos

Nome	Descrição
Condicionador	Descrição do produto

## Aula 16 – Primeiro cadastro no banco de dados

### web.php

```
<?php

Route::get('/painel/produtos/tests', 'Painel\ProdutoController@tests');

Route::resource('/painel/produtos', 'Painel\ProdutoController');

Route::group(['namespace' => 'Site'], function(){
    Route::get('/', 'SiteController@index');
    Route::get('/contato', 'SiteController@contato');
    Route::get('/categoria/{id}', 'SiteController@categoria');
    Route::get('/categoria2/{id?}', 'SiteController@categoriaOp');
});
```

### ProdutoController.php

```
<?php

namespace App\Http\Controllers\Painel;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Models\Painel\Product;

class ProdutoController extends Controller
{
    private $product;

    public function __construct(Product $product){
        $this->product = $product;
    }

    public function index()
    {
        $products = $this->product->all();

        return view('painel.products.index', compact('products'));
    }
}
```

```

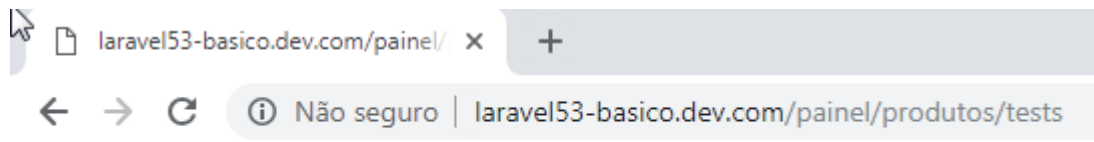
public function create()
{
    //
}
public function store(Request $request)
{
    //
}
public function show($id)
{
    //
}
public function edit($id)
{
    //
}
public function update(Request $request, $id)
{
    //
}
public function destroy($id)
{
    //
}
public function tests(){
    // return 'tests';
    // $this->product->insert();

    $prod = $this->product;

    $prod->name="Nome do produto";
    $prod->number=131231;
    $prod->active=true;
    $prod->category='eletronicos';
    $prod->description='Descrição do produto';
    $insert = $prod->save();

    if($insert){
        return 'Inserido com sucesso!';
    } else {
        return 'Falha ao inserir!';
    }
}
}

```



Inserido com sucesso!

Servidor: localhost Banco de dados: curso-laravel53 Tabela: products Dados Consulta								
curso-laravel53.products: 2 registros totais (aproximadamente)								
Próximo Mostrar todos Ordem								
id	name	number	active	image	category	description	created_at	updated_at
1	Condicionador	1.311.312	1	(NULL)	banho	Descrição do produto	(NULL)	(NULL)
2	Nome do produto	131.231	1	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:29:16	2019-02-08 17:29:16

```
public function tests(){

    $insert = $this->product->insert([
        'name'      => 'Nome do produto 2',
        'number'    => 434435,
        'active'    => false,
        'category'  => 'eletronicos',
        'description' => 'Descrição do produto'
    ]);

    if($insert){
        return 'Inserido com sucesso!';
    } else {
        return 'Falha ao inserir!';
    }
}
```

Servidor: localhost Banco de dados: curso-laravel53 Tabela: products Dados Consulta								
curso-laravel53.products: 3 registros totais (aproximadamente)								
Próximo Mostrar todos								
id	name	number	active	image	category	description	created_at	updated_at
1	Condicionador	1.311.312	1	(NULL)	banho	Descrição do produto	(NULL)	(NULL)
2	Nome do produto	131.231	1	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:29:16	2019-02-08 17:29:16
3	Nome do produto 2	434.435	0	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:32:03	2019-02-08 17:32:03



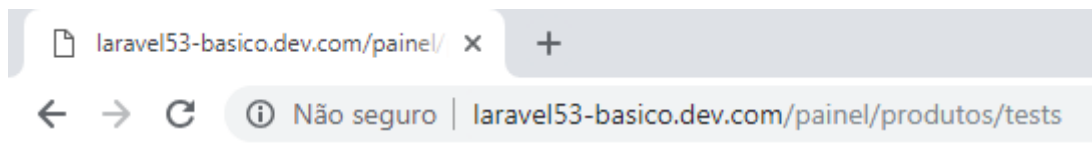
```

public function tests(){

    $insert = $this->product->create([
        'name'      => 'Nome do produto 2',
        'number'    => 434435,
        'active'    => false,
        'category'  => 'eletronicos',
        'description' => 'Descrição do produto'
    ]);

    if($insert){
        return "Inserido com sucesso, ID: {$insert -> id}";
    } else {
        return 'Falha ao inserir!';
    }
}

```



Servidor: localhost

Banco de dados: curso-laravel53

Tabela: products

Dados

Consulta

curso-laravel53.products: 6 registros totais (aproximadamente)

Próximo

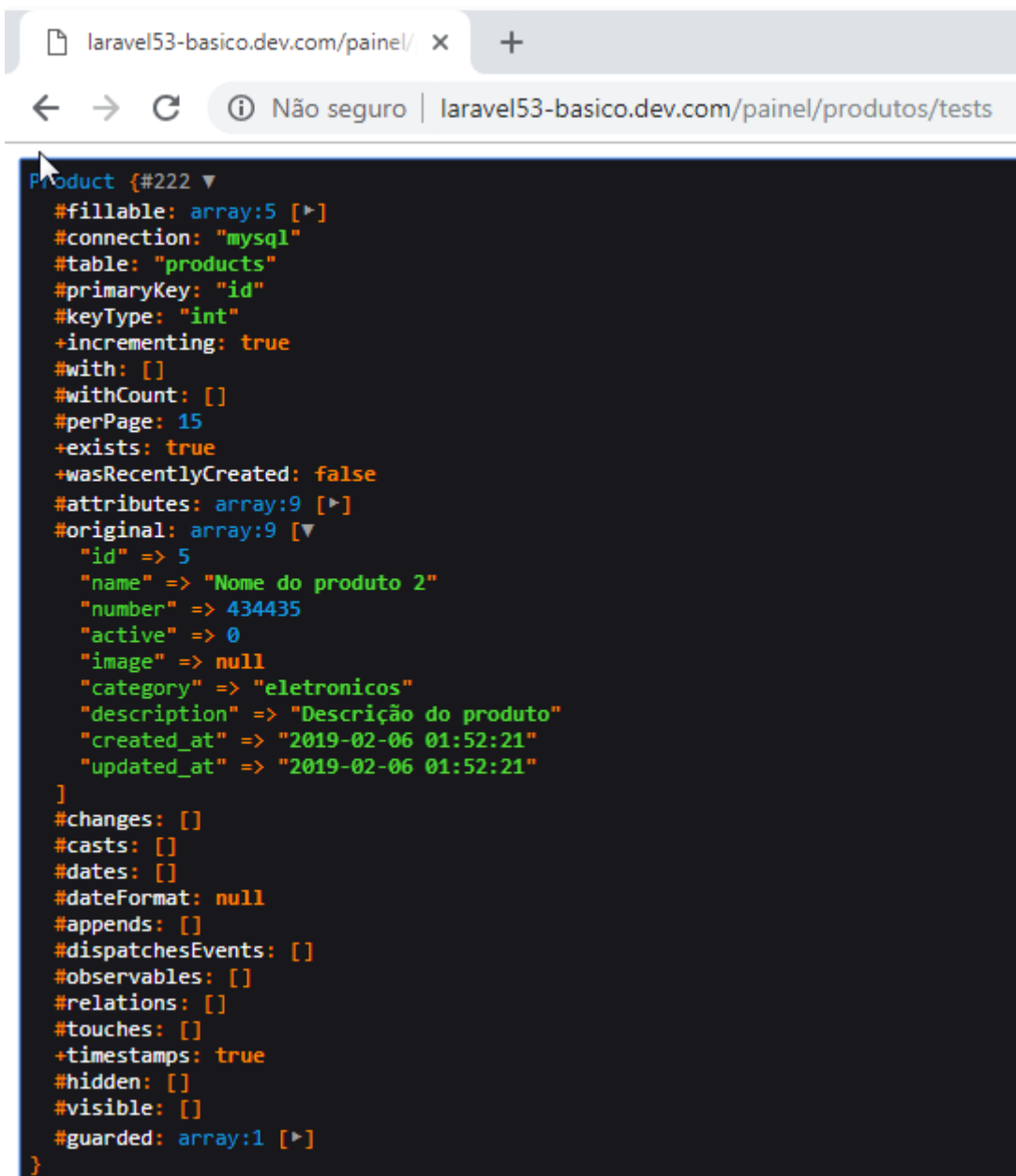
Mostrar todos

id	name	number	active	image	category	description	created_at	updated_at
1	Condicionador	1.311.312	1	(NULL)	banho	Descrição do produto	(NULL)	(NULL)
2	Nome do produto	131.231	1	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:29:16	2019-02-08 17:29:16
3	Nome do produto 2	434.435	0	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:32:03	2019-02-08 17:32:03
4	Nome do produto 2	434.435	0	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:35:52	2019-02-08 17:35:52
5	Nome do produto 2	434.435	0	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:36:32	2019-02-08 17:36:32
6	Nome do produto 2	434.435	0	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:37:06	2019-02-08 17:37:06

## Aula 17 – Update

```
public function tests(){  
  
    $prod = $this->product->find(5);  
    dd($prod);  
  
}
```

find() retorna um item pelo seu id.



