

# Curso Laravel 5.4 - Aulas 68 a 84

## SYS4SOFT (João Ribeiro)

[https://www.youtube.com/watch?v=0T5gM1WRNsY&list=PLXik\\_5Br-zO893qVjjP7a4gg4NYrl33w1&index=1](https://www.youtube.com/watch?v=0T5gM1WRNsY&list=PLXik_5Br-zO893qVjjP7a4gg4NYrl33w1&index=1)

Resumo do curso feito por Roberto Pinheiro

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

### SISTEMA DE USUÁRIOS

Sistema para inserção de usuários na BD;  
Formulário de login;  
Sistema de recuperação de senha.

### SISTEMA DE USUÁRIOS

Route, Model, View, Controller, Migration  
Form validation, Hashing, Eloquent ORM, Sessions  
Blade, Mail, Criação de classes próprias, etc...



The image shows a web form for a user system. At the top, there is a header with a logo of a person in a suit and the text "SISTEMA DE USUÁRIOS". Below the header, there are two input fields: "Usuário:" and "Senha:". The "Usuário:" field has a placeholder text "Usuário". The "Senha:" field has a placeholder text "Senha". Below the input fields, there is a blue button labeled "Entrar". At the bottom, there are two links: "Recuperar senha" and "Criar nova conta".

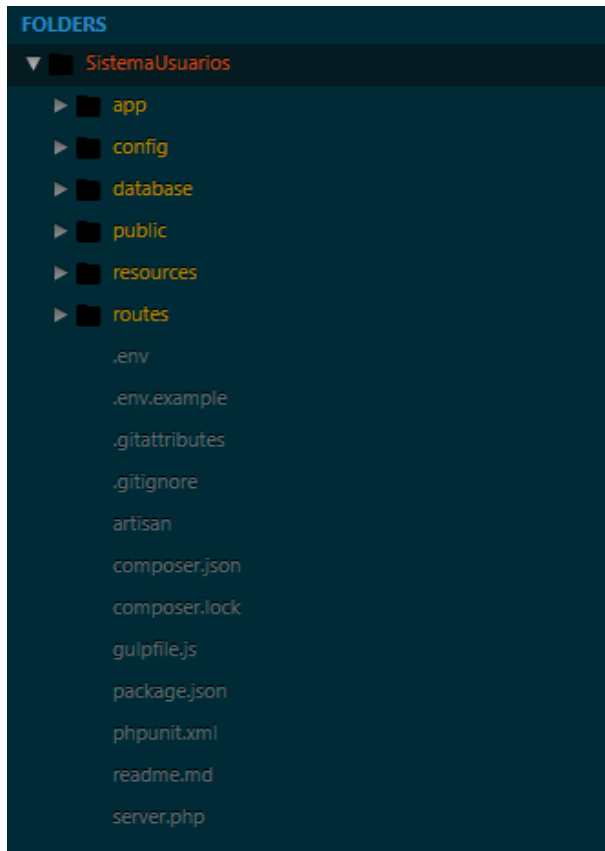
No Laragon, clicar com o botão direito na área de trabalho e selecionar "**Criar rapidamente um website**" → "**Laravel (zip)**". Digitar como nome do projeto: **SistemaUsuarios** e clicar no botão Ok.

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

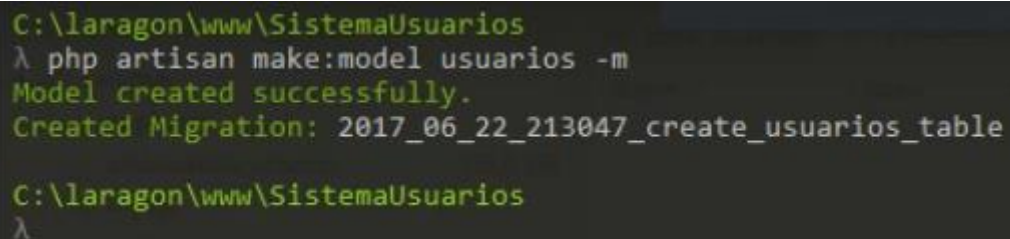
**.env**

```
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



