

ENSINO MÉDIO INTEGRADO - INFORMÁTICA

Disciplina de Desenvolvimento Web

Aula 11: Autenticação / Breeze

Gil Eduardo de Andrade

Conceitos Preliminares

(<https://laravel.com/docs/8.x>)

Autenticação (<https://laravel.com/docs/8.x/authentication>)

Atualmente, diversas aplicações voltadas a plataforma web fornecem um modo de autenticação aos seus usuários, para que os mesmos possam efetuar o “processo login”. A implementação desse recurso, normalmente, requer tempo e possui um grau considerável de complexidade, além de ser potencialmente arriscado. Por esta razão, o Laravel disponibiliza ferramentas que possibilitam a implementação de autenticação de forma rápida, segura e fácil.

Kits - Laravel Breeze (<https://laravel.com/docs/8.x/starter-kits>)

O Laravel fornece alguns kits para iniciantes que permitem trabalhar com o processo de autenticação de maneira prática. Estes kits organizam a aplicação em questão para trabalhar com autenticação, visto que já definem as rotas, as classes de controle e as telas de visualização necessárias para registrar e autenticar os usuários. O Laravel Breeze é uma implementação básica e simples de todos os recursos de autenticação do Laravel, incluindo login, registro, redefinição de senha, verificação de e-mail e confirmação de senha. A camada de visualização padrão do Laravel Breeze é composta de modelos Blade simples.



Codificação – Laravel / Autenticação

(<https://laravel.com/docs/8.x/authentication>)

Instalando o Breeze (No terminal - dentro da pasta do projeto)

```
composer require laravel/breeze --dev
```

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/dwii/10
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/10-Autenticação$ composer
require laravel/breeze --dev
```

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/dwii/10
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
> @php artisan vendor:publish --tag=laravel-assets --ansi --force
No publishable resources for tag [laravel-assets].
Publishing complete.
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/10-Autenticação$
```

`php artisan breeze:install`

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/dwii/10
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/10-Autenticação$ php artis
an breeze:install
Breeze scaffolding installed successfully.
Please execute the "npm install" && "npm run dev" commands to build your
assets.
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/10-Autenticação$
```

`npm install`

`npm run dev`

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/dwii/10
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/10-Autenticação$ npm insta
ll
```

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/dwii/10
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/10-Autenticação$ npm run d
ev
```

** Todas as rotas do Breeze são definidas no arquivo /routes/auth.php*

Arquivos Gerados (Rotas)

- **/routes/auth.php** - contém todas as rotas necessárias para executar o processo de autenticação dentro da aplicação em questão.

(No terminal - dentro da pasta do projeto)

```
php artisan route:list
```

(Arquivo: "/routes/auth.php")

```
...  
Route::middleware('guest')->group(function () {  
    Route::get('register', [RegisteredUserController::class, 'create'])  
        ->name('register');  
    Route::post('register', [RegisteredUserController::class, 'store']);  
    Route::get('login', [AuthenticatedSessionController::class, 'create'])  
        ->name('login');  
    Route::post('login', [AuthenticatedSessionController::class, 'store']);  
    ...  
});  
...
```

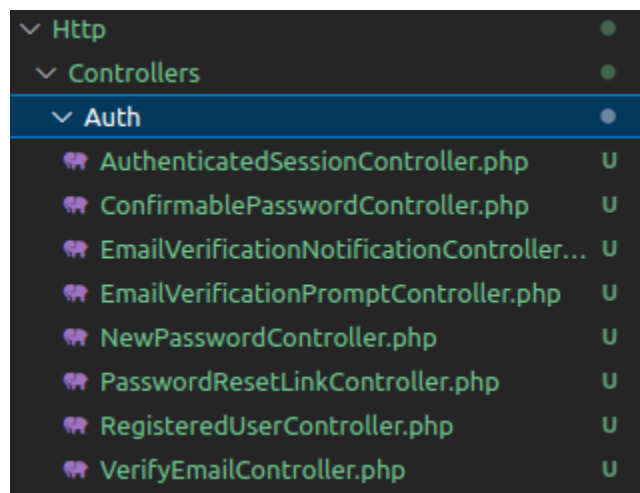
Arquivo de Rotas - Breeze.

(Arquivo-fonte: 10-Autenticação/routes/auth.php)

Arquivos Gerados (Controle)

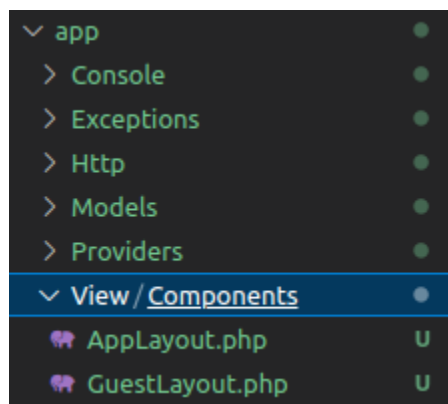
- **/app/Http/Controllers/auth/AuthenticatedSessionController.php** - responsável por configurar a sessão de autenticação do usuário.
- **/app/Http/Controllers/auth/ConfirmablePasswordController.php** - responsável pelos eventos de confirmação de senha.

- **/app/Http/Controllers/auth/EmailVerificationNotificationController.php** - responsável pelos eventos de verificação de e-mail do usuário que se registrou recentemente.
- **/app/Http/Controllers/auth/EmailVerificationPromptController.php**
- **/app/Http/Controllers/auth/NewPasswordController.php** - responsável pelos eventos de recebimento e atualização da nova senha do usuário.
- **/app/Http/Controllers/auth/PasswordResetLinkController.php** - responsável pelos eventos de solicitação de redefinição de senha - envio do link de reset.
- **/app/Http/Controllers/auth/RegisteredUserController.php** - responsável...
- **/app/Http/Controllers/auth/VerifyEmailController.php** - responsável pelos eventos de verificação de e-mail do usuário que se registrou recentemente.

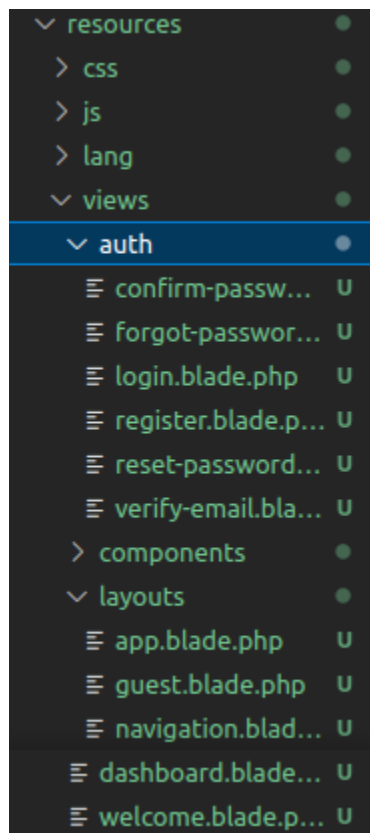


Arquivos Gerados (View)

- **/app/View/Components/AppLayout.php** - define um componente que será utilizado na renderização do layout da aplicação para usuários, com autenticação.
- **/app/View/Components/GuestLayout.php** - define um componente que será utilizado na renderização do layout da aplicação para visitantes, sem autenticação.



- **/resources/views/dashboard.blade.php** - tela apresentada logo após a autenticação ser efetuada com sucesso.
- **/resources/views/auth/confirm-password.blade.php** - tela utilizada para confirmação de senha.
- **/resources/views/forgot-password.blade.php** - tela utilizada para recuperação de senha, quando usuário a esqueceu.
- **/resources/views/auth/login.blade.php** - tela de autenticação (solicita e-mail e senha) apresentada ao usuário.
- **/resources/views/auth/register.blade.php** - tela de registro de novo usuário, apresentada quando um novo cadastro é solicitado.
- **/resources/views/auth/reset-password.blade.php** - tela que permite ao usuário redefinir sua nova senha.
- **/resources/views/auth/verify-email.blade.php** - tela que permite efetuar a confirmação de e-mail do usuário que se registrou recentemente.



Migração - Criação das Tabelas de Usuários

(No terminal - dentro da pasta do projeto) (Considerando que o “.env” já está configurado)

```
php artisan migrate
```

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/dwii/10
Arquivo Editar Ver Terminal Abas Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/10-Autenticação$ php artisan migrate
Migration table created successfully.
```

(Arquivo: “/database/migrations/..._create_users_table.php”)

```
...
// Arquivo Original - Não foi editado
public function up() {
    Schema::create('users', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('email')->unique();
        $table->timestamp('email_verified_at')->nullable();
        $table->string('password');
        $table->rememberToken();
        $table->timestamps();
    });
}
...

Arquivo de Migração “users” - Gerado durante a criação do projeto.

(Arquivo-fonte: 10-Autenticação/database/migrations/..._create_users_table.php)
```

Middleware Auth nas Rotas

(Arquivo: “/routes/web.php”)

```
...
Route::get('/dashboard', function () {
```

```
return view('dashboard');  
  
})->middleware(['auth']->name('dashboard'));  
  
require __DIR__ . '/auth.php';
```

Arquivo de Rotas - Middleware “auth” na rota “dashboard”.

[middleware('auth')] - Aplica o “auth” na rota “dashboard”

(Arquivo-fonte: 10-Authenticação/routes/web.php)

Obs.: Assim como visto na aula anterior (“Middleware”), também é possível aplicar o “auth” via definição no construtor da classe que manipula as rotas que devem ser restritas.

Atualizando os Arquivos de Layout do Breeze (removendo vite)

(Arquivo: “/resources/views/layouts/app.blade.php”)

```
...  
  
<head>  
  
  <meta charset="utf-8">  
  
  <meta name="viewport" content="width=device-width, initial-scale=1">  
  
  <meta name="csrf-token" content="{{ csrf_token() }}">  
  
  <title>{{ config('app.name', 'Laravel') }}</title>  
  
  <!-- Fonts -->  
  
  <link rel="stylesheet"  
href="https://fonts.googleapis.com/css2?family=Nunito:wght@400;600;700&display=swap">  
  
  <!-- Styles -->  
  
  <link rel="stylesheet" href="{{ asset('css/app.css') }}">  
  
  <!-- Scripts -->  
  
  <script src="{{ asset('js/app.js') }}" defer></script>
```

```
</head>
```

```
...
```

Arquivo de Layout do Breeze - Aplicação.

(Arquivo-fonte: 10-Authenticação/resources/views/layouts/app.blade.php)

(Arquivo: “/resources/views/layouts/guest.blade.php”)

```
...
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<meta name="csrf-token" content="{{ csrf_token() }}">
```

```
<title>{{ config('app.name', 'Laravel') }}</title>
```

```
<!-- Fonts -->
```

```
<link rel="stylesheet"
```

```
href="https://fonts.googleapis.com/css2?family=Nunito:wght@400;600;700&display=swap">
```

```
<!-- Styles -->
```

```
<link rel="stylesheet" href="{{ asset('css/app.css') }}">
```

```
<!-- Scripts -->
```

```
<script src="{{ asset('js/app.js') }}" defer></script>
```

```
</head>
```

```
...
```

Arquivo de Layout do Breeze - Convidado.

(Arquivo-fonte: 10-Authenticação/resources/views/layouts/guest.blade.php)

Executando e Visualizando a Aplicação

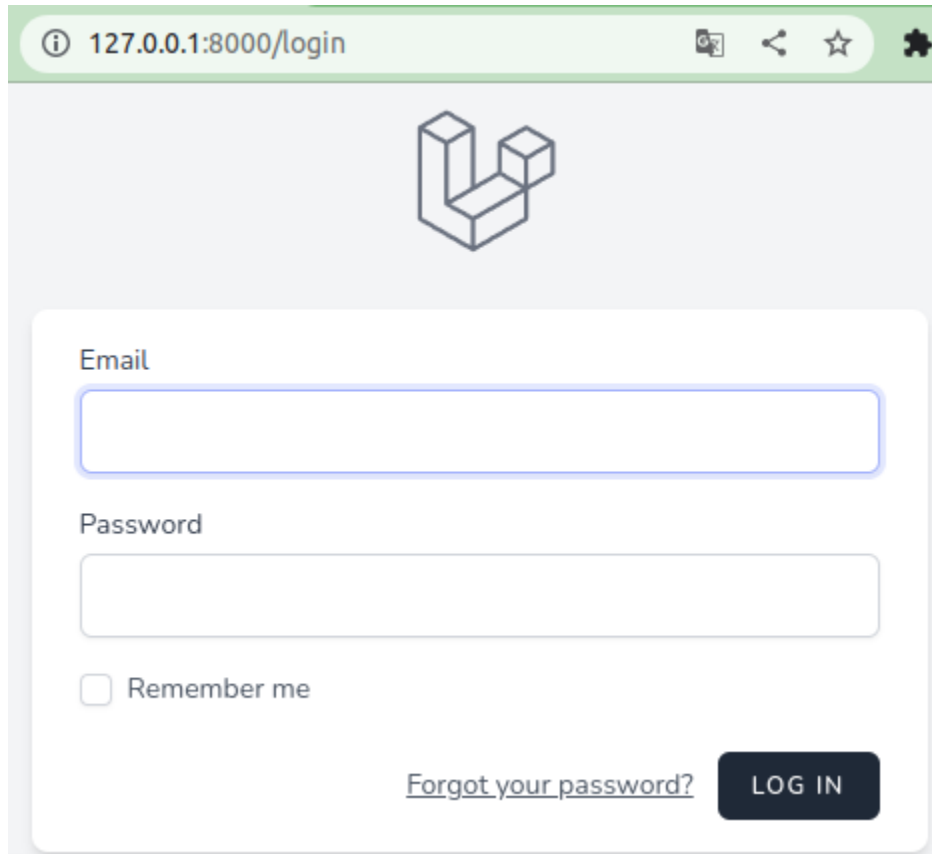
(No terminal - dentro da pasta do projeto)

```
php artisan serve
```



** Ao acessarmos a rota principal da aplicação já é possível visualizar as opções “Log in” e “Register” na parte superior direita da tela. As duas opções fazem parte dos recursos disponibilizados pelo pacote Breeze.*

Acessando a Rota “/dashboard” - Requer Autenticação

A screenshot of a web browser window showing a login form. The address bar displays "127.0.0.1:8000/login". The form includes an "Email" input field, a "Password" input field, a "Remember me" checkbox, a "Forgot your password?" link, and a "LOG IN" button.

Email

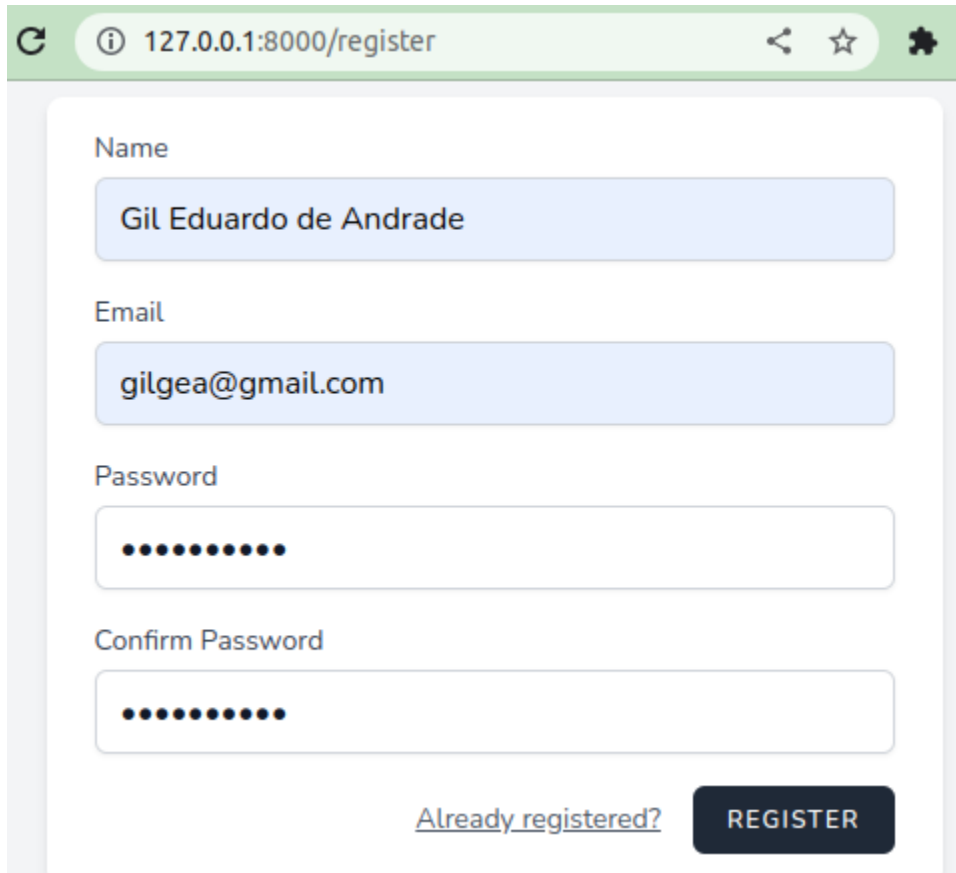
Password

Remember me

[Forgot your password?](#) **LOG IN**

** Ao acessarmos a rota “/dashboard” a aplicação é redirecionada, automaticamente, para tela de autenticação, rota “/login”. Isso ocorre porque o middleware “auth” foi aplicado a rota “/dashboard”, tornando necessária a autenticação do usuário para que o acesso a mesma seja permitido.*

Acessando a Rota de Registro de Usuário / Efetuando Registro



127.0.0.1:8000/register

Name
Gil Eduardo de Andrade

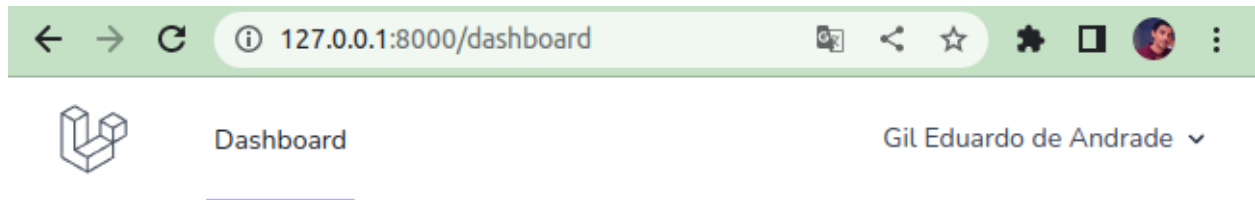
Email
gilgea@gmail.com

Password
.....

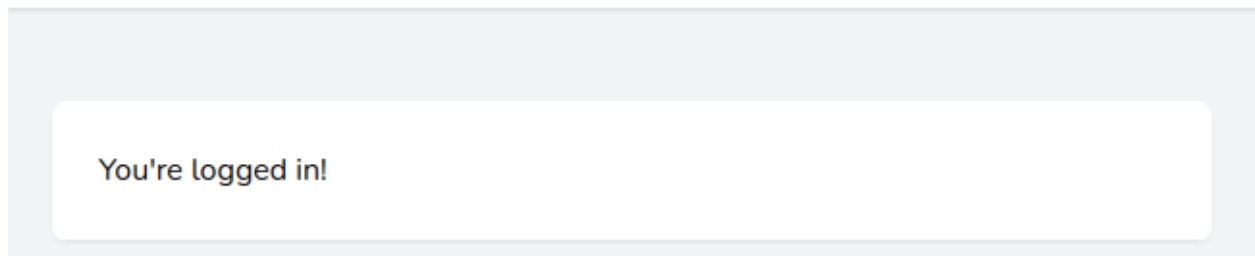
Confirm Password
.....

[Already registered?](#) REGISTER

(Execução da codificação anterior)

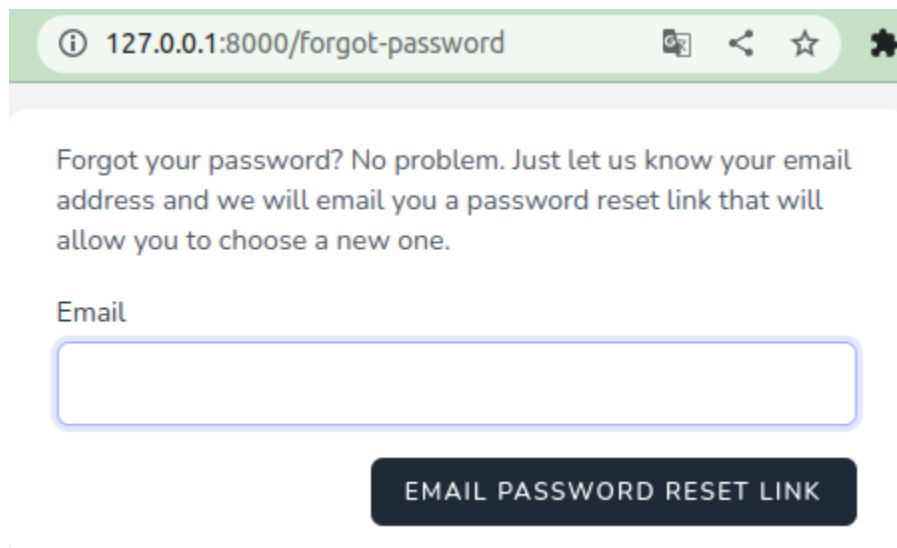


Dashboard



(Execução da codificação anterior)

Acessando a Rota “Esqueceu a sua Senha?” / “Forgot your password?”



(Execução da codificação anterior)

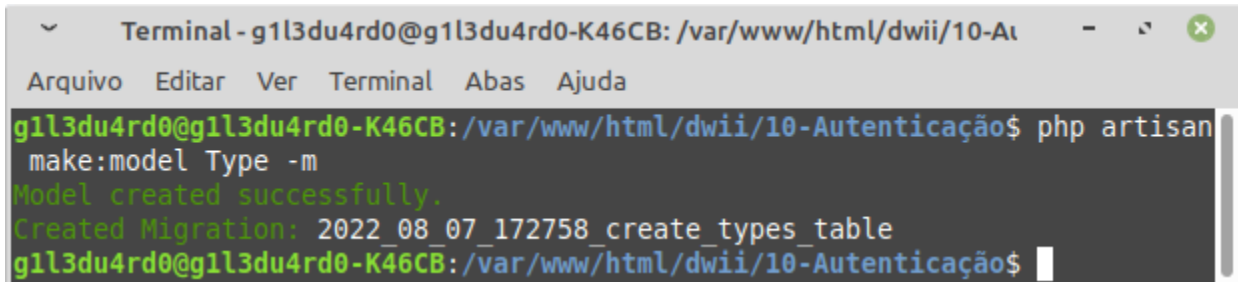
** Para que seja possível enviar, ao usuário, o e-mail de recuperação de senha, as configurações do serviço SMTP responsável para tal devem ser feitas no arquivo “.env”. Este procedimento será visto posteriormente na aula “Laravel: E-mails/Fila”.*

NESTA AULA, É DE CONHECIMENTO QUE O LARAVEL DISPONIBILIZA RECURSOS DE AUTORIZAÇÃO, TAL COMO “*POLICE*”, OBJETO DE ESTUDO DA PRÓXIMA AULA. OUTROS CONCEITOS IMPORTANTES COMO PAPÉIS, REGRAS E PERMISSÕES TAMBÉM SERÃO ABORDADOS POSTERIORMENTE. SENDO ASSIM, O OBJETIVO, NESTE MOMENTO, É TRABALHAR/PRATICAR O USO DOS RECURSOS DE AUTENTICAÇÃO ESTUDADOS.

Criando os Arquivos de Modelo/Migração “Types”

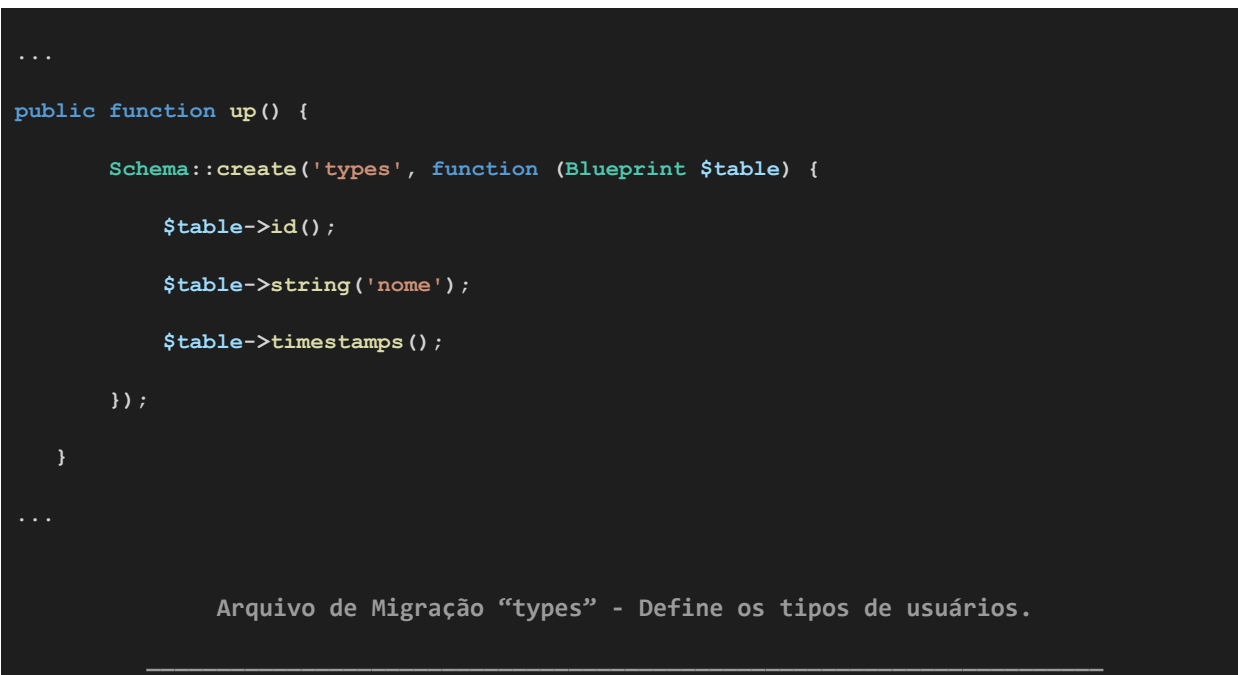
(No terminal - dentro da pasta do projeto)

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



```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/dwii/10-Au
Arquivo Editar Ver Terminal Abas Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/10-Autenticação$ php artisan
make:model Type -m
Model created successfully.
Created Migration: 2022_08_07_172758_create_types_table
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/10-Autenticação$
```

(Arquivo: “/database/migrations/..._create_types_table.php”)



```
...
public function up() {
    Schema::create('types', function (Blueprint $table) {
        $table->id();
        $table->string('nome');
        $table->timestamps();
    });
}
...

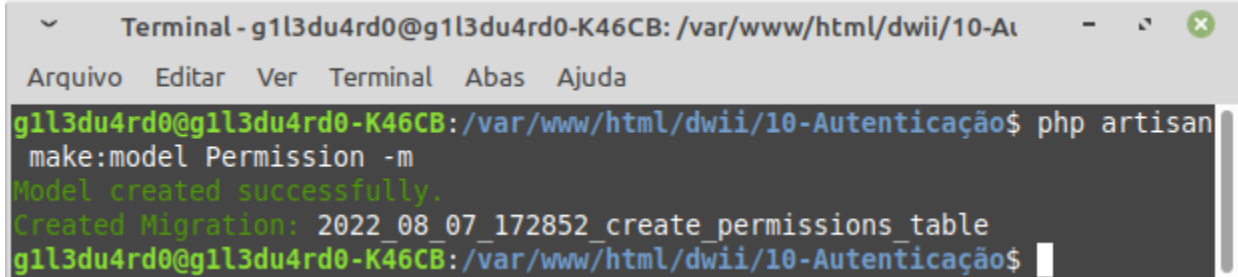
Arquivo de Migração “types” - Define os tipos de usuários.
```

(Arquivo-fonte: 10-Autenticação/database/migrations/..._create_types_table.php)

Criando os Arquivos de Migração “Permissions”

(No terminal - dentro da pasta do projeto)

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



```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/dwii/10-Au
Arquivo Editar Ver Terminal Abas Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/10-Autenticação$ php artisan
make:model Permission -m
Model created successfully.
Created Migration: 2022_08_07_172852_create_permissions_table
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/10-Autenticação$
```

(Arquivo: “/database/migrations/..._create_permissions_table.php”)

```
...
public function up() {
    Schema::create('permissions', function (Blueprint $table) {
        $table->id();
        $table->string('regra');
        $table->boolean('permissao');
        $table->unsignedBigInteger('type_id');
        $table->foreign('type_id')->references('id')->on('types');
        $table->timestamps();
    });
}
...
```

Arquivo de Migração “permissions” - Define as permissões para cada tipo de usuário.

(Arquivo-fonte: 10-Autenticação/database/migrations/..._create_permissions_table.php)

** É necessário renomear os arquivos de migração criados anteriormente, para que as tabelas sejam criadas no banco de dados antes da tabela de usuários (users).*

Alterando o Arquivo de Migração “Users”

(Arquivo: “/database/migrations/..._create_users_table.php”)

```
...  
  
public function up() {  
  
    Schema::create('users', function (Blueprint $table) {  
  
        $table->id();  
  
        $table->string('name');  
  
        $table->string('email')->unique();  
  
        $table->timestamp('email_verified_at')->nullable();  
  
        $table->string('password');  
  
        $table->unsignedBigInteger('type_id');  
  
        $table->foreign('type_id')->references('id')->on('types');  
  
        $table->rememberToken();  
  
        $table->timestamps();  
  
    });  
  
}  
  
...
```

Arquivo de Migração “users” - Gerado durante a criação do projeto.

(Arquivo-fonte: 10-Authenticação/database/migrations/..._create_users_table.php)

Refazendo a Migração

(No terminal - dentro da pasta do projeto)

```
php artisan migrate:fresh
```

```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/dwii/10-Au
Arquivo Editar Ver Terminal Abas Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/10-Autenticação$ php artisan
migrate:fresh
Dropped all tables successfully.
Migration table created successfully.
Migrating: 2014_08_07_172758_create_types_table
Migrated: 2014_08_07_172758_create_types_table (34.88ms)
Migrating: 2014_08_07_172852_create_permissions_table
Migrated: 2014_08_07_172852_create_permissions_table (123.63ms)
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (258.39ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (105.96ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (69.55ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (93.47ms)
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/10-Autenticação$
```

Criando Tipos de Usuários e Permissões Direto no Banco (Via phpMyAdmin)

+ Opções

	id	nome	created_at	updated_at
<input type="checkbox"/> Edita Copiar Apagar	1	PROFESSOR	NULL	NULL
<input type="checkbox"/> Edita Copiar Apagar	2	COORDENADOR	NULL	NULL
<input type="checkbox"/> Edita Copiar Apagar	3	DIRETOR	NULL	NULL

TABELA - TIPOS DE USUÁRIO

+ Opções












































		id	regra	permissao	type_id	created_at	updated_at
<input type="checkbox"/>	 Edita  Copiar  Apagar	6	cursos.index	1	1	NULL	NULL
<input type="checkbox"/>	 Edita  Copiar  Apagar	7	cursos.create	0	1	NULL	NULL
<input type="checkbox"/>	 Edita  Copiar  Apagar	8	cursos.edit	0	1	NULL	NULL
<input type="checkbox"/>	 Edita  Copiar  Apagar	9	cursos.show	1	1	NULL	NULL
<input type="checkbox"/>	 Edita  Copiar  Apagar	10	cursos.destroy	0	1	NULL	NULL
<input type="checkbox"/>	 Edita  Copiar  Apagar	11	cursos.index	1	2	NULL	NULL
<input type="checkbox"/>	 Edita  Copiar  Apagar	12	cursos.create	1	2	NULL	NULL
<input type="checkbox"/>	 Edita  Copiar  Apagar	13	cursos.edit	0	2	NULL	NULL
<input type="checkbox"/>	 Edita  Copiar  Apagar	14	cursos.show	1	2	NULL	NULL
<input type="checkbox"/>	 Edita  Copiar  Apagar	15	cursos.destroy	0	2	NULL	NULL
<input type="checkbox"/>	 Edita  Copiar  Apagar	16	cursos.index	1	3	NULL	NULL
<input type="checkbox"/>	 Edita  Copiar  Apagar	17	cursos.create	1	3	NULL	NULL
<input type="checkbox"/>	 Edita  Copiar  Apagar	18	cursos.edit	1	3	NULL	NULL
<input type="checkbox"/>	 Edita  Copiar  Apagar	19	cursos.show	1	3	NULL	NULL
<input type="checkbox"/>	 Edita  Copiar  Apagar	20	cursos.destroy	1	3	NULL	NULL

TABELA - PERMISSÕES

Alterando a Model “User”

(Arquivo: “/app/Models/User.php”)

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

Arquivo de Modelo “User” - Adicionando Campo ‘type_id’ - fillable.

(Arquivo-fonte: 10-Autenticação/app/Models/User.php)

Alterando a Controller “Users”

(Arquivo: “/app/Http/Controllers/Auth/RegisteredUserController.php”)

```
...  
use App\Models\Type;  
...  
public function create() {  
    $types = Type::orderBy('nome')->get();  
    return view('auth.register', compact('types'));  
}  
...  
public function store(Request $request) {  
...  
    $user = User::create([  
        'name' => $request->name,  
        'email' => $request->email,  
        'type_id' => $request->type_id,  
        'password' => Hash::make($request->password),  
    ]);  
...  
}
```

Arquivo de Controle “users” - Adicionando Campo ‘type_id’ - Método “create()”.

(Arquivo-fonte: 10-Autenticação/app/Http/Controllers/Auth/RegisteredUserController.php)

Alterando a View “Users”

(Arquivo: “/resources/views/auth/register.blade.php”)

```
...  
  
<!-- User Type -->  
  
<div class="mt-4">  
  
    <select name="type_id" class="rounded-md shadow-sm border-gray-300  
focus:border-indigo-300 focus:ring focus:ring-indigo-200 focus:ring-opacity-50">  
  
        @foreach($types as $item)  
  
            <option value="{{ $item->id }}">  
  
                {{ $item->nome }}  
  
            </option>  
  
        @endforeach  
  
    </select>  
  
</div>  
  
...
```

Arquivo de Visualização "Register" - Adicionando <select> Type.

(Arquivo-fonte: 10-Authenticação/resources/views/auth/register.blade.php)

Name

Gil Eduardo de Andrade

Email

gilgea@gmail.com

Password

.....

Confirm Password

.....

COORDENADOR ▾

[Already registered?](#)

REGISTER

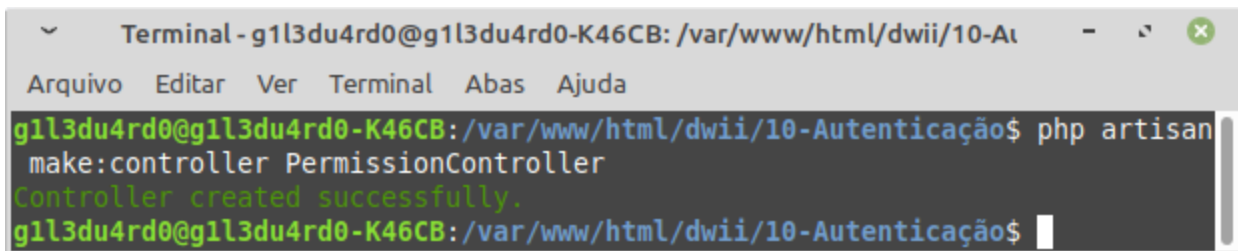
(Execução da codificação anterior)

id	name	email	type_id
1	Gil Eduardo de Andrade	gilgea@gmail.com	2

Criando uma Classe para Gerenciar as Permissões do Usuário

(No terminal - dentro da pasta do projeto)

```
php artisan make:controller PermissionController
```



```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/dwii/10-Au
Arquivo  Editar  Ver  Terminal  Abas  Ajuda
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/10-Autenticação$ php artisan
make:controller PermissionController
Controller created successfully.
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/10-Autenticação$
```

(Arquivo: "/app/Http/Controllers/PermissionController.php")

```
...
use App\Models\Permission;
...

public static function loadPermissions($user_type) {

    $sess = Array();

    $perm = Permission::where('type_id', $user_type)->get();

    foreach($perm as $item) {

        $sess[$item->regra] = (boolean) $item->permissao;

    }

    session(['user_permissions' => $sess]);

}

public static function isAuthorized($rule) {

    $permissions = session('user_permissions');

    return $permissions[$rule];

}
```

...

Classe de Controle - Permissões do Usuário.

(Arquivo-fonte: 10-Autenticação/app/Http/Controllers/PermissionController.php)

Carregando as Permissões no Momento da Autenticação

(Arquivo: "/app/Http/Controllers/Auth/AuthenticaredSessionController.php")

...

```
use App\Models\Permission;
```

...

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

...

```
// Carregando as Permissões do Usuário / Sessão
```

```
PermissionController::loadPermissions(Auth::user()->type_id);
```

```
return redirect()->intended(RouteServiceProvider::HOME);
```

...

Classe de Controle "Sessão Autenticado" - Carregando Permissões do Usuário.

(Arquivo-fonte: 10-Autenticação/app/Http/Controllers/Auth/AuthenticatedSessionController.php)

Carregando as Permissões no Momento do Registro

(Arquivo: "/app/Http/Controllers/Auth/RegisteredUserController.php")

...

```
use App\Models\Permission;
```

...

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

```
...  
  
    // Carregando as Permissões do Usuário / Sessão  
  
    PermissionController::loadPermissions(Auth::user()->type_id);  
  
    return redirect(RouteServiceProvider::HOME);  
  
...
```

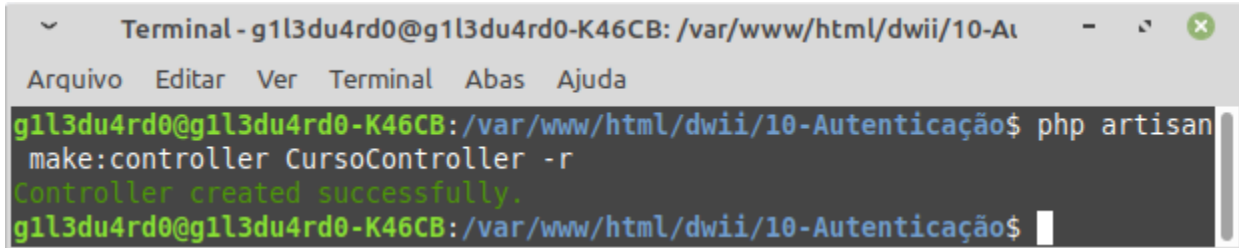
Classe de Controle “Usuário Registrado” - Carregando Permissões do Usuário.

(Arquivo-fonte: 10-Authenticação/app/Http/Controllers/AuthRegisteredUserController.php)

Criando a Controle para Cursos

(No terminal - dentro da pasta do projeto)

```
php artisan make:controller CursoController -r
```



```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/dwii/10-Authenticação  
Arquivo Editar Ver Terminal Abas Ajuda  
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/10-Authenticação$ php artisan  
make:controller CursoController -r  
Controller created successfully.  
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/10-Authenticação$
```

Codificando a Classe “CursoController”

(Arquivo: “/app/Http/Controllers/CursoController.php”)

```
...  
  
    public function index() {  
  
        $permissions = session('user_permissions');  
  
        return view('cursos.index', compact('permissions'));  
  
    }  
  
    public function create() {  
  
        return view('cursos.create');  
  
    }  
  
}
```

```
public function store(Request $request) {  
    }  
  
public function show($id) {  
    return view('cursos.show');  
}  
  
public function edit($id) {  
    return view('cursos.edit');  
}  
  
public function update(Request $request, $id) {  
    }  
  
public function destroy($id) {  
    return view('cursos.destroy');  
}  
...  
...  
...
```

Classe de Controle dos Cursos - Codificando os Métodos.

(Arquivo-fonte: 10-Authenticação/app/Http/Controllers/CursoController.php)

Especificando as Rotas

(Arquivo: "/routes/web.php")

```
...  
  
Route::get('/dashboard', function () {  
    return view('templates.middleware')->with('titulo', "");  
})->middleware(['auth'])->name('dashboard');  
  
Route::resource('/cursos', '\App\Http\Controllers\CursoController')  
->middleware(['auth']);
```

...

Arquivo de Rotas - Especificando as Rotas.

(Arquivo-fonte: 10-Autenticação/routes/web.php)

** Os arquivos de visualização utilizados nesse momento, para cursos, são os mesmos disponibilizados na aula anterior - por isso serão usados/copiados sem necessidade de codificação.*

Desabilitando as Ações na View Index Cursos / Considerando as Permissões

(Arquivo: "/resources/views/cursos/index.blade.php")

...

```
@extends('templates.middleware', [  
    'titulo' => "Cursos",  
    'rota' => "cursos.create",  
    'permission' => $permissions['cursos.create']  
)  
...  
<td>TÉCNICO EM INFORMÁTICA</td>  
<td class="d-none d-md-table-cell">ENSINO MÉDIO INTEGRADO</td>  
<td>  
    @if($permissions['cursos.edit'])  
        <a href= "{{ route('cursos.edit', '1') }}" class="btn btn-success">  
            ...  
        </a>  
    @endif  
    @if($permissions['cursos.show'])  
        <a href= "{{ route('cursos.show', '1') }}" class="btn btn-primary">  
            ...  
    @endif  
...
```



```
</a>

@endif

@if($permissions['cursos.destroy'])

    <a nohref style="cursor:pointer" onclick="document.getElementById('form_1').submit()" class="btn btn-danger">

        ...

    </a>

@endif

</td>

...

Arquivo de Visualização “Cursos Index” - Verificando Permissões.



---



(Arquivo-fonte: 10-Authenticação/resources/views/cursos/index.blade.php)
```

(Arquivo: “/resources/views/templates/middleware.blade.php”)

```
...

@if(isset($rota) && $permission)

    <div class="col d-flex justify-content-end">

        <a href= "{{ route($rota) }}" class="btn btn-secondary">

            ...

        </a>

    </div>

@endif

...

Arquivo de Visualização “Template” - Verificando Permissões.



---




(Arquivo-fonte: 10-Authenticação/resources/views/templates/middleware.blade.php)
```



Cursos

Tabela de Cursos

Nome	Descrição	Ações
TÉCNICO EM INFORMÁTICA	ENSINO MÉDIO INTEGRADO	

(Execução da codificação anterior)

Bloqueando Acesso Direto as Rotas - “CursoController”

(Arquivo: “/app/Http/Controllers/CursoController.php”)

```
...  
  
public function index() {  
  
    if(!PermissionController::isAuthorized('cursos.index')) {  
  
        abort(403);  
  
    }  
  
    $permissions = session('user_permissions');  
  
    return view('cursos.index', compact('permissions'));  
  
}  
  
public function create() {  
  
    if(!PermissionController::isAuthorized('cursos.create')) {  
  
        abort(403);  
  
    }  
  
    return view('cursos.create');  
  
}  
  
public function store(Request $request) {
```

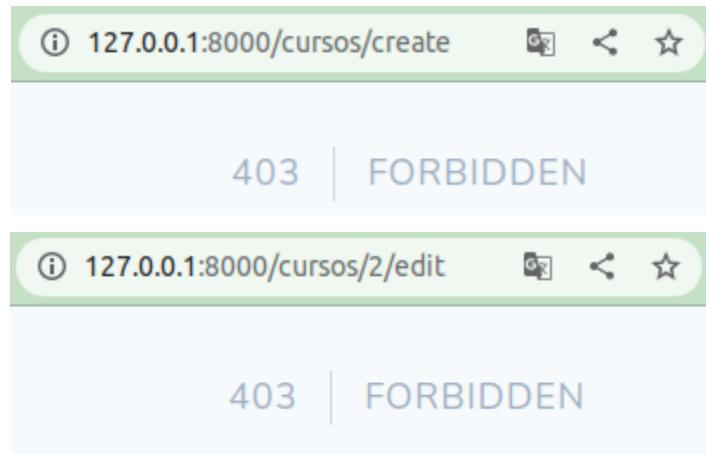


```
}  
  
public function show($id) {  
  
    if(!PermissionController::isAuthorized('cursos.show')) {  
  
        abort(403);  
  
    }  
  
    return view('cursos.show');  
  
}  
  
public function edit($id) {  
  
    if(!PermissionController::isAuthorized('cursos.edit')) {  
  
        abort(403);  
  
    }  
  
    return view('cursos.edit');  
  
}  
  
public function update(Request $request, $id) {  
  
}  
  
public function destroy($id) {  
  
    if(!PermissionController::isAuthorized('cursos.destroy')) {  
  
        abort(403);  
  
    }  
  
    return view('cursos.destroy');  
  
}  
  
...  

```

Classe de Controle dos Cursos - Recusando Acesso Direto as Rotas.

(Arquivo-fonte: 10-Authenticação/app/Http/Controllers/CursoController.php)



(Execução da codificação anterior)

Criando uma “Facade” para Gerenciar as Permissões

Criando a Classe

(Arquivo: “/app/Facades/UserPermissions.php”)

```
namespace App\Facades;

use App\Models\Permission;

class UserPermissions {

    public static function loadPermissions($user_type) {

        $sess = Array();

        $perm = Permission::where('type_id', $user_type)->get();

        foreach($perm as $item) {

            $sess[$item->regra] = (boolean) $item->permissao;

        }

        session(['user_permissions' => $sess]);

    }

    public static function isAuthorized($rule) {

        $permissions = session('user_permissions');

        return $permissions[$rule];

    }

}
```

```
}  
  
public static function test() {  
  
    return "<h1>UserPermissionsFacade - Running!!</h1>";  
  
}  
  
}
```

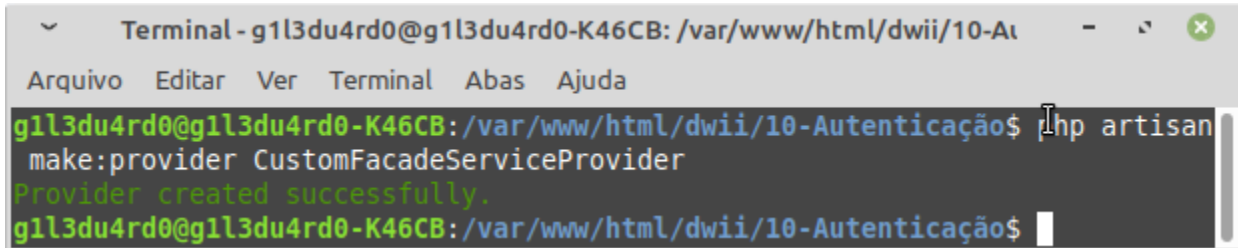
Classe "UserPermissions".

(Arquivo-fonte: 10-Autenticação/app/Facades/UserPermissions.php)

Criando ServiceProvider

(No terminal - dentro da pasta do projeto)

```
php artisan make:provider CustomFacadeServiceProvider
```



```
Terminal - g1l3du4rd0@g1l3du4rd0-K46CB: /var/www/html/dwii/10-Au  
Arquivo Editar Ver Terminal Abas Ajuda  
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/10-Autenticação$ php artisan  
make:provider CustomFacadeServiceProvider  
Provider created successfully.  
g1l3du4rd0@g1l3du4rd0-K46CB:/var/www/html/dwii/10-Autenticação$
```

Vinculando o Facade no ServiceProvider Criado

(Arquivo: "/app/Providers/CustomFacadeServiceProvider.php")

```
use App\Facades\UserPermissions;  
  
...  
  
public function register() {  
  
    $this->app->bind('userpermissions',function() {  
  
        return new UserPermissions();  
  
    });  
  
}  
  
...  
  
}
```

Classe Facade “UserPermissions”.

(Arquivo-fonte: 10-Autenticação/app/Facades/UserPermissions.php)

Registrando o ServiceProvider Criado

(Arquivo: “/config/app.php”)

```
...  
  
'providers' => [  
  
    ...  
  
    /*  
     * Application Service Providers...  
     */  
  
    App\Providers\AppServiceProvider::class,  
  
    App\Providers\AuthServiceProvider::class,  
  
    // App\Providers\BroadcastServiceProvider::class,  
  
    App\Providers\EventServiceProvider::class,  
  
    App\Providers\RouteServiceProvider::class,  
  
    App\Providers\CustomFacadeServiceProvider::class,  
  
],  
  
...
```

Registrando o ServiceProvider Criado - CustomFacadesServiceProvider.

(Arquivo-fonte: 10-Autenticação/app/config.php)

Criando a Classe Facade

(Arquivo: “/app/Facades/UserPermissionsFacade.php”)

```
<?php  
  
namespace App\Facades;
```

```
use Illuminate\Support\Facades\Facade;

class UserPermissionsFacade extends Facade {

    protected static function getFacadeAccessor() {

        return 'userpermissions'; // mesmo nome do método vinculado no service provider

    }

}
```

Classe Facade - “UserPermissionsFacade”.

(Arquivo-fonte: 10-Authenticação/app/Facades/UserPermissionsFacade.php)

Testando o Facade Criado

(Arquivo: “/routes/web.php”)

```
...

Route::get('/testfacade', function () {

    return App\Facades\UserPermissionsFacade::test();

});

...
```

Arquivo de Rotas - “web.php”.

(Arquivo-fonte: 10-Authenticação/routes/web.php)



← → ↻ ⓘ 127.0.0.1:8000/testfacade

UserPermissionsFacade - Running!!

(Execução da codificação anterior)

Definindo um “Alias” (apelido) para Classe Facade

(Arquivo: “/config/app.php”)

```
...  
  
'aliases' => [  
  
    'App' => Illuminate\Support\Facades\App::class,  
  
    ...  
  
    'View' => Illuminate\Support\Facades\View::class,  
  
    'UserPermissions' => App\Facades\UserPermissionsFacade::class,  
  
],  
  
...
```

Criando um “Alias” - UserPermissionsFacade.

(Arquivo-fonte: 10-Authenticação/config/app.php)

Testando o Alias Definido

(Arquivo: “/routes/web.php”)

```
...  
  
Route::get('/testfacade', function () {  
  
    return UserPermissions::test();  
  
});  
  
...
```

Arquivo de Rotas - “web.php”.

(Arquivo-fonte: 10-Authenticação/routes/web.php)



← → ↻ ⓘ 127.0.0.1:8000/testfacade

UserPermissionsFacade - Running!!

(Execução da codificação anterior)

Utilizando Facade - Carregando as Permissões Durante a Autenticação

(Arquivo: “/app/Http/Controllers/Auth/AuthenticatedSessionController.php”)

```
...  
use App\Facades\UserPermissions;  
...  
public function store(LoginRequest $request) {  
...  
    // Carregando as Permissões do Usuário / Sessão  
    UserPermissions::loadPermissions(Auth::user()->type_id);  
...  
}
```

Utilizando Facade - Carregando Permissões do Usuário.

(Arquivo-fonte: 10-Autenticação/app/Http/Controllers/Auth/AuthenticatedSessionController.php)

Utilizando Facade - Carregando as Permissões Durante o Registro

(Arquivo: “/app/Http/Controllers/Auth/RegisteredUserController.php”)

```
...  
use App\Facades\UserPermissions;  
...  
public function store(Request $request) {  
...  
    // Carregando as Permissões do Usuário / Sessão  
    UserPermissions::loadPermissions(Auth::user()->type_id);  
...  
}
```

Utilizando Facade - Carregando Permissões do Usuário.

(Arquivo-fonte: 10-Autenticação/app/Http/Controllers/Auth/AuthenticatedSessionController.php)

Utilizando Facade - “CursoController”

(Arquivo: “/app/Http/Controllers/CursoController.php”)

```
use App\Facades\UserPermissions;

...

public function index() {

    if(!UserPermissions::isAuthorized('cursos.index')) {

        abort(403);

    }

    return view('cursos.index');

}

public function create() {

    if(!UserPermissions::isAuthorized('cursos.create')) {

        abort(403);

    }

    return view('cursos.create');

}

public function store(Request $request) {

}

public function show($id) {

    if(!UserPermissions::isAuthorized('cursos.show')) {

        abort(403);

    }

    return view('cursos.show');

}

public function edit($id) {
```



```
@endif

@if(UserPermissions::isAuthorized('cursos.show'))

    <a href= "{{ route('cursos.show', '1') }}" class="btn btn-primary">

        ...

    </a>

@endif

@if(UserPermissions::isAuthorized('cursos.destroy'))

    <a nohref style="cursor:pointer" onclick="document.getElementById('form_1').submit()" class="btn btn-danger">

        ...

    </a>

@endif

...
```

Arquivo de Visualização “cursos.index” - Verificando Permissões.

(Arquivo-fonte: 10-Authenticação/resources/views/cursos/index.blade.php)

Utilizando Facade - “View templates.middleware”

(Arquivo: “/resources/views/templates/middleware.blade.php”)

```
...

@if(isset($rota) && UserPermissions::isAuthorized('cursos.create'))

    <div class="col d-flex justify-content-end">

        <a href= "{{ route($rota) }}" class="btn btn-secondary">

            ...

        </a>

    </div>

@endif

...
```

Arquivo de Visualização "Template" - Verificando Permissões.

(Arquivo-fonte: 10-Autenticação/resources/views/templates/middleware.blade.php)