

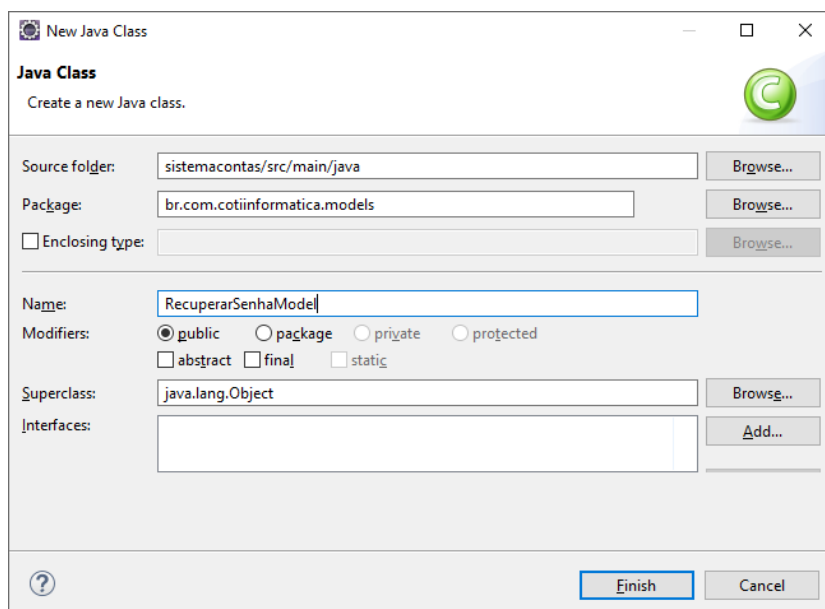
Feature: **Esqueci minha senha**

- 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**



```
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() {

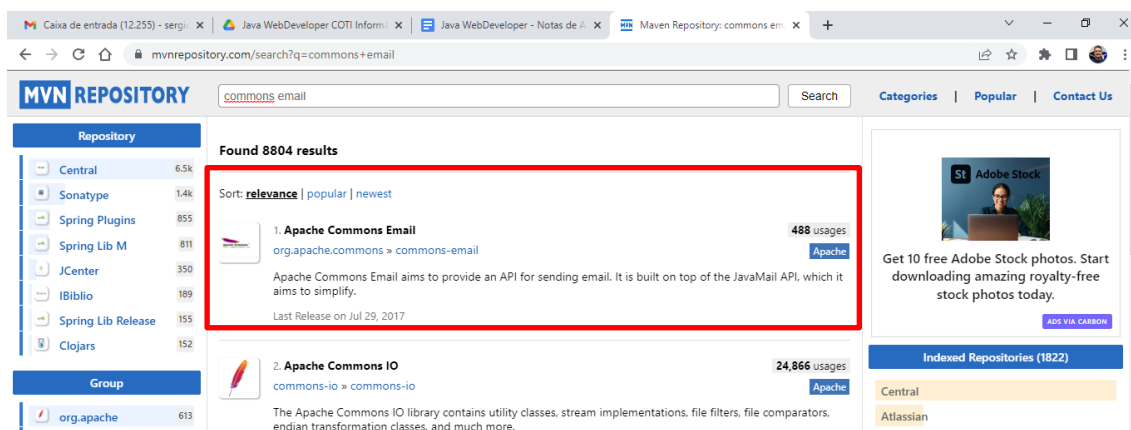
    $("#form_recuperarsenha").validate({
        rules: {
            'email' : { required: true, email : true }
        }
    });

})
</script>

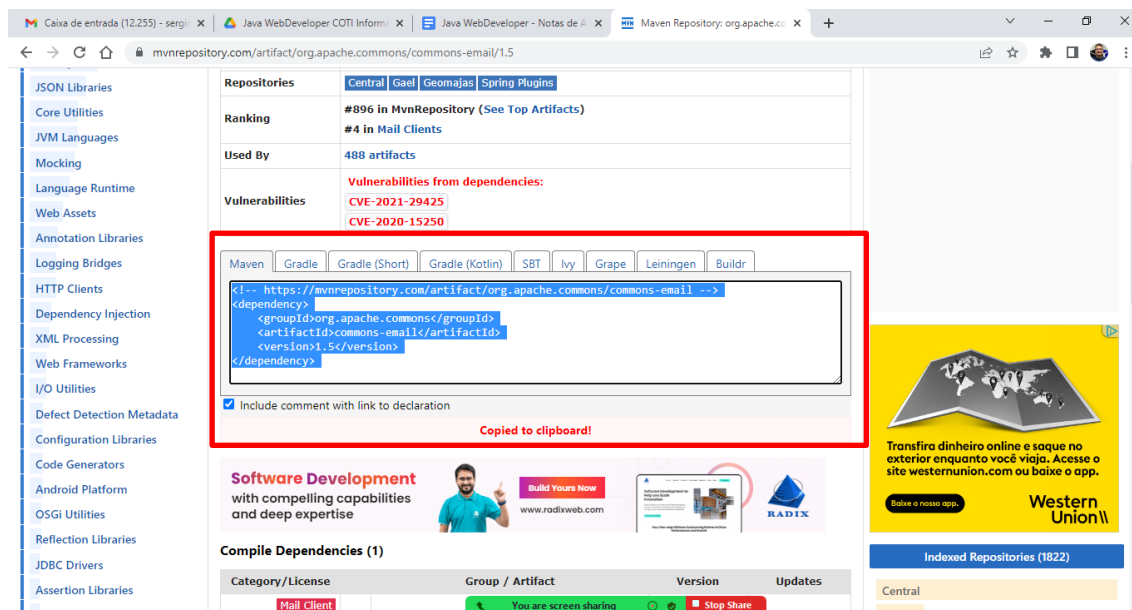
</body>
</html>
```

/pom.xml

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



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



The screenshot shows the Maven Repository website for the artifact `org.apache.commons:commons-email:1.5`. The page includes a sidebar with navigation links, a main content area with repository details, and a code snippet for the dependency declaration.

Repositories: Central, Gael, Geomajas, Spring Plugins

Ranking: #896 in MvnRepository (See Top Artifacts)

Used By: 488 artifacts

Vulnerabilities: CVE-2021-29425, CVE-2020-15250

Code Snippet:

```
<!-- https://mvnrepository.com/artifact/org.apache.commons/commons-email -->
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-email</artifactId>
  <version>1.5</version>
</dependency>
```

Include comment with link to declaration

Copied to clipboard!

Software Development with compelling capabilities and deep expertise

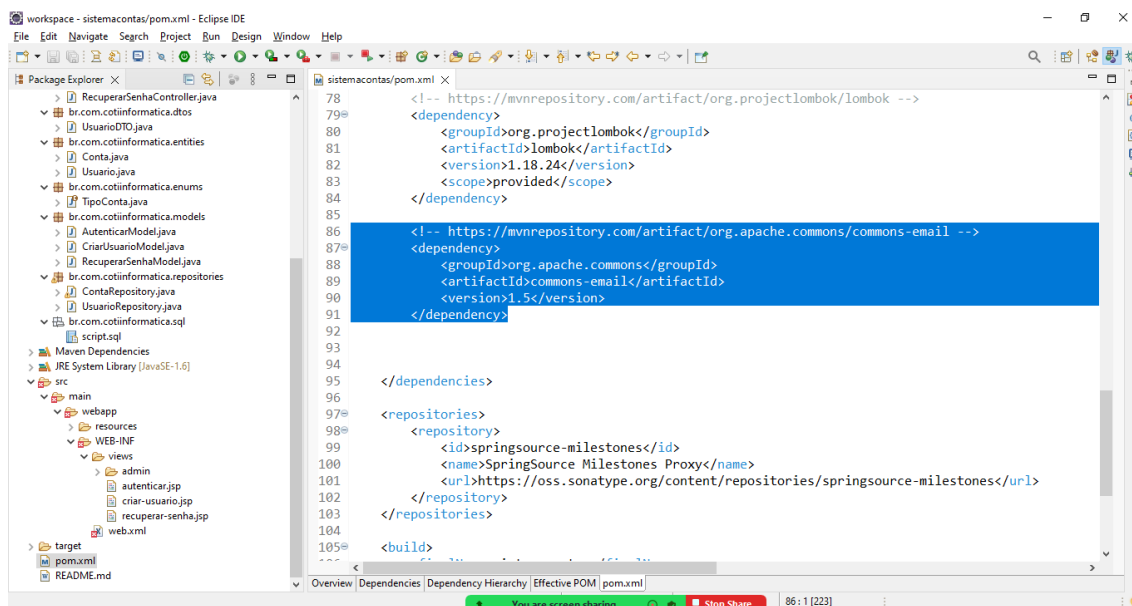
Build Your Now

www.radioweb.com

Compile Dependencies (1)

Category/License	Group / Artifact	Version	Updates
Mail Client	org.apache.commons:commons-email	1.5	

```
<!-- https://mvnrepository.com/artifact
/org.apache.commons/commons-email -->
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-email</artifactId>
  <version>1.5</version>
</dependency>
```



The screenshot shows the Eclipse IDE with the `pom.xml` file for the `sistemacontas` project. The file contains the dependency declaration for `org.apache.commons:commons-email:1.5`.

```
<!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <version>1.18.24</version>
  <scope>provided</scope>
</dependency>

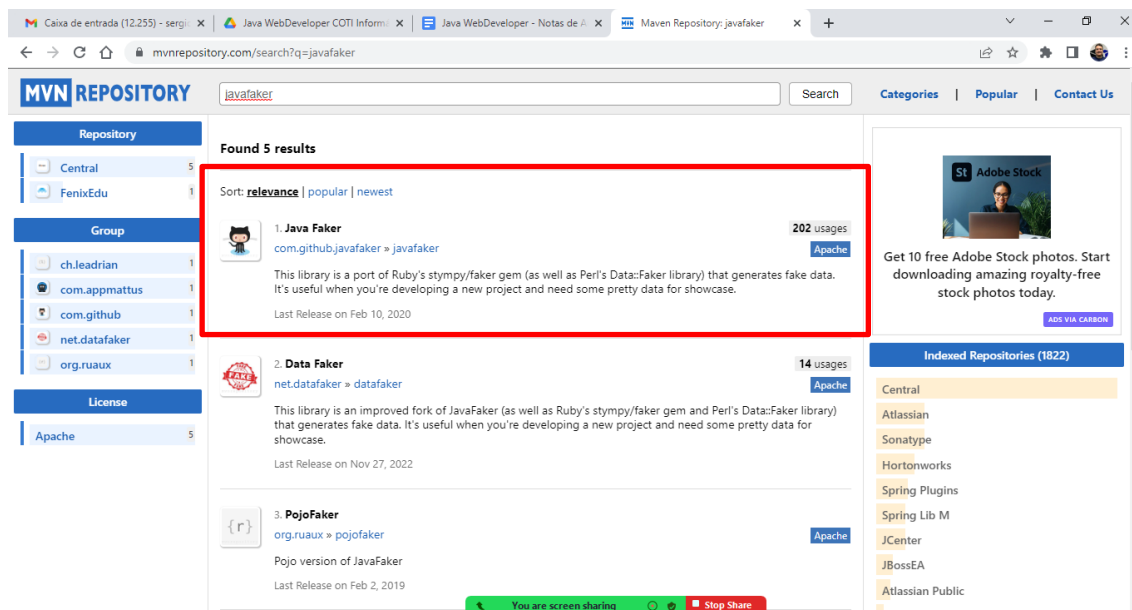
<!-- https://mvnrepository.com/artifact/org.apache.commons/commons-email -->
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-email</artifactId>
  <version>1.5</version>
</dependency>

</dependencies>

<repositories>
  <repository>
    <id>springsource-milestones</id>
    <name>SpringSource Milestones Proxy</name>
    <url>https://oss.sonatype.org/content/repositories/springsource-milestones</url>
  </repository>
</repositories>

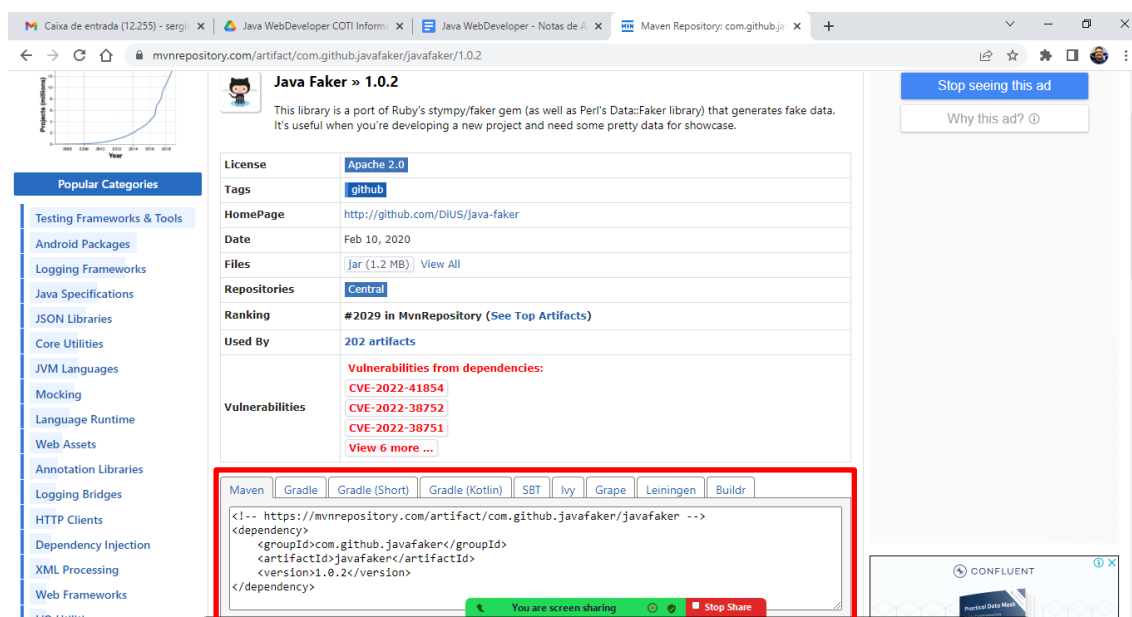
<build>
```

Instalando a biblioteca Java Faker:



The screenshot shows the Maven Repository search results for 'javafaker'. The search bar contains 'javafaker'. The results are sorted by relevance. The first result is 'Java Faker' by 'com.github.javafaker', which has 202 usages and is licensed under Apache. The description states: 'This library is a port of Ruby's stymphy/faker gem (as well as Perl's Data:Faker library) that generates fake data. It's useful when you're developing a new project and need some pretty data for showcase.' The last release was on Feb 10, 2020. The second result is 'Data Faker' by 'net.datafaker', which has 14 usages and is licensed under Apache. The description states: 'This library is an improved fork of JavaFaker (as well as Ruby's stymphy/faker gem and Perl's Data:Faker library) that generates fake data. It's useful when you're developing a new project and need some pretty data for showcase.' The last release was on Nov 27, 2022. The third result is 'PojoFaker' by 'org.ruauux', which is a Pojo version of JavaFaker and has a last release on Feb 2, 2019.

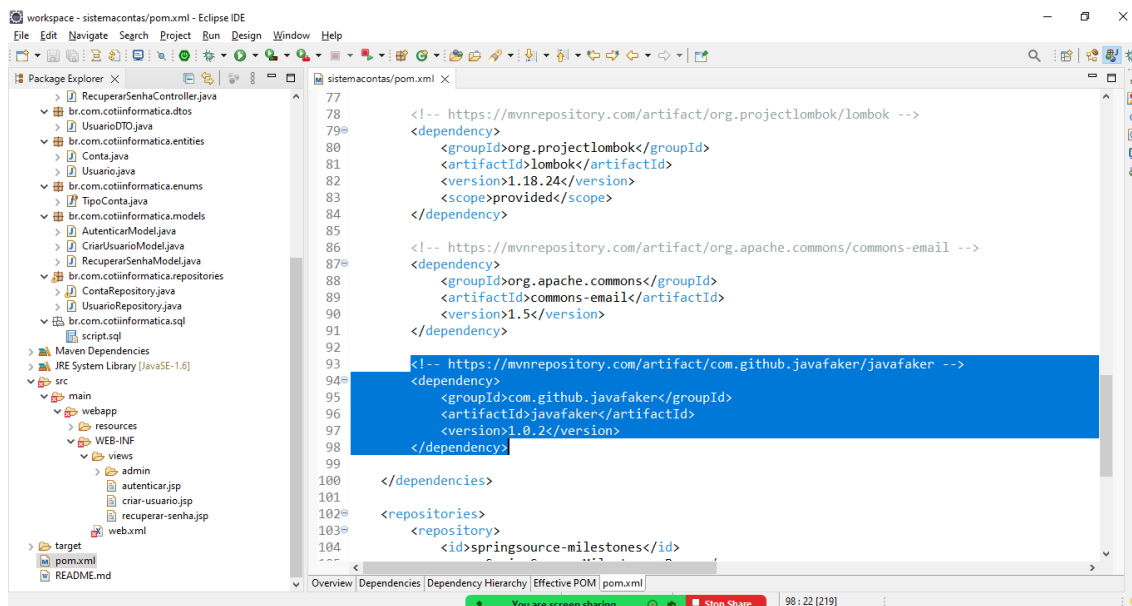
<https://mvnrepository.com/artifact/com.github.javafaker/javafaker/1.0.2>



The screenshot shows the Maven Repository artifact page for 'com.github.javafaker:java-faker:1.0.2'. The page displays the license (Apache 2.0), tags (github), homepage (http://github.com/DIUS/java-faker), date (Feb 10, 2020), files (jar (1.2 MB)), repositories (Central), ranking (#2029 in MvnRepository), and used by (202 artifacts). The vulnerabilities section lists three vulnerabilities: CVE-2022-41854, CVE-2022-38752, and CVE-2022-38751. The dependencies section is highlighted with a red box and shows the following XML snippet:

```
<!-- https://mvnrepository.com/artifact/com.github.javafaker/java-faker -->
<dependency>
  <groupId>com.github.javafaker</groupId>
  <artifactId>java-faker</artifactId>
  <version>1.0.2</version>
</dependency>
```

```
<!-- https://mvnrepository.com/artifact
/com.github.javafaker/java-faker -->
<dependency>
  <groupId>com.github.javafaker</groupId>
  <artifactId>java-faker</artifactId>
  <version>1.0.2</version>
</dependency>
```



```
<?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>
```

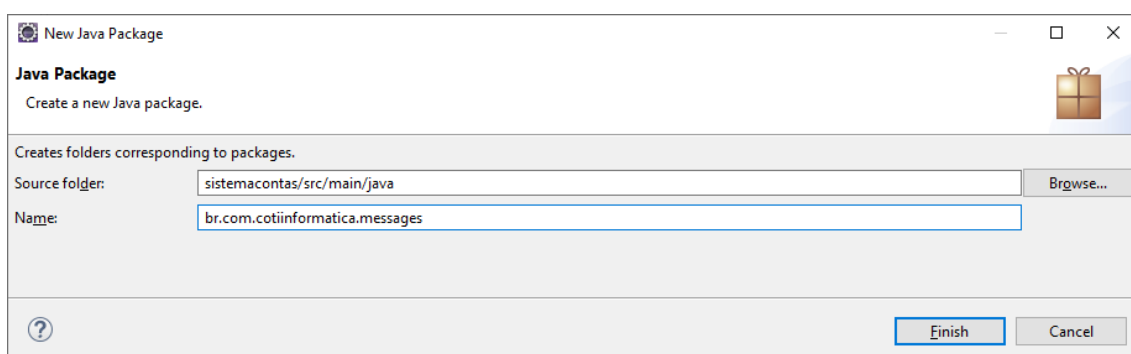


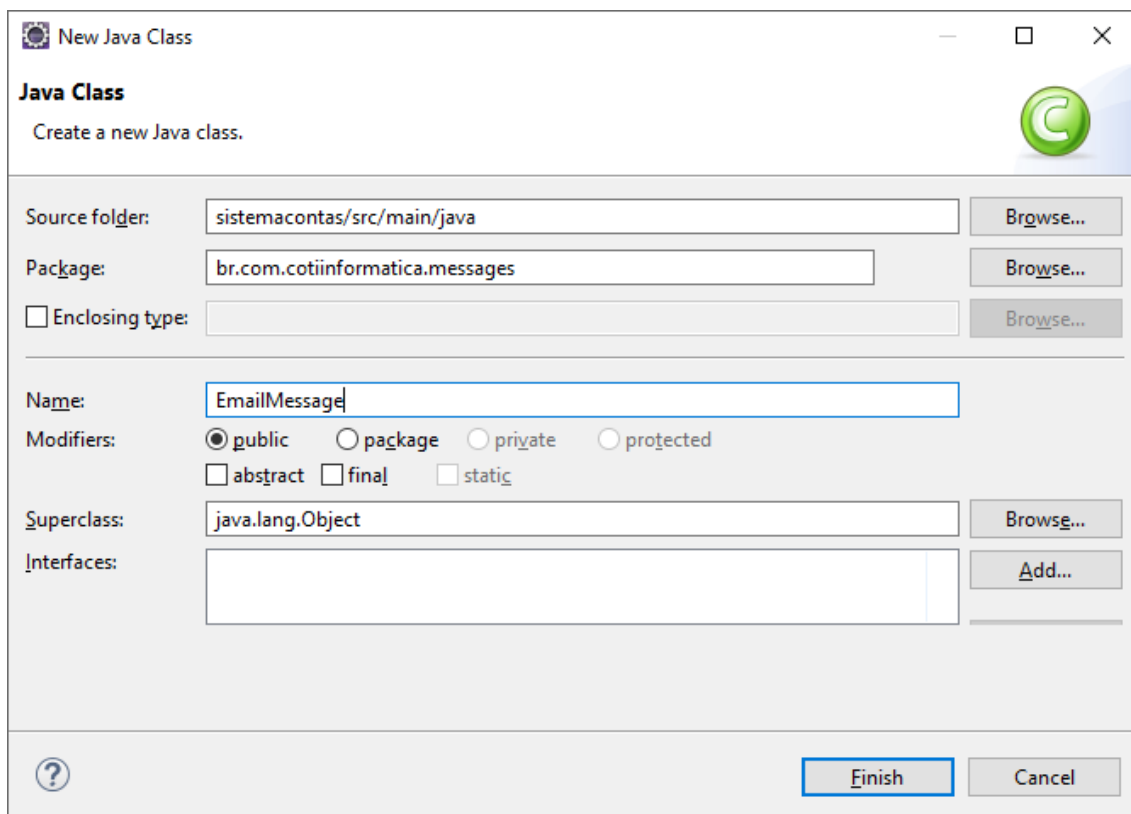
```
<!-- https://mvnrepository.com/artifact/org.apache.commons
/commons-email -->
<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-email</artifactId>
  <version>1.5</version>
</dependency>

<!-- 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:





New Java Class

Java Class
Create a new Java class.

Source folder: sistemacontas/src/main/java Browse...

Package: br.com.cotiinformatica.messages Browse...

☐ Enclosing type: Browse...

Name: EmailMessage

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add...

? Finish Cancel

```
package br.com.cotiinformatica.messages;
```

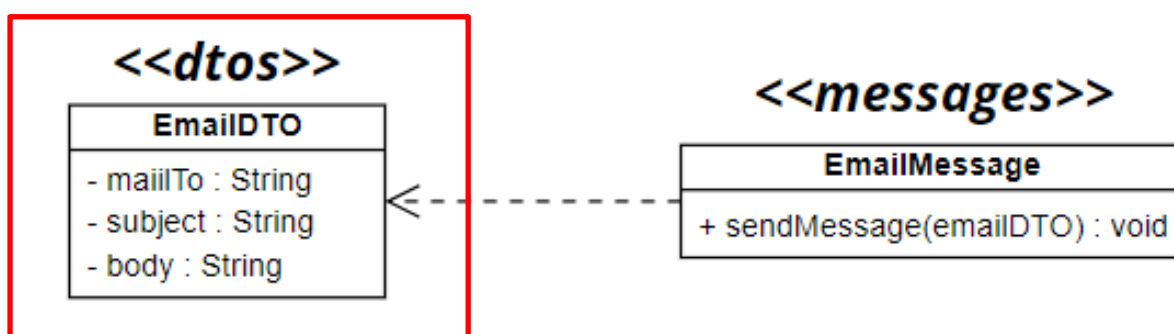
```
public class EmailMessage {
```

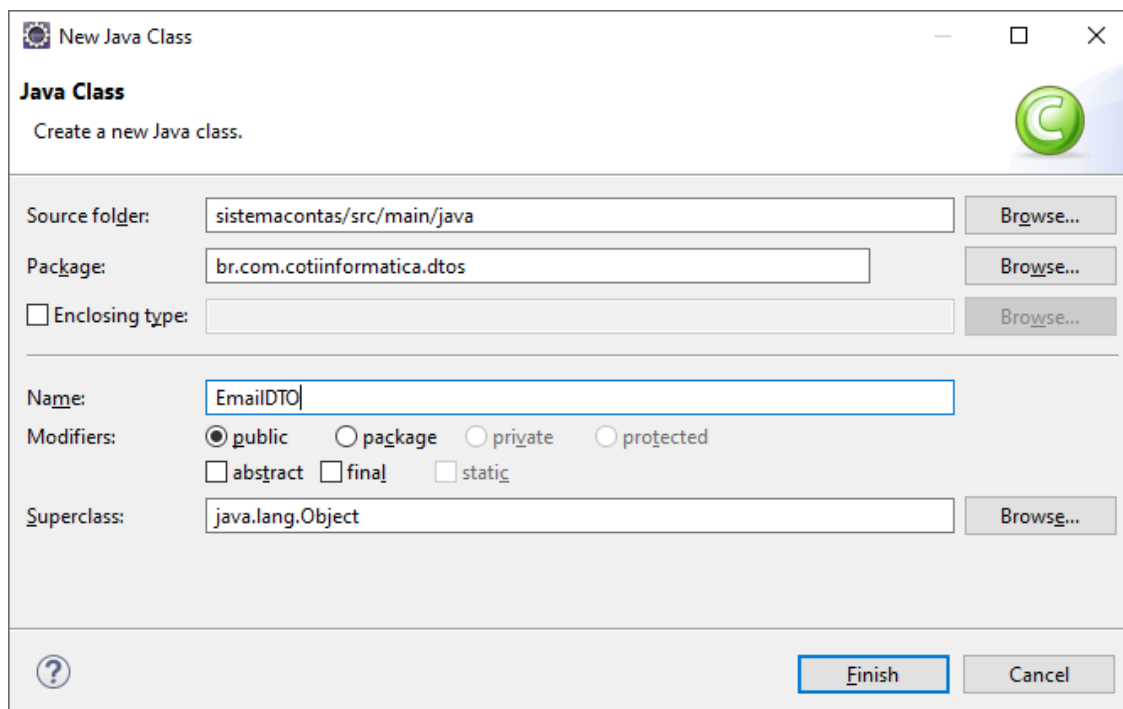
```
    // método estático para envio de emails
    public static void sendMessage() throws Exception {
        // TODO
    }
```

```
}
```

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.





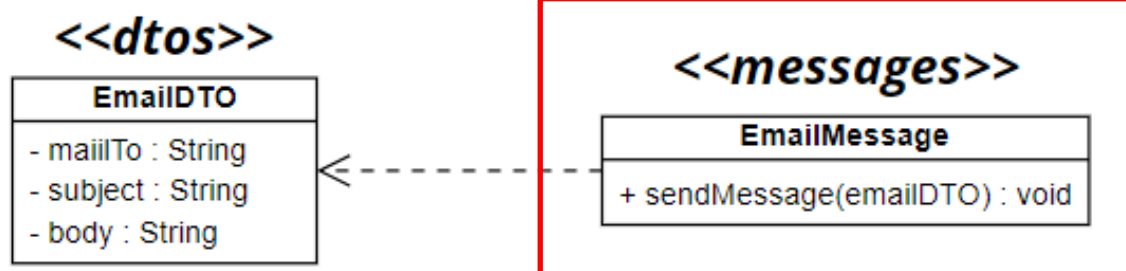
package br.com.cotiinformatica.dtos;

```
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;
```

```
}
```



```
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 recuperarSenha(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);

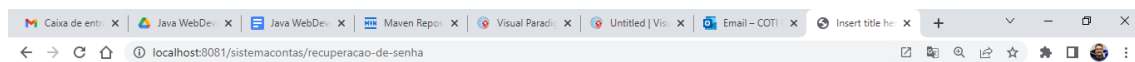
            //atualizando a senha no banco de dados
            usuarioRepository.update
                (usuario.getIdUsuario(), novaSenha);

            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:



Sistema de Contas

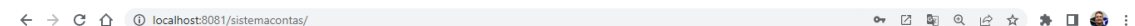
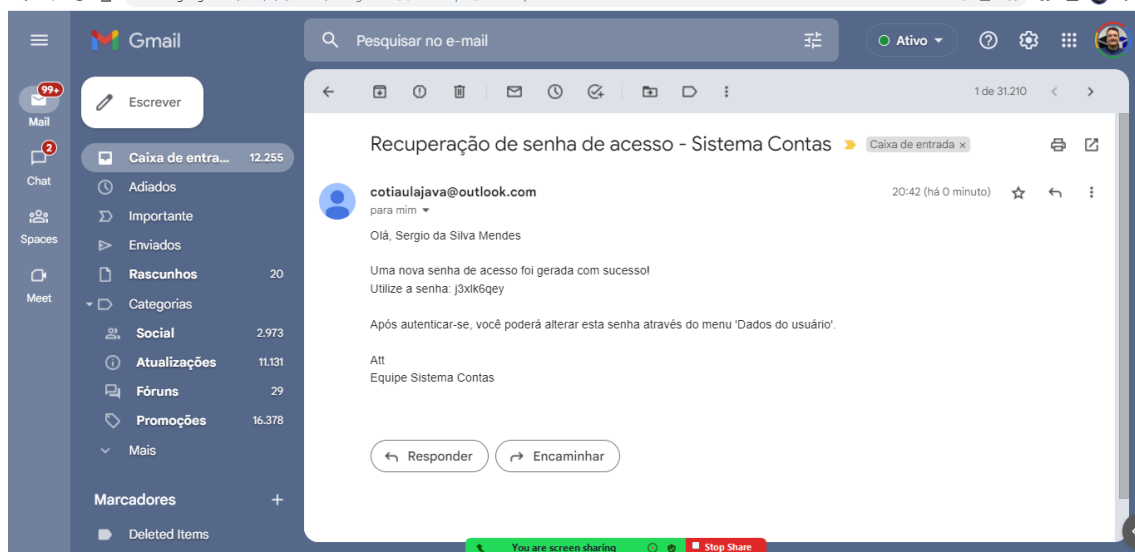
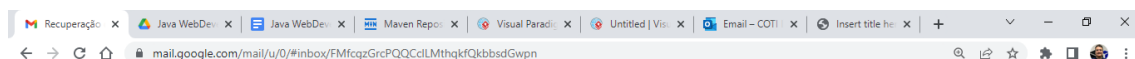
Recuperação de Senha

Recuperação de senha realizada com sucesso.

Entre com o seu email:

[Confirmar](#)

Já possui cadastro? [Acesse aqui!](#)



Sistema de Contas

Autenticação de usuários

Entre com o seu email:

Entre com a sua senha

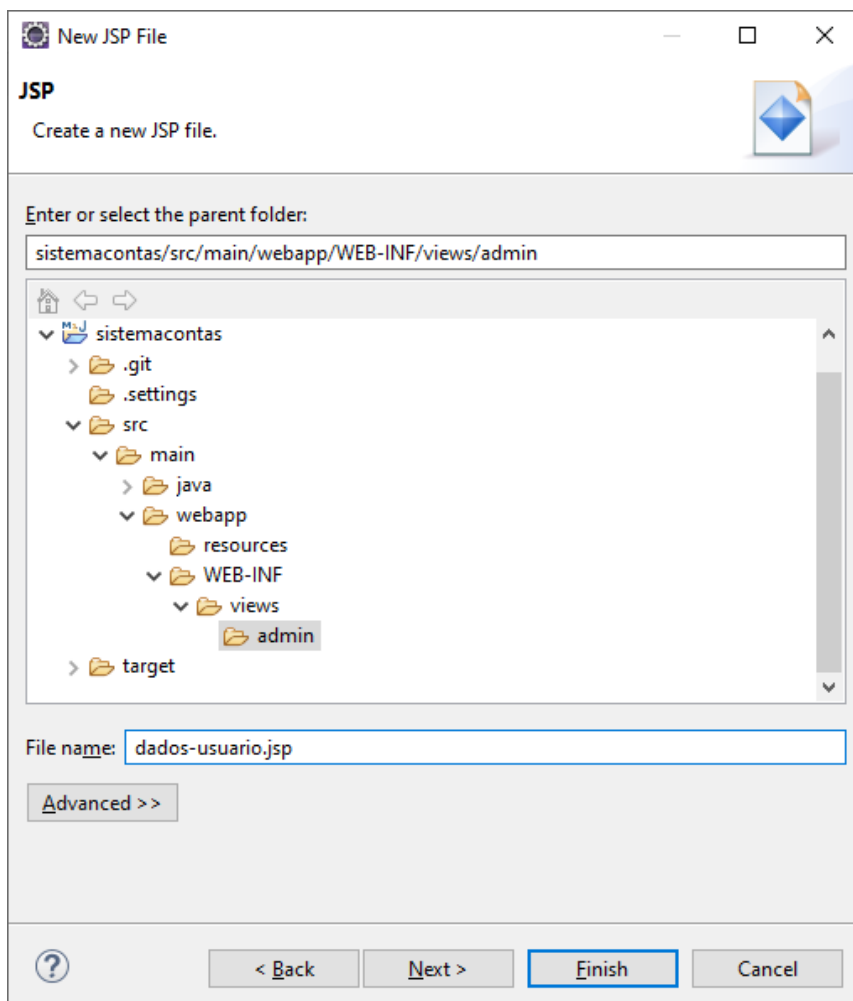
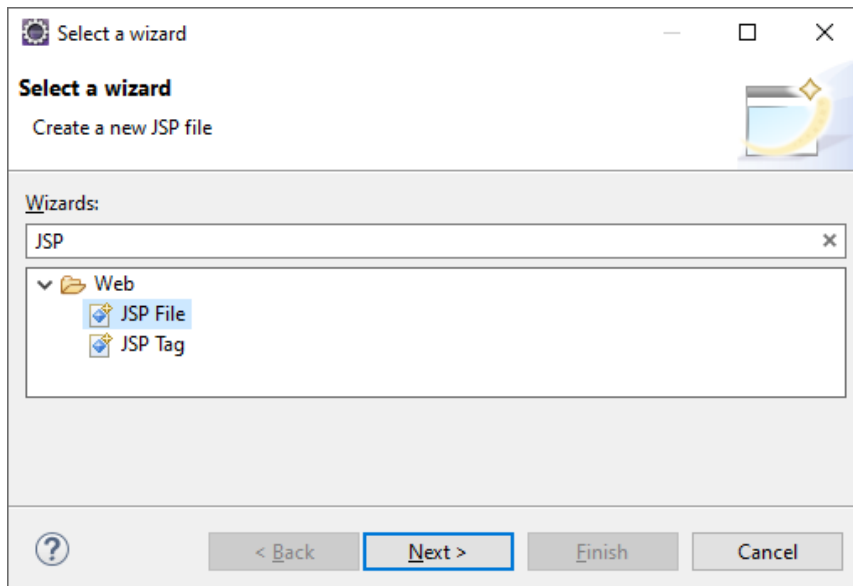
[Esqueci minha senha?](#)

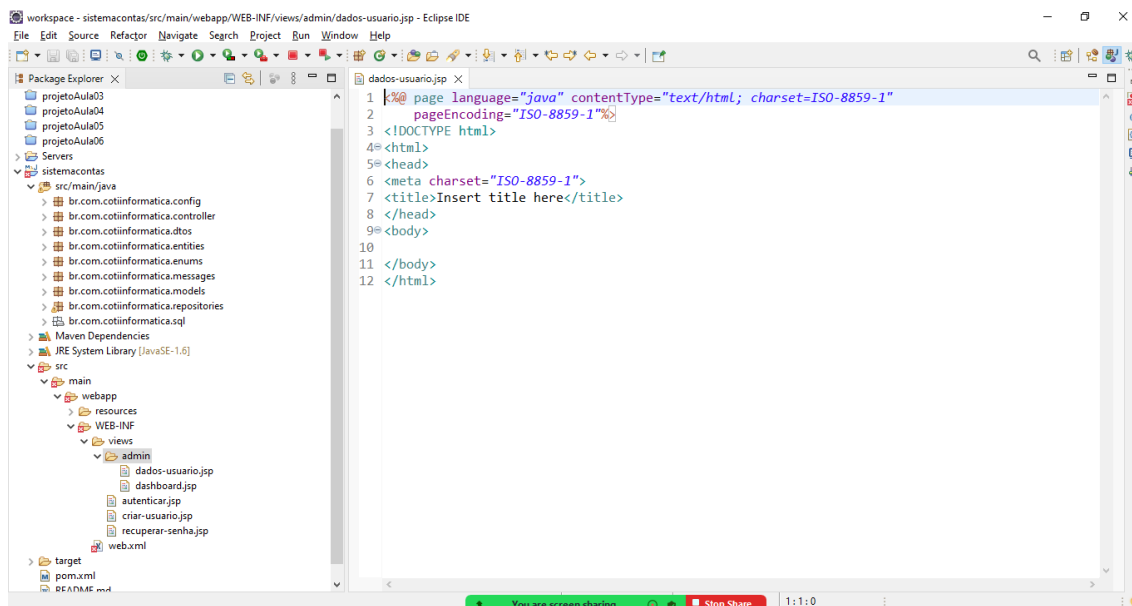
[Entrar](#)

Não possui cadastro? [Clique aqui!](#)

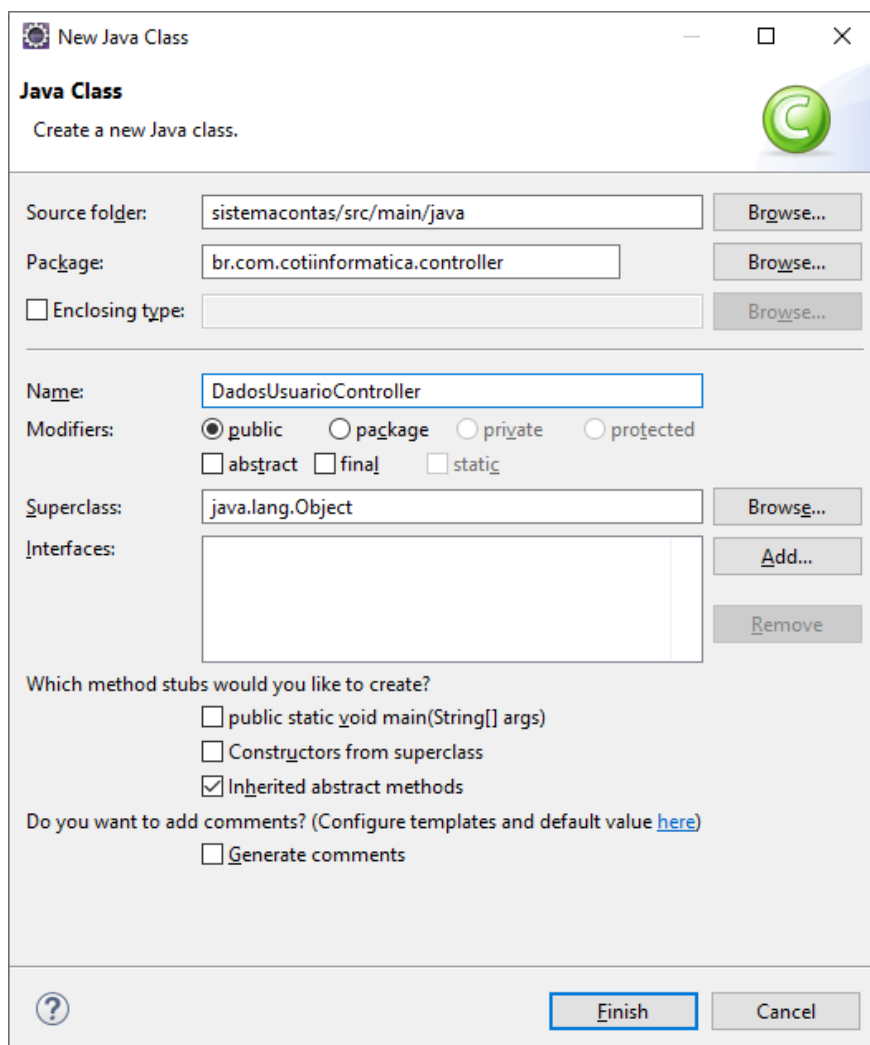
Criando uma página que permita ao usuário alterar a sua senha:

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





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



```
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() {

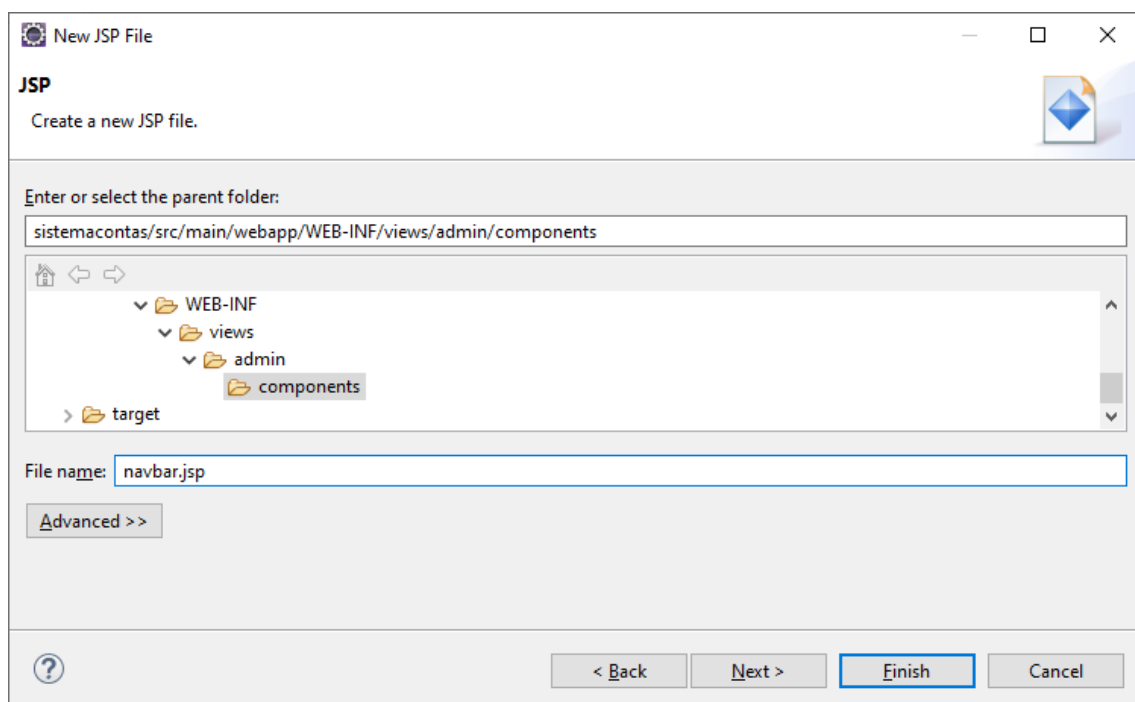
        // WEB-INF/views/admin/dados-usuario.jsp
        ModelAndView modelAndView
            = new ModelAndView("admin/dados-usuario");

        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



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>
```

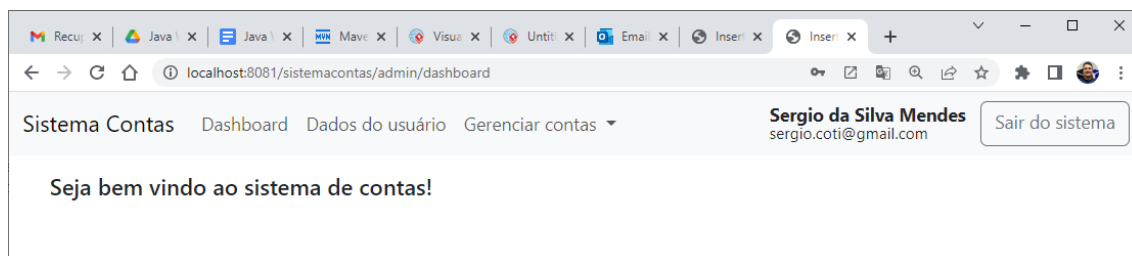
```
</div>
```

```
<!-- 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>
```



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>
            </form>
        </div>
    </div>

</div>
```

```
<div class="row mb-2">
    <div class="col-md-6">
        <input type="submit" class="btn
            btn-success"
            value="Salvar alterações"/>
        </div>
    </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>

<!-- 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>

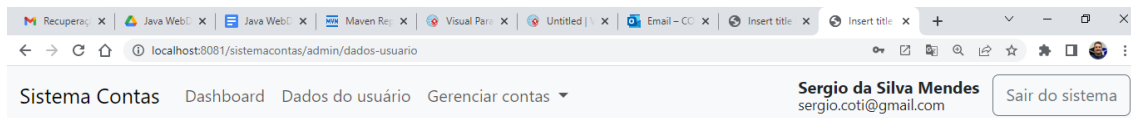
<script>
    $(document).ready(function() {

        $("#form_alterar_senha").validate({
            rules: {
                'novaSenha' : { required: true,
                    minlength : 8, maxlength: 20},
                'novaSenhaConfirmacao' : { required: true,
                    equalTo : "#novaSenha" }
            }
        });

    })
</script>

</body>
</html>
```

Executando:



Dados do usuário

Nome do usuário: **Sergio da Silva Mendes**
Email de acesso: **sergio.coti@gmail.com**

Alterar senha de acesso

Nova senha:

Confirme a nova senha:

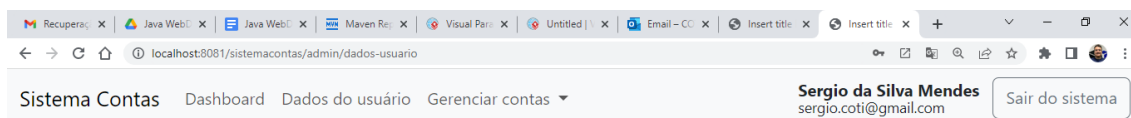
Este campo é requerido.

Este campo é requerido.

Salvar alterações

You are screen sharing

Stop Share



Dados do usuário

Nome do usuário: **Sergio da Silva Mendes**
Email de acesso: **sergio.coti@gmail.com**

Alterar senha de acesso

Nova senha:

Confirme a nova senha:

Por favor, forneça ao menos 8 caracteres.

Por favor, forneça o mesmo valor novamente.

Salvar alterações

You are screen sharing

Stop Share

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
```

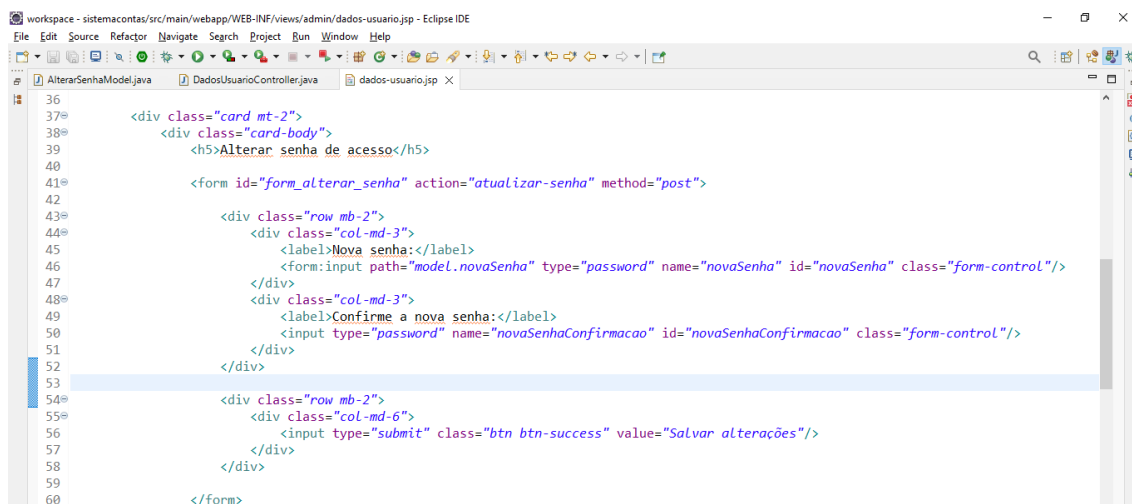
```
        ModelAndView modelAndView = new ModelAndView("admin/dados-usuario");
```

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

```
        return modelAndView;
```

```
    }
}
```

Voltando na página:



```
workspace - sistemacontas/src/main/webapp/WEB-INF/views/admin/dados-usuario.jsp - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
36
37<div class="card mt-2">
38<div class="card-body">
39<h5>Alterar senha de acesso</h5>
40
41<form id="form_alterar_senha" action="atualizar-senha" method="post">
42
43<div class="row mb-2">
44<div class="col-md-3">
45<label>Nova senha:</label>
46<form:input path="model.novaSenha" type="password" name="novaSenha" id="novaSenha" class="form-control"/>
47</div>
48<div class="col-md-3">
49<label>Confirme a nova senha:</label>
50<input type="password" name="novaSenhaConfirmacao" id="novaSenhaConfirmacao" class="form-control"/>
51</div>
52</div>
53
54<div class="row mb-2">
55<div class="col-md-6">
56<input type="submit" class="btn btn-success" value="Salvar alterações"/>
57</div>
58</div>
59
60</form>
```


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());
        }
    }
}
```

```

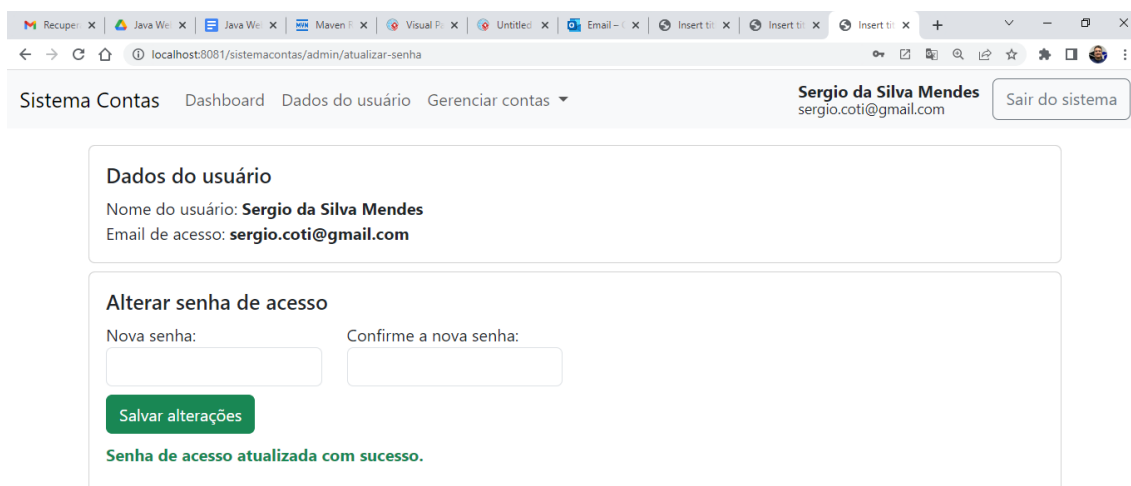
        modelAndView.addObject("mensagem_sucesso",
            "Senha de acesso atualizada com sucesso.");
    }
    catch(Exception e) {
        modelAndView.addObject("mensagem_erro",
            "Erro ao alterar senha: " + e.getMessage());
    }

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

    return modelAndView;
}
}

```

Executando:



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

```