```
C:\laragon\www\SistemaUsuarios
λ php artisan make:model usuarios -m
Model created successfully.
Created Migration: 2017_06_22_213047_create_usuarios_table

C:\laragon\www\SistemaUsuarios
λ
```

## 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;

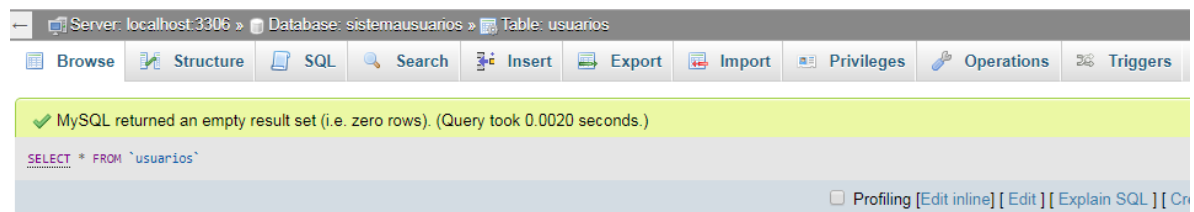
class usuarios extends Model
{
    //alterar a identificação da PK
    protected $primaryKey = 'id_usuario';
}
```

## Criando a tabela

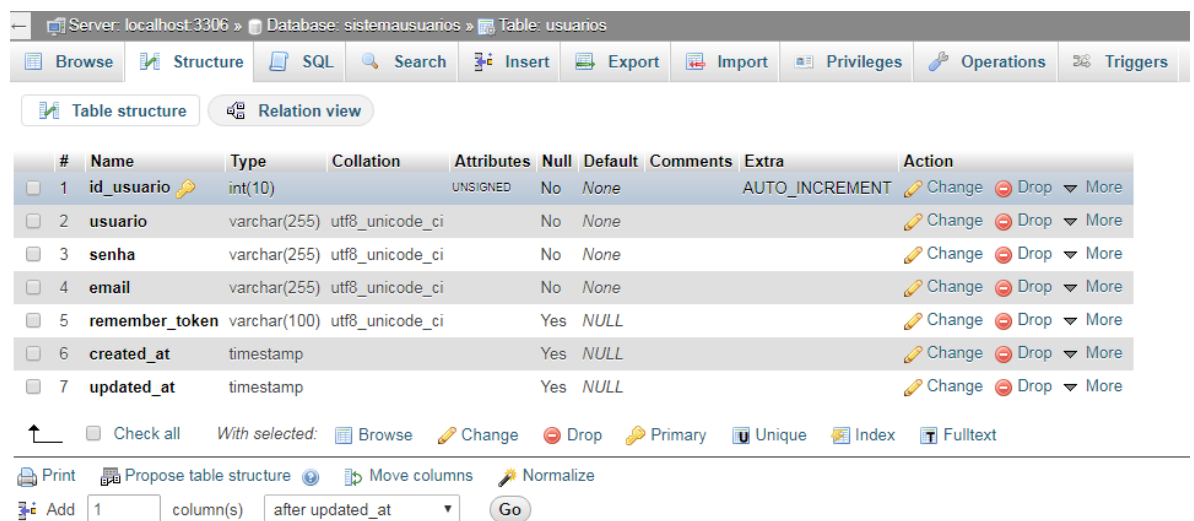
php artisan migrate

```
C:\laragon\www\SistemaUsuarios
λ php artisan migrate
Migration table created successfully.
Migrated: 2017_06_22_213047_create_usuarios_table

C:\laragon\www\SistemaUsuarios
λ |
```



id\_usuario usuario senha email remember\_token created\_at updated\_at



## 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 separados 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">
    
  </div>
</div>
```

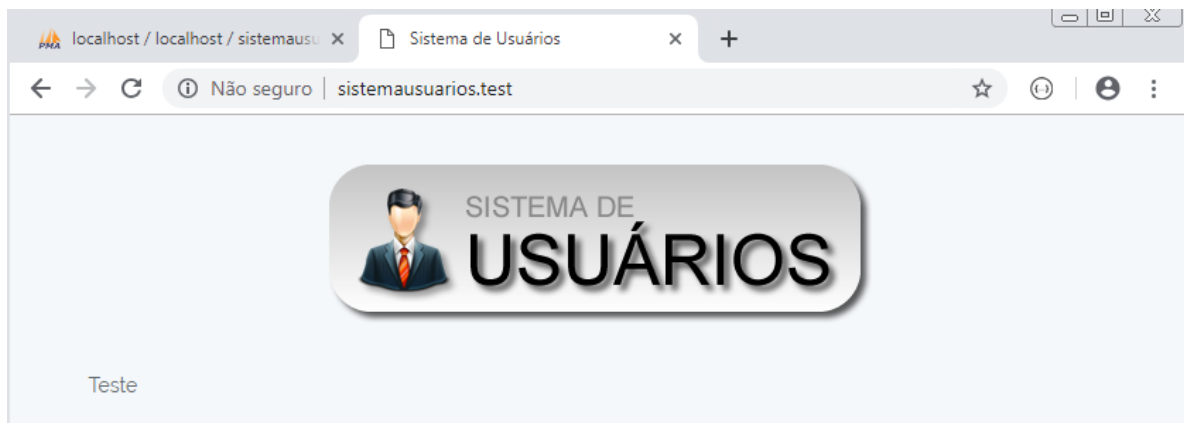
**resources/views/teste.blade.php**

```
@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

localhost / localhost / sistemausu X Sistema de Usuários X +

← → ↻ ⓘ Não seguro | sistemausuarios.test ☆ ⓘ | 👤 ⋮



SISTEMA DE  
**USUÁRIOS**

Usuário:

Senha:

Entrar

[Recuperar senha](#)  
[Criar nova conta](#)

## 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>

        </form>

    </div>

</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>

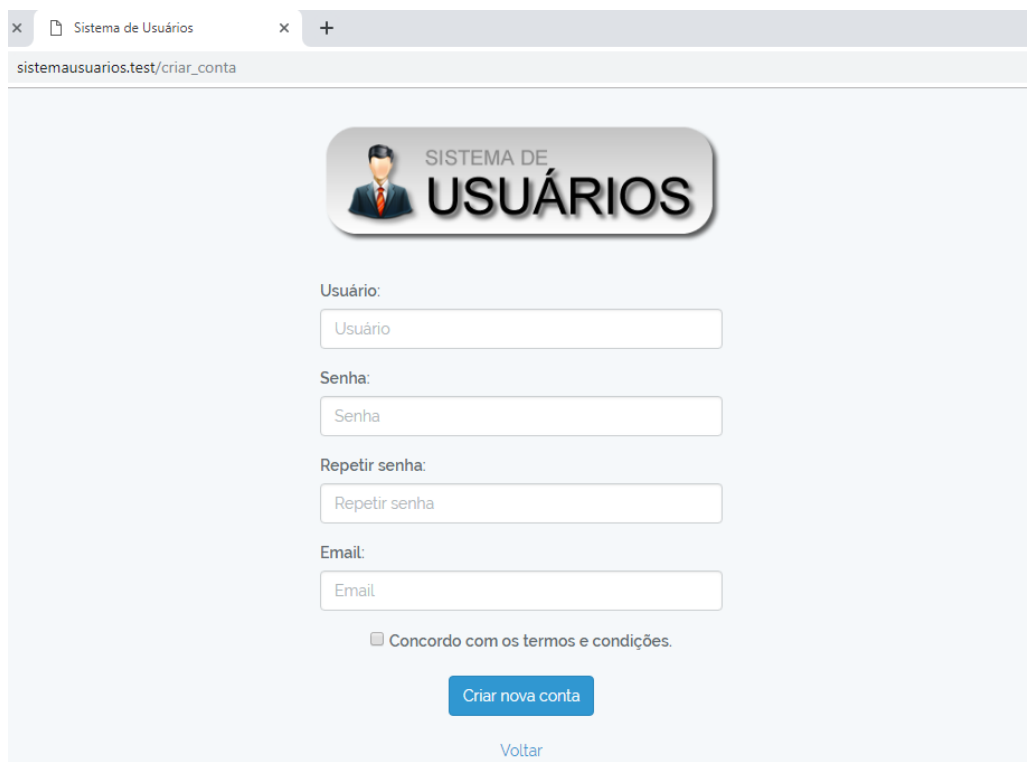
</form>

</div>

</div>

```

@endsection



The screenshot shows a web browser window with the address bar displaying 'sistemausuarios.test/criar\_conta'. The page features a header with a logo of a person in a suit and the text 'SISTEMA DE USUÁRIOS'. Below the header, there are four input fields for registration: 'Usuário:', 'Senha:', 'Repetir senha:', and 'Email:'. Each field has a placeholder text matching its label. Below these fields is a checkbox labeled 'Concordo com os termos e condições.' and a blue button labeled 'Criar nova conta'. At the bottom of the page, there is a blue link labeled 'Voltar'.

## 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

```
C:\laragon\www\SistemaUsuarios
λ php artisan make:controller usuariosController
Controller created successfully.

C:\laragon\www\SistemaUsuarios
λ
```

### 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>
```

Sistema de Usuários

sistemausuarios.test/usuario\_frm\_criar\_conta



SISTEMA DE  
**USUÁRIOS**

Usuário:

Senha:

Repetir senha:

Email:

☐ Concordo com os termos e condições.

Criar nova conta



Email:

Recuperar senha

[Voltar](#)

## 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>
```

× Sistema de Usuários × +

sistemausuarios.test

 SISTEMA DE  
**USUÁRIOS**

0

Usuário:

Senha:

[Entrar](#)

[Recuperar senha](#)  
[Criar nova conta](#)

Clicando no botão "Entrar":

 SISTEMA DE  
**USUÁRIOS**

2

Usuário:

Senha:

[Entrar](#)

[Recuperar senha](#)  
[Criar nova conta](#)

 SISTEMA DE  
**USUÁRIOS**

0

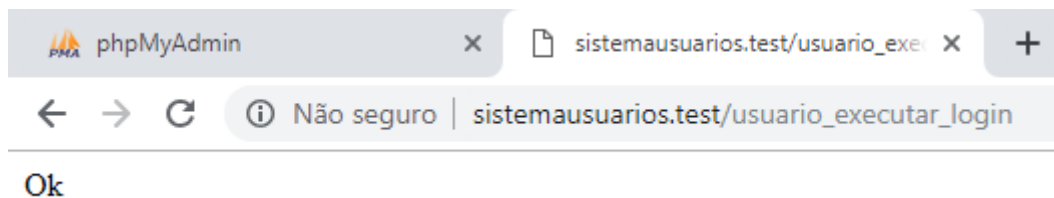
Usuário:

Senha:

Entrar

[Recuperar senha](#)  
[Criar nova conta](#)

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;
}
```



## SISTEMA DE USUÁRIOS

### ERROS:

- The text usuario field is required.
- The text senha field is required.

Usuário:

Senha:

Entrar

[Recuperar senha](#)

[Criar nova conta](#)



## SISTEMA DE USUÁRIOS

Usuário:

Senha:

Entrar

[Recuperar senha](#)

[Criar nova conta](#)



ERRO:

- The text senha field is required.

Usuário:

Senha:

Entrar

[Recuperar senha](#)

[Criar nova conta](#)

## 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

```



Sistema de Usuários

sistemausuarios.test/usuario\_frm\_criar\_conta



SISTEMA DE  
USUÁRIOS

Usuário:

joao

Senha:

.....

Repetir senha:

.....|

Email:

teste@teste.pt

☒ Concordo com os termos e condições.

Criar nova conta

[Voltar](#)

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

SELECT \* FROM `usuarios`

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

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

+ Options

	id_usuario	usuario	senha	email	remember_token	created_at	updated_at
<input type="checkbox"/> Edit Copy Delete	1	joao	\$2y\$10\$SnmlNuEGVIXNY9X4y9Fk2urh0FVU9wpjHij7fC0JXh5...	teste@teste.pt	NULL	2019-02-19 01:51:42	2019-02-19 01:51:42

☐ Check all | With selected: Edit Copy Delete Export

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

id_usuario	usuario	senha	email	remember_token	created_at	updated_at
1	joao	\$2y\$10\$SnmlNuEGVIXNY9X4y9Fk2urh0FVU9wpjHij7fC0JXh5...	teste@teste.pt	NULL	2019-02-19 01:51:42	2019-02-19 01:51:42



SISTEMA DE  
**USUÁRIOS**

Usuário:

ana

Senha:

.....

Repetir senha:

.....

Email:

ana@teste.pt

☒ Concordo com os termos e condições.

Criar nova conta

Voltar

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

SELECT \* FROM `usuarios`

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

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

+ Options

	id_usuario	usuario	senha	email	remember_token	created_at	updated_at
<input type="checkbox"/>	1	joao	\$2y\$10\$Snm1NuEGVIXNY9X4y9Fk2urh0FVU9wpjHij7fC0JXh5...	teste@teste.pt	NULL	2019-02-19 01:51:42	2019-02-19 01:51:42
<input type="checkbox"/>	2	ana	\$2y\$10\$yjBtr1ZR0siqViirpVJYN.jHHqR940zTzUQKV7wdCN...	ana@teste.pt	NULL	2019-02-19 01:58:09	2019-02-19 01:58:09

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

id_usuario	usuario	senha	email	remember_token	created_at	updated_at
1	joao	\$2y\$10\$Snm1NuEGVIXNY9X4y9Fk2urh0FVU9wpjHij7fC0JXh5...	teste@teste.pt	NULL	2019-02-19 01:51:42	2019-02-19 01:51:42
2	ana	\$2y\$10\$yjBtr1ZR0siqViirpVJYN.jHHqR940zTzUQKV7wdCN...	ana@teste.pt	NULL	2019-02-19 01:58:09	2019-02-19 01:58:09

 SISTEMA DE  
**USUÁRIOS**

Usuário:

Senha:

Repetir senha:

Email:

☒ Concordo com os termos e condições.

[Criar nova conta](#)

[Voltar](#)

 SISTEMA DE  
**USUÁRIOS**

Já existe um usuário com o mesmo nome ou com o mesmo email.

Usuário:

Senha:

Repetir senha:

Email:

☐ Concordo com os termos e condições.

[Criar nova conta](#)

[Voltar](#)

## 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';
}
```



## SISTEMA DE USUÁRIOS

Usuário:

Senha:

Entrar

[Recuperar senha](#)

[Criar nova conta](#)



## SISTEMA DE USUÁRIOS

A senha está incorreta.

Usuário:

Senha:

Entrar

[Recuperar senha](#)

[Criar nova conta](#)



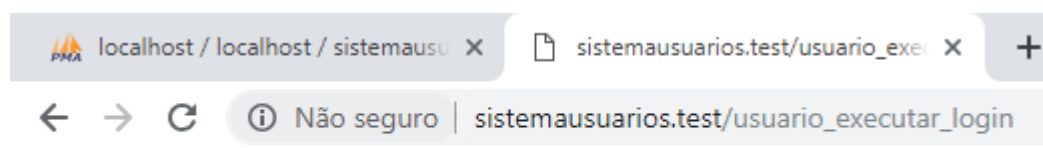
Usuário:

Senha:

Entrar

[Recuperar senha](#)

[Criar nova conta](#)



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
```



The login form features a header with a user icon and the text 'SISTEMA DE USUÁRIOS'. Below this, there are two input fields: one for the username labeled 'Usuário:' containing the text 'joao', and another for the password labeled 'Senha:' with masked characters '.....'. A blue 'Entrar' button is positioned below the password field. At the bottom, there are two links: 'Recuperar senha' and 'Criar nova conta'.

Usuário:

joao

Senha:

.....

Entrar

[Recuperar senha](#)

[Criar nova conta](#)



## 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
```




The image shows a login form for a system titled "SISTEMA DE USUÁRIOS". The form has a header with a user icon and the title. Below the header, there are two input fields: "Usuário:" with the value "joao" and "Senha:" with masked characters ".....". A blue "Entrar" button is positioned below the password field. At the bottom, there are two links: "Recuperar senha" and "Criar nova conta".



phpMyAdmin x Sistema de Usuários x +

← → ↻ http://sistemausuarios.test/aplicacao\_index

 SISTEMA DE  
**USUÁRIOS**

Usuário:

Senha:

[Entrar](#)

[Recuperar senha](#)  
[Criar nova conta](#)

## 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';

}
```

phpMyAdmin x Sistema de Usuários x +

← → ↻ ⚠ Não seguro | sistemausuarios.test/usuario\_frm\_recuperar\_senha ☆ ⓘ ⓘ ⋮

 SISTEMA DE  
**USUÁRIOS**


Email:

Recuperar senha

[Voltar](#)

phpMyAdmin x Sistema de Usuários x +

← → ↻ ⓘ Não seguro | sistemausuarios.test/usuario\_executar\_recuperar\_senha ☆ ⓘ ⓘ ⋮

 SISTEMA DE  
**USUÁRIOS**

O email não está associado a nenhuma conta de usuário.

Email:

Recuperar senha

[Voltar](#)



# SISTEMA DE USUÁRIOS

Email:

ana@teste.pt

Recuperar senha

[Voltar](#)

phpMyAdmin

sistemausuarios.test/usuario\_exe

Não seguro | sistemausuarios.test/usuario\_executar\_recuperar\_senha

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 = "";
        $caracteres = 'abcdefghijklmnopqrstuvwxyz_ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!?$%';
        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'));
```

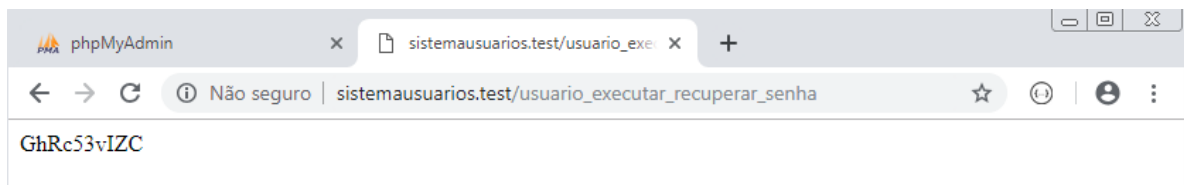
```
    }
```

```
    //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;

}
```

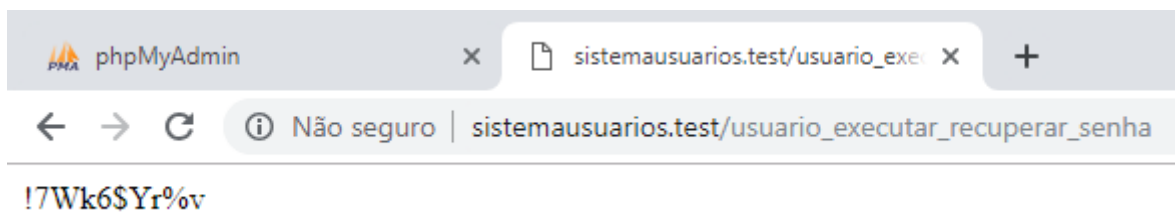


Email:

teste@teste.pt

Recuperar senha

[Voltar](#)



Senha: aaabbb

Senha original encriptada:

\$2y\$10\$SnmINuEGVIXNY9X4y9Fk2urh0FVU9wpjHij7fC0JXh5W33Y8UWeou

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**

Servidor de recebimento de e-mails (POP)	pop.gmail.com Requer SSL: Sim Porta: 995
Servidor de envio de e-mails (SMTP)	smtp.gmail.com Requer SSL: Sim Requer TLS: Sim (se disponível) Requer autenticação: sim Porta para TLS/STARTTLS: 587  Se você usa o Gmail com sua conta do trabalho ou da escola, verifique a configuração correta do SMTP com o <a href="#">administrador</a> .
Tempos limite do servidor	Mais que um minuto (recomenda-se cinco)
Nome completo ou Nome de exibição	Seu nome
Nome da conta, Nome do usuário ou E-mail	Seu endereço de e-mail
Senha	Sua senha do Gmail

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 Inclua uma nova conta, com senha: 1234567

 SISTEMA DE  
**USUÁRIOS**

Usuário:

Senha:

Repetir senha:

Email:

☒ Concordo com os termos e condições.

[Criar nova conta](#)

[Voltar](#)

Servidor: localhost

Banco de dados: sistemausuarios

Tabela: usuarios

Dados

Consulta

sistemausuarios.usuarios: 3 registros totais (aproximadamente)

Próximo

Mostrar todos

Ordem

Colunas (7/7)

Filtro

id_usuario	usuario	senha	email	remember_token	created_at	updated_at
1	joao	\$2y\$10\$Fz2tW5nLH98eYZHDFYT.A9Tfnd3AF/P6jOKg8grXMIISN3DM18C	teste@teste.pt	(NULL)	2019-02-19 01:51:42	2019-02-19 18:34:17
2	ana	\$2y\$10\$yj8tr1ZR0siqV/rpVJYN.jHqR940ztZuQKVs7vdCNO.BQM4AMC2	ana@teste.pt	(NULL)	2019-02-19 01:58:09	2019-02-19 01:58:09
3	betopinheiro	\$2y\$10\$pvh8jd/uo9S7NDONZ0ednOdLJMN/9VgP6FjxS0DcpPbleMsxx5ia	pinheiroberto61@gmail.com	(NULL)	2019-02-19 20:16:45	2019-02-19 20:16:45

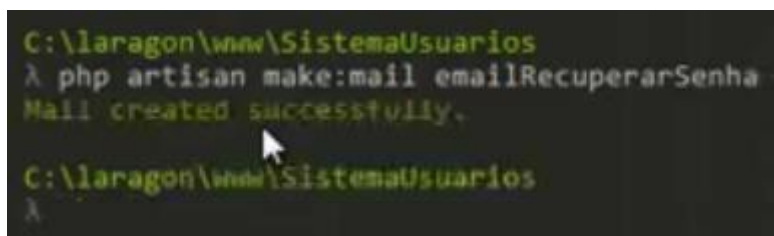
## Aula 083 – Exercício 2 - Sistema de usuários - Parte 16

### Enviando o email para o usuário

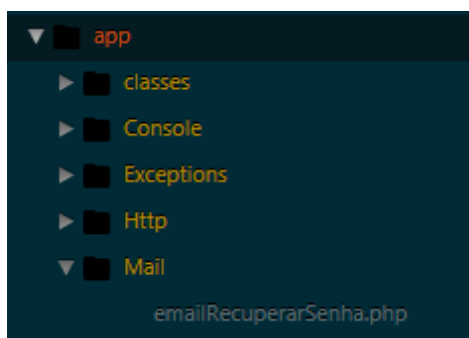
#### Criando a classe de email

No terminal, entre com o seguinte comando:

```
php artisan make:mail emailRecuperarSenha
```



```
C:\laragon\www\SistemaUsuarios
λ php artisan make:mail emailRecuperarSenha
Mail created successfully.
C:\laragon\www\SistemaUsuarios
λ
```



- 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';

}
```

Sistema de Usuários

Não seguro | sistemausuarios.test/usuario\_frm\_recuperar\_senha

 SISTEMA DE  
**USUÁRIOS**

Email:

pinheiroberto61@gmail.com



Recuperar senha


[Voltar](#)










sistemausuarios.test/usuario\_exe

Não seguro | sistemausuarios.test/usuario\_executar\_recuperar\_senha

Ok

| Principal   | Social   | Promoções |
|---|--|-----------|
| <input type="checkbox"/>  eu     | <b>Email Recuperar Senha</b> - Este email foi-lhe enviado por ter solicitado a recuperação da senha, ... 19:08 |           |
| <input type="checkbox"/>  Google | <b>Alerta de segurança crítico</b> - O acesso para aplicações menos seguras foi ativado pinheirober... 17:02   |           |

 Pesquisar correio

## Email Recuperar Senha Caixa de entrada x



**Example** <pinheiroberto61@gmail.com>  
para eu ▾

Este email foi-lhe enviado por ter solicitado a recuperação da senha.

Nova senha: lbd%KdFK4w

Na próxima visita ao site SistemaUsuarios, use a nova senha.

 Responder

 Encaminhar

Senha: 1234567

Senha original encriptada:


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

Senha aleatória: lbd%KdFK4w

Senha aleatória encriptada:

\$2y\$10\$Kqd7c./0pUj/BWzivMgl5.sKCvdGdDKJm24U9rbSZIH0jT8DZONdy

Entrando com a senha: 1234567



SISTEMA DE  
**USUÁRIOS**

A senha está incorreta.

Usuário:

Senha:

Entrar

[Recuperar senha](#)  
[Criar nova conta](#)

Entrando com a senha enviada: lbd%KdFK4w

 SISTEMA DE  
**USUÁRIOS**

Usuário:


Senha:

[Entrar](#)

[Recuperar senha](#)  
[Criar nova conta](#)

Sistema de Usuários x +

← → ↻ ⓘ Não seguro | sistemausuarios.test 🔑 ☆

 SISTEMA DE  
**USUÁRIOS**

Estou cá dentro!

Usuário: **betopinheiro**

[Logout](#)

## 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>
```

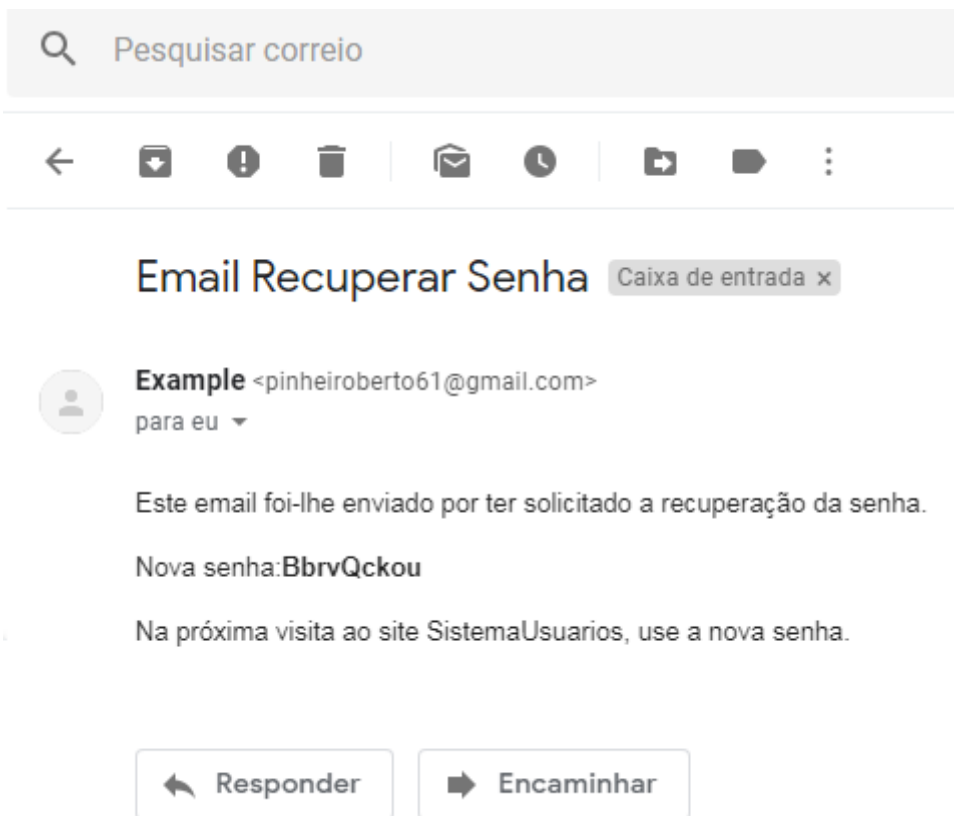
```
        <p><a href="/">Voltar</a></p>
```

```
    </div>
```

```
</div>
```

```
@endsection
```





Senha anterior: lbd%KdFK4w

Nova senha: BbrvQckou

Entrando com a nova senha:

