

CURSO LARAVEL 5.5

Larast

https://www.youtube.com/watch?v=aOjXDiqNIPs&list=PLsY_JS2Bx44k2pY9vihW_ok9fA6Z6L7CF

Resumo do curso feito por Roberto Pinheiro

Aula 1 - Rotas e Vistas

As rotas são os pontos de entrada de nossa aplicação web. A partir delas podemos invocar os views ou vistas que contém a informação que será exibida no navegador.

Retornando um texto plano

routes/web.php

```
Route::get('/', function () {
    return 'Minhas anotações';
});

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

The image contains two screenshots of a web browser window. The top screenshot shows the URL 'laravel-larast.test' in the address bar, with the page content 'Minhas anotações'. The bottom screenshot shows the URL 'laravel-larast.test/contact' in the address bar, with the page content 'Contato'.

Retornando vistas

routes/web.php

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

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

resources/views/contact.blade.php

```
<!doctype html>
<html lang="{{ app()->getLocale() }}>
    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1">

        <title>Laravel</title>

        <!-- Fonts -->
        <link href="https://fonts.googleapis.com/css?family=Raleway:100,600" rel="stylesheet" type="text/css">

        <!-- Styles -->
        <style>
            html, body {
                background-color: #fff;
                color: #636b6f;
                font-family: 'Raleway', sans-serif;
                font-weight: 100;
                height: 100vh;
                margin: 0;
            }

            .full-height {
                height: 100vh;
            }

            .flex-center {
                align-items: center;
                display: flex;
                justify-content: center;
            }
        </style>
    </head>
    <body>
```

```
.position-ref {
    position: relative;
}

.top-right {
    position: absolute;
    right: 10px;
    top: 18px;
}

.content {
    text-align: center;
}

.title {
    font-size: 84px;
}

.links > a {
    color: #636b6f;
    padding: 0 25px;
    font-size: 12px;
    font-weight: 600;
    letter-spacing: .1rem;
    text-decoration: none;
    text-transform: uppercase;
}

.m-b-md {
    margin-bottom: 30px;
}

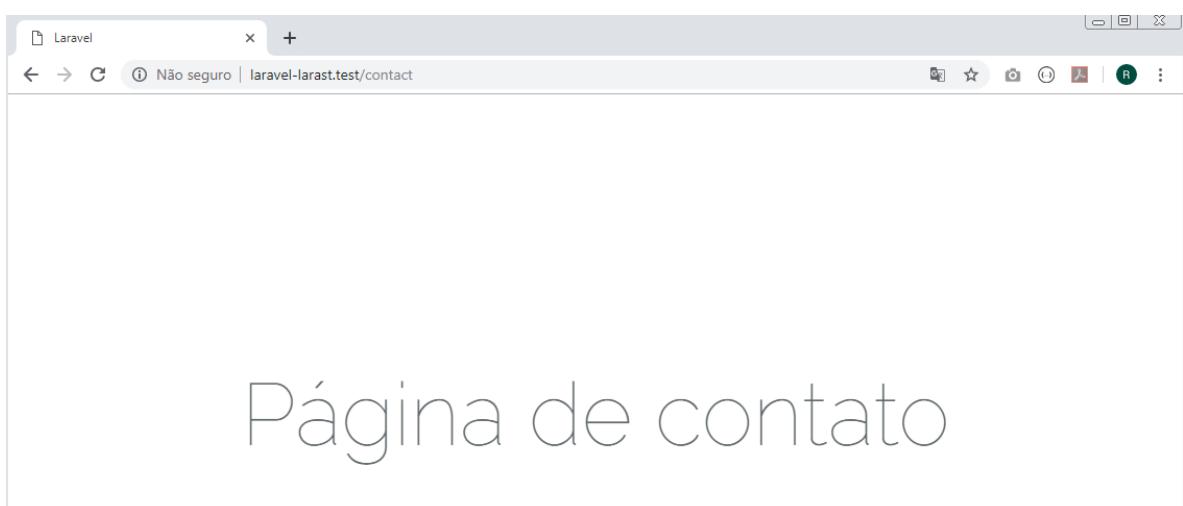
```

</style>

```
</head>
<body>
    <div class="flex-center position-ref full-height">
        @if (Route::has('login'))
            <div class="top-right links">
                @auth
                    <a href="{{ url('/home') }}">Home</a>
                @else
                    <a href="{{ route('login') }}">Login</a>
                    <a href="{{ route('register') }}">Register</a>
                @endauth
            </div>
        @endif
    </div>

```

```
<div class="content">  
    <div class="title m-b-md">  
        Página de contato  
    </div>  
    </div>  
    </body>  
</html>
```



Aula 2 - Vistas dinâmicas com Blade

Para fazer com que as vistas de nossa aplicação sejam dinâmicas, é necessário compartilhar com elas dados e informações que podem ser exibidos. Isso é muito simples de se fazer usando Blade, o motor de templates do Laravel.

routes/web.php

```
<?php

/*
|--------------------------------------------------------------------------
| Web Routes
|--------------------------------------------------------------------------
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/

/* Vistas dinâmicas com Blade */
/* Passando dados de um array para o view */

Route::get('/', function () {
    $name = 'Larast';
    $message = 'Vistas dinâmicas com Blade';
    $date = date('Y-m-d H:i:s');
    $notes = [
        [
            'title' => 'Rotas Laravel',
            'body' => 'As rotas são definidas no arquivo web.php',
            'important' => true
        ],
        [
            'title' => 'Blade',
            'body' => 'Blade é o motor de templates do Laravel',
            'important' => false
        ]
    ];
    return view('welcome', [ 'welcome' => 'Olá Mundo!', 'name' => $name, 'date'
=> $date, 'message' => $message, 'notes' => $notes ]);
});
```



Minhas anotações

Olá Mundo!

Larast

2019-04-19 05:16:46

Vistas dinâmicas com Blade

Rotas Laravel *

As rotas são definidas no arquivo web.php

Blade

Blade é o motor de templates do Laravel.

[DOCUMENTATION](#)

[LARACASTS](#)

[NEWS](#)

[FORGE](#)

[GITHUB](#)

Aula 3 - Introdução a Migrations

As migrações permitem criar tabelas nas bases de dados, escrevendo código PHP orientado a objetos. É uma maneira muito conveniente de manipular e versionar até certo ponto as alterações na base de dados.

Configurando o acesso a base de dados

.env

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:M29FqBdy1ir30TerdGYjv3YPRsMN+3k8vDvIgM+stTw=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=laravel-larast
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
SESSION_DRIVER=file
SESSION_LIFETIME=120
QUEUE_DRIVER=sync

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_DRIVER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null

PUSHER_APP_ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
PUSHER_APP_CLUSTER=mt1

MIX_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
MIX_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
```

Criando as tabelas

php artisan migrate

```
C:\laragon\www\laravel-larast 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

C:\laragon\www\laravel-larast λ | api_rest_laravel
```

Table	Action	Rows	Type	Collation	Size	Overhead
migrations	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_unicode_ci	16 KiB	-
password_resets	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16 KiB	-
users	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16 KiB	-
3 tables	Sum			2	InnoDB latin1_swedish_ci 48 KiB	0 B

Criando uma migration

```
php artisan make:migration create_notes_table
```

```
C:\laragon\www\laravel-larast
λ php artisan make:migration create_notes_table
Created Migration: 2019_04_19_071011_create_notes_table

C:\laragon\www\laravel-larast
λ |
```

database/migrations/2019_04_19_071011_create_notes_table.php

```
<?php

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

class CreateNotesTable extends Migration
{
    public function up()
    {
        Schema::create('notes', function (Blueprint $table) {
            $table->increments('id');
            $table->string('title');
            $table->text('body');
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('notes');
    }
}
```

Criando a tabela notes

php artisan migrate

```
C:\laragon\www\laravel-larast
λ php artisan migrate
Migrating: 2019_04_19_071011_create_notes_table
Migrated: 2019_04_19_071011_create_notes_table

C:\laragon\www\laravel-larast
λ |
```

Table	Action	Rows	Type	Collation	Size	Overhead
migrations	★ Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_unicode_ci	16 KiB	-
notes	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16 KiB	-
password_resets	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16 KiB	-
users	★ Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_unicode_ci	16 KiB	-
4 tables	Sum	3	InnoDB	latin1_swedish_ci	64 KiB	0 B

Aula 4 - Migrations

Ao trabalhar com migrações podemos encontrar com situações em que necessitamos fazer alterações em uma tabela. Nesta aula veremos alguns comandos e sugestões úteis.

Inserindo um campo na tabela notes

database/migrations/2019_04_19_071011_create_notes_table.php

```
<?php

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

class CreateNotesTable extends Migration
{
    public function up()
    {
        Schema::create('notes', function (Blueprint $table) {
            $table->increments('id');
            $table->string('title');
            $table->text('body');
            $table->boolean('important');
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('notes');
    }
}
```

Executando a inserção do campo na tabela

```
php artisan migrate:refresh
```

```
C:\laragon\www\laravel-larast
λ php artisan migrate:refresh
Rolling back: 2019_04_19_071011_create_notes_table
Rolled back: 2019_04_19_071011_create_notes_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_04_19_071011_create_notes_table
Migrated: 2019_04_19_071011_create_notes_table

C:\laragon\www\laravel-larast
λ |
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	title	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	body	text	utf8mb4_unicode_ci		No	None			Change Drop More
4	important	tinyint(1)			No	None			Change Drop More
5	created_at	timestamp			Yes	NULL			Change Drop More
6	updated_at	timestamp			Yes	NULL			Change Drop More

Criando migration para relacionamento entre tabelas

```
php artisan make:migration add_user_id_column_to_notes_table --table=notes
```

```
C:\laragon\www\laravel-larast
λ php artisan make:migration add_user_id_column_to_notes_table --table=notes
Created Migration: 2019_04_19_080304_add_user_id_column_to_notes_table

C:\laragon\www\laravel-larast
λ |
```

database/migrations/ 2019_04_19_075602_add_user_id_column_to_notes_table.php

```
<?php

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

class AddUserIdColumnToNotesTable extends Migration
{
    public function up()
    {
        Schema::table('notes', function (Blueprint $table) {
            $table->integer('user_id')->unsigned();
            $table->foreign('user_id')->references('id')->on('users');
        });
    }

    public function down()
    {
        Schema::table('notes', function (Blueprint $table) {
            // ...
        });
    }
}
```

Inserindo o campo

php artisan migrate

```
C:\laragon\www\laravel-larast
λ php artisan migrate
Migrating: 2019_04_19_080304_add_user_id_column_to_notes_table
Migrated: 2019_04_19_080304_add_user_id_column_to_notes_table
2019_04_19_080304_add_user_id_column_to_notes_table

C:\laragon\www\laravel-larast
λ |
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	title	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	body	text	utf8mb4_unicode_ci		No	None			Change Drop More
4	important	tinyint(1)			No	None			Change Drop More
5	created_at	timestamp			Yes	NULL			Change Drop More
6	updated_at	timestamp			Yes	NULL			Change Drop More
7	user_id	int(10)		UNSIGNED	No	None			Change Drop More

Mudando a posição do campo na tabela

```
database/migrations/
2019_04_19_075602_add_user_id_column_to_notes_table.php

<?php

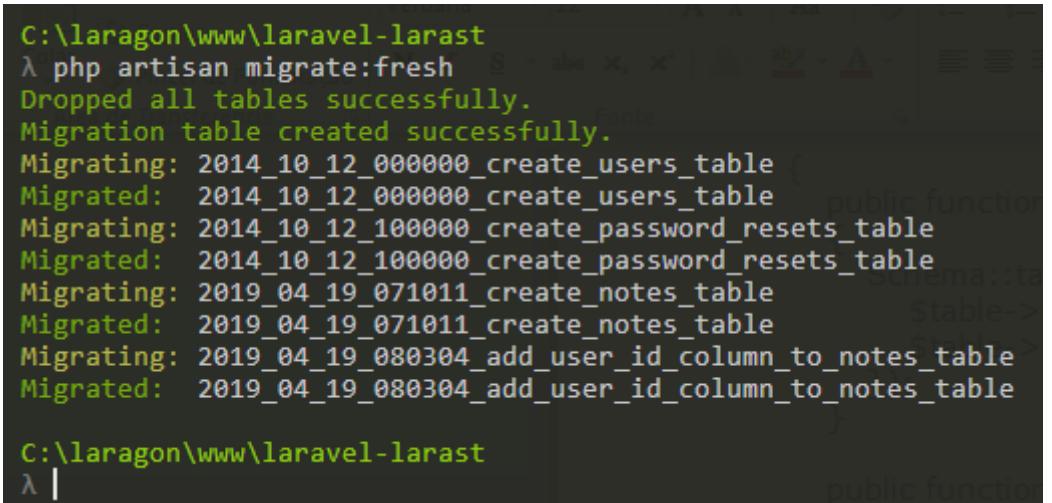
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class AddUserIdColumnToNotesTable extends Migration
{
    public function up()
    {
        Schema::table('notes', function (Blueprint $table) {
            $table->integer('user_id')->unsigned()->after('id');
            $table->foreign('user_id')->references('id')->on('users');
        });
    }

    public function down()
    {
        Schema::table('notes', function (Blueprint $table) {
            // ...
        });
    }
}
```

Apagando todas as tabelas e criando-as novamente

```
php artisan migrate:fresh
```



The screenshot shows a terminal window with the following text output:

```
C:\laragon\www\laravel-larast
λ 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
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table
Migrating: 2019_04_19_071011_create_notes_table
Migrated: 2019_04_19_071011_create_notes_table
Migrating: 2019_04_19_080304_add_user_id_column_to_notes_table
Migrated: 2019_04_19_080304_add_user_id_column_to_notes_table

C:\laragon\www\laravel-larast
λ |
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id 	int(10)		UNSIGNED	No	None		AUTO_INCREMENT	 Change  Drop  More
2	user_id 	int(10)		UNSIGNED	No	None			 Change  Drop  More
3	title	varchar(255)	utf8mb4_unicode_ci		No	None			 Change  Drop  More
4	body	text	utf8mb4_unicode_ci		No	None			 Change  Drop  More
5	important	tinyint(1)			No	None			 Change  Drop  More
6	created_at	timestamp			Yes	NULL			 Change  Drop  More
7	updated_at	timestamp			Yes	NULL			 Change  Drop  More

Aula 5 - Query Builder

O construtor de consultas do Laravel é uma classe que nos brinda uma maneira muito conveniente de fazer consultas à base de dados. Isto sem a necessidade de escrever código SQL de maneira direta, pois a lógica para construir as cadeias SQL está encapsulada dentro desta classe.

**database/migrations/
2019_04_19_075602_add_user_id_column_to_notes_table.php**

```
<?php

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

class AddUserIdColumnToNotesTable extends Migration
{
    public function up()
    {
        Schema::table('notes', function (Blueprint $table) {
            // $table->integer('user_id')->unsigned()->after('id');
            // $table->foreign('user_id')->references('id')->on('users');
        });
    }
    public function down()
    {
        Schema::table('notes', function (Blueprint $table) {
            //
        });
    }
}
```

php artisan migrate:fresh

```
C:\laragon\www\laravel-larast
λ 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
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table
Migrating: 2019_04_19_071011_create_notes_table
Migrated: 2019_04_19_071011_create_notes_table
Migrating: 2019_04_19_080304_add_user_id_column_to_notes_table
Migrated: 2019_04_19_080304_add_user_id_column_to_notes_table

C:\laragon\www\laravel-larast
λ |
```

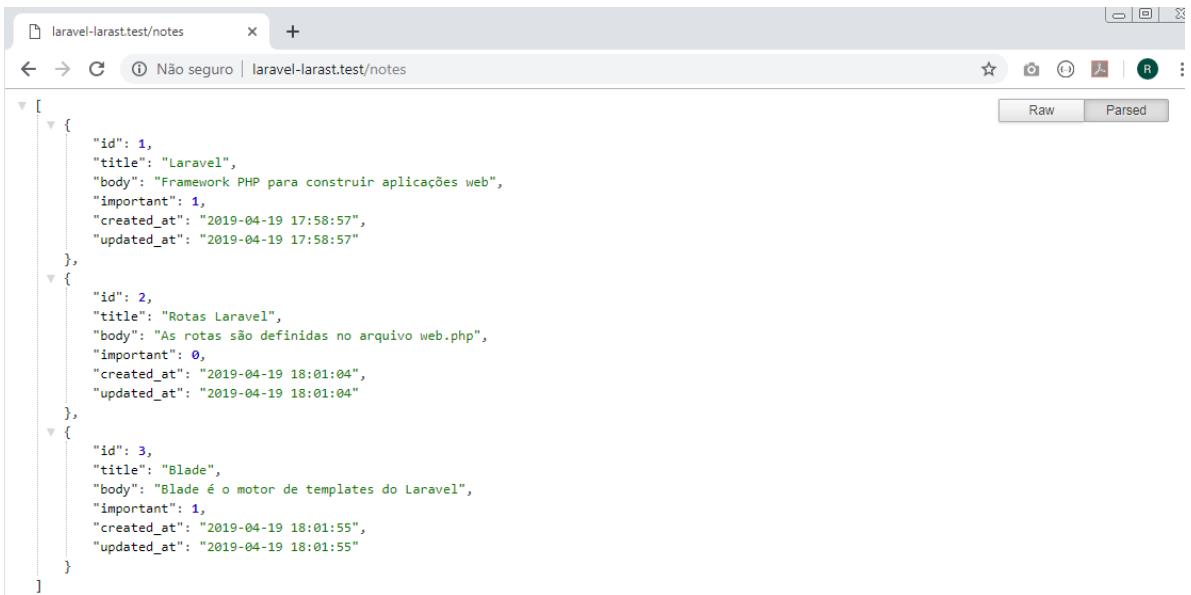
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	title	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	body	text	utf8mb4_unicode_ci		No	None			Change Drop More
4	important	tinyint(1)			No	None			Change Drop More
5	created_at	timestamp			Yes	NULL			Change Drop More
6	updated_at	timestamp			Yes	NULL			Change Drop More

Na tabela **notes** insira manualmente os seguintes registros:

+ Options		← →	id	title	body	important	created_at	updated_at
Edit	Copy	Delete	1	Laravel	Framework PHP para construir aplicações web	1	2019-04-19 17:58:57	2019-04-19 17:58:57
Edit	Copy	Delete	2	Rotas Laravel	As rotas são definidas no arquivo web.php	0	2019-04-19 18:01:04	2019-04-19 18:01:04
Edit	Copy	Delete	3	Blade	Blade é o motor de templates do Laravel	1	2019-04-19 18:01:55	2019-04-19 18:01:55

routes/web.php

```
Route::get('/notes', function () {
    $notes = DB::table('notes')->get();
    return $notes;
});
```



The screenshot shows a browser window with the URL `laravel-larastest/test/notes`. The page displays a JSON array of three notes:

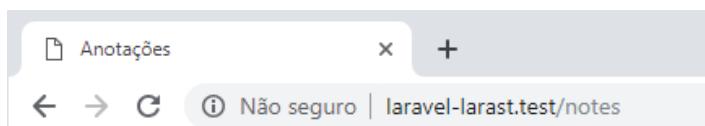
```
[
  {
    "id": 1,
    "title": "Laravel",
    "body": "Framework PHP para construir aplicações web",
    "important": 1,
    "created_at": "2019-04-19 17:58:57",
    "updated_at": "2019-04-19 17:58:57"
  },
  {
    "id": 2,
    "title": "Rotas Laravel",
    "body": "As rotas são definidas no arquivo web.php",
    "important": 0,
    "created_at": "2019-04-19 18:01:04",
    "updated_at": "2019-04-19 18:01:04"
  },
  {
    "id": 3,
    "title": "Blade",
    "body": "Blade é o motor de templates do Laravel",
    "important": 1,
    "created_at": "2019-04-19 18:01:55",
    "updated_at": "2019-04-19 18:01:55"
  }
]
```

routes/web.php

```
Route::get('/notes', function () {
    $notes = DB::table('notes')->get();
    return view('notes/index', [ 'notes' => $notes ]);
});
```

resources/views/notes/index.blade.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Anotações</title>
</head>
<body>
    <div>
        @foreach($notes as $note)
            <h2>{{ $note->title }}</h2>
            @if($note->important)
                *
            @endif
            </h2>
            <h3>{{ $note->body }}</h3>
        @endforeach
    </div>
</body>
</html>
```



Laravel *

Framework PHP para construir aplicações web

Rotas Laravel

As rotas são definidas no arquivo web.php

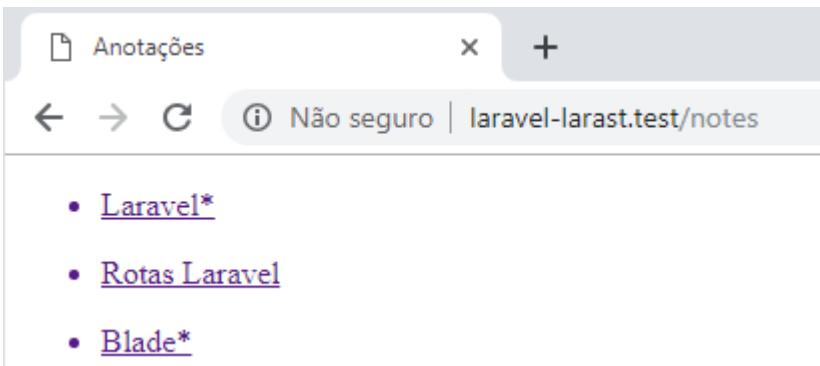
Blade *

Blade é o motor de templates do Laravel

routes/web.php

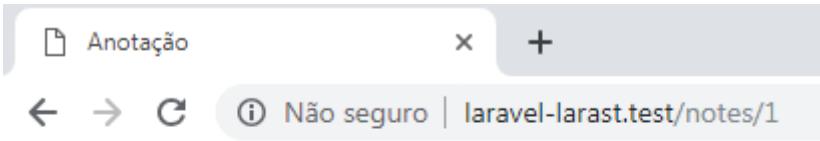
```
Route::get('/notes', function () {
    $notes = DB::table('notes')->get();
    return view('notes/index', [ 'notes' => $notes ]);
});

Route::get('/notes/{id}', function ($id) {
    $note = DB::table('notes')->find($id);
    return view('notes/show', [ 'note' => $note ]);
});
```



A screenshot of a web browser window titled "Anotações". The address bar shows "laravel-larast.test/notes". The page content is a list of links:

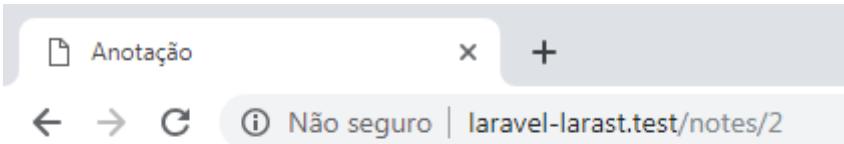
- [Laravel*](#)
- [Rotas Laravel](#)
- [Blade*](#)



A screenshot of a web browser window titled "Anotação". The address bar shows "laravel-larast.test/notes/1". The page content displays the text "Laravel".

Laravel

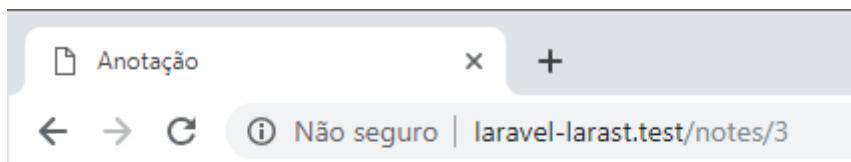
Framework PHP para construir aplicações web



A screenshot of a web browser window titled "Anotação". The address bar shows "laravel-larast.test/notes/2". The page content displays the text "Rotas Laravel".

Rotas Laravel

As rotas são definidas no arquivo web.php

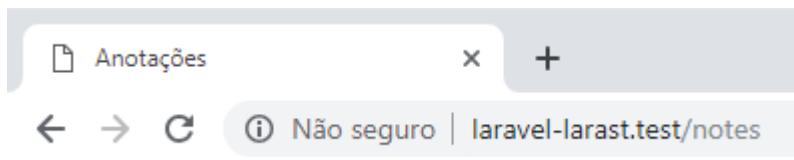


Blade

Blade é o motor de templates do Laravel

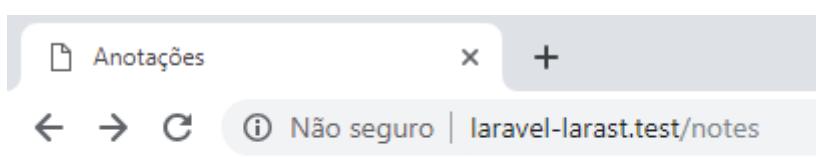
routes/web.php

```
Route::get('/notes', function () {
    $notes = DB::table('notes')->where('important', 1)->get();
    return view('notes/index', [ 'notes' => $notes ]);
});
```



routes/web.php

```
Route::get('/notes', function () {
    $notes = DB::table('notes')->where('important', 0)->get();
    return view('notes/index', [ 'notes' => $notes ]);
});
```



- [Rotas Laravel](#)

Aula 6 - Tinker

Tinker é uma console que permite interagir com os elementos e classes da aplicação, a partir da linha de comandos. Muito útil quando necessitamos ver a saída ou o resultado de uma função, método, etc.

php artisan tinker

```
C:\laragon\www\laravel-larast
λ php artisan tinker
Psy Shell v0.9.6 (PHP 7.2.11 - cli) by Justin Hileman
>>> 4+4
=> 8
>>> snake_case('Olá mundo!')
=> "olá_mundo!"
>>> str_slug('Olá mundo!')
=> "ola-mundo"
>>> DB::table('notes')->get();
=> Illuminate\Support\Collection {#2865
    all: [
        #2871
        +"id": 1,
        +"title": "Laravel",
        +"body": "Framework PHP para construir aplicações web",
        +"important": 1,
        +"created_at": "2019-04-19 17:58:57",
        +"updated_at": "2019-04-19 17:58:57",
    ],
    #2873
    #2873
        +"id": 2,
        +"title": "Rotas Laravel",
        +"body": "As rotas são definidas no arquivo web.php",
        +"important": 0,
        +"created_at": "2019-04-19 18:01:04",
        +"updated_at": "2019-04-19 18:01:04",
    },
    #2874
    #2874
        +"id": 3,
        +"title": "Blade",
        +"body": "Blade é o motor de templates do Laravel",
        +"important": 1,
        +"created_at": "2019-04-19 18:01:55",
        +"updated_at": "2019-04-19 18:01:55",
    ],
}
```

```
>>> DB::table('notes')->find(2);
=> #2861
    +"id": 2,
    +"title": "Rotas Laravel",
    +"body": "As rotas são definidas no arquivo web.php",
    +"important": 0,
    +"created_at": "2019-04-19 18:01:04",
    +"updated_at": "2019-04-19 18:01:04",
}
>>> DB::table('notes')->where('important', 1)->get();
=> Illuminate\Support\Collection {#2868
    all: [
        #2876
            +"id": 1,
            +"title": "Laravel",
            +"body": "Framework PHP para construir aplicações web",
            +"important": 1,
            +"created_at": "2019-04-19 17:58:57",
            +"updated_at": "2019-04-19 17:58:57",
        },
        #2879
            +"id": 3,
            +"title": "Blade",
            +"body": "Blade é o motor de templates do Laravel",
            +"important": 1,
            +"created_at": "2019-04-19 18:01:55",
            +"updated_at": "2019-04-19 18:01:55",
        ],
    ],
}
>>> DB::table('notes')->where('important', 0)->get();
=> Illuminate\Support\Collection {#2880
    all: [
        #2865
            +"id": 2,
            +"title": "Rotas Laravel",
            +"body": "As rotas são definidas no arquivo web.php",
            +"important": 0,
            +"created_at": "2019-04-19 18:01:04",
            +"updated_at": "2019-04-19 18:01:04",
        ],
    ],
}
>>> |
```

Aula 7 - Eloquent - Primeiros passos

É hora de falar de Eloquent, o ORM do Laravel. Os modelos do Eloquent são classes que nos facilitarão enormemente a tarefa de trazer informação da base de dados.

Criando um model

```
php artisan make:model Note
```

```
C:\laragon\www\laravel-larast
λ php artisan make:model Note
Model created successfully.

C:\laragon\www\laravel-larast
λ |
```

app/Note.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Note extends Model
{
    //
}
```

Usando o Tinker para executar métodos do Eloquent

php artisan tinker

```
C:\laragon\www\laravel-larast
λ php artisan tinker
Psy Shell v0.9.6 (PHP 7.2.11 - cli) by Justin Hileman
>>> App\Note::all()
=> Illuminate\Database\Eloquent\Collection {#2877
    all: [
        App\Note {#2878
            id: 1,
            title: "Laravel",
            body: "Framework PHP para construir aplicações web",
            important: 1,
            created_at: "2019-04-19 17:58:57",
            updated_at: "2019-04-19 17:58:57",
        },
        App\Note {#2879
            id: 2,
            title: "Rotas Laravel",
            body: "As rotas são definidas no arquivo web.php",
            important: 0,
            created_at: "2019-04-19 18:01:04",
            updated_at: "2019-04-19 18:01:04",
        },
        App\Note {#2880
            id: 3,
            title: "Blade",
            body: "Blade é o motor de templates do Laravel",
            important: 1,
            created_at: "2019-04-19 18:01:55",
            updated_at: "2019-04-19 18:01:55",
        },
    ],
}
```

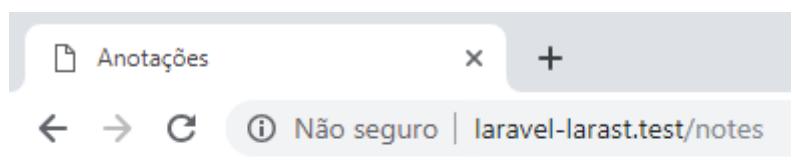
```
>>> App\Note::where('important', '=', 1)->get()
=> Illuminate\Database\Eloquent\Collection {#2869
    all: [
        App\Note {#2881
            id: 1,
            title: "Laravel",
            body: "Framework PHP para construir aplicações web",
            important: 1,
            created_at: "2019-04-19 17:58:57",
            updated_at: "2019-04-19 17:58:57",
        },
        App\Note {#2863
            id: 3,
            title: "Blade",
            body: "Blade é o motor de templates do Laravel",
            important: 1,
            created_at: "2019-04-19 18:01:55",
            updated_at: "2019-04-19 18:01:55",
        },
    ],
}
```

```
>>> App\Note::where('important', 1)->get()
=> Illuminate\Database\Eloquent\Collection {#2878
    all: [
        App\Note {#2873
            id: 1,
            title: "Laravel",
            body: "Framework PHP para construir aplicações web",
            important: 1,
            created_at: "2019-04-19 17:58:57",
            updated_at: "2019-04-19 17:58:57",
        },
        App\Note {#2880
            id: 3,
            title: "Blade",
            body: "Blade é o motor de templates do Laravel",
            important: 1,
            created_at: "2019-04-19 18:01:55",
            updated_at: "2019-04-19 18:01:55",
        },
    ],
}
>>> App\Note::pluck('title')
=> Illuminate\Support\Collection {#2884
    all: [
        "Laravel",
        "Rotas Laravel",
        "Blade",
    ],
}
>>> |
```

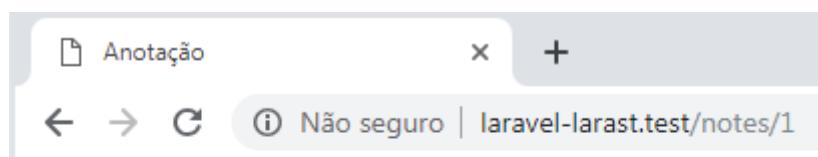
routes/web.php

```
Route::get('/notes', function () {
    $notes = App\Note::all();
    return view('notes/index', [ 'notes' => $notes ]);
});

Route::get('/notes/{id}', function ($id) {
    $note = App\Note::find($id);
    return view('notes/show', [ 'note' => $note ]);
});
```

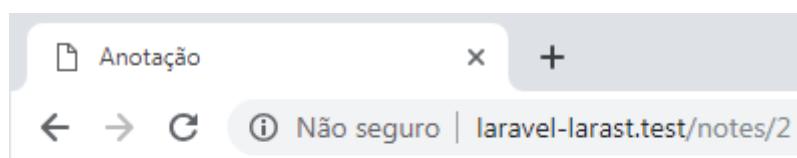


- [Laravel*](#)
- [Rotas Laravel](#)
- [Blade*](#)



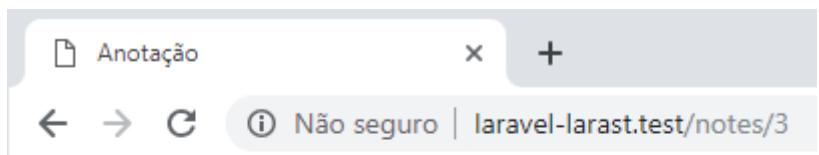
Laravel

Framework PHP para construir aplicações web



Rotas Laravel

As rotas são definidas no arquivo web.php



Blade é o motor de templates do Laravel

app/Note.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Note extends Model
{
    public function isImportant()
    {
        return false;
    }
}
```

```
C:\laragon\www\laravel-larast
λ php artisan tinker
Psy Shell v0.9.6 (PHP 7.2.11 - cli) by Justin Hileman
>>> $note = App\Note::find(2)
=> App\Note {#2875
    id: 2,
    title: "Rotas Laravel",
    body: "As rotas são definidas no arquivo web.php",
    important: 0,
    created_at: "2019-04-19 18:01:04",
    updated_at: "2019-04-19 18:01:04",
}
>>> $note->isImportant()
=> false
>>> $note->title
=> "Rotas Laravel"
>>> $note->body
=> "As rotas são definidas no arquivo web.php" use Illum
>>> $note->important
=> 0
>>> |
```

app/Note.php

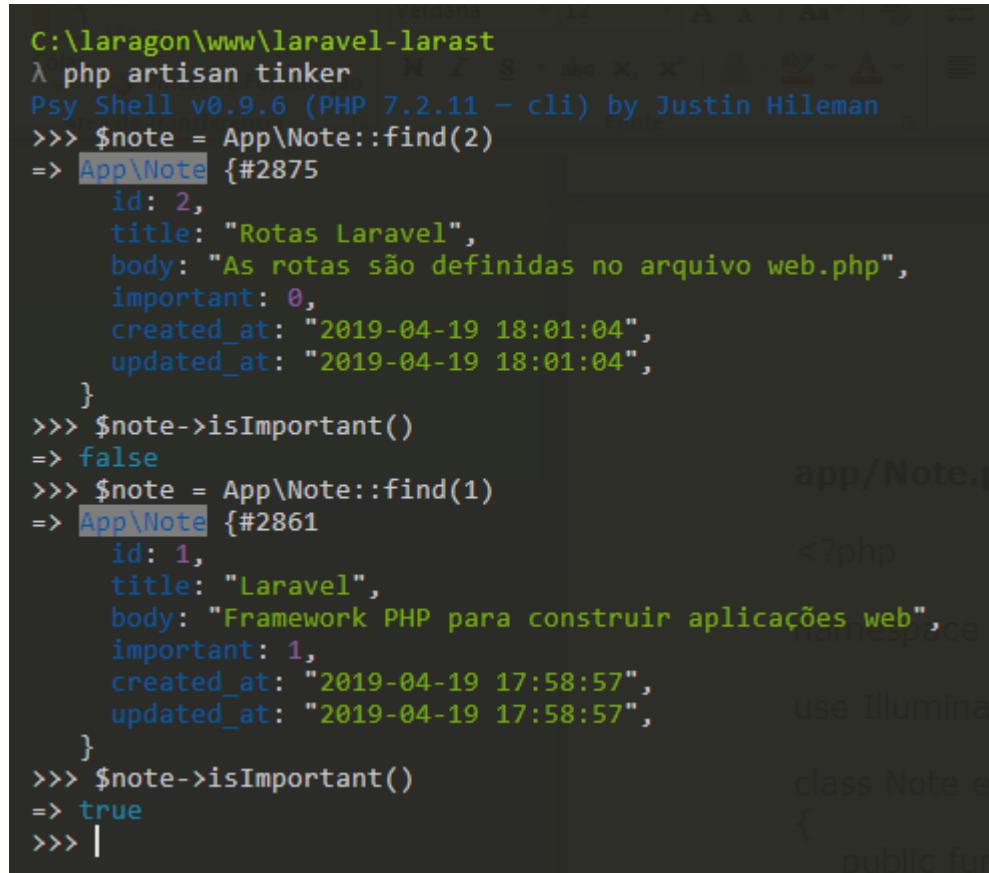
```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Note extends Model
{
    // protected $table = 'anotacoes';

    public function isImportant()
    {
        return $this->important == 1;
    }
}
```



C:\laragon\www\laravel-larast
λ php artisan tinker
Psy Shell v0.9.6 (PHP 7.2.11 - cli) by Justin Hileman
>>> \$note = App\Note::find(2)
=> App\Note {#2875
 id: 2,
 title: "Rotas Laravel",
 body: "As rotas são definidas no arquivo web.php",
 important: 0,
 created_at: "2019-04-19 18:01:04",
 updated_at: "2019-04-19 18:01:04",
}
>>> \$note->isImportant()
=> false
>>> \$note = App\Note::find(1)
=> App\Note {#2861
 id: 1,
 title: "Laravel",
 body: "Framework PHP para construir aplicações web",
 important: 1,
 created_at: "2019-04-19 17:58:57",
 updated_at: "2019-04-19 17:58:57",
}
>>> \$note->isImportant()
=> true
>>> |

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Note extends Model
{
    public function isImportant()
    {
        return $this->important == 1;
    }
}
```

resources/views/notes/index.blade.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Anotações</title>
</head>
<body>
    <div>
        @foreach($notes as $note)
            <ul>
                <a href="/notes/{{ $note->id }}">
                    <li>{{ $note->title }}</li>
                    @if($note->isImportant())
                        *
                    @endif
                </a>
            </ul>
        @endforeach
    </div>
</body>
</html>
```

routes/web.php

```
use App\Note;

Route::get('/notes', function () {
    $notes = Note::all();
    return view('notes/index', [ 'notes' => $notes ]);
});

Route::get('/notes/{id}', function ($id) {
    $note = Note::find($id);
    return view('notes/show', [ 'note' => $note ]);
});
```

Aula 8 - Controllers

Veremos agora como agrupar dentro de Controladores a lógica que temos visto escrevendo o arquivo de rotas. Isto permitirá organizar de uma maneira melhor nosso código e deixar o arquivo de rotas com a única responsabilidade de disparar a lógica adequada dependendo da URL a que se acesse.