The screenshot shows a web browser window with the address bar displaying 'laravel53-basico.dev.com/painel/' and a sub-page 'produtos/tests'. The page content is a JSON object representing a product, with a dark background and syntax highlighting. The JSON object includes metadata like '#fillable', '#connection', and '#table', as well as product details like 'name', 'number', 'category', and 'description'.

```
Product {#222 ▾  
  #fillable: array:5 [▸]  
  #connection: "mysql"  
  #table: "products"  
  #primaryKey: "id"  
  #keyType: "int"  
  +incrementing: true  
  #with: []  
  #withCount: []  
  #perPage: 15  
  +exists: true  
  +wasRecentlyCreated: false  
  #attributes: array:9 [▸]  
  #original: array:9 [▾  
    "id" => 5  
    "name" => "Nome do produto 2"  
    "number" => 434435  
    "active" => 0  
    "image" => null  
    "category" => "eletronicos"  
    "description" => "Descrição do produto"  
    "created_at" => "2019-02-06 01:52:21"  
    "updated_at" => "2019-02-06 01:52:21"  
  ]  
  #changes: []  
  #casts: []  
  #dates: []  
  #dateFormat: null  
  #appends: []  
  #dispatchesEvents: []  
  #observables: []  
  #relations: []  
  #touches: []  
  +timestamps: true  
  #hidden: []  
  #visible: []  
  #guarded: array:1 [▸]  
}
```

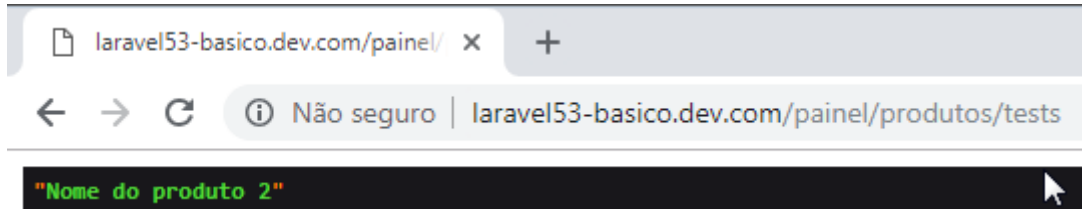
```

public function tests(){

    $prod = $this->product->find(5);
    dd($prod->name);

}

```



```

public function tests(){

    $prod = $this->product->find(5);
    $prod->name="Update";
    $prod->number=79789;
    $prod->active=true;
    $prod->category='eletronicos';
    $prod->description='Desc. Update';
    $update = $prod->save();

    if($update){
        return "Alterado com sucesso!";
    } else {
        return 'Falha ao alterar!';
    }

}

```

Servidor: localhost Banco de dados: curso-laravel53 Tabela: products Dados Consulta

curso-laravel53.products: 6 registros totais (aproximadamente) Próximo Mostrar todos

id	name	number	active	image	category	description	created_at	updated_at
1	Condicionador	1.311.312	1	(NULL)	banho	Descrição do produto	(NULL)	(NULL)
2	Nome do produto	131.231	1	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:29:16	2019-02-08 17:29:16
3	Nome do produto 2	434.435	0	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:32:03	2019-02-08 17:32:03
4	Nome do produto 2	434.435	0	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:35:52	2019-02-08 17:35:52
5	Update	79.789	0	(NULL)	eletronicos	Desc.Update	2019-02-08 17:36:32	2019-02-08 17:39:45
6	Nome do produto 2	434.435	0	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:37:06	2019-02-08 17:37:06

```

public function tests(){
    $prod = $this->product->find(5);
    $prod->name="Update 2";
    $prod->number=797890;
    $update = $prod->save();

    if($update){
        return "Alterado com sucesso!";
    } else {
        return 'Falha ao alterar!';
    }
}

```

Servidor: localhost

Banco de dados: curso-laravel53

Tabela: products

Dados

Consulta

curso-laravel53.products: 6 registros totais (aproximadamente)

Próximo

Mostrar todos

id	name	number	active	image	category	description	created_at	updated_at
1	Condicionador	1.311.312	1	(NULL)	banho	Descrição do produto	(NULL)	(NULL)
2	Nome do produto	131.231	1	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:29:16	2019-02-08 17:29:16
3	Nome do produto 2	434.435	0	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:32:03	2019-02-08 17:32:03
4	Nome do produto 2	434.435	0	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:35:52	2019-02-08 17:35:52
5	Update 2	797.890	0	(NULL)	eletronicos	Desc.Update	2019-02-08 17:36:32	2019-02-08 17:41:43
6	Nome do produto 2	434.435	0	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:37:06	2019-02-08 17:37:06

```

public function tests(){

    $prod = $this->product->find(6);
    $update = $prod->update([
        'name'      => 'Update test',
        'number'    => 6765756,
        'active'    => true
    ]);

    if($update){
        return "Alterado com sucesso!";
    } else {
        return 'Falha ao alterar!';
    }
}

```

Servidor: localhost Banco de dados: curso-laravel53 Tabela: products Dados Consulta

curso-laravel53.products: 6 registros totais (aproximadamente) Próximo Mostrar todos

id	name	number	active	image	category	description	created_at	updated_at
1	Condicionador	1.311.312	1	(NULL)	banho	Descrição do produto	(NULL)	(NULL)
2	Nome do produto	131.231	1	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:29:16	2019-02-08 17:29:16
3	Nome do produto 2	434.435	0	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:32:03	2019-02-08 17:32:03
4	Nome do produto 2	434.435	0	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:35:52	2019-02-08 17:35:52
5	Update 2	797.890	0	(NULL)	eletronicos	Desc.Update	2019-02-08 17:36:32	2019-02-08 17:41:43
6	Update test	6.765.756	1	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:37:06	2019-02-08 17:43:43

```

public function tests(){
    $prod = $this->product->where('number',6765756);
    $update = $prod->update([
        'name'      => 'Update test 2',
        'number'    => 67657560,
        'active'    => false
    ]);

    if($update){
        return "Alterado com sucesso 2!";
    } else {
        return 'Falha ao alterar!';
    }
}

```

Servidor: localhost Banco de dados: curso-laravel53 Tabela: products Dados Consulta

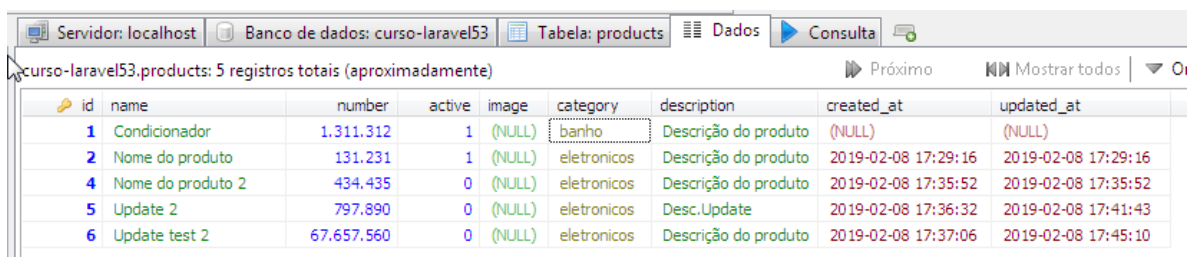
curso-laravel53.products: 6 registros totais (aproximadamente) Próximo Mostrar todos Ordem

id	name	number	active	image	category	description	created_at	updated_at
1	Condicionador	1.311.312	1	(NULL)	banho	Descrição do produto	(NULL)	(NULL)
2	Nome do produto	31.231	1	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:29:16	2019-02-08 17:29:16
3	Nome do produto 2	434.435	0	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:32:03	2019-02-08 17:32:03
4	Nome do produto 2	434.435	0	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:35:52	2019-02-08 17:35:52
5	Update 2	797.890	0	(NULL)	eletronicos	Desc.Update	2019-02-08 17:36:32	2019-02-08 17:41:43
6	Update test 2	67.657.560	0	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:37:06	2019-02-08 17:45:10

