Curso Laravel 5.2 com ACL

EspecializaTi (Carlos Ferreira)

https://www.youtube.com/watch?v=-AVa6GNpEL8&list=PLVSNL1PHDWvTch1r8uTSluw9SkzSA9cDJ

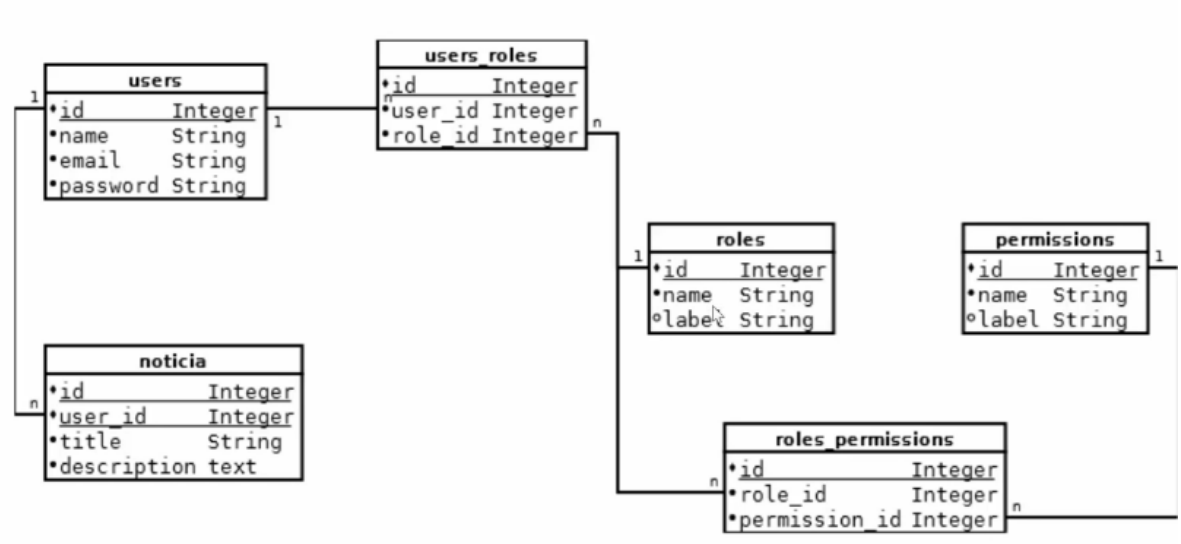
Resumo do curso feito por Roberto Pinheiro

Aula 01 - Apresentação do curso

Possui nível de acesso a usuários.

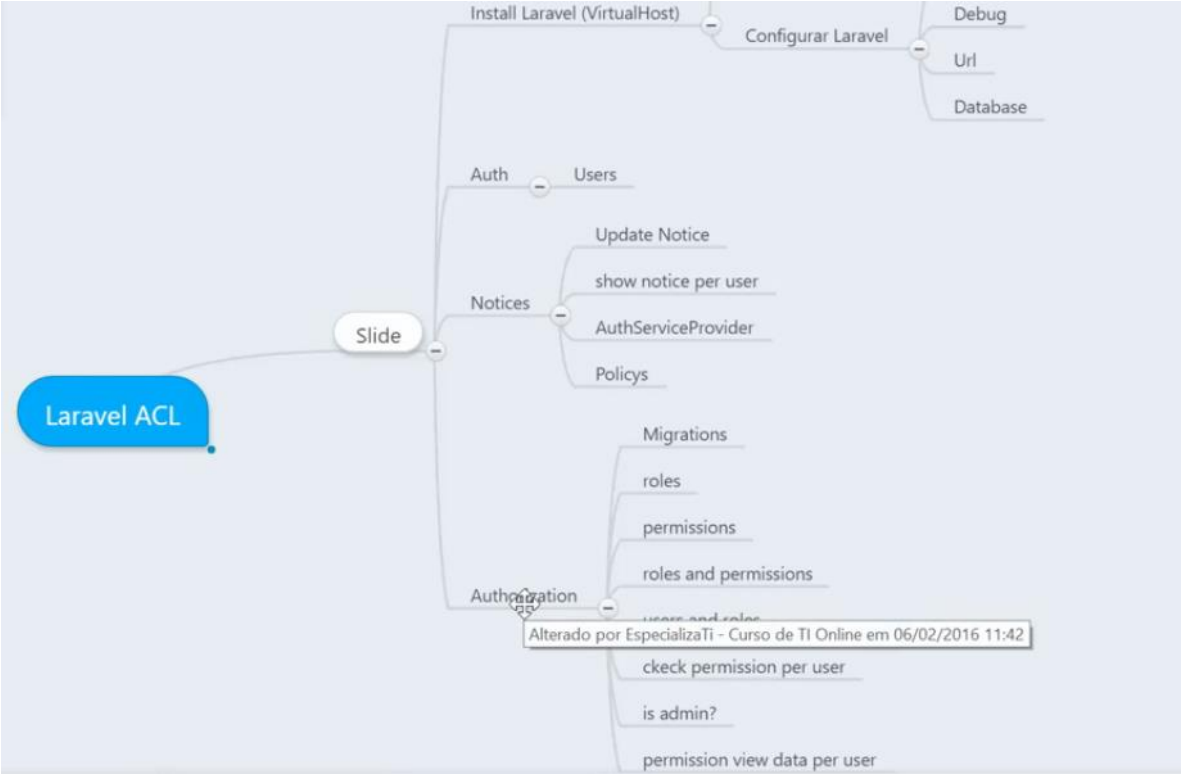
Funções permitidas para cada nível de usuário

Relacionamento entre as tabelas da aplicação



Aula 02 - Organização do Curso Laravel com ACL

Mapa mental



LaraACL
Usuários
Notícias
Funções
Permissões
Sair
Pesquisar?
Encontrar
123
423
3534
90
EspecializaTi - Todos os direitos reservados

Aula 03 - Instalando Laravel e configurando Virtual Host

Instalando o Laravel e criando o projeto

composer create-project --prefer-dist laravel/laravel laravel-acl

Configurando Virtual Host

c:/Windows/System32/drivers/etc/hosts

```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
# 102.54.94.97 rhino.acme.com # source server
# 38.25.63.10 x.acme.com # x client host

# localhost name resolution is handled within DNS itself.
#   127.0.0.1 localhost
#   ::1 localhost
127.0.0.1 localhost
::1 localhost
127.0.0.1     appLogin.test       #laragon magic!
127.0.0.1     blog.test           #laragon magic!
127.0.0.1     laravel-acl.test    #laragon magic!
127.0.0.1     laravel53-basico.test #laragon magic!
127.0.0.1     NewsApp.test        #laragon magic!
127.0.0.1     primeiro.test       #laragon magic!
127.0.0.1     rest-laravel.test   #laragon magic!
127.0.0.1     rest_api.test       #laragon magic!
127.0.0.1     SistemaUsuarios.test #laragon magic!
```

config/app.php

```
'timezone' => 'America/Sao_Paulo',
```

.env

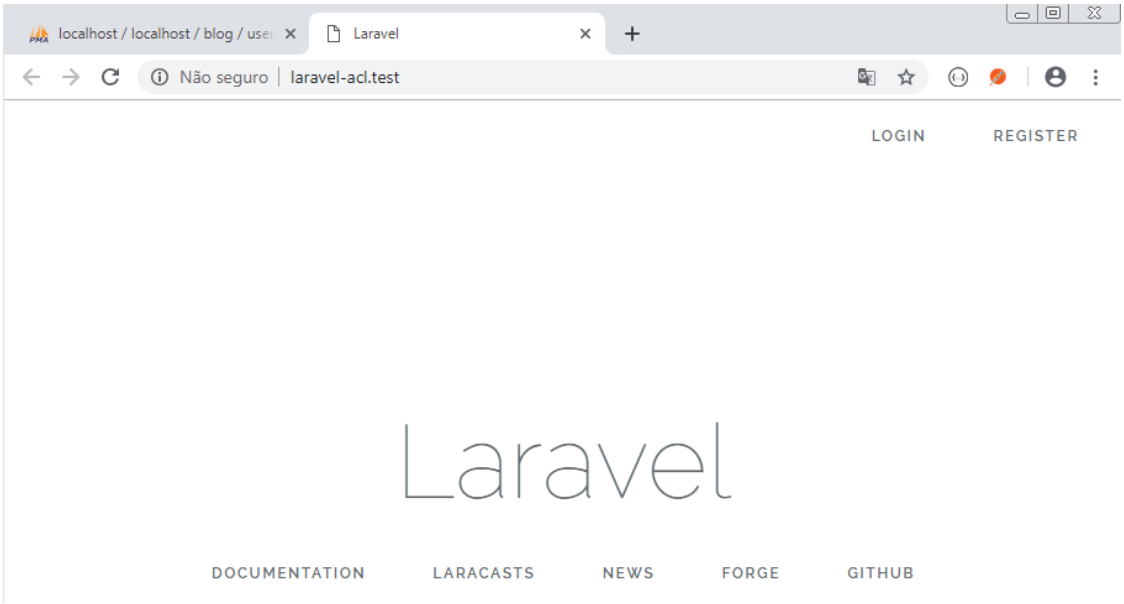
```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:M29FqBdy1ir30TerdGYjv3YPRsMN+3k8vDvIgM+stTw=
APP_DEBUG=true
APP_URL=http://laravel-acl.test
```

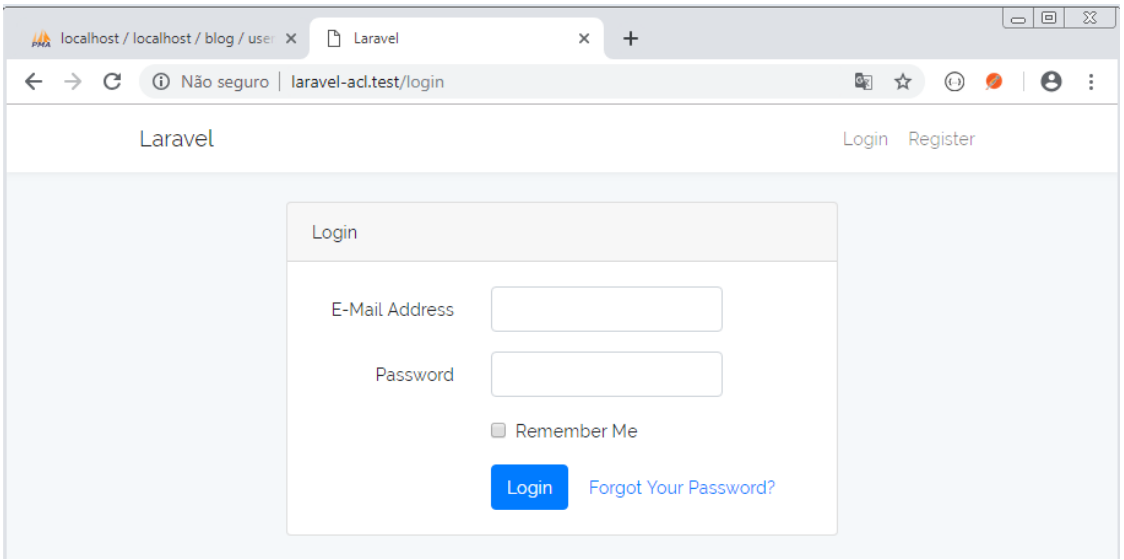
Aula 04 - Auth Laravel

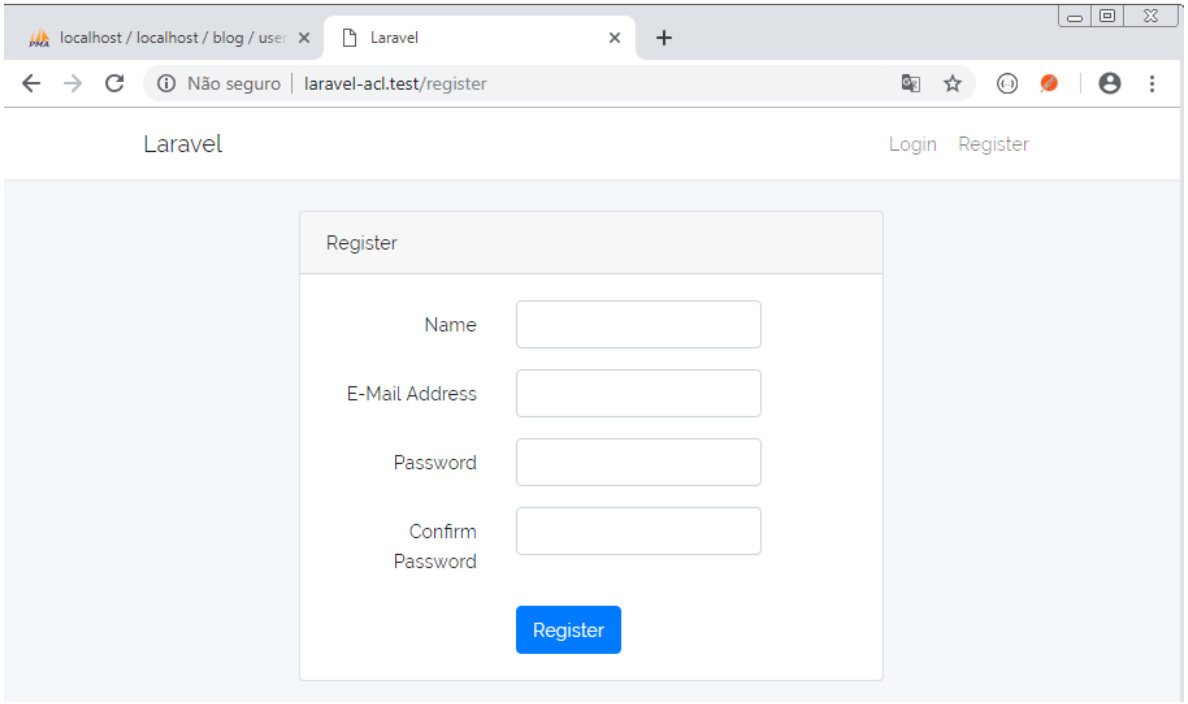
Criando sistema de autenticação com o Laravel

php artisan make:auth

```
C:\laragon\www\laravel-acl
λ php artisan make:auth
Authentication scaffolding generated successfully.

C:\laragon\www\laravel-acl
λ
```

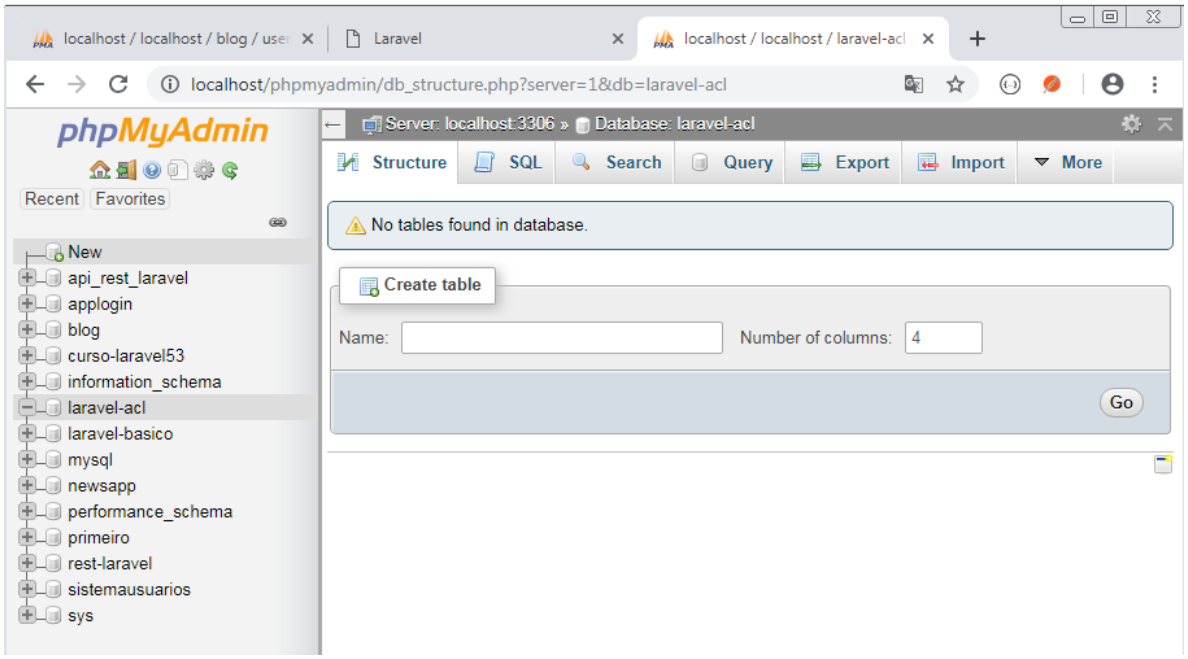






Criando o banco de dados

Crie o banco de dados chamado laravel-acl



Configurando o acesso ao banco de dados

Abrir o arquivo .env e fazer a seguinte configuração:

DB_CONNECTION=mysql
DB_HOST=localhost
DB_PORT=3306
DB_DATABASE=laravel-acl
DB_USERNAME=root
DB_PASSWORD=

Criar model e migration

php artisan make:model Post -m

```
C:\laragon\www\laravel-acl
λ php artisan make:model Post -m
Model created successfully.
Created Migration: 2019_02_24_140124_create_posts_table

C:\laragon\www\laravel-acl
λ
```

database/migrations/2019_02_24_140124_create_posts_table.php

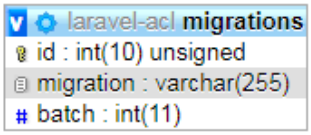
```php
<?php

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

class CreatePostsTable extends Migration
{
   public function up()
   {
      Schema::create('posts', function (Blueprint $table) {
         $table->increments('id');
         $table->integer('user_id')->unsigned();
         $table->foreign('user_id')->references('id')->on('users')->onDelete('cascade');
         $table->string('title', 200);
         $table->text('description');
         $table->timestamps();
      });
   }

   public function down()
   {
      Schema::drop('posts');
   }
}
```

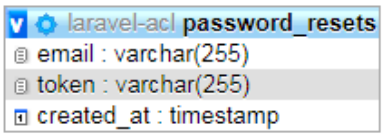
Criando as tabelas

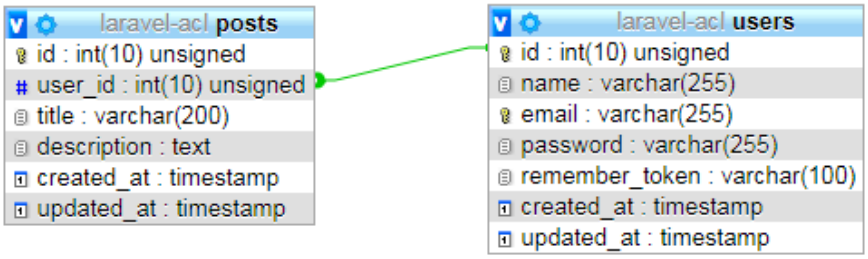
php artisan migrate

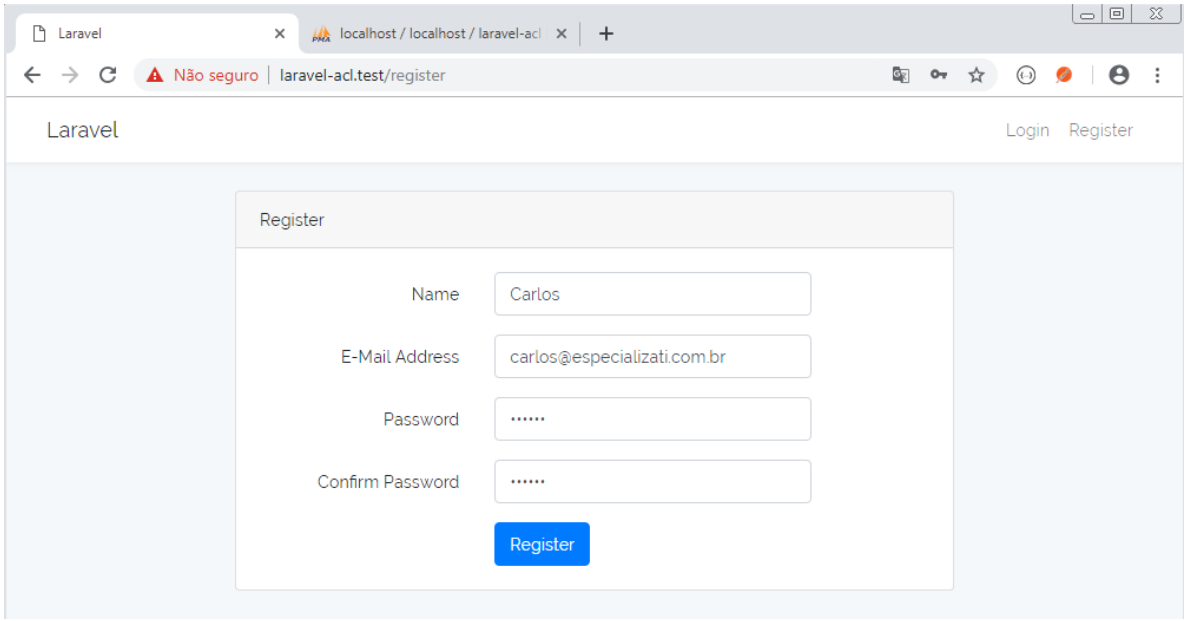
```
C:\laragon\www\laravel-acl
λ php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated:  2014_10_12_000000_create_users_table
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated:  2014_10_12_100000_create_password_resets_table
Migrating: 2019_02_24_140124_create_posts_table
Migrated:  2019_02_24_140124_create_posts_table

C:\laragon\www\laravel-acl
λ
```

No phpmyadmin clique em Design

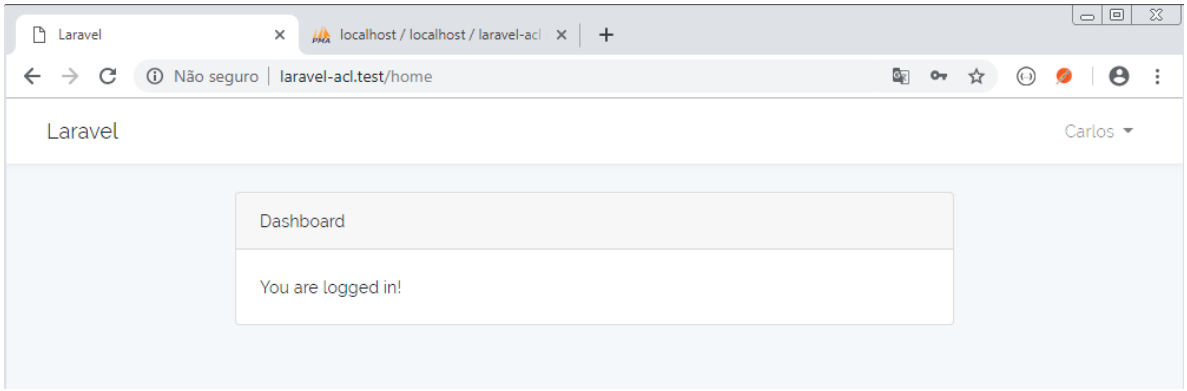


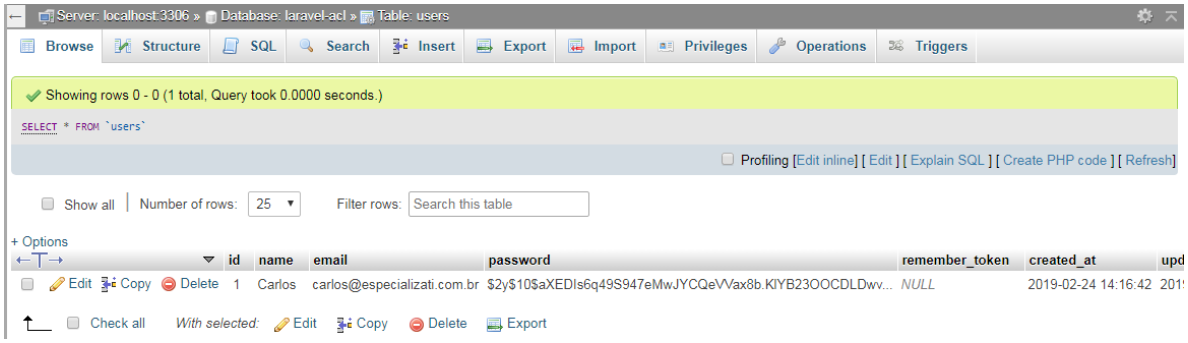




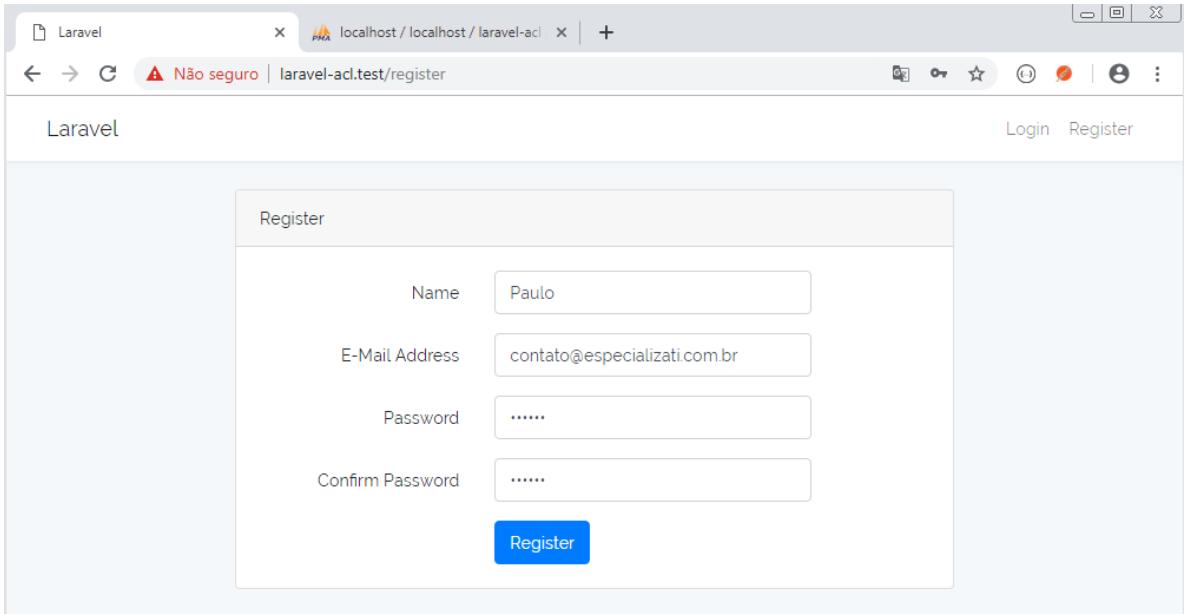


Senha: 123456

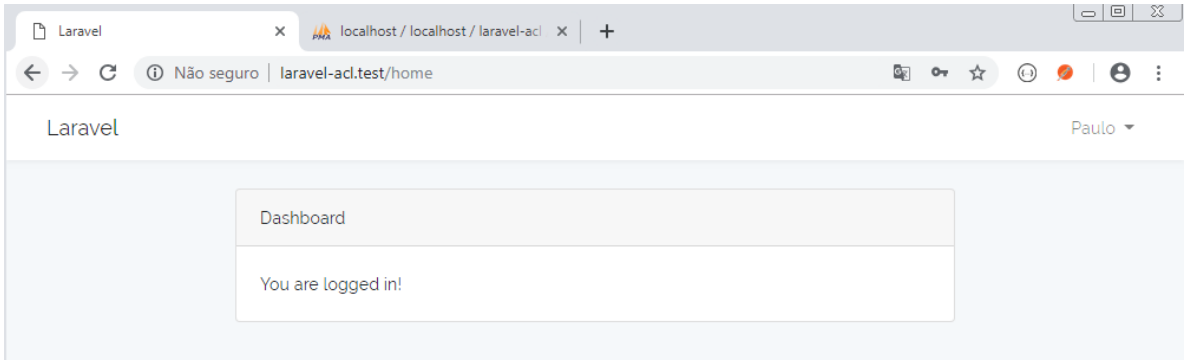




Faça logout e cadastre um novo usuário



Senha: 123456



Aula 05 - Cadastrando Posts e listando

app/Post.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Post extends Model
{
   public function user()
   {
      // Relacionamento de muitos para um
      return $this->belongsTo(User::class);
   }
}
```

app/http/Controllers/HomeController.php

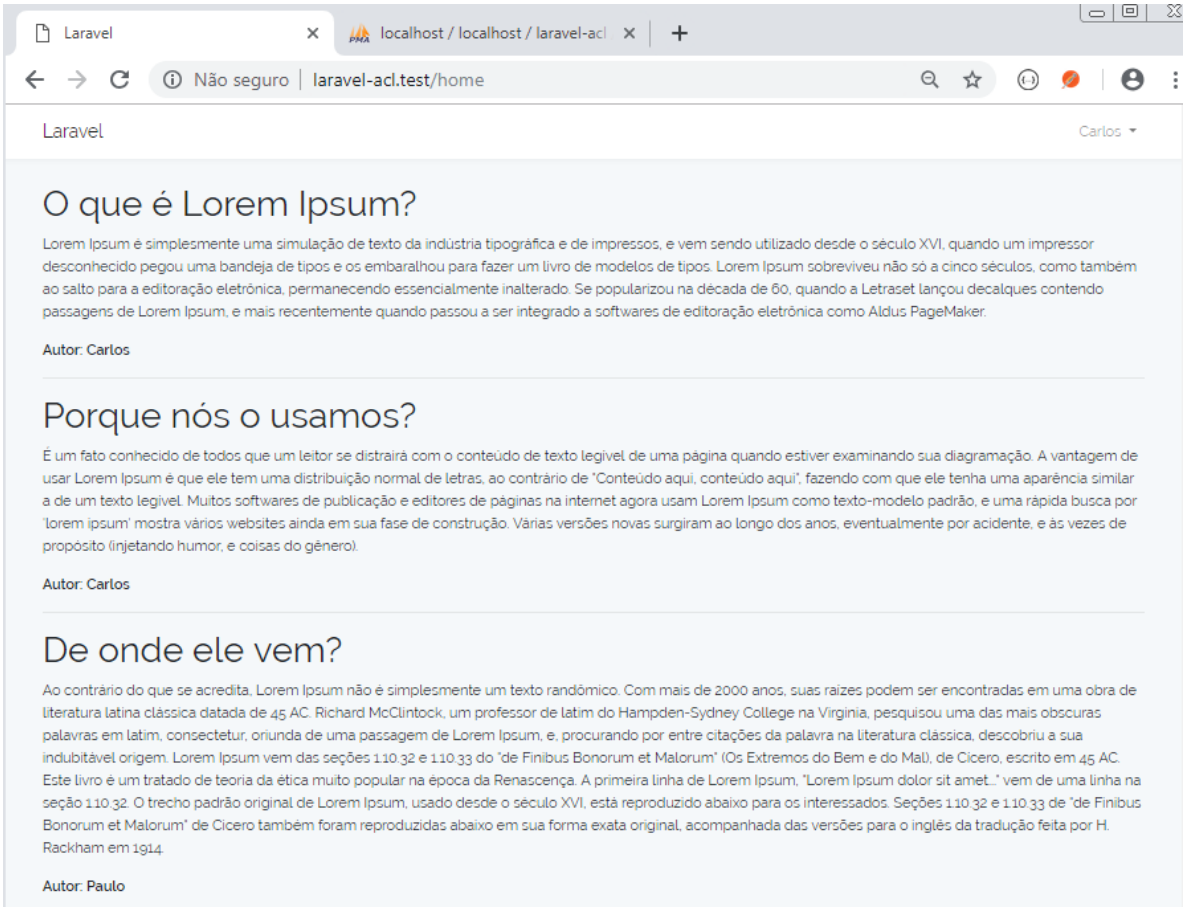
```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Post;

class HomeController extends Controller
{
   public function __construct()
   {
      $this->middleware('auth');
   }

   public function index(Post $post)
   {
      $posts = $post->all();
      return view('home', compact('posts'));
   }
}
```

resources/views/home.blade.php

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

@section('content')
<div class="container">
   @forelse($posts as $post)
      <h1>{{$post->title}}</h1>
      <p>{{$post->description}}</p>
      <p><b>Autor: {{$post->user->name}}</b></p>
      <hr>
   @empty
      <p>Nenhum post cadastrado</p>
   @endforelse
</div>
@endsection
```

routes/web.php

```
// Foi inserido automaticamente ao criar auth
Auth::routes();
Route::get('/home', 'HomeController@index')->name('home');
```

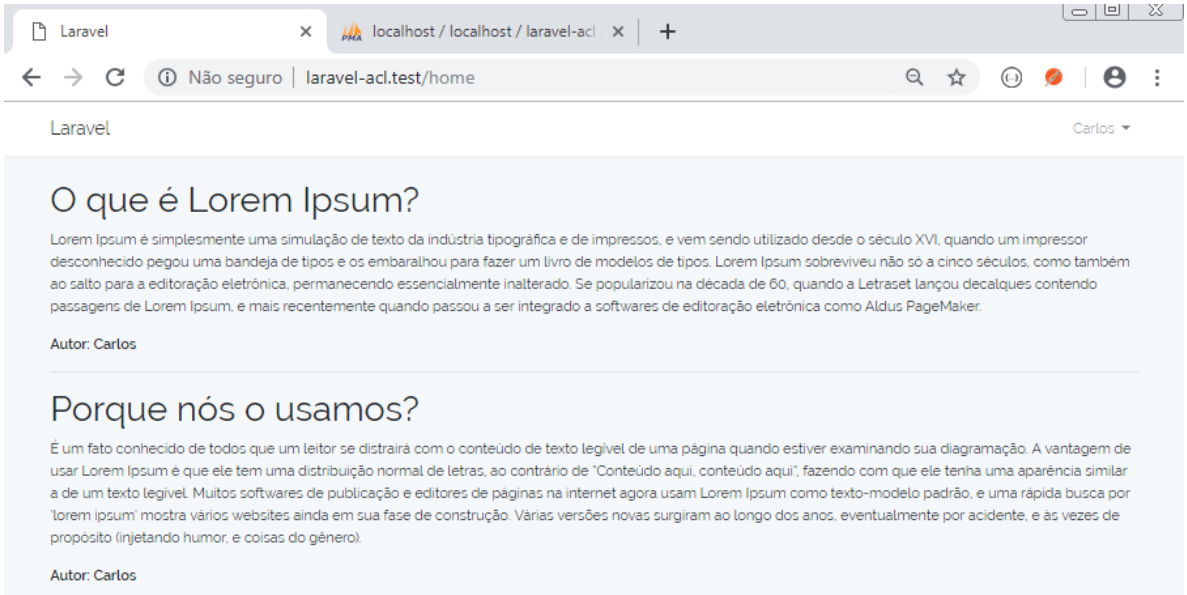


Aula 06 - Iniciando ACL com Laravel

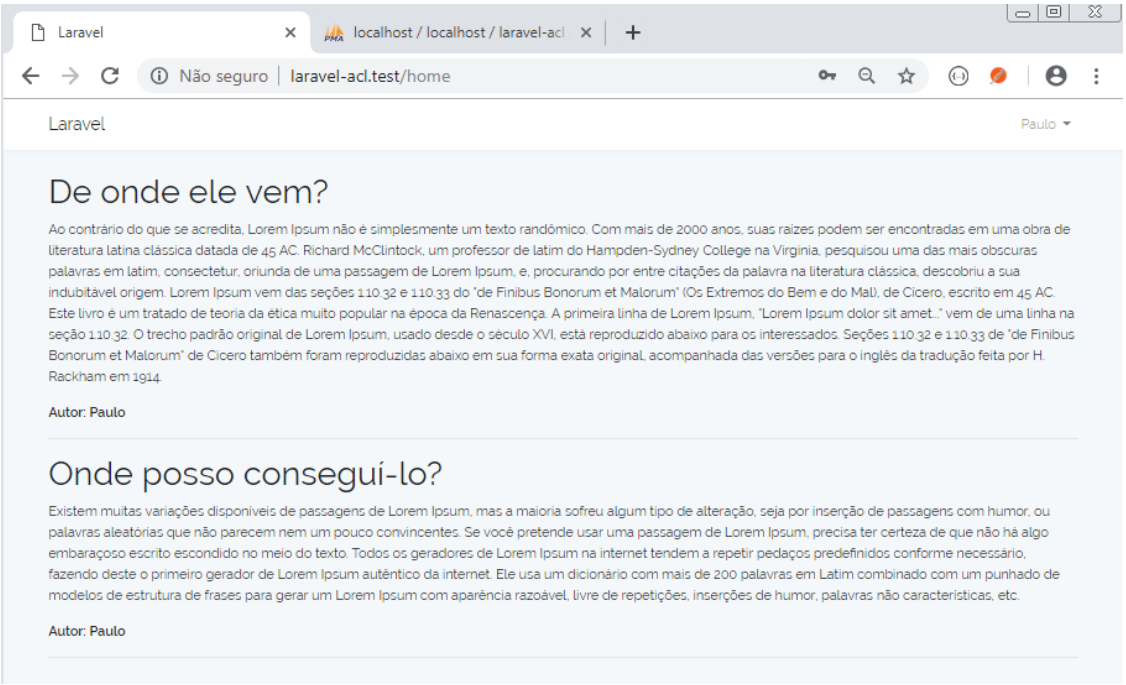
Exibindo apenas os posts do usuário logado

app/http/Controllers/HomeController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Post;

class HomeController extends Controller
{
   public function __construct()
   {
      $this->middleware('auth');
   }

   public function index(Post $post)
   {
      // Seleciona todos os posts do usuário logado
      $posts = $post->where('user_id', auth()->user()->id)->get();
      return view('home', compact('posts'));
  }
}
```



Usuário logado como Paulo



Preparando edição do post

routes/web.php