Criando um controller para as anotações

```
php artisan make:controller NotesController
```

```
C:\laragon\www\laravel-larast
λ php artisan make:controller NotesController
Controller created successfully.

C:\laragon\www\laravel-larast
λ |
```

app/Http/Controllers/NotesController.php

```
<?php

namespace App\Http\Controllers;

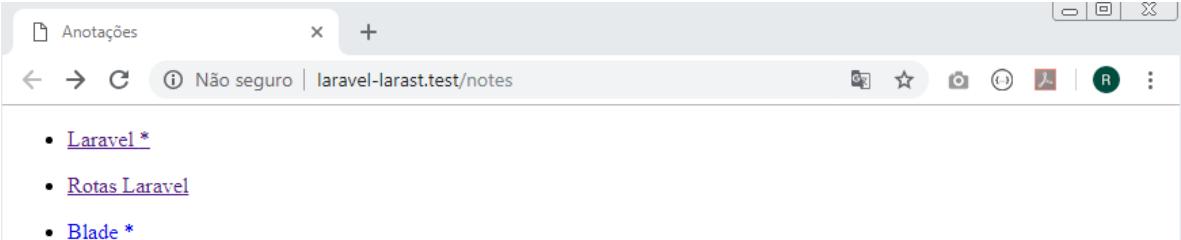
use App>Note;
use Illuminate\Http\Request;

class NotesController extends Controller
{
    public function index(){
        $notes = Note::all();
        return view('notes/index', [ 'notes' => $notes ]);
    }

    public function show($id){
        $note = Note::find($id);
        return view('notes/show', [ 'note' => $note ]);
    }
}
```

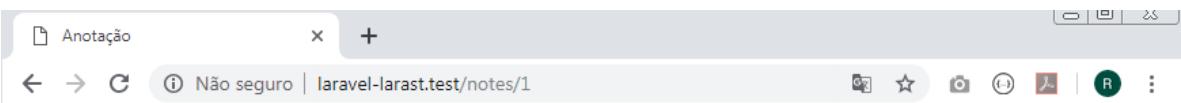
routes/web.php

```
Route::get('/notes', 'NotesController@index');  
Route::get('/notes/{id}', 'NotesController@show');
```



A screenshot of a web browser window. The title bar says "Anotações". The address bar shows "Não seguro | laravel-larast.test/notes". The main content area contains a bulleted list:

- [Laravel *](#)
- [Rotas Laravel](#)
- [Blade *](#)



A screenshot of a web browser window. The title bar says "Anotação". The address bar shows "Não seguro | laravel-larast.test/notes/1".

Laravel

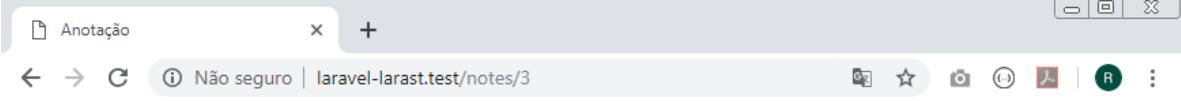
Framework PHP para construir aplicações web



A screenshot of a web browser window. The title bar says "Anotação". The address bar shows "Não seguro | laravel-larast.test/notes/2".

Rotas Laravel

As rotas são definidas no arquivo web.php



A screenshot of a web browser window. The title bar says "Anotação". The address bar shows "Não seguro | laravel-larast.test/notes/3".

Blade

Blade é o motor de templates do Laravel

Aula 9 - Laravel mix + Bootstrap

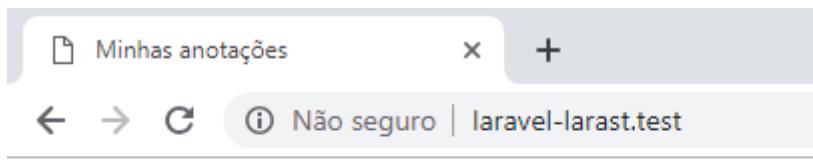
Começaremos a desenvolver a parte visual de nossa aplicação fazendo uso dos estilos de Bootstrap. E faremos usando Laravel mix, um pacote que nos oferece uma maneira conveniente de usar webpack para compilar css, sass, less, js, etc.

resources/views/welcome.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Minhas anotações</title>
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-
bootstrap/3.3.7/css/bootstrap.min.css">
</head>
<body>
    <div class="container">
        <h1>Minhas anotações</h1>
        <button type="button" class="btn btn-primary">Primary</button>
        <!-- Button trigger modal -->
        <button type="button" class="btn btn-primary btn-lg" data-
toggle="modal" data-target="#myModal">
            Launch demo modal
        </button>

        <!-- Modal -->
        <div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-
labelledby="myModalLabel">
            <div class="modal-dialog" role="document">
                <div class="modal-content">
                    <div class="modal-header">
                        <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
                        <h4 class="modal-title" id="myModalLabel">Modal title</h4>
                    </div>
                    <div class="modal-body">
                        ...
                    </div>
                    <div class="modal-footer">
                        <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
                        <button type="button" class="btn btn-primary">Save
changes</button>
                    </div>
                </div>
            </div>
        </div>
    </div>
</body>
```

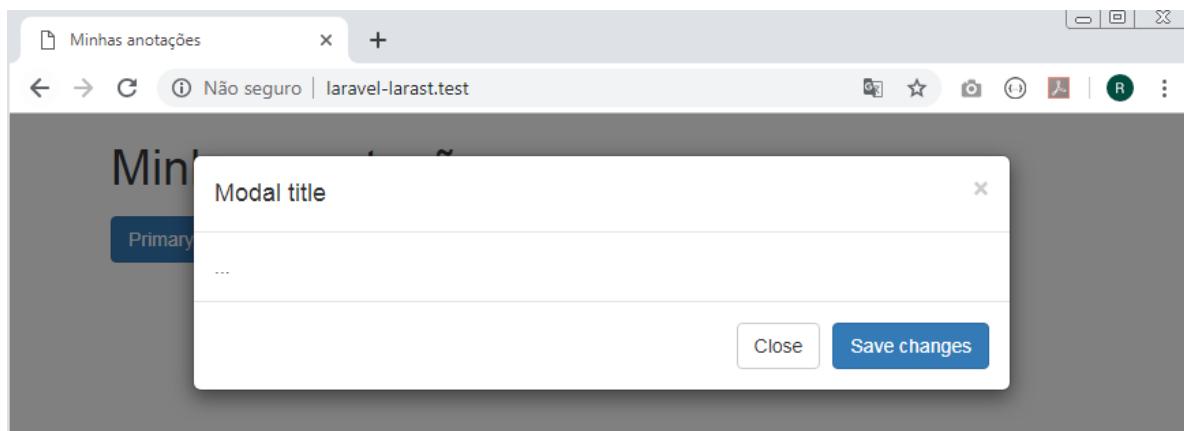
```
<script  
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>  
>  
  <script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-  
bootstrap/3.3.7/js/bootstrap.min.js"></script>  
</body>  
</html>
```



Minhas anotações

Primary

Launch demo modal

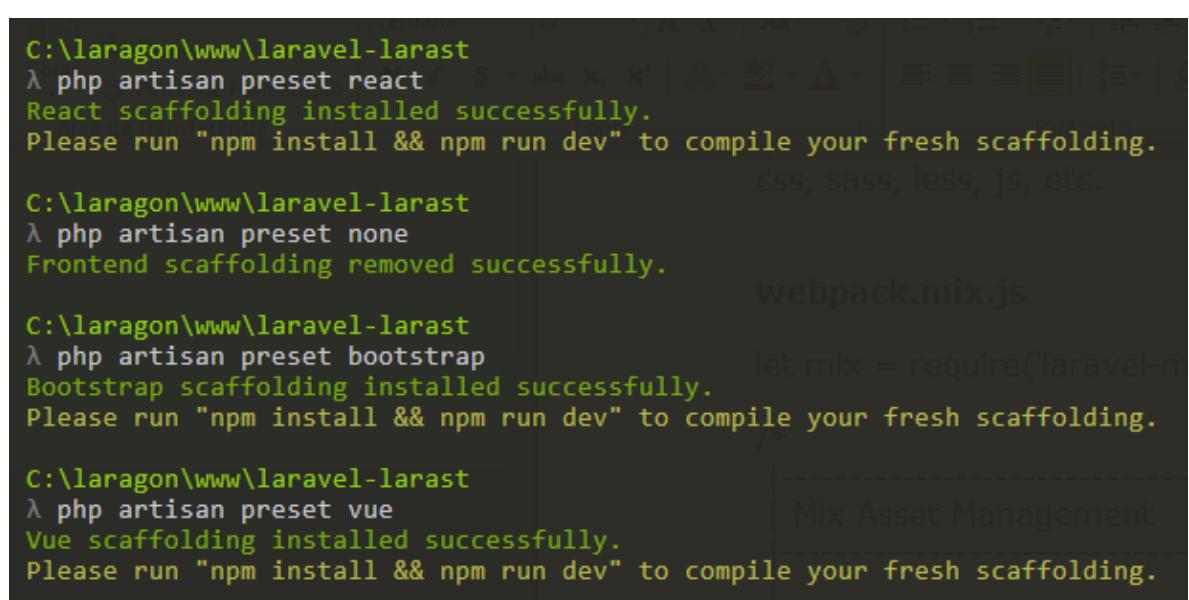


webpack.mix.js

```
let mix = require('laravel-mix');

/*
|-----|
| Mix Asset Management |
|-----|
|
| Mix provides a clean, fluent API for defining some Webpack build steps
| for your Laravel application. By default, we are compiling the Sass
| file for the application as well as bundling up all the JS files.
|
*/
```

```
mix.js('resources/assets/js/app.js', 'public/js')
    .sass('resources/assets/sass/app.scss', 'public/css');
```



The screenshot shows a terminal window with four separate command executions:

- `C:\laragon\www\laravel-larast`
λ php artisan preset react
React scaffolding installed successfully.
Please run "npm install && npm run dev" to compile your fresh scaffolding.
- `C:\laragon\www\laravel-larast`
λ php artisan preset none
Frontend scaffolding removed successfully.
- `C:\laragon\www\laravel-larast`
λ php artisan preset bootstrap
Bootstrap scaffolding installed successfully.
Please run "npm install && npm run dev" to compile your fresh scaffolding.
- `C:\laragon\www\laravel-larast`
λ php artisan preset vue
Vue scaffolding installed successfully.
Please run "npm install && npm run dev" to compile your fresh scaffolding.

On the right side of the terminal, there are two vertical panels with titles:

- webpack.mix.js**: Contains the code snippet from above.
- Mix Asset Management**: Contains the explanatory text about Mix's purpose and usage.

Instalando as dependências

npm install

```
C:\laragon\www\laravel-larast
λ npm install
npm WARN deprecated browserslist@2.11.3: Browserslist 2 could fail on reading Browserslist >3.0 config used in other tools.
npm WARN deprecated browserslist@1.7.7: Browserslist 2 could fail on reading Browserslist >3.0 config used in other tools.

> node-sass@4.11.0 install C:\laragon\www\laravel-larast\node_modules\node-sass
> node scripts/install.js

Cached binary found at C:\Users\Beto\AppData\Roaming\npm-cache\node-sass\4.11.0\win32-ia32-64_binding.node

> uglifyjs-webpack-plugin@0.4.6 postinstall C:\laragon\www\laravel-larast\node_modules\webpack\node_modules\uglifyjs-webpack-plugin
> node lib/post_install.js

> node-sass@4.11.0 postinstall C:\laragon\www\laravel-larast\node_modules\node-sass
> node scripts/build.js

Binary found at C:\laragon\www\laravel-larast\node_modules\node-sass\vendor\win32-ia32-64\binding.node
Testing binary
Binary is fine
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN img-loader@3.0.1 requires a peer of imagemin@^5.0.0 || ^6.0.0 but none is installed. You must install peer dependencies yourself.
npm WARN ajv-keywords@3.4.0 requires a peer of ajv@^6.9.1 but none is installed. You must install peer dependencies yourself.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.8 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.8: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"ia32"})

added 1169 packages from 696 contributors and audited 12566 packages in 669.062s
found 9 vulnerabilities (4 moderate, 5 high)
  run `npm audit fix` to fix them, or `npm audit` for details

C:\laragon\www\laravel-larast
λ |
```

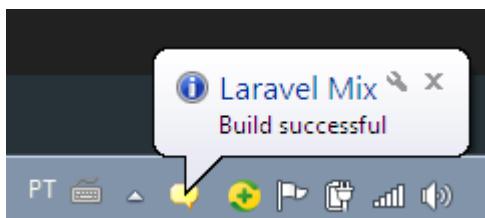
Compilando

npm run dev

```
C:\laragon\www\laravel-larast
λ npm run dev
> @ dev C:\laragon\www\laravel-larast
> npm run development
    npm install

> @ development C:\laragon\www\laravel-larast
> cross-env NODE_ENV=development node_modules/webpack/bin/webpack.js --progress --hide-modules --config=node_modules/laravel-mix/setup/webpack.config.js
10% building modules 8/12 modules 4 active ...ts\components\ExampleComponent.vue[ parser: "babylon" ] is deprecated; we now treat it as { parser: "babel" }.
95% emitting
 DONE Compiled successfully in 27734ms
03:40:11

      Asset            Size  Chunks          Chunk Names
  /js/app.js       1.43 MB        0  [emitted]  [big]  /js/app
 /css/app.css     219 kB         0  [emitted]
C:\laragon\www\laravel-larast
λ |
```



resources/views/welcome.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Minhas anotações</title>
    <link rel="stylesheet" href="/css/app.css">
</head>
<body>
    <div class="container">
        <h1>Minhas anotações</h1>
        <button type="button" class="btn btn-primary">Primary</button>
        <!-- Button trigger modal -->
        <button type="button" class="btn btn-primary btn-lg" data-
toggle="modal" data-target="#myModal">
            Launch demo modal
        </button>

        <!-- Modal -->
        <div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-
labelledby="myModalLabel">
            <div class="modal-dialog" role="document">
                <div class="modal-content">
                    <div class="modal-header">
                        <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
                        <h4 class="modal-title" id="myModalLabel">Modal title</h4>
                    </div>
                    <div class="modal-body">
                        ...
                    </div>
                    <div class="modal-footer">
                        <button type="button" class="btn btn-default" data-
dismiss="modal">Close</button>
                        <button type="button" class="btn btn-primary">Save
changes</button>
                    </div>
                </div>
            </div>
        </div>
        <script src="/js/app.js"></script>
    </body>
</html>
```

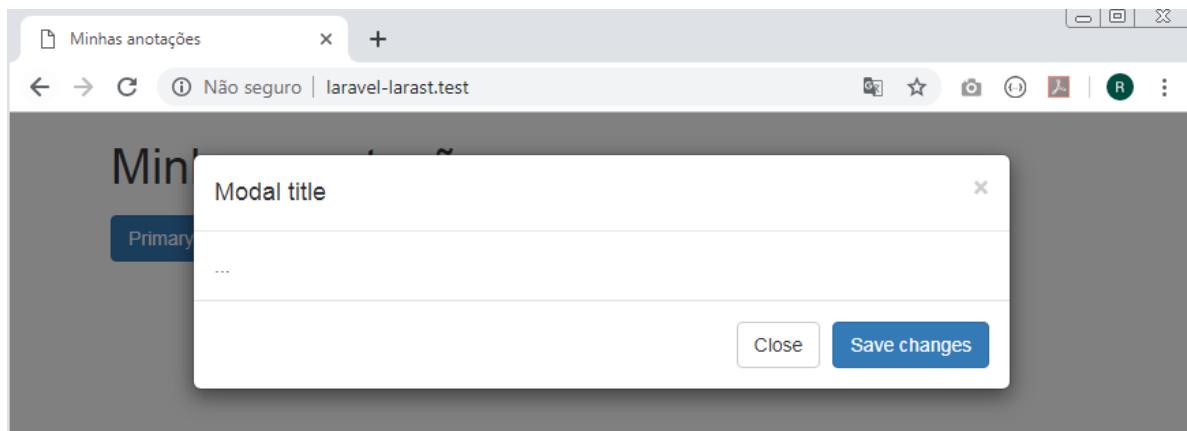
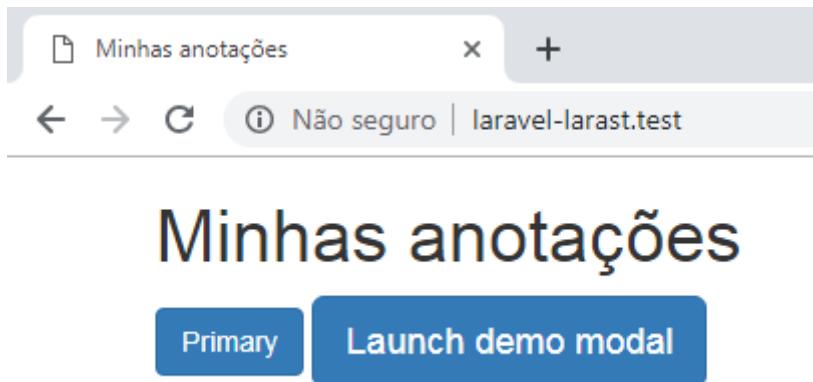
npm run dev

```
C:\laragon\www\laravel-larast
$ npm run dev
> @ dev C:\laragon\www\laravel-larast
> npm run development

> @ development C:\laragon\www\laravel-larast
> cross-env NODE_ENV=development node_modules/webpack/bin/webpack.js --progress --hide-modules --config=node_modules/laravel-mix/setup/webpack.config.js
10% building modules 8/12 modules 4 active ...ts\components\ExampleComponent.vue{ parser: "babylon" } is deprecated; we now treat it as { parser: "babel" }.
95% emitting
 DONE Compiled successfully in 35219ms
 Asset      Size  Chunks   Chunk Names
 /js/app.js  1.43 MB    0  [emitted]  [big]  /js/app
 /css/app.css 219 kB     0  [emitted]           /js/app
04:01:40
```

npm run watch

Com este comando toda vez que for feita alguma alteração nos arquivos css ou js, é feita automaticamente uma recompilação.



Aula 10 - Layouts

Agora começaremos a dar algo mais de estrutura aos views usando Bootstrap. Ademais, com o fim de evitar a duplicação de código ao escrever views, faremos uso de layouts ou de templates básicos.

routes/web.php

```
Route::get('/notes', 'NotesController@index');
Route::get('/notes/{id}', 'NotesController@show');

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

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

resources/views/welcome.blade.php

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

@section('title')
    {{ config('app.name') }}
@endsection

@section('content')
    <h1>Bem-vindo às {{ config('app.name') }}</h1>
@endsection
```

resources/views/contact.blade.php

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

@section('title')
    Contato
@endsection

@section('content')
    <h1>Contatos</h1>
@endsection
```

resources/views/notes/show.blade.php

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

@section('title')
    Anotação: {{ $note->title }}
@endsection

@section('content')
    <h1>{{ $note->title }}</h1>
    <pre>{{ $note->body }}</pre>
@endsection
```

resources/views/notes/index.blade.php

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

@section('title')
    Listagem de anotações
@endsection

@section('content')
    <div>
        @foreach($notes as $note)
            <ul>
                <a href="notes/{{ $note->id }}">
                    <li>{{ $note->title }}</li>
                    @if($note->isImportant())
                        *
                    @endif
                </a>
            </ul>
        @endforeach
    </div>
@endsection
```

resources/views/layouts/app.blade.php

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>@yield('title')</title>
    <link rel="stylesheet" href="/css/app.css">
</head>
<body>

    <div class="container">
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <a class="navbar-brand" href="/">{{ config('app.name') }}</a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav">
                    <li class="{{ Request::is('notes') ? 'active' : '' }}>
                        <a class="nav-link" href="/notes">Anotações <span class="sr-only">(current)</span></a>
                    </li>
                    <li class="{{ Request::is('contact') ? 'active' : '' }}>
                        <a class="nav-link" href="/contact">Contato</a>
                    </li>
                </ul>
            </div>
        </nav>
    </div>

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

    <footer>
        <div class="container">
            {{ config('app.name') }} . Todos os direitos reservados
        </div>
    </footer>

        <script src="/js/app.js"></script>
</body>
</html>
```

resources/assets/sass/app.blade.php

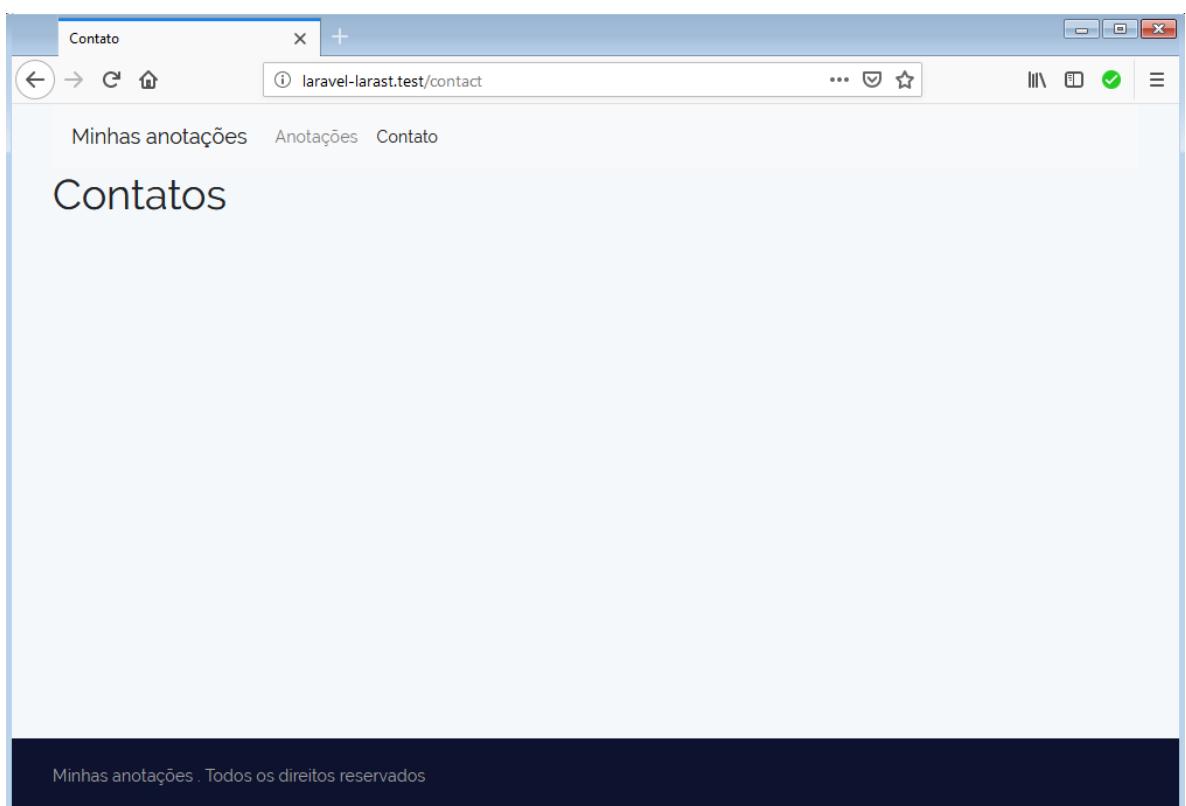
```
// Fonts
@import
url("https://fonts.googleapis.com/css?family=Raleway:300,400,600");

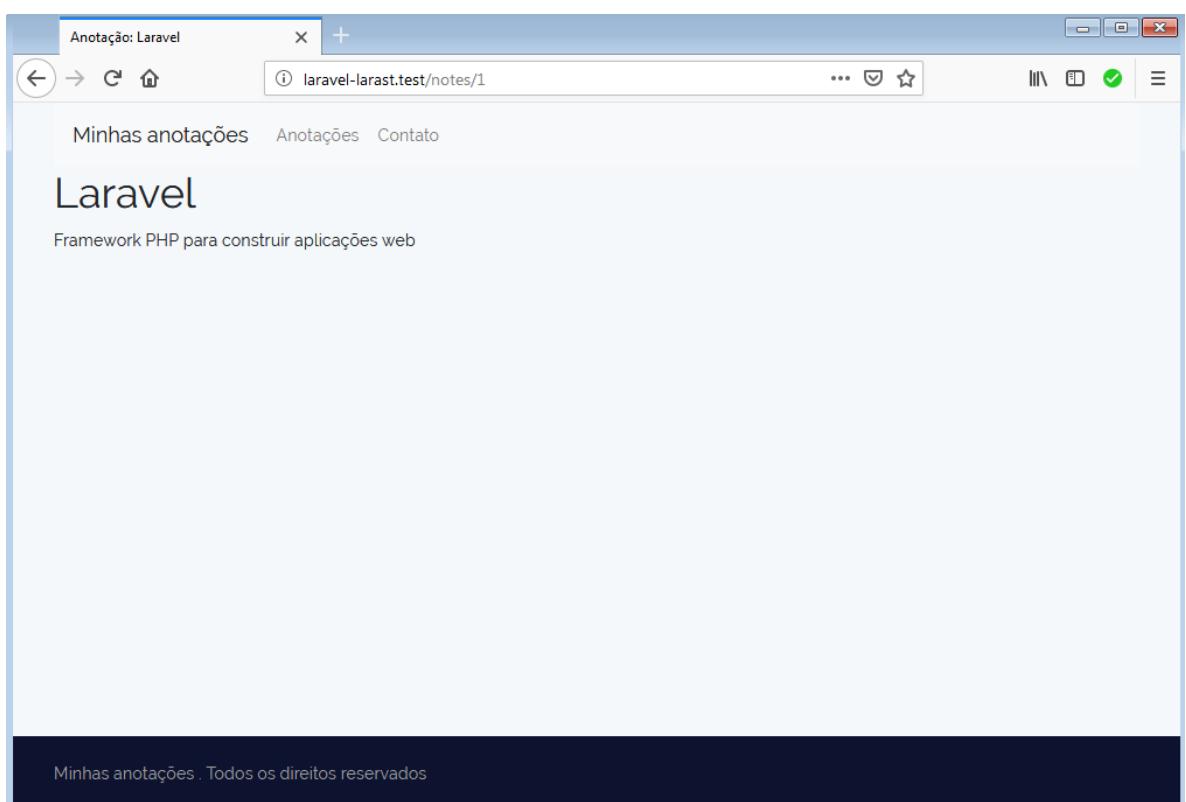
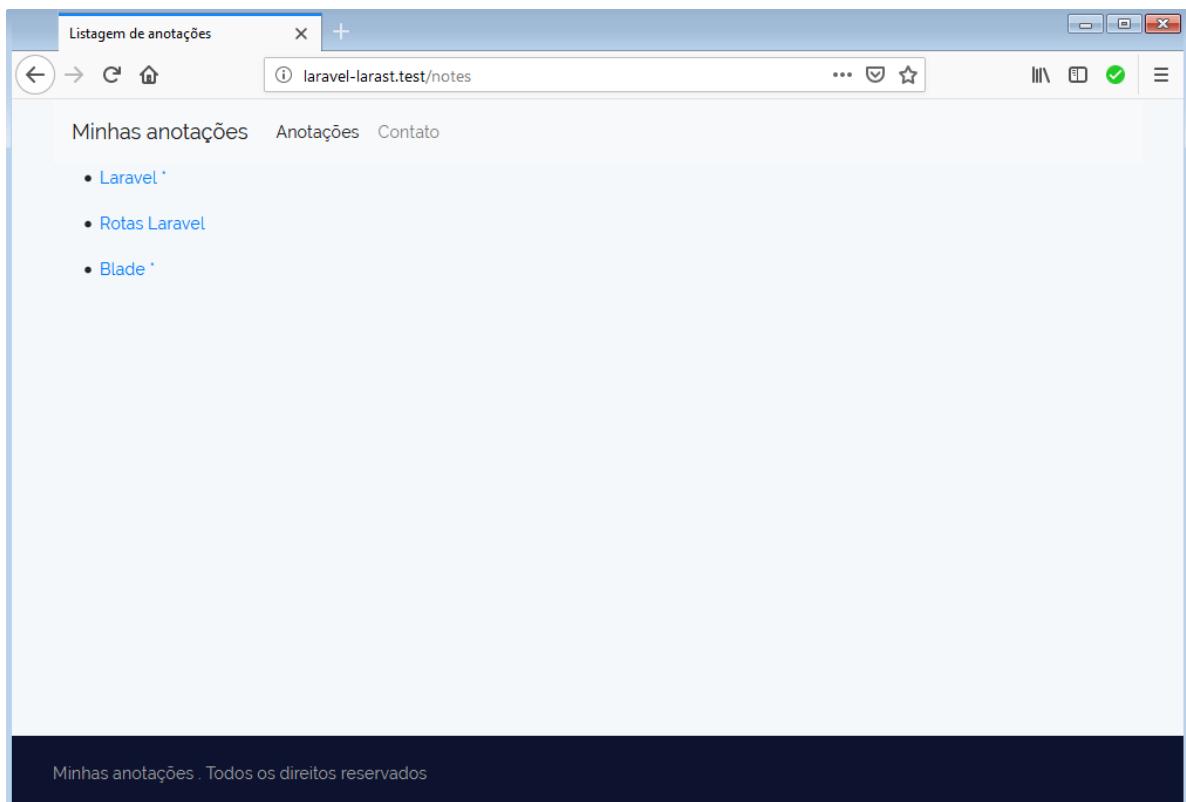
// Variables
@import "variables";

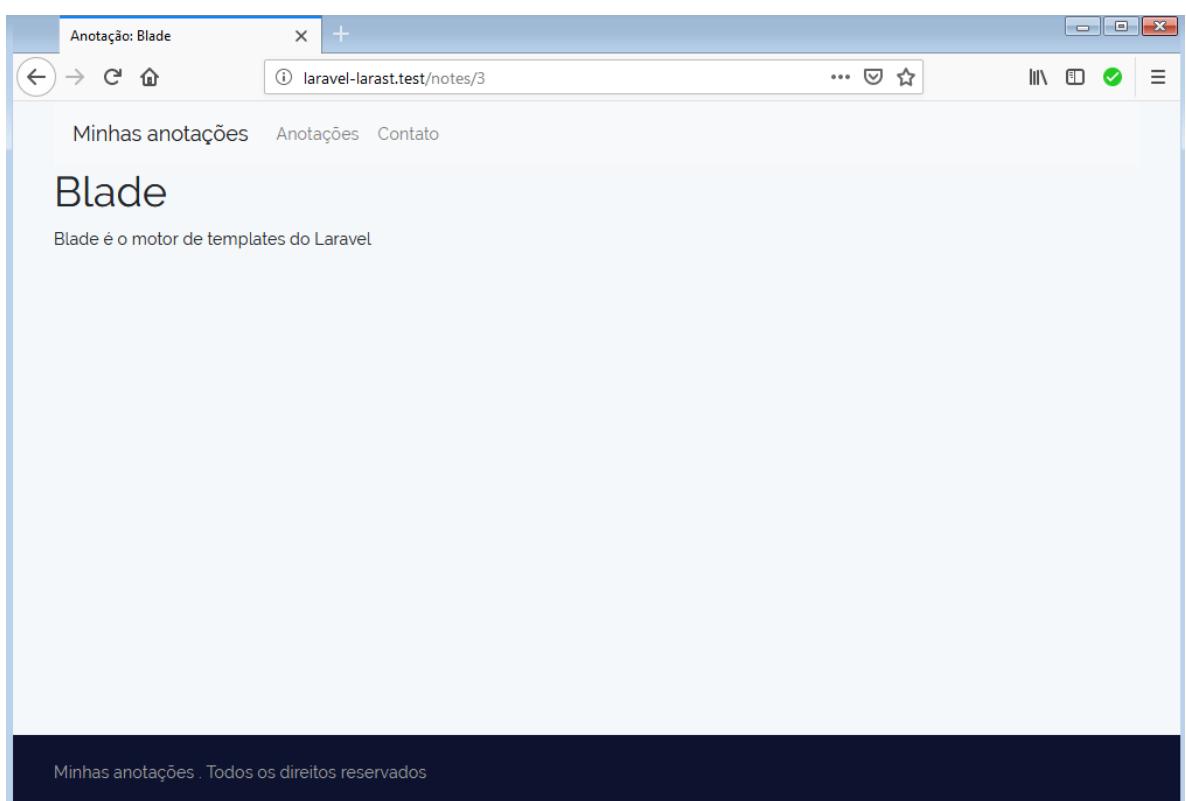
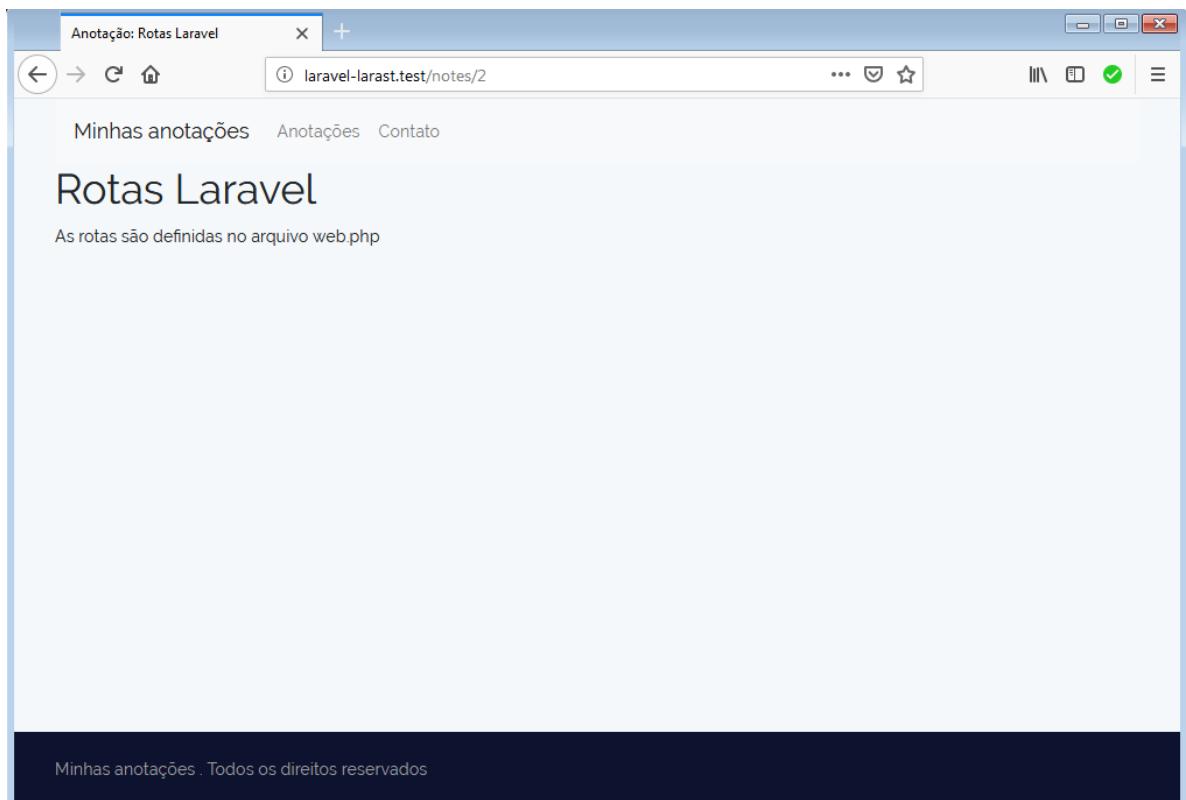
// Bootstrap
@import '~bootstrap/scss/bootstrap';
// @import '~bootstrap-sass/assets/stylesheets/bootstrap';

// .navbar-laravel {
//   background-color: #fff;
//   box-shadow: 0 2px 4px rgba(0, 0, 0, 0.04);
// }

footer {
  color: #9F9F9F;
  background-color: #0D122E;
  padding: 20px 0 20px 0;
  position: fixed;
  bottom: 0;
  width: 100%;
}
```