## Aula 18 – Delete

```
public function tests(){
    $prod = $this->product->find(3);
    $delete = $prod->delete();

    if($delete){
        return "Deletado com sucesso!";
    } else {
        return 'Falha ao deletar!';
    }
}
```



curso-laravel53.products: 5 registros totais (aproximadamente)

id	name	number	active	image	category	description	created_at	updated_at
1	Condicionador	1.311.312	1	(NULL)	banho	Descrição do produto	(NULL)	(NULL)
2	Nome do produto	131.231	1	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:29:16	2019-02-08 17:29:16
4	Nome do produto 2	434.435	0	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:35:52	2019-02-08 17:35:52
5	Update 2	797.890	0	(NULL)	eletronicos	Desc.Update	2019-02-08 17:36:32	2019-02-08 17:41:43
6	Update test 2	67.657.560	0	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:37:06	2019-02-08 17:45:10

O registro de id 3 também poderia ser deletado da seguinte forma:

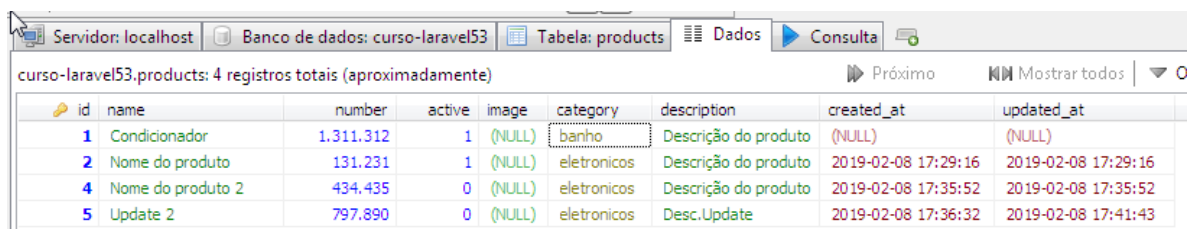
```
public function tests(){
    $prod = $this->product->destroy(3);
}
```

Os registros de id 3, 4 e 5 podem ser deletados da seguinte forma:

```
public function tests(){
    $prod = $this->product->destroy([3, 4, 5]);
}
```

Deletando pelo número (number):

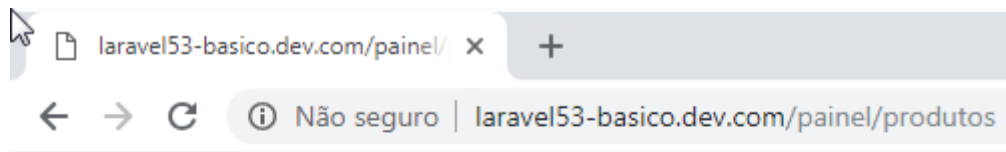
```
public function tests(){
    $delete = $this->product->where('number',67657560)->delete();
}
```



curso-laravel53.products: 4 registros totais (aproximadamente)

id	name	number	active	image	category	description	created_at	updated_at
1	Condicionador	1.311.312	1	(NULL)	banho	Descrição do produto	(NULL)	(NULL)
2	Nome do produto	131.231	1	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:29:16	2019-02-08 17:29:16
4	Nome do produto 2	434.435	0	(NULL)	eletronicos	Descrição do produto	2019-02-08 17:35:52	2019-02-08 17:35:52
5	Update 2	797.890	0	(NULL)	eletronicos	Desc.Update	2019-02-08 17:36:32	2019-02-08 17:41:43

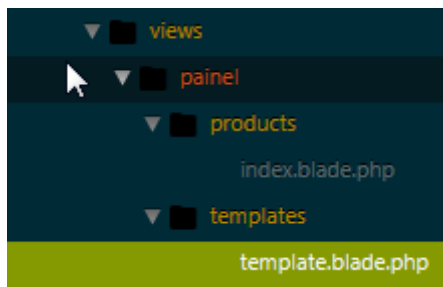
## Aula 19 – Melhorar a listagem dos itens (CSS)



### Listagem dos produtos

Nome	Descrição
Condicionador	Descrição do produto
Nome do produto	Descrição do produto
Nome do produto 2	Descrição do produto
Update 2	Desc. Update

- Em views/painel criar a pasta templates. E dentro dessa pasta criar o arquivo template.blade.php



## template.blade.php

```
<!DOCTYPE html>
<html>
<head>
    <title>{{ $title ?? 'Painel curso' }}</title>
    <!-- Latest compiled and minified CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="{{ url('assets/painel/css/style.css') }}">
</head>
<body>

    <div class="container">
        @yield('content')
    </div>

</body>
</html>
```

## Em **ProdutoController.php**

```
public function index()
{

    $title = "Listagem dos produtos";
    $products = $this->product->all();

    return view('painel.products.index', compact('products', 'title'));
}
```



## index.blade.php

```
@extends('painel.templates.template')
```

```
@section('content')
```

```
<h1 class="title-pg">Listagem dos produtos</h1>
```

```
<a href="" class="btn btn-primary btn-add"><span class="glyphicon glyphicon-plus"></span>  
Cadastrar</a>
```

```
<table class="table table-striped">
```

```
<tr>
```

```
<th>Nome</th>
```

```
<th>Descrição</th>
```

```
<th width="100px">Ações</th>
```

```
</tr>
```

```
@foreach($products as $product)
```

```
<tr>
```

```
<td>{{ $product->name }}</td>
```

```
<td>{{ $product->description }}</td>
```

```
<td>
```

```
<a href="" class="actions edit">
```

```
<span class="glyphicon glyphicon-pencil"></span>
```

```
</a>
```

```
<a href="" class="actions delete">
```

```
<span class="glyphicon glyphicon-trash"></span>
```

```
</a>
```

```
</td>
```

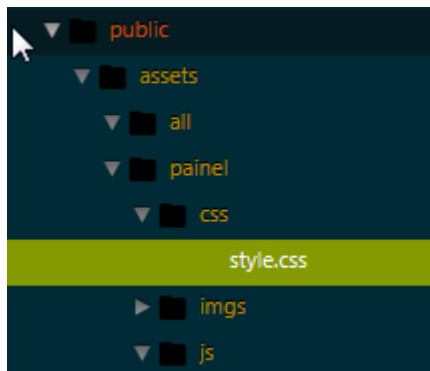
```
</tr>
```

```
@endforeach
```

```
</table>
```

```
@endsection
```

- Dentro da pasta public criar as seguintes subpastas e o arquivo style.css:



### style.css

```
.title-pg{
  font-size: 2em;
  font-weight: 900;
  text-transform: uppercase;
}
.edit{
  background: #242063;
}
.delete{
  background: #b12020;
}
.actions{
  padding: 6px;
  color: #fff;
  text-decoration: none;
  font-size: 1em;
  margin: 0px 2px;
}
.actions:hover{
  text-decoration: none;
  color: #fff;
}
.btn-add{
  margin-bottom: 10px;
}
```

# LISTAGEM DOS PRODUTOS

+
Cadastrar

Nome	Descrição	Ações
Condicionador	Descrição do produto	<div>✎</div> <div>🗑</div>
Nome do produto	Descrição do produto	<div>✎</div> <div>🗑</div>
Nome do produto 2	Descrição do produto	<div>✎</div> <div>🗑</div>
Update 2	Desc.Update	<div>✎</div> <div>🗑</div>

## Aula 20 – Formulários

Em **ProdutoController.php**:

```
public function create()
{
    // return "#form Cad";

    $title = "Cadastrar novo produto";
    $categorys = ['eletronicos','moveis','limpeza','banho'];

    return view('painel.products.create', compact('title','categorys'));
}
public function store(Request $request)
{
    return "Cadastrando...";
}
```

## create.blade.php

```
@extends('painel.templates.template')
```

```
@section('content')
```

```
<h1 class="title-pg">Gestão produto</h1>
```

```
<form class="form" method="post" action="{{route('produtos.store')}}">
```

```
<!-- <input type="hidden" name="_token" value="{{csrf_token()}}"> -->
```

```
{!! csrf_field() !!}
```

```
<div class="form-group">
```

```
<input type="text" name="name" placeholder="Nome" class="form-control">
```

```
</div>
```

```
<div class="form-group">
```

```
<label>
```

```
<input type="checkbox" name="active" value="1">
```

```
Ativo?
```

```
</label>
```

```
</div>
```

```
<div class="form-group">
```

```
<input type="text" name="number" placeholder="Número" class="form-control">
```

```
</div>
```

```
<div class="form-group">
```

```
<select name="category" class="form-control">
```

```
<option>Escolha a categoria</option>
```

```
@foreach($categorys as $category)
```

```
<option value="{{ $category }}">{{ $category }}</option>
```

```
@endforeach
```

```
</select>
```

```
</div>
```

```
<div class="form-group">
```

```
<textarea name="description" placeholder="Descrição" class="form-control"></textarea>
```

```
</div>
```

```
<button class="btn btn-primary">Enviar</button>
```

```
</form>
```

```
@endsection
```

Editar Produto: Shampoo e Cond

Não seguro | laravel53-basico.test/painel/produtos/1/edit

## GESTÃO PRODUTO

Shampoo e Condicionador

☒ Ativo?

