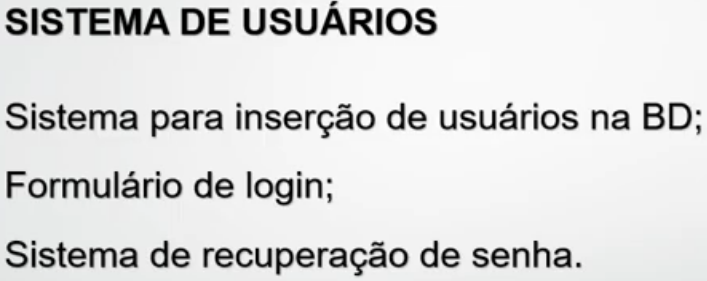
**Curso Laravel 5.4 - Aulas 68 a 84**

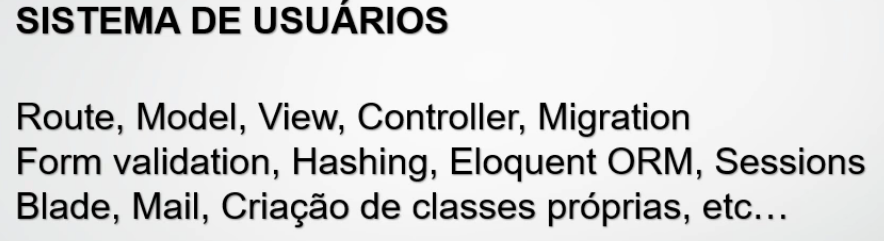
**SYS4SOFT (João Ribeiro)**

<https://www.youtube.com/watch?v=0T5gM1WRNsY&list=PLXik_5Br-zO893qVjjP7a4qg4NYrl33w1&index=1>

Resumo do curso feito por Roberto Pinheiro

# Aula 068 – Exercício 2 - Sistema de usuários - Parte 1







No Laragon, clicar com o botão direito na área de trabalho e selecionar "Criar rapidamente um website" 🡪 "Laravel (zip)". Digitar como nome do projeto: SistemaUsuarios e clicar no botão Ok.

No Sublime Text, apague as duas migrations padrões e na pasta app apague o arquivo User.php.

## Ocultando pastas no Sublime Text

Em Preferences 🡪 Settings:

{

"color\_scheme": "Packages/Color Scheme - Default/Monokai.sublime-color-scheme",

"font\_size": 14,

"ignored\_packages":

[

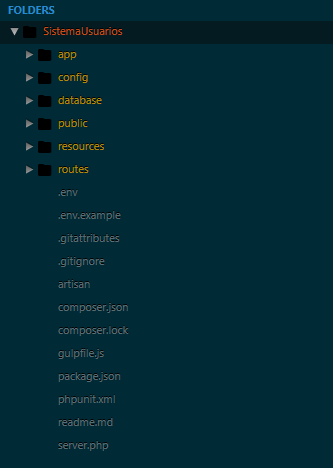
"Vintage"

],

"theme": "Soda SolarizedDark.sublime-theme",

**"folder\_exclude\_patterns":["vendor","tests","storage","bootstrap"]**

}



# Aula 069 – Exercício 2 - Sistema de usuários - Parte 2

## Configurando o acesso a base de dados

**.env**

DB\_CONNECTION=mysql

DB\_HOST=localhost

DB\_PORT=3306

DB\_DATABASE=sistemausuarios

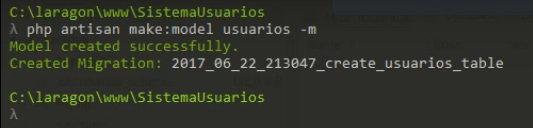
DB\_USERNAME=root

DB\_PASSWORD=

## Criando o model e a migration

No terminal crie o model e a migration, entrando com o seguinte comando:

**php artisan make:model usuarios -m**



**database/migrations/2017\_06\_22\_213047\_create\_usuarios\_table.php**

<?php

use Illuminate\Support\Facades\Schema;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Database\Migrations\Migration;

class CreateUsuariosTable extends Migration

{

public function up()

{

Schema::create('usuarios', function (Blueprint $table) {

$table->increments('id\_usuario');

$table->string('usuario');

$table->string('senha');

$table->string('email');

$table->rememberToken();

$table->timestamps();

});

}

public function down()

{

Schema::dropIfExists('usuarios');

}

}

**app/usuarios.php**

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class usuarios extends Model

{

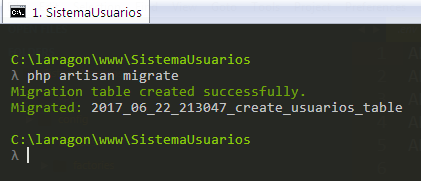
//alterar a identificação da PK

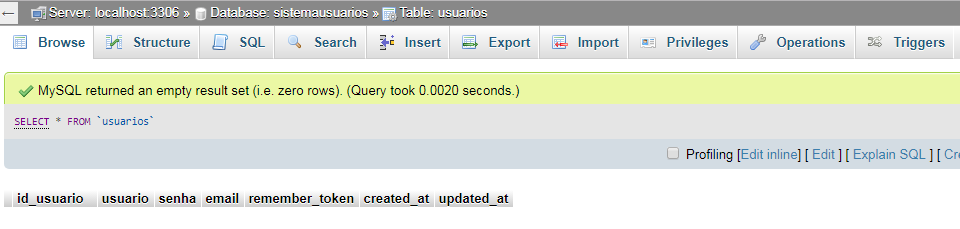
protected $primaryKey = 'id\_usuario';

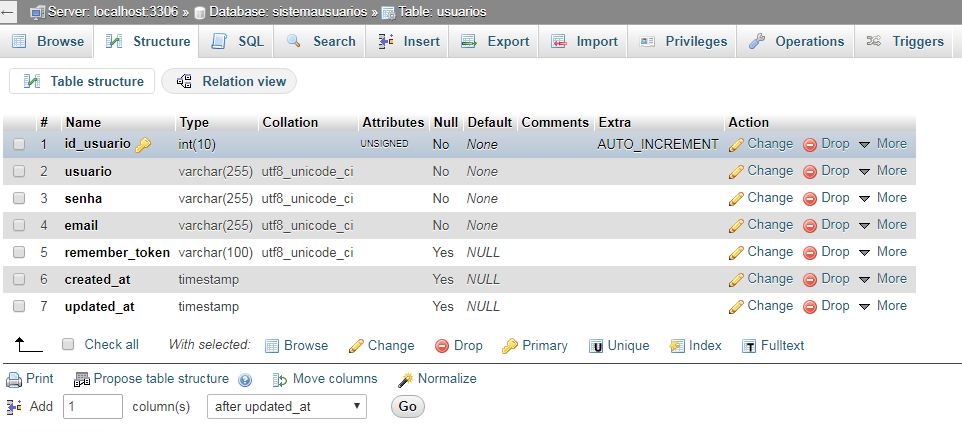
}

## Criando a tabela

**php artisan migrate**







## Route

**routes/web.php**

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

return view('teste');

});

## Criando a master page

**resources/views/layouts/app.blade.php**

<!DOCTYPE html>

<html lang="pt-br">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

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

<!-- As 3 meta tags acima \*devem\* vir em primeiro lugar dentro do `head`; qualquer outro conteúdo deve vir \*após\* essas tags -->

<title>Sistema de Usuários</title>

<!-- Bootstrap -->

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

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

<!-- HTML5 shim e Respond.js para suporte no IE8 de elementos HTML5 e media queries -->

<!-- ALERTA: Respond.js não funciona se você visualizar uma página file:// -->

<!--[if lt IE 9]>

<script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>

<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>

<![endif]-->

</head>

<body>

<div class="container">

{{-- header --}}

@include('inc.header')

{{-- conteúdo --}}

@yield('conteudo')

</div>

<!-- jQuery (obrigatório para plugins JavaScript do Bootstrap) -->

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>

<!-- Inclui todos os plugins compilados (abaixo), ou inclua arquivos separadados se necessário -->

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

</body>

</html>

**resources/views/inc/header.blade.php**

<div class="row">

<div class="col-md-4 col-md-offset-4 col-sm-8 col-sm-offset-2 col-xs-12">

<img src="{{ asset('images/logo.png') }}" class="img-responsive img-logo">

</div>

</div>

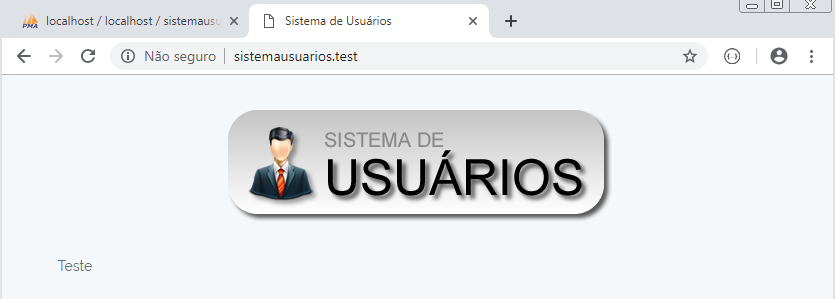
**resources/views/teste.blade.php**

@extends('layouts.app')

