

## Spring Web MVC

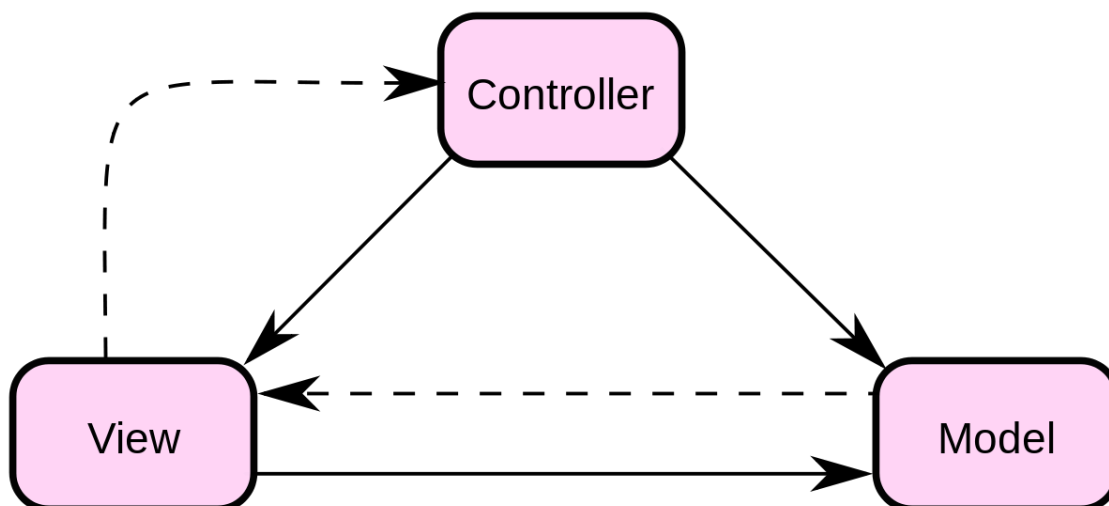
O Spring MVC é um framework que nos auxilia no desenvolvimento de aplicações web. Com ele, nós conseguimos ter facilidade e flexibilidade para trabalhar com requisições web.

O Spring MVC é uma excelente implementação do padrão MVC. Já temos aqui no blog um post específico sobre o padrão MVC, mas de forma resumida, MVC (acrônimo para Model-View-Controller) sugere uma maneira para você pensar na divisão de responsabilidades, principalmente dentro de um software web, dividindo as camadas em model, view, controller. Com ele podemos separar o código relativo à interface do usuário das regras de negócio. Ele já possui as principais funcionalidades que precisamos para o desenvolvimento: atender as [requisições HTTP](#), delegar responsabilidades de processamento de dados para outros componentes e preparar as respostas HTTP.



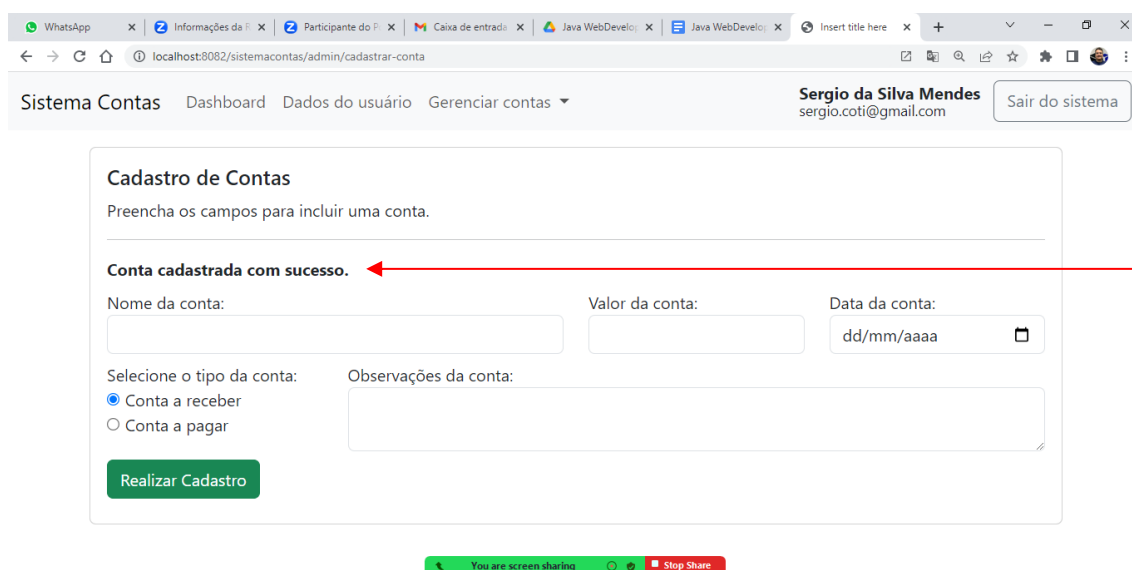
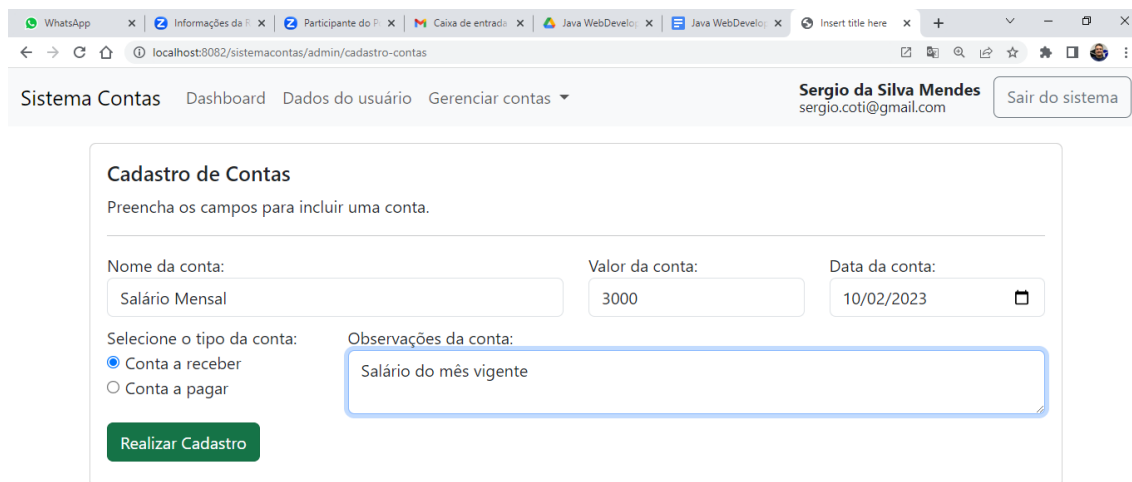
O Spring MVC trabalha de forma síncrona, ou seja, processa requisições uma a uma, por ordem de chegada. Enquanto uma requisição não for finalizada, ele não processa a seguinte.

O Spring MVC, assim como vários outros frameworks Java trabalha com Servlets, que trabalha ou de modo síncrono (versão anterior ao Servlet 3), ou de modo assíncrono (versão a partir do Servlet 3).



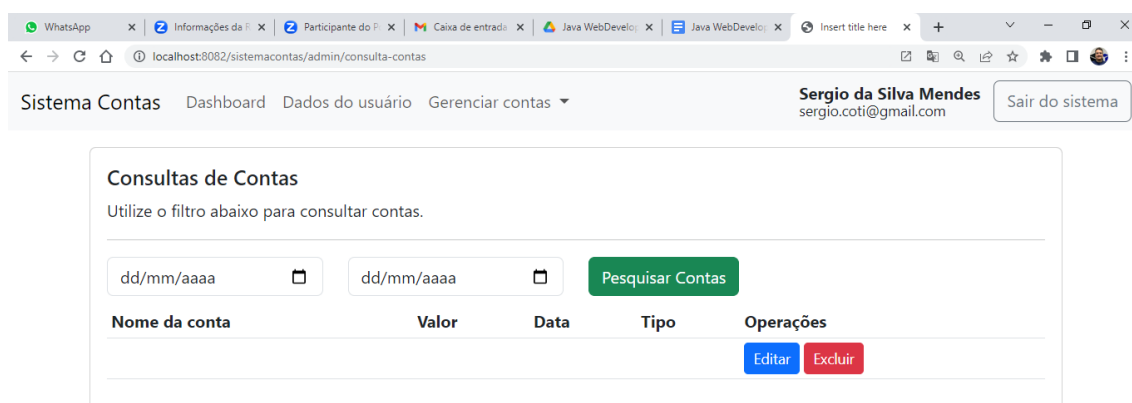
### Testando o cadastro de contas:

<http://localhost:8082/sistemacontas/admin/cadastro-contas>

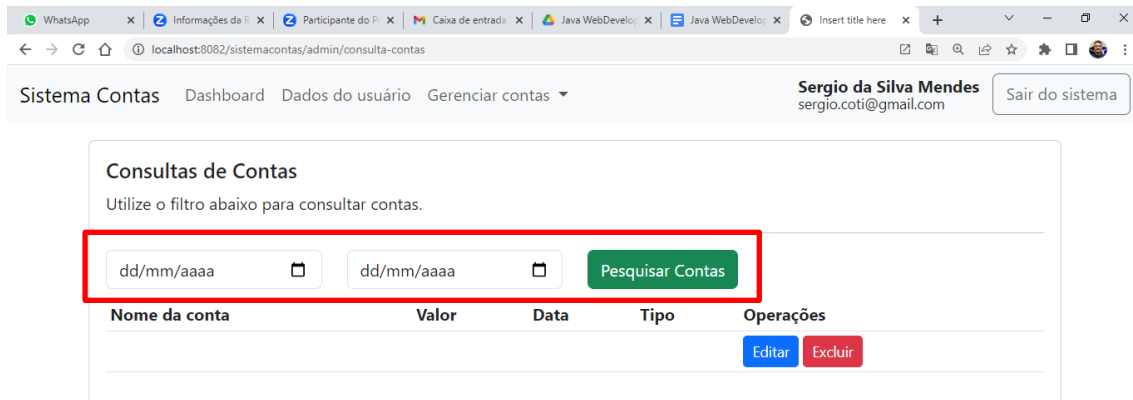
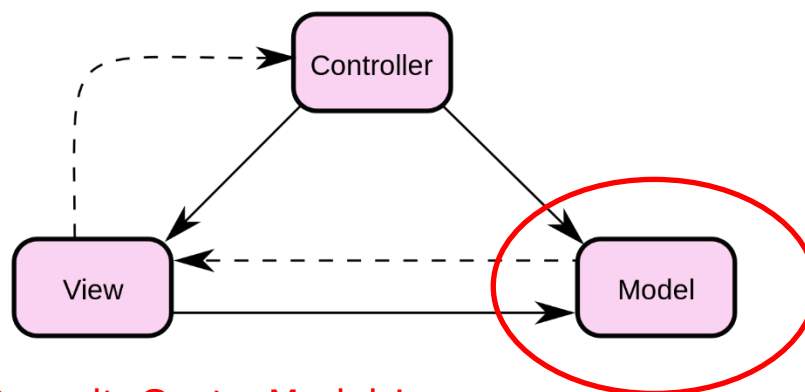


### Página de consulta:

<http://localhost:8082/sistemacontas/admin/consulta-contas>



Segundo o padrão MVC, para capturarmos os campos do formulário da página de consulta de contas, precisamos de uma classe de **modelo de dados (model)**:

/models/**ConsultaContasModel.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 ConsultaContasModel {
```

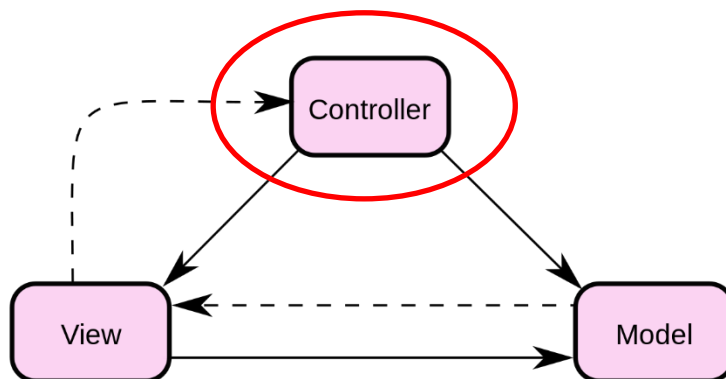
```
    private String dataIni;
```

```
    private String dataFim;
```

```
}
```

Em seguida, precisamos utilizar esta classe para capturar os campos da página de consulta de contas, e para isso vamos utilizar o controlador e a **view**:

/controllers/**ConsultaContasController.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.ConsultaContasModel;
```

```
@Controller
```

```
public class ConsultaContasController {
```

```
    @RequestMapping(value = "/admin/consulta-contas")
```

```
    public ModelAndView consultaContas() {
```

```
        //WEB-INF/views/admin/consulta-contas.jsp
```

```
        ModelAndView modelAndView = new ModelAndView
```

```
            ("admin/consulta-contas");
```

```
        modelAndView.addObject
```

```
            ("model", new ConsultaContasModel());
```

```
        return modelAndView;
```

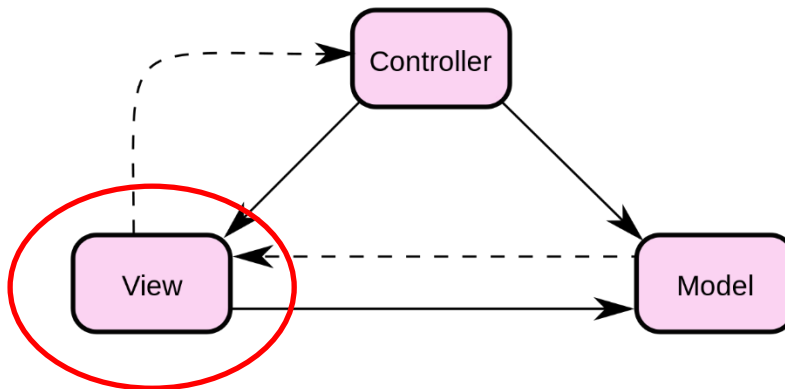
```
    }
```

```
}
```

Por último, vamos acessar a página JSP para então fazermos a captura dos campos do formulário através da model.

Estes campos são:

- Data de início
- Data de fim



/WEB-INF/views/**consulta-contas.jsp**

Aqui, precisamos utilizar uma biblioteca de tags do spring chamada: **/forms** para fazer a captura dos campos do formulário:

```
<%@ 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, textarea.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>Consultas de Contas</h5>
                <p>Utilize o filtro abaixo para consultar contas.</p>
                <hr/>

                <div class="mb-2">
                    <strong>${mensagem}</strong>
                </div>

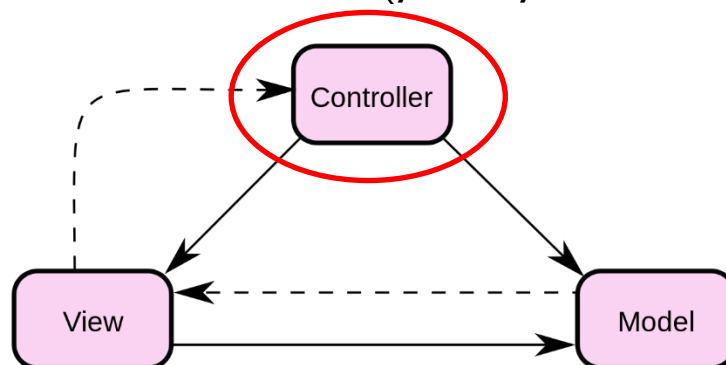
                <form id="form_consulta"
                    action="consultar-contas" method="post">
```

```
<div class="row mb-2">
  <div class="col-md-3">
    <form:input type="date"
      path="model.dataIni" id="dataIni"
      name="dataIni" class="form-
        control"/>
  </div>
  <div class="col-md-3">
    <form:input type="date"
      path="model.dataFim" id="dataFim"
      name="dataFim" class="form-
        control"/>
  </div>
  <div class="col-md-6">
    <input type="submit" class="btn
      btn-success"
      value="Pesquisar Contas"/>
  </div>
</div>

</form>

(...)
```

Agora, precisamos voltar no controlador e criar um método **POST** para capturar o SUBMIT do formulário (**/admin/consultar-contas**)



```
package br.com.cotiinformatica.controller;
```

```
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.models.ConsultaContasModel;
```

```
@Controller
```

```
public class ConsultaContasController {
```

```
    @RequestMapping(value = "/admin/consulta-contas")
```

```
    public ModelAndView consultaContas() {
```

```
        //WEB-INF/views/admin/consulta-contas.jsp
```

```
        ModelAndView modelAndView = new ModelAndView
```

```
            ("admin/consulta-contas");
```

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

```
        return modelAndView;
```

```
    }
```

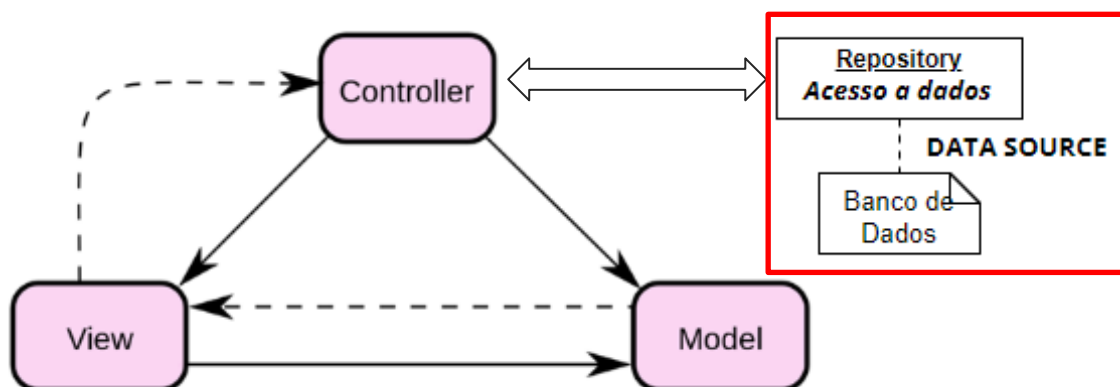
```
@RequestMapping(value = "/admin/consultar-contas",
    method = RequestMethod.POST)
public ModelAndView consultarContas(ConsultaContasModel model) {

    ModelAndView modelAndView = new ModelAndView
        ("admin/consulta-contas");

    //TODO

    modelAndView.addObject("model", model);
    return modelAndView;
}
```

**Agora, vamos consultar as contas no banco de dados para o usuário autenticado e dentro do período de datas selecionado:**



```
package br.com.cotiinformatica.controller;

import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.List;

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.dtos.UsuarioDTO;
import br.com.cotiinformatica.entities.Conta;
import br.com.cotiinformatica.models.ConsultaContasModel;
import br.com.cotiinformatica.repositories.ContaRepository;

@Controller
public class ConsultaContasController {

    @Autowired //inicialização automática (injeção de dependência)
    private ContaRepository contaRepository;
```

```
@RequestMapping(value = "/admin/consulta-contas")
public ModelAndView consultaContas() {

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

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

    return modelAndView;
}

@RequestMapping(value = "/admin/consultar-contas",
    method = RequestMethod.POST)
public ModelAndView consultarContas(ConsultaContasModel model,
    HttpServletRequest request) {

    ModelAndView modelAndView = new ModelAndView
        ("admin/consulta-contas");

    try {

        //capturar os dados do usuário gravado em sessão
        UsuarioDTO usuarioDto = (UsuarioDTO) request.getSession()
            .getAttribute("usuario");

        //capturar as datas selecionadas no formulário
        Date dataIni = new SimpleDateFormat
            ("yyyy-MM-dd").parse(model.getDataIni());

        Date dataFim = new SimpleDateFormat
            ("yyyy-MM-dd").parse(model.getDataFim());

        //consultar as contas do usuário no banco de dados
        List<Conta> contas = contaRepository.findByUsuarioAndData
            (usuarioDto.getIdUsuario(), dataIni, dataFim);

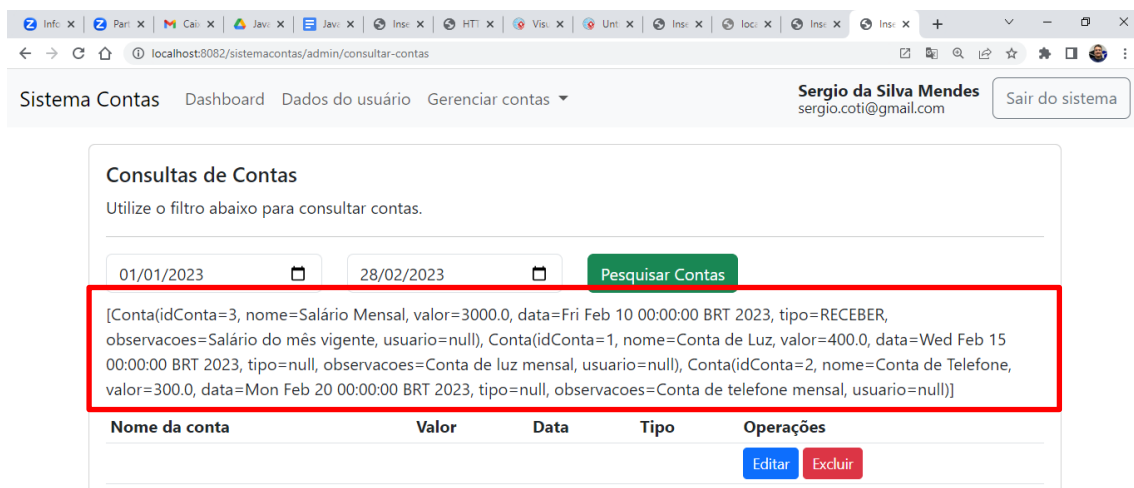
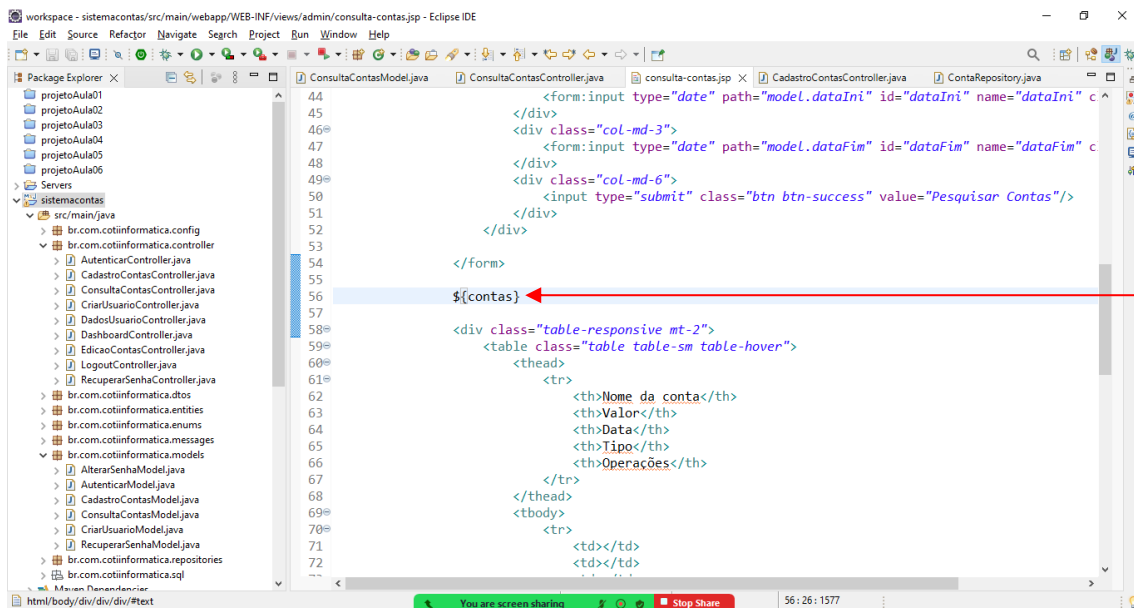
        //enviando a lista de contas para a página exibir
        modelAndView.addObject("contas", contas);
    }
    catch(Exception e) {
        modelAndView.addObject
            ("mensagem", "Falha ao consultar contas: " +
            e.getMessage());
    }

    modelAndView.addObject("model", model);
    return modelAndView;
}

}
```



Exibindo os dados na página:

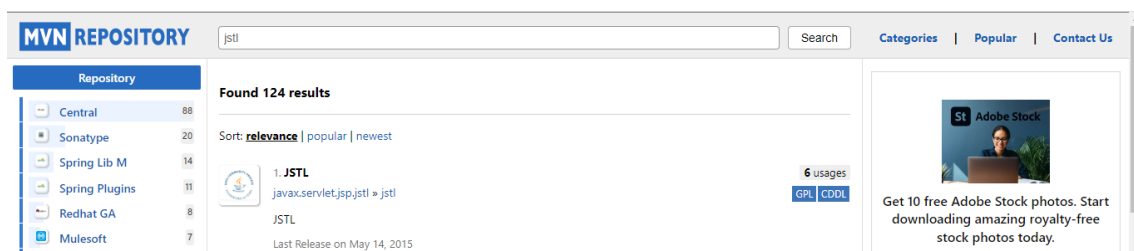


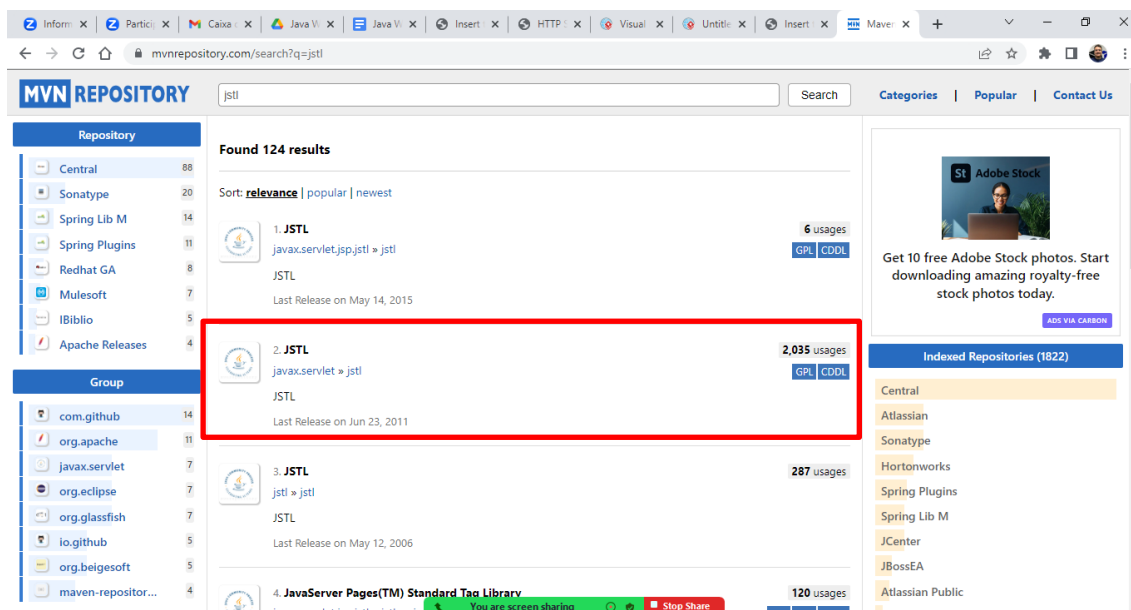
## JSTL – Java Standart Tag Libraries

Conjunto de bibliotecas de tags para páginas JSP que possuem uma série de funções e recursos capazes de manipular dados obtidos de um controlador de maneira e exibi-los nas páginas JSP.

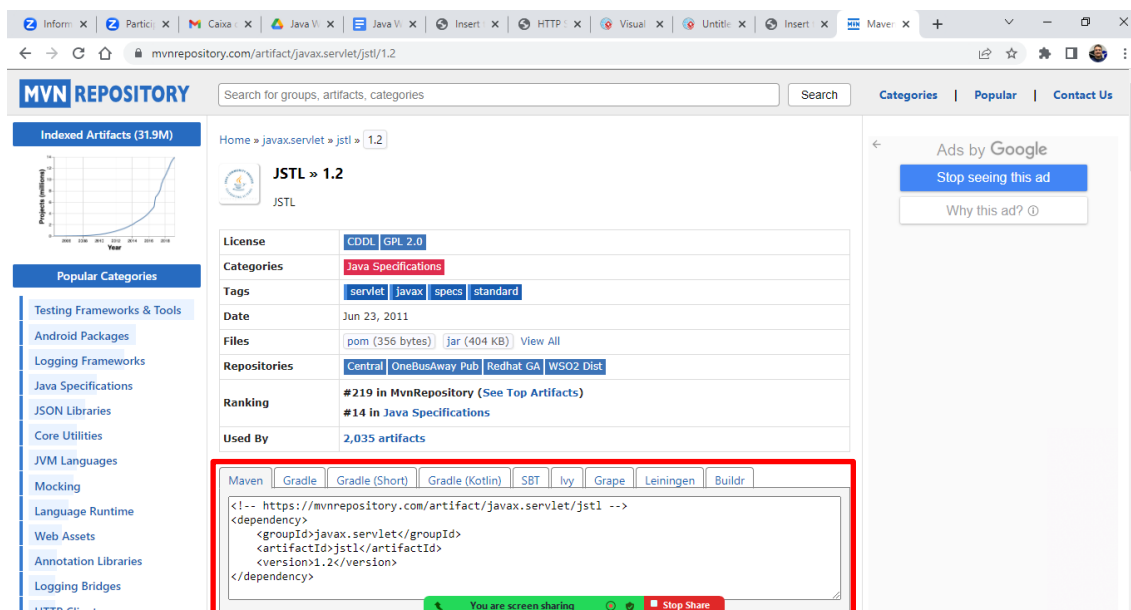
Instalando:

<https://mvnrepository.com/search?q=jstl>



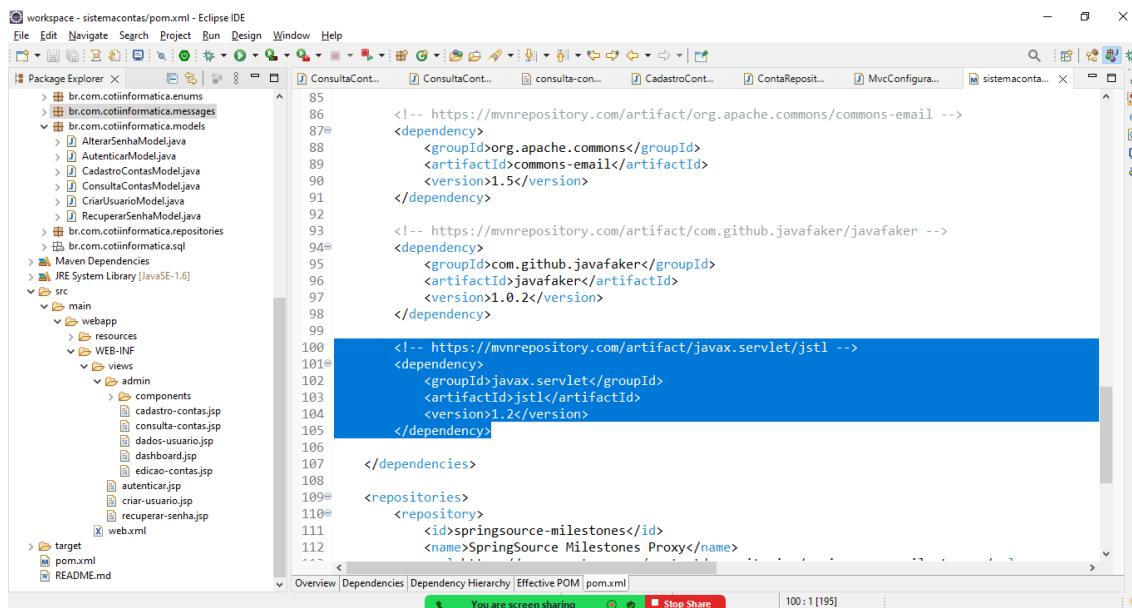


<https://mvnrepository.com/artifact/javax.servlet/jstl/1.2>



**Cole o XML abaixo no arquivo /pom.xml do projeto:**

```
<!-- https://mvnrepository.com/artifact/javax.servlet/jstl -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
</dependency>
```



Voltando para a página de consulta:

/WEB-INF/views/**consulta-contas.jsp**

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
```

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

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

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
```

```
<!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, textarea.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>Consultas de Contas</h5>
            <p>Utilize o filtro abaixo para consultar contas.</p>
            <hr/>

            <div class="mb-2">
                <strong>${mensagem}</strong>
            </div>

            <form id="form_consulta" action="consultar-contas"
                  method="post">

                <div class="row mb-2">
                    <div class="col-md-3">
                        <form:input type="date"
                                   path="model.dataIni" id="dataIni"
                                   name="dataIni" class="form-
                                   control"/>
                    </div>
                    <div class="col-md-3">
                        <form:input type="date"
                                   path="model.dataFim" id="dataFim"
                                   name="dataFim" class="form-
                                   control"/>
                    </div>
                    <div class="col-md-6">
                        <input type="submit" class="btn
                        btn-success"
                        value="Pesquisar Contas"/>
                    </div>
                </div>
            </div>

            </form>

            <c:if test="${contas.size() > 0}">

                <div class="table-responsive mt-2">
                    <table class="table table-sm table-hover">
                        <thead>
                            <tr>
                                <th>Nome da conta</th>
                                <th>Valor</th>
                                <th>Data</th>
                                <th>Tipo</th>
                                <th>Operações</th>
                            </tr>
                        </thead>
                        <tbody>

                            <c:forEach items="${contas}"
                                    var="conta">
                                <tr>

                                    <td>${conta.nome}</td>
                                    <td><fmt:formatNumber
                                    value="${conta.valor}"
                                    type="currency"/></td>
                                    <td><fmt:formatDate
```

```

        value="${conta.data}"
        pattern="EE dd/MM/yyyy"
    /></td>

    <td>${conta.tipo}</td>
    <td>
        <a
            href="/sistemacontas/
            admin/edicao-contas"
            class="btn btn-
            primary btn-sm">
                Editar
            </a>
            <a
            href="/sistemacontas/
            admin/exclusao-
            contas" class="btn
            btn-danger btn-sm">
                Excluir
            </a>
        </td>
    </tr>
</c:forEach>

</tbody>
</table>
</div>

</c:if>

</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>

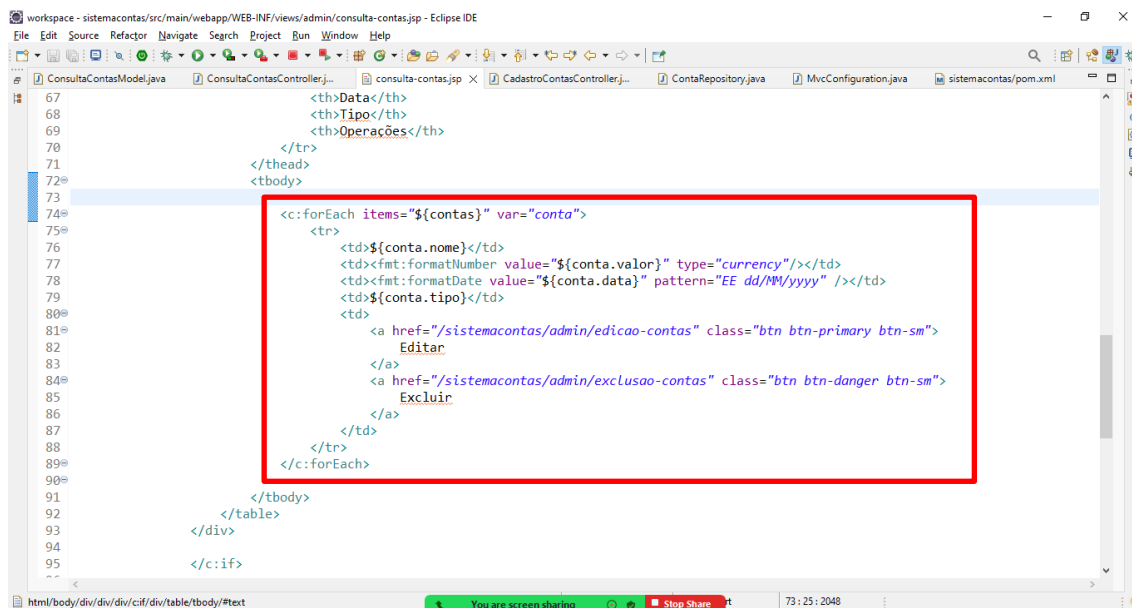
<!-- CDN dos arquivos da biblioteca JQuery Validation -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-
validate/1.19.5/jquery.validate.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-
validate/1.19.5/additional-methods.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-
validate/1.19.5/localization/messages_pt_BR.min.js"></script>

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

    $("#form_consulta").validate({
        rules: {
            'dataIni' : { required: true },
            'dataFim' : { required: true }
        }
    });

})
</script>
</body>
</html>

```



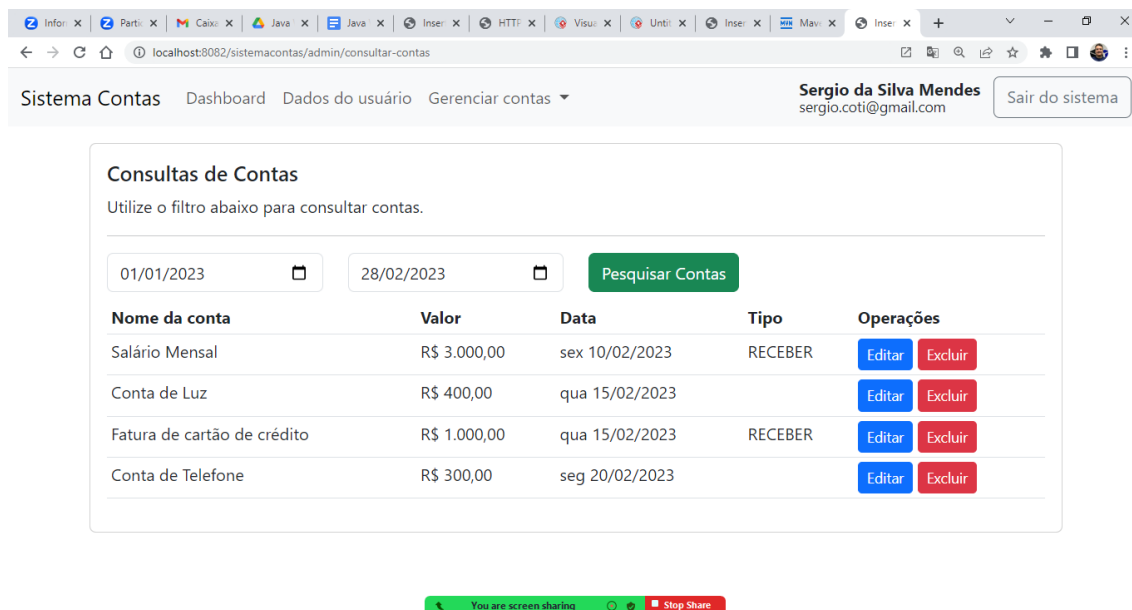
```

67 <th>Data</th>
68 <th>Tipo</th>
69 <th>Operações</th>
70 </tr>
71 </thead>
72 <tbody>
73
74 <:forEach items="${contas}" var="conta">
75 <tr>
76 <td>${conta.nome}</td>
77 <td><fmt:formatNumber value="${conta.valor}" type="currency"/></td>
78 <td><fmt:formatDate value="${conta.data}" pattern="EE dd/MM/yyyy" /></td>
79 <td>${conta.tipo}</td>
80 <td>
81 <a href="/sistemacontas/admin/edicao-contas" class="btn btn-primary btn-sm">
82 Editar
83 </a>
84 <a href="/sistemacontas/admin/exclusao-contas" class="btn btn-danger btn-sm">
85 Excluir
86 </a>
87 </td>
88 </tr>
89 </c:forEach>
90
91 </tbody>
92 </table>
93 </div>
94 </c:if>
95

```

### Testando:

<http://localhost:8082/sistemacontas/admin/consultar-contas>



Sistema Contas Dashboard Dados do usuário Gerenciar contas

Sergio da Silva Mendes  
sergio.coti@gmail.com Sair do sistema

#### Consultas de Contas

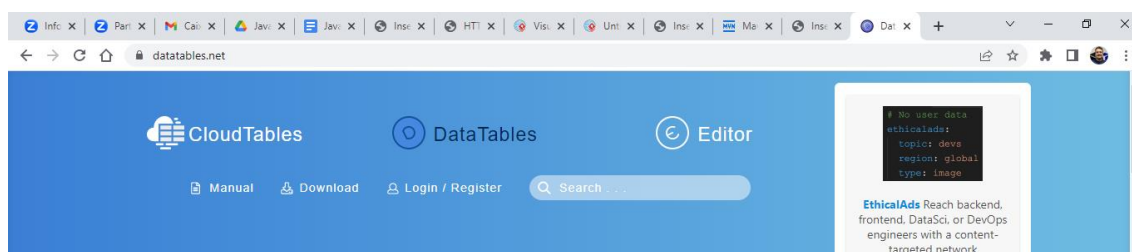
Utilize o filtro abaixo para consultar contas.

01/01/2023 28/02/2023 Pesquisar Contas

Nome da conta	Valor	Data	Tipo	Operações
Salário Mensal	R\$ 3.000,00	sex 10/02/2023	RECEBER	<button>Editar</button> <button>Excluir</button>
Conta de Luz	R\$ 400,00	qua 15/02/2023		<button>Editar</button> <button>Excluir</button>
Fatura de cartão de crédito	R\$ 1.000,00	qua 15/02/2023	RECEBER	<button>Editar</button> <button>Excluir</button>
Conta de Telefone	R\$ 300,00	seg 20/02/2023		<button>Editar</button> <button>Excluir</button>

## JQuery DataTables

Biblioteca para formatação de tabelas em páginas HTML já adicionando funcionalidades de paginação, filtros, ordenação etc.

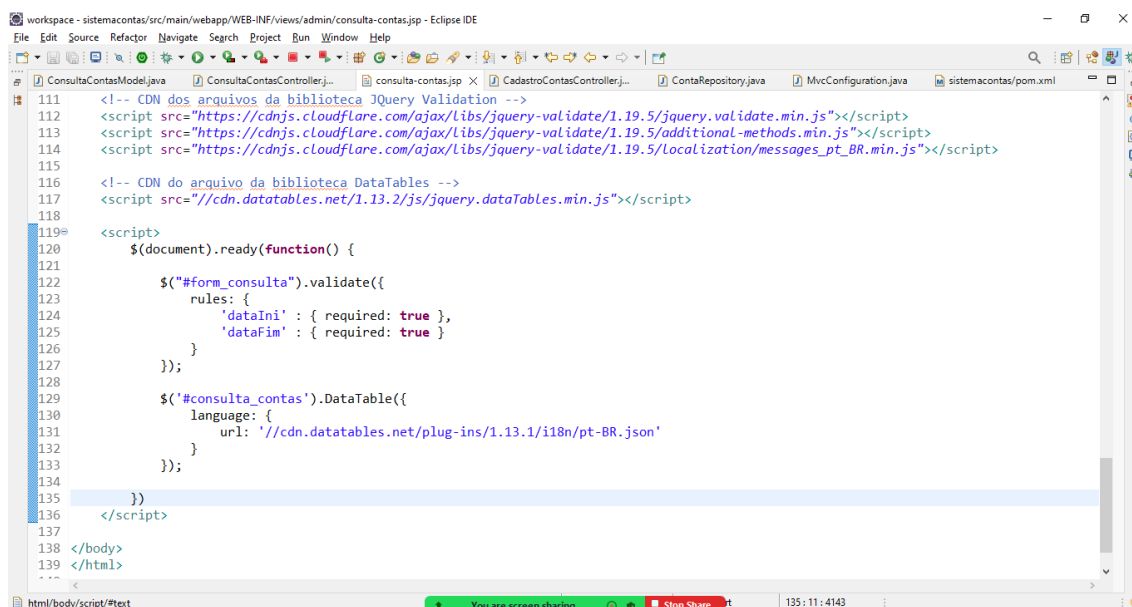
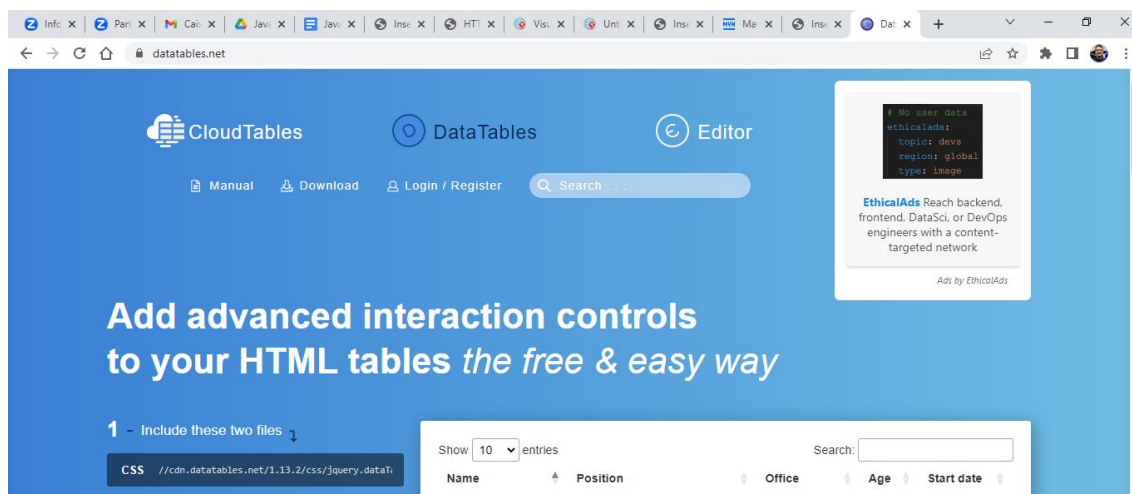


CloudTables DataTables Editor

Manual Download Login / Register Search

! No user data  
ethicalads: topier devops region: global type: inpage

EthicalAds Reach backend, frontend, DataSci, or DevOps engineers with a content-targeted network



```

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

        $('#form_consulta').validate({
            rules: {
                'dataIni' : { required: true },
                'dataFim' : { required: true }
            }
        });

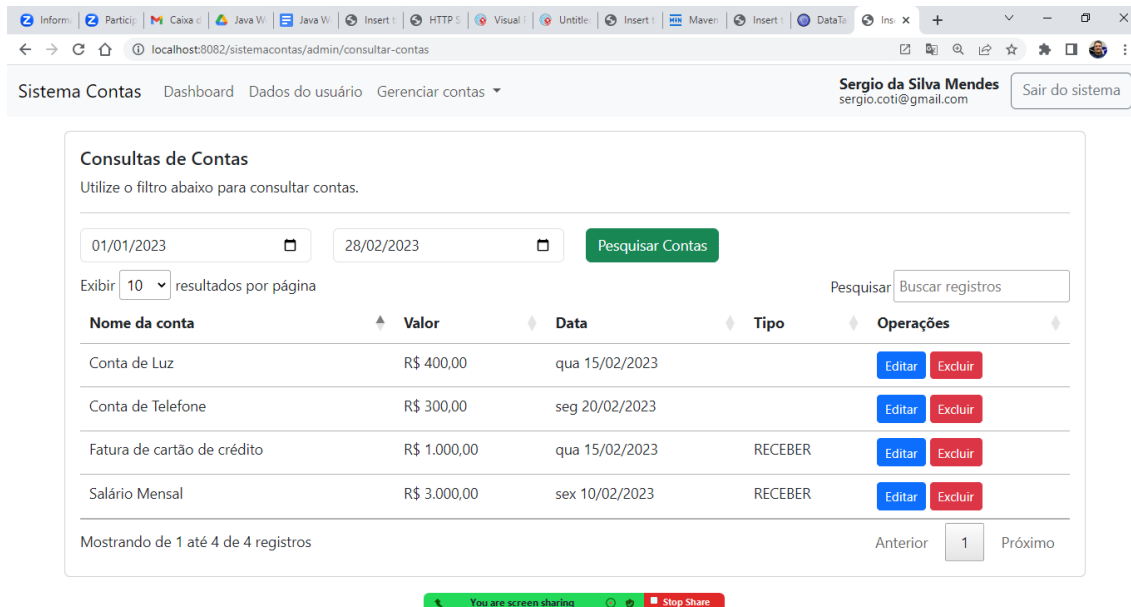
        $('#consulta_contas').DataTable({
            language: {
                url: '//cdn.datatables.net
                    /plug-ins/1.13.1/i18n/pt-BR.json'
            }
        });

    })
</script>

```

### Resultado:

<http://localhost:8082/sistemacontas/admin/consultar-contas>