Aula 11 - Route Model Binding

Uma operação muito comum na maior parte de projetos, é de fazer consultas a base de dados baseados em um id que chega na URL. Laravel oferece uma maneira de automatizar este processo de busca.

routes/web.php

```
Route::get('/notes', 'NotesController@index');
Route::get('/notes/{note}', 'NotesController@show');

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

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

app/Http/Controllers/NotesController.php

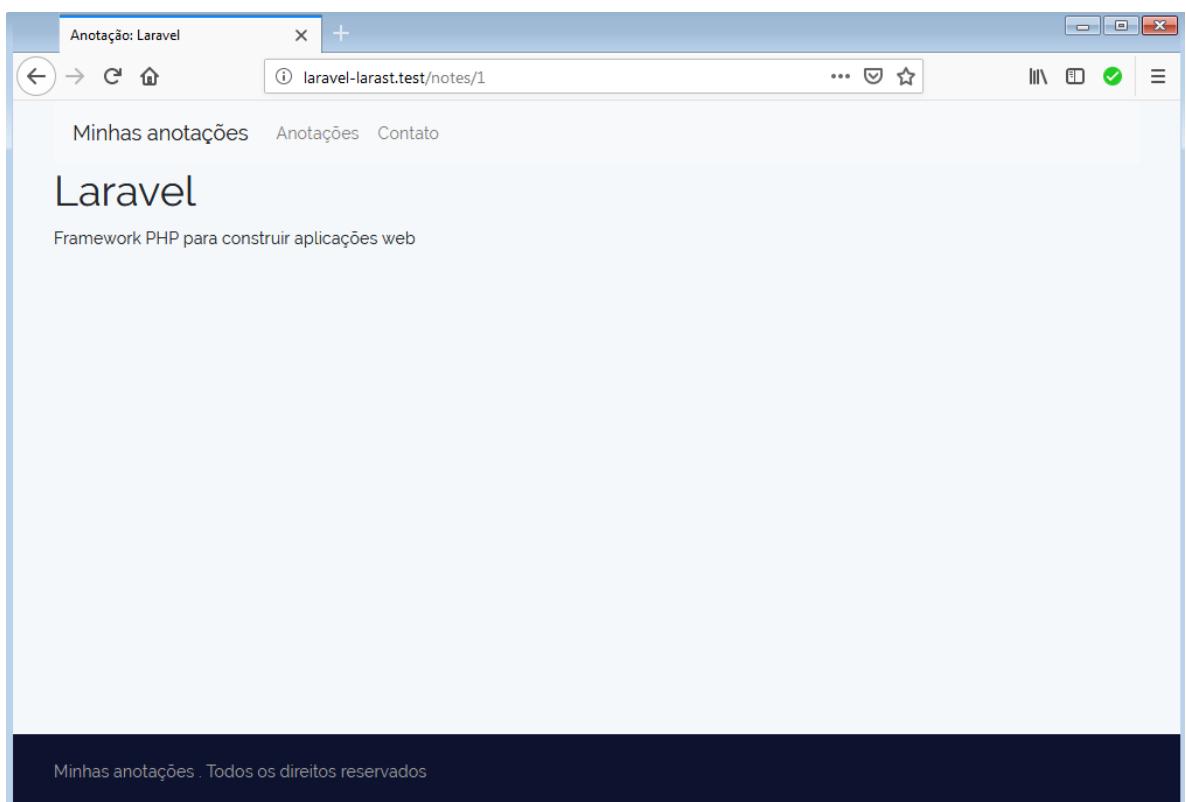
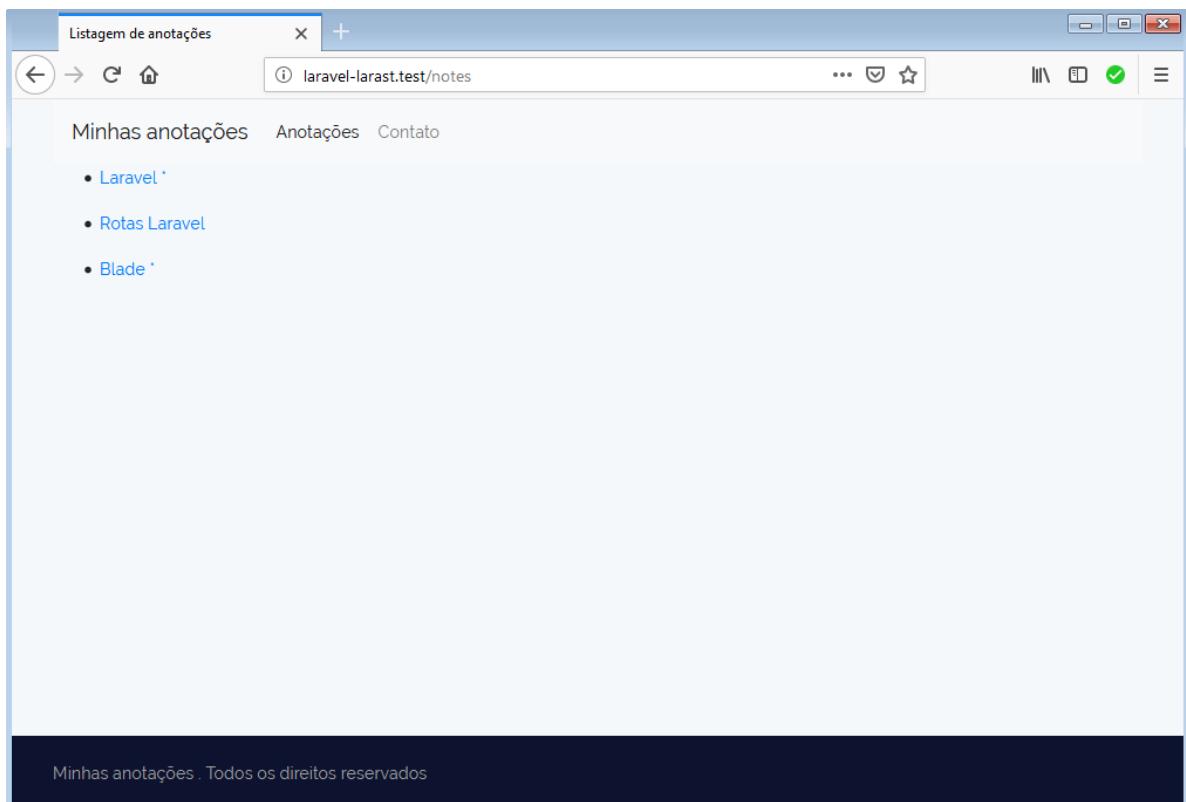
```
<?php

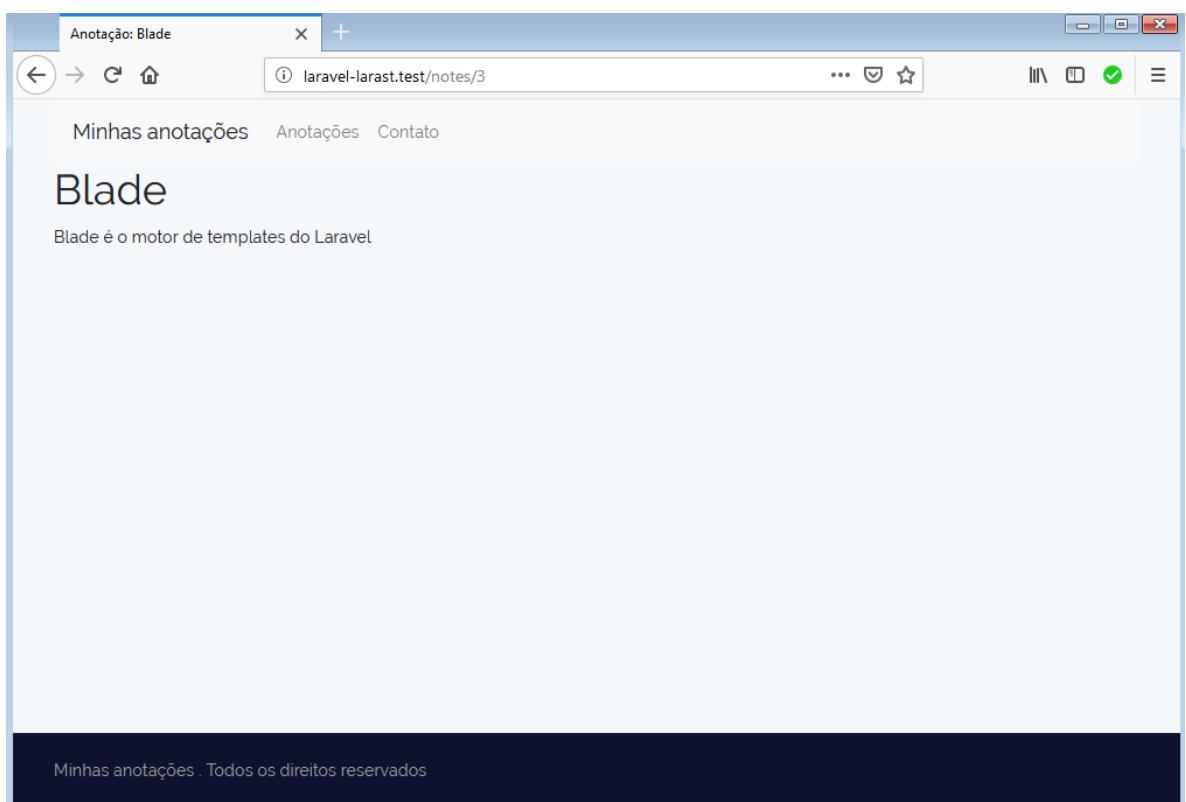
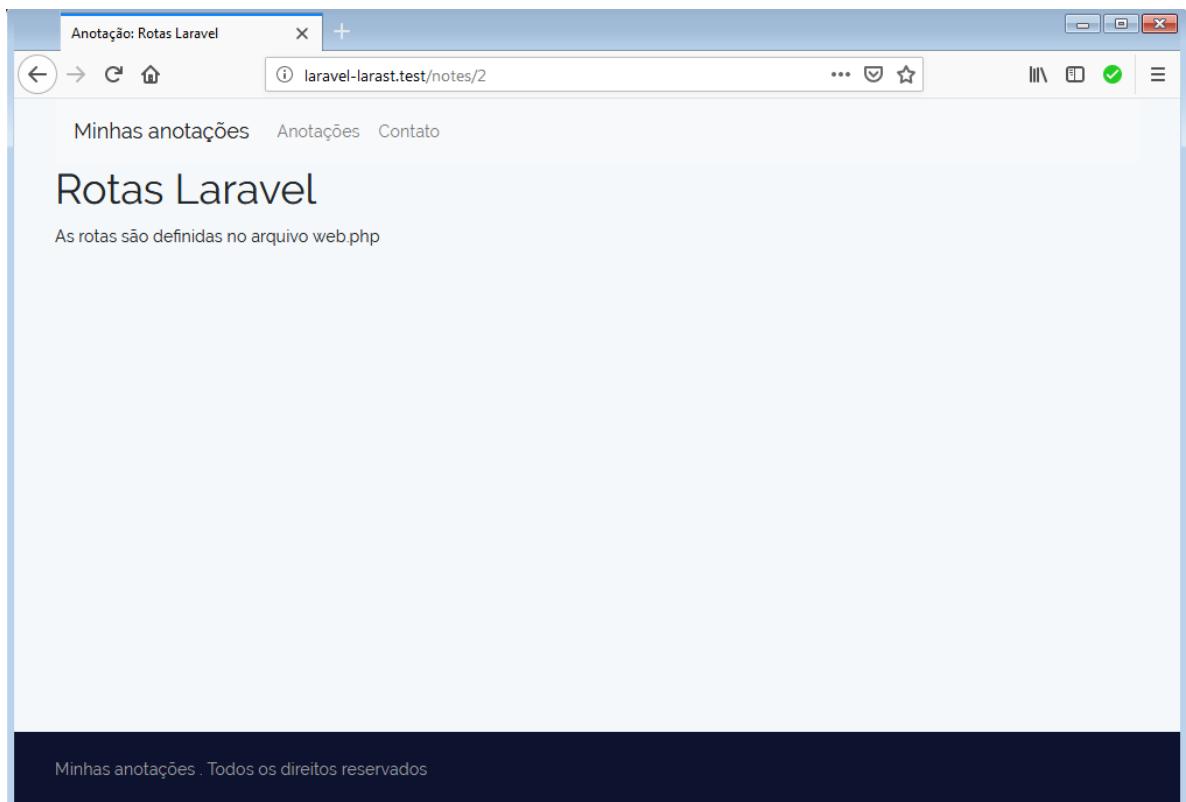
namespace App\Http\Controllers;

use App\Note;
use Illuminate\Http\Request;

class NotesController extends Controller
{
    public function index(){
        $notes = Note::all();
        return view('notes/index', [ 'notes' => $notes ]);
    }

    public function show(Note $note){
        return view('notes/show', [ 'note' => $note ]);
    }
}
```





Aula 12 - Guardar formulário na base de dados

Passemos agora a manipular formulários e armazenar essa informação na base de dados. Esta é uma das maneiras mais comuns nas quais a aplicação receberá informação dos usuários. E claro, sempre que recebe informação do exterior, devemos contemplar a possibilidade de receber diferentes tipos de ataques. Felizmente Laravel por padrão nos protege de alguns dos ataques mais comuns.

Instale no Sublime Text o pacote [Bootstrap 3 Snippets](#).

database/migrations/2019_04_19_071011_create_notes_table.php

```
<?php

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

class CreateNotesTable extends Migration
{
    public function up()
    {
        Schema::create('notes', function (Blueprint $table) {
            $table->increments('id');
            $table->string('title');
            $table->text('body');
            $table->boolean('important')->default(false);
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('notes');
    }
}
```

```
php artisan migrate:refresh
```

```
C:\laragon\www\laravel-larast
λ php artisan migrate:refresh
Rolling back: 2019_04_19_080304_add_user_id_column_to_notes_table
Rolled back: 2019_04_19_080304_add_user_id_column_to_notes_table
Rolling back: 2019_04_19_071011_create_notes_table
Rolled back: 2019_04_19_071011_create_notes_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_04_19_071011_create_notes_table
Migrated: 2019_04_19_071011_create_notes_table
Migrating: 2019_04_19_080304_add_user_id_column_to_notes_table
Migrated: 2019_04_19_080304_add_user_id_column_to_notes_table
```

routes/web.php

```
<?php

Route::get('/notes', 'NotesController@index');
Route::get('/notes/create', 'NotesController@create');
// Route::get('/notes/{id}', 'NotesController@show');
Route::get('/notes/{note}', 'NotesController@show');
Route::post('/notes', 'NotesController@store');

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

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

app/Http/Controllers/NotesController.php

```
<?php

namespace App\Http\Controllers;

use App\Note;
use Illuminate\Http\Request;

class NotesController extends Controller
{
    public function index(){
```

```

$notes = Note::all();
return view('notes.index', [ 'notes' => $notes ]);
}

public function create(){
    return view('notes.create');
}

// public function show($id){
//     $note = Note::find($id);
//     return view('notes/show', [ 'note' => $note ]);
// }

public function show(Note $note){
    return view('notes.show', [ 'note' => $note ]);
}

// public function store(){
//     $note = new Note;
//     $note->title = request()->title;
//     $note->body = request()->body;
//     $note->important = is_null(request()->important) ? 0 : 1;
//     $note->save();
// }

public function store(){
    Note::create( request()->all() );
    return redirect('/notes');
}
}

```

resources/views/notes/create.blade.php

```

@extends('layouts.app')

@section('title')
    Criar uma nova anotação
@endsection

@section('content')

<form action="/notes" method="POST" role="form">

{{ csrf_field() }}

<legend>Criar nova anotação</legend>

```

```

<div class="form-group">
    <label for="">Título</label>
    <input name="title" type="text" class="form-control" id=""
placeholder="Digite o título">
</div>

<div class="form-group">
    <label for="">Anotação</label>
    <textarea name="body" type="text" class="form-control" id=""
placeholder="Digite a anotação"></textarea>
</div>

<div class="form-group">
    <input name="important" type="checkbox" value="1"> É importante
</div>

    <button type="submit" class="btn btn-primary">Criar</button>
</form>
@endsection

```

app/Note.php

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Note extends Model
{
    // protected $table = 'anotacoes';

    protected $fillable = [
        'title', 'body', 'important'
    ];

    public function isImportant()
    {
        return $this->important == 1;
    }
}

```

Criar uma nova anotação

Minhas anotações Anotações Contato

Criar nova anotação

Titulo

Digite o título

Anotação

Digite a anotação

É importante

Criar

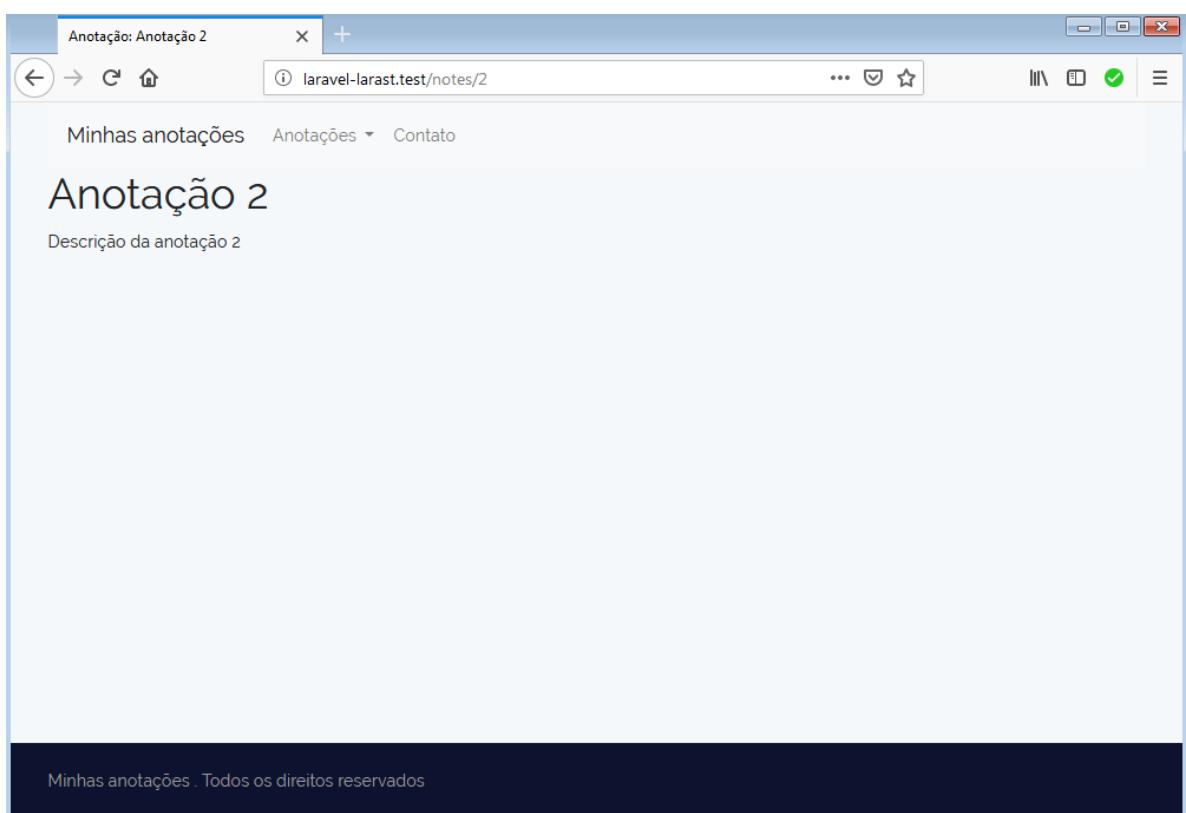
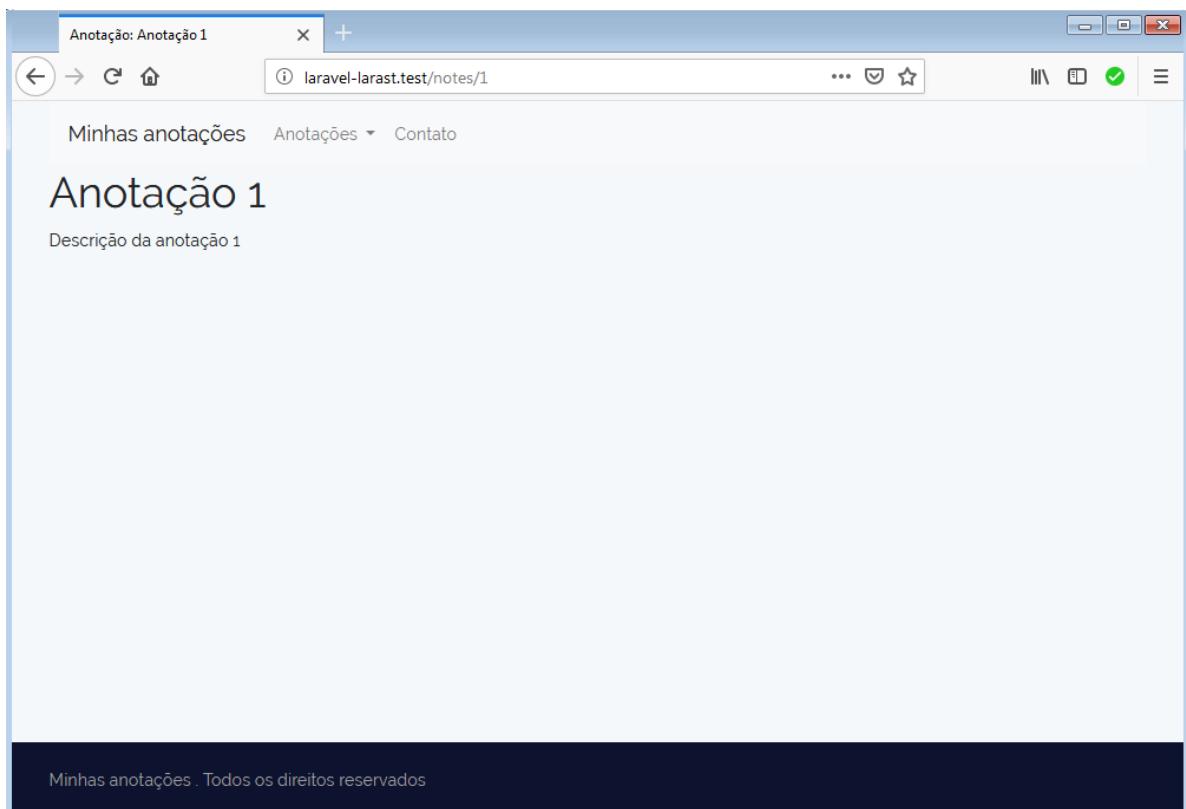
Minhas anotações . Todos os direitos reservados

Listagem de anotações

Minhas anotações Anotações Contato

- Anotação 1
- Anotação 2 *
- Anotação 3

Minhas anotações . Todos os direitos reservados



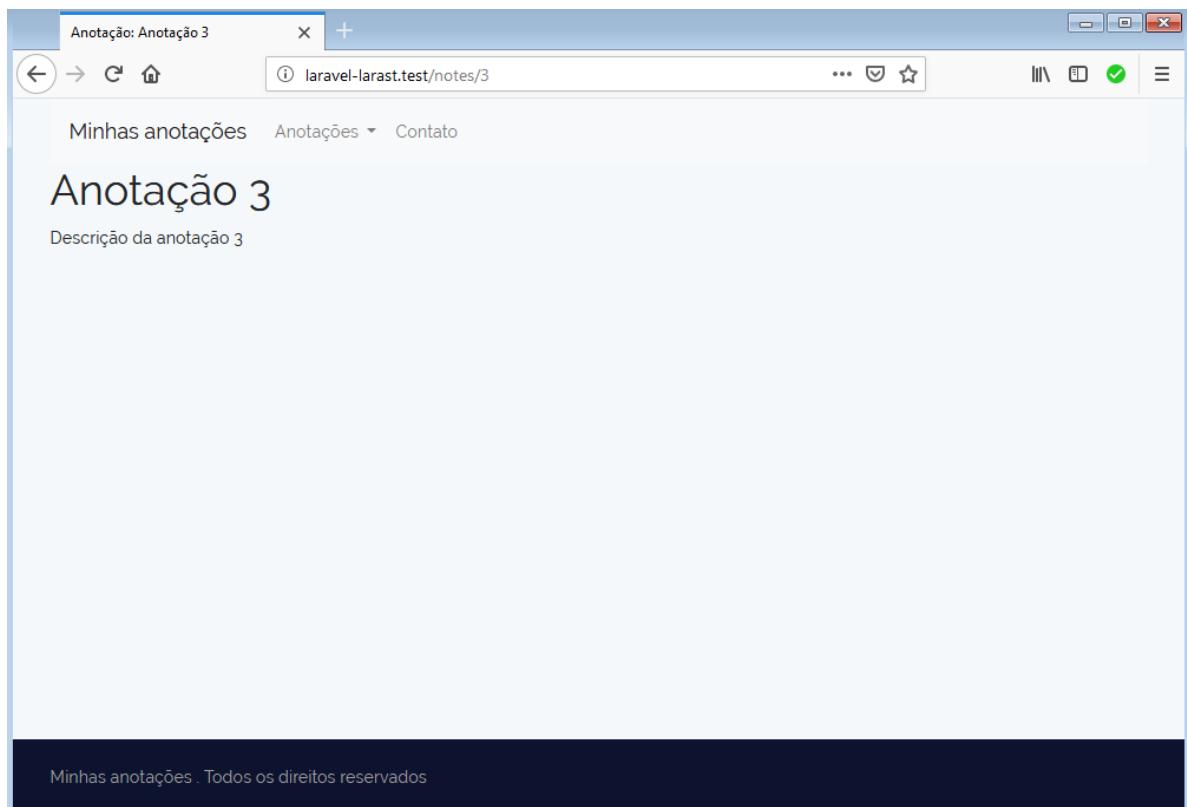
Anotação: Anotação 3

Minhas anotações Anotações Contato

Anotação 3

Descrição da anotação 3

Minhas anotações . Todos os direitos reservados



Server: localhost:3306 » Database: laravel-larast » Table: notes

Browse Structure SQL Search Insert Export Import Privileges Operations

Showing rows 0 - 2 (3 total, Query took 0.0000 seconds.)

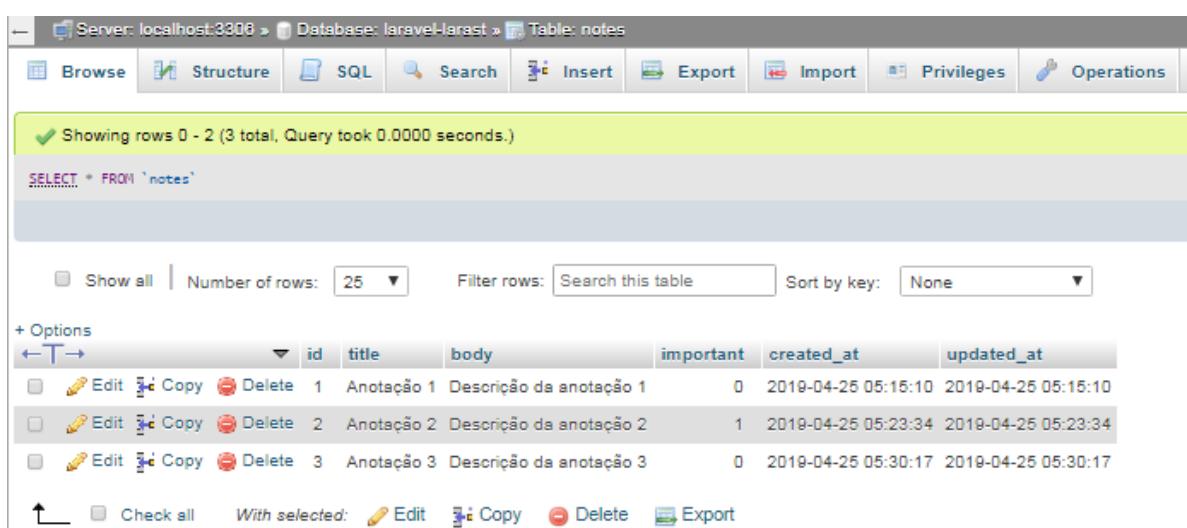
```
SELECT * FROM `notes`
```

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	id	title	body	important	created_at	updated_at
<input type="checkbox"/>	1	Anotação 1	Descrição da anotação 1	0	2019-04-25 05:15:10	2019-04-25 05:15:10
<input type="checkbox"/>	2	Anotação 2	Descrição da anotação 2	1	2019-04-25 05:23:34	2019-04-25 05:23:34
<input type="checkbox"/>	3	Anotação 3	Descrição da anotação 3	0	2019-04-25 05:30:17	2019-04-25 05:30:17

Check all With selected: Edit Copy Delete Export



Aula 13 - View Partials + Formulários

O uso de view partials evita que dupliquemos código em várias situações. Uma delas é quando se manipula formulários de criação e edição, nos quais a maior parte das vezes são muito similares.

resources/views/layouts/app.blade.php

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <title>@yield('title')</title>
  <link rel="stylesheet" href="/css/app.css">
</head>
<body>

  <div class="container">
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
      <a class="navbar-brand" href="/">{{ config('app.name') }}</a>
      <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
          <li class="nav-item dropdown {{ Request::is('notes') ? 'active' : '' }}>
            <a class="nav-link dropdown-toggle" href="#" id="navbardrop" data-toggle="dropdown">
              Anotações
            </a>
            <div class="dropdown-menu">
              <a class="dropdown-item" href="/notes">Ver</a>
              <a class="dropdown-item" href="/notes/create">Criar</a>
            </div>
          </li>
          <li class="{{ Request::is('contact') ? 'active' : '' }}>
            <a class="nav-link" href="/contact">Contato</a>
          </li>
        </ul>
      </div>
    </nav>
  </div>

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

```

<footer>
    <div class="container">
        {{ config('app.name') }} . Todos os direitos reservados
    </div>
</footer>

<script src="/js/app.js"></script>
</body>
</html>

```

resources/views/notes/index.blade.php

```

@extends('layouts.app')

@section('title')
    Listagem de anotações
@endsection

@section('content')
    <table class="table table-striped table-hover">
        <thead>
            <th>Anotação</th>
            <th>Editar</th>
        </thead>
        <tbody>
            @foreach($notes as $note)
                <tr>
                    <td>
                        <a href="/notes/{{ $note->id }}">
                            {{ $note->title }}
                            @if($note->isImportant())
                                *
                            @endif
                        </a>
                    </td>
                    <td>
                        <a href="/notes/{{ $note->id }}/edit" class="btn btn-primary btn-sm">Editar</a>
                    </td>
                </tr>
            @endforeach
        </tbody>
    </table>
@endsection

```

routes/web.php

```
<?php

Route::get('/notes', 'NotesController@index');
Route::get('/notes/create', 'NotesController@create');
// Route::get('/notes/{id}', 'NotesController@show');
Route::get('/notes/{note}', 'NotesController@show');
Route::get('/notes/{note}/edit', 'NotesController@edit');
Route::post('/notes', 'NotesController@store');

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

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

app/Http/Controllers/NotesController.php

```
<?php

namespace App\Http\Controllers;

use App\Note;
use Illuminate\Http\Request;

class NotesController extends Controller
{
    public function index(){
        $notes = Note::all();
        return view('notes.index', [ 'notes' => $notes ]);
    }

    public function create(){
        return view('notes.create');
    }

    // public function show($id){
    // $note = Note::find($id);
    // return view('notes/show', [ 'note' => $note ]);
    // }

    public function show(Note $note){
        return view('notes.show', [ 'note' => $note ]);
    }
}
```

```

// public function store(){
// $note = new Note;
// $note->title = request()->title;
// $note->body = request()->body;
// $note->important = is_null(request()->important) ? 0 : 1;
// $note->save();
// }

public function store(){
    Note::create( request()->all() );
    return redirect('/notes');
}

public function edit(Note $note){
    return view('notes.edit', compact('note'));
}

}

```

resources/views/notes/edit.blade.php

```

@extends('layouts.app')

@section('title')
    Editar: {{ $note->title }}
@endsection

@section('content')
    <form action="/notes/{{ $note->id }}" method="PATCH" role="form">
        {{ csrf_field() }}
        <legend>Editar anotação</legend>
        @include('notes._form')
        <button type="submit" class="btn btn-primary">Editar</button>
    </form>
@endsection

```

resources/views/notes/create.blade.php

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

@section('title')
    Criar uma nova anotação
@endsection

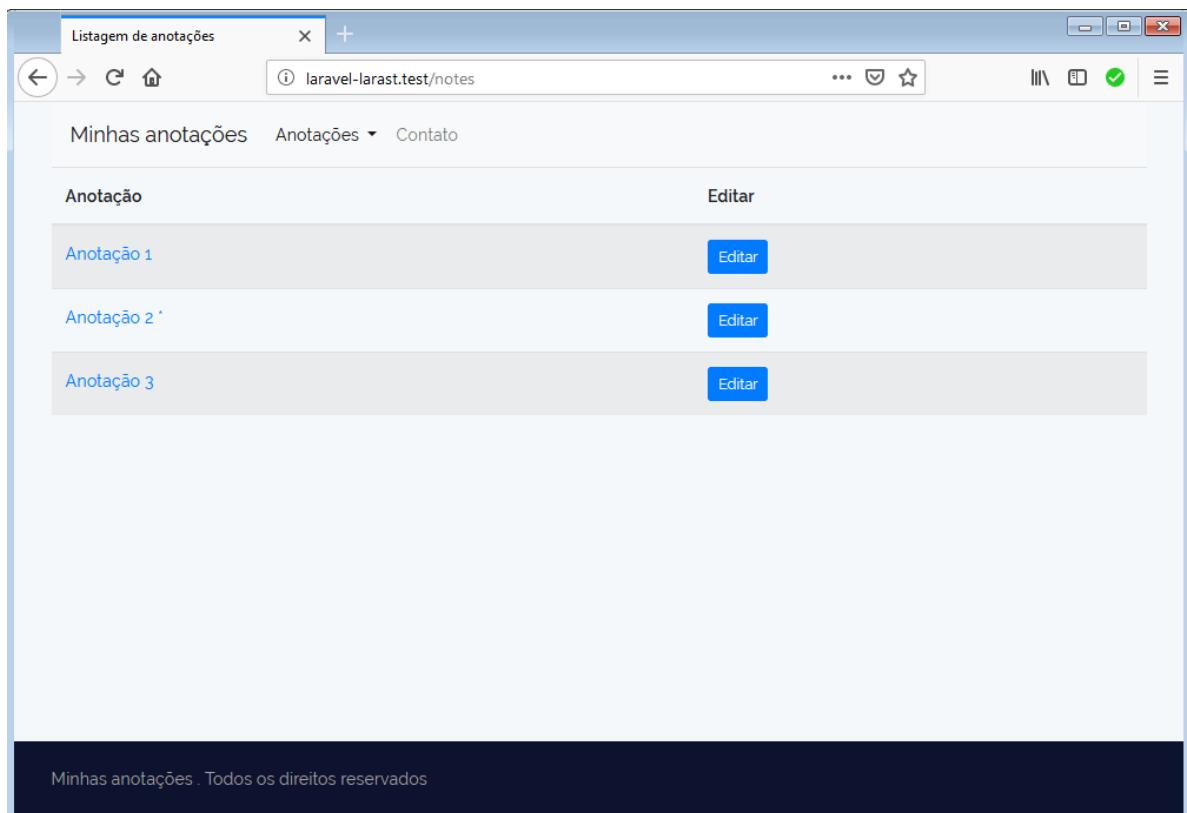
@section('content')
    <form action="/notes" method="POST" role="form">
        {{ csrf_field() }}
        <legend>Criar nova anotação</legend>
        @include('notes._form')
        <button type="submit" class="btn btn-primary">Criar</button>
    </form>
@endsection
```

resources/views/notes/_form.blade.php

```
<div class="form-group">
    <label for="">Título</label>
    <input name="title" value="{{ isset($note) ? $note->title : '' }}" type="text"
    class="form-control" id="" placeholder="Digite o título">
</div>

<div class="form-group">
    <label for="">Anotação</label>
    <textarea name="body" type="text" class="form-control" id=""
    placeholder="Digite a anotação">{{ isset($note) ? $note->body : '' }}</textarea>
</div>

<div class="checkbox">
    <label>
        <input type="checkbox" value="1" name="important"
        @if(isset($note))
            {{ $note->isImportant() ? 'checked' : ''}}
        @endif
        > É importante
    </label>
</div>
```



Editor: Anotação 2

Minhas anotações Anotações Contato

Editar anotação

Titulo

Anotação

É importante

Editar

Minhas anotações . Todos os direitos reservados

Criar uma nova anotação

Minhas anotações Anotações Contato

Criar nova anotação

Titulo

Anotação

É importante

Criar

Minhas anotações . Todos os direitos reservados

Aula 14 - Edição de registros

Com o formulário de edição pronto, veremos como completar a ação de editar/atualizar os dados de uma anotação. Ademais veremos uma forma de solucionar o clássico problema de atualizar campos que se representam mediante controle do tipo checkbox.

routes/web.php

```
<?php

Route::get('/notes', 'NotesController@index');
Route::get('/notes/create', 'NotesController@create');
// Route::get('/notes/{id}', 'NotesController@show');
Route::get('/notes/{note}', 'NotesController@show');
Route::get('/notes/{note}/edit', 'NotesController@edit');
Route::patch('/notes/{note}', 'NotesController@update');
Route::post('/notes', 'NotesController@store');

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

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

resources/views/notes/edit.blade.php

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

@section('title')
    Editar: {{ $note->title }}
@endsection

@section('content')
    <form action="/notes/{{ $note->id }}" method="POST" role="form">
        {{ method_field('PATCH') }}
        {{ csrf_field() }}
        <legend>Editar anotação</legend>
        @include('notes._form')
        <button type="submit" class="btn btn-primary">Editar</button>
    </form>
@endsection
```

app/Http/Controllers/NotesController.php

```
<?php

namespace App\Http\Controllers;

use App\Note;
use Illuminate\Http\Request;

class NotesController extends Controller
{
    public function index(){
        $notes = Note::all();
        return view('notes.index', [ 'notes' => $notes ]);
    }
    public function create(){
        return view('notes.create');
    }
    // public function show($id){
    //     $note = Note::find($id);
    //     return view('notes/show', [ 'note' => $note ]);
    // }
    public function show(Note $note){
        return view('notes.show', [ 'note' => $note ]);
    }
    // public function store(){
    //     $note = new Note;
    //     $note->title = request()->title;
    //     $note->body = request()->body;
    //     $note->important = is_null(request()->important) ? 0 : 1;
    //     $note->save();
    // }
    public function store(){
        Note::create( request()->all() );
        return redirect('/notes');
    }
    public function edit(Note $note){
        return view('notes.edit', compact('note'));
    }
    public function update(Note $note){
        $note->update(request()->all());
        // $note->title = request()->title;
        // $note->body = request()->body;
        // $note->important = request()->important;
        // $note->save();
        return redirect('/notes');
    }
}
```

resources/views/notes/_form.blade.php

```
<div class="form-group">
    <label for="">Título</label>
    <input name="title" value="{{ isset($note) ? $note->title : '' }}" type="text"
        class="form-control" id="" placeholder="Digite o título">
</div>

<div class="form-group">
    <label for="">Anotação</label>
    <textarea name="body" type="text" class="form-control" id=""
        placeholder="Digite a anotação">{{ isset($note) ? $note->body : '' }}
```

Minhas anotações Anotações Contato

Editar anotação

Tituto

Anotação

Descrição da anotação 1 - EDITADO

É importante

Editar

Minhas anotações . Todos os direitos reservados

Server: localhost:3306 » Database: laravel-Harast » Table: notes

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 0 - 2 (3 total, Query took 0.0000 seconds.)

```
SELECT * FROM 'notes'
```

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	id	title	body	important	created_at	updated_at
<input type="checkbox"/>	1	Anotação 1 - EDITADO	Descrição da anotação 1 - EDITADO	1	2019-04-25 05:15:10	2019-04-26 03:57:12
<input type="checkbox"/>	2	Anotação 2	Descrição da anotação 2	1	2019-04-25 05:23:34	2019-04-25 05:23:34
<input type="checkbox"/>	3	Anotação 3	Descrição da anotação 3	0	2019-04-25 05:30:17	2019-04-25 05:30:17

Check all With selected: Edit Copy Delete Export

Aula 15 - Validação de formulários

Validar os dados que chegam de um formulário é muito útil para "obrigar" ao usuário a prover certa informação. Laravel faz que esta tarefa seja muito simples pois traz por padrão várias regras de validação prontas para seu uso nos projetos.

app/Http/Controllers/NotesController.php

```
<?php

namespace App\Http\Controllers;

use App>Note;
use App\Http\Requests\NotesRequest;
use Illuminate\Http\Request;

class NotesController extends Controller
{
    public function index(){
        $notes = Note::all();
        return view('notes.index', [ 'notes' => $notes ]);
    }
    public function create(){
        return view('notes.create');
    }
    public function show(Note $note){
        return view('notes.show', [ 'note' => $note ]);
    }
    public function store(NotesRequest $request){
        // request()->validate([
        //     'title' => 'required',
        //     'body' => 'required|min: 20'
        // ]);
        Note::create( $request->all() );
        return redirect('/notes');
    }
    public function edit(Note $note){
        return view('notes.edit', compact('note'));
    }
    public function update(Note $note, NotesRequest $request){
        // request()->validate([
        //     'title' => 'required',
        //     'body' => 'required|min: 20'
        // ]);
        $note->update( $request->all() );
        // $note->title = request()->title;
        // $note->body = request()->body;
        // $note->important = request()->important;
        // $note->save();
        return redirect('/notes');
    }
}
```

resources/views/notes/create.blade.php

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

@section('title')
    Criar uma nova anotação
@endsection

@section('content')
    <form action="/notes" method="POST" role="form">
        {{ csrf_field() }}
        <legend>Criar nova anotação</legend>
        @include('partials.errors')
        @include('notes._form')
        <button type="submit" class="btn btn-primary">Criar</button>
    </form>
@endsection
```

resources/views/notes/edit.blade.php

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

@section('title')
    Editar: {{ $note->title }}
@endsection

@section('content')
    <form action="/notes/{{ $note->id }}" method="POST" role="form">
        {{ method_field('PATCH') }}
        {{ csrf_field() }}
        <legend>Editar anotação</legend>
        @include('partials.errors')
        @include('notes._form')
        <button type="submit" class="btn btn-primary">Editar</button>
    </form>
@endsection
```

resources/views/partials/errors.blade.php

```
@if($errors->any())
    <div class="alert alert-danger">
        <ul>
            @foreach($errors->all() as $error)
                <li>{{ $error }}</li>
            @endforeach
        </ul>
    </div>
@endif
```

Criando um request

```
php artisan make:request NotesRequest
```

```
C:\laragon\www\laravel-larast
λ php artisan make:request NotesRequest
Request created successfully.
```

app/Http/Requests/NotesRequests.php

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class NotesRequest extends FormRequest
{
    public function authorize()
    {
        return true;
    }

    public function rules()
    {
        return [
            'title' => 'required',
            'body' => 'required|min: 20'
        ];
    }
}
```

Editar: Anotação 1 - EDITADO

Minhas anotações Anotações Contato

Editar anotação

The body field is required.

Titulo

Anotação 1 - EDITADO

Anotação

Descrição da anotação 1 - EDITADO

É importante

Editar

Minhas anotações . Todos os direitos reservados

Criar uma nova anotação

Minhas anotações Anotações Contato

Criar nova anotação

The title field is required.
The body field is required.

Titulo

Digite o título

Anotação

Digite a anotação

É importante

Criar

Minhas anotações . Todos os direitos reservados

Aula 16 - Conservar dados do formulários

Nesta lição solucionaremos o problema dos formulários que não mantém a informação ao ser redirecionados de volta quando ocorrem erros de validação. No processo aprenderemos como criar nossos próprios helpers personalizados.

resources/views/notes/_form.blade.php

```
<div class="form-group">
    <label for="">Título</label>
    <input name="title" value="{{ old('title', isset($note) ? $note->title : '') }}" type="text" class="form-control" id="" placeholder="Digite o título">
</div>

<div class="form-group">
    <label for="">Anotação</label>
    <textarea name="body" type="text" class="form-control" id="" placeholder="Digite a anotação">{{ old('body', isset($note) ? $note->body : '') }}</textarea>
</div>

<div class="checkbox">
    <label>
        <input type="hidden" name="important" value="0">
        <input type="checkbox" value="1" name="important" {{ oldCheckbox('important', ( isset($note) ? $note : null) ) }}> É importante
    </label>
</div>
```

helpers/html.php

```
<?php

function oldCheckbox($checkName, $model = null){
    if(!is_null(old('important'))){
        return old('important') == '1' ? 'checked' : '';
    } else {
        if(isset($note)) {
            return $note->isImportant() ? 'checked' : '';
        }
    }
}
```

composer.json

```
.
.
.

"require-dev": {
    "filp/whoops": "^2.0",
    "fzaninotto/faker": "^1.4",
    "mockery/mockery": "^1.0",
    "nunomaduro/collision": "^2.0",
    "phpunit/phpunit": "^7.0"
},
"autoload": {
    "classmap": [
        "database/seeds",
        "database/factories"
    ],
    "psr-4": {
        "App\\": "app/"
    },
    "files": [
        "helpers/html.php"
    ]
},
```

composer dump-autoload

```
C:\laragon\www\laravel-larast
λ composer dump-autoload
Generating optimized autoload files> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover
Discovered Package: fideloper/proxy
Discovered Package: laravel/tinker
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
Generated optimized autoload files containing 3378 classes

C:\laragon\www\laravel-larast
λ |
```

Editar: Anotação 1 - EDITADO

Minhas anotações Anotações Contato

Editar anotação

Titulo

Anotação

É importante

Editar

Minhas anotações . Todos os direitos reservados

Editar: Anotação 1 - EDITADO

Minhas anotações Anotações Contato

Editar anotação

• The body field is required.

Titulo

Anotação

É importante

Editar

Minhas anotações . Todos os direitos reservados

Aula 17 - Mensagens de validação em português

Laravel traz as mensagens das regras de validação por padrão em inglês. Nesta aula veremos um par de maneiras de mostrar estas mensagens em português.

app/Http/Requests/NotesRequests.php

```
<?php

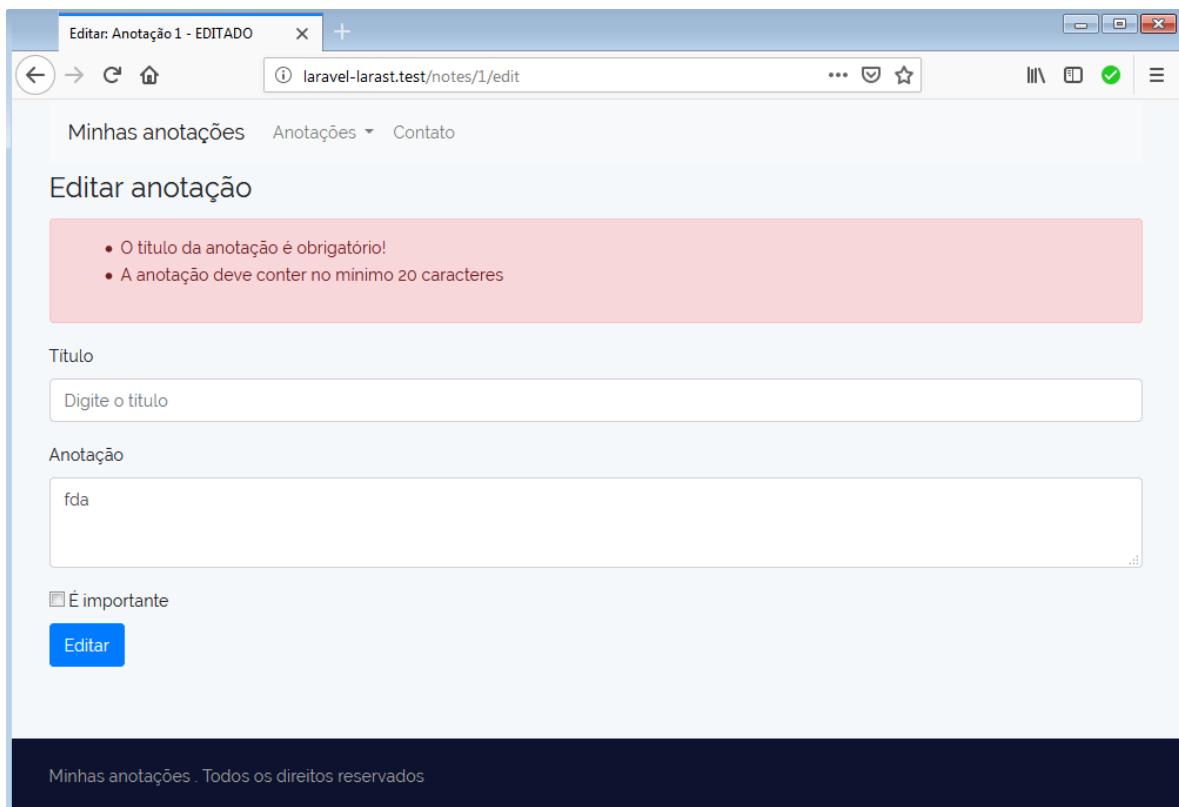
namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class NotesRequest extends FormRequest
{
    public function authorize()
    {
        return true;
    }

    public function rules()
    {
        return [
            'title' => 'required',
            'body' => 'required|min: 20'
        ];
    }

    public function messages()
    {
        return [
            'title.required' => 'O título da anotação é obrigatório!',
            'body.required' => 'O conteúdo da anotação é obrigatório!',
            'body.min' => 'A anotação deve conter no mínimo :min caracteres'
        ];
    }
}
```



config/app.php

```
.  
. .  
'locale' => 'pt-br',  
.
```

resources/lang/pt-br/validation.php

```
<?php  
return [  
    /*  
     |-----  
     | Validation Language Lines  
     |-----  
     |  
     | The following language lines contain the default error messages used by  
     | the validator class. Some of these rules have multiple versions such  
     | as the size rules. Feel free to tweak each of these messages here.  
     |  
*/
```

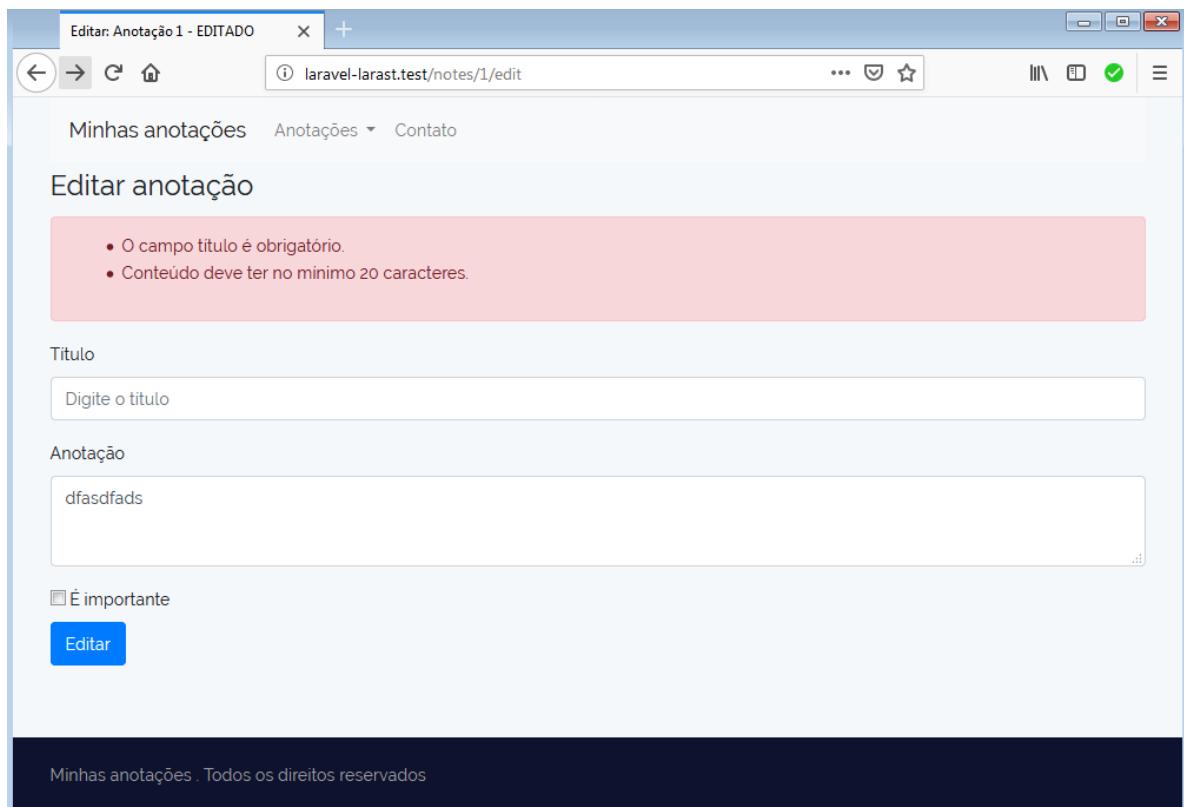
```

'accepted'          => ':Attribute deve ser aceito.',
'active_url'       => ':Attribute não é uma URL válida.',
'after'            => ':Attribute deve ser uma data depois de :date.',
'after_or_equal'   => ':Attribute deve ser uma data posterior ou igual à :date.',
'alpha'             => ':Attribute deve conter somente letras.',
'alpha_dash'        => ':Attribute deve conter letras, números e traços.',
'alpha_num'         => ':Attribute deve conter somente letras e números.',
'array'             => ':Attribute deve ser um array.',
'before'            => ':Attribute deve ser uma data antes de :date.',
'before_or_equal'  => ':Attribute deve ser uma data anterior ou igual à :date.',
'between'           => [
    'numeric' => ':Attribute deve estar entre :min e :max.',
    'file'    => ':Attribute deve estar entre :min e :max kilobytes.',
    'string'  => ':Attribute deve estar entre :min e :max caracteres.',
    'array'   => ':Attribute deve ter entre :min e :max itens.',
],
'boolean'           => ':Attribute deve ser verdadeiro ou falso.',
'confirmed'        => 'A confirmação de :attribute não confere.',
'date'              => ':Attribute não é uma data válida.',
'date_format'      => ':Attribute não confere com o formato :format.',
'different'        => ':Attribute e :other devem ser diferentes.',
'digits'            => ':Attribute deve ter :digits dígitos.',
'digits_between'   => ':Attribute deve ter entre :min e :max dígitos.',
'dimensions'        => ':Attribute tem dimensões de imagem inválidas.',
'distinct'          => ':Attribute tem um valor duplicado.',
'email'             => ':Attribute deve ser um endereço de e-mail válido.',
'exists'            => 'O :attribute selecionado é inválido.',
'file'              => ':Attribute deve ser um arquivo.',
'filled'            => ':Attribute é um campo obrigatório.',
'image'            => ':Attribute deve ser uma imagem.',
'in'                => ':Attribute é inválido.',
'in_array'          => ':Attribute não existe em :other.',
'integer'           => ':Attribute deve ser um inteiro.',
'ip'                => ':Attribute deve ser um endereço IP válido.',
'ipv4'              => ':Attribute deve ser um endereço IPv4 válido.',
'ipv6'              => ':Attribute deve ser um endereço IPv6 válido.',
'json'              => ':Attribute deve ser um JSON válido.',
'max'               => [
    'numeric' => ':Attribute não deve ser maior que :max.',
    'file'    => ':Attribute não deve ter mais que :max kilobytes.',
    'string'  => ':Attribute não deve ter mais que :max caracteres.',
    'array'   => ':Attribute não deve ter mais que :max itens.',
],
'mimes'             => ':Attribute deve ser um arquivo do tipo: :values.',
'mimetypes'         => ':Attribute deve ser um arquivo do tipo: :values.',
'min'               => [
    'numeric' => ':Attribute deve ser no mínimo :min.',
    'file'    => ':Attribute deve ter no mínimo :min kilobytes.',
    'string'  => ':Attribute deve ter no mínimo :min caracteres.',
    'array'   => ':Attribute deve ter no mínimo :min itens.',
],
'not_in'            => 'O :attribute selecionado é inválido.',
'numeric'           => ':Attribute deve ser um número.',
'present'          => 'O :attribute deve estar presente.',
'regex'             => 'O formato de :attribute é inválido.',
'required'          => 'O campo :attribute é obrigatório.',
'required_if'       => 'O campo :attribute é obrigatório quando :other é :value.';
```

```

'required_unless'      => 'O :attribute é necessário a menos que :other esteja em
:values.',
'required_with'        => 'O campo :attribute é obrigatório quando :values está
presente.',
'required_with_all'   => 'O campo :attribute é obrigatório quando :values estão
presentes.',
'required_without'    => 'O campo :attribute é obrigatório quando :values não está
presente.',
'required_without_all'=> 'O campo :attribute é obrigatório quando nenhum destes
estão presentes: :values.',
'same'                 => ':Attribute e :other devem ser iguais.',
'size'                 => [
  'numeric' => ':Attribute deve ser :size.',
  'file'     => ':Attribute deve ter :size kilobytes.',
  'string'   => ':Attribute deve ter :size caracteres.',
  'array'    => ':Attribute deve conter :size itens.',
],
'string'               => ':Attribute deve ser uma string',
'timezone'             => ':Attribute deve ser uma timezone válida.',
'unique'               => ':Attribute já está em uso.',
'uploaded'              => 'O :attribute falhou ao ser enviado.',
'url'                  => 'O formato de :attribute é inválido.',
/*
|-----|
| Custom Validation Language Lines
|-----|
|
| Here you may specify custom validation messages for attributes using the
| convention "attribute.rule" to name the lines. This makes it quick to
| specify a specific custom language line for a given attribute rule.
|
*/
'custom' => [
  'attribute-name' => [
    'rule-name' => 'custom-message',
  ],
],
/*
|-----|
| Custom Validation Attributes
|-----|
|
| The following language lines are used to swap attribute place-holders
| with something more reader friendly such as E-Mail Address instead
| of "email". This simply helps us make messages a little cleaner.
|
*/
'attributes' => [
  'title' => 'título',
  'body'  => 'conteúdo'
],
];

```



Aula 18 - Eliminação de registros

Nesta aula veremos como é simples eliminar registros da base de dados. Ademais, agregaremos a funcionalidade de pedir uma confirmação de eliminação ao usuário antes de executar a ação.

routes/web.php

```
<?php

Route::get('/notes', 'NotesController@index');
Route::get('/notes/create', 'NotesController@create');
// Route::get('/notes/{id}', 'NotesController@show');
Route::get('/notes/{note}', 'NotesController@show');
Route::get('/notes/{note}/edit', 'NotesController@edit');
Route::patch('/notes/{note}', 'NotesController@update');
Route::post('/notes', 'NotesController@store');
Route::delete('/notes/{note}', 'NotesController@destroy');

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

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

app/Http/Requests/NotesRequests.php

```
<?php

namespace App\Http\Controllers;

use App>Note;
use App\Http\Requests\NotesRequest;
use Illuminate\Http\Request;

class NotesController extends Controller
{
    public function index(){
        $notes = Note::all();
        return view('notes.index', [ 'notes' => $notes ]);
    }

    public function create(){
        return view('notes.create');
    }

    public function show(Note $note){
        return view('notes.show', [ 'note' => $note ]);
    }

    public function store(NotesRequest $request){
        Note::create( $request->all() );
        return redirect('/notes');
    }

    public function edit(Note $note){
        return view('notes.edit', compact('note'));
    }

    public function update(Note $note, NotesRequest $request){
        $note->update( $request->all() );
        return redirect('/notes');
    }

    public function destroy(Note $note){
        $note->delete();
        return redirect('/notes');
    }
}
```

resources/views/layouts/app.blade.php

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>@yield('title')</title>
    <link rel="stylesheet" href="/css/app.css">
</head>
<body>

    <div class="container">
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <a class="navbar-brand" href="/">{{ config('app.name') }}</a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav">
                    <li class="nav-item dropdown {{ Request::is('notes') ? 'active' : '' }}>
                        <a class="nav-link dropdown-toggle" href="#" id="navbardrop" data-toggle="dropdown">
                            Anotações
                        </a>
                        <div class="dropdown-menu">
                            <a class="dropdown-item" href="/notes">Ver</a>
                            <a class="dropdown-item" href="/notes/create">Criar</a>
                        </div>
                    </li>
                    <li class="{{ Request::is('contact') ? 'active' : '' }}>
                        <a class="nav-link" href="/contact">Contato</a>
                    </li>
                </ul>
            </div>
        </nav>
    </div>

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

    <footer>
        <div class="container">
            {{ config('app.name') }} . Todos os direitos reservados
        </div>
    </footer>
```

```

<script src="/js/app.js"></script>

@yield('scripts')

</body>
</html>

```

resources/views/notes/index.blade.php

```

@extends('layouts.app')

@section('title')
    Listagem de anotações
@endsection

@section('content')
    <table class="table table-striped table-hover">
        <thead>
            <th>Anotação</th>
            <th>Editar</th>
            <th>Eliminar</th>
        </thead>
        <tbody>
            @foreach($notes as $note)
                <tr>
                    <td>
                        <a href="notes/{{ $note->id }}">
                            {{ $note->title }}
                            @if($note->isImportant())
                                *
                            @endif
                        </a>
                    </td>
                    <td>
                        <a href="/notes/{{ $note->id }}/edit" class="btn btn-primary btn-sm">Editar</a>
                    </td>
                    <td>
                        <form action="/notes/{{ $note->id }}" method="POST">
                            {{ method_field('DELETE') }}
                            {{ csrf_field() }}
                            <button type="button" class="btn btn-danger btn-sm btn-delete">Eliminar</button>
                        </form>
                    </td>
                </tr>
            @endforeach
        </tbody>
    </table>
@endsection

```

```
@section('scripts')
<script>
    $('.btn-delete').on('click', function(){
        if(confirm('Tem certeza que deseja eliminar a anotação?')){
            $(this).parents('form:first').submit();
        }
    })
</script>
@endsection
```

Screenshot of a web browser showing a list of notes. The title bar says "Listagem de anotações". The address bar shows "laravel-larast.test/notes". The page content includes a header with "Minhas anotações", "Anotações", and "Contato". Below is a table with three rows:

Anotação	Editar	Eliminar
Anotação 1 - EDITADO	Editar	Eliminar
Anotação 2 *	Editar	Eliminar
Anotação 3	Editar	Eliminar

At the bottom, a dark footer bar contains the text "Minhas anotações . Todos os direitos reservados".

Eliminando a anotação 3:

Screenshot of the same web browser after clicking the "Eliminar" button for note 3. A confirmation dialog box is displayed in the center of the screen with the message "Tem certeza que deseja eliminar a anotação?". It has two buttons: "OK" and "Cancelar". The background of the browser window is dimmed.

Listagem de anotações

Minhas anotações Anotações Contato

Anotação	Editar	Eliminar
Anotação 1 - EDITADO	Editar	Eliminar
Anotação 2 *	Editar	Eliminar

Minhas anotações . Todos os direitos reservados

The screenshot shows a web browser window titled "Listagem de anotações". The address bar displays the URL "laravel-larast.test/notes". The main content area is titled "Minhas anotações" and includes a dropdown menu for "Anotações" and a link for "Contato". Below this, there is a table listing two notes. The first note is "Anotação 1 - EDITADO" and the second is "Anotação 2 *". Each note row contains two buttons: "Editar" (in blue) and "Eliminar" (in red). At the bottom of the page, there is a dark footer bar with the text "Minhas anotações . Todos os direitos reservados".

Aula 19 - Relação entre tabelas - Parte 1

Nesta primeira parte aprenderemos como criar campos com chave estrangeira a nível de base de dados para estabelecer relações. Ademais, deixaremos nossa interface pronta para implementar uma nova funcionalidade na próxima aula que nos permitirá aprender sobre relações entre modelos do Eloquent.

resources/views/notes/_form.blade.php

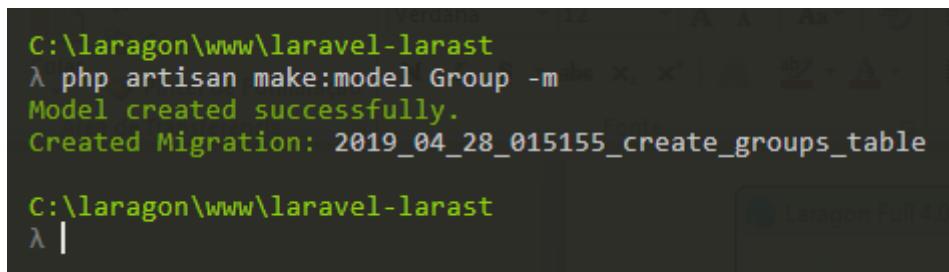
```
<div class="form-group">
    <label for="">Título</label>
    <input name="title" value="{{ old('title', isset($note) ? $note->title : '')}" type="text" class="form-control" id="" placeholder="Digite o título">
</div>

<div class="form-group">
    <label for="">Anotação</label>
    <textarea name="body" type="text" class="form-control" id="" placeholder="Digite a anotação">{{ old('body', isset($note) ? $note->body : '')}}</textarea>
</div>

<div class="form-group">
    <label>Grupo</label>
    <select class="form-control" name="group_id">
        <option value="">--Nenhum--</option>
        @foreach($groups as $group)
            <option value="{{ $group->id }}>{{ $group->name }}</option>
        @endforeach
    </select>
</div>

<div class="checkbox">
    <label>
        <input type="hidden" name="important" value="0">
        <input type="checkbox" value="1" name="important" {{ oldCheckbox('important', ( isset($note) ? $note : null) ) }}> É importante
    </label>
</div>
```

```
php artisan make:model Group -m
```



```
C:\laragon\www\laravel-larast
λ php artisan make:model Group -m
Model created successfully.
Created Migration: 2019_04_28_015155_create_groups_table

C:\laragon\www\laravel-larast
λ | Laragon Full 4.0
```

database/migrations/2019_04_28_015155_create_groups_table.php

```
<?php

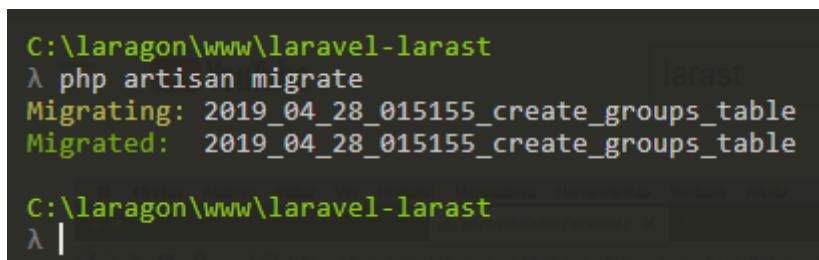
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateGroupsTable extends Migration
{
    public function up()
    {
        Schema::create('groups', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists('groups');
    }
}
```

Criando a tabela

```
php artisan migrate
```



```
C:\laragon\www\laravel-larast
λ php artisan migrate
Migrating: 2019_04_28_015155_create_groups_table
Migrated: 2019_04_28_015155_create_groups_table

C:\laragon\www\laravel-larast
λ | Laragon Full 4.0
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<code>id</code>	int(10)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	<code>name</code>	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
3	<code>created_at</code>	timestamp			Yes	NULL			Change Drop More
4	<code>updated_at</code>	timestamp			Yes	NULL			Change Drop More

↑ Check all With selected: Browse Change Drop Primary Unique Index Fulltext

Criando migration para adicionar campo na tabela notes

```
php artisan make:migration add_group_id_to_note_tables --table=notes
```

```
C:\laragon\www\laravel-larast
λ php artisan make:migration add_group_id_to_note_tables --table=notes
Created Migration: 2019_04_28_020913_add_group_id_to_note_tables

C:\laragon\www\laravel-larast
λ |
```

database/migrations/2019_04_28_020913_add_group_id_to_note_tables.php

```
<?php

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

class AddGroupIdToNoteTables extends Migration
{
    public function up()
    {
        Schema::table('notes', function (Blueprint $table) {
            $table->unsignedInteger('group_id')->nullable()->after('id');

            $table->foreign('group_id')->references('id')->on('groups')-
>onDelete('cascade');
        });
    }

    public function down()
    {
        Schema::table('notes', function (Blueprint $table) {
            //
```

```
}
```

Criando o campo na tabela notes

php artisan migrate

```
C:\laragon\www\laravel-larast
λ php artisan migrate
Migrating: 2019_04_28_020913_add_group_id_to_note_tables
Migrated: 2019_04_28_020913_add_group_id_to_note_tables

C:\laragon\www\laravel-larast
λ |
```

Showing rows 0 - 1 (2 total, Query took 0.0020 seconds.)

[SELECT * FROM `notes`](#)

[Profiling](#) [Edit inline](#) [Edit](#) [Explain SQL](#) [Create PHP](#)

Show all Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	id	group_id	title	body	important	created_at	updated_at
<input type="checkbox"/>	1	NULL	Anotação 1 - EDITADO	Descrição da anotação 1 - EDITADO	0	2019-04-25 05:15:10	2019-04-26 16:52:10
<input type="checkbox"/>	2	NULL	Anotação 2	Descrição da anotação 2	1	2019-04-25 05:23:34	2019-04-25 05:23:34

[Check all](#) With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

← Server: localhost:3306 » Database: laravel-larast » Table: notes

Browse Structure SQL Search Insert Export Import Privileges Operations

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 group_id	int(10)		UNSIGNED	Yes	NULL			Change Drop More
<input type="checkbox"/>	3 title	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	4 body	text	utf8mb4_unicode_ci		No	None			Change Drop More
<input type="checkbox"/>	5 important	tinyint(1)			No	0			Change Drop More
<input type="checkbox"/>	6 created_at	timestamp			Yes	NULL			Change Drop More
<input type="checkbox"/>	7 updated_at	timestamp			Yes	NULL			Change Drop More

[Check all](#) With selected: [Browse](#) [Change](#) [Drop](#) [Primary](#) [Unique](#) [Index](#) [Fulltext](#)

[Print](#) [Propose table structure](#) [Move columns](#) [Normalize](#)

Add 1 column(s) after updated_at [Go](#)

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	2	A	No	
Edit Drop	notes_group_id_foreign	BTREE	No	No	group_id	1	A	Yes	

Na tabela groups, insira manualmente dois grupos: Família e Trabalho.

The screenshot shows a MySQL query results page. At the top, a green bar displays the message: "Showing rows 0 - 1 (2 total, Query took 0.0020 seconds.)". Below this, a SQL query is shown: "SELECT * FROM `groups`". The main area displays the data from the 'groups' table:

	id	name	created_at	updated_at
<input type="checkbox"/>	1	Familia	2019-04-27 23:26:06	2019-04-27 23:26:06
<input type="checkbox"/>	2	Trabalho	2019-04-27 23:26:06	2019-04-27 23:26:06

Below the table are various navigation and search controls.

app/Http/Requests/NotesRequests.php

```
<?php

namespace App\Http\Controllers;

use App\Note;
use App\Group;
use App\Http\Requests\NotesRequest;
use Illuminate\Http\Request;

class NotesController extends Controller
{
    public function index(){
        $groups = Group::all();
        $notes = Note::all();
        return view('notes.index', compact('notes', 'groups'));
    }

    public function create(){
        $groups = Group::all();
        return view('notes.create', compact('groups'));
    }

    public function show(Note $note){
        return view('notes.show', [ 'note' => $note ]);
    }

    public function store(NotesRequest $request){
        Note::create( $request->all() );
        return redirect('/notes');
    }
}
```

```

public function edit(Note $note){
    $groups = Group::all();
    return view('notes.edit', compact('note', 'groups'));
}

public function update(Note $note, NotesRequest $request){
    $note->update( $request->all() );
    return redirect('/notes');
}

public function destroy(Note $note){
    $note->delete();
    return redirect('/notes');
}

}

```

app/Notes.php

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Note extends Model
{
    // protected $table = 'anotacoes';

    protected $fillable = [
        'title', 'body', 'important', 'group_id'
    ];

    public function isImportant()
    {
        return $this->important == 1;
    }
}

```

resources/views/notes/index.blade.php

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

@section('title')
    Listagem de anotações
@endsection

@section('content')
    <div class="row">
        <div class="col-md-2">
            <h4>Grupos</h4>
            <ul class="list-group">
                <li class="list-group-item"><a href="">--Nenhum--</a></li>
                @foreach($groups as $group)
                    <li class="list-group-item"><a href="">{{ $group->name }}</a></li>
                @endforeach
            </ul>
        </div>
        <div class="col-md-10">
            <table class="table table-striped table-hover">
                <thead>
                    <th>Anotação</th>
                    <th>Editar</th>
                    <th>Eliminar</th>
                </thead>
                <tbody>
                    @foreach($notes as $note)
                        <tr>
                            <td>
                                <a href="/notes/{{ $note->id }}">
                                    {{ $note->title }}
                                    @if($note->isImportant())
                                        *
                                    @endif
                                </a>
                            </td>
                            <td>
                                <a href="/notes/{{ $note->id }}/edit" class="btn btn-primary
btn-sm">Editar</a>
                            </td>
                            <td>
                                <form action="/notes/{{ $note->id }}" method="POST">
                                    {{ method_field('DELETE') }}
                                    {{ csrf_field() }}
                                    <button type="button" class="btn btn-danger btn-sm btn-
delete">Eliminar</button>
                                </form>
                            </td>
                        </tr>
                    @endforeach
                </tbody>
            </table>
        </div>
    </div>
@endsection
```

```

@section('scripts')
<script>
  $('.btn-delete').on('click', function(){
    if(confirm('Tem certeza que deseja eliminar a anotação?')){
      $(this).parents('form:first').submit();
    }
  })
</script>
@endsection

```

Criar nova anotação

Título

Digite o título

Anotação

Digite a anotação

Grupo

--Nenhum--

É importante

Criar

Minhas anotações. Todos os direitos reservados

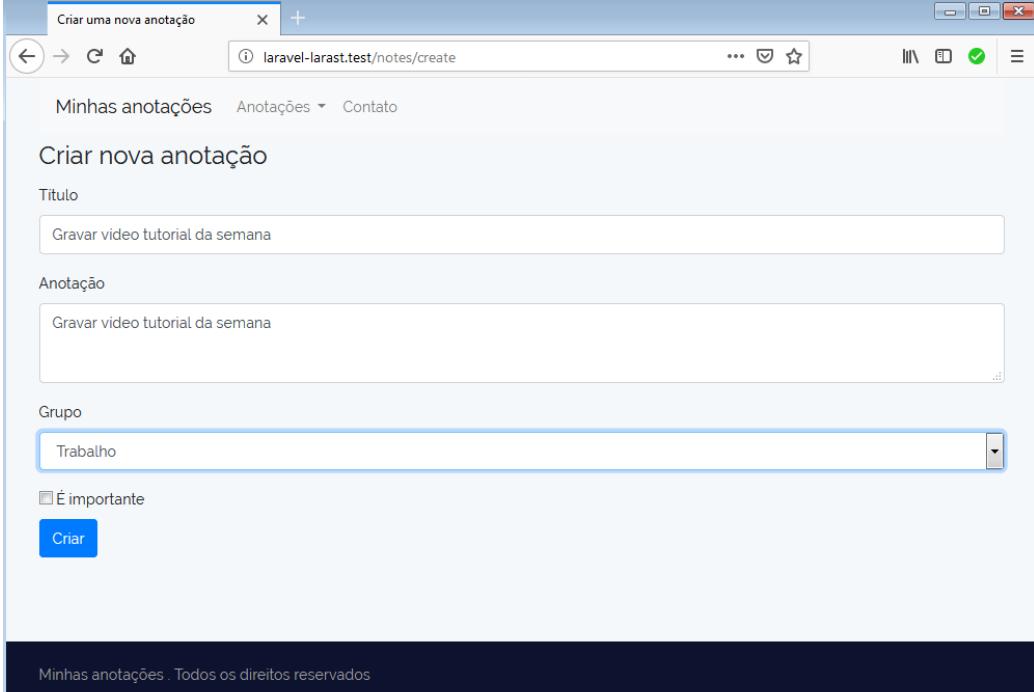
Anotação	Editar	Eliminar
Anotação 1 - EDITADO	Editar	Eliminar
Anotação 2 *	Editar	Eliminar

Minhas anotações. Todos os direitos reservados

Aula 20 - Relação entre tabelas - Parte 2

Nesta segunda parte veremos como definir a nível de Eloquent, as relações entre dois modelos. Ademais pularemos certos problemas que se apresentam ao usar estas relações nas vistas para mostrar informação.

Inicialmente realize um `truncate` na tabela `notes`. Em seguida insira as seguintes anotações:



Criar uma nova anotação

Minhas anotações Anotações Contato

Criar nova anotação

Título

Gravar video tutorial da semana

Anotação

Gravar video tutorial da semana

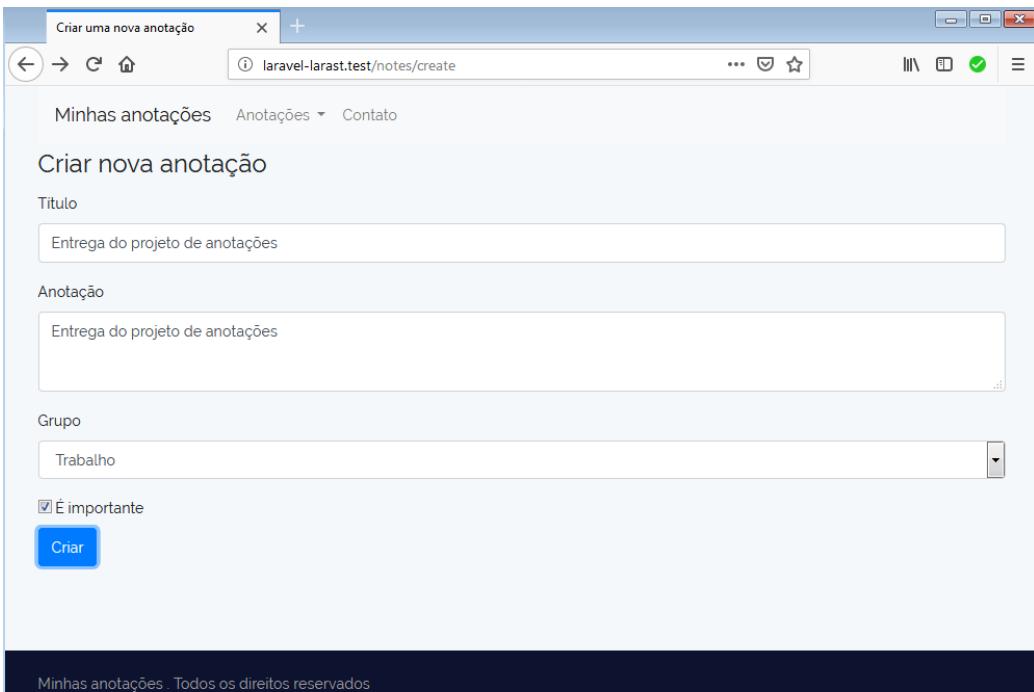
Grupo

Trabalho

É importante

Criar

Minhas anotações. Todos os direitos reservados



Criar uma nova anotação

Minhas anotações Anotações Contato

Criar nova anotação

Título

Entrega do projeto de anotações

Anotação

Entrega do projeto de anotações

Grupo

Trabalho

É importante

Criar

Minhas anotações. Todos os direitos reservados

Criar uma nova anotação

Minhas anotações Anotações Contato

Criar nova anotação

Título

Anotação

Grupo

É importante

Criar

Minhas anotações . Todos os direitos reservados

Criar uma nova anotação

Minhas anotações Anotações Contato

Criar nova anotação

Título

Anotação

Grupo

É importante

Criar

Minhas anotações . Todos os direitos reservados

Criar uma nova anotação +

Minhas anotações Anotações Contato

Criar nova anotação

Titulo

Anotação

Ir ao supermercado para fazer compras.

Grupo

--Nenhum--

É importante

Criar

Minhas anotações . Todos os direitos reservados

Listagem de anotações +

Minhas anotações Anotações Contato

Grupos	Anotação	Editar	Eliminar
--Nenhum--	Gravar video tutorial da semana	Editar	Eliminar
Família	Entrega do projeto de anotações *	Editar	Eliminar
Trabalho	Levar Boby ao veterinário *	Editar	Eliminar
	Graduação de Susan *	Editar	Eliminar
	Fazer compras	Editar	Eliminar

Minhas anotações . Todos os direitos reservados

Showing rows 0 - 4 (5 total, Query took 0.0000 seconds.)

```
SELECT * FROM `notes`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	id	group_id	title	body	important	created_at	updated_at
	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	1	2	Gravar vídeo tutorial da semana	Gravar vídeo tutorial da semana	0	2019-04-28 03:47:48	2019-04-28 03:47:48
	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	2	2	Entrega do projeto de anotações	Entrega do projeto de anotações	1	2019-04-28 03:49:08	2019-04-28 03:49:08
	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	3	1	Levar Boby ao veterinário	Levar Boby ao veterinário	1	2019-04-28 03:53:56	2019-04-28 03:53:56
	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	4	1	Graduação de Susan	Comparecer a graduação de Susan	1	2019-04-28 03:58:13	2019-04-28 03:58:13
	<input type="checkbox"/>	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	5	NULL	Fazer compras	Ir ao supermercado para fazer compras.	0	2019-04-28 04:00:25	2019-04-28 04:00:25

Check all | With selected: Edit | Copy | Delete | Export

Definindo relação entre a tabela grupo e a tabela notes

app/Group.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Group extends Model
{
    public function notes(){
        return $this->hasMany(Note::class); // group_id
    }
}
```

Usando o tinker para testes

php artisan tinker

```
C:\laragon\www\laravel-larast
λ php artisan tinker
Psy Shell v0.9.6 (PHP 7.2.11 - cli) by Justin Hileman
>>> $g = App\Group::find(1)
=> App\Group {#2882
    id: 1,
    name: "Família",
    created_at: "2019-04-27 23:26:06",
    updated_at: "2019-04-27 23:26:06",
}
>>> $g->notes
=> Illuminate\Database\Eloquent\Collection {#2885
    all: [
        App\Note {#2883
            id: 3,
            group_id: 1,
            title: "Levar Boby ao veterinário",
            body: "Levar Boby ao veterinário",
            important: 1,
            created_at: "2019-04-28 03:53:56",
            updated_at: "2019-04-28 03:53:56",
        },
        App\Note {#2884
            id: 4,
            group_id: 1,
            title: "Graduação de Susan",
            body: "Comparecer a graduação de Susan",
            important: 1,
            created_at: "2019-04-28 03:58:13",
            updated_at: "2019-04-28 03:58:13",
        },
    ],
}
>>> | ⟲ ⟳ ⟴ ⟵ ⟶ 3:26 / 13:24
```

routes/web.php

```
<?php

Route::get('/notes', 'NotesWithoutGroupController@index');
Route::get('/groups/{group}/notes', 'NotesController@index');

Route::get('/notes/create', 'NotesController@create');
// Route::get('/notes/{id}', 'NotesController@show');
Route::get('/notes/{note}', 'NotesController@show');
Route::get('/notes/{note}/edit', 'NotesController@edit');
Route::patch('/notes/{note}', 'NotesController@update');
Route::post('/notes', 'NotesController@store');
Route::delete('/notes/{note}', 'NotesController@destroy');

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

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

Definindo um controlador para anotações sem grupo

```
php artisan make:controller NotesWithoutGroupController
```

```
C:\laragon\www\laravel-larast
λ php artisan make:controller NotesWithoutGroupController
Controller created successfully.

C:\laragon\www\laravel-larast
λ |
```

app/Http/Controllers/NotesWithoutGroupRequests.php

```
<?php

namespace App\Http\Controllers;

use App>Note;
use App\Group;
use Illuminate\Http\Request;

class NotesWithoutGroupController extends Controller
{
    public function index(Group $group){
        $groups = Group::all();
        $notes = Note::all();
        return view('notes.index', compact('notes', 'groups'));
    }
}
```

resources/views/notes/index.blade.php

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

@section('title')
    Listagem de anotações
@endsection

@section('content')
    <div class="row">
        <div class="col-md-2">
            <h4>Grupos</h4>
            <ul class="list-group">
                <li class="list-group-item"><a href="/notes">--Nenhum--</a></li>
                @foreach($groups as $group)
                    <li class="list-group-item"><a href="/groups/{{ $group->id }}/notes">{{ $group->name }}</a></li>
```

```

        @endforeach
    </ul>
</div>
<div class="col-md-10">
    <table class="table table-striped table-hover">
        <thead>
            <th>Anotação</th>
            <th>Grupo</th>
            <th>Editar</th>
            <th>Eliminar</th>
        </thead>
        <tbody>
            @foreach($notes as $note)
                <tr>
                    <td>
                        <a href="/notes/{{ $note->id }}">
                            {{ $note->title }}
                            @if($note->isImportant())
                                *
                            @endif
                        </a>
                    </td>
                    <td>
                        @if($note->group_id != null)
                            <label class="btn btn-info btn-sm">{{ $note-
                        >group->name }}</label>
                        @endif
                    </td>
                    <td>
                        <a href="/notes/{{ $note->id }}/edit" class="btn btn-
primary btn-sm">Editar</a>
                    </td>
                    <td>
                        <form action="/notes/{{ $note->id }}" method="POST">
                            {{ method_field('DELETE') }}
                            {{ csrf_field() }}
                            <button type="button" class="btn btn-danger btn-
sm btn-delete">Eliminar</button>
                        </form>
                    </td>
                </tr>
            @endforeach
        </tbody>
    </table>
</div>
</div>
@endsection

```

```

@section('scripts')
<script>
  $('.btn-delete').on('click', function(){
    if(confirm('Tem certeza que deseja eliminar a anotação?')){
      $(this).parents('form:first').submit();
    }
  })
</script>
@endsection

```

The screenshot shows a web browser window titled "Listagem de anotações". The URL is "laravel-larast.test/notes". The page displays a table of notes with columns: Anotação, Grupo, Editar, and Eliminar. On the left, there is a sidebar titled "Grupos" with options: "--Nenhum--", "Família", and "Trabalho". The notes listed are:

Anotação	Grupo	Editar	Eliminar
Gravar video tutorial da semana	Trabalho	Editar	Eliminar
Entrega do projeto de anotações *	Trabalho	Editar	Eliminar
Levar Boby ao veterinário *	Família	Editar	Eliminar
Graduação de Susan *	Família	Editar	Eliminar
Fazer compras		Editar	Eliminar

At the bottom of the page, there is a footer bar with the text "Minhas anotações . Todos os direitos reservados".

Aula 21 - Scopes

Os scopes permitem escrever código muito mais expressivo e fácil de ler pois podem encapsular a lógica que desejamos aplicar na obtenção de informação a partir da base de dados.

app/Http/Controllers/NotesWithoutGroupRequests.php

```
<?php

namespace App\Http\Controllers;

use App\Note;
use App\Group;
use Illuminate\Http\Request;

class NotesWithoutGroupController extends Controller
{
    public function index(Group $group){
        $groups = Group::all();
        // $notes = Note::where('group_id', '=', null)->get();
        // $notes = Note::whereNull('group_id')->get();
        $notes = Note::withoutGroup()->latest()->get();
        // $notes = Note::withoutGroup()->oldest()->get();
        return view('notes.index', compact('notes', 'groups'));
    }
}
```

app/Note.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Note extends Model
{
    // protected $table = 'anotacoes';

    protected $fillable = [
        'title', 'body', 'important', 'group_id'
    ];

    public function isImportant()
    {
        return $this->important == 1;
    }
}
```

```
public function group(){
    return $this->belongsTo(Group::class); // group_id
}

public function scopeWithoutGroup($query){
    return $query->whereNull('group_id');
}

// public function scopeOrdemDesc($query){
//     return $query->orderBy('created_at', 'DESC');
// }

}
```

The screenshot shows a web application window titled "Listagem de anotações". The URL in the address bar is "laravel-larast.test/notes". The interface includes a header with back, forward, search, and other navigation icons. Below the header, there are tabs for "Minhas anotações", "Anotações", and "Contato".

The main content area displays a table with the following data:

Grupos	Anotação	Grupo	Editar	Eliminar
--Nenhum--	Fazer compras		Editar	Eliminar
Família				
Trabalho				

At the bottom of the page, there is a footer bar with the text "Minhas anotações. Todos os direitos reservados".

Aula 22 - Coleções

As coleções (collections) são muito utilizadas em Laravel. Apresentam muitas vantagens ao utilizá-las como alternativa/complemento ao uso de Arrays.

php artisan tinker

```
C:\laragon\www\laravel-larast
λ php artisan tinker
Psy Shell v0.9.6 (PHP 7.2.11 - cli) by Justin Hileman
>>> $personas = ['luis', 'juan', 'andres']
=> [
    "luis",
    "juan",
    "andres",
]
>>> $personas = collect(['luis', 'juan', 'andres'])
=> Illuminate\Support\Collection {#2883
    all: [
        "luis",
        "juan",
        "andres",
    ],
}
>>> count($personas)
=> 3
>>> $personas[1]
=> "juan"
>>> $personas->count()
=> 3
>>> $personas->first()
=> "luis"
>>> $personas->last()
=> "andres"
>>> $personas->reverse()
=> Illuminate\Support\Collection {#2873
    all: [
        2 => "andres",
        1 => "juan",
        0 => "luis",
    ],
}
>>> $personas->map(function($persona){ return ucfirst($persona); })
=> Illuminate\Support\Collection {#2884
    all: [
        "Luis",
        "Juan",
        "Andres",
    ],
}
```

```
>>> App\Note::all()
=> Illuminate\Database\Eloquent\Collection {#2889
    all: [
        App\Note {#2890
            id: 1,
            group_id: 2,
            title: "Gravar vídeo tutorial da semana",
            body: "Gravar vídeo tutorial da semana",
            important: 0,
            created_at: "2019-04-28 03:47:48",
            updated_at: "2019-04-28 03:47:48",
        },
        App\Note {#2891
            id: 2,
            group_id: 2,
            title: "Entrega do projeto de anotações",
            body: "Entrega do projeto de anotações",
            important: 1,
            created_at: "2019-04-28 03:49:08",
            updated_at: "2019-04-28 03:49:08",
        },
        App\Note {#2892
            id: 3,
            group_id: 1,
            title: "Levar Boby ao veterinário",
            body: "Levar Boby ao veterinário",
            important: 1,
            created_at: "2019-04-28 03:53:56",
            updated_at: "2019-04-28 03:53:56",
        },
        App\Note {#2893
            id: 4,
            group_id: 1,
            title: "Graduação de Susan",
            body: "Comparecer a graduação de Susan",
            important: 1,
            created_at: "2019-04-28 03:58:13",
            updated_at: "2019-04-28 03:58:13",
        },
        App\Note {#2894
            id: 5,
            group_id: null,
            title: "Fazer compras",
            body: "Ir ao supermercado para fazer compras.",
            important: 0,
            created_at: "2019-04-28 04:00:25",
            updated_at: "2019-04-28 04:00:25",
        },
    ],
}
>>> |
```

```
>>> $notes = App\Note::all()
=> Illuminate\Database\Eloquent\Collection {#2869
    all: [
        App\Note {#2881
            id: 1,
            group_id: 2,
            title: "Gravar vídeo tutorial da semana",
            body: "Gravar vídeo tutorial da semana",
            important: 0,
            created_at: "2019-04-28 03:47:48",
            updated_at: "2019-04-28 03:47:48",
        },
        App\Note {#2900
            id: 2,
            group_id: 2,
            title: "Entrega do projeto de anotações",
            body: "Entrega do projeto de anotações",
            important: 1,
            created_at: "2019-04-28 03:49:08",
            updated_at: "2019-04-28 03:49:08",
        },
        App\Note {#2901
            id: 3,
            group_id: 1,
            title: "Levar Boby ao veterinário",
            body: "Levar Boby ao veterinário",
            important: 1,
            created_at: "2019-04-28 03:53:56",
            updated_at: "2019-04-28 03:53:56",
        },
        App\Note {#2902
            id: 4,
            group_id: 1,
            title: "Graduação de Susan",
            body: "Comparecer a graduação de Susan",
            important: 1,
            created_at: "2019-04-28 03:58:13",
            updated_at: "2019-04-28 03:58:13",
        },
        App\Note {#2903
            id: 5,
            group_id: null,
            title: "Fazer compras",
            body: "Ir ao supermercado para fazer compras.",
            important: 0,
            created_at: "2019-04-28 04:00:25",
            updated_at: "2019-04-28 04:00:25",
        },
    ],
}
22. C:\Users\joseph\Documents\GitHub\laravel\laravel\app\database\seeds\NotesTableSeeder.php:22
>>> $notes->first()
=> App\Note {#2881
    id: 1,
    group_id: 2,
    title: "Gravar vídeo tutorial da semana",
    body: "Gravar vídeo tutorial da semana",
    important: 0,
    created_at: "2019-04-28 03:47:48",
    updated_at: "2019-04-28 03:47:48",
}
>>> $notes->count()
=> 5
>>> |
```

```
>>> $notes->filter(function($note){ return $note->isImportant(); })
=> Illuminate\Database\Eloquent\Collection {#2878
    all: [
        1 => App\Note {#2900
            id: 2,
            group_id: 2,
            title: "Entrega do projeto de anotações",
            body: "Entrega do projeto de anotações",
            important: 1,
            created_at: "2019-04-28 03:49:08",
            updated_at: "2019-04-28 03:49:08",
        },
        2 => App\Note {#2901
            id: 3,
            group_id: 1,
            title: "Levar Boby ao veterinário",
            body: "Levar Boby ao veterinário",
            important: 1,
            created_at: "2019-04-28 03:53:56",
            updated_at: "2019-04-28 03:53:56",
        },
        3 => App\Note {#2902
            id: 4,
            group_id: 1,
            title: "Graduação de Susan",
            body: "Comparecer a graduação de Susan",
            important: 1,
            created_at: "2019-04-28 03:58:13",
            updated_at: "2019-04-28 03:58:13",
        },
    ],
}
>>> |
```

```
>>> $c = [18, 7, 8, 9]
=> [
    18,
    7,
    8,
    9,
]
>>> $c = collect([18, 7, 8, 9])
=> Illuminate\Support\Collection {#2871
    all: [
        18,
        7,
        8,
        9,
    ],
}
>>> $c->sum()
=> 42
>>> $c->sum() / $c->count()
=> 10.5
>>> |
```

Aula 23 - Formulários com Laravel Collective

Ao manejar formulários de edição nos enfrentamos com alguns problemas; por exemplo carregar automaticamente os dados do modelo que se está editando e as vezes manter as edições no formulário quando há erros de validação e somos redirecionados de volta a página do formulário. Em vez de escrever uma solução própria, vamos instalar um pacote muito útil para estes casos.

`resources/views/notes/_form.blade.php`

```
<div class="form-group">
    <label for="">Título</label>
    <input name="title" value="{{ old('title', isset($note) ? $note->title : '')}" type="text" class="form-control" id="" placeholder="Digite o título">
</div>

<div class="form-group">
    <label for="">Anotação</label>
    <textarea name="body" type="text" class="form-control" id="" placeholder="Digite a anotação">{{ old('body', isset($note) ? $note->body : '')}}</textarea>
</div>
<div class="form-group">
    <label>Grupo</label>
    <select class="form-control" name="group_id">
        <option value="">--Nenhum--</option>
        @foreach($groups as $group)
            <option value="{{ $group->id }}"
                @if(!is_null(old('group_id')))
                    {{ old('group_id') == $group->id ? 'selected' : '' }}
                @else
                    @if(isset($note))
                        {{ $note->group_id == $group->id ? "selected" : ''}}
                    @endif
                @endif
            >{{ $group->name }}</option>
        @endforeach
    </select>
</div>
<div class="checkbox">
    <label>
        <input type="hidden" name="important" value="0">
        <input type="checkbox" value="1" name="important"
            {{ oldCheckbox('important', ( isset($note) ? $note : null) ) }}>
        É importante
    </label>
</div>
```

Listagem de anotações

Minhas anotações Anotações Contato

Grupos	Anotação	Grupo	Editar	Eliminar
--Nenhum--	Levar Boby ao veterinário *	Família	Editar	Eliminar
Família	Graduação de Susan *	Família	Editar	Eliminar
Trabalho				

Minhas anotações . Todos os direitos reservados

Editar: Graduação de Susan

Minhas anotações Anotações Contato

Editar anotação

Titulo

Anotação

Grupo

É importante

[Editar](#)

Minhas anotações . Todos os direitos reservados

Instalando o Laravel Collective

composer require "laravelcollective/html":"^5.5.0"

```
C:\laragon\www\laravel-larast
λ composer require "laravelcollective/html":"^5.5.0"
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
  - Installing laravelcollective/html (v5.6.10): Loading from cache
Writing lock file
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover
Discovered Package: fideloper/proxy
Discovered Package: laravel/tinker
Discovered Package: laravelcollective/html
Discovered Package: nunomaduro/collision
Package manifest generated successfully.

C:\laragon\www\laravel-larast
λ |
```

resources/views/notes/_form.blade.php

```
<div class="form-group">
    <label for="">Título</label>
    {{-- <input name="title" value="{{ old('title', isset($note) ? $note->title : '') }}" type="text" class="form-control" id="" placeholder="Digite o título"> --}}
    {!! Form::text('title', null, ['class' => 'form-control', 'placeholder' => 'Digite o título']) !!}
</div>

<div class="form-group">
    <label for="">Anotação</label>
    {{-- <textarea name="body" type="text" class="form-control" id="" placeholder="Digite a anotação">{{ old('body', isset($note) ? $note->body : '') }} --}}
    {!! Form::textarea('body', null, ['class' => 'form-control', 'placeholder' => 'Digite a anotação', 'rows' => 6]) !!}
</div>

<div class="form-group">
    <label>Grupo</label>
    {{-- <select class="form-control" name="group_id">
        <option value="">--Nenhum--</option>
        @foreach($groups as $group)
            <option value="{{ $group->id }}"
                @if(!is_null(old('group_id')))
                    {{ old('group_id') == $group->id ? 'selected' : '' }}
                @else
                    @if(isset($note))
                        {{ $note->group_id == $group->id ? "selected" : ''}}
                    @endif
                @endif
            >{{ $group->name }}</option>
        @endforeach
        </select> --}}
    {!! Form::select('group_id', $groups, null, ['class' => 'form-control', 'placeholder' => '--Nenhum--']) !!}
</div>

<div class="checkbox">
    <label>
        <input type="hidden" name="important" value="0">
        {{-- <input type="checkbox" value="1" name="important"
            {{ oldCheckbox('important', ( isset($note) ? $note : null) ) }} --}}
        -->
        {!! Form::checkbox('important', 1) !!}
        É importante
    </label>
</div>
```

resources/views/notes/edit.blade.php

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

@section('title')
    Editar: {{ $note->title }}
@endsection

@section('content')
    {{-- <form action="/notes/{{ $note->id }}" method="POST"
role="form"> --}}
    {!! Form::model($note, ['url' => "/notes/{$note->id}", "method" =>
"PATCH"]) !!}
        {{-- {{ method_field('PATCH') }}}}
        {{ csrf_field() --}}
        <legend>Editar anotação</legend>
        @include('partials.errors')
        @include('notes._form')
        <button type="submit" class="btn btn-primary">Editar</button>
    {!! Form::close() !!}
    {{-- </form> --}}
@endsection
```

resources/views/notes/create.blade.php

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

@section('title')
    Criar uma nova anotação
@endsection

@section('content')
    {{-- <form action="/notes" method="POST" role="form"> --}}
    {!! Form::open(['url' => "/notes", "method" => "POST"]) !!}
        {{-- {{ csrf_field() }} --}}
        <legend>Criar nova anotação</legend>
        @include('partials.errors')
        @include('notes._form')
        <button type="submit" class="btn btn-primary">Criar</button>
    {!! Form::close() !!}
    {{-- </form> --}}
@endsection
```

app/Http/Requests/NotesRequests.php

```
<?php

namespace App\Http\Controllers;

use App\Note;
use App\Group;
use App\Http\Requests\NotesRequest;
use Illuminate\Http\Request;

class NotesController extends Controller
{
    public function index(Group $group){
        $groups = Group::all();
        $notes = $group->notes;
        return view('notes.index', compact('notes', 'groups'));
    }

    public function create(){
        $groups = Group::pluck('name', 'id');
        return view('notes.create', compact('groups'));
    }

    public function show(Note $note){
        return view('notes.show', [ 'note' => $note ]);
    }

    public function store(NotesRequest $request){
        Note::create( $request->all() );
        return redirect('/notes');
    }

    public function edit(Note $note){
        $groups = Group::pluck('name', 'id');
        return view('notes.edit', compact('note', 'groups'));
    }

    public function update(Note $note, NotesRequest $request){
        $note->update( $request->all() );
        return redirect('/notes');
    }

    public function destroy(Note $note){
        $note->delete();
        return redirect('/notes');
    }
}
```

Editor: Graduação de Susan

Minhas anotações Anotações Contato

Editar anotação

Título

Anotação

Grupo

É importante]

Editar

Minhas anotações . Todos os direitos reservados

Criar uma nova anotação

Minhas anotações Anotações Contato

Criar nova anotação

Título

Anotação

Grupo

É importante]

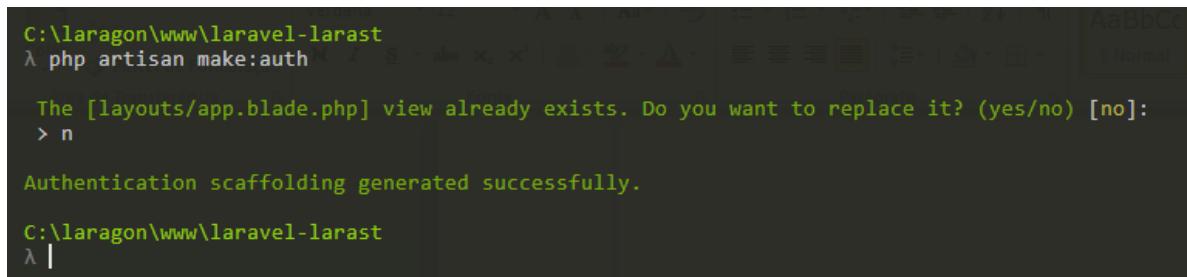
Criar

Minhas anotações . Todos os direitos reservados

Aula 24 - Sistema de autenticação

A maioria das aplicações necessitam de um sistema de autenticação (registro, login, logout, e reset de senha). Laravel permite mediante o comando make:auth, ter um sistema de autenticação completo em questão de minutos.

Criando autenticação



```
C:\laragon\www\laravel-larast
λ php artisan make:auth
The [layouts/app.blade.php] view already exists. Do you want to replace it? (yes/no) [no]:
> n

Authentication scaffolding generated successfully.

C:\laragon\www\laravel-larast
λ |
```

routes/web.php

```
<?php

Route::get('/notes', 'NotesWithoutGroupController@index');
Route::get('/groups/{group}/notes', 'NotesController@index');

Route::get('/notes/create', 'NotesController@create');
// Route::get('/notes/{id}', 'NotesController@show');
Route::get('/notes/{note}', 'NotesController@show');
Route::get('/notes/{note}/edit', 'NotesController@edit');
Route::patch('/notes/{note}', 'NotesController@update');
Route::post('/notes', 'NotesController@store');
Route::delete('/notes/{note}', 'NotesController@destroy');

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

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

Auth::routes();

Route::get('/home', 'HomeController@index')->name('home');
```

php artisan route:list

Domain	Method	URI	Name	Action	Middleware
	GET HEAD	/		Closure	web
	GET HEAD	api/user		Closure @annotation () { return view(''); }	api,auth:api
	GET HEAD	contact		Closure	web
	GET HEAD	groups/{group}/notes		App\Http\Controllers\NotesController@index	web
	GET HEAD	home	home	App\Http\Controllers\HomeController@index	web,auth
	GET HEAD	login	login	App\Http\Controllers\Auth\LoginController@showLoginForm	web,guest
	POST	login		App\Http\Controllers\Auth\LoginController@login	web,guest
	POST	logout	logout	App\Http\Controllers\Auth\LoginController@logout	web
	POST	notes		App\Http\Controllers\NotesController@store	web
	GET HEAD	notes		App\Http\Controllers\NotesWithoutGroupController@index	web
	GET HEAD	notes/create		App\Http\Controllers\NotesController@create	web
	DELETE	notes/{note}		App\Http\Controllers\NotesController@destroy	web
	PATCH	notes/{note}		App\Http\Controllers\NotesController@update	web
	GET HEAD	notes/{note}/edit		App\Http\Controllers\NotesController@show	web
	POST	password/email	password.email	App\Http\Controllers\NotesController@edit	web
	GET HEAD	password/reset	password.request	App\Http\Controllers\Auth\ForgotPasswordController@sendResetLinkEmail	web,guest
	POST	password/reset		App\Http\Controllers\Auth\ForgotPasswordController@showLinkRequestForm	web,guest
	GET HEAD	password/reset/{token}	password.reset	App\Http\Controllers\Auth\ResetPasswordController@reset	web,guest
	GET HEAD	register	register	App\Http\Controllers\Auth\RegisterController@showRegistrationForm	web,guest
	POST	register		App\Http\Controllers\Auth\RegisterController@register	web,guest

No arquivo `resources/views/layouts/app.blade.php` insira o seguinte código em vermelho:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>@yield('title')</title>
    <link rel="stylesheet" href="/css/app.css">
</head>
<body>

    <div class="container">
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <a class="navbar-brand" href="/">{{ config('app.name') }}</a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav">
                    <li class="nav-item dropdown {{ Request::is('notes') ? 'active' : '' }}>
                        <a class="nav-link dropdown-toggle" href="#" id="navbardrop" data-toggle="dropdown">
                            Anotações
                        </a>
                        <div class="dropdown-menu">
                            <a class="dropdown-item" href="/notes">Ver</a>
                            <a class="dropdown-item" href="/notes/create">Criar</a>
                        </div>
                    </li>
                    <li class="{{ Request::is('contact') ? 'active' : '' }}>
                        <a class="nav-link" href="/contact">Contato</a>
                    </li>
                </ul>
```

```

<!-- Right Side Of Navbar -->
<ul class="navbar-nav ml-auto">
    <!-- Authentication Links -->
    @guest
        <li class="nav-item">
            <a class="nav-link" href="{{ route('login') }}>{{ __('Login') }}</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="{{ route('register') }}>{{ __('Register') }}</a>
        </li>
    @else
        <li class="nav-item dropdown">
            <a id="navbarDropdown" class="nav-link dropdown-toggle" href="#" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false" v-pre>
                {{ Auth::user()->name }} <span class="caret"></span>
            </a>

            <div class="dropdown-menu dropdown-menu-right" aria-labelledby="navbarDropdown">
                <a class="dropdown-item" href="{{ route('logout') }}"
                    onclick="event.preventDefault();
                               document.getElementById('logout-form').submit();">
                    {{ __('Logout') }}
                </a>

                <form id="logout-form" action="{{ route('logout') }}"
                    method="POST" style="display: none;">
                    @csrf
                </form>
            </div>
        </li>
    @endguest
</ul>

</div>
</nav>
</div>

<div class="container">
{{--    <ul>
        <li>{{ Auth::user() -> name }}</li>
        <li>{{ Auth::user() -> email }}</li>
    </ul>
--}}
    @yield('content')
}
</div>

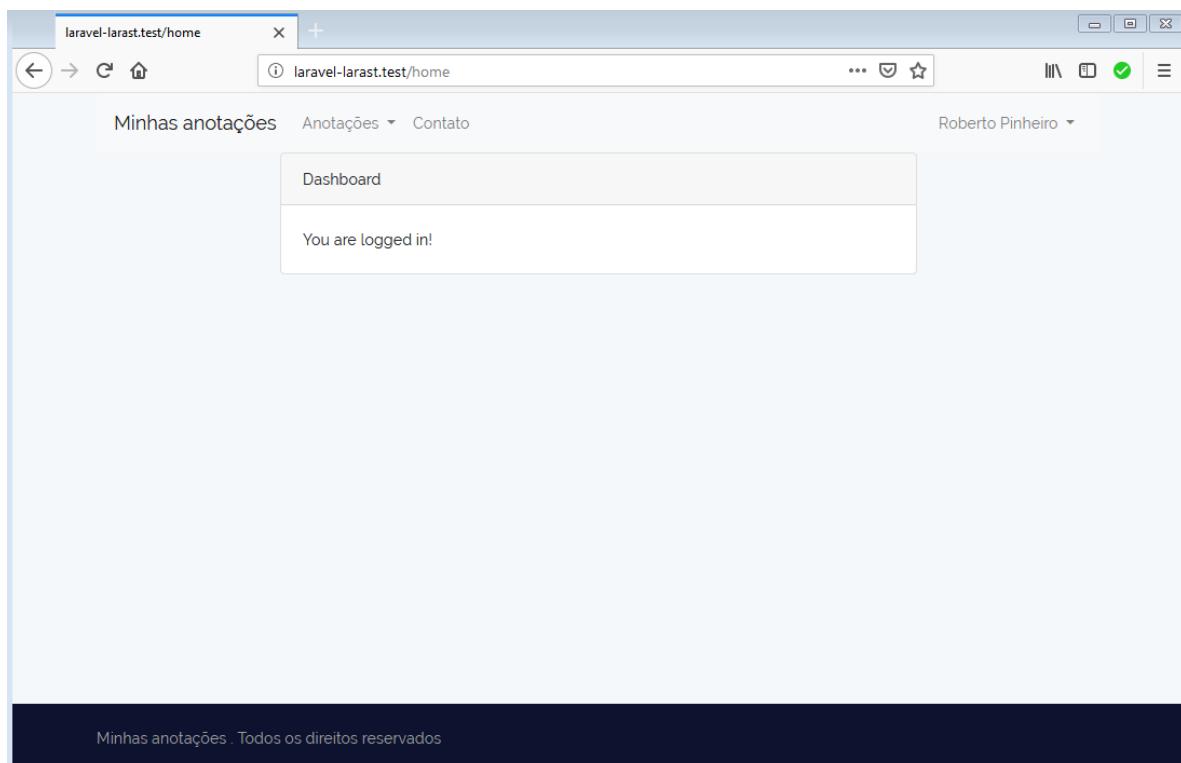
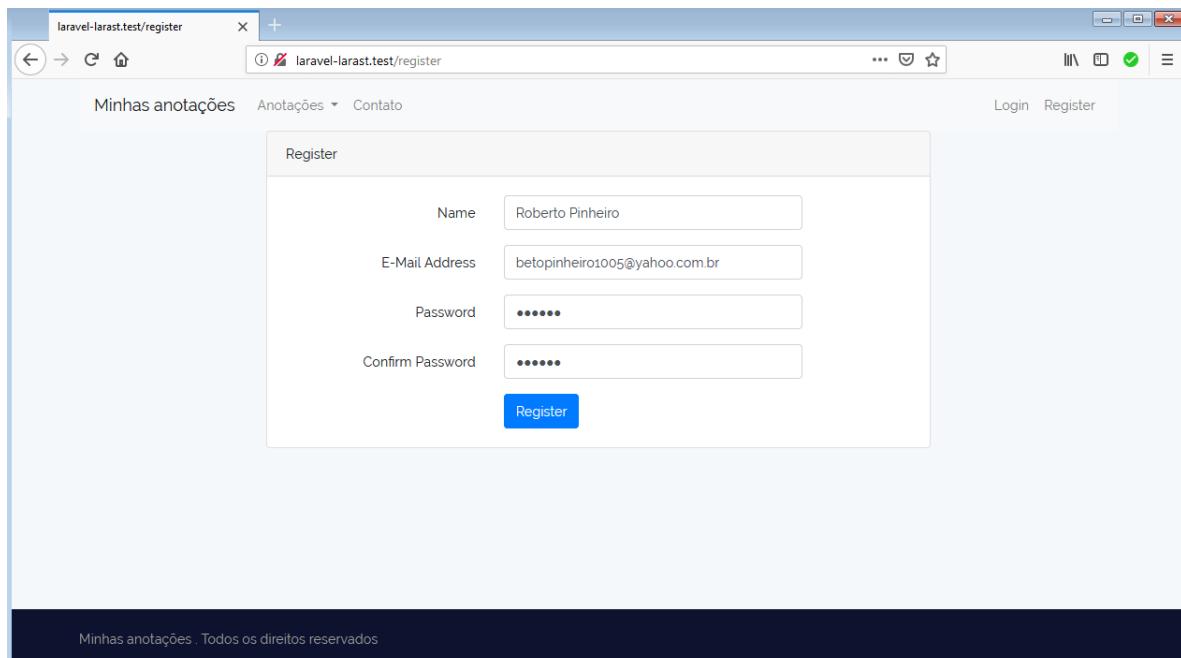
<footer>
    <div class="container">
        {{ config('app.name') }} . Todos os direitos reservados
    </div>
</footer>

<script src="/js/app.js"></script>

```

```
@yield('scripts')  
</body>  
</html>
```

Senha: 123456



Showing rows 0 - 0 (1 total. Query took 0.0030 seconds.)

```
SELECT * FROM `users`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options id name email password remember_token created_at updated_at

1 Roberto Pinheiro betopinheiro1005@yahoo.com.br \$2y\$10\$trxFm66UZT/mcV5rQ1xO7u1DPhsuXu6hrp2kf.CIRL... NULL 2019-05-01 01:48:14 2019-05-01 01:48:14

Check all With selected:

Resetando a senha

Acesse a url:

<https://mailtrap.io/>

Faça o login com a conta do google:

Log In

Email

Password [Forgot password?](#)

Remember me

Mailtrap renders emails in the same way browsers do, meaning that no additional stylesheet or CSS reset is applied.

Log in

Haven't received the confirmation email yet? [Resend the email](#)

OR

 Use Google account

 Use Github account

The screenshot shows the Mailtrap.io inbox interface. At the top, there's a navigation bar with links for 'Shared Inboxes' and 'Billing'. On the right, it shows the user's name 'Roberto Pinheiro Beto' and a 'Upgrade' button. Below the navigation is a search bar labeled 'Inbox name' and a 'Create Inbox' button. The main area is titled 'My Inboxes' and shows a table with one inbox entry: 'Testes'. The table columns include 'Inboxes', 'Total Sent', 'Messages', 'Max size', 'Last message', and 'Action'. The 'Testes' row shows 0 messages sent, 0/1 messages received, a max size of 50, and a last message received 2 months ago. The 'Action' column contains icons for reply, forward, and delete.

Clique no link do seu inbox:

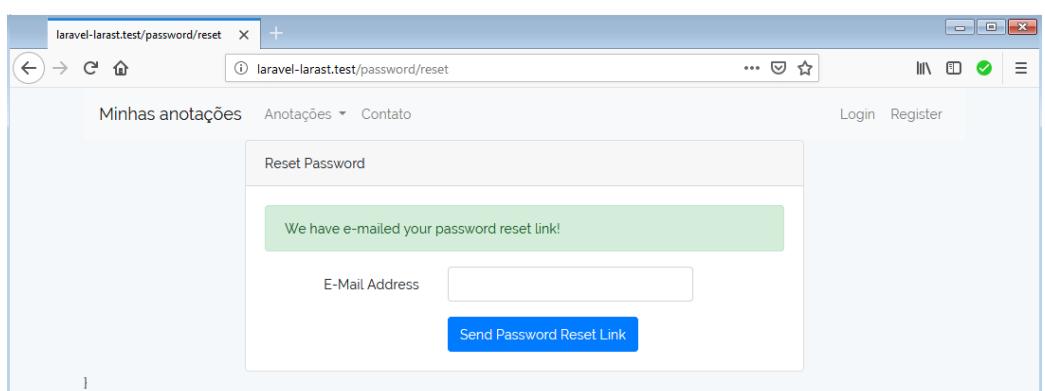
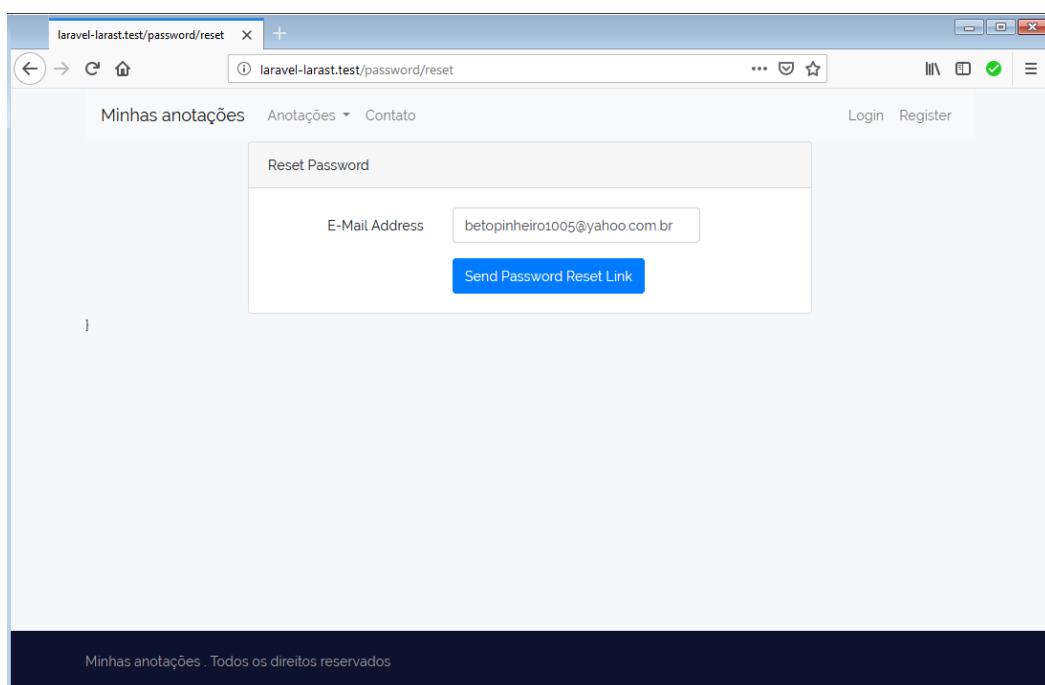
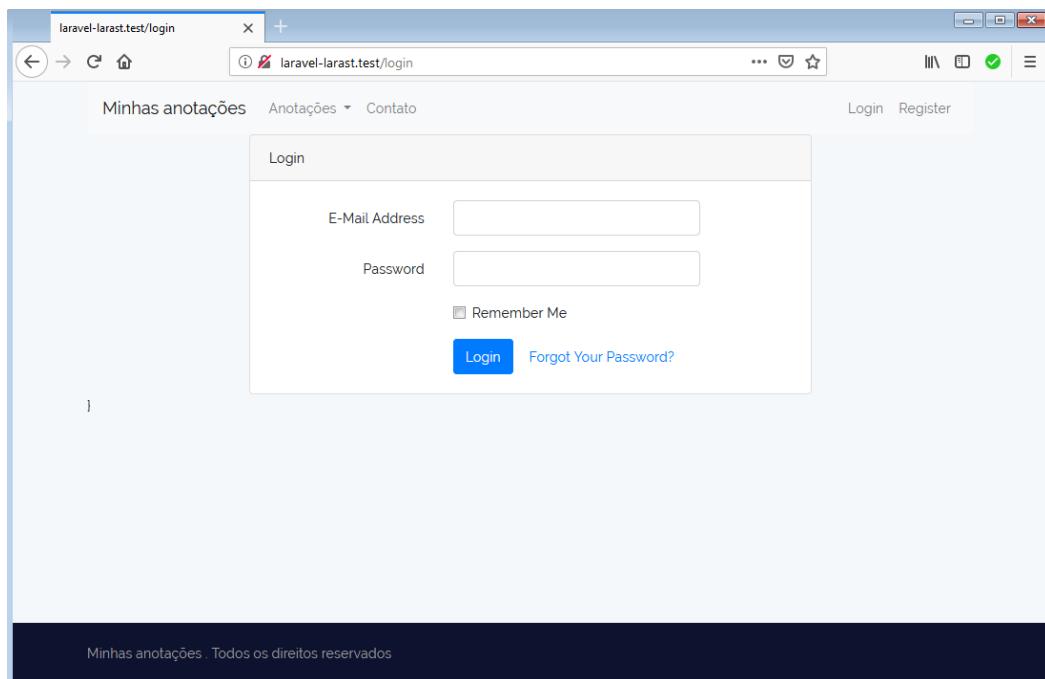
The screenshot shows the 'Testes' inbox details page. At the top, it displays 'Home / Testes' and 'Total messages sent: 0'. Below this is a navigation bar with tabs: 'SMTP Settings' (which is active), 'Email Address', 'Auto Forward', 'Manual Forward', and 'Team Members'. The main section is titled 'Credentials' with a 'Reset SMTP/POP3' link. It is divided into two columns: 'SMTP' and 'POP3'. The 'SMTP' section lists: Host: smtp.mailtrap.io, Port: 25 or 465 or 587 or 2525, Username: f7f9b1c636f954, Password: c6263786f720f6, Auth: PLAIN, LOGIN and CRAM-MD5, and TLS: Optional (STARTTLS on all ports). The 'POP3' section lists: Host: pop3.mailtrap.io, Port: 1100 or 9950, Username: f7f9b1c636f954, Password: c6263786f720f6, Auth: USER/PASS, PLAIN, LOGIN, APOP and CRAM-MD5, and TLS: Optional (STARTTLS on all ports). Below this is a section titled 'Integrations' with a dropdown menu set to 'Ruby on Rails'. It contains code snippets for ActionMailer configuration:

```
config.action_mailer.delivery_method = :smtp
config.action_mailer.smtp_settings = {
  :user_name => 'f7f9b1c636f954',
  :password => 'c6263786f720f6'.
```

Abra o arquivo de configuração **.env** e informe os dados de acordo com suas credenciais:

```
APP_NAME="Minhas anotações"
APP_ENV=local
APP_KEY=base64:M29FqBdy1ir30TerdGYjv3YPRsMN+3k8vDvIgM+stTw=
APP_DEBUG=true
APP_URL=http://laravel-larast.test
```

```
MAIL_DRIVER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=f7f9b1c636f954
MAIL_PASSWORD=c6263786f720f6
MAIL_ENCRYPTION=tls
```



The screenshot shows a browser window displaying the Mailtrap.io inbox at <https://mailtrap.io/inboxes/563109/messages/1250965186>. The inbox lists several messages, including a "Reset Password Notification" from "Example <hello@example.com>" to "betopinheiro1005@yahoo.com.br" received 2 minutes ago. The message content is displayed in an HTML viewer, showing a "Hello!" greeting, a password reset request, and a prominent blue "Reset Password" button.

Altere a senha para 1234567

The screenshot shows a browser window with a password reset form. The URL is <https://laravel-larast.test/password/reset/ea813cd6e08c4e229dc13d26d3b61733113b06a0d0...>. The form is titled "Reset Password" and contains fields for "E-Mail Address" (filled with "betopinheiro1005@yahoo.com.br"), "Password" (filled with "....."), and "Confirm Password" (filled with "....."). A blue "Reset Password" button is at the bottom. The page includes navigation links for "Minhas anotações", "Anotações", "Contato", "Login", and "Register".

← → C ⓘ Não seguro | laravel-larast.test/home

Minhas anotações Anotações Contato Roberto Pinheiro

Dashboard

Your password has been reset!

You are logged in!

]

Minhas anotações . Todos os direitos reservados

Aula 25 - Personalizar o sistema de autenticação

Nesta lição, ajustaremos alguns detalhes do sistema de autenticação que Laravel nos entregou por padrão.

app/User.php

```
<?php

namespace App;

use App\Notifications\ResetPassword;
use Illuminate\Notifications\Notifiable;
use Illuminate\Foundation\Auth\User as Authenticatable;

class User extends Authenticatable
{
    use Notifiable;

    protected $fillable = [
        'name', 'email', 'password',
    ];

    protected $hidden = [
        'password', 'remember_token',
    ];

    public function sendPasswordResetNotification($token)
    {
        $this->notify(new ResetPassword($token));
    }
}
```

app/Notifications/ResetPassword.php

```
<?php

namespace App\Notifications;

use Illuminate\Support\Facades\Lang;
use Illuminate\Notifications\Notification;
use Illuminate\Notifications\Messages\MailMessage;

class ResetPassword extends Notification
{
    public $token;

    public static $toMailCallback;

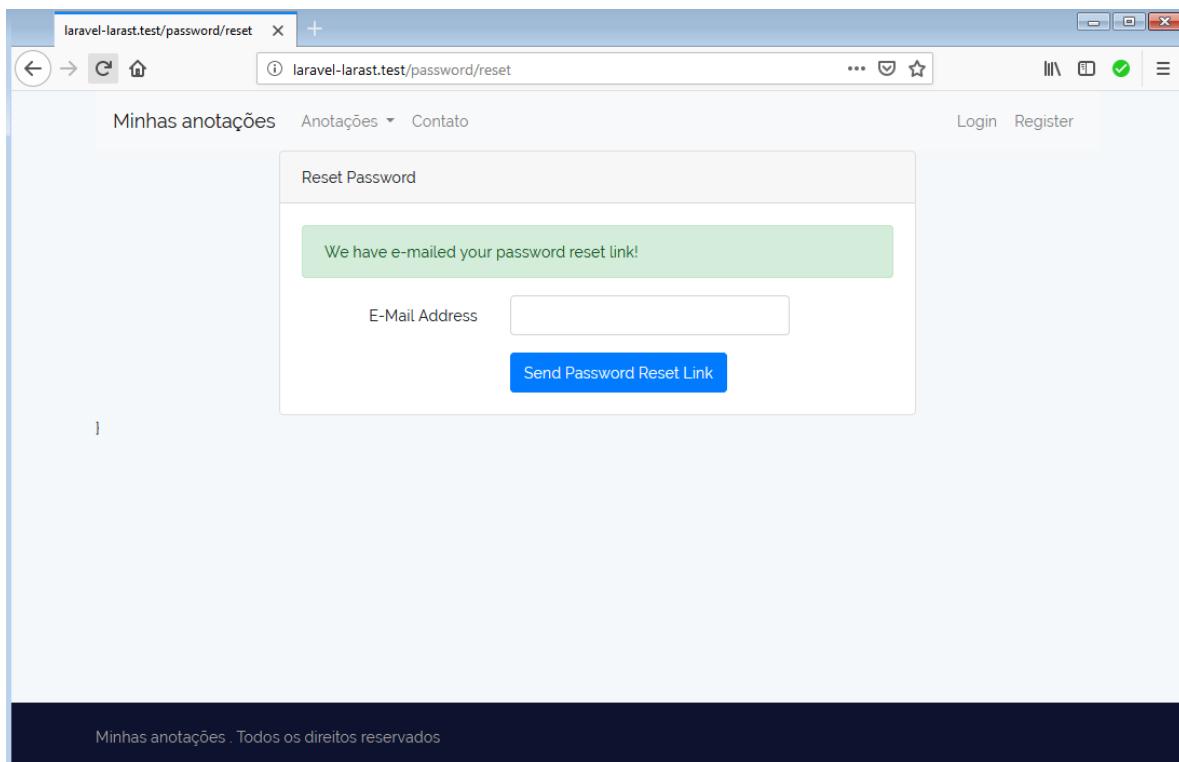
    public function __construct($token)
    {
        $this->token = $token;
    }

    public function via($notifiable)
    {
        return ['mail'];
    }

    public function toMail($notifiable)
    {
        if (static::$toMailCallback) {
            return call_user_func(static::$toMailCallback, $notifiable, $this->token);
        }

        return (new MailMessage)
            ->subject(Lang::getFromJson('Notificação de reset de senha'))
            ->line(Lang::getFromJson('Você está recebendo esta mensagem porque
recebemos uma solicitação de reset de senha.'))
            ->action(Lang::getFromJson('Resetar Senha'),
url(config('app.url').route('password.reset', $this->token, false)))
            ->line(Lang::getFromJson('Se você não solicitou o reset de senha,
ignore essa mensagem.'));
    }

    public static function toMailUsing($callback)
    {
        static::$toMailCallback = $callback;
    }
}
```



Home / Testes / Notificação de reset de senha

Notificação de reset de senha

From: Example <hello@example.com>
To: <betopinheiro1005@yahoo.com.br>
[More info](#)

2019-05-01 20:18
(11 minutes ago)
Size: 9.8 KB

HTML HTML Source Text Raw Analysis Check HTML

Minhas anotações

Hello!

Você está recebendo esta mensagem porque recebemos uma solicitação de reset de senha.

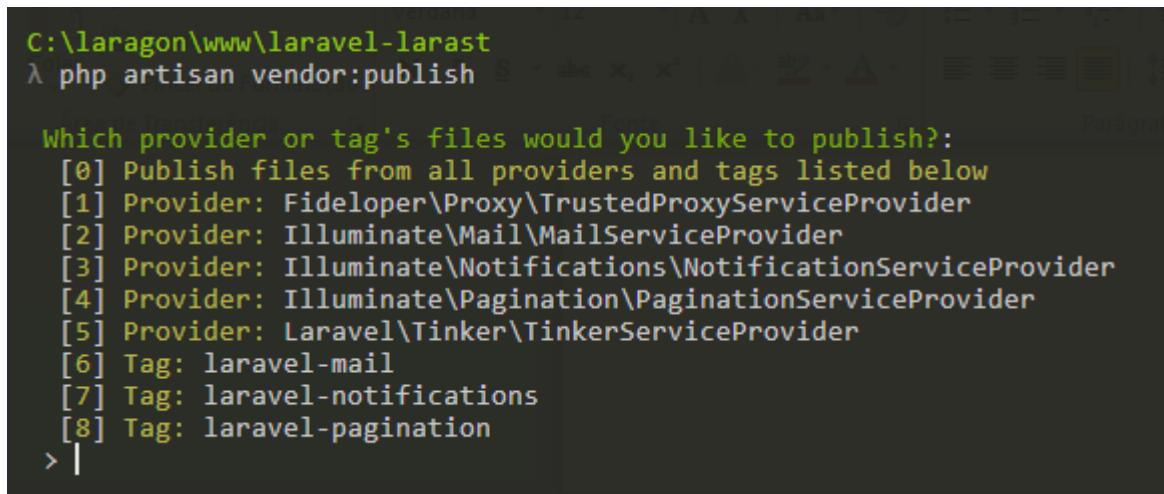
[Resetar Senha](#)

Se você não solicitou o reset de senha, ignore essa mensagem.

Regards,
Minhas anotações

Publicando notificação

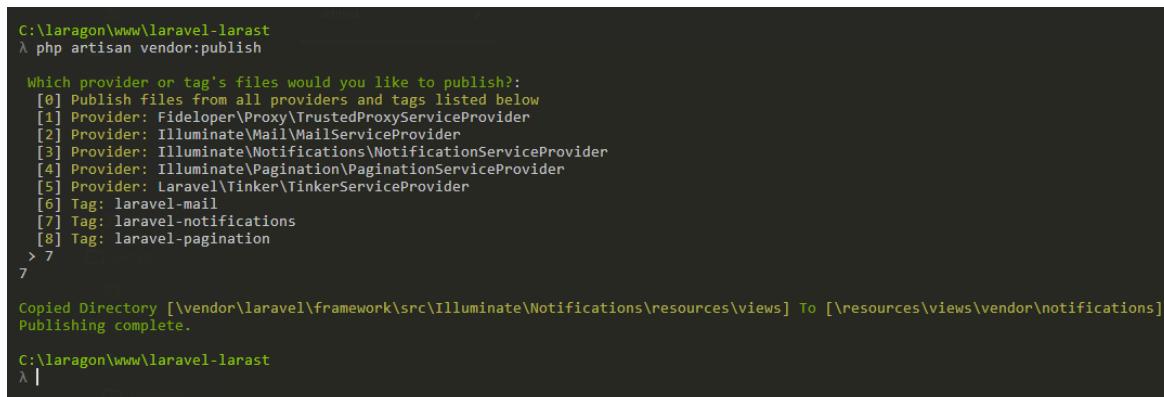
php artisan vendor:publish



```
C:\laragon\www\laravel-larast
λ php artisan vendor:publish

Which provider or tag's files would you like to publish?:
[0] Publish files from all providers and tags listed below
[1] Provider: Fideloper\Proxy\TrustedProxyServiceProvider
[2] Provider: Illuminate\Mail\MailServiceProvider
[3] Provider: Illuminate\Notifications\NotificationServiceProvider
[4] Provider: Illuminate\Pagination\PaginationServiceProvider
[5] Provider: Laravel\Tinker\TinkerServiceProvider
[6] Tag: laravel-mail
[7] Tag: laravel-notifications
[8] Tag: laravel-pagination
> |
```

Selecione a opção 7 (Laravel Notifications)



```
C:\laragon\www\laravel-larast
λ php artisan vendor:publish

Which provider or tag's files would you like to publish?:
[0] Publish files from all providers and tags listed below
[1] Provider: Fideloper\Proxy\TrustedProxyServiceProvider
[2] Provider: Illuminate\Mail\MailServiceProvider
[3] Provider: Illuminate\Notifications\NotificationServiceProvider
[4] Provider: Illuminate\Pagination\PaginationServiceProvider
[5] Provider: Laravel\Tinker\TinkerServiceProvider
[6] Tag: laravel-mail
[7] Tag: laravel-notifications
[8] Tag: laravel-pagination
> 7
7

Copied Directory [\"vendor\\laravel\\framework\\src\\Illuminate\\Notifications\\resources\\views\"] To [\"resources\\views\\vendor\\notifications\"]
Publishing complete.

C:\laragon\www\laravel-larast
λ |
```

resources/views/vendor/notifications/email.php

```
@component('mail::message')
{{-- Greeting --}}
@if (! empty($greeting))
# {{ $greeting }}
@else
@if ($level == 'error')
# @lang('Whoops!')
@else
# @lang('Olá!')
@endif
@endif
```

```

{{-- Intro Lines --}}
@foreach ($introLines as $line)
{{ $line }}

@endforeach

{{-- Action Button --}}
@isset($actionText)
<?php
    switch ($level) {
        case 'success':
            $color = 'green';
            break;
        case 'error':
            $color = 'red';
            break;
        default:
            $color = 'blue';
    }
?>
@component('mail::button', ['url' => $actionUrl, 'color' => $color])
{{ $actionText }}
@endcomponent
@endisset

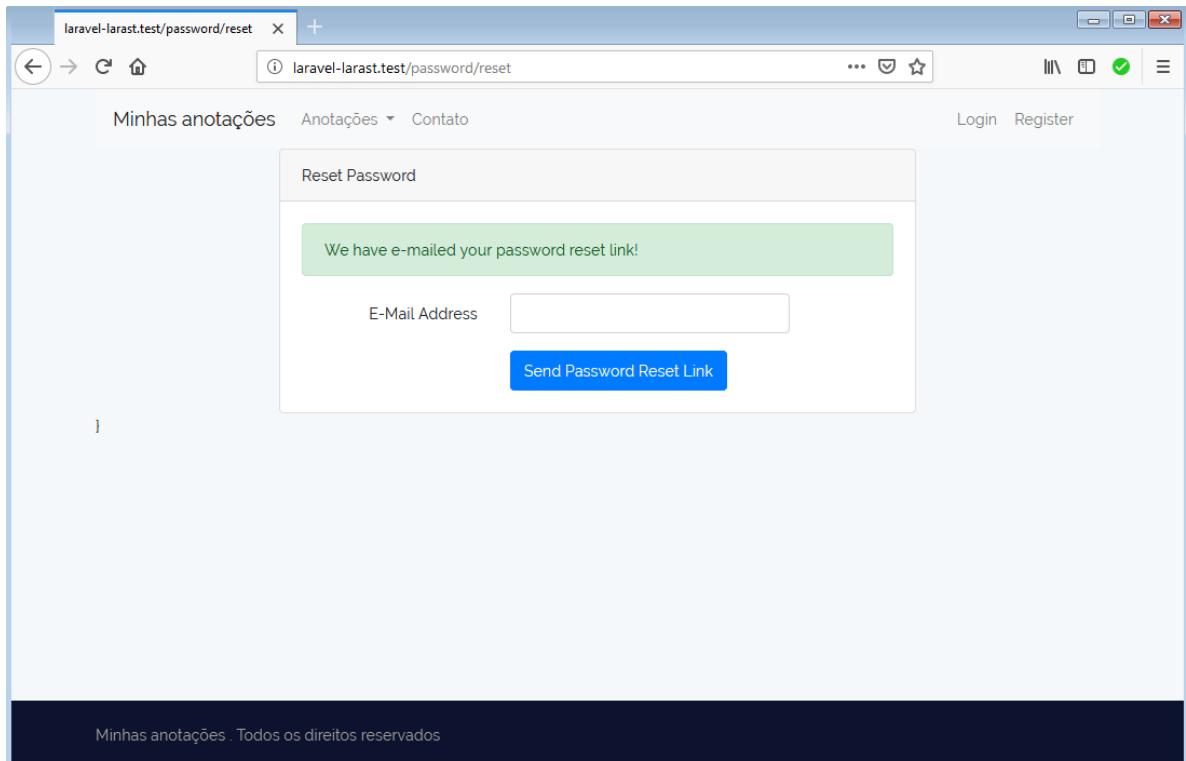
{{-- Outro Lines --}}
@foreach ($outroLines as $line)
{{ $line }}

@endforeach

{{-- Salutation --}}
@if (! empty($salutation))
{{ $salutation }}
@else
@lang('Saudações'),<br>{{ config('app.name') }}
@endif

{{-- Subcopy --}}
@isset($actionText)
@component('mail::subcopy')
@lang(
    "Caso tenha problemas clicando no botão \"{:actionText}\", copie e cole a URL
    abaixo\n".
    'em seu navegador: [:actionURL](:actionURL)',
    [
        'actionText' => $actionText,
        'actionURL' => $actionUrl
    ]
)
@endcomponent
@endisset
@endcomponent

```



A screenshot of an email client interface. The subject of the email is "Notificação de reset de senha". The email content is as follows:

Notificação de reset de senha

From: Example <hello@example.com>
To: <betopinheiro1005@yahoo.com.br>
More info

2019-05-01 20:55
(a few seconds ago)
Size: 9.8 KB

HTML HTML Source Text Raw Analysis Check HTML

Minhas anotações

Olá!
Você está recebendo esta mensagem porque recebemos uma solicitação de reset de senha.

Resetar Senha

Se você não solicitou o reset de senha, ignore essa mensagem.

Saudações,
Minhas anotações

Caso tenha problemas clicando no botão "Resetar Senha", copie e cole a URL abaixo em seu navegador: <http://laravel-larast.test/password/reset/283de164c51ecfe7407334b5bc0ef045592240e2e0b5b09248da10e328c1629>

php artisan vendor:publish

```
C:\laragon\www\laravel-larast
\ php artisan vendor:publish

Which provider or tag's files would you like to publish?:
[0] Publish files from all providers and tags listed below
[1] Provider: Fideloper\Proxy\TrustedProxyServiceProvider
[2] Provider: Illuminate\Mail\MailServiceProvider
[3] Provider: Illuminate\Notifications\NotificationServiceProvider
[4] Provider: Illuminate\Pagination\PaginationServiceProvider
[5] Provider: Laravel\Tinker\TinkerServiceProvider
[6] Tag: laravel-mail
[7] Tag: laravel-notifications
[8] Tag: laravel-pagination
> 6

Copied Directory [vendor\laravel\framework\src\Illuminate\Mail\resources\views] To [\resources\views\vendor\mail]
Publishing complete.

C:\laragon\www\laravel-larast
\ |
```

Selecione a opção 6 (laravel mail)

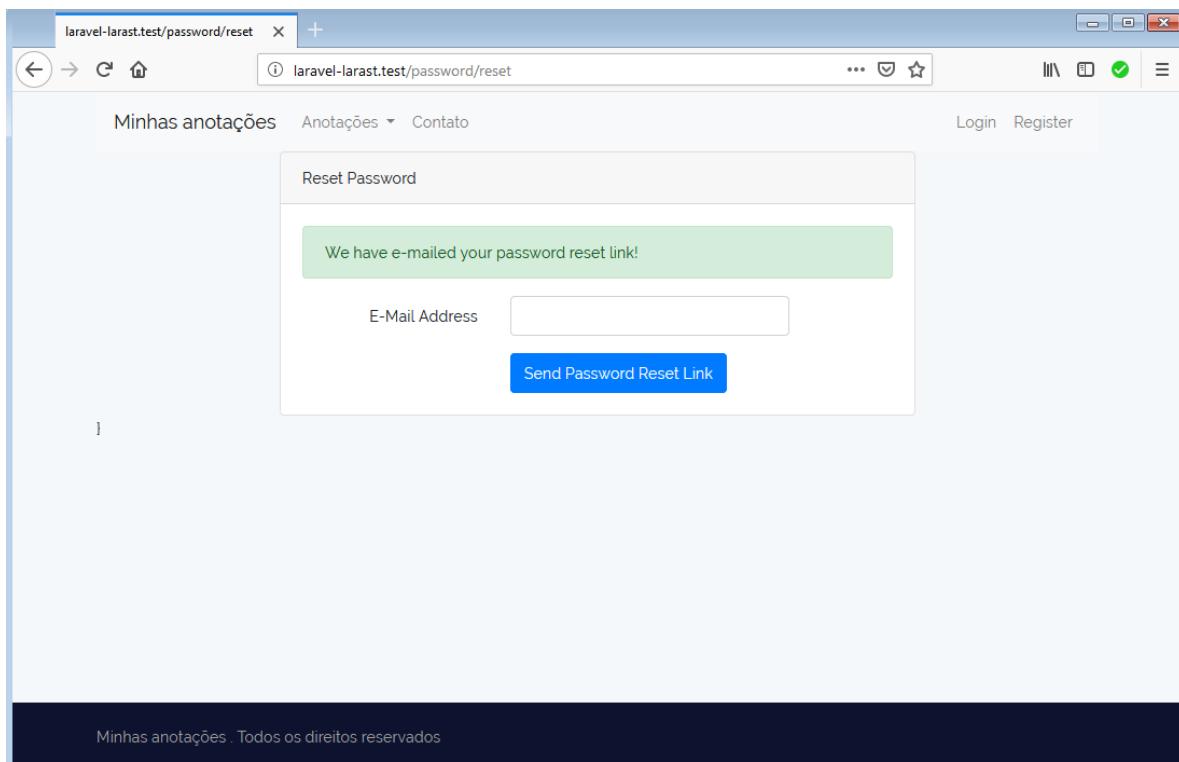
resources/views/vendor/mail/html/message.blade.php

```
@component('mail::layout')
{{-- Header --}}
@slot('header')
    @component('mail::header', ['url' => config('app.url')])
        {{ config('app.name') }}
    @endcomponent
@endslot

{{-- Body --}}
{{ $slot }}

{{-- Subcopy --}}
@if($subcopy)
    @slot('subcopy')
        @component('mail::subcopy')
            {{ $subcopy }}
        @endcomponent
    @endslot
@endif

{{-- Footer --}}
@slot('footer')
    @component('mail::footer')
        &copy; {{ date('Y') }} {{ config('app.name') }}. Todos os direitos
        reservados.
    @endcomponent
@endslot
@endcomponent
```



A screenshot of an email client interface. The subject of the email is "Notificação de reset de senha". The email content is as follows:

From: Example <hello@example.com>
To: <betopinheiro1005@yahoo.com.br>
More info

2019-05-02 00:27
(a few seconds ago)
Size: 9.8 kB

Você está recebendo esta mensagem porque recebemos uma solicitação de reset de senha.

[Resetar Senha](#)

Se você não solicitou o reset de senha, ignore essa mensagem.

Saudações,
Minhas anotações

Caso tenha problemas clicando no botão "Resetar Senha", copie e cole a URL abaixo em seu navegador: <http://laravel-larast.test/password/reset/eb76080b3faed88d8a78015520c60278dd14a580e8483b7dddc49fb3acd56463>

© 2019 Minhas anotações. Todos os direitos reservados.

resources/lang/pt-br/auth.php

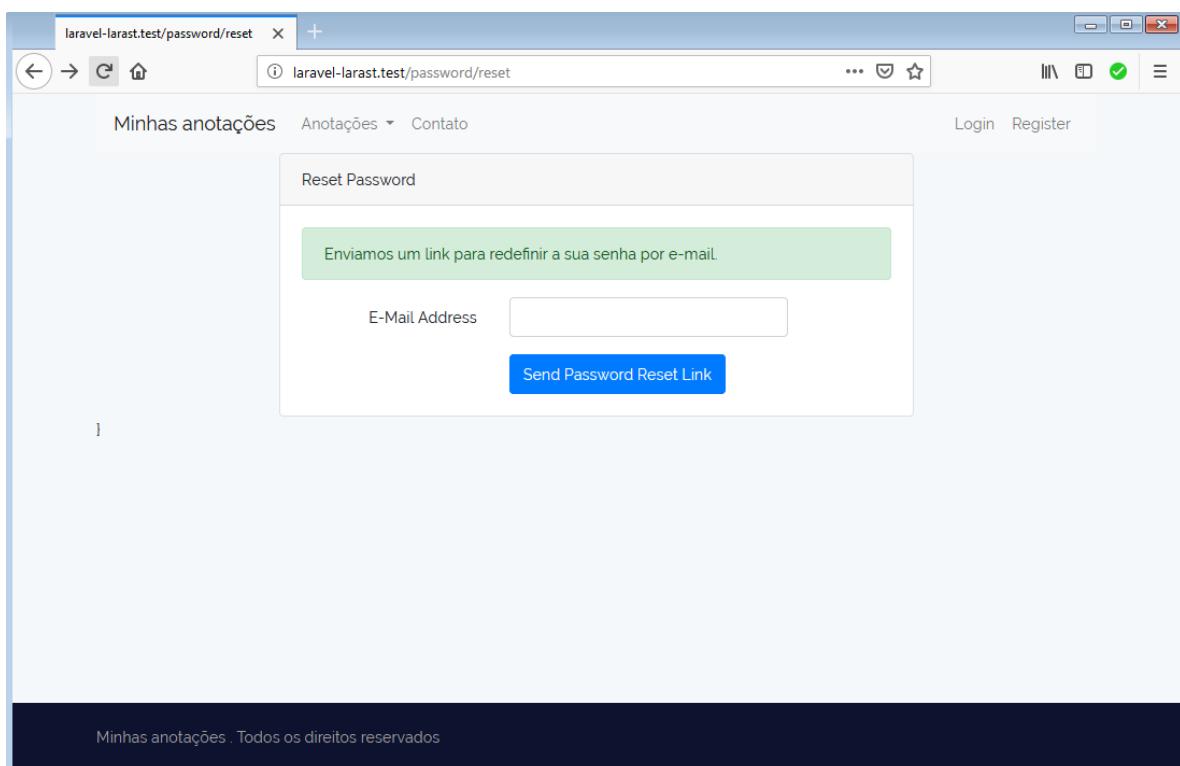
```
<?php
return [
/*
| -----
| Authentication Language Lines
| -----
|
| The following language lines are used during authentication for various
| messages that we need to display to the user. You are free to modify
| these language lines according to your application's requirements.
|
*/
'failed' => 'Credenciais informadas não correspondem com nossos registros.',
'throttle' => 'Você realizou muitas tentativas de login. Por favor, tente novamente em
:seconds segundos.',
];
];
```

resources/lang/pt-br/passwords.php

```
<?php
return [
/*
| -----
| Password Reminder Language Lines
| -----
|
| The following language lines are the default lines which match reasons
| that are given by the password broker for a password update attempt
| has failed, such as for an invalid token or invalid new password.
|
*/
'password' => 'A senha deve conter pelo menos seis caracteres e ser igual à
confirmação.',
'reset' => 'Sua senha foi redefinida!',
'sent' => 'Enviamos um link para redefinir a sua senha por e-mail.',
'token' => 'Esse código de redefinição de senha é inválido.',
'user' => 'Não conseguimos encontrar nenhum usuário com o endereço de e-mail
informado.',
];
];
```

resources/lang/pt-br/pagination.php

```
<?php
return [
/*
-----
| Pagination Language Lines
-----
|
| The following language lines are used by the paginator library to build
| the simple pagination links. You are free to change them to anything
| you want to customize your views to better match your application.
|
*/
'previous' => '&laquo; Anterior',
'next'     => 'Próxima &raquo;',
];
];
```



laravel-larast.test/login

Minhas anotações Anotações Contato Login Register

Login

E-Mail Address betopinheiro1005@yahoo.com.br ×

Credenciais informadas não correspondem com nossos registros.

Password

Remember Me

[Login](#) [Forgot Your Password?](#)

Minhas anotações . Todos os direitos reservados

This screenshot shows a web browser window with a blue header bar containing the URL 'laravel-larast.test/login'. Below the header is a navigation bar with links for 'Minhas anotações', 'Anotações', 'Contato', 'Login', and 'Register'. The main content area has a title 'Login' and two input fields: 'E-Mail Address' and 'Password'. The 'E-Mail Address' field contains 'betopinheiro1005@yahoo.com.br' with a red 'X' icon to its right. Below the field is the error message 'Credenciais informadas não correspondem com nossos registros.' (The credentials provided do not match our records). The 'Password' field is empty. There is also a 'Remember Me' checkbox and two buttons at the bottom: 'Login' (blue) and 'Forgot Your Password?'. At the very bottom of the page is a dark footer bar with the text 'Minhas anotações . Todos os direitos reservados'.

← → C Não seguro | laravel-larast.test/home

Minhas anotações Anotações Contato Roberto Pinheiro

Dashboard

Sua senha foi redefinida!

You are logged in!

Minhas anotações . Todos os direitos reservados

This screenshot shows a web browser window with a grey header bar containing the URL 'laravel-larast.test/home'. Below the header is a navigation bar with links for 'Minhas anotações', 'Anotações', 'Contato', and a user profile 'Roberto Pinheiro'. The main content area has a title 'Dashboard' and a green message box containing 'Sua senha foi redefinida!' (Your password was reset). Below the message box is the text 'You are logged in!'. At the bottom of the page is a dark footer bar with the text 'Minhas anotações . Todos os direitos reservados'.

Redirecionando a página inicial para anotações

app/http/Controllers/Auth/LoginController.php

```
<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use Illuminate\Foundation\Auth\ResetsPasswords;

class ResetPasswordController extends Controller
{
    /*
    |--------------------------------------------------------------------------
    | Password Reset Controller
    |--------------------------------------------------------------------------
    |

    This controller is responsible for handling password reset requests
    and uses a simple trait to include this behavior. You're free to
    explore this trait and override any methods you wish to tweak.
    |

    */
    use ResetsPasswords;

    protected $redirectTo = '/notes';

    public function __construct()
    {
        $this->middleware('guest');
    }
}
```

app/http/Controllers/Auth/RegisterController.php

```
<?php

namespace App\Http\Controllers\Auth;

use App\User;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;
use Illuminate\Foundation\Auth\RegistersUsers;
```

```
class RegisterController extends Controller
{
    /*
    -----
    | Register Controller
    -----
    |
    | This controller handles the registration of new users as well as their
    | validation and creation. By default this controller uses a trait to
    | provide this functionality without requiring any additional code.
    |
    */
    use RegistersUsers;

    protected $redirectTo = '/notes';

    public function __construct()
    {
        $this->middleware('guest');
    }

    protected function validator(array $data)
    {
        return Validator::make($data, [
            'name' => 'required|string|max:255',
            'email' => 'required|string|email|max:255|unique:users',
            'password' => 'required|string|min:6|confirmed',
        ]);
    }

    protected function create(array $data)
    {
        return User::create([
            'name' => $data['name'],
            'email' => $data['email'],
            'password' => Hash::make($data['password']),
        ]);
    }
}
```

app/http/Controllers/Auth/ResetPasswordController.php

```
<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use Illuminate\Foundation\Auth\ResetsPasswords;

class ResetPasswordController extends Controller
{
    /*
    |--------------------------------------------------------------------------
    | Password Reset Controller
    |--------------------------------------------------------------------------
    |

    | This controller is responsible for handling password reset requests
    | and uses a simple trait to include this behavior. You're free to
    | explore this trait and override any methods you wish to tweak.
    |

    */

    use ResetsPasswords;

    protected $redirectTo = '/notes';

    public function __construct()
    {
        $this->middleware('guest');
    }
}
```

app/http/Controllers/Middleware/RedirectIfAuthenticated.php

```
<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Support\Facades\Auth;

class RedirectIfAuthenticated
{
    public function handle($request, Closure $next, $guard = null)
    {
        if (Auth::guard($guard)->check()) {
            return redirect('/notes');
        }

        return $next($request);
    }
}
```

laravel-larast.test/login

Minhas anotações Anotações Contato

Login

E-Mail Address betopinheiro1005@yahoo.com.br

Password *****

Remember Me

[Login](#) [Forgot Your Password?](#)

Minhas anotações . Todos os direitos reservados

Listagem de anotações

laravel-larast.test/notes

Minhas anotações Anotações Contato Roberto Pinheiro

Grupos

Anotação	Grupo	Editar	Eliminar
Comprar livro		Editar	Eliminar
Fazer compras		Editar	Eliminar

--Nenhum--

Familia

Trabalho

Minhas anotações . Todos os direitos reservados

Inserindo campo adicional no formulário de registro

resources/views/auth/register.blade.php

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

@section('content')


{{ __('Register') }}



<form method="POST" action="{{ route('register') }}" aria-label="{{ __('Register') }}">
@csrf

<div class="form-group row">
<label for="name" class="col-md-4 col-form-label text-md-right">{{ __('Name') }}
<div class="col-md-6">
<input id="name" type="text" class="form-control{{ $errors->has('name') ? ' is-invalid' : '' }}" name="name" value="{{ old('name') }}" required autofocus>

@if ($errors->has('name'))
<span class="invalid-feedback" role="alert">
<strong>{{ $errors->first('name') }}</strong>
</span>
@endif
</div>
</div>

<div class="form-group row">
<label for="cpf" class="col-md-4 col-form-label text-md-right">{{ __('CPF') }}
<div class="col-md-6">
<input id="cpf" type="text" placeholder="Só numeros" maxlength="11" class="form-control{{ $errors->has('cpf') ? ' is-invalid' : '' }}" name="cpf" value="{{ old('cpf') }}" required autofocus>

@if ($errors->has('cpf'))
<span class="invalid-feedback" role="alert">
<strong>{{ $errors->first('cpf') }}</strong>
</span>
@endif
</div>
</div>


```

```

<div class="form-group row">
    <label for="email" class="col-md-4 col-form-label text-md-right">{{ __('E-Mail Address') }}</label>

    <div class="col-md-6">
        <input id="email" type="email" class="form-control{{ $errors->has('email') ? ' is-invalid' : '' }}" name="email" value="{{ old('email') }}" required>

        @if ($errors->has('email'))
            <span class="invalid-feedback" role="alert">
                <strong>{{ $errors->first('email') }}</strong>
            </span>
        @endif
    </div>
</div>

<div class="form-group row">
    <label for="password" class="col-md-4 col-form-label text-md-right">{{ __('Password') }}</label>

    <div class="col-md-6">
        <input id="password" type="password" class="form-control{{ $errors->has('password') ? ' is-invalid' : '' }}" name="password" required>

        @if ($errors->has('password'))
            <span class="invalid-feedback" role="alert">
                <strong>{{ $errors->first('password') }}</strong>
            </span>
        @endif
    </div>
</div>

<div class="form-group row">
    <label for="password-confirm" class="col-md-4 col-form-label text-md-right">{{ __('Confirm Password') }}</label>

    <div class="col-md-6">
        <input id="password-confirm" type="password" class="form-control" name="password_confirmation" required>
    </div>
</div>

<div class="form-group row mb-0">
    <div class="col-md-6 offset-md-4">
        <button type="submit" class="btn btn-primary">
            {{ __('Register') }}
        </button>
    </div>
</div>
</form>
</div>
</div>
</div>
</div>
@endsection

```

app/http/Controllers/RegisterController.php

```
<?php

namespace App\Http\Controllers\Auth;

use App\User;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Validator;
use Illuminate\Foundation\Auth\RegistersUsers;

class RegisterController extends Controller
{
    /*
    |--------------------------------------------------------------------------
    | Register Controller
    |--------------------------------------------------------------------------
    |
    | This controller handles the registration of new users as well as their
    | validation and creation. By default this controller uses a trait to
    | provide this functionality without requiring any additional code.
    |
    */

    use RegistersUsers;

    protected $redirectTo = '/notes';

    public function __construct()
    {
        $this->middleware('guest');
    }

    protected function validator(array $data)
    {
        return Validator::make($data, [
            'name' => 'required|string|max:255',
            'cpf' => 'required|string|max:11',
            'email' => 'required|string|email|max:255|unique:users',
            'password' => 'required|string|min:6|confirmed',
        ]);
    }
}
```

```
protected function create(array $data)
{
    return User::create([
        'name' => $data['name'],
        'cpf' => $data['cpf'],
        'email' => $data['email'],
        'password' => Hash::make($data['password']),
    ]);
}
```

app/user.php

```
<?php

namespace App;
use App\Notifications\ResetPassword;
use Illuminate\Notifications\Notifiable;
use Illuminate\Foundation\Auth\User as Authenticatable;

class User extends Authenticatable
{
    use Notifiable;

    protected $fillable = [
        'name', 'email', 'password', 'cpf'
    ];

    protected $hidden = [
        'password', 'remember_token',
    ];

    public function sendPasswordResetNotification($token)
    {
        $this->notify(new ResetPassword($token));
    }
}
```

```
php artisan make:migration add_cpf_to_users_table --table=users
```

```
C:\laragon\www\laravel-larast
λ php artisan make:migration add_cpf_to_users_table --table=users
Created Migration: 2019_05_02_042334_add_cpf_to_users_table

C:\laragon\www\laravel-larast
λ |
```

database/migrations/ 2019_05_02_042334_add_cpf_to_users_table.php

```
<?php

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

class AddCpfToUsersTable extends Migration
{
    public function up()
    {
        Schema::table('users', function (Blueprint $table) {
            $table->string('cpf')->after('name')->nullable();
        });
    }

    public function down()
    {
        Schema::table('users', function (Blueprint $table) {
            // ...
        });
    }
}
```

```
C:\laragon\www\laravel-larast
λ php artisan migrate
Migrating: 2019_05_02_042334_add_cpf_to_users_table
Migrated: 2019_05_02_042334_add_cpf_to_users_table

C:\laragon\www\laravel-larast
λ |
```

Showing rows 0 - 0 (1 total, Query took 0.0020 seconds.)

SELECT * FROM `users`

Show all Number of rows: 25 Filter rows: Search this table

+ Options

	id	name	cpf	email	password	remember_token
<input type="checkbox"/>	1	Roberto Pinheiro	NULL	betopinheiro1005@yahoo.com.br	\$2y\$10\$OrdqZ0XCqwBi0cIKUxwusOnpWPb5pBt2zc2iZSzfPp...	LoTzBbMvPhI3U7H2oV2xm6g6RXeTw...

Check all With selected: Edit Copy Delete Export

laravel-larast.test/register +

Minhas anotações Anotações Contato Login Register

Register

Name: Leonardo Levi Nascimento

CPF: 67752313505

E-Mail Address: leonardolevinascimento@gmail.com

Password: *****

Confirm Password: *****

Register

The screenshot shows a web browser window titled "Listagem de anotações". The URL is "laravel-larast.test/notes". The page has a header with "Minhas anotações", "Anotações", and "Contato" buttons, and a user profile "Leonardo Levi Nascimento". On the left, there's a sidebar titled "Grupos" with options: "--Nenhum--", "Família", and "Trabalho". The main content area displays a table with two rows of notes:

Anotação	Grupo	Editar	Eliminar
Comprar livro		Editar	Eliminar
Fazer compras		Editar	Eliminar

Showing rows 0 - 1 (2 total, Query took 0.0020 seconds.)

SELECT * FROM `users`						
		Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]				
<input type="checkbox"/> Show all	Number of rows:	25	Filter rows:	Search this table	Sort by key:	None
+ Options Edit Copy Delete 1 Roberto Pinheiro NULL betopinheiro1005@yahoo.com.br \$2y\$10\$OrdqZ0XCqwBl0cIKUxwusOn.pWPb5pBt2zc2iZSzfP... LoTzBbMvPhl3U7H...						
 Edit Copy Delete 2 Leonardo Levi Nascimento 67752313505 leonardolevinascimento@gmail.com \$2y\$10\$8e8WxjITVNd7RyZNmncLAOzdyK.OjOJoLIC1eK9UE1J... NULL						
<input type="checkbox"/> Check all With selected: Edit Copy Delete Export						

Aula 26 - O usuário com sessão iniciada

Esta aula mostrará como podemos obter e utilizar o modelo que corresponde ao usuário que tenha iniciado sessão. Isto permitirá que cada usuário manipule suas próprias anotações dentro da aplicação.

Criando o campo de usuário na tabela de anotações

```
php artisan make:migration add_user_to_notes_table --table=notes
```

```
C:\laragon\www\laravel-larast
λ php artisan make:migration add_user_to_notes_table --table=notes
Created Migration: 2019_05_02_073112_add_user_to_notes_table
      Area de transferencia          Par

C:\laragon\www\laravel-larast
λ |
```

```
php artisan migrate:fresh
```

```
2019_05_02_073112_add_user_to_notes_table.php
C:\laragon\www\laravel-larast
λ php artisan migrate:fresh
Dropped all tables successfully. Loss: AddUserToNotesTable
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_04_19_071011_create_notes_table
Migrated: 2019_04_19_071011_create_notes_table
Migrating: 2019_04_19_080304_add_user_id_column_to_notes_table
Migrated: 2019_04_19_080304_add_user_id_column_to_notes_table
Migrating: 2019_04_28_015155_create_groups_table
Migrated: 2019_04_28_015155_create_groups_table
Migrating: 2019_04_28_020913_add_group_id_to_note_tables
Migrated: 2019_04_28_020913_add_group_id_to_note_tables
Migrating: 2019_05_02_042334_add_cpf_to_users_table
Migrated: 2019_05_02_042334_add_cpf_to_users_table
Migrating: 2019_05_02_073112_add_user_to_notes_table
Migrated: 2019_05_02_073112_add_user_to_notes_table
      Schema::create('notes', function (Blueprint $table) {
        $table->id();
        $table->integer('user_id');
        $table->string('content');
        $table->timestamps();
      });

C:\laragon\www\laravel-larast
λ |
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	<code>id</code>	int(10)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	<code>group_id</code>	int(10)		UNSIGNED	Yes	NULL			Change Drop More
3	<code>user_id</code>	int(10)		UNSIGNED	No	None			Change Drop More
4	<code>title</code>	varchar(255)	utf8mb4_unicode_ci		No	None			Change Drop More
5	<code>body</code>	text	utf8mb4_unicode_ci		No	None			Change Drop More
6	<code>important</code>	tinyint(1)			No	0			Change Drop More
7	<code>created_at</code>	timestamp			Yes	NULL			Change Drop More
8	<code>updated_at</code>	timestamp			Yes	NULL			Change Drop More

↑ Check all With selected: Change Drop Primary Unique Index Fulltext

Print Propose table structure Move columns Normalize

Add 1 column(s) after updated_at

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
Edit Drop	PRIMARY	BTREE	Yes	No	id	0	A	No	
Edit Drop	notes_group_id_foreign	BTREE	No	No	group_id	0	A	Yes	
Edit Drop	notes_user_id_foreign	BTREE	No	No	user_id	0	A	No	

Registrando um usuário

laravel-larast.test/register

Minhas anotações Anotações Contato

Login Register

Register

Name: Roberto Pinheiro

CPF: 78557628099

E-Mail Address: betopinheiro1005@yahoo.com.br

Password: *****

Confirm Password: *****

Register

Minhas anotações. Todos os direitos reservados

Showing rows 0 - 0 (1 total, Query took 0.0010 seconds.)

```
SELECT * FROM `users`
```

Profiling [Edit inline](#) [Edit](#) [Explain SQL](#) [Create PHP code](#) [Refresh](#)

Show all Number of rows: 25 Filter rows: Search this table

+ Options id name cpf email password remember_token created_a

1 Roberto 78557628099 betopinheiro1005@yahoo.com.br \$2y\$10\$OhjqvDNPt1n4KbDnSthJORlqCWjtv12uTOjkFi0poF... NULL 2019-05-0:

↑ Check all With selected: Export

app/Http/Controllers/NotesController.php

```
<?php

namespace App\Http\Controllers;

use App\Note;
use App\Group;
use App\Http\Requests\NotesRequest;
use Illuminate\Http\Request;

class NotesController extends Controller
{
    public function index(Group $group){
        $groups = Group::all();
        $notes = $group->notes;
        return view('notes.index', compact('notes', 'groups'));
    }

    public function create(){
        $groups = Group::pluck('name', 'id');
        return view('notes.create', compact('groups'));
    }

    public function show(Note $note){
        return view('notes.show', [ 'note' => $note ]);
    }

    public function store(NotesRequest $request){
        $data = request()->all();
        $data['user_id'] = \Auth::user()->id;
        Note::create($data);
        return redirect('/notes');
    }

    public function edit(Note $note){
        $groups = Group::pluck('name', 'id');
        return view('notes.edit', compact('note', 'groups'));
    }

    public function update(Note $note, NotesRequest $request){
        $note->update( $request->all() );
        return redirect('/notes');
    }

    public function destroy(Note $note){
        $note->delete();
        return redirect('/notes');
    }
}
```

app/Note.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Note extends Model
{
    // protected $table = 'anotacoes';

    protected $fillable = [
        'title', 'body', 'important', 'group_id', 'user_id'
    ];

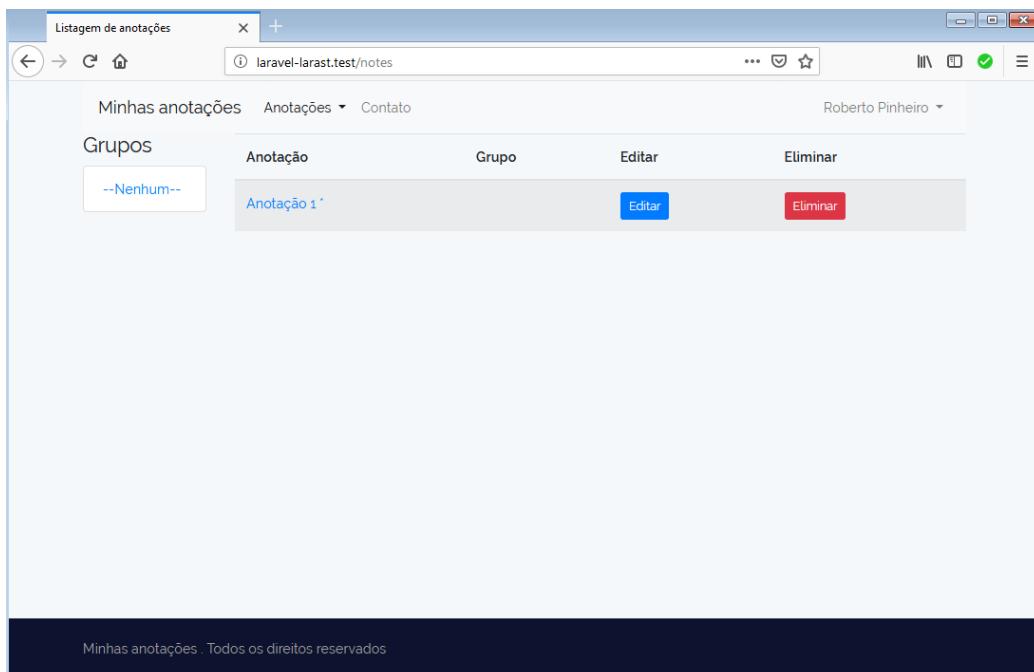
    public function isImportant()
    {
        return $this->important == 1;
    }

    public function group(){
        return $this->belongsTo(Group::class); // group_id
    }

    public function scopeWithoutGroup($query){
        return $query->whereNull('group_id');
    }
}
```

Cadastrando uma anotação

The screenshot shows a web browser window with the title 'Criar uma nova anotação'. The URL in the address bar is 'laravel-larast.test/notes/create'. The page content is a form for creating a new note. It includes fields for 'Título' (Title) containing 'Anotação 1', 'Anotação' (Content) containing 'Descrição da anotação 1', 'Grupo' (Group), and a checked checkbox for 'É importante' (Is important). A blue 'Criar' (Create) button is at the bottom. The top navigation bar shows 'Minhas anotações', 'Anotações', and 'Contato'. The user 'Roberto Pinheiro' is logged in.



Showing rows 0 - 0 (1 total, Query took 0.0020 seconds.)									
SELECT * FROM `notes`									
<input type="checkbox"/> Profiling [Edit inline] [Edit] [Explain SQL									
<input type="checkbox"/> Show all Number of rows: 25 ▾ Filter rows: Search this table									
+ Options	id	group_id	user_id	title	body	important	created_at	updated_at	
<input type="checkbox"/>	1	NULL	1	Anotação 1	Descrição da anotação 1	1	2019-05-02 08:34:07	2019-05-02 08:34:07	
	<input type="checkbox"/> Check all	With selected:	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	<input type="checkbox"/> Export			

Outra forma

app/Http/Controllers/NotesController.php

```
<?php

namespace App\Http\Controllers;

use App>Note;
use App\Group;
use App\Http\Requests\NotesRequest;
use Illuminate\Http\Request;

class NotesController extends Controller
{
    public function index(Group $group){
        $groups = Group::all();
        $notes = $group->notes;
        return view('notes.index', compact('notes', 'groups'));
    }
}
```

```

public function create(){
    $groups = Group::pluck('name', 'id');
    return view('notes.create', compact('groups'));
}

public function show(Note $note){
    return view('notes.show', [ 'note' => $note ]);
}

public function store(NotesRequest $request){

// $data = request()->all();
// $data['user_id'] = \Auth::user()->id;
// Note::create($data);

$note = new Note(request()->all());
$note->user_id = \Auth::user()->id;
$note->save();

return redirect('/notes');
}

public function edit(Note $note){
    $groups = Group::pluck('name', 'id');
    return view('notes.edit', compact('note', 'groups'));
}

public function update(Note $note, NotesRequest $request){
    $note->update( $request->all() );
    return redirect('/notes');
}

public function destroy(Note $note){
    $note->delete();
    return redirect('/notes');
}
}

```

app/Note.php

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Note extends Model
{
    // protected $table = 'anotacoes';

    protected $fillable = [
        'title', 'body', 'important', 'group_id'
    ];
}

```

```

public function isImportant()
{
    return $this->important == 1;
}

public function group(){
    return $this->belongsTo(Group::class); // group_id
}

public function scopeWithoutGroup($query){
    return $query->whereNull('group_id');
}

}

```

Screenshot of a web application showing a list of notes. The interface includes a header with a back button, a search bar containing 'laravel-larast.test/notes', and a user dropdown for 'Roberto Pinheiro'. Below the header is a navigation bar with 'Minhas anotações', 'Anotações', and 'Contato'.

Grupos	Anotação	Grupo	Editar	Eliminar
--Nenhum--	Anotação 2		Editar	Eliminar
	Anotação 1 *		Editar	Eliminar

At the bottom of the page, there is a footer bar with the text 'Minhas anotações . Todos os direitos reservados'.

MySQL query results for the 'notes' table:

```

Showing rows 0 - 1 (2 total, Query took 0.0020 seconds.)

SELECT * FROM `notes`

```

Table structure and data:

+ Options	id	group_id	user_id	title	body	important	created_at	updated_at
<input type="checkbox"/>	1	NULL	1	Anotação 1	Descrição da anotação 1	1	2019-05-02 08:34:07	2019-05-02 08:34:07
<input type="checkbox"/>	2	NULL	1	Anotação 2	Descrição da anotação 2	0	2019-05-02 08:46:01	2019-05-02 08:46:01

Action buttons at the bottom include: Check all, With selected: Edit, Copy, Delete, Export.

Outra forma

app/Http/Controllers/NotesController.php

```
<?php

namespace App\Http\Controllers;

use App\Note;
use App\Group;
use App\Http\Requests\NotesRequest;
use Illuminate\Http\Request;

class NotesController extends Controller
{
    public function index(Group $group){
        $groups = Group::all();
        $notes = $group->notes;
        return view('notes.index', compact('notes', 'groups'));
    }

    public function create(){
        $groups = Group::pluck('name', 'id');
        return view('notes.create', compact('groups'));
    }

    public function show(Note $note){
        return view('notes.show', [ 'note' => $note ]);
    }

    public function store(NotesRequest $request){

        // $data = request()->all();
        // $data['user_id'] = \Auth::user()->id;
        // Note::create($data);

        // $note = new Note(request()->all());
        // $note->user_id = \Auth::user()->id;
        // $note->save();

        $note = new Note(request()->all());
        \Auth::user()->notes()->save($note);

        return redirect('/notes');
    }

    public function edit(Note $note){
        $groups = Group::pluck('name', 'id');
        return view('notes.edit', compact('note', 'groups'));
    }

    public function update(Note $note, NotesRequest $request){
        $note->update( $request->all() );
        return redirect('/notes');
    }
}
```

```

public function destroy(Note $note){
    $note->delete();
    return redirect('/notes');
}

```

Criar uma nova anotação

Minhas anotações Anotações Contato Roberto Pinheiro

Criar nova anotação

Título
Anotação 3

Anotação
Descrição da anotação 3

Grupo
--Nenhum--

É importante]

Criar

Minhas anotações . Todos os direitos reservados

Listagem de anotações

Minhas anotações Anotações Contato Roberto Pinheiro

Grupos	Anotação	Grupo	Editar	Eliminar
--Nenhum--	Anotação 3 *		Editar	Eliminar
	Anotação 2		Editar	Eliminar
	Anotação 1 *		Editar	Eliminar

Minhas anotações . Todos os direitos reservados

Registrando um novo usuário

A screenshot of a web browser window titled "laravel-larast.test/register". The address bar shows the URL "laravel-larast.test/register". The page content is a registration form titled "Register". The form fields are:

Name	Ricardo Souza
CPF	44932324049
E-Mail Address	ricardosouza@gmail.com
Password	*****
Confirm Password	*****

Below the form is a blue "Register" button. At the top right of the page, there are "Login" and "Register" links. The bottom of the page has a dark footer bar with the text "Minhas anotações . Todos os direitos reservados".

A screenshot of a web browser window titled "Listagem de anotações". The address bar shows the URL "laravel-larast.test/notes". The page content is a table listing notes. The table has columns: Grupos, Anotação, Grupo, Editar, and Eliminar. The "Grupos" column shows a single entry: "--Nenhum--". The "Anotação" column is empty. The "Grupo" column is also empty. There are "Editar" and "Eliminar" buttons for each row. At the top right, there is a dropdown menu showing "Ricardo Souza". The bottom of the page has a dark footer bar with the text "Minhas anotações . Todos os direitos reservados".

Grupos	Anotação	Grupo	Editar	Eliminar
--Nenhum--				

Criar uma nova anotação

Minhas anotações Anotações Contato Ricardo Souza

Criar nova anotação

Titulo

Anotação

Grupo

É importante]

Criar

Minhas anotações . Todos os direitos reservados

Insira manualmente na tabela groups:

Família
Trabalho

```
✓ 1 row affected.

UPDATE `groups` SET `created_at` = CURRENT_TIME(), `updated_at` = CURRENT_TIME() WHERE `groups`.

✓ Showing rows 0 - 1 (2 total, Query took 0.0010 seconds.)

SELECT * FROM `groups`

>Show all | Number of rows: 25 | Filter rows: Search this table

+ Options
← → ▾ id name created_at updated_at
  Edit  Copy  Delete 1 Família 2019-05-02 06:23:46 2019-05-02 06:23:46
  Edit  Copy  Delete 2 Trabalho 2019-05-02 06:23:20 2019-05-02 06:23:20

 Check all With selected:  Edit  Copy  Delete  Export
```

resources/views/layouts/app.blade.php

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>@yield('title')</title>
    <link rel="stylesheet" href="/css/app.css">
</head>
<body>

    <div class="container">
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <a class="navbar-brand" href="/">{{ config('app.name') }}</a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                @if(Auth::check())
                    <ul class="navbar-nav">
                        <li class="nav-item dropdown {{ Request::is('notes') ? 'active' : '' }}>
                            <a class="nav-link dropdown-toggle" href="#" id="navbardrop" data-toggle="dropdown">
                                Anotações
                            </a>
                            <div class="dropdown-menu">
                                <a class="dropdown-item" href="/notes">Ver</a>
                                <a class="dropdown-item" href="/notes/create">Criar</a>
                            </div>
                        </li>
                        <li class="{{ Request::is('contact') ? 'active' : '' }}>
                            <a class="nav-link" href="/contact">Contato</a>
                        </li>
                    </ul>
                @endif

                <!-- Right Side Of Navbar -->
                <ul class="navbar-nav ml-auto">
                    <!-- Authentication Links -->
                    @guest
                        <li class="nav-item">
                            <a class="nav-link" href="{{ route('login') }}>{{ __('Login') }}</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link" href="{{ route('register') }}>{{ __('Register') }}</a>
                        </li>
                    @else
                        <li class="nav-item dropdown">
                            <a id="navbarDropdown" class="nav-link dropdown-toggle" href="#" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false" v-pre>
                                {{ Auth::user()->name }} <span class="caret"></span>
                            </a>
                        </li>
                    @endguest
                </ul>
            </div>
        </nav>
    </div>

```

```
<div class="dropdown-menu dropdown-menu-right" aria-
labelledby="navbarDropdown">
    <a class="dropdown-item" href="{{ route('logout') }}"
        onclick="event.preventDefault();
                  document.getElementById('logout-form').submit();">
        {{ __('Logout') }}
    </a>

    <form id="logout-form" action="{{ route('logout') }}"
method="POST" style="display: none;">
        @csrf
    </form>
</div>
</li>
@endguest
</ul>

</div>
</nav>
</div>

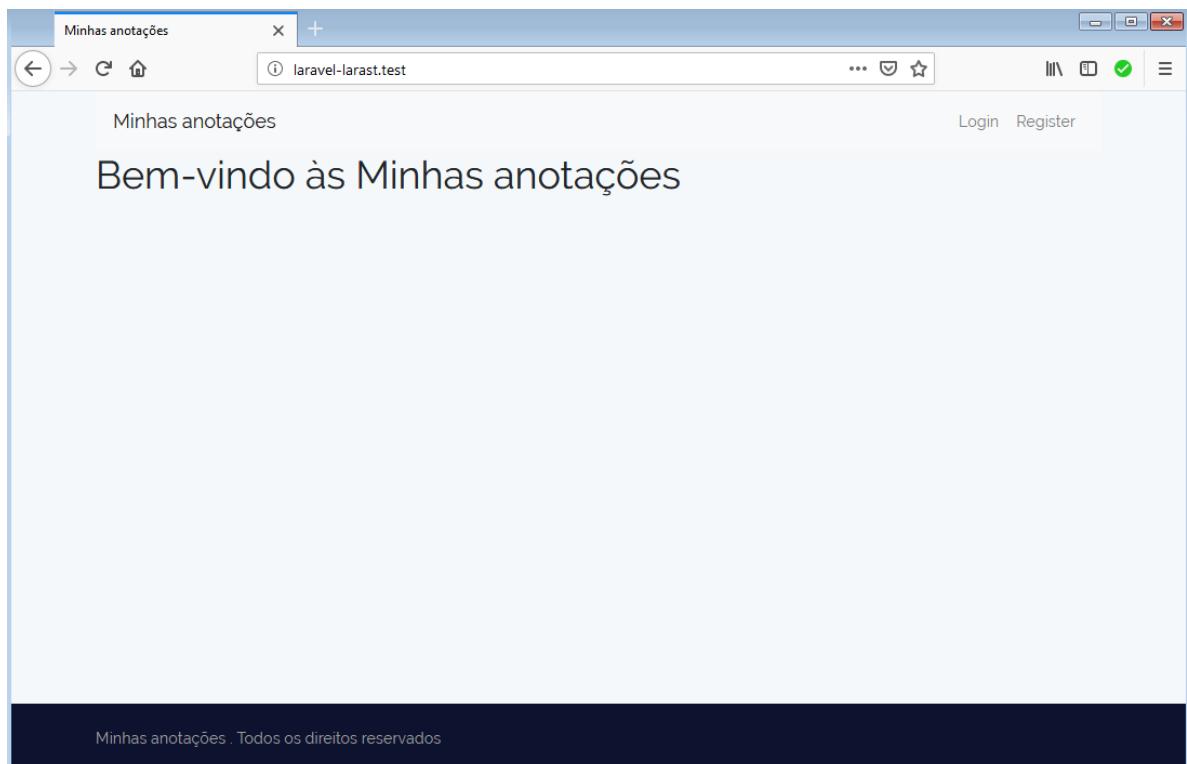
<div class="container">
{{-- <ul>
    <li>{{ Auth::user() -> name }}</li>
    <li>{{ Auth::user() -> email }}</li>
</ul>
--}} @yield('content')
</div>

<footer>
    <div class="container">
        {{ config('app.name') }} . Todos os direitos reservados
    </div>
</footer>

<script src="/js/app.js"></script>

@yield('scripts')

</body>
</html>
```



The screenshot shows a web browser window titled "Listagem de anotações". The address bar contains "laravel-larast.test/notes". The page content includes a header with "Minhas anotações", "Anotações", "Contato", and a user profile "Roberto Pinheiro". On the left, there is a sidebar titled "Grupos" with options "--Nenhum--", "Familia", and "Trabalho". The main area displays a table with three rows of notes:

Anotação	Grupo	Editar	Eliminar
Anotação 1 *		Editar	Eliminar
Anotação 2		Editar	Eliminar
Anotação 3 *		Editar	Eliminar

At the bottom, a dark footer bar displays "Minhas anotações . Todos os direitos reservados".

app/Controllers/NotesWithoutGroupController.php

```
<?php

namespace App\Http\Controllers;

use App\Note;
use App\Group;
use Illuminate\Http\Request;

class NotesWithoutGroupController extends Controller
{

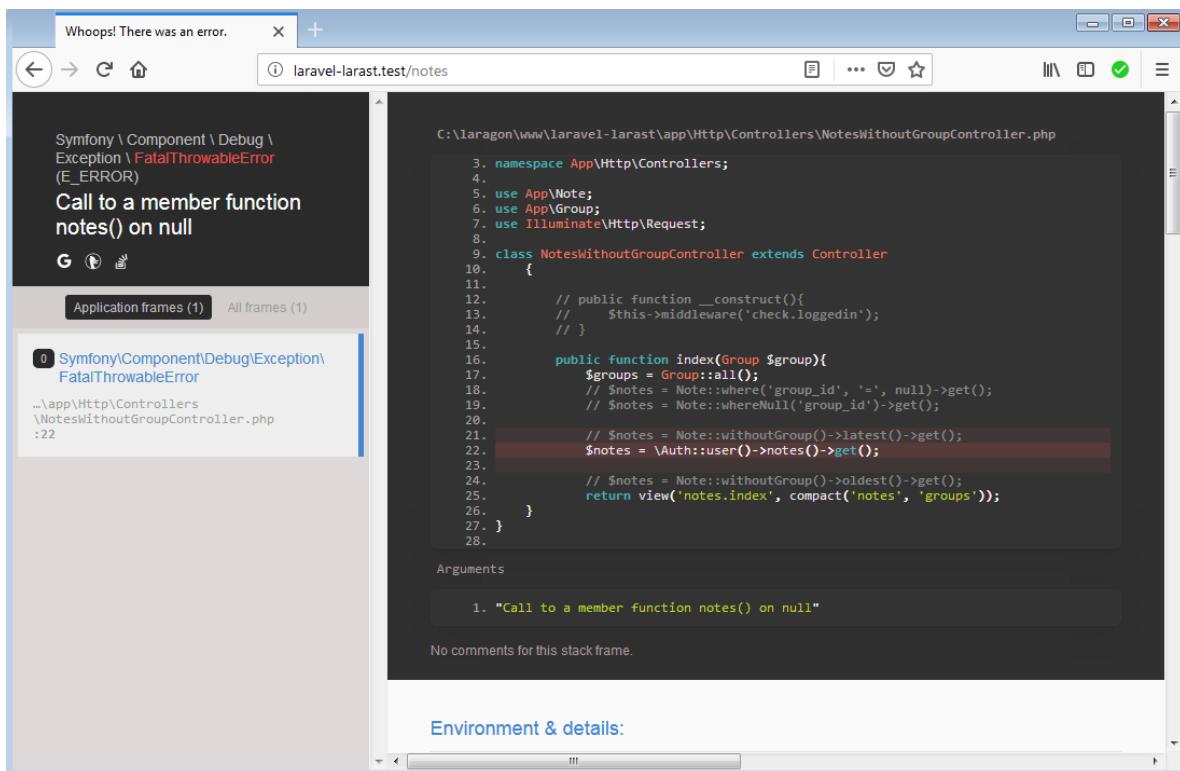
    public function index(Group $group){
        $groups = Group::all();
        $notes = \Auth::user()->notes()->withoutGroup()->get();

        return view('notes.index', compact('notes', 'groups'));
    }
}
```

Aula 27 - Middlewares

Às vezes necessitamos proteger certas rotas em nossa aplicação para que estejam disponíveis unicamente para certos usuários. Os middlewares nos ajudam nessa tarefa e são sumamente úteis quando queremos escrever lógica e comprovações antes de sequer tocar os controladores.

<http://laravel-larast.test/notes>



Criando um middleware

`php artisan make:middleware CheckIfLoggedIn`

```
C:\laragon\www\laravel-larast
λ php artisan make:middleware CheckIfLoggedIn
Middleware created successfully.

C:\laragon\www\laravel-larast
λ |
```

app/Middleware/CheckIfLoggedIn.php

```
<?php

namespace App\Http\Middleware;

use Closure;

class CheckIfLoggedIn
{
    public function handle($request, Closure $next)
    {
        if(\Auth::guest()){
            return redirect('/login');
        }
        return $next($request);
    }
}
```

app/Controllers/Kernel.php

```
<?php

namespace App\Http;

use Illuminate\Foundation\Http\Kernel as HttpKernel;

class Kernel extends HttpKernel
{
    protected $middleware = [
        \Illuminate\Foundation\Http\Middleware\CheckForMaintenanceMode::class,
        \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,
        \App\Http\Middleware\TrimStrings::class,
        \Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,
        \App\Http\Middleware\TrustProxies::class,
    ];

    protected $middlewareGroups = [
        'web' => [
            \App\Http\Middleware\EncryptCookies::class,
            \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
            \Illuminate\Session\Middleware\StartSession::class,
            // \Illuminate\Session\Middleware\AuthenticateSession::class,
            \Illuminate\View\Middleware\ShareErrorsFromSession::class,
            \App\Http\Middleware\VerifyCsrfToken::class,
            \Illuminate\Routing\Middleware\SubstituteBindings::class,
        ],
    ];
}
```

```

'api' => [
    'throttle:60,1',
    'bindings',
],
];

protected $routeMiddleware = [
    'auth' => \Illuminate\Auth\Middleware\Authenticate::class,
    'auth.basic' =>
\Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
    'bindings' => \Illuminate\Routing\Middleware\SubstituteBindings::class,
    'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
    'can' => \Illuminate\Auth\Middleware\Authorize::class,
    'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
    'signed' => \Illuminate\Routing\Middleware\ValidateSignature::class,
    'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
    'check.loggedin' => \App\Http\Middleware\CheckIfLoggedIn::class,
];
}

```

app/Http/Controllers/NotesWithoutGroupController.php

```

<?php

namespace App\Http\Controllers;

use App\Note;
use App\Group;
use Illuminate\Http\Request;

class NotesWithoutGroupController extends Controller
{

    public function __construct(){
        $this->middleware('check.loggedin');
    }

    public function index(Group $group){
        $groups = Group::all();
        // $notes = Note::where('group_id', '=', null)->get();
        // $notes = Note::whereNull('group_id')->get();

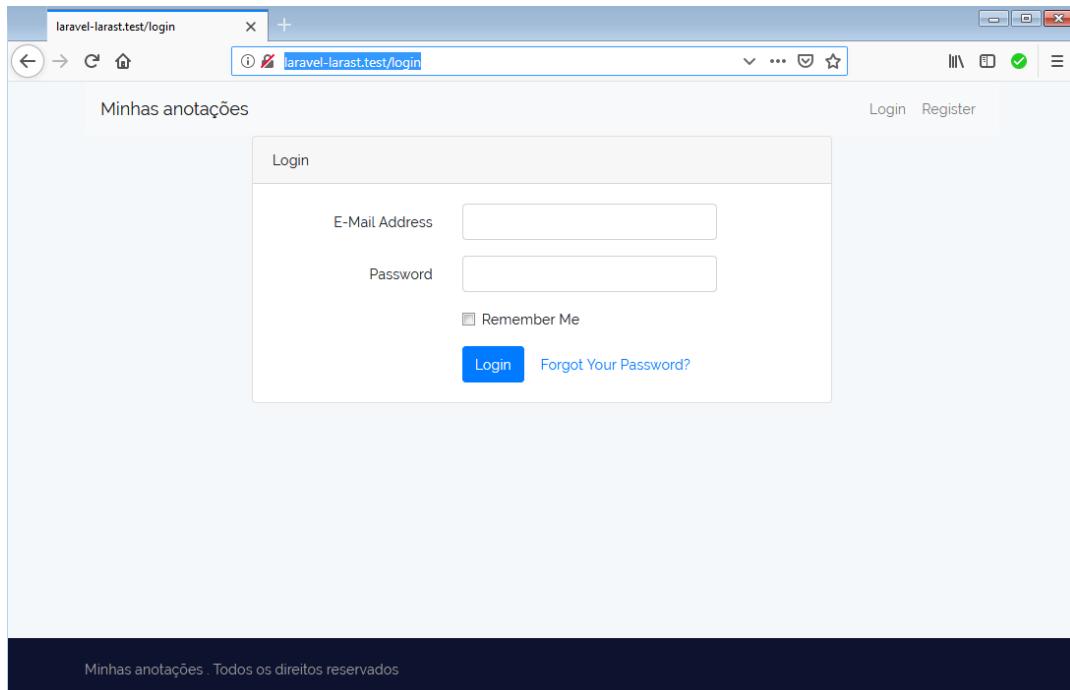
        // $notes = Note::withoutGroup()->latest()->get();
        $notes = \Auth::user()->notes()->get();

        // $notes = Note::withoutGroup()->oldest()->get();
        return view('notes.index', compact('notes', 'groups'));
    }
}

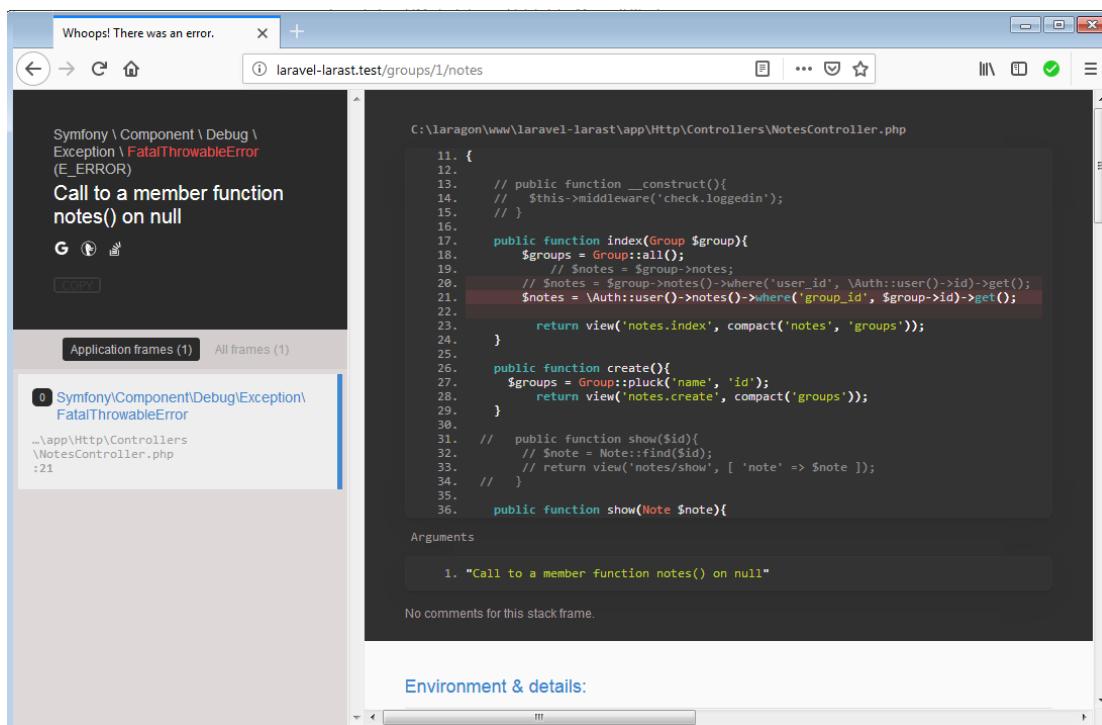
```

Se não estiver logado, ao executar:
`http://laravel-larast.test/notes`

será redirecionado para a tela de login:



`laravel-larast.test/groups/1/notes`



app/Http/Controllers/NotesController.php

```
<?php

namespace App\Http\Controllers;

use App\Note;
use App\Group;
use App\Http\Requests\NotesRequest;
use Illuminate\Http\Request;

class NotesController extends Controller
{

    public function __construct(){
        $this->middleware('check.loggedin');
    }

    public function index(Group $group){
        $groups = Group::all();
        $notes = \Auth::user()->notes()->where('group_id', $group->id)->get();
        return view('notes.index', compact('notes', 'groups'));
    }

    public function create(){
        $groups = Group::pluck('name', 'id');
        return view('notes.create', compact('groups'));
    }

    public function show(Note $note){
        return view('notes.show', [ 'note' => $note ]);
    }

    public function store(NotesRequest $request){
        $note = new Note(request()->all());
        \Auth::user()->notes()->save($note);

        return redirect('/notes');
    }

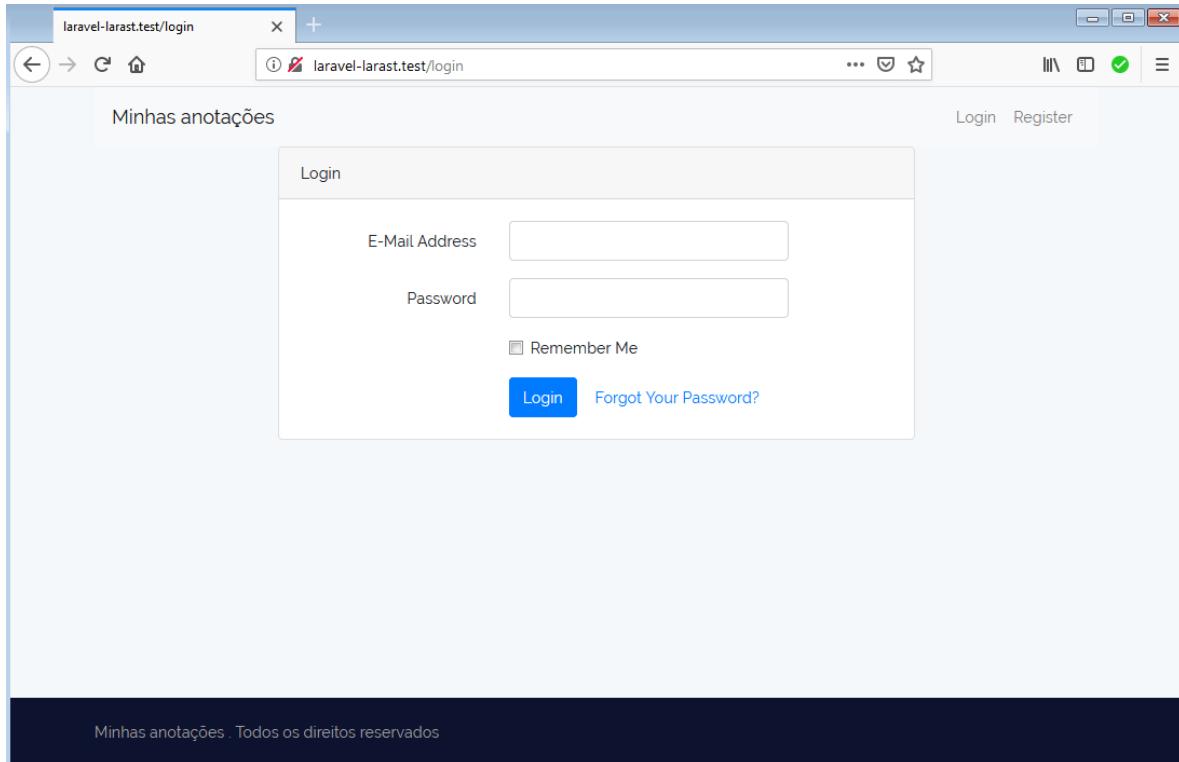
    public function edit(Note $note){
        $groups = Group::pluck('name', 'id');
        return view('notes.edit', compact('note', 'groups'));
    }

    public function update(Note $note, NotesRequest $request){
        $note->update( $request->all() );
        return redirect('/notes');
    }

    public function destroy(Note $note){
        $note->delete();
        return redirect('/notes');
    }
}
```

Se não estiver logado, ao executar:
laravel-larast.test/groups/1/notes

será redirecionado para a tela de login:



Outra forma melhor

Remova o construtor dos dois controladores e altere o arquivo de rotas:

routes/web.php

```
<?php

Route::group(['middleware' => 'check.loggedin'], function(){
    Route::get('/notes', 'NotesWithoutGroupController@index');
    Route::get('/groups/{group}/notes', 'NotesController@index');

    Route::get('/notes/create', 'NotesController@create');
    // Route::get('/notes/{id}', 'NotesController@show');
    Route::get('/notes/{note}', 'NotesController@show');
    Route::get('/notes/{note}/edit', 'NotesController@edit');
    Route::patch('/notes/{note}', 'NotesController@update');
    Route::post('/notes', 'NotesController@store');
    Route::delete('/notes/{note}', 'NotesController@destroy');
});
```

```

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

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

Auth::routes();

Route::get('/home', 'HomeController@index')->name('home');

```

Usando o middleware do próprio Laravel

Remova a linha inserida no arquivo **app/Controllers/Kernel.php**

```

protected $routeMiddleware = [
    'auth' => \Illuminate\Auth\Middleware\Authenticate::class,
    'auth.basic' =>
\Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
    'bindings' => \Illuminate\Routing\Middleware\SubstituteBindings::class,
    'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
    'can' => \Illuminate\Auth\Middleware\Authorize::class,
    'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
    'signed' => \Illuminate\Routing\Middleware\ValidateSignature::class,
    'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
    // 'check.loggedin' => \App\Http\Middleware\CheckIfLoggedIn::class,
];

```

routes/web.php

```

<?php

Route::group(['middleware' => 'auth'], function(){
    Route::get('/notes', 'NotesWithoutGroupController@index');
    Route::get('/groups/{group}/notes', 'NotesController@index');

    Route::get('/notes/create', 'NotesController@create');
    // Route::get('/notes/{id}', 'NotesController@show');
    Route::get('/notes/{note}', 'NotesController@show');
    Route::get('/notes/{note}/edit', 'NotesController@edit');
    Route::patch('/notes/{note}', 'NotesController@update');
    Route::post('/notes', 'NotesController@store');
    Route::delete('/notes/{note}', 'NotesController@destroy');
});

```

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

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

Auth::routes();

Route::get('/home', 'HomeController@index')->name('home');
```

Middleware para checar se existe assinatura paga

Se o usuário tem assinatura paga, pode acessar o aplicativo, caso contrário, não.

Incluindo o campo paid na tabela de usuários

```
php artisan make:migration add_paid_field_to_users_table --table=users
```

```
C:\laragon\www\laravel-larast
λ php artisan make:migration add_paid_field_to_users_table --table=users
Created Migration: 2019_05_03_024213_add_paid_field_to_users_table
Parâmetro
```



```
C:\laragon\www\laravel-larast
λ |
```

database/migrations/2019_05_03_024213_add_paid_field_to_users_table.php

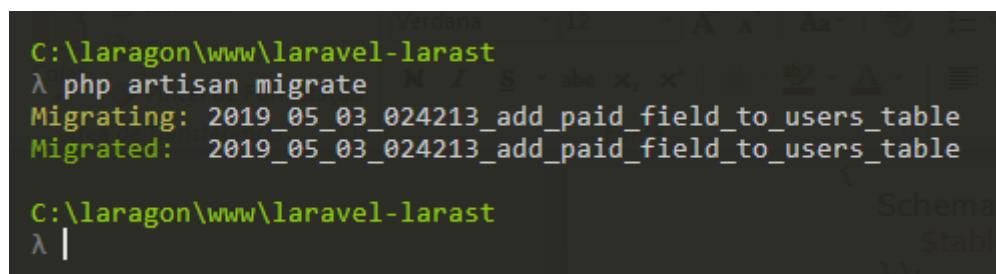
```
<?php

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

class AddPaidFieldToUsersTable extends Migration
{
    public function up()
    {
        Schema::table('users', function (Blueprint $table) {
            $table->integer('paid')->after('cpf');
        });
    }

    public function down()
    {
        Schema::table('users', function (Blueprint $table) {
            // ...
        });
    }
}
```

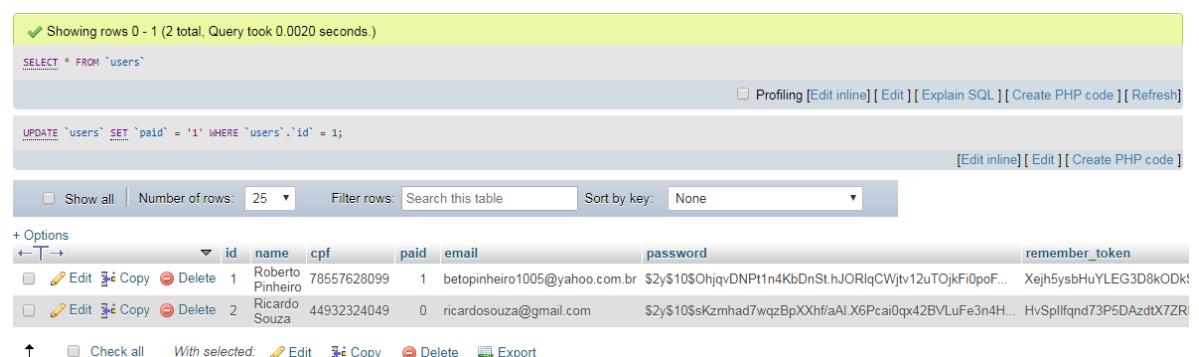
php artisan migrate



```
C:\laragon\www\laravel-larast
λ php artisan migrate
Migrating: 2019_05_03_024213_add_paid_field_to_users_table
Migrated: 2019_05_03_024213_add_paid_field_to_users_table

C:\laragon\www\laravel-larast
λ |
```

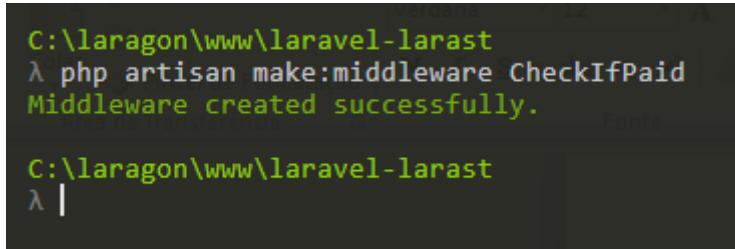
No campo paid deixe 1 para o usuário de id 1 e 0 para o usuário de id 2.



	id	name	cpf	paid	email	password	remember_token
<input type="checkbox"/>	1	Roberto Pinheiro	78557628099	1	betopinheiro1005@yahoo.com.br	\$2y\$10\$OhjqvDNPt1n4KbDnSt.hJORlqCW/tv12uT0jkFj0poF...	Xejh5ysbHuYLEG3D8kODk...
<input type="checkbox"/>	2	Ricardo Souza	44932324049	0	ricardosouza@gmail.com	\$2y\$10\$sKzmhad7wqzBpXXhf/aAI/X6Pca10qx42BV/LuFe3nH...	HvSpIlfqnd73P5DAzdtX7ZR...

Criando o middleware

```
php artisan make:middleware CheckIfPaid
```



```
C:\laragon\www\laravel-larast
λ php artisan make:middleware CheckIfPaid
Middleware created successfully.

C:\laragon\www\laravel-larast
λ |
```

app/Middleware/CheckIfPaid.php

```
<?php

namespace App\Http\Middleware;

use Closure;

class CheckIfPaid
{
    public function handle($request, Closure $next)
    {
        if(\Auth::user()->paid == 0){
            return redirect('/subscription-cancelled');
        }
        return $next($request);
    }
}
```

app/Controllers/Kernel.php

```
protected $routeMiddleware = [
    'auth' => \Illuminate\Auth\Middleware\Authenticate::class,
    'auth.basic' =>
\Illuminate\Auth\Middleware\AuthenticateWithBasicAuth::class,
    'bindings' => \Illuminate\Routing\Middleware\SubstituteBindings::class,
    'cache.headers' => \Illuminate\Http\Middleware\SetCacheHeaders::class,
    'can' => \Illuminate\Auth\Middleware\Authorize::class,
    'guest' => \App\Http\Middleware\RedirectIfAuthenticated::class,
    'signed' => \Illuminate\Routing\Middleware\ValidateSignature::class,
    'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
    'check.paid' => \App\Http\Middleware\CheckIfPaid::class,
];
```

routes/web.php

```
<?php

Route::group(['middleware' => ['auth', 'check.paid']], function(){
    Route::get('/notes', 'NotesWithoutGroupController@index');
    Route::get('/groups/{group}/notes', 'NotesController@index');

    Route::get('/notes/create', 'NotesController@create');
    // Route::get('/notes/{id}', 'NotesController@show');
    Route::get('/notes/{note}', 'NotesController@show');
    Route::get('/notes/{note}/edit', 'NotesController@edit');
    Route::patch('/notes/{note}', 'NotesController@update');
    Route::post('/notes', 'NotesController@store');
    Route::delete('/notes/{note}', 'NotesController@destroy');
});

Route::get('/subscription-cancelled', function () {
    return view('bloqueado');
});

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

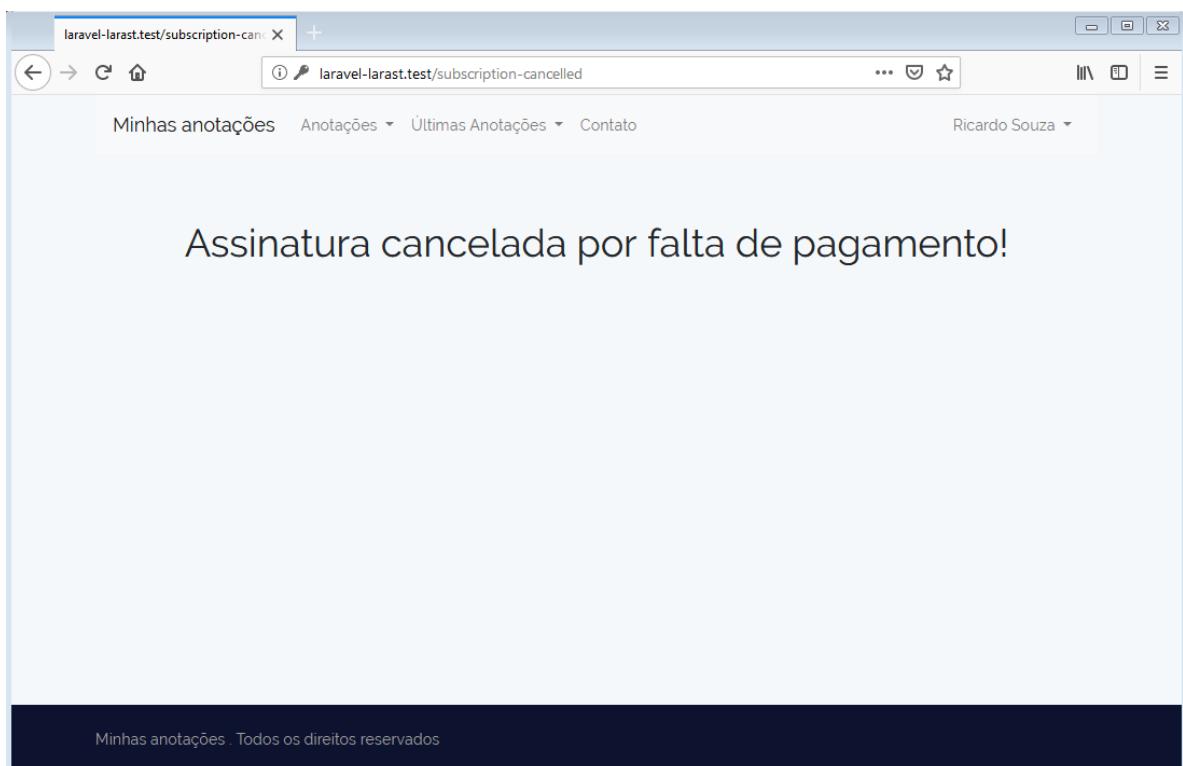
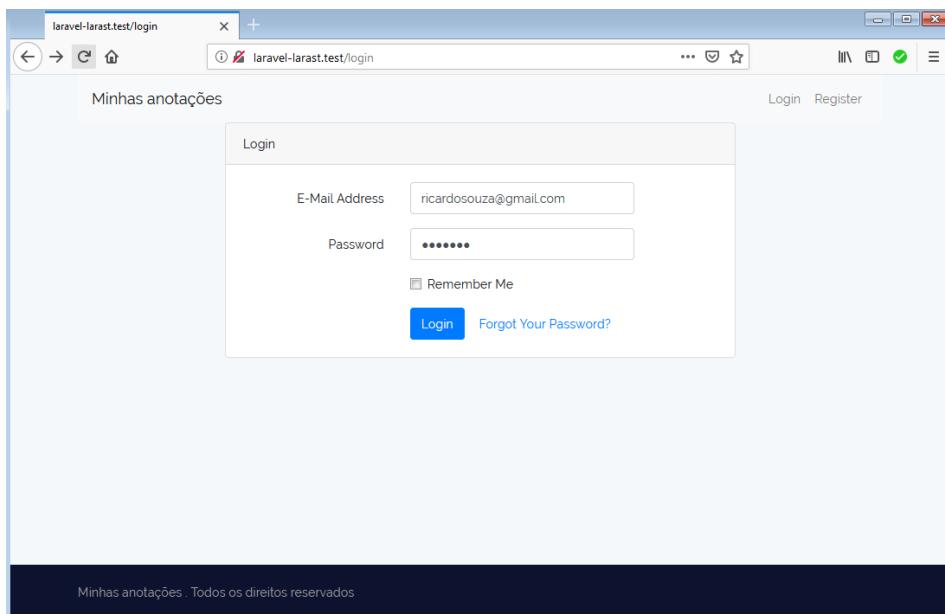
Route::get('/contact', function () {
    return view('contact');
});

Auth::routes();

Route::get('/home', 'HomeController@index')->name('home');
```

resources/views/bloqueado.blade.php

```
@include('layouts.app')
<h1 class="text-center" style="margin-top:60px;">Assinatura cancelada
por falta de pagamento!</h1>
```



Laravel-larast.test/login

Minhas anotações

Login

E-Mail Address betopinheiro1005@yahoo.com.br

Password *****

Remember Me

[Login](#) [Forgot Your Password?](#)

Minhas anotações . Todos os direitos reservados

Listagem de anotações

Minhas anotações Anotações Contato Roberto Pinheiro

Grupos

Anotação	Grupo	Editar	Eliminar
Anotação 1 *		Editar	Eliminar
Anotação 2		Editar	Eliminar
Anotação 3 *		Editar	Eliminar

Minhas anotações . Todos os direitos reservados

laravel-larast.test

Na tabela users, altere o campo paid para o valor 1 para o usuário Ricardo Souza (id 2).

Aula 28 - View Composers

Alguma vez você esteve na situação de necessitar a mesma informação em todas as views? Por exemplo um item de menu que deve mostrar a mesma informação em todas e cada uma das páginas. Qual é a melhor maneira de manejar esta situação? Repetimos o mesmo código em todos e cada um dos controladores? Vejamos como os view composers nos são de grande utilidade em situações como esta.

resources/views/layouts/app.blade.php

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>@yield('title')</title>
    <link rel="stylesheet" href="/css/app.css">
</head>
<body>

    <div class="container">
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <a class="navbar-brand" href="/">{{ config('app.name') }}</a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                @if(Auth::check())
                    <ul class="navbar-nav">
                        <li class="nav-item dropdown {{ Request::is('notes') ? 'active' : '' }}>
                            <a class="nav-link dropdown-toggle" href="#" id="navbardrop" data-toggle="dropdown">
                                Anotações
                            </a>
                            <div class="dropdown-menu">
                                <a class="dropdown-item" href="/notes">Ver</a>
                                <a class="dropdown-item" href="/notes/create">Criar</a>
                            </div>
                        </li>
                        <li class="dropdown">
                            <a class="nav-link dropdown-toggle" href="#" data-toggle="dropdown">
                                Últimas Anotações
                            </a>
                            <ul class="dropdown-menu">
                                @foreach ($lastNotes as $note)
                                    <li class="list-group-item"><a href="/notes/{{ $note->id }}>{{ $note->title }}</a></li>
                                @endforeach
                            </ul>
                        </li>
                @endif
            </div>
        </nav>
    </div>
</body>
```

```

<li class="{{ Request::is('contact') ? 'active' : ''}}>
    <a class="nav-link" href="/contact">Contato</a>
</li>
</ul>
@endif

<!-- Right Side Of Navbar -->
<ul class="navbar-nav ml-auto">
    <!-- Authentication Links -->
    @guest
        <li class="nav-item">
            <a class="nav-link" href="{{ route('login') }}>{{ __('Login') }}</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="{{ route('register') }}>{{ __('Register') }}</a>
        </li>
    @else
        <li class="nav-item dropdown">
            <a id="navbarDropdown" class="nav-link dropdown-toggle" href="#" role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false" v-pre>
                {{ Auth::user()->name }} <span class="caret"></span>
            </a>

            <div class="dropdown-menu dropdown-menu-right" aria-labelledby="navbarDropdown">
                <a class="dropdown-item" href="{{ route('logout') }}"
                    onclick="event.preventDefault();
                            document.getElementById('logout-form').submit();">
                    {{ __('Logout') }}</a>
            <form id="logout-form" action="{{ route('logout') }}"
                method="POST" style="display: none;">
                @csrf
                </form>
            </div>
        </li>
    @endguest
</ul>

</div>
</nav>
</div>

<div class="container">
{{--    <ul>
        <li>{{ Auth::user() -> name }}</li>
        <li>{{ Auth::user() -> email }}</li>
    </ul>
--}}    @yield('content')
</div>

```

```

<footer>
    <div class="container">
        {{ config('app.name') }} . Todos os direitos reservados
    </div>
</footer>

<script src="/js/app.js"></script>

@yield('scripts')

</body>
</html>

```

app/Providers/AppServiceProvider.php

```

<?php

namespace App\Providers;

use Illuminate\Support\ServiceProvider;

class AppServiceProvider extends ServiceProvider
{
    public function boot()
    {
        \View::composer('layouts.app', function($view){
            if(\Auth::check()){
                $lastNotes = \Auth::user()->notes()->orderBy('created_at', 'desc')->take(2)->get();
                $view->with('lastNotes', $lastNotes);
            }
        });
    }

    public function register()
    {
        //
    }
}

```

Listagem de anotações

laravel-larast.test/notes

Minhas anotações Anotações Últimas Anotações Contato Roberto Pinheiro

Grupos

Anotação	Grupo	Editar	Eliminar
Frutas *		Editar	Eliminar

--Nenhum--

Família

Trabalho

Minhas anotações . Todos os direitos reservados

Anotação: Frutas

laravel-larast.test/notes/6

Minhas anotações Anotações Últimas Anotações Contato Roberto Pinheiro

Frutas

- Macã
- Banana
- Melancia
- Pera
- Uva
- Laranja
- Pêssego
- Goiaba
- Melão
- Mamão
- Limão
- Cereja
- Morango
- Abacate
- Amora
- Ameixa

Minhas anotações . Todos os direitos reservados

Listagem de anotações

Minhas anotações Anotações Últimas Anotações Contato Roberto Pinheiro

Grupos

Anotação	Grupo	Editar	Eliminar
Anotação 1 *	Família	Editar	Eliminar
Anotação 2	Família	Editar	Eliminar

Minhas anotações . Todos os direitos reservados

This screenshot shows a web application interface for managing notes. On the left, there's a sidebar titled 'Grupos' with options like '--Nenhum--', 'Família', and 'Trabalho'. The main area displays two notes: 'Anotação 1' and 'Anotação 2', both associated with the 'Família' group. Each note has 'Editar' and 'Eliminar' buttons. The bottom of the page includes a footer with copyright information.

Anotação: Anotação 1

Minhas anotações Anotações Últimas Anotações Contato Roberto Pinheiro

Anotação 1

Descrição da anotação 1

This screenshot shows the details of a specific note titled 'Anotação 1'. The title is displayed prominently at the top of the page. Below it, there is a placeholder text 'Descrição da anotação 1'. The page structure is identical to the one in the first screenshot, indicating it is a detailed view of the note listed in the first image.

Listagem de anotações

laravel-larast.test/groups/2/notes

Minhas anotações Anotações Últimas Anotações Contato Roberto Pinheiro

Grupos

	Anotação	Grupo	Editar	Eliminar
--Nenhum--				
Família				
Trabalho	Anotação 3 *	Trabalho	Editar	Eliminar

Minhas anotações . Todos os direitos reservados

Anotação: Anotação 3

laravel-larast.test/notes/3

Minhas anotações Anotações Últimas Anotações Contato Roberto Pinheiro

Anotação 3

Descrição da anotação 3