```
Auth::routes();
Route::get('/home', 'HomeController@index')->name('home');
Route::get('/post/{id}/update', 'HomeController@update');
```

resources/views/home.blade.php

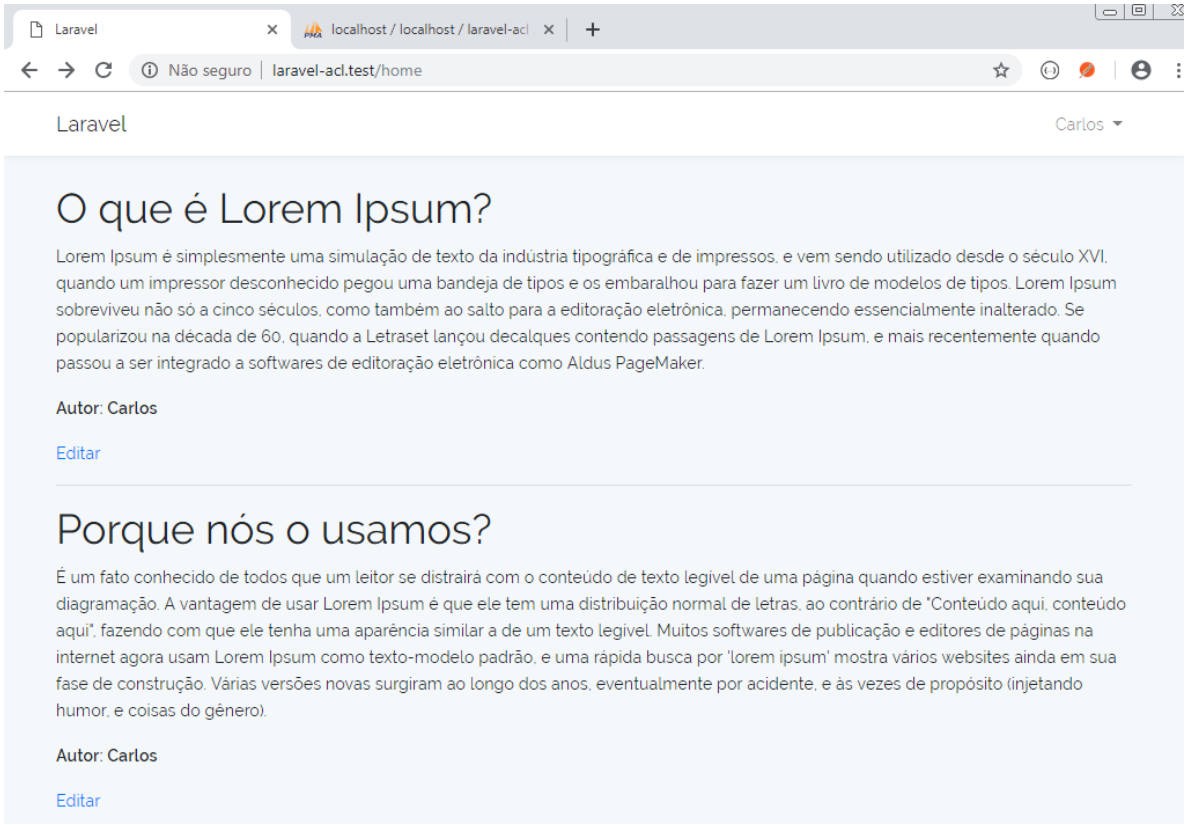
```
@extends('layouts.app')

@section('content')
<div class="container">
  @forelse($posts as $post)
    <h1>{{$post->title}}</h1>
    <p>{{$post->description}}</p>
    <p><b>Autor: {{$post->user->name}}</b></p>
    <p><a href="{{url("/post/$post->id/update")}}">Editar</a></p>
    <hr>
  @empty
    <p>Nenhum post cadastrado</p>
  @endforelse
</div>
@endsection
```

app/http/Controllers/HomeController.php

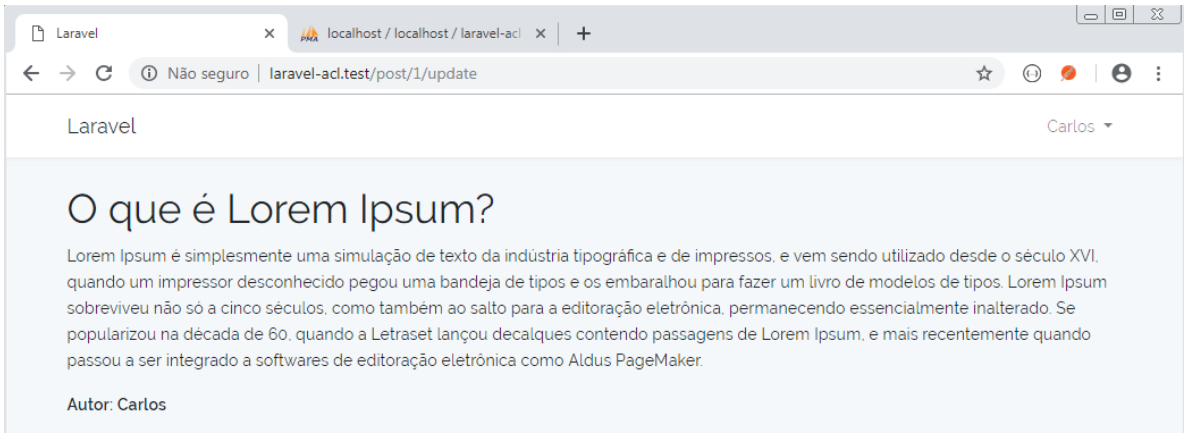
```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Post;

class HomeController extends Controller
{
   public function __construct()
   {
      $this->middleware('auth');
   }

   public function index(Post $post)
   {
      // Seleciona todos os posts
      // $posts = $post->all();

      // Seleciona todos os posts do usuário logado
      $posts = $post->where('user_id', auth()->user()->id)->get();

      return view('home', compact('posts'));
   }

   public function update($idpost){
      $post = Post::find($idpost);
      return view('post-update', compact('post'));
   }

}
```

resources/views/post-update.blade.php

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

@section('content')
<div class="container">
      <h1>{{$post->title}}</h1>
      <p>{{$post->description}}</p>
      <p><b>Autor: {{$post->user->name}}</b></p>
</div>
@endsection
```



Clique no link "Editar" do primeiro post:



Bloqueando edição de posts que não são do usuário

Definindo a primeira ACL

app/Providers/AuthServiceProvider.php

```
<?php

namespace App\Providers;

// use Illuminate\Support\Facades\Gate;
use Illuminate\Contracts\Auth\Access\Gate as GateContract;
use Illuminate\Foundation\Support\Providers\AuthServiceProvider as ServiceProvider;
use App\Post;
use App\User;

class AuthServiceProvider extends ServiceProvider
{
    protected $policies = [
        'App\Model' => 'App\Policies\ModelPolicy',
    ];

    public function boot(GateContract $gate)
    {
        $this->registerPolicies();

        $gate->define('update-post', function(User $user, Post $post){
            return $user->id == $post->user_id;
        });
    }
}
```

app/http/Controllers/HomeController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Post;
use Gate;

class HomeController extends Controller
{
   public function __construct()
   {
      $this->middleware('auth');
   }

   public function index(Post $post)
   {
      // Seleciona todos os posts
      // $posts = $post->all();

      // Seleciona todos os posts do usuário logado
      $posts = $post->where('user_id', auth()->user()->id)->get();

      return view('home', compact('posts'));
   }

   public function update($idpost){

      $post = Post::find($idpost);

      // $this->authorize('update-post', $post);

      if( Gate::denies('update-post', $post) )
         abort(403, 'Unauthorized');

      return view('post-update', compact('post'));

   }

}
```

Laravel | localhost / localhost / laravel-acl

Não seguro | laravel-acl.test/post/1/update

Laravel

Carlos

O que é Lorem Ipsum?

Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos, e vem sendo utilizado desde o século XVI, quando um impressor desconhecido pegou uma bandeja de tipos e os embaralhou para fazer um livro de modelos de tipos. Lorem Ipsum sobreviveu não só a cinco séculos, como também ao salto para a editoração eletrônica, permanecendo essencialmente inalterado. Se popularizou na década de 60, quando a Letraset lançou decalques contendo passagens de Lorem Ipsum, e mais recentemente quando passou a ser integrado a softwares de editoração eletrônica como Aldus PageMaker.

Autor: Carlos

Whoops! There was an error. | localhost / localhost / laravel-acl

Não seguro | laravel-acl.test/post/3/update

Symfony \ Component \ HttpKernel \ Exception \ HttpException

Unauthorized

COPY

Application frames (2) | All frames (57)

56 Symfony\Component\HttpKernel\Exception\HttpException
…\vendor\laravel\framework\src\Illuminate\Foun
:942

55 Illuminate\Foundation\Application abort
…\vendor\laravel\framework\src\Illuminate\Foun
:44

54 abort
…\app\Http\Controllers\HomeController.php
:44

53 App\Http\Controllers\HomeController update
…\vendor\laravel\framework\src\Illuminate\Rout
:54

C:\laragon\www\laravel-acl\vendor\laravel\framework\src\Illuminate\Foundation\Application.php

```
932.     * @return void
933.     *
934.     * @throws \Symfony\Component\HttpKernel\Exception\HttpException
935.     */
936.    public function abort($code, $message = '', array $headers = [])
937.    {
938.        if ($code == 404) {
939.            throw new NotFoundHttpException($message);
940.        }
941.
942.        throw new HttpException($code, $message, null, $headers);
943.    }
944.
945.    /**
946.     * Register a terminating callback with the application.
947.     *
948.     * @param  \Closure  $callback
949.     * @return $this
950.     */
951.    public function terminating(Closure $callback)
952.    {
953.        $this->terminatingCallbacks[] = $callback;
954.
955.        return $this;
956.    }
957.
```

Arguments

1. "Unauthorized"

No comments for this stack frame.

Environment & details:

Aula 07 - Laravel ACL na View

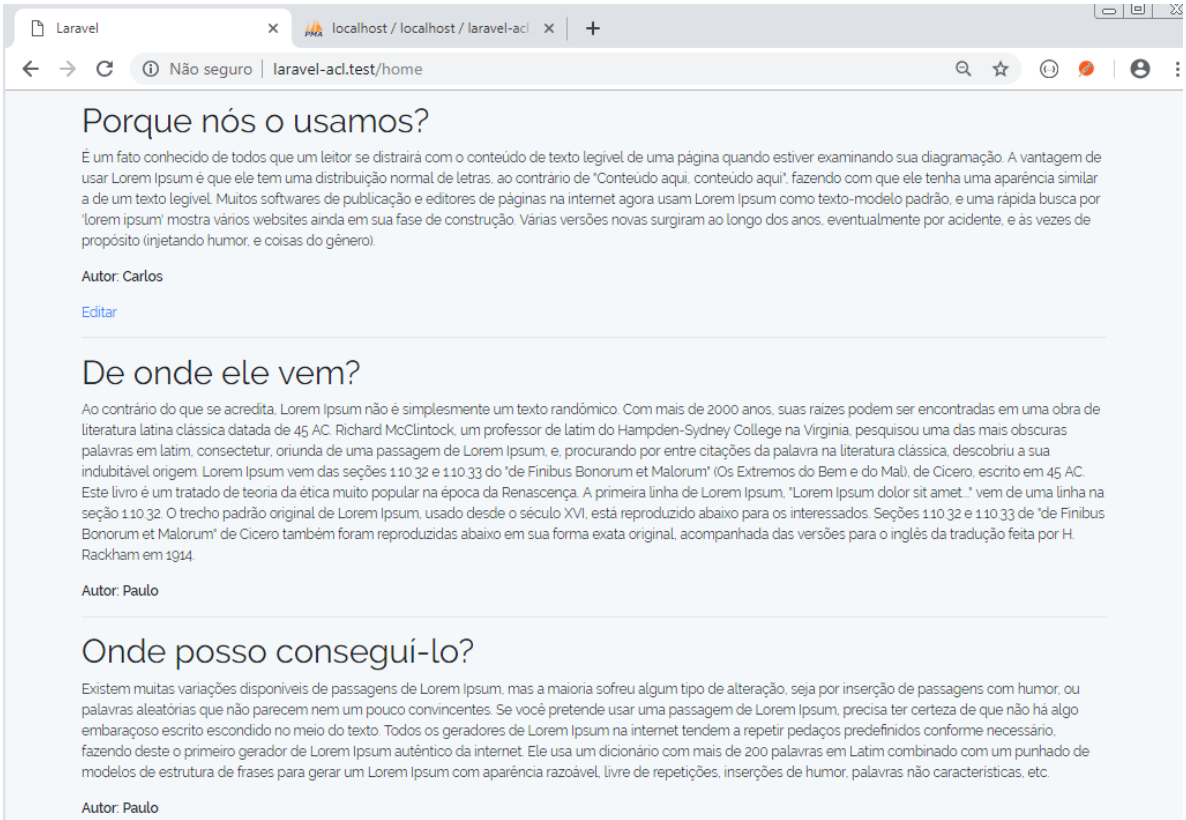
Exibindo todos os posts e permitindo editar apenas os do usuário

app/http/Controllers/HomeController.php

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Post;
use Gate;

class HomeController extends Controller
{
   public function __construct()
   {
      $this->middleware('auth');
   }

   public function index(Post $post)
   {
      // Seleciona todos os posts
      $posts = $post->all();

      // Seleciona todos os posts do usuário logado
      // $posts = $post->where('user_id', auth()->user()->id)->get();

      return view('home', compact('posts'));
   }

   public function update($idpost){

      $post = Post::find($idpost);

      // $this->authorize('update-post', $post);

      if( Gate::denies('update-post', $post) )
         abort(403, 'Unauthorized');

      return view('post-update', compact('post'));

   }

}
```

resources/views/home.blade.php

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

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

   @forelse($posts as $post)
      <h1>{{$post->title}}</h1>
      <p>{{$post->description}}</p>
      <p><b>Autor: {{$post->user->name}}</b></p>
      @can('update-post', $post)
         <p><a href="{{url("/post/$post->id/update")}}">Editar</a></p>
      @endcan
      <hr>
   @empty
      <p>Nenhum post cadastrado</p>
   @endforelse

</div>
@endsection
```



Aula 08 - Definindo políticas de acesso ao sistema

Criando políticas de acesso ao sistema

php artisan make:policy PostPolicy

```
C:\laragon\www\laravel-acl
λ php artisan make:policy PostPolicy
Policy created successfully.

C:\laragon\www\laravel-acl
λ
```

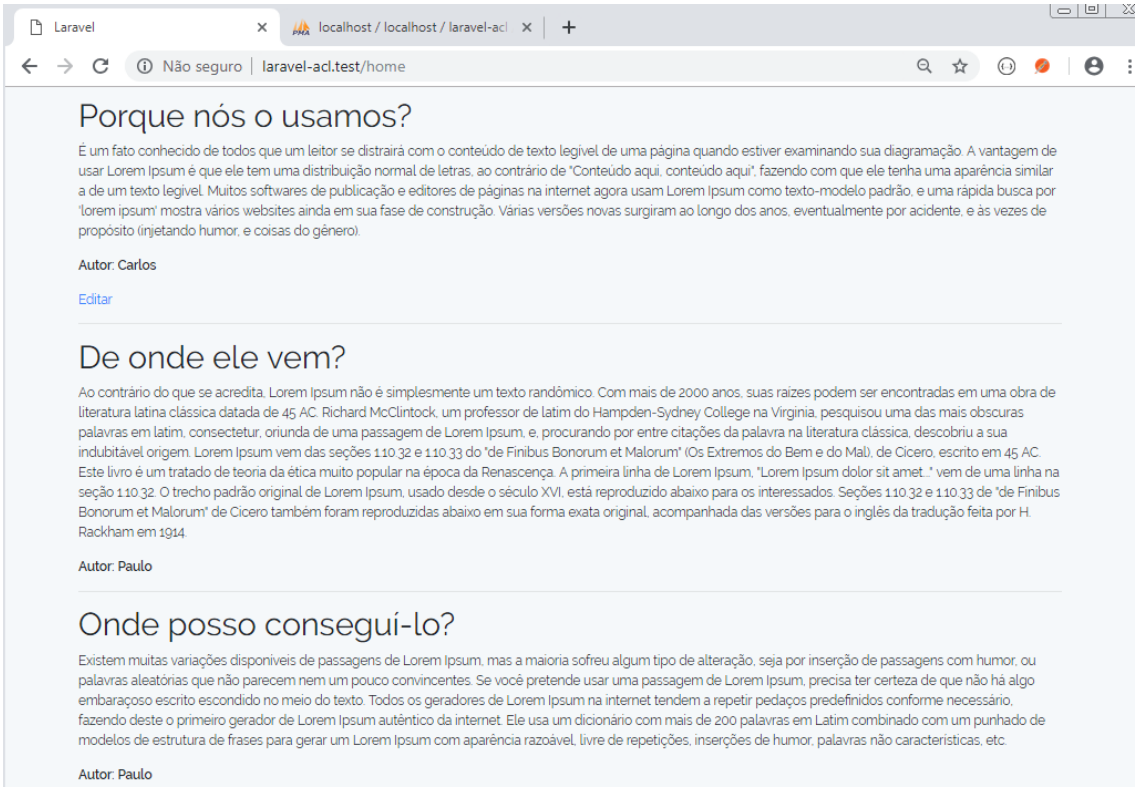
Organizando as políticas de acesso

app/Policies/PostPolicy.php

```
<?php

namespace App\Policies;

use Illuminate\Auth\Access\HandlesAuthorization;
use App\User;
use App\Post;

class PostPolicy
{
   use HandlesAuthorization;

   public function __construct()
   {
      //
   }

   public function updatePost(User $user, Post $post)
   {
      return $user->id == $post->user_id;
   }

}
```

app/Providers/AuthServiceProvider.php

```php
<?php

namespace App\Providers;

// use Illuminate\Support\Facades\Gate;
use Illuminate\Contracts\Auth\Access\Gate as GateContract;
use Illuminate\Foundation\Support\Providers\AuthServiceProvider as ServiceProvider;
use App\Post;
use App\User;

class AuthServiceProvider extends ServiceProvider
{
   protected $policies = [
      // 'App\Model' => 'App\Policies\ModelPolicy',
      \App\Post::class => \App\Policies\PostPolicy::class
   ];
   public function boot(GateContract $gate)
   {
      $this->registerPolicies();
      /*
      $gate->define('update-post', function(User $user, Post $post){
         return $user->id == $post->user_id;
      });
      */
   }
}
```



Laravel | localhost / localhost / laravel-acl

Não seguro | laravel-acl.test/post/2/update

Laravel

Carlos

Porque nós o usamos?

É um fato conhecido de todos que um leitor se distrairá com o conteúdo de texto legível de uma página quando estiver examinando sua diagramação. A vantagem de usar Lorem Ipsum é que ele tem uma distribuição normal de letras, ao contrário de "Conteúdo aqui, conteúdo aqui", fazendo com que ele tenha uma aparência similar a de um texto legível. Muitos softwares de publicação e editores de páginas na internet agora usam Lorem Ipsum como texto-modelo padrão, e uma rápida busca por 'lorem ipsum' mostra vários websites ainda em sua fase de construção. Várias versões novas surgiram ao longo dos anos, eventualmente por acidente, e às vezes de propósito (injetando humor, e coisas do gênero).

Autor: Carlos

Whoops! There was an error. | localhost / localhost / laravel-acl

Não seguro | laravel-acl.test/post/3/update

Symfony \ Component \ HttpKernel \ Exception \ HttpException

Unauthorized

COPY

Application frames (2) | All frames (57)

56 Symfony\Component\HttpKernel\Exception\HttpException
…\vendor\laravel\framework\src\Illuminate\Foun
:942

55 Illuminate\Foundation\Application abort
…\vendor\laravel\framework\src\Illuminate\Foun
:44

54 abort
…\app\Http\Controllers\HomeController.php
:44

53 App\Http\Controllers\HomeController update
…\vendor\laravel\framework\src\Illuminate\Rout
:54

C:\laragon\www\laravel-acl\vendor\laravel\framework\src\Illuminate\Foundation\Application.php

```
932.     * @return void
933.     *
934.     * @throws \Symfony\Component\HttpKernel\Exception\HttpException
935.     */
936.    public function abort($code, $message = '', array $headers = [])
937.    {
938.        if ($code == 404) {
939.            throw new NotFoundHttpException($message);
940.        }
941.
942.        throw new HttpException($code, $message, null, $headers);
943.    }
944.
945.    /**
946.     * Register a terminating callback with the application.
947.     *
948.     * @param  \Closure  $callback
949.     * @return $this
950.     */
951.    public function terminating(Closure $callback)
952.    {
953.        $this->terminatingCallbacks[] = $callback;
954.
955.        return $this;
956.    }
957.
```

Arguments

1. "Unauthorized"

No comments for this stack frame.

Environment & details:

Aula 09 - Criando migrations ACL FOR LARGE SYSTEMS

Criando model e migration para as funções

php artisan make:model Role -m

```
C:\laragon\www\laravel-acl
λ php artisan make:model Role -m
Model created successfully.
Created Migration: 2019_02_25_005005_create_roles_table

C:\laragon\www\laravel-acl
λ
```

php artisan make:model Permission -m

```
C:\laragon\www\laravel-acl
λ php artisan make:model Permission -m
Model created successfully.
Created Migration: 2019_02_25_005221_create_permissions_table

C:\laragon\www\laravel-acl
λ
```

database/migrations/2019_02_25_005005_create_roles_table.php

```php
<?php

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

class CreateRolesTable extends Migration
{
    public function up()
    {
        Schema::create('roles', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name', 50);
            $table->string('label', 200);
            $table->timestamps();
        });

        Schema::create('role_user', function (Blueprint $table) {
            $table->increments('id');
            $table->integer('role_id')->unsigned();
            $table->integer('user_id')->unsigned();

            $table->foreign('role_id')
                ->references('id')
                ->on('roles')
                ->onDelete('cascade');

            $table->foreign('user_id')
                ->references('id')
                ->on('users')
                ->onDelete('cascade');
        });
    }

    public function down()
    {
        Schema::drop('role_user');
        Schema::drop('roles');
    }
}
```

database/migrations/2019_02_25_005221_create_permissions_table.php

```php
<?php

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

class CreatePermissionsTable extends Migration
{
    public function up()
    {
        Schema::create('permissions', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name', 50);
            $table->string('label', 200);
            $table->timestamps();
        });

        Schema::create('permission_role', function (Blueprint $table) {
            $table->increments('id');
            $table->integer('permission_id')->unsigned();
            $table->integer('role_id')->unsigned();

            $table->foreign('permission_id')
                ->references('id')
                ->on('permissions')
                ->onDelete('cascade');

            $table->foreign('role_id')
                ->references('id')
                ->on('roles')
                ->onDelete('cascade');
        });
    }

    public function down()
    {
        Schema::drop('permission_role');
        Schema::drop('permissions');
    }
}
```

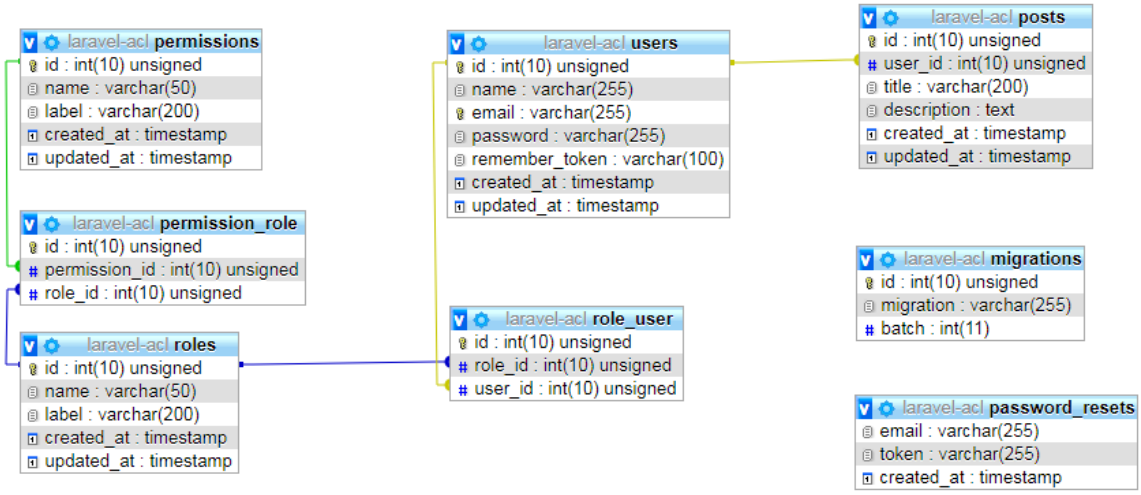
Criando as tabelas

php artisan migrate

```
C:\laragon\www\laravel-acl
λ php artisan migrate
Migrating: 2019_02_25_005005_create_roles_table
Migrated:  2019_02_25_005005_create_roles_table
Migrating: 2019_02_25_005221_create_permissions_table
Migrated:  2019_02_25_005221_create_permissions_table

C:\laragon\www\laravel-acl
λ
```

Design

No phpmyadmin clique em Design



Aula 10 - Definindo permissão dinamicamente

Preencha manualmente as tabelas a seguir:

Tabela roles

+ Options

			id	name	label	created_at	updated_at
Edit	Copy	Delete	1	manager	Manager	2019-02-25 01:33:04	2019-02-25 01:33:04
Edit	Copy	Delete	2	editor	Editor do sistema	2019-02-25 01:33:52	2019-02-25 01:33:52
Edit	Copy	Delete	3	adm	Administrador do Sistema	2019-02-25 01:34:47	2019-02-25 01:34:47

Check all *With selected:* Edit Copy Delete Export

Tabela permissions

+ Options

			id	name	label	created_at	updated_at
Edit	Copy	Delete	1	view_post	Visualiza o post	2019-02-25 01:37:03	2019-02-25 01:37:03
Edit	Copy	Delete	2	edit_post	Editar post	2019-02-25 01:38:02	2019-02-25 01:38:02
Edit	Copy	Delete	3	delete_post	Deletar post	2019-02-25 01:39:17	2019-02-25 01:39:17

Check all *With selected:* Edit Copy Delete Export

Tabela role_user

+ Options

			id	role_id	user_id
Edit	Copy	Delete	1	3	1
Edit	Copy	Delete	2	2	2

Check all *With selected:* Edit Copy Delete Export

Tabela permission_role

+ Options

				id	permission_id	role_id
	Edit	Copy	Delete	1	1	1
	Edit	Copy	Delete	2	1	2
	Edit	Copy	Delete	3	2	2

Check all *With selected:* Edit Copy Delete Export

app/Permission.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Permission extends Model
{
   public function roles()
   {
      return $this->belongsToMany(\App\Role::class);
   }
}
```

app/Providers/AuthServiceProvider.php

```php
<?php

namespace App\Providers;

use Illuminate\Contracts\Auth\Access\Gate as GateContract;
use Illuminate\Foundation\Support\Providers\AuthServiceProvider as ServiceProvider;
use App\Post;
use App\User;
use App\Permission;

class AuthServiceProvider extends ServiceProvider
{
    protected $policies = [
        /*
         \App\Post::class => \App\Policies\PostPolicy::class,
         */
    ];

    public function boot(GateContract $gate)
    {
        $this->registerPolicies($gate);

        $permissions = Permission::with('roles')->get();
        foreach( $permissions as $permission )
        {
            $gate->define($permission->name, function(User $user) use ($permission){
                return $user->hasPermission($permission);
            });
        }

    }
}
```

Aula 11 - Criando lógica de ACL

#users
Carlos
Paulo

#roles
Manager
Adm
Editor
Publish

#users <-> roles
Paulo -> Manager

#permissions
edit_post
edit
delete_post
edit_fin
view_rel_fin

#roles <-> permissions
Manager -> edit_post, edit, delete_post

```
@can('edit_post', $post)

@endcan
```

app/User.php

```php
<?php

namespace App;

use Illuminate\Foundation\Auth\User as Authenticatable;
use App\Permission;

class User extends Authenticatable
{
    protected $fillable = [
        'name', 'email', 'password',
    ];

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

    public function roles()
    {
        // Relacionamento de muitos para muitos
        return $this->belongsToMany(\App\Role::class);
    }

    public function hasPermission(Permission $permission)
    {
        return $this->hasAnyRoles($permission->roles);
    }

    public function hasAnyRoles($roles)
    {
        if(is_array($roles) || is_object($roles) ) {
            foreach($roles as $role){
                return $this->roles->contains('name', $role->name);
            }
        }

        return $this->roles->contains('name', $roles);
    }
}
```

Aula 12 - Mostrar Posts por Perfil

resources/views/home.blade.php

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

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

  @forelse($posts as $post)
    @can('view_post', $post)
      <h1>{{$post->title}}</h1>
      <p>{{$post->description}}</p>
      <p><b>Autor: {{$post->user->name}}</b></p>
      <p><a href="{{url("/post/$post->id/update")}}">Editar</a></p>
    @endcan
    <hr>
  @empty
    <p>Nenhum post cadastrado</p>
  @endforelse

</div>
@endsection
```

Na tabela role_user inserir o registro 3:

+ Options

				id	role_id	user_id
☐	Edit	Copy	Delete	1	3	1
☐	Edit	Copy	Delete	2	2	2
☐	Edit	Copy	Delete	3	1	2

Check all *With selected:* Edit Copy Delete Export

O que é Lorem Ipsum?

Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos, e vem sendo utilizado desde o século XVI, quando um impressor desconhecido pegou uma bandeja de tipos e os embaralhou para fazer um livro de modelos de tipos. Lorem Ipsum sobreviveu não só a cinco séculos, como também ao salto para a editoração eletrônica, permanecendo essencialmente inalterado. Se popularizou na década de 60, quando a Letraset lançou decalques contendo passagens de Lorem Ipsum, e mais recentemente quando passou a ser integrado a softwares de editoração eletrônica como Aldus PageMaker.

Autor: Carlos

Editar

Porque nós o usamos?

É um fato conhecido de todos que um leitor se distrairá com o conteúdo de texto legível de uma página quando estiver examinando sua diagramação. A vantagem de usar Lorem Ipsum é que ele tem uma distribuição normal de letras, ao contrário de "Conteúdo aqui, conteúdo aqui", fazendo com que ele tenha uma aparência similar a de um texto legível. Muitos softwares de publicação e editores de páginas na internet agora usam Lorem Ipsum como texto-modelo padrão, e uma rápida busca por 'lorem ipsum' mostra vários websites ainda em sua fase de construção. Várias versões novas surgiram ao longo dos anos, eventualmente por acidente, e às vezes de propósito (injetando humor, e coisas do gênero).

Autor: Carlos

Editar

De onde ele vem?

Ao contrário do que se acredita, Lorem Ipsum não é simplesmente um texto randômico. Com mais de 2000 anos, suas raízes podem ser encontradas em uma obra de literatura latina clássica datada de 45 AC. Richard McClintock, um professor de latim do Hampden-Sydney College na Virginia, pesquisou uma das mais obscuras palavras em latim, consectetur, oriunda de uma passagem de Lorem Ipsum, e, procurando por entre citações da palavra na literatura clássica, descobriu a sua indubitável origem. Lorem Ipsum vem das seções 1.10.32 e 1.10.33 do "de Finibus Bonorum et Malorum" (Os Extremos do Bem e do Mal), de Cicero, escrito em 45 AC. Este livro é um tratado de teoria da ética muito popular na época da Renascença. A primeira linha de Lorem Ipsum, "Lorem Ipsum dolor sit amet..", vem de uma linha na seção 1.10.32. O trecho padrão original de Lorem Ipsum, usado desde o século XVI, está reproduzido abaixo para os interessados. Seções 1.10.32 e 1.10.33 de "de Finibus Bonorum et Malorum" de Cicero também foram reproduzidas abaixo em sua forma exata original, acompanhada das versões para o inglês da tradução feita por H. Rackham em 1914.

Autor: Paulo

Editar

Aula 13 - Debug perfis e melhorar código

routes/web.php

```
Auth::routes();

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

Route::get('/post/{id}/update', 'HomeController@update');

Route::get('/roles-permissions', 'HomeController@rolesPermissions');
```

app/http/Controllers/HomeController.php

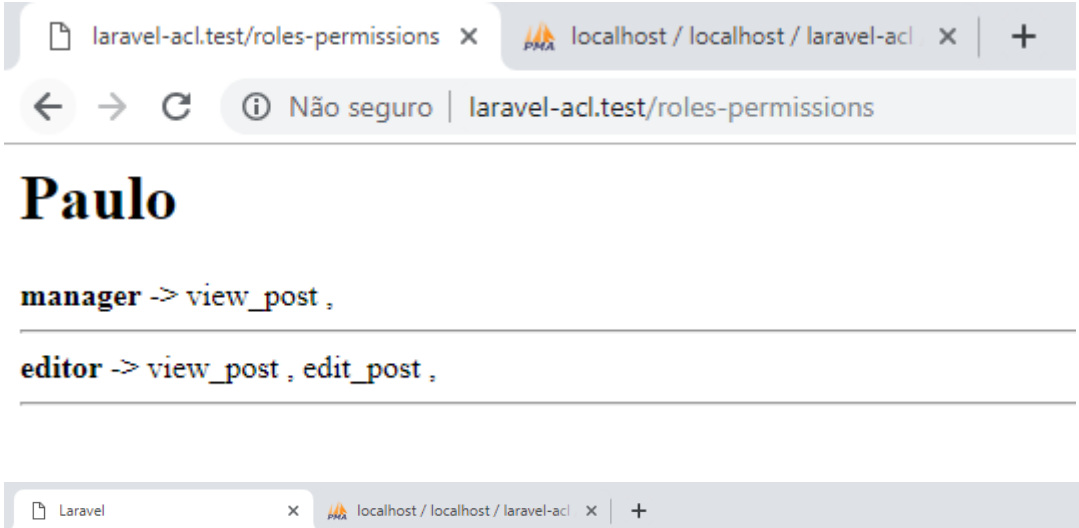
```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Post;
use Gate;

class HomeController extends Controller
{
   public function __construct()
   {
      $this->middleware('auth');
   }
   public function index(Post $post)
   {
      // Seleciona todos os posts
      $posts = $post->all();
      // Seleciona todos os posts do usuário logado
      // $posts = $post->where('user_id', auth()->user()->id)->get();
      return view('home', compact('posts'));
   }

   public function update($idpost){

      $post = Post::find($idpost);
      // $this->authorize('update-post', $post);
      if( Gate::denies('update-post', $post) )
         abort(403, 'Unauthorized');
      return view('post-update', compact('post'));
   }

   public function rolesPermissions(){
      // return 'Roles e Permissions to users';

      $nameUser = auth()->user()->name;
      echo "<h1>{$nameUser}</h1>";

      foreach(auth()->user()->roles as $role){
         echo "<b>$role->name</b> -> ";
         $permissions = $role->permissions;
         foreach($permissions as $permission){
            echo " $permission->name , ";
         }
         echo "<hr>";
      }
   }

}
```

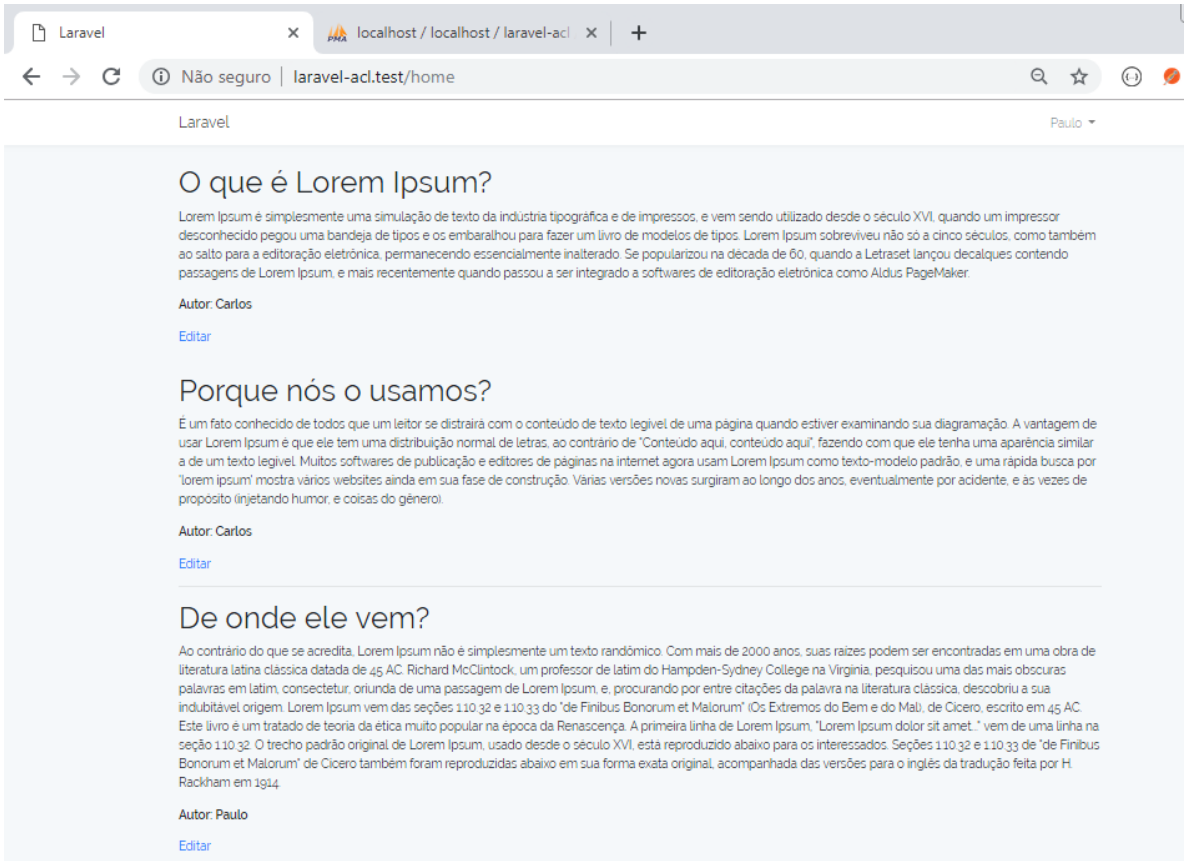
app/Role.php

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Role extends Model
{
   public function permissions()
   {
      return $this->belongsToMany(\App\Permission::class);
   }
}
```

app/User.php

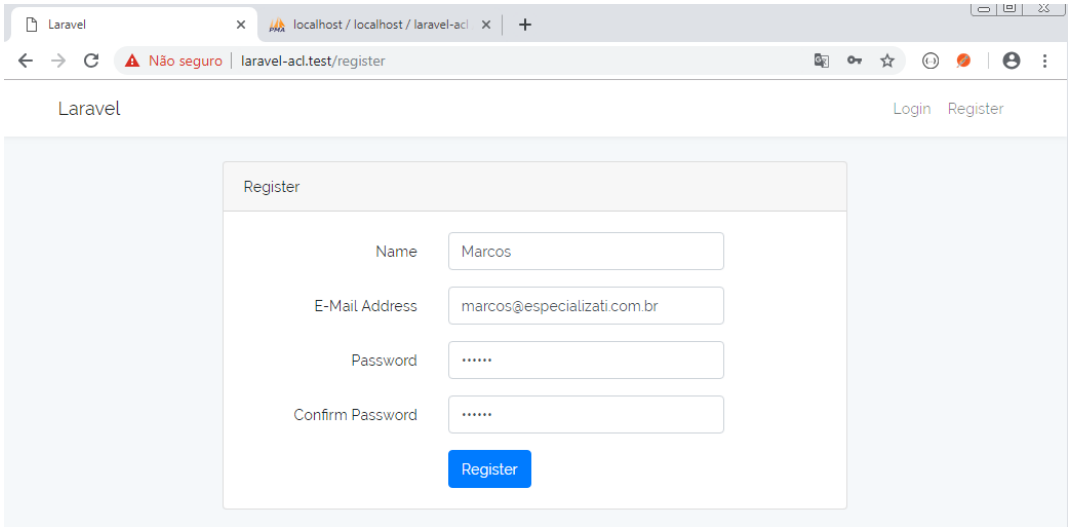
```php
<?php

namespace App;

use Illuminate\Foundation\Auth\User as Authenticatable;
use App\Permission;

class User extends Authenticatable
{
   protected $fillable = [
      'name', 'email', 'password',
   ];

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

   public function roles()
   {
      // Relacionamento de muitos para muitos
      return $this->belongsToMany(\App\Role::class);
   }

   public function hasPermission(Permission $permission)
   {
      return $this->hasAnyRoles($permission->roles);
   }

   public function hasAnyRoles($roles)
   {
      if(is_array($roles) || is_object($roles) ) {
         // foreach($roles as $role){
         //     return $this->roles->contains('name', $role->name);
         // }
         return !! $roles->intersect($this->roles)->count();
      }

      return $this->roles->contains('name', $roles);
   }
}
```


laravel-acl.test/roles-permissions
localhost / localhost / laravel-acl
Não seguro | laravel-acl.test/roles-permissions
Paulo
manager -> view_post ,
editor -> view_post , edit_post ,


Laravel
localhost / localhost / laravel-acl
Não seguro | laravel-acl.test/home
Laravel
Paulo
O que é Lorem Ipsum?
Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos, e vem sendo utilizado desde o século XVI, quando um impressor desconhecido pegou uma bandeja de tipos e os embaralhou para fazer um livro de modelos de tipos. Lorem Ipsum sobreviveu não só a cinco séculos, como também ao salto para a editoração eletrônica, permanecendo essencialmente inalterado. Se popularizou na década de 60, quando a Letraset lançou decalques contendo passagens de Lorem Ipsum, e mais recentemente quando passou a ser integrado a softwares de editoração eletrônica como Aldus PageMaker.
Autor: Carlos
Editar
Porque nós o usamos?
É um fato conhecido de todos que um leitor se distrairá com o conteúdo de texto legível de uma página quando estiver examinando sua diagramação. A vantagem de usar Lorem Ipsum é que ele tem uma distribuição normal de letras, ao contrário de "Conteúdo aqui, conteúdo aqui", fazendo com que ele tenha uma aparência similar a de um texto legível. Muitos softwares de publicação e editores de páginas na internet agora usam Lorem Ipsum como texto-modelo padrão, e uma rápida busca por 'lorem ipsum' mostra vários websites ainda em sua fase de construção. Várias versões novas surgiram ao longo dos anos, eventualmente por acidente, e às vezes de propósito (injetando humor, e coisas do gênero).
Autor: Carlos
Editar
De onde ele vem?
Ao contrário do que se acredita, Lorem Ipsum não é simplesmente um texto randômico. Com mais de 2000 anos, suas raízes podem ser encontradas em uma obra de literatura latina clássica datada de 45 AC. Richard McClintock, um professor de latim do Hampden-Sydney College na Virginia, pesquisou uma das mais obscuras palavras em latim, consectetur, oriunda de uma passagem de Lorem Ipsum, e, procurando por entre citações da palavra na literatura clássica, descobriu a sua indubitável origem. Lorem Ipsum vem das seções 1.10.32 e 1.10.33 do "de Finibus Bonorum et Malorum" (Os Extremos do Bem e do Mal), de Cicero, escrito em 45 AC. Este livro é um tratado de teoria da ética muito popular na época da Renascença. A primeira linha de Lorem Ipsum, "Lorem Ipsum dolor sit amet..", vem de uma linha na seção 1.10.32. O trecho padrão original de Lorem Ipsum, usado desde o século XVI, está reproduzido abaixo para os interessados. Seções 1.10.32 e 1.10.33 de "de Finibus Bonorum et Malorum" de Cicero também foram reproduzidas abaixo em sua forma exata original, acompanhada das versões para o inglês da tradução feita por H. Rackham em 1914.
Autor: Paulo
Editar

Aula 14 - Definindo Super Admin Sistema

Insira um novo usuário:



Senha: 123456

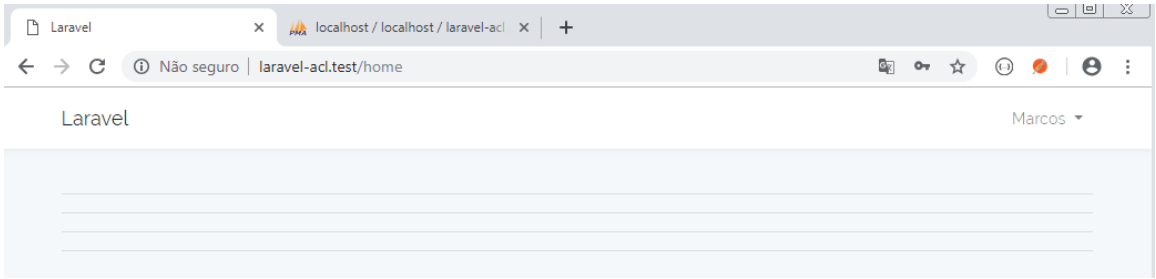
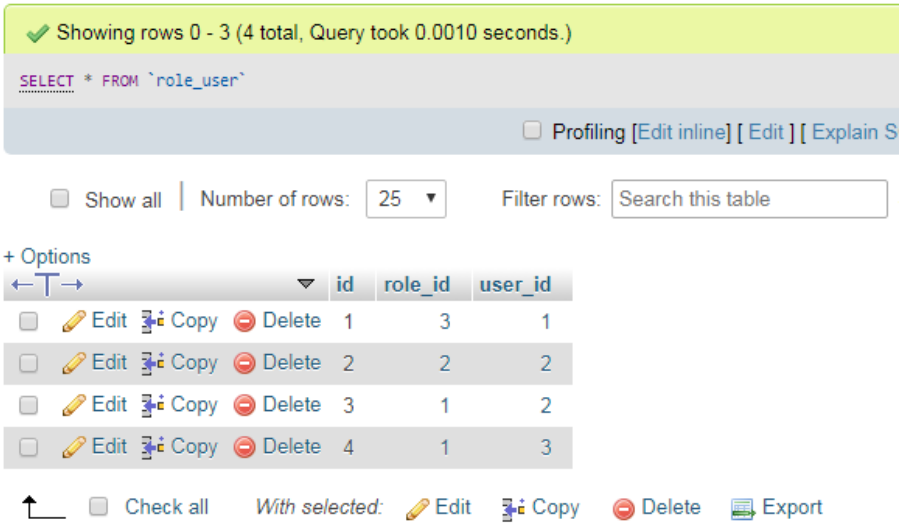


tabela role_user

Insira o registro 4:



laravel-acl.test/roles-permissions | localhost / localhost / laravel-acl

Não seguro | laravel-acl.test/roles-permissions

Marcos

manager -> view_post ,

Laravel | localhost / localhost / laravel-acl

Não seguro | laravel-acl.test/home

Laravel

Marcos

O que é Lorem Ipsum?

Lorem Ipsum é simplesmente uma simulação de texto da indústria tipográfica e de impressos, e vem sendo utilizado desde o século XVI, quando um impressor desconhecido pegou uma bandeja de tipos e os embaralhou para fazer um livro de modelos de tipos. Lorem Ipsum sobreviveu não só a cinco séculos, como também ao salto para a editoração eletrônica, permanecendo essencialmente inalterado. Se popularizou na década de 60, quando a Letraset lançou decalques contendo passagens de Lorem Ipsum, e mais recentemente quando passou a ser integrado a softwares de editoração eletrônica como Aldus PageMaker.

Autor: Carlos

Editar

Porque nós o usamos?

É um fato conhecido de todos que um leitor se distrairá com o conteúdo de texto legível de uma página quando estiver examinando sua diagramação. A vantagem de usar Lorem Ipsum é que ele tem uma distribuição normal de letras, ao contrário de "Conteúdo aqui, conteúdo aqui", fazendo com que ele tenha uma aparência similar a de um texto legível. Muitos softwares de publicação e editores de páginas na internet agora usam Lorem Ipsum como texto-modelo padrão, e uma rápida busca por 'lorem ipsum' mostra vários websites ainda em sua fase de construção. Várias versões novas surgiram ao longo dos anos, eventualmente por acidente, e às vezes de propósito (injetando humor, e coisas do gênero).

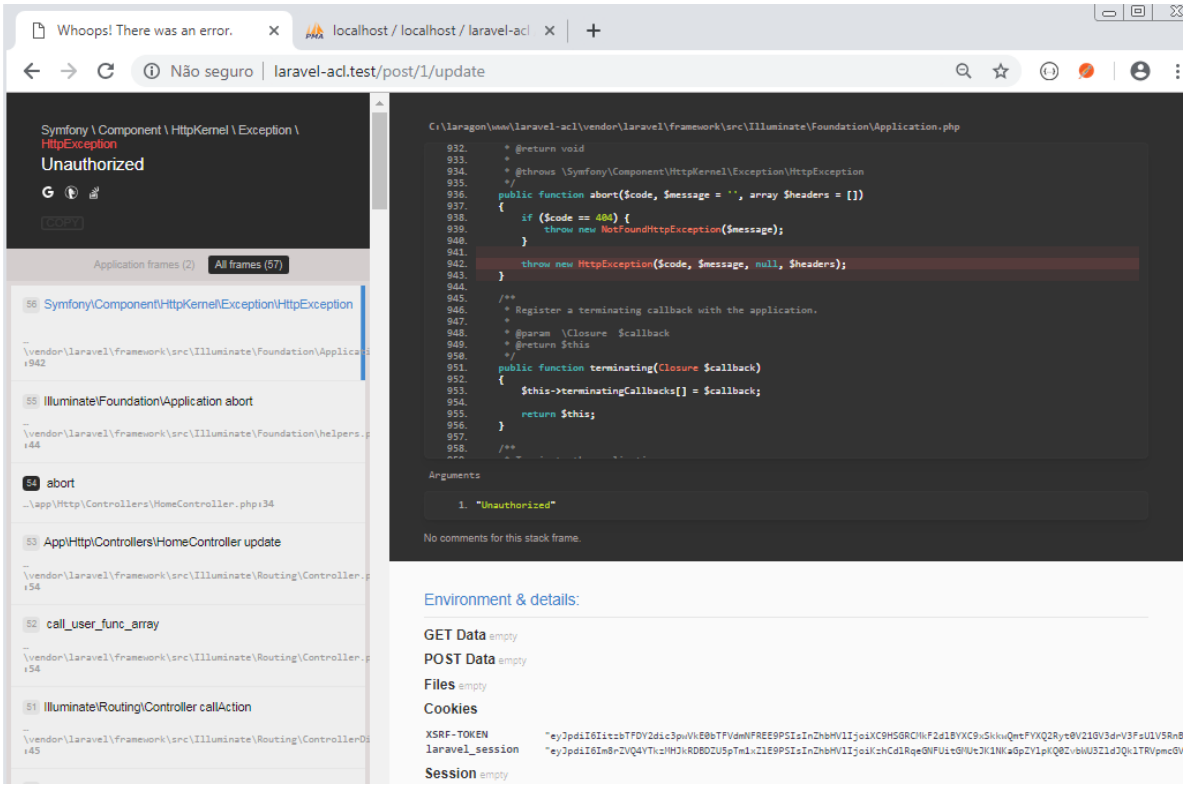
Autor: Carlos

Editar

De onde ele vem?

Ao contrário do que se acredita, Lorem Ipsum não é simplesmente um texto randômico. Com mais de 2000 anos, suas raízes podem ser encontradas em uma obra de literatura latina clássica datada de 45 AC. Richard McClintock, um professor de latim do Hampden-Sydney College na Virginia, pesquisou uma das mais obscuras palavras em latim, consectetur, oriunda de uma passagem de Lorem Ipsum, e, procurando por entre citações da palavra na literatura clássica, descobriu a sua indubitável origem. Lorem Ipsum vem das seções 1.10.32 e 1.10.33 do "de Finibus Bonorum et Malorum" (Os Extremos do Bem e do Mal), de Cicero, escrito em 45 AC. Este livro é um tratado de teoria da ética muito popular na época da Renascença. A primeira linha de Lorem Ipsum, "Lorem Ipsum dolor sit amet..." vem de uma linha na seção 1.10.32. O trecho padrão original de Lorem Ipsum, usado desde o século XVI, está reproduzido abaixo para os interessados. Seções 1.10.32 e 1.10.33 de "de Finibus Bonorum et Malorum" de Cicero também foram reproduzidas abaixo em sua forma exata original, acompanhada das versões para o inglês da tradução feita por H. Rackham em 1914.

Ao tentar editar o post 1



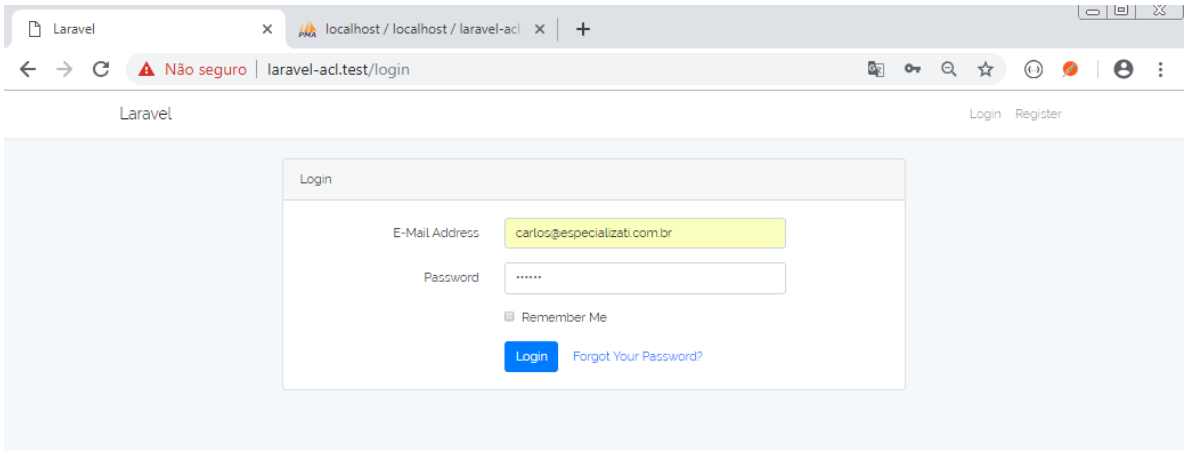
app/Providers/AuthServiceProvider.php

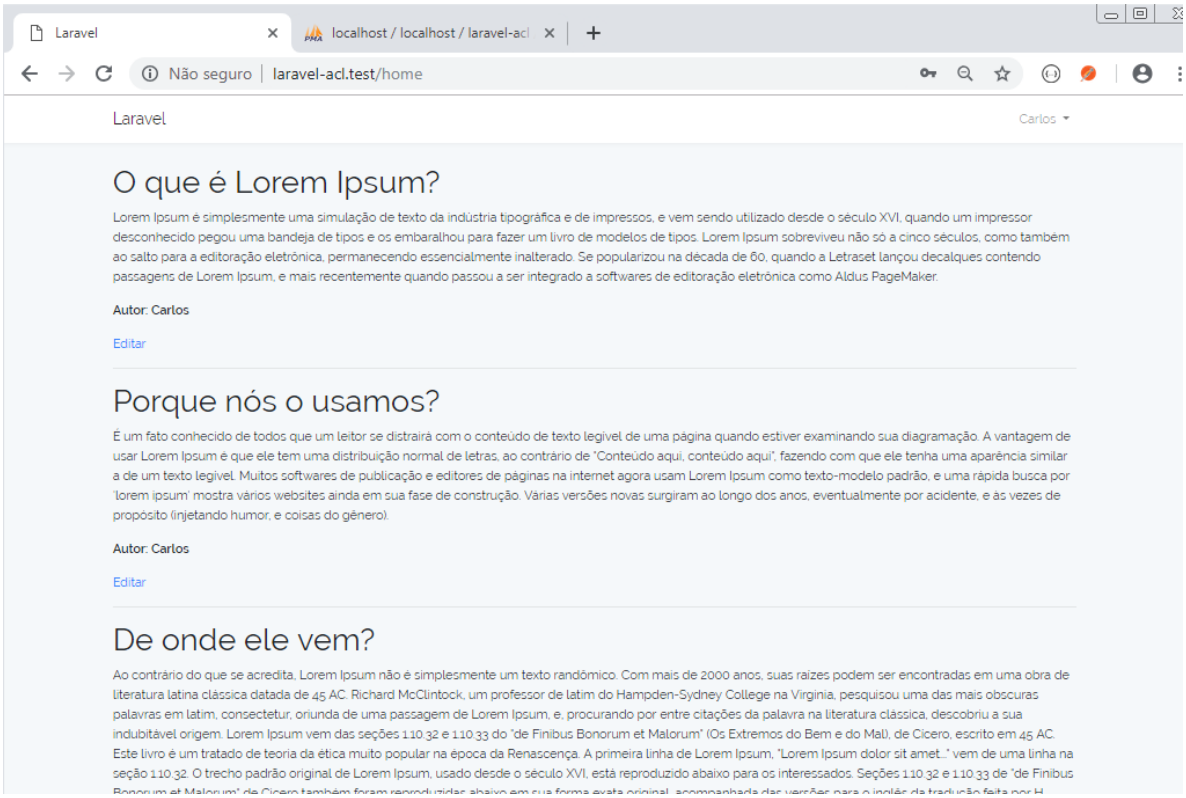
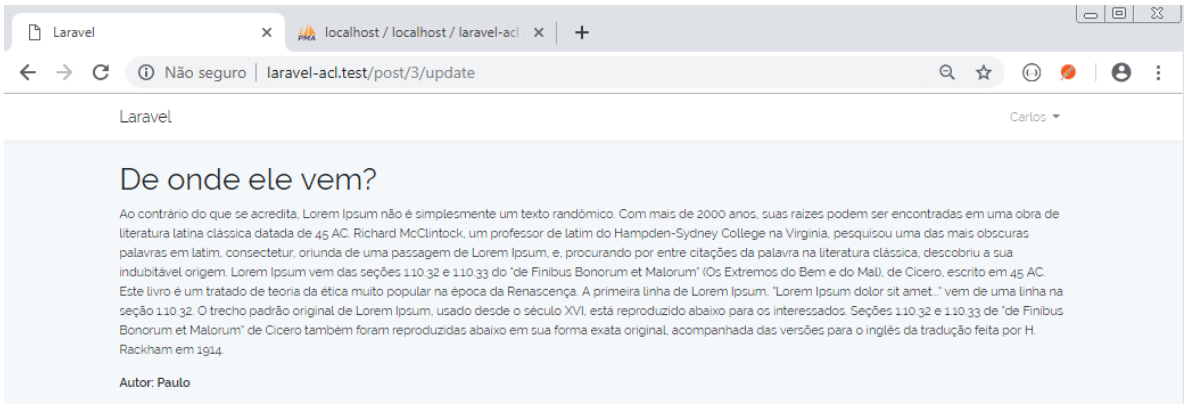
```php
<?php

namespace App\Providers;

use Illuminate\Contracts\Auth\Access\Gate as GateContract;
use Illuminate\Foundation\Support\Providers\AuthServiceProvider as ServiceProvider;
use App\Post;
use App\User;
use App\Permission;

class AuthServiceProvider extends ServiceProvider
{
    protected $policies = [
        /*
         \App\Post::class => \App\Policies\PostPolicy::class,
         */
    ];

    public function boot(GateContract $gate)
    {
        $this->registerPolicies($gate);

        /*
        $gate->define('update-post', function(User $user, Post $post){
            return $user->id == $post->user_id;
        });
         */

        $permissions = Permission::with('roles')->get();
        foreach( $permissions as $permission )
        {
            $gate->define($permission->name, function(User $user) use ($permission){
                return $user->hasPermission($permission);
            });
        }

        $gate->before(function(User $user, $ability){

            if ( $user->hasAnyRoles('adm') )
                return true;

        });
    }
}
```

app/Policies/PostPolicy.php

```php
<?php

namespace App\Policies;

use Illuminate\Auth\Access\HandlesAuthorization;
use App\User;
use App\Post;

class PostPolicy
{
   use HandlesAuthorization;

   public function __construct()
   {
      //
   }

   public function updatePost(User $user, Post $post)
   {
      return $user->id == $post->user_id;
   }

   public function before(User $user)
   {
      return $user->name == 'Carlos';
   }
}
```

Logando como um administrador do sistema





Clique no link Editar do terceiro post