123456

banho

Descrição do produto para banho

Enviar

laravel53-basico.dev.com/painel/

Não seguro | laravel53-basico.dev.com/painel/produtos

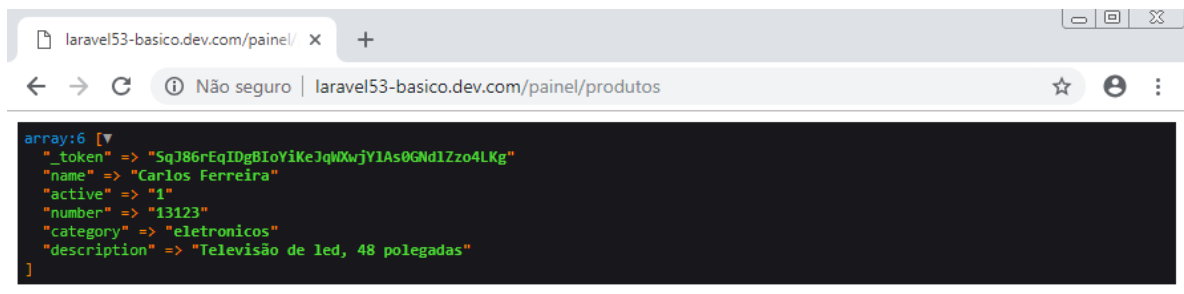
Cadastrando...

## Aula 21 – Requests

Em “ProdutoController.php”:

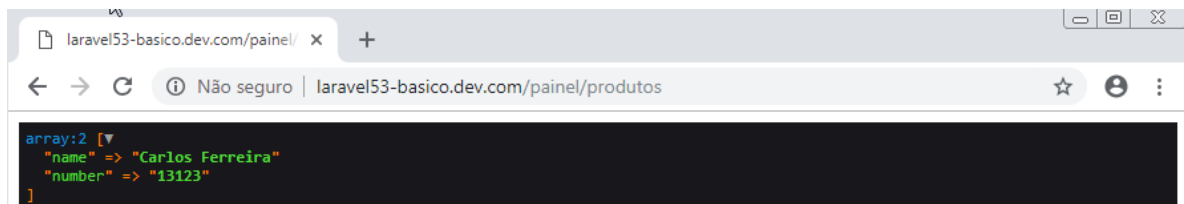
```
public function store(Request $request)
{
    dd($request->all());

    return "Cadastrando...";
}
```



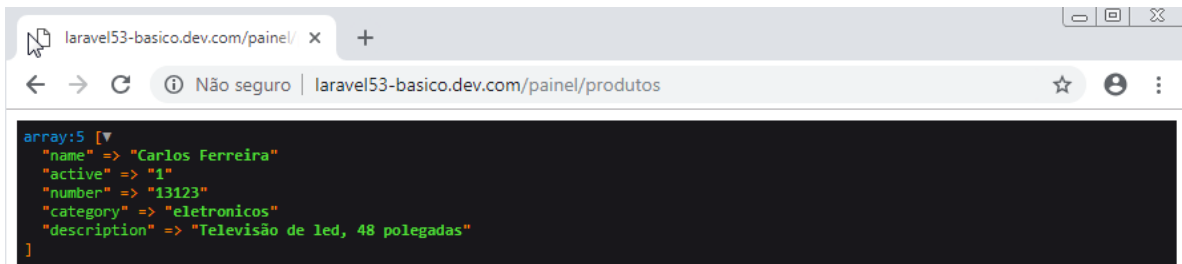
```
public function store(Request $request)
{
    // dd($request->all());
    dd($request->only(['name','number']));

    return "Cadastrando...";
}
```



```
public function store(Request $request)
{
    // dd($request->all());
    // dd($request->only(['name','number']));
    dd($request->except(['_token']));

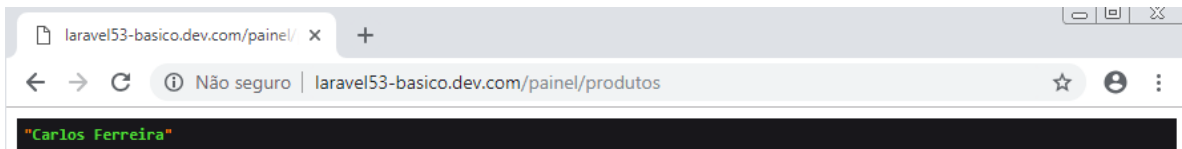
    return "Cadastrando...";
}
```



A screenshot of a web browser window. The address bar shows the URL `laravel53-basico.dev.com/painel/`. The page content displays a JSON array of 5 elements. The first element is an object with the following properties: `"name" => "Carlos Ferreira"`, `"active" => "1"`, `"number" => "43123"`, `"category" => "eletronicos"`, and `"description" => "Televisão de led, 48 polegadas"`.

```
array:5 [▼  
  "name" => "Carlos Ferreira"  
  "active" => "1"  
  "number" => "43123"  
  "category" => "eletronicos"  
  "description" => "Televisão de led, 48 polegadas"  
]
```

```
public function store(Request $request)  
{  
    // dd($request->all());  
    // dd($request->only(['name','number']));  
    // dd($request->except(['_token']));  
    dd($request->input('name'));  
  
    return "Cadastrando...";  
}
```



A screenshot of a web browser window. The address bar shows the URL `laravel53-basico.dev.com/painel/`. The page content displays the output of the `dd` function, which is the string `"Carlos Ferreira"`.

```
"Carlos Ferreira"
```



## Aula 22 – Cadastrar dados do formulário

- Em ProdutoController:

```
public function store(Request $request)
{
    // dd($request->all());
    // dd($request->only(['name','number']));
    // dd($request->except(['_token']));
    // dd($request->input('name'));

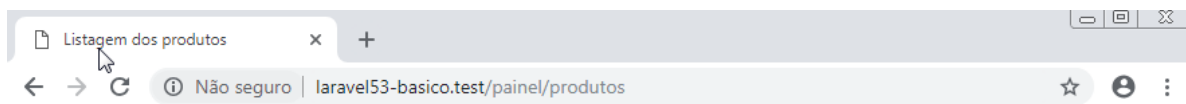
    /* Pega todos os dados que vem do formulário */
    // $dataForm = $request->except('_token');
    $dataForm = $request->all();

    $dataForm['active'] = (!isset($dataForm['active'])) ? 0 : 1;

    /* Faz o cadastro */
    // $insert = $this->product->insert($dataForm);
    $insert = $this->product->create($dataForm);

    if($insert)
        return redirect()->route('produtos.index');
    else
        return redirect()->route('produtos.create');

    // return "Cadastrando...";
}
```



### LISTAGEM DOS PRODUTOS

[+ Cadastrar](#)

Nome	Descrição	Ações
Condicionador	Descrição do produto	 
Nome do produto	Descrição do produto	 
Nome do produto 2	Descrição do produto	 
Update 2	Desc.Update	 
Shampoo	Shampoo anti-caspa	 
Aparelho celular	Smartphone	 
Sofá	Sofá com 4 lugares	 

## Aula 23 – Validação de dados

- Em ProdutoController.php:

```
public function store(Request $request)
{
    /* Pega todos os dados que vem do formulário */
    // $dataForm = $request->except('_token');
    $dataForm = $request->all();

    $dataForm['active'] = (!isset($dataForm['active'])) ? 0 : 1;

    /* Valida os dados */
    $this->validate($request, $this->product->rules);

    /* Faz o cadastro */
    // $insert = $this->product->insert($dataForm);
    $insert = $this->product->create($dataForm);

    if($insert)
        return redirect()->route('produtos.index');
    else
        return redirect()->route('produtos.create');

    // return "Cadastrando...";
}
```

- Na pasta 'Models → Painei:

### Product.php

```
<?php

namespace App\Models\Painei;

use Illuminate\Database\Eloquent\Model;

class Product extends Model
{
    protected $fillable = [
        'name', 'number', 'active', 'category', 'description'
    ];

    // protected $guarded = ['admin'];

    public $rules = [
        'name' => 'required|min:3|max:100',
        'number' => 'required|numeric',
        'category' => 'required',
        'description' => 'min:3|max:1000'
    ];
}
```

## create.blade.php

```
@extends('painel.templates.template')

@section('content')

    <h1 class="title-pg">Gestão produto</h1>

    @if(isset($errors) && count($errors) > 0)
        <div class="alert alert-danger">
            @foreach($errors->all() as $error)
                <p>{{ $error }}</p>
            @endforeach
        </div>
    @endif

    <form class="form" method="post" action="{{route('produtos.store')}}">

        <!-- <input type="hidden" name="_token" value="{{csrf_token()}}"> -->

        {!! csrf_field() !!}

        <div class="form-group">
            <input type="text" name="name" placeholder="Nome" class="form-control"
value="{{old('name')}}">
        </div>

        <div class="form-group">
            <label>
                <input type="checkbox" name="active" value="1">
                Ativo?
            </label>
        </div>

        <div class="form-group">
            <input type="text" name="number" placeholder="Número" class="form-control"
value="{{old('number')}}">
        </div>

        <div class="form-group">
            <select name="category" class="form-control">
                <option>Escolha a categoria</option>
                @foreach($categorys as $category)
                    <option value="{{ $category }}">{{ $category }}</option>
                @endforeach
            </select>
        </div>

        <div class="form-group">
            <textarea name="description" placeholder="Descrição" class="form-
control">{{old('description')}}</textarea>
        </div>

        <button class="btn btn-primary">Enviar</button>
    </form>
@endsection
```

Cadastrar novo produto

← → ↺ Não seguro | laravel53-basico.test/painel/produtos/create

## GESTÃO PRODUTO

O campo Nome é de preenchimento obrigatório!  
O campo Número é de preenchimento obrigatório!  
The description must be at least 3 characters.

☐ Ativo?

Escolha a categoria

Descrição

Enviar

Cadastrar novo produto

← → ↺ Não seguro | laravel53-basico.test/painel/produtos/create

## GESTÃO PRODUTO

The name must be at least 3 characters.  
No campo Número, digite apenas números!  
The description must be at least 3 characters.

☐ Ativo?

Escolha a categoria

c

Enviar

## Aula 24 – Personalizar mensagens de erro

- Em ProdutoController.php:

```
public function store(Request $request)
{

    /* Pega todos os dados que vem do formulário */
    // $dataForm = $request->except('_token');
    $dataForm = $request->all();

    $dataForm['active'] = (!isset($dataForm['active'])) ? 0 : 1;

    /* Valida os dados */
    // $this->validate($request, $this->product->rules);

    $messages = [
        'name.required' => 'O campo Nome é de preenchimento obrigatório!',
        'number.numeric' => "No campo Número, digite apenas números!",
        'number.required' => 'O campo Número é de preenchimento obrigatório!',
    ];

    $validate = validator($dataForm, $this->product->rules, $messages);

    if( $validate->fails()){
        return redirect()
            ->route('produtos.create')
            ->withErrors($validate)
            ->withInput();
    }

    /* Faz o cadastro */
    // $insert = $this->product->insert($dataForm);
    $insert = $this->product->create($dataForm);

    if($insert)
        return redirect()->route('produtos.index');
    else
        return redirect()->route('produtos.create');

    // return "Cadastrando...";
}
```

Cadastrar novo produto

Não seguro

laravel53-basico.test/painel/produtos/cre...

## GESTÃO PRODUTO

O campo Nome é de preenchimento obrigatório!

O campo Número é de preenchimento obrigatório!

The category field is required.

The description must be at least 3 characters.

Nome

☐ Ativo?

Número

Escolha a categoria

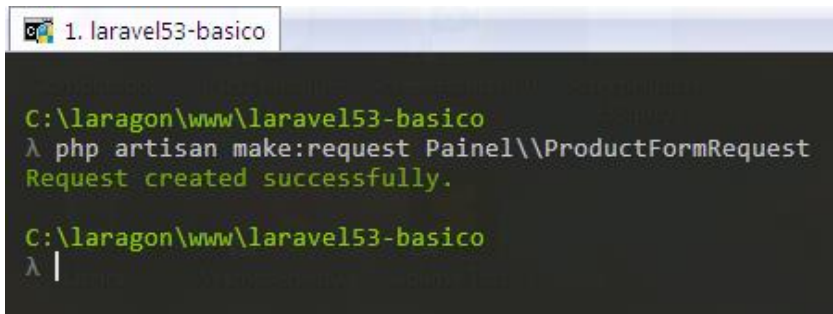
Descrição

Enviar

## Aula 25 – Form Request Validation

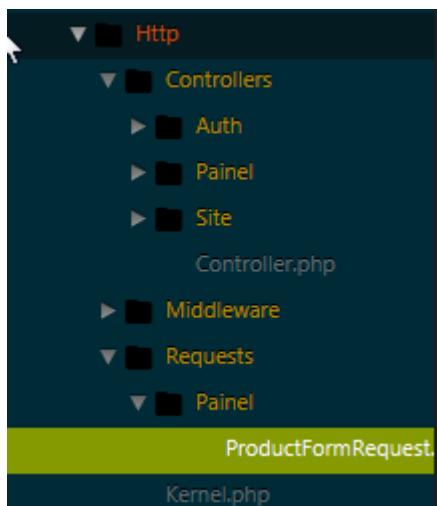
No terminal entre com o seguinte comando:

`php artisan make:request Paine1\\ProductFormRequest`

A screenshot of a terminal window titled "1. laravel53-basico". The terminal shows the command `php artisan make:request Paine1\\ProductFormRequest` being executed, with the output `Request created successfully.`. The prompt `C:\laragon\www\laravel53-basico` is visible at the top and bottom of the terminal.

```
C:\laragon\www\laravel53-basico
λ php artisan make:request Paine1\\ProductFormRequest
Request created successfully.

C:\laragon\www\laravel53-basico
λ |
```



### ProductFormRequest

```
<?php
```

```
namespace App\Http\Requests\Paine1;
use Illuminate\Foundation\Http\FormRequest;
```

```
class \ProductFormRequest extends FormRequest
{
    public function authorize()
    {
        return false;
    }
    public function rules()
    {
        return [
            //
        ];
    }
}
```



- Altere o arquivo **ProductFormRequest.php** para:

```
<?php
```

```
namespace App\Http\Requests\Painel;
```

```
use Illuminate\Foundation\Http\FormRequest;
```

```
class ProductFormRequest extends FormRequest
```

```
{
```

```
    public function authorize()
```

```
    {
```

```
        return true;
```

```
    }
```

```
    public function rules(){
```

```
        return [
```

```
            'name' => 'required|min:3|max:100',
```

```
            'number' => 'required|numeric',
```

```
            'category' => 'required',
```

```
            'description' => 'min:3|max:1000'
```

```
        ];
```

```
    }
```

```
    public function messages(){
```

```
        return [
```

```
            'name.required' => 'O campo Nome é de preenchimento obrigatório!',
```

```
            'number.numeric' => "No campo Número, digite apenas números!",
```

```
            'number.required' => 'O campo Número é de preenchimento obrigatório!',
```

```
        ];
```

```
    }
```

```
}
```

- Altere o arquivo **Product.php** para:

```
<?php

namespace App\Models\Painel;

use Illuminate\Database\Eloquent\Model;

class Product extends Model
{
    protected $fillable = [
        'name', 'number', 'active', 'category', 'description'
    ];

    // public $rules = [
    //     'name' => 'required|min:3|max:100',
    //     'number' => 'required|numeric',
    //     'category' => 'required',
    //     'description' => 'min:3|max:1000'
    // ];
}
```

- Em ProdutoController.php:

```
public function store(ProductFormRequest $request)
{
    /* Pega todos os dados que vem do formulário */
    $dataForm = $request->all();

    $dataForm['active'] = (!isset($dataForm['active'])) ? 0 : 1;

    /* Faz o cadastro */
    // $insert = $this->product->insert($dataForm);
    $insert = $this->product->create($dataForm);

    if($insert)
        return redirect()->route('produtos.index');
    else
        return redirect()->route('produtos.create');
}
```

Cadastrar novo produto

←

→

↻

ⓘ Não seguro | laravel53-basico.test/painel/produtos/cre...

☆

👤

⋮

## GESTÃO PRODUTO

O campo Nome é de preenchimento obrigatório!

O campo Número é de preenchimento obrigatório!

The category field is required.

The description must be at least 3 characters.

Nome

☐ Ativo?

Número

Escolha a categoria ▼

Descrição

Enviar

## Aula 26 – Preparando para editar dados

- Renomear o arquivo **create.blade.php** para **create-edit.blade.php**

- Em ProdutoController.php

```
public function create()
{
    // return "#form Cad";

    $title = "Cadastrar novo produto";
    $categorys = ['eletronicos', 'moveis', 'limpeza', 'banho'];

    return view('painel.products.create-edit', compact('title', 'categorys'));
}

public function edit($id)
{
    /* Recupera o produto pelo seu id */

    $title = "Editar produto {$id}";
    $categorys = ['eletronicos', 'moveis', 'limpeza', 'banho'];

    $product = $this->product->find($id);

    return view('painel.products.create-edit', compact('title', 'categorys', 'product'));

    // return "Editar item {$id}";
}

public function update(Request $request, $id)
{
    return "Editando o item {$id}";
}
```

## create-edit.blade.php

```
@extends('painel.templates.template')
```

```
@section('content')
```

```
<h1 class="title-pg">Gestão produto</h1>
```

```
@if(isset($errors) && count($errors) > 0)
```

```
<div class="alert alert-danger">
```

```
@foreach($errors->all() as $error)
```

```
<p>{{ $error }}</p>
```

```
@endforeach
```

```
</div>
```

```
@endif
```

```
@if(isset($product) )
```

```
<form class="form" method="post" action="{{route('produtos.update', $product->id)}}">
```

```
{!! method_field('PUT') !!}
```

```
@else
```

```
<form class="form" method="post" action="{{route('produtos.store')}}">
```

```
@endif
```

```
<!-- <input type="hidden" name="_token" value="{{csrf_token()}}" -->
```

```
{!! csrf_field() !!}
```

```
<div class="form-group">
```

```
<input type="text" name="name" placeholder="Nome" class="form-control" value="{{ $product->name ?? old('name') }}">
```

```
</div>
```

```
<div class="form-group">
```

```
<label>
```

```
<input type="checkbox" name="active" value="1" @if(isset($product) && $product->active == 1) checked @endif>
```

```
Ativo?
```

```
</label>
```

```
</div>
```

```

<div class="form-group">
    <input type="text" name="number" placeholder="Número" class="form-control"
value="{{ $product->number ?? old('number') }}">
</div>

<div class="form-group">
    <select name="category" class="form-control">
        <option value="">Escolha a categoria</option>
        @foreach($categorys as $category)
            <option value="{{ $category }}" @if(isset($product) && $product->category ==
$category) selected @endif>{{ $category }}</option>
        @endforeach
    </select>
</div>

<div class="form-group">
    <textarea name="description" placeholder="Descrição" class="form-
control">{{ $product->description ?? old('description') }}</textarea>
</div>

<button class="btn btn-primary">Enviar</button>

</form>
@endsection

```

## index.blade.php (Products)

```
@section('content')
    <h1 class="title-pg">Listagem dos produtos</h1>

    <!-- <a href="{{url('/painel/produtos/create')}}" class="btn btn-primary btn-add"><span
    class="glyphicon glyphicon-plus"></span> Cadastrar</a>
    -->
    <a href="{{route('produtos.create')}}" class="btn btn-primary btn-add"><span
    class="glyphicon glyphicon-plus"></span> Cadastrar</a>

    <table class="table table-striped">
        <tr>
            <th>Nome</th>
            <th>Descrição</th>
            <th width="100px">Ações</th>
        </tr>

        @foreach($products as $product)
            <tr>
                <td>{{ $product->name }}</td>
                <td>{{ $product->description }}</td>
                <td>
                    <!-- <a href="{{url("/painel/produtos/{{$product->id}}/edit')}}" class="actions
edit"> -->
                    <a href="{{route('produtos.edit', $product->id)}}" class="actions edit">

                        <span class="glyphicon glyphicon-pencil"></span>
                    </a>
                    <a href="" class="actions delete">
                        <span class="glyphicon glyphicon-trash"></span>
                    </a>
                </td>
            </tr>
        @endforeach

    </table>
@endsection
```

## LISTAGEM DOS PRODUTOS

+ Cadastrar

Nome	Descrição	Ações
Shampoo e Condicionador	Descrição do produto para banho	 
Nome do produto	Descrição do produto	 
Nome do produto 2	Descrição do produto	 
Update 2	Desc. Update	 
Shampoo	Shampoo anti-caspa	 
Aparelho celular	Smartphone	 
Sofá	Sofá com 4 lugares	 
Microfone	Microfone de lapela	 
Estante	Estante com 4 prateleiras	 

- Clicando no botão “Editar” do primeiro produto:

## GESTÃO PRODUTO

Condicionador

☒ Ativo?

1311312

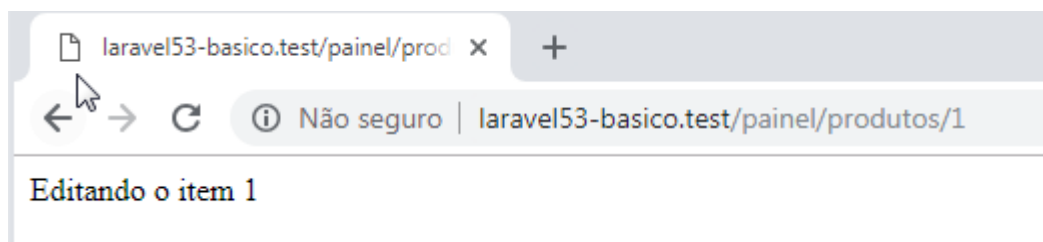
banho ▼

Descrição do produto

Enviar



- Clicando no botão “Enviar”:



## Aula 27 – Update dados laravel

- Em ProdutoController.php:

```
public function update(ProductFormRequest $request, $id)
{

    // Recupera todos os dados do formulário
    $dataForm = $request -> all();

    // Recupera o item para editar
    $product = $this->product->find($id);

    // Verifica se o produto está ativado
    $dataForm['active'] = (!isset($dataForm['active']) ) ? 0 : 1;

    // Altera os itens
    $update = $product->update($dataForm);

    // Verifica se realmente editou
    if($update){
        return redirect()->route('produtos.index');
    } else {
        return redirect()->route('produtos.edit', $id)->with(['errors'=>'Falha ao editar!']);
    }

    // return "Editando o item {$id}";
}
```

Editar Produto: Condicionador

Não seguro | laravel53-basico.test/painel/produtos/1/edit

## GESTÃO PRODUTO

Shampoo e Condicionador

☒ Ativo?

123456

banho

Descrição do produto para banho

Enviar

Listagem dos produtos

Não seguro | laravel53-basico.test/painel/produtos

LISTAGEM DOS PRODUTOS		
<div>+ Cadastrar</div>		
Nome	Descrição	Ações
Shampoo e Condicionador	Descrição do produto para banho	<div><div></div><div></div></div>
Nome do produto	Descrição do produto	<div><div></div><div></div></div>
Nome do produto 2	Descrição do produto	<div><div></div><div></div></div>
Update 2	Desc.Update	<div><div></div><div></div></div>
Shampoo	Shampoo anti-caspa	<div><div></div><div></div></div>
Aparelho celular	Smartphone	<div><div></div><div></div></div>
Sofá	Sofá com 4 lugares	<div><div></div><div></div></div>
Microfone	Microfone de lapela	<div><div></div><div></div></div>
Estante	Estante com 4 prateleiras	<div><div></div><div></div></div>

- Clicando no botão editar deste produto:

Editar Produto: Shampoo e Cond

Não seguro | laravel53-basico.test/painel/produtos/1/edit

## GESTÃO PRODUTO

Shampoo e Condicionador

☒ Ativo?

123456

banho

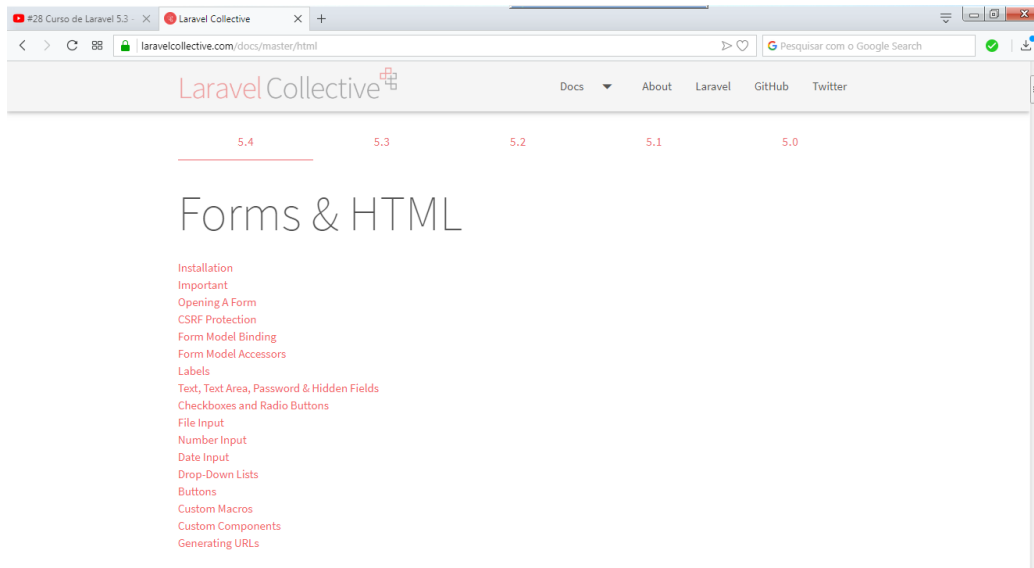
Descrição do produto para banho

Enviar

# Aula 28 - Pacote Collective Form and HTML

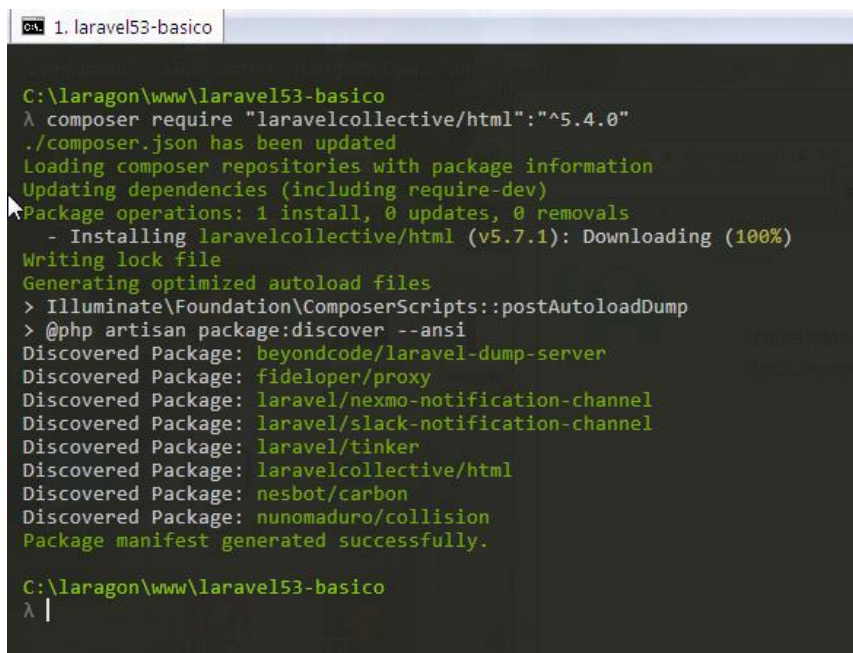
## Instalando o pacote Laravel Collective

- Acessar a página:  
<https://laravelcollective.com/docs/master/html>



- Clique na aba que corresponde a versão do Laravel que está usando.
- Entre no terminal com o seguinte comando:

```
composer require "laravelcollective/html":"^5.4.0"
```



- Em config → app.php, incluir:

Em Providers:

```
Collective\Html\HtmlServiceProvider::class,
```

Em aliases:

```
'Form' => Collective\Html\FormFacade::class,
```

```
'Html' => Collective\Html\HtmlFacade::class,
```

Alterar o arquivo **create-edit.blade.php** para:

```
@extends('painel.templates.template')
```

```
@section('content')
```

```
<h1 class="title-pg"><a href="{{route('produtos.index')}}"><span class="glyphicon glyphicon-fast-backward"></span></a> Gestão produto: <b>{{ $product->name ?? 'Novo' }}</b></h1>
```

```
@if(isset($errors) && count($errors) > 0)
```

```
<div class="alert alert-danger">
```

```
@foreach($errors->all() as $error)
```

```
<p>{{ $error }}</p>
```

```
@endforeach
```

```
</div>
```

```
@endif
```

```
@if(isset($product) )
```

```
{!! Form::model($product, ['route' => ['produtos.update', $product->id], 'class'=>'form', 'method'=>'put']) !!}
```

```
@else
```

```
{!! Form::open(['route' => 'produtos.store', 'class'=>'form']) !!}
```

```
@endif
```

```
<div class="form-group">
```

```
{!! Form::text('name', null, ['class'=>'form-control', 'placeholder' => 'Nome']) !!}
```

```
</div>
```

```
<div class="form-group">
```

```
<label>
```

```
{!! Form::checkbox('active') !!}
```

```
Ativo?
```

```
</label>
```

```
</div>
```

```

<div class="form-group">
    {!! Form::text('number', null, ['class'=>'form-control', 'placeholder' => 'Número']) !!}
</div>

<div class="form-group">
    {!! Form::select('category', $categorys, null, ['class'=>'form-control']) !!}
</div>

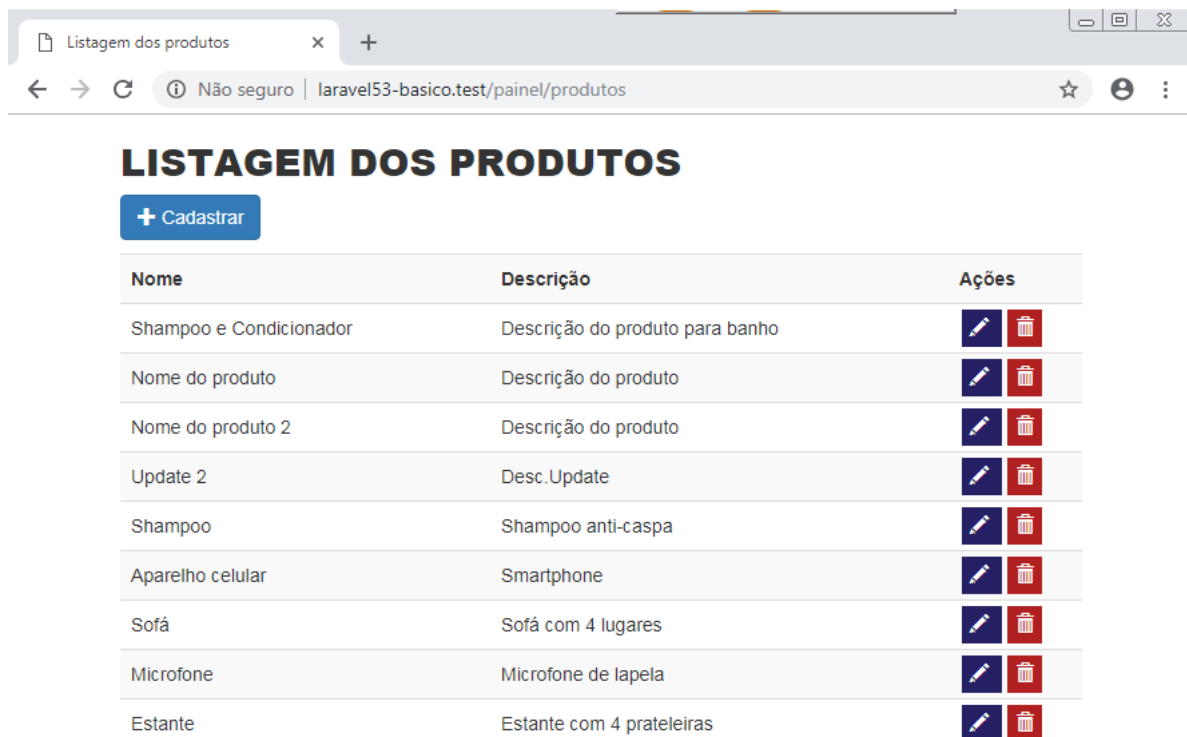
<div class="form-group">
    {!! Form::textarea('description', null, ['class'=>'form-control', 'placeholder' => 'Descrição'])
!!}
</div>



















    {!! Form::submit('Enviar', ['class'=>'btn btn-primary']) !!}

{!! Form::close() !!}

```

@endsection



Nome	Descrição	Ações
Shampoo e Condicionador	Descrição do produto para banho	 
Nome do produto	Descrição do produto	 
Nome do produto 2	Descrição do produto	 
Update 2	Desc.Update	 
Shampoo	Shampoo anti-caspa	 
Aparelho celular	Smartphone	 
Sofá	Sofá com 4 lugares	 
Microfone	Microfone de lapela	 
Estante	Estante com 4 prateleiras	 

Cadastrar novo produto

laravel53-basico.test/painel/produtos/create

## GESTÃO PRODUTO: NOVO

☐ Ativo?

Escolha a categoria

Descrição

Enviar

Editar Produto: Shampoo e Cond

laravel53-basico.test/painel/produtos/1/edit

## GESTÃO PRODUTO: SHAMPOO E CONDICIONADOR

☒ Ativo?

banho

Descrição do produto para banho

Enviar



## Aula 29 - Deletar

- Em "ProdutoControle.php":

```
public function destroy($id)
{
    $product = $this->product->find($id);

    $delete = $product->delete();

    if( $delete )
        return redirect()->route('produtos.index');
    else
        return redirect()->route('produtos.show', $id)->with(['errors' => 'Falha ao deletar']);

    // return "Deletando o item {$id}";
}
```

- Criar em "view → painel → products" o arquivo "show.blade.php"

**show.blade.php**

```
@extends('painel.templates.template')
@section('content')

<h1 class="title-pg">
    <a href="{{route('produtos.index')}}"><span class="glyphicon glyphicon-fast-
backward"></span></a>
    Produto: <b>{{ $product->name }}</b>
</h1>
<p><b>Ativo:</b> {{ $product->active }}</p>
<p><b>Número:</b> {{ $product->number }}</p>
<p><b>Categoria:</b> {{ $product->category }}</p>
<p><b>Descrição:</b> {{ $product->description }}</p>

<hr>

@if( isset($errors) && count($errors) > 0 )
    <div class="alert alert-danger">
        @foreach( $errors->all() as $error )
            <p>{{ $error }}</p>
        @endforeach
    </div>
@endif

{!! Form::open(['route' => ['produtos.destroy', $product->id], 'method' => 'DELETE']) !!}
{!! Form::submit("Deletar Produto: $product->name", ['class' => 'btn btn-danger']) !!}
{!! Form::close() !!}

@endsection
```

**index.blade.php (view → painel → products)**

```
@extends('painel.templates.template')
```

```
@section('content')
```

```
<h1 class="title-pg">Listagem dos produtos</h1>
```

```
<!-- <a href="{{url('/painel/produtos/create')}}" class="btn btn-primary btn-add"><span
class="glyphicon glyphicon-plus"></span> Cadastrar</a>
-->
```

```
<a href="{{route('produtos.create')}}" class="btn btn-primary btn-add"><span class="glyphicon
glyphicon-plus"></span> Cadastrar</a>
```

```
<a href="{{route('produtos.create')}}" class="btn btn-primary btn-add"><span class="glyphicon
glyphicon-plus"></span> Cadastrar</a>
```

```
<table class="table table-striped">
```

```
<tr>
```

```
<th>Nome</th>
```

```
<th>Descrição</th>
```

```
<th width="100px">Ações</th>
```

```
</tr>
```

```
@foreach($products as $product)
```

```
<tr>
```

```
<td>{{ $product->name }}</td>
```

```
<td>{{ $product->description }}</td>
```

```
<td>
```

```
<a href="{{route('produtos.edit', $product->id)}}" class="actions edit">
```

```
<span class="glyphicon glyphicon-pencil"></span>
```

```
</a>
```

```
<a href="{{route('produtos.show', $product->id)}}" class="actions delete">
```

```
<span class="glyphicon glyphicon-eye-open"></span>
```

```
</a>
```

```
</td>
```

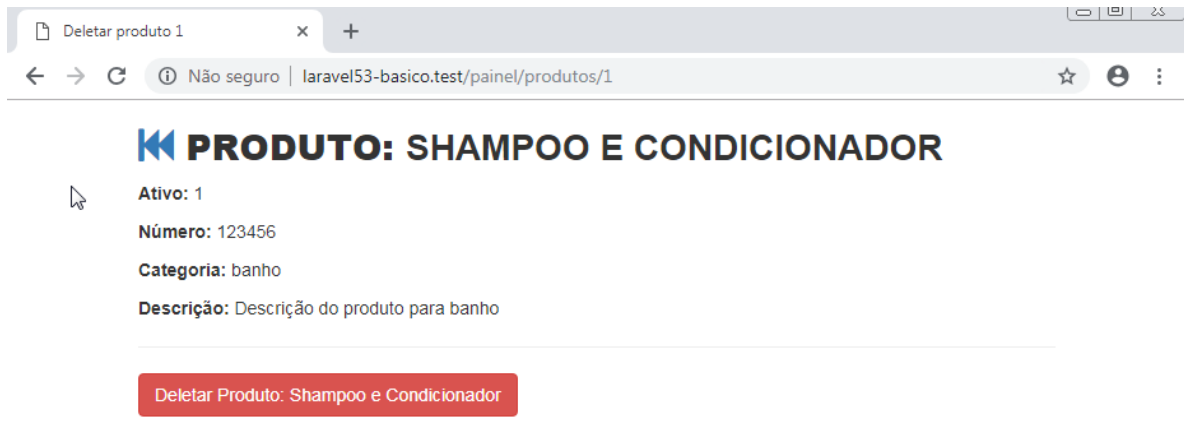
```
</tr>
```

```
@endforeach
```

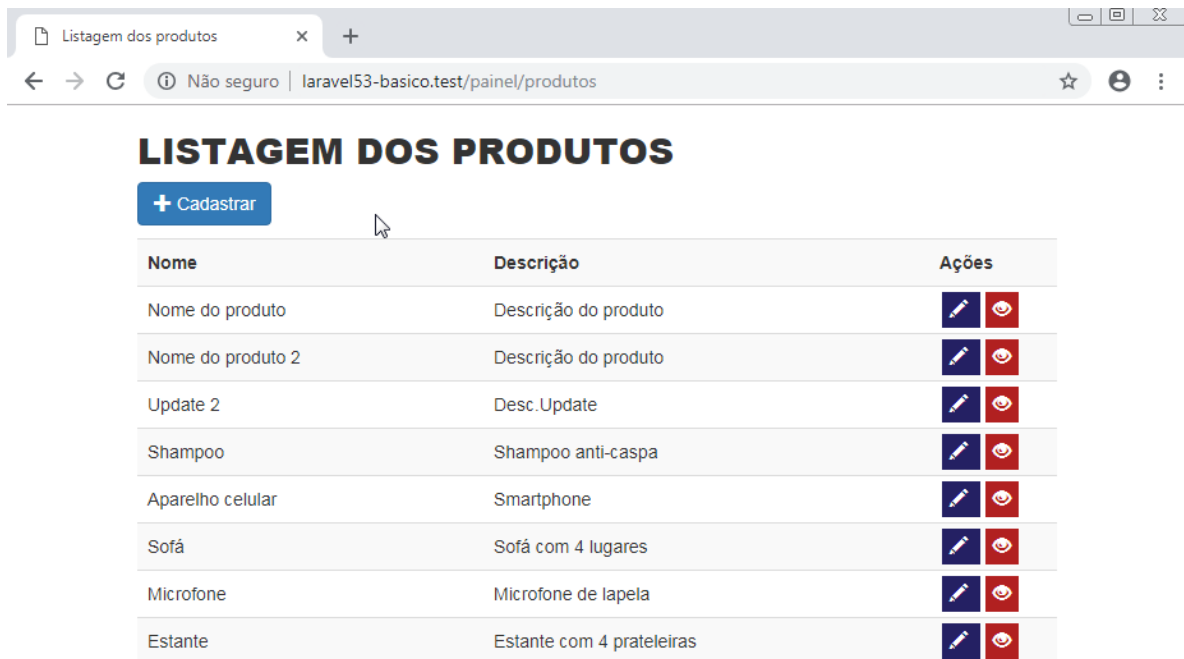
```
</table>
```

```
@endsection
```

- Clique no botão “Visualizar dados / Deletar” do primeiro produto:



- Clique no botão “Deletar Produto”:



## Aula 30 - Paginação

- Em “**ProdutoController.php**”;

```
private $product;
private $totalPage = 3;

public function index()
{

    $title = "Listagem dos produtos";
    // $products = $this->product->all();
    $products = $this->product->paginate($this -> totalPage);

    return view('painel.products.index', compact('products','title'));
}
```

- Em “**index.blade.php**” (view → painel → products):

```
@extends('painel.templates.template')
```

```
@section('content')
```

```
<h1 class="title-pg">Listagem dos produtos</h1>
```

```
<a href="{{route('produtos.create')}}" class="btn btn-primary btn-add"><span
class="glyphicon glyphicon-plus"></span> Cadastrar</a>
```

```
<table class="table table-striped">
```

```
<tr>
```

```
<th>Nome</th>
```

```
<th>Descrição</th>
```

```
<th width="100px">Ações</th>
```

```
</tr>
```

```

@foreach($products as $product)
    <tr>
        <td>{{ $product->name }}</td>
        <td>{{ $product->description }}</td>
        <td>
            <!-- <a href="{{url('/painel/produtos/{ $product->id }/edit')}}" class="actions
edit"> -->
                <a href="{{route('produtos.edit', $product->id)}}" class="actions edit">
                    <span class="glyphicon glyphicon-pencil"></span>
                </a>

            <!--
                <a href="" class="actions delete">
                    <span class="glyphicon glyphicon-trash"></span>
                </a>
            -->
                <a href="{{route('produtos.show', $product->id)}}" class="actions delete">
                    <span class="glyphicon glyphicon-eye-open"></span>
                </a>

        </td>
    </tr>
@endforeach

</table>

@if( isset($dataForm) )
    {!! $products->appends($dataForm)->links() !!}
@else
    {!! $products->links() !!}
@endif

@endsection

```

Listagem dos produtos

+







← → ↻

ⓘ Não seguro | laravel53-basico.test/painel/produtos

☆ 👤 ⋮

## LISTAGEM DOS PRODUTOS

+ Cadastrar

Nome	Descrição	Ações
Nome do produto	Descrição do produto	 
Nome do produto 2	Descrição do produto	 
Update 2	Desc.Update	 

<

1

2

3

>

Listagem dos produtos

+







← → ↻

ⓘ Não seguro | laravel53-basico.test/painel/produtos?page=2

☆ 👤 ⋮

## LISTAGEM DOS PRODUTOS

+ Cadastrar

Nome	Descrição	Ações
Shampoo	Shampoo anti-caspa	 
Aparelho celular	Smartphone	 
Sofá	Sofá com 4 lugares	 

<

1

2

3

>

Listagem dos produtos

+

← → ↻

ⓘ Não seguro | laravel53-basico.test/painel/produtos?page=3

☆ 👤 ⋮

## LISTAGEM DOS PRODUTOS

+ Cadastrar

Nome	Descrição	Ações
Microfone	Microfone de lapela	 
Estante	Estante com 4 prateleiras	 

<

1

2

3

>