@section('conteudo')

<p>Teste</p>

@endsection



# Aula 070 – Exercício 2 - Sistema de usuários - Parte 3

**routes/web.php**

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

return view('usuario\_frm\_login');

});

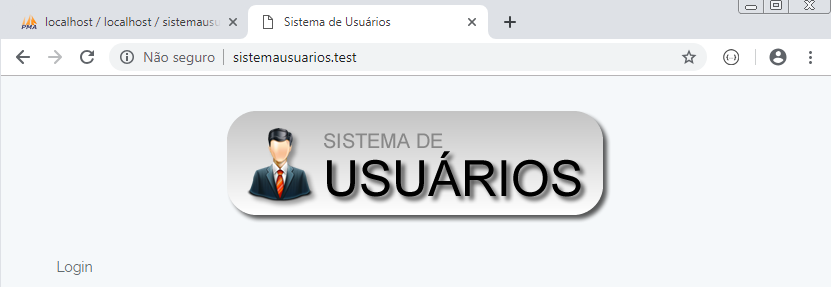
**resources/views/usuario\_frm\_login.blade.php**

@extends('layouts.app')

@section('conteudo')

<p>Login</p>

@endsection



**public/css/main.css**

/\*o meu código CSS\*/

.img-logo{

margin: auto;

margin-top: 30px;

margin-bottom: 30px;

}

**resources/views/usuario\_frm\_login.blade.php**

@extends('layouts.app')

{{-- FORMULÁRIO DE LOGIN --}}

@section('conteudo')

<div class="row">

<div class="col-md-4 col-md-offset-4 col-sm-8 col-sm-offset-2 col-xs-12">

<form method="POST" action="/usuario\_executar\_login">

{{-- csrf --}}

{{ csrf\_field() }}

{{-- usuario --}}

<div class="form-group">

<label for="id\_text\_usuario">Usuário:</label>

<input type="text" class="form-control" id="id\_text\_usuario" name="text\_usuario" placeholder="Usuário">

</div>

{{-- senha --}}

<div class="form-group">

<label for="id\_text\_senha">Senha:</label>

<input type="password" class="form-control" id="id\_text\_senha" name="text\_senha" placeholder="Senha">

</div>

{{-- submeter --}}

<div class="text-center">

<button type="submit" class="btn btn-primary">Entrar</button>

</div>

{{-- link para recuperar senha --}}

<div class="text-center margem-top-20">

<a href="# ">Recuperar senha</a>

</div>

{{-- link para criar nova conta de usuário --}}

<div class="text-center">

<a href="#">Criar nova conta</a>

</div>

</form>

</div>

</div>

@endsection



# Aula 071 – Exercício 2 - Sistema de usuários - Parte 4

## Formulário para criar nova conta

**routes/web.php**

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

return view('usuario\_frm\_login');

});

Route::get('/criar\_conta', function(){

return view('usuario\_frm\_criar\_conta');

});

**resources/views/usuario\_frm\_login.blade.php**

{{-- link para recuperar senha --}}

<div class="text-center margem-top-20">

<a href="/recuperar\_senha">Recuperar senha</a>

</div>

{{-- link para criar nova conta de usuário --}}

<div class="text-center">

<a href="/criar\_conta">Criar nova conta</a>

</div>

**resources/views/usuario\_frm\_criar\_conta.blade.php**

@extends('layouts.app')

{{-- CRIAR NOVA CONTA DE USUÁRIO --}}

@section('conteudo')

<div class="row">

<div class="col-md-4 col-md-offset-4 col-sm-8 col-sm-offset-2 col-xs-12">

{{-- apresentação de erros de validação... e não só --}}

{{-- ========================================================= --}}

<form method="POST" action="/usuario\_executar\_criar\_conta">

{{-- csrf --}}

{{ csrf\_field() }}

{{-- usuario --}}

<div class="form-group">

<label for="id\_text\_usuario">Usuário:</label>

<input type="text" class="form-control" id="id\_text\_usuario" name="text\_usuario" placeholder="Usuário">

</div>

{{-- senha --}}

<div class="form-group">

<label for="id\_text\_senha">Senha:</label>

<input type="password" class="form-control" id="id\_text\_senha" name="text\_senha" placeholder="Senha">

</div>

{{-- repetir senha --}}

<div class="form-group">

<label for="id\_text\_senha\_repetida">Repetir senha:</label>

<input type="password" class="form-control" id="id\_text\_senha\_repetida" name="text\_senha\_repetida" placeholder="Repetir senha">

</div>

{{-- email --}}

<div class="form-group">

<label for="id\_text\_email">Email:</label>

<input type="text" class="form-control" id="id\_text\_email" name="text\_email" placeholder="Email">

</div>

{{-- aceitação das condições do serviço --}}

<div class="form-group text-center">

<input type="checkbox" id="id\_check\_termos\_condicoes" name="check\_termos\_condicoes" value="1">

<label for="id\_check\_termos\_condicoes"> Concordo com os termos e condições.</label>

</div>

{{-- submeter --}}

<div class="text-center">

<button type="submit" class="btn btn-primary">Criar nova conta</button>

</div>

{{-- voltar ao início --}}

<div class="text-center margem-top-20">

<a href="/">Voltar</a>

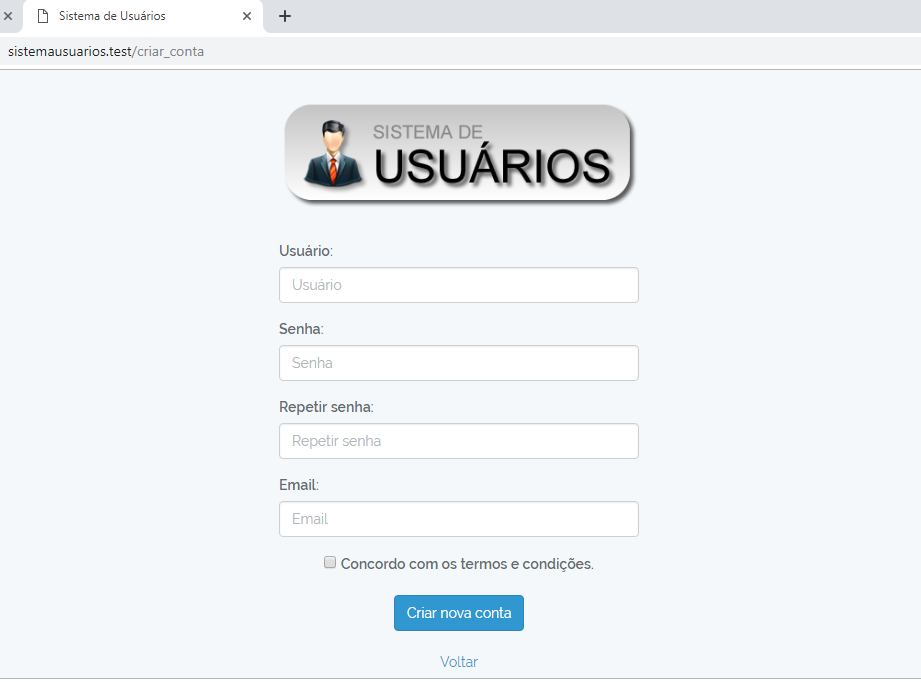
</div>

</form>

</div>

</div>

@endsection



# Aula 072 – Exercício 2 - Sistema de usuários - Parte 5

## Formulário para recuperar senha

**routes/web.php**

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

return view('usuario\_frm\_login');

});

Route::get('/criar\_conta', function(){

return view('usuario\_frm\_criar\_conta');

});

Route::get('/recuperar\_senha', function(){

return view('usuario\_frm\_recuperar\_senha');

});

**resources/views/usuario\_frm\_recuperar\_senha.blade.php**

@extends('layouts.app')

{{-- RECUPERAR SENHA --}}

@section('conteudo')

<div class="row">

<div class="col-md-4 col-md-offset-4 col-sm-8 col-sm-offset-2 col-xs-12">

{{-- apresentação de erros de validação --}}

{{-- ========================================================= --}}

<form method="POST" action="/usuario\_executar\_recuperar\_senha">

{{-- csrf --}}

{{ csrf\_field() }}

{{-- email --}}

<div class="form-group">

<label for="id\_text\_email">Email:</label>

<input type="text" class="form-control" id="id\_text\_email" name="text\_email" placeholder="Email">

</div>

{{-- submeter --}}

<div class="text-center">

<button type="submit" class="btn btn-primary">Recuperar senha</button>

</div>

{{-- voltar ao início --}}

<div class="text-center margem-top-20">

<a href="/">Voltar</a>

</div>

</form>

</div>

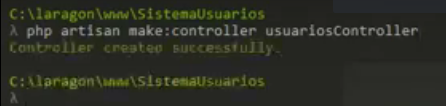
</div>

@endsection



## Criando o controller

**php artisan make:controller usuariosController**



**routes/web.php**

//------------------------------------------------------------

Route::get('/', 'usuariosController@index');

//------------------------------------------------------------

// usuario - login

//------------------------------------------------------------

// usuario - recuperar senha

Route::get('/usuario\_frm\_recuperar\_senha', 'usuariosController@frmRecuperarSenha');

//------------------------------------------------------------

// usuario - nova conta

Route::get('/usuario\_frm\_criar\_conta', 'usuariosController@frmCriarNovaConta');

**app/Http/Controllers/usuariosController.php**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Requests;

use Illuminate\Support\Facades\Mail;

class usuariosController extends Controller

{

//================================================================

public function index(){

return view('usuario\_frm\_login');

}

//================================================================

// LOGIN

//================================================================

//================================================================

// RECUPERAR SENHA

//================================================================

public function frmRecuperarSenha(){

//apresentar o formulário para recuperação da senha

return view('usuario\_frm\_recuperar\_senha');

}

//================================================================

public function executarRecuperarSenha(Request $request){

//================================================================

// CRIAÇÃO DE NOVA CONTA

//================================================================

public function frmCriarNovaConta(){

//apresentar o formulário de criação de nova conta

return view('usuario\_frm\_criar\_conta');

}

//================================================================

public function executarCriarNovaConta(Request $request){

}

**resources/views/usuario\_frm\_login.blade.php**

{{-- link para recuperar senha --}}

<div class="text-center margem-top-20">

<a href="/usuario\_frm\_recuperar\_senha">Recuperar senha</a>

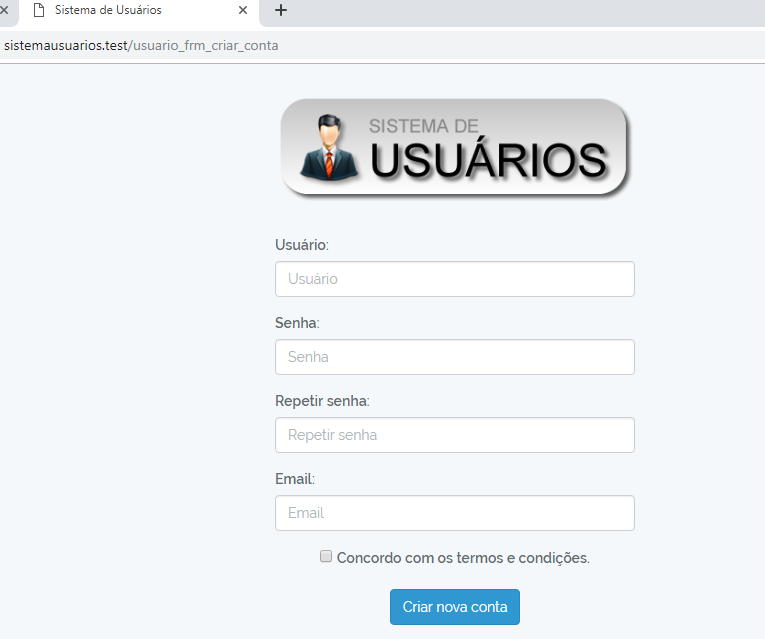
</div>

{{-- link para criar nova conta de usuário --}}

<div class="text-center">

<a href="/usuario\_frm\_criar\_conta">Criar nova conta</a>

</div>





# Aula 073 – Exercício 2 - Sistema de usuários - Parte 6

## Validação do formulário de Login

routes/web.php

//------------------------------------------------------------

Route::get('/', 'usuariosController@index');

//------------------------------------------------------------

// usuario - login

Route::get('/usuario\_frm\_login', 'usuariosController@frmLogin');

Route::post('/usuario\_executar\_login', 'usuariosController@executarLogin');

//------------------------------------------------------------

// usuario - recuperar senha

Route::get('/usuario\_frm\_recuperar\_senha', 'usuariosController@frmRecuperarSenha');

//------------------------------------------------------------

// usuario - nova conta

Route::get('/usuario\_frm\_criar\_conta', 'usuariosController@frmCriarNovaConta');

Route::post('/usuario\_executar\_criar\_conta', 'usuariosController@executarCriarNovaConta');

**app/Http/Controllers/usuariosController.php**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Requests;

class usuariosController extends Controller

{

//================================================================

public function index(){

return $this->frmLogin();

}

//================================================================

// LOGIN

//================================================================

public function frmLogin(){

//apresenta o formulário de login

return view('usuario\_frm\_login');

}

//================================================================

public function executarLogin(Request $request){

/\*

1 - verificar se os dados foram preenchidos corretamente (Validation)

2 - ir à procura do usuário na bd (Eloquent ORM)

3 - verificar se usuário e senha correspondem a usuário e senha inserido no frm (Hashing)

4 - ultrapassadas as fases todas, criar sessão de usuário (Sessions)

\*/

// Validação

$this->validate($request, [

'text\_usuario' => 'required',

'text\_senha' => 'required'

]);

return 'Ok';

}

//================================================================

// RECUPERAR SENHA

//================================================================

public function frmRecuperarSenha(){

//apresentar o formulário para recuperação da senha

return view('usuario\_frm\_recuperar\_senha');

}

//================================================================

public function executarRecuperarSenha(Request $request){

}

//================================================================

// CRIAÇÃO DE NOVA CONTA

//================================================================

public function frmCriarNovaConta(){

//apresentar o formulário de criação de nova conta

return view('usuario\_frm\_criar\_conta');

}

//================================================================

public function executarCriarNovaConta(Request $request){

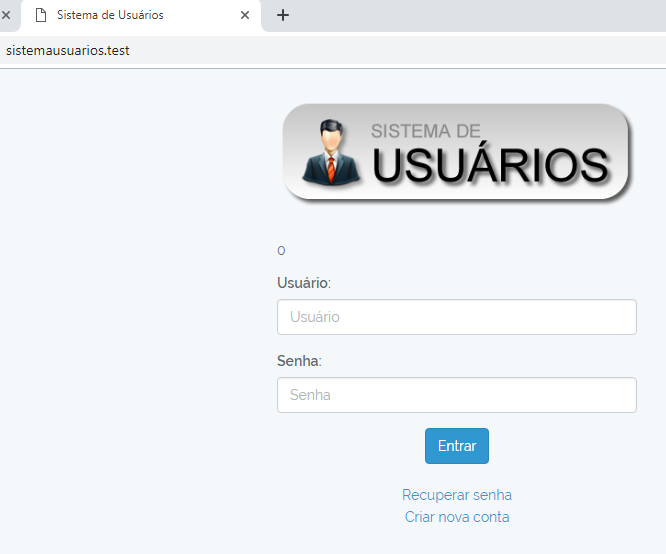
//executar os procedimentos e verificações para criação de uma nova conta

}

**resources/views/usuario\_frm\_login.blade.php**

{{-- apresentação de erros de validação... e não só --}}

<p>{{count($errors)}}</p>

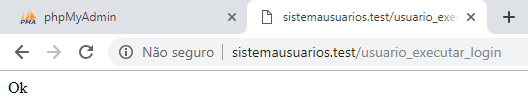


Clicando no botão "Entrar":





Clicando no botão "Entrar":



# Aula 074 – Exercício 2 - Sistema de usuários - Parte 7

## Validando os formulários Recuperar Senha e Criar Conta

**routes/web.php**

//------------------------------------------------------------

Route::get('/', 'usuariosController@index');

//------------------------------------------------------------

// usuario - login

Route::get('/usuario\_frm\_login', 'usuariosController@frmLogin');

Route::post('/usuario\_executar\_login', 'usuariosController@executarLogin');

//------------------------------------------------------------

// usuario - recuperar senha

Route::get('/usuario\_frm\_recuperar\_senha', 'usuariosController@frmRecuperarSenha');

Route::post('/usuario\_executar\_recuperar\_senha', 'usuariosController@executarRecuperarSenha');

// //------------------------------------------------------------

// usuario - nova conta

Route::get('/usuario\_frm\_criar\_conta', 'usuariosController@frmCriarNovaConta');

Route::post('/usuario\_executar\_criar\_conta', 'usuariosController@executarCriarNovaConta');

**app/Http/Controllers/usuariosController.php**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Requests;

class usuariosController extends Controller

{

//================================================================

public function index(){

return $this->frmLogin();

}

//================================================================

// LOGIN

//================================================================

public function frmLogin(){

//apresenta o formulário de login

return view('usuario\_frm\_login');

}

//================================================================

public function executarLogin(Request $request){

/\*

1 - verificar se os dados foram preenchidos corretamente (Validation)

2 - ir à procura do usuário na bd (Eloquent ORM)

3 - verificar se usuário e senha correspondem a usuário e senha inserido no frm (Hashing)

4 - ultrapassadas as fases todas, criar sessão de usuário (Sessions)

\*/

// Validação

$this->validate($request, [

'text\_usuario' => 'required',

'text\_senha' => 'required'

]);

return 'Ok';

}

//================================================================

// RECUPERAR SENHA

//================================================================

public function frmRecuperarSenha(){

//apresentar o formulário para recuperação da senha

return view('usuario\_frm\_recuperar\_senha');

}

//================================================================

public function executarRecuperarSenha(Request $request){

//validacao

$this->validate($request,[

'text\_email' => 'required|email'

]);

return 'Ok';

}

//================================================================

// CRIAÇÃO DE NOVA CONTA

//================================================================

public function frmCriarNovaConta(){

//apresentar o formulário de criação de nova conta

return view('usuario\_frm\_criar\_conta');

}

//================================================================

public function executarCriarNovaConta(Request $request){

//executar os procedimentos e verificações para criação de uma nova conta

// ---------------------------------------------------

//validacao

$this->validate($request,[

'text\_usuario' => 'required|between:3,30|alpha\_num',

'text\_senha' => 'required|between:6,15',

'text\_senha\_repetida' => 'required|same:text\_senha',

'text\_email' => 'required|email',

'check\_termos\_condicoes' => 'accepted'

]);

return 'Ok';

}

}

## Incluindo um arquivo de erros nos formulários

**resources/views/usuario\_frm\_recuperar\_senha.blade.php**

{{-- apresentação de erros de validação... e não só --}}

@include('inc.erros')

**resources/views/usuario\_frm\_criar\_conta.blade.php**

{{-- apresentação de erros de validação... e não só --}}

@include('inc.erros')

**resources/views/usuario\_frm\_login.blade.php**

{{-- apresentação de erros de validação... e não só --}}

@include('inc.erros')

**resources/views/inc/erros.blade.php**

{{-- apresentação dos erros de validação --}}

@if(count($errors) != 0)

<div class="alert alert-danger">

{{-- título da caixa de erros --}}

@if(count($errors) == 1)

<p class="titulo-erro">ERRO:</p>

@else

<p class="titulo-erro">ERROS:</p>

@endif

{{-- apresentar os erros (concatenação) --}}

<ul>

@foreach($errors->all() as $erro)

<li>{{ $erro }}</li>

@endforeach

</ul>

</div>

@endif

{{-- apresentação dos erros de comunicação com a bd --}}

**public/css/main.css**

/\*o meu código CSS\*/

.img-logo{

margin: auto;

margin-top: 30px;

margin-bottom: 30px;

}

.margem-top-20{

margin-top: 20px;

}

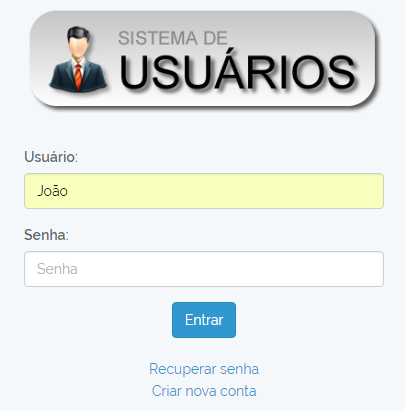
.titulo-erro{

font-weight: bolder;

color: black;

}







# Aula 075 – Exercício 2 - Sistema de usuários - Parte 8

## Cadastrando usuários na base de dados

**app/Http/Controllers/usuariosController.php**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Requests;

use Illuminate\Support\Facades\Hash;

use App\usuarios;

class usuariosController extends Controller

{

//================================================================

public function executarCriarNovaConta(Request $request){

//executar os procedimentos e verificações para criação de uma nova conta

// ---------------------------------------------------

//validacao

$this->validate($request,[

'text\_usuario' => 'required|between:3,30|alpha\_num',

'text\_senha' => 'required|between:6,15',

'text\_senha\_repetida' => 'required|same:text\_senha',

'text\_email' => 'required|email',

'check\_termos\_condicoes' => 'accepted'

]);

// ---------------------------------------------------

// verifica se já existe um usuário com o mesmo nome ou com o mesmo email

$dados = usuarios::where('usuario', "=", $request->text\_usuario)

->orWhere('email', '=', $request->text\_email)

->get();

if($dados->count() != 0){

$erros\_bd = ['Já existe um usuário com o mesmo nome ou com o mesmo email.'];

return view('usuario\_frm\_criar\_conta', compact('erros\_bd'));

}

// ---------------------------------------------------

//inserir o novo usuário na base de dados

$novo = new usuarios;

$novo->usuario = $request->text\_usuario;

$novo->senha = Hash::make($request->text\_senha);

$novo->email = $request->text\_email;

$novo->save();

return redirect('/');

}

}

**resources/views/inc/erros.blade.php**

{{-- apresentação dos erros de validação --}}

@if(count($errors) != 0)

<div class="alert alert-danger">

{{-- título da caixa de erros --}}

@if(count($errors) == 1)

<p class="titulo-erro">ERRO:</p>

@else

<p class="titulo-erro">ERROS:</p>

@endif

{{-- apresentar os erros (concatenação) --}}

<ul>

@foreach($errors->all() as $erro)

<li>{{ $erro }}</li>

@endforeach

</ul>

</div>

@endif

{{-- apresentação dos erros de comunicação com a bd --}}

@if(isset($erros\_bd))

<div class="alert alert-danger">

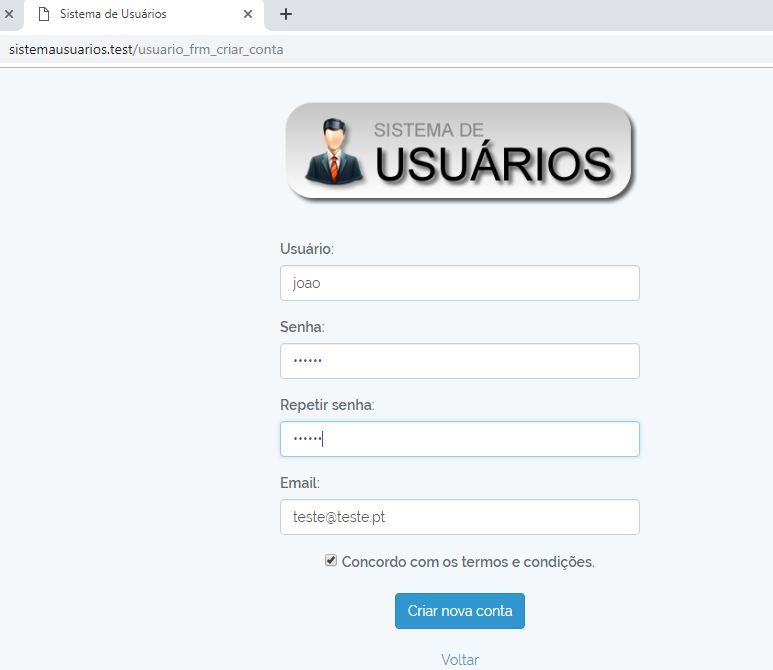
@foreach ($erros\_bd as $erro)

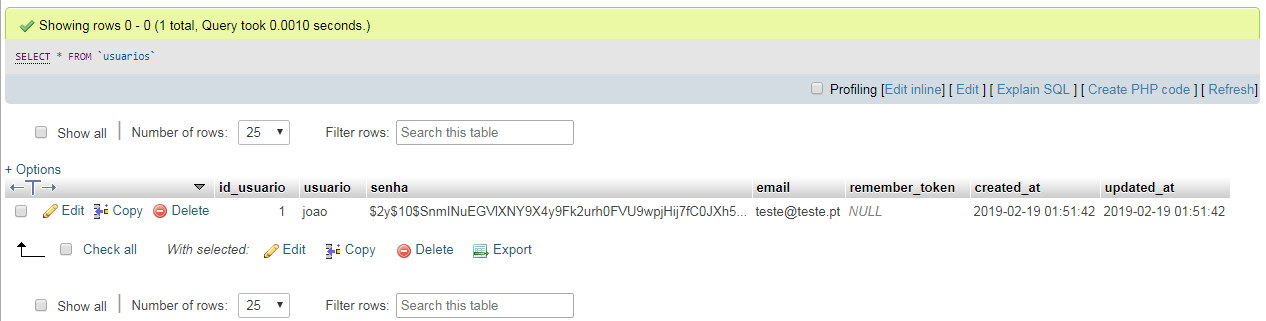
<p>{{ $erro }}</p>

@endforeach

</div>

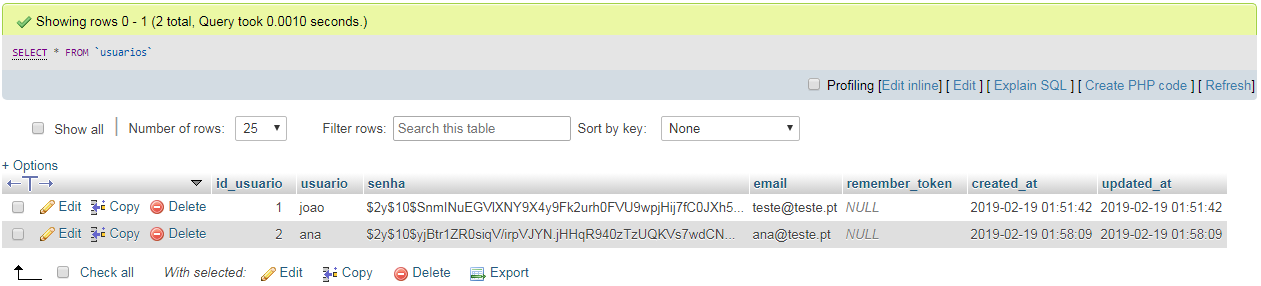
@endif

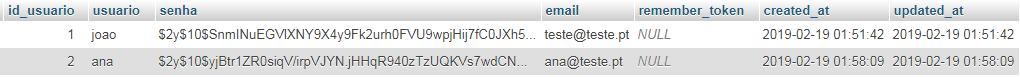


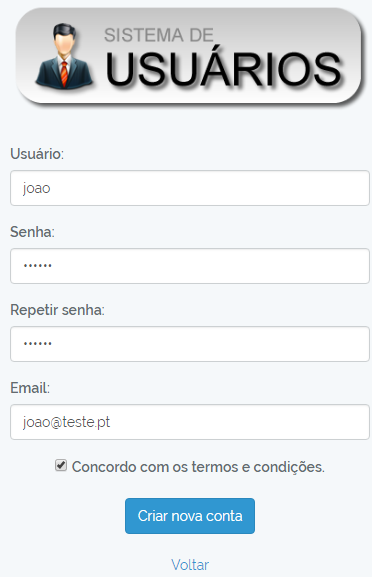


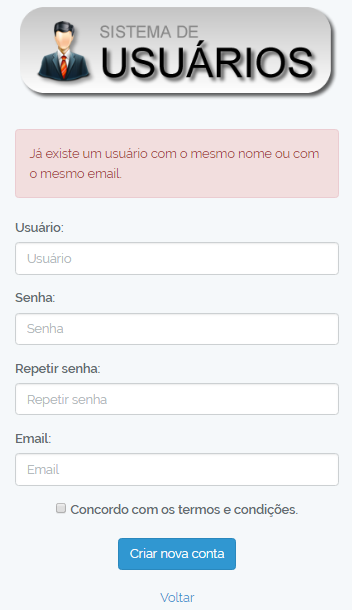












# Aula 076 – Exercício 2 - Sistema de usuários - Parte 9

## Login - Verificando usuário e senha

**app/Http/Controllers/usuariosController.php**

public function executarLogin(Request $request){

/\*

1 - verificar se os dados foram preenchidos corretamente (Validation)

2 - ir à procura do usuário na bd (Eloquent ORM)

3 - verificar se usuário e senha correspondem a usuário e senha inserido no frm (Hashing)

4 - ultrapassadas as fases todas, criar sessão de usuário (Sessions)

\*/

// Validação

$this->validate($request, [

'text\_usuario' => 'required',

'text\_senha' => 'required'

]);

//verificar se o usuário existe

$usuario = usuarios::where('usuario', $request->text\_usuario)->first();

//verifica se existe usuário

// if(count($usuario)==0){

if($usuario->count()==0){

$erros\_bd = ['Essa conta de usuário não existe.'];

return view('usuario\_frm\_login', compact('erros\_bd'));

}

//verifica se a senha corresponde ao guardado na bd

if(!Hash::check($request->text\_senha, $usuario->senha)){

$erros\_bd = ['A senha está incorreta.'];

return view('usuario\_frm\_login', compact('erros\_bd'));

}

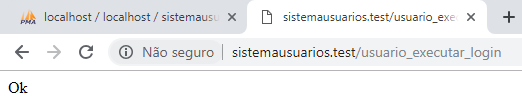
return 'Ok';

}









# Aula 077 – Exercício 2 - Sistema de usuários - Parte 10

## Sessões - Parte 1

**app/Http/Controllers/usuariosController.php**

use Session;

public function executarLogin(Request $request){

/\*

1 - verificar se os dados foram preenchidos corretamente (Validation)

2 - ir à procura do usuário na bd (Eloquent ORM)

3 - verificar se usuário e senha correspondem a usuário e senha inserido no frm (Hashing)

4 - ultrapassadas as fases todas, criar sessão de usuário (Sessions)

\*/

// Validação

$this->validate($request, [

'text\_usuario' => 'required',

'text\_senha' => 'required'

]);

//verificar se o usuário existe

$usuario = usuarios::where('usuario', $request->text\_usuario)->first();

//verifica se existe usuário

// if(count($usuario)==0){

if($usuario->count()==0){

$erros\_bd = ['Essa conta de usuário não existe.'];

return view('usuario\_frm\_login', compact('erros\_bd'));

}

//verifica se a senha corresponde ao guardado na bd

if(!Hash::check($request->text\_senha, $usuario->senha)){

$erros\_bd = ['A senha está incorreta.'];

return view('usuario\_frm\_login', compact('erros\_bd'));

}

//criar/abrir sessão de usuário

Session::put('login', 'sim');

Session::put('usuario', $usuario->usuario);

return redirect('/');

}

**routes/web.php**

//------------------------------------------------------------

Route::get('/', 'usuariosController@index');

//------------------------------------------------------------

// usuario - login

Route::get('/usuario\_frm\_login', 'usuariosController@frmLogin');

Route::post('/usuario\_executar\_login', 'usuariosController@executarLogin');

//------------------------------------------------------------

// usuario - recuperar senha

Route::get('/usuario\_frm\_recuperar\_senha', 'usuariosController@frmRecuperarSenha');

Route::post('/usuario\_executar\_recuperar\_senha', 'usuariosController@executarRecuperarSenha');

// Route::get('/usuario\_email\_enviado', 'usuariosController@emailEnviado');

//------------------------------------------------------------

// usuario - nova conta

Route::get('/usuario\_frm\_criar\_conta', 'usuariosController@frmCriarNovaConta');

Route::post('/usuario\_executar\_criar\_conta', 'usuariosController@executarCriarNovaConta');

//------------------------------------------------------------

// interior da aplicação

Route::get('/aplicacao\_index', 'aplicacaoController@apresentarPaginaInicial');

**app/Http/Controllers/aplicacaoController.php**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Requests;

use Session;

class aplicacaoController extends Controller

{

public function apresentarPaginaInicial(){

//verifica se a sessão está ativa

if(!Session::has('login')){ return redirect('/');}

//apresenta o interior da aplicação

return view('aplicacao');

}

}

**resources/views/aplicacao.blade.php**

@extends('layouts.app')

@section('conteudo')

<p>Estou cá dentro!</p>

@endsection





# Aula 078 – Exercício 2 - Sistema de usuários - Parte 11

## Sessões - Parte 2

**routes/web.php**

//------------------------------------------------------------

Route::get('/', 'usuariosController@index');

//------------------------------------------------------------

// usuario - login

Route::get('/usuario\_frm\_login', 'usuariosController@frmLogin');

Route::post('/usuario\_executar\_login', 'usuariosController@executarLogin');

//------------------------------------------------------------

// usuário - logout

Route::get('/usuario\_logout', 'usuariosController@logout');

//------------------------------------------------------------

// usuario - recuperar senha

Route::get('/usuario\_frm\_recuperar\_senha', 'usuariosController@frmRecuperarSenha');

Route::post('/usuario\_executar\_recuperar\_senha', 'usuariosController@executarRecuperarSenha');

// Route::get('/usuario\_email\_enviado', 'usuariosController@emailEnviado');

//------------------------------------------------------------

// usuario - nova conta

Route::get('/usuario\_frm\_criar\_conta', 'usuariosController@frmCriarNovaConta');

Route::post('/usuario\_executar\_criar\_conta', 'usuariosController@executarCriarNovaConta');

//------------------------------------------------------------

// interior da aplicação

Route::get('/aplicacao\_index', 'aplicacaoController@apresentarPaginaInicial');

**app/Http/Controllers/aplicacaoController.php**

public function logout(){

//logout da sessão (destruir a sessão e redirecionar para o quadro de login)

//destruir a sessão

Session::flush();

return redirect('/');

}

**resources/views/aplicacao.blade.php**

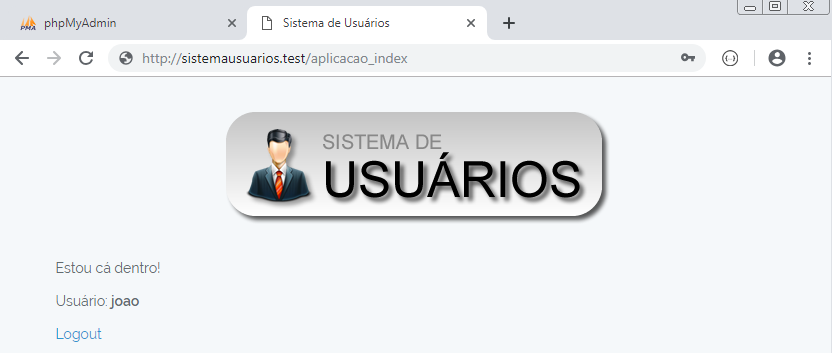
@extends('layouts.app')

@section('conteudo')

<p>Estou cá dentro!</p>

@endsection







# Aula 079 – Exercício 2 - Sistema de usuários - Parte 12

## Recuperação de senha - Parte 1

**app/Http/Controllers/aplicacaoController.php**

public function executarRecuperarSenha(Request $request){

//validacao

$this->validate($request,[

'text\_email' => 'required|email'

]);

/\*

1 - Ter usuário com email válido.

2 - Verificar se o email inserido corresponde ao do usuário

3 - Sistema cria senha aleatória

4 - Envia email com a nova senha para o email do usuário

5 - Informa (view) o usuário que foi enviada senha para o email dele.

\*/

//vai buscar o usuário que contém a conta de email indicada

$usuario = usuarios::where('email', $request->text\_email)->get();

if($usuario->count() == 0){

$erros\_bd = ['O email não está associado a nenhuma conta de usuário.'];

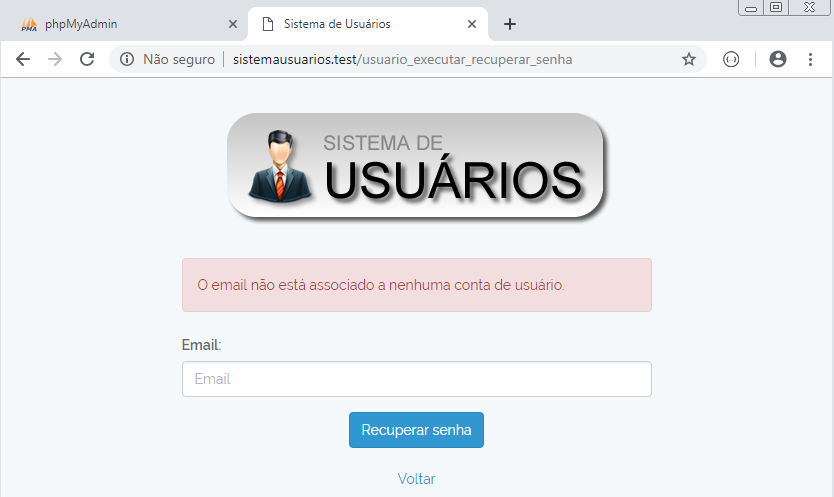
return view('usuario\_frm\_recuperar\_senha', compact('erros\_bd'));

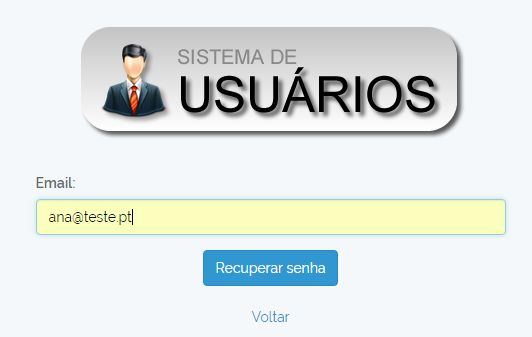
}

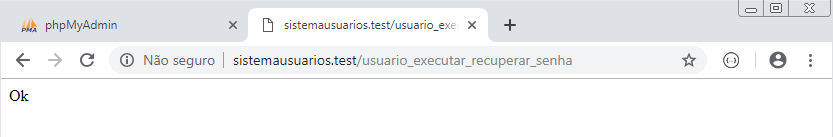
return 'Ok';

}









# Aula 080 – Exercício 2 - Sistema de usuários - Parte 13

## Recuperação de senha - Criando senha aleatória

### Criando uma classe

Dentro da pasta app criar uma pasta chamada classes e dentro dessa pasta criar um arquivo chamado minhaclasse.php

**app/classes/minhaclasse.php**

<?php

namespace App\classes;

class minhaClasse{

public static function criarCodigo(){

//criar um código aleatório (senha) com 10 caracteres

$valor = '';

$caracteres = 'abcdefghijklmnopqrstuvwyxz\_ABCDEFGHIJKLMNOPQRSTUVWYXZ1234567890!?#$%';

for($m=0; $m<10; $m++){

$index = rand(0, strlen($caracteres));

$valor .= substr($caracteres, $index, 1);

}

return $valor;

}

}

?>

**app/Http/Controllers/aplicacaoController.php**

use App\classes\minhaClasse;

public function executarRecuperarSenha(Request $request){

//validacao

$this->validate($request,[

'text\_email' => 'required|email'

]);

/\*

1 - Ter usuário com email válido.

2 - Verificar se o email inserido corresponde ao do usuário

3 - Sistema cria senha aleatória

3a - Registrar (alterar) a senha do usuário na base de dados

4 - Envia email com a nova senha para o email do usuário

5 - Informa (view) o usuário que foi enviada senha para o email dele.

\*/

//vai buscar o usuário que contém a conta de email indicada

$usuario = usuarios::where('email', $request->text\_email)->get();

if($usuario->count() == 0){

$erros\_bd = ['O email não está associado a nenhuma conta de usuário.'];

return view('usuario\_frm\_recuperar\_senha', compact('erros\_bd'));

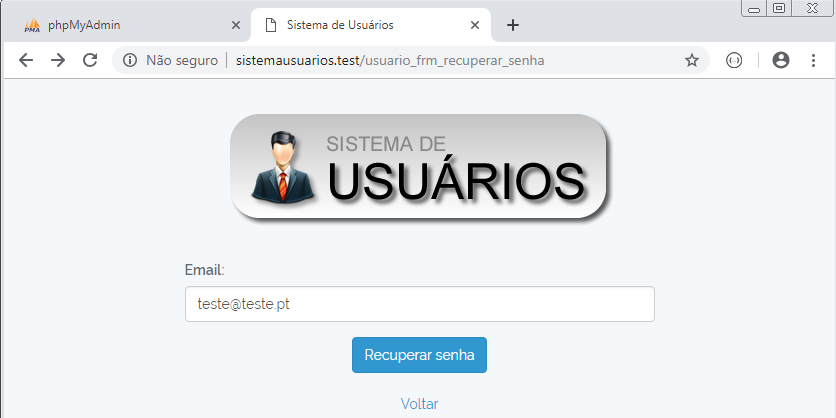
}

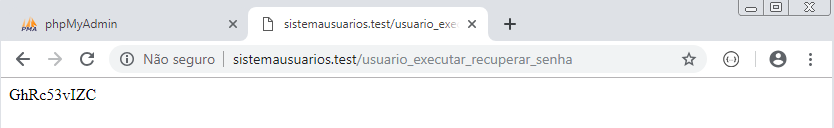
//criar uma nova senha aleatória

$nova\_senha = minhaClasse::criarCodigo();

return $nova\_senha;

}





# Aula 081 – Exercício 2 - Sistema de usuários - Parte 14

## Atualizar a senha do usuário a senha de recuperação

**app/Http/Controllers/aplicacaoController.php**

public function executarRecuperarSenha(Request $request){

//validacao

$this->validate($request,[

'text\_email' => 'required|email'

]);

/\*

1 - Ter usuário com email válido.

2 - Verificar se o email inserido corresponde ao do usuário

3 - Sistema cria senha aleatória

3a - Registrar (alterar) a senha do usuário na base de dados

4 - Envia email com a nova senha para o email do usuário

5 - Informa (view) o usuário que foi enviada senha para o email dele.

\*/

//vai buscar o usuário que contém a conta de email indicada

$usuario = usuarios::where('email', $request->text\_email)->get();

if($usuario->count() == 0){

$erros\_bd = ['O email não está associado a nenhuma conta de usuário.'];

return view('usuario\_frm\_recuperar\_senha', compact('erros\_bd'));

}

//atualizar a senha do usuário para a nova senha (senha de recuperação)

$usuario = $usuario->first();

// criar uma nova senha aleatória

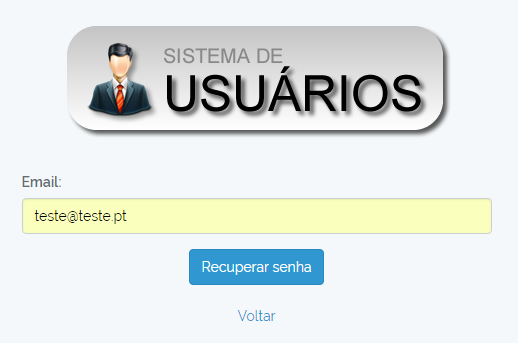
$nova\_senha = minhaClasse::criarCodigo();

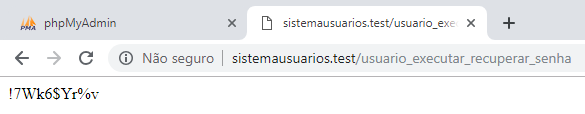
$usuario->senha = Hash::make($nova\_senha);

$usuario->save();

return $nova\_senha;

}





Senha: aaabbb

Senha original encriptada:

$2y$10$SnmINuEGVlXNY9X4y9Fk2urh0FVU9wpjHij7fC0JXh5W33Y8UWeou

Senha aleatória: !7Wk6$Yr%v

Senha aleatória encriptada:

$2y$10$F2j2zTw5nLH98eYZHDFYT.A9TfnD3AF/P6jOKg8grXMIiSN3DMI8C

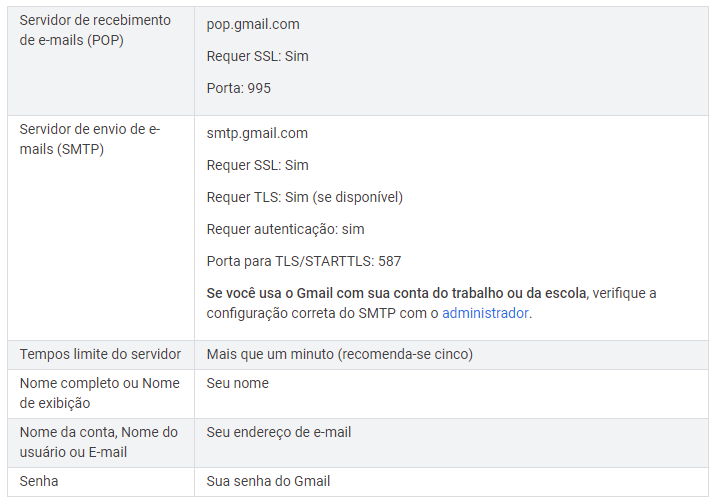
# Aula 082 – Exercício 2 - Sistema de usuários - Parte 15

## Configurando o email para envio de mensagens

- Acesse a conta de emails do gmail.

- Clique em Definições e posteriormente na aba Encaminhamento e POP IMAP

- No campo Configure o seu cliente de email clique no link instruções de configuração



Configurar email no arquivo **.env**

MAIL\_DRIVER=smtp

MAIL\_HOST=smtp.gmail.com

MAIL\_PORT=587

MAIL\_USERNAME=pinheiroberto61@gmail.com

MAIL\_PASSWORD=angstron1005

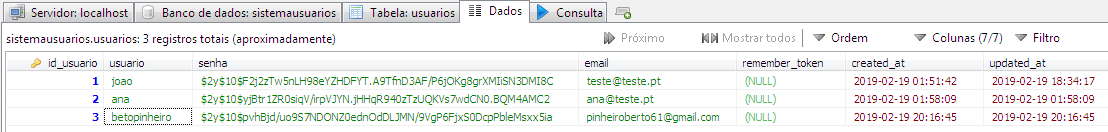
MAIL\_ENCRYPTION=tls

- Em Google Conta clique em Segurança

- No campo Acesso a aplicações menos seguras clique em Ativar o acesso (não recomendado)

- No sistema de usuários Incluia uma nova conta, com senha: 1234567





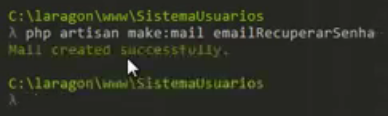
# Aula 083 – Exercício 2 - Sistema de usuários - Parte 16

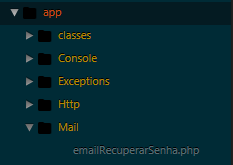
## Enviando o email para o usuário

### Criando a classe de email

No terminal, entre com o seguinte comando:

**php artisan make:mail emailRecuperarSenha**





- Dentro da pasta views criar uma subpasta chamada emails com o seguinte arquivo: emailRecuperarSenha.blade.php

**resources/views/emails/emailRecuperarSenha**

<p>Este email foi-lhe enviado por ter solicitado a recuperação da senha.</p>

<p>Nova senha:<strong>{{ $nova\_senha }}</strong></p>

<p>Na próxima visita ao site SistemaUsuarios, use a nova senha.</p>

**app/Mail/emailRecuperarSenha**

<?php

namespace App\Mail;

use Illuminate\Bus\Queueable;

use Illuminate\Mail\Mailable;

use Illuminate\Queue\SerializesModels;

use Illuminate\Contracts\Queue\ShouldQueue;

class emailRecuperarSenha extends Mailable

{

use Queueable, SerializesModels;

protected $nova\_senha;

//=====================================

public function \_\_construct($nova\_senha)

{

$this->nova\_senha = $nova\_senha;

}

//=====================================

public function build()

{

return $this->view('emails.emailRecuperarSenha')->with(['nova\_senha' => $this->nova\_senha]);

}

}

**app/Http/Controllers/aplicacaoController.php**

public function executarRecuperarSenha(Request $request){

//validacao

$this->validate($request,[

'text\_email' => 'required|email'

]);

/\*

1 - Ter usuário com email válido.

2 - Verificar se o email inserido corresponde ao do usuário

3 - Sistema cria senha aleatória

3a - Registrar (alterar) a senha do usuário na base de dados

4 - Envia email com a nova senha para o email do usuário

5 - Informa (view) o usuário que foi enviada senha para o email dele.

\*/

//vai buscar o usuário que contém a conta de email indicada

$usuario = usuarios::where('email', $request->text\_email)->get();

if($usuario->count() == 0){

$erros\_bd = ['O email não está associado a nenhuma conta de usuário.'];

return view('usuario\_frm\_recuperar\_senha', compact('erros\_bd'));

}

//atualizar a senha do usuário para a nova senha (senha de recuperação)

$usuario = $usuario->first();

// criar uma nova senha aleatória

$nova\_senha = minhaClasse::criarCodigo();

$usuario->senha = Hash::make($nova\_senha);

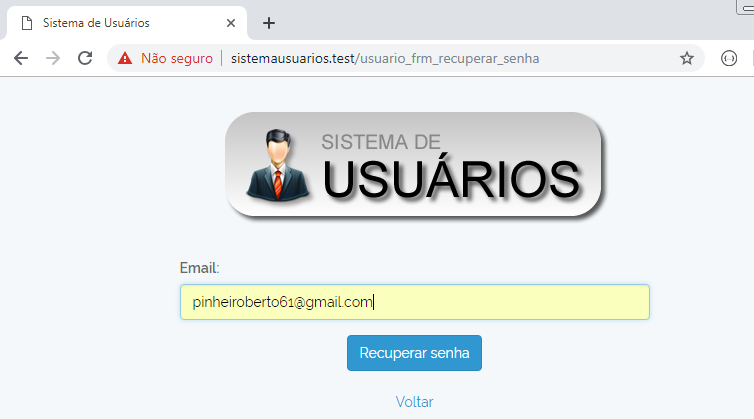
$usuario->save();

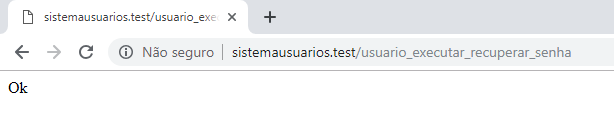
//enviar email ao usuário com a nova senha

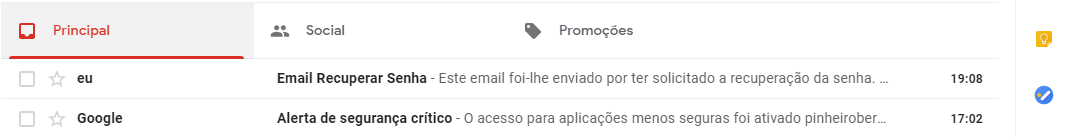
Mail::to($usuario->email)->send(new emailRecuperarSenha($nova\_senha));

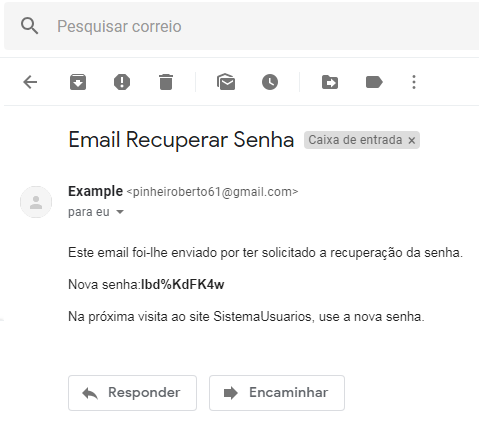
return 'Ok';

}









Senha: 1234567

Senha original encriptada:

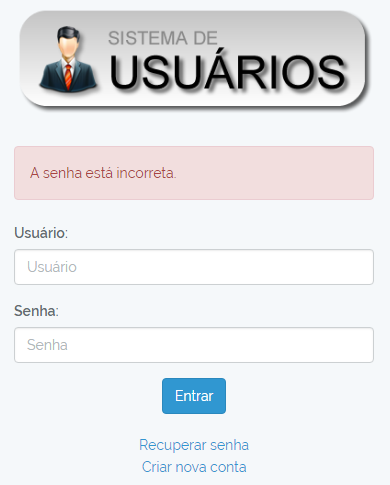
$2y$10$pvhBjd/uo9S7NDONZ0ednOdDLJMN/9VgP6FjxS0DcpPbleMsxx5ia

Senha aleatória: **Ibd%KdFK4w**

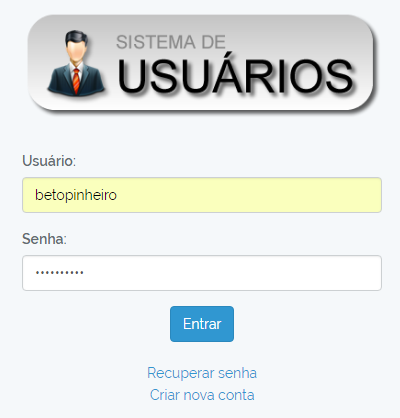
Senha aleatória encriptada:

$2y$10$Kqd7c./0pUj/BWzivMgl5.sKCvdGdDKJm24U9rbSZlH0jT8DZONdy

Entrando com a senha: 1234567



Entrando com a senha enviada: **Ibd%KdFK4w**





# Aula 084 – Exercício 2 - Sistema de usuários - Parte 17

**routes/web.php**

//------------------------------------------------------------

Route::get('/', 'usuariosController@index');

//------------------------------------------------------------

// usuario - login

Route::get('/usuario\_frm\_login', 'usuariosController@frmLogin');

Route::post('/usuario\_executar\_login', 'usuariosController@executarLogin');

//------------------------------------------------------------

// usuário - logout

Route::get('/usuario\_logout', 'usuariosController@logout');

//------------------------------------------------------------

// usuario - recuperar senha

Route::get('/usuario\_frm\_recuperar\_senha', 'usuariosController@frmRecuperarSenha');

Route::post('/usuario\_executar\_recuperar\_senha', 'usuariosController@executarRecuperarSenha');

Route::get('/usuario\_email\_enviado', 'usuariosController@emailEnviado');

//------------------------------------------------------------

// usuario - nova conta

Route::get('/usuario\_frm\_criar\_conta', 'usuariosController@frmCriarNovaConta');

Route::post('/usuario\_executar\_criar\_conta', 'usuariosController@executarCriarNovaConta');

//------------------------------------------------------------

// interior da aplicação

Route::get('/aplicacao\_index', 'aplicacaoController@apresentarPaginaInicial');

**app/Http/Controllers/aplicacaoController.php**

//================================================================

public function executarRecuperarSenha(Request $request){

//validacao

$this->validate($request,[

'text\_email' => 'required|email'

]);

/\*

1 - Ter usuário com email válido.

2 - Verificar se o email inserido corresponde ao do usuário

3 - Sistema cria senha aleatória

3a - Registrar (alterar) a senha do usuário na base de dados

4 - Envia email com a nova senha para o email do usuário

5 - Informa (view) o usuário que foi enviada senha para o email dele.

\*/

//vai buscar o usuário que contém a conta de email indicada

$usuario = usuarios::where('email', $request->text\_email)->get();

if($usuario->count() == 0){

$erros\_bd = ['O email não está associado a nenhuma conta de usuário.'];

return view('usuario\_frm\_recuperar\_senha', compact('erros\_bd'));

}

//atualizar a senha do usuário para a nova senha (senha de recuperação)

$usuario = $usuario->first();

// criar uma nova senha aleatória

$nova\_senha = minhaClasse::criarCodigo();

$usuario->senha = Hash::make($nova\_senha);

$usuario->save();

//enviar email ao usuário com a nova senha

Mail::to($usuario->email)->send(new emailRecuperarSenha($nova\_senha));

//apresentar uma view informando o usuário que foi enviado email

return redirect('/usuario\_email\_enviado');

}

public function emailEnviado(){

return view('usuario\_email\_enviado');

}

**resources/views/usuario\_email\_enviado.blade.php**

@extends('layouts.app')

@section('conteudo')

<div class="row">

<div class="col-md-12 text-center">

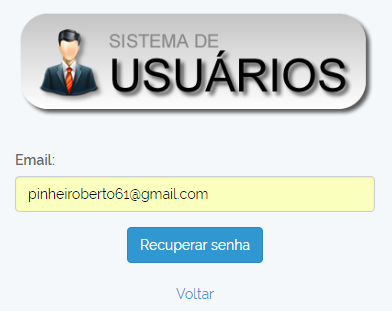
<p>Foi enviado um email com a nova senha.</p>

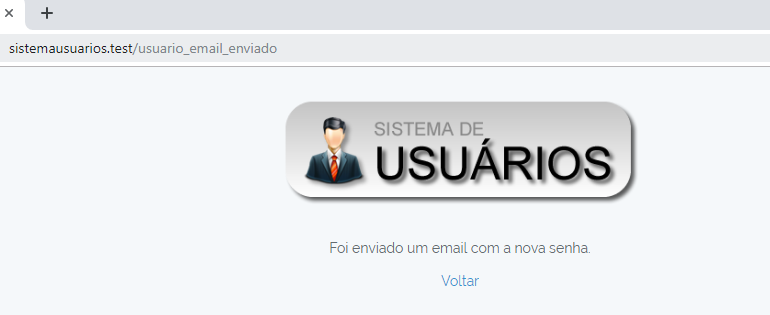
<p><a href="/">Voltar</a></p>

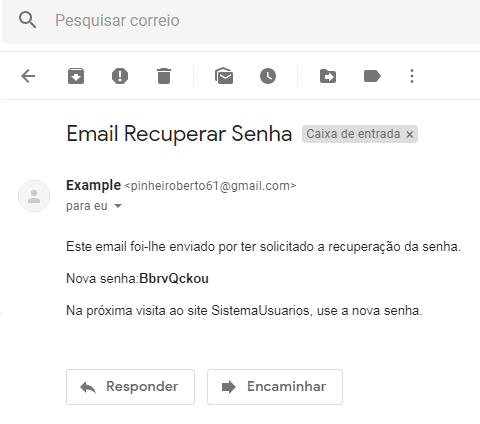
</div>

</div>

@endsection







Senha anterior: **Ibd%KdFK4w**

Nova senha: **BbrvQckou**

Entrando com a nova senha:

