Feature: Esqueci minha senha

* Caso o usuário tenha esquecido sua senha de acesso, o mesmo deverá acessar a página de recuperação de senha e informar o seu email cadastrado na aplicação.
* O Sistema deverá validar o email informado e então gerar uma nova senha no banco de dados para o usuário. Esta senha deverá ser enviada para o email do usuário.
* O Usuário irá utilizar esta senha recebida por email para acessar o sistema, e dentro do sistema o usuário poderá alterar sua senha.

**Primeiro passo:**

Criar uma classe de modelo para capturar os campos do formulário de recuperação de senha:

/Models/RecuperarSenhaModel.java

Interface gráfica do usuário, Aplicativo, Email

Descrição gerada automaticamente

**package** br.com.cotiinformatica.models;

**import** lombok.AllArgsConstructor;

**import** lombok.Getter;

**import** lombok.NoArgsConstructor;

**import** lombok.Setter;

**import** lombok.ToString;

@Setter

@Getter

@NoArgsConstructor

@AllArgsConstructor

@ToString

**public** **class** RecuperarSenhaModel {

**private** String email;

}

No controlador, precisamos utilizar a classe Model para fazer a captura dos campos do formulário:

/controllers/RecuperarSenhaController.java

package br.com.cotiinformatica.controller;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.servlet.ModelAndView;

import br.com.cotiinformatica.models.RecuperarSenhaModel;

@Controller

public class RecuperarSenhaController {

@RequestMapping(value = "/recuperar-senha") //ROTA (navegação)

public ModelAndView recuperarSenha() {

// WEB-INF/views/recuperar-senha.jsp

ModelAndView modelAndView = new ModelAndView

("recuperar-senha");

**modelAndView.addObject("model",**

**new RecuperarSenhaModel());**

return modelAndView;

}

}

Agora, na página, vamos utilizar este objeto **model** para fazer a captura do formulário de recuperação de senha.

/WEB-INF/views/recuperar-senha.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<%@ taglib uri=*"http://www.springframework.org/tags/form"* prefix=*"form"* %>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

<!-- CDN da folha de estilos CSS do bootstrap -->

<link href=*"https://cdn.jsdelivr.net/npm/bootstrap*

*@5.3.0-alpha1/dist/css/bootstrap.min.css"* rel=*"stylesheet"* />

<style>

**label***.error* { color: *#df4759*; }

**input***.error* { border: *2px solid #df4759*; }

</style>

</head>

<body>

<div class=*"row"*>

<div class=*"col-md-4 offset-md-4"*>

<div class=*"card mt-5"*>

<div class=*"card-body"*>

<div class=*"text-center"*>

<h2>Sistema de Contas</h2>

<h5>Recuperação de Senha</h5>

</div>

<hr/>

**<div class=*"text-success"*>**

**<strong>${mensagem\_sucesso}</strong>**

**</div>**

**<div class=*"text-danger"*>**

**<strong>${mensagem\_erro}</strong>**

**</div>**

**<form id=*"form\_recuperarsenha"***

**action=*"recuperacao-de-senha"* method=*"post"*>**

<div class=*"mb-2"*>

<label>Entre com o seu

email:</label>

**<form:input path=*"model.email"***

**type=*"text"* name=*"email"* class=*"form-control"*/>**

</div>

<div class=*"mb-2 d-grid"*>

<input type=*"submit"*

value=*"Confirmar"* class=*"btn btn-primary"*/>

</div>

<div class=*"mb-2 d-grid"*>

<a href=*"/sistemacontas/"*

class=*"btn btn-light"*>

Já possui cadastro?

<strong>Acesse aqui!</strong>

</a>

</div>

</form>

</div>

</div>

</div>

</div>

<!-- CDN do arquivo javascript do bootstrap -->

<script src=*"https://cdn.jsdelivr.net/npm/bootstrap*

*@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"*></script>

<!-- CDN do arquivo javascript do JQuery -->

<script src=*"https://code.jquery.com/jquery-3.6.3.min.js"*></script>

<!-- CDN dos arquivos da biblioteca JQuery Validation -->

<script src=*"https://cdnjs.cloudflare.com/ajax/libs/jquery-*

*validate/1.19.5/jquery.validate.min.js"*></script>

<script src=*"https://cdnjs.cloudflare.com/ajax/libs/jquery-*

*validate/1.19.5/additional-methods.min.js"*></script>

<script src=*"https://cdnjs.cloudflare.com/ajax/libs/jquery-*

*validate/1.19.5/localization/messages\_pt\_BR.min.js"*></script>

<script>

$(document).ready(**function**() {

$("#form\_recuperarsenha").validate({

rules: {

'email' : { required: **true**, email : **true** }

}

});

})

</script>

</body>

</html>

/pom.xml

Arquivo de configuração do projeto utilizado pelo framework **Apache Maven** para instalação das bibliotecas e dependências utilizadas pelo projeto.

Desenho de um animal

Descrição gerada automaticamente com confiança média

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

<https://mvnrepository.com/artifact/org.apache.commons/commons-email/1.5>

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

**<!-- https://mvnrepository.com/artifact**

**/org.apache.commons/commons-email -->**

**<dependency>**

**<groupId>org.apache.commons</groupId>**

**<artifactId>commons-email</artifactId>**

**<version>1.5</version>**

**</dependency>**

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**Instalando a biblioteca Java Faker:**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

<https://mvnrepository.com/artifact/com.github.javafaker/javafaker/1.0.2>

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

**<!-- https://mvnrepository.com/artifact  
 /com.github.javafaker/javafaker -->**

**<dependency>**

**<groupId>com.github.javafaker</groupId>**

**<artifactId>javafaker</artifactId>**

**<version>1.0.2</version>**

**</dependency>**

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente com confiança média

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>sistemacontas</groupId>

<artifactId>sistemacontas</artifactId>

<version>1.0</version>

<packaging>war</packaging>

<name>sistemacontas</name>

<url>http://maven.apache.org</url>

<properties>

<java.version>1.6</java.version>

<spring.version>3.1.0.RELEASE</spring.version>

<cglib.version>2.2.2</cglib.version>

</properties>

<dependencies>

<!-- Spring core & mvc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-orm</artifactId>

<version>${spring.version}</version>

<type>jar</type>

<scope>compile</scope>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-test</artifactId>

<version>${spring.version}</version>

<type>jar</type>

<scope>test</scope>

</dependency>

<!-- CGLib for @Configuration -->

<dependency>

<groupId>cglib</groupId>

<artifactId>cglib-nodep</artifactId>

<version>${cglib.version}</version>

<scope>runtime</scope>

</dependency>

<!-- Servlet Spec -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>servlet-api</artifactId>

<version>2.4</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>javax.servlet.jsp</groupId>

<artifactId>jsp-api</artifactId>

<version>2.1</version>

<scope>provided</scope>

</dependency>

<!-- https://mvnrepository.com/artifact

/org.postgresql/postgresql -->

<dependency>

<groupId>org.postgresql</groupId>

<artifactId>postgresql</artifactId>

<version>42.5.1</version>

</dependency>

<!-- https://mvnrepository.com/artifact

/org.projectlombok/lombok -->

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<version>1.18.24</version>

<scope>provided</scope>

</dependency>

**<!-- https://mvnrepository.com/artifact/org.apache.commons  
/commons-email -->**

**<dependency>**

**<groupId>org.apache.commons</groupId>**

**<artifactId>commons-email</artifactId>**

**<version>1.5</version>**

**</dependency>**

**<!-- https://mvnrepository.com/artifact  
/com.github.javafaker/javafaker -->**

**<dependency>**

**<groupId>com.github.javafaker</groupId>**

**<artifactId>javafaker</artifactId>**

**<version>1.0.2</version>**

**</dependency>**

</dependencies>

<repositories>

<repository>

<id>springsource-milestones</id>

<name>SpringSource Milestones Proxy</name>

<url><https://oss.sonatype.org/content>

/repositories/springsource-milestones</url>

</repository>

</repositories>

<build>

<finalName>sistemacontas</finalName>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>2.0.2</version>

<configuration>

<source>${java.version}</source>

<target>${java.version}</target>

</configuration>

</plugin>

</plugins>

</build>

</project>

Criando uma classe no projeto capaz de enviar   
email para os usuários da aplicação:

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

**package** br.com.cotiinformatica.messages;

**public** **class** EmailMessage {

// método estático para envio de emails

**public** **static** **void** sendMessage() **throws** Exception {

// **TODO**

}

}

**DTO – Data Transfer Object**

Conceito utilizado para classes que terão como finalidade fazer o transporte de dados entre camadas de um sistema. É por padrão um JavaBean.

Interface gráfica do usuário, Aplicativo, Tabela, Excel

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

**package** br.com.cotiinformatica.dtos;

**import** lombok.AllArgsConstructor;

**import** lombok.Getter;

**import** lombok.NoArgsConstructor;

**import** lombok.Setter;

**import** lombok.ToString;

@Setter

@Getter

@NoArgsConstructor

@AllArgsConstructor

@ToString

**public** **class** EmailDTO {

**private** String mailTo;

**private** String subject;

**private** String body;

}

Interface gráfica do usuário, Aplicativo, Tabela, Excel

Descrição gerada automaticamente

**package** br.com.cotiinformatica.messages;

**import** java.util.Properties;

**import** javax.mail.internet.MimeMessage;

**import** org.springframework.mail.javamail.JavaMailSenderImpl;

**import** org.springframework.mail.javamail.MimeMessageHelper;

**import** org.springframework.mail.javamail.MimeMessagePreparator;

**import** br.com.cotiinformatica.dtos.EmailDTO;

**public** **class** EmailMessage {

// atributos

**private** **static** **final** String ***ACCOUNT***

= "cotiaulajava@outlook.com";

**private** **static** **final** String ***PASSWORD*** = "@Admin123456";

**private** **static** **final** String ***SMTP*** = "smtp-mail.outlook.com";

**private** **static** **final** Integer ***PORT*** = 587;

// método estático para envio de emails

**public** **static** **void** sendMessage(**final** EmailDTO emailDTO)

**throws** Exception {

//Conta que irá fazer o envio dos emails deste sistema

JavaMailSenderImpl sender = **new** JavaMailSenderImpl();

sender.setUsername(***ACCOUNT***);

sender.setPassword(***PASSWORD***);

sender.setHost(***SMTP***);

sender.setPort(***PORT***);

//Configurações necessárias para envio do email

Properties properties = **new** Properties();

properties.put("mail.smtp.starttls.enable", "true");

properties.put("mail.smtp.auth", "true");

properties.put("mail.smtp.transport.protocol", "smtp");

sender.setJavaMailProperties(properties);

//Montando o conteúdo do email

MimeMessagePreparator preparator

= **new** MimeMessagePreparator() {

@Override

**public** **void** prepare(MimeMessage mimeMessage)

**throws** Exception {

MimeMessageHelper helper

= **new** MimeMessageHelper(mimeMessage);

helper.setFrom(***ACCOUNT***);

helper.setTo(emailDTO.getMailTo());

helper.setSubject(emailDTO.getSubject());

helper.setText(emailDTO.getBody());

}

};

//enviando o email..

sender.send(preparator);

}

}

Voltando no controlador:

/controller/RecuperarSenhaController.java

package br.com.cotiinformatica.controller;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.servlet.ModelAndView;

import com.github.javafaker.Faker;

import br.com.cotiinformatica.dtos.EmailDTO;

import br.com.cotiinformatica.entities.Usuario;

import br.com.cotiinformatica.messages.EmailMessage;

import br.com.cotiinformatica.models.RecuperarSenhaModel;

import br.com.cotiinformatica.repositories.UsuarioRepository;

@Controller

public class RecuperarSenhaController {

@Autowired

private UsuarioRepository usuarioRepository;

@RequestMapping(value = "/recuperar-senha") // ROTA (navegação)

public ModelAndView recuperarSenha() {

// WEB-INF/views/recuperar-senha.jsp

ModelAndView modelAndView = new ModelAndView("recuperar-senha");

modelAndView.addObject("model", new RecuperarSenhaModel());

return modelAndView;

}

@RequestMapping(value = "/recuperacao-de-senha",

method = RequestMethod.POST)

public ModelAndView recuperacaodesenha(RecuperarSenhaModel model) {

// WEB-INF/views/recuperar-senha.jsp

ModelAndView modelAndView = new ModelAndView("recuperar-senha");

try {

//consultando o usuário no banco de dados através do email

Usuario usuario = usuarioRepository.findByEmail(model.getEmail());

//verificando se o usuário foi encontrado

if(usuario != null) {

//gerando uma nova senha para o usuário

String novaSenha = new Faker().internet().password(8, 10);

//escrevendo a mensagem

EmailDTO emailDTO = new EmailDTO();

emailDTO.setMailTo(usuario.getEmail());

emailDTO.setSubject

("Recuperação de senha de acesso - Sistema Contas");

emailDTO.setBody("Olá, " + usuario.getNome()

+ "\n\nUma nova senha de acesso

foi gerada com sucesso!"

+ "\nUtilize a senha: " + novaSenha

+ "\n\nApós autenticar-se, você

poderá alterar esta senha através do menu 'Dados do usuário'."

+ "\n\nAtt\nEquipe Sistema

Contas");

//enviando a mensagem por email

EmailMessage.sendMessage(emailDTO);

//atualizando a senha no banco de dados

usuarioRepository.update

(usuario.getIdUsuario(), novaSenha);

modelAndView.addObject("mensagem\_sucesso",

"Recuperação de senha realizada com sucesso.");

model = new RecuperarSenhaModel();

}

else {

modelAndView.addObject("mensagem\_erro",

"Usuário não encontrado, verifique o email informado.");

}

}

catch(Exception e) {

modelAndView.addObject("mensagem\_erro",

"Erro: " + e.getMessage());

}

modelAndView.addObject("model", model);

return modelAndView;

}

}

**Executando:**

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Criando uma página que permita ao usuário alterar a sua senha:

/WEB-INF/views/admin/dados-usuario.jsp

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Para esta página, precisamos de um controlador que crie   
a sua rota de navegação **/admin/dados-usuario**

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**package** br.com.cotiinformatica.controller;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.servlet.ModelAndView;

@Controller

**public** **class** DadosUsuarioController {

@RequestMapping(value = "/admin/dados-usuario")

**public** ModelAndView dadosUsuario() {

// WEB-INF/views/admin/dados-usuario.jsp

ModelAndView modelAndView

= **new** ModelAndView("admin/dados-usuario");

**return** modelAndView;

}

}

**Componentização dos elementos**

**de Layout das páginas**

Qualquer conteúdo HTML que precisamos utilizar em

várias páginas JSP deverão ser isolados em componentes.

/WEB-INF/views/admin/components/navbar.jsp

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

<nav class=*"navbar navbar-expand-lg bg-body-tertiary"*>

<div class=*"container-fluid"*>

<a class=*"navbar-brand"* href=*"#"*>Sistema Contas</a>

<button class=*"navbar-toggler"* type=*"button"* data-bs-

toggle=*"collapse"*

data-bs-target=*"#navbarSupportedContent"*

aria-controls=*"navbarSupportedContent"* aria-

expanded=*"false"*

aria-label=*"Toggle navigation"*>

<span class=*"navbar-toggler-icon"*></span>

</button>

<div class=*"collapse navbar-collapse"*

id=*"navbarSupportedContent"*>

<ul class=*"navbar-nav me-auto mb-2 mb-lg-0"*>

<li class=*"nav-item"*><a class=*"nav-link"* aria-

current=*"page"*

href=*"/sistemacontas*

*/admin/dashboard"*>Dashboard</a></li>

<li class=*"nav-item"*><a class=*"nav-link"* aria-

current=*"page"*

href=*"/sistemacontas/admin/dados-*

*usuario"*>Dados do usuário</a></li>

<li class=*"nav-item dropdown"*><a

class=*"nav-link dropdown-toggle"* href=*"#"*

role=*"button"*

data-bs-toggle=*"dropdown"* aria-

expanded=*"false"*> Gerenciar

contas </a>

<ul class=*"dropdown-menu"*>

<li><a class=*"dropdown-item"*

href=*"/sistemacontas/admin*

*/cadastrar-contas"*>

Cadastrar contas</a></li>

<li><a class=*"dropdown-item"*

href=*"/sistemacontas*

*/admin/consultar-contas"*>Consultar contas</a></li>

</ul></li>

</ul>

<div class=*"d-flex"*>

<div>

<div>

<strong>${usuario.nome}</strong>

</div>

<div style="margin-top: *-8px !important*;">

<small>${usuario.email}</small>

</div>

</div>

&nbsp;&nbsp;&nbsp; <a href=*"/sistemacontas/logout"*

class=*"btn btn-outline-secondary"*

onclick="return confirm

('Deseja realmente sair do sistema?');">

Sair do sistema </a>

</div>

</div>

</div></nav>

Inserindo o componente **navbar** nas demais páginas:

/WEB-INF/views/admin/dashboard.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

<!-- CDN da folha de estilos CSS do bootstrap -->

<link href=*"https://cdn.jsdelivr.net/npm/bootstrap*

*@5.3.0-alpha1/dist/css/bootstrap.min.css"* rel=*"stylesheet"* />

</head>

<body>

**<!-- menu superior da aplicação -->**

**<jsp:include page=*"/WEB-INF/views/admin/components/navbar.jsp"*>**

**</jsp:include>**

<div class=*"container mt-3"*>

<h5>Seja bem vindo ao sistema de contas!</h5>

</div>

<!-- CDN do arquivo javascript do bootstrap -->

<script src=*"https://cdn.jsdelivr.net/npm/bootstrap*

*@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"*></script>

</body>

</html>

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Inserindo o componente **navbar** nas demais páginas:

/WEB-INF/views/admin/dados-usuario.jsp

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"ISO-8859-1"*>

<title>Insert title here</title>

<!-- CDN da folha de estilos CSS do bootstrap -->

<link href=*"https://cdn.jsdelivr.net/npm/bootstrap*

*@5.3.0-alpha1/dist/css/bootstrap.min.css"* rel=*"stylesheet"* />

<style>

**label***.error* { color: *#df4759*; }

**input***.error* { border: *2px solid #df4759*; }

</style>

</head>

<body>

<!-- menu superior da aplicação -->

<jsp:include page=*"/WEB-INF/views/admin/components/navbar.jsp"*>

</jsp:include>

<div class=*"container mt-3"*>

<div class=*"card"*>

<div class=*"card-body"*>

<h5>Dados do usuário</h5>

<div>Nome do usuário:

<strong>${usuario.nome}</strong></div>

<div>Email de acesso:

<strong>${usuario.email}</strong></div>

</div>

</div>

<div class=*"card mt-2"*>

<div class=*"card-body"*>

<h5>Alterar senha de acesso</h5>

<form id=*"form\_alterar\_senha"*>

<div class=*"row mb-2"*>

<div class=*"col-md-3"*>

<label>Nova senha:</label>

<input type=*"password"*

name=*"novaSenha"* id=*"novaSenha"* class=*"form-control"*/>

</div>

<div class=*"col-md-3"*>

<label>Confirme a nova

senha:</label>

<input type=*"password"*

name=*"novaSenhaConfirmacao"* id=*"novaSenhaConfirmacao"* class=*"form-control"*/>

</div>

</div>

<div class=*"row mb-2"*>

<div class=*"col-md-6"*>

<input type=*"submit"* class=*"btn*

*btn-success"*

value=*"Salvar alterações"*/>

</div>

</div>

</form>

<div class=*"mt-2 text-success"*>

<strong>${mensagem\_sucesso}</strong>

</div>

<div class=*"mt-2 text-danger"*>

<strong>${mensagem\_erro}</strong>

</div>

</div>

</div>

</div>

<!-- CDN do arquivo javascript do bootstrap -->

<script src=*"https://cdn.jsdelivr.net/npm/bootstrap*

*@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"*></script>

<!-- CDN do arquivo javascript do JQuery -->

<script src=*"https://code.jquery.com/jquery-3.6.3.min.js"*></script>

<!-- CDN dos arquivos da biblioteca JQuery Validation -->

<script src=*"https://cdnjs.cloudflare.com/ajax/libs/jquery-*

*validate/1.19.5/jquery.validate.min.js"*></script>

<script src=*"https://cdnjs.cloudflare.com/ajax/libs/jquery-*

*validate/1.19.5/additional-methods.min.js"*></script>

<script src=*"https://cdnjs.cloudflare.com/ajax/libs/jquery-*

*validate/1.19.5/localization/messages\_pt\_BR.min.js"*></script>

<script>

$(document).ready(**function**() {

$("#form\_alterar\_senha").validate({

rules: {

'novaSenha' : { required: **true**,

minlength : 8, maxlength: 20},

'novaSenhaConfirmacao' : { required: **true**,

equalTo : "#novaSenha" }

}

});

})

</script>

</body>

</html>

**Executando:**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Para implementarmos essa funcionalidade, precisamos

criar uma classe de modelo de dados:

/models/AlterarSenhaModel.java

**package** br.com.cotiinformatica.models;

**import** lombok.AllArgsConstructor;

**import** lombok.Getter;

**import** lombok.NoArgsConstructor;

**import** lombok.Setter;

**import** lombok.ToString;

@Setter

@Getter

@NoArgsConstructor

@AllArgsConstructor

@ToString

**public** **class** AlterarSenhaModel {

**private** String novaSenha;

}

No controlador, precisamos criar um objeto para fazer a captura dos campos do formulário:

package br.com.cotiinformatica.controller;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.servlet.ModelAndView;

import br.com.cotiinformatica.models.AlterarSenhaModel;

@Controller

public class DadosUsuarioController {

@RequestMapping(value = "/admin/dados-usuario")

public ModelAndView dadosUsuario() {

// WEB-INF/views/admin/dados-usuario.jsp

ModelAndView modelAndView = new ModelAndView("admin/dados-usuario");

**modelAndView.addObject("model", new AlterarSenhaModel());**

return modelAndView;

}

}

Voltando na página:

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**Finalizando, vamos implementar o método POST no controlador:**

**package** br.com.cotiinformatica.controller;

**import** javax.servlet.http.HttpServletRequest;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Controller;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

**import** org.springframework.web.servlet.ModelAndView;

**import** br.com.cotiinformatica.entities.Usuario;

**import** br.com.cotiinformatica.models.AlterarSenhaModel;

**import** br.com.cotiinformatica.repositories.UsuarioRepository;

@Controller

**public** **class** DadosUsuarioController {

@Autowired

**private** UsuarioRepository usuarioRepository;

@RequestMapping(value = "/admin/dados-usuario")

**public** ModelAndView dadosUsuario() {

// WEB-INF/views/admin/dados-usuario.jsp

ModelAndView modelAndView = **new** ModelAndView

("admin/dados-usuario");

modelAndView.addObject("model", **new** AlterarSenhaModel());

**return** modelAndView;

}

@RequestMapping(value = "/admin/atualizar-senha",

method = RequestMethod.***POST***)

**public** ModelAndView atualizarSenha(AlterarSenhaModel model,

HttpServletRequest request) {

ModelAndView modelAndView

= **new** ModelAndView("admin/dados-usuario");

**try** {

//capturar o usuário autenticado em sessão

UsuarioDTO usuarioDTO = (UsuarioDTO)

request.getSession()

.getAttribute("usuario");

//atualizar a senha do usuário

usuarioRepository.update(usuario.getIdUsuario(),

model.getNovaSenha());

modelAndView.addObject("mensagem\_sucesso",

"Senha de acesso atualizada com sucesso.");

}

**catch**(Exception e) {

modelAndView.addObject("mensagem\_erro",

"Erro ao alterar senha: " + e.getMessage());

}

modelAndView.addObject("model", **new** AlterarSenhaModel());

**return** modelAndView;

}

}

**Executando:**

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

**Publicando o trabalho no GITHUB:**

Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 - Java WebDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas (main)

**$ git add .**

Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 - Java WebDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas (main)

**$ git commit -m 'Recuperação e alteração de senha do usuário'**

Samsung@DESKTOP-P9F6D9F MINGW64 ~/Desktop/COTI - Aulas/2023 - Java WebDeveloper SQS 18h as 22h (Início em 09.01)/workspace/sistemacontas (main)

**$ git push -u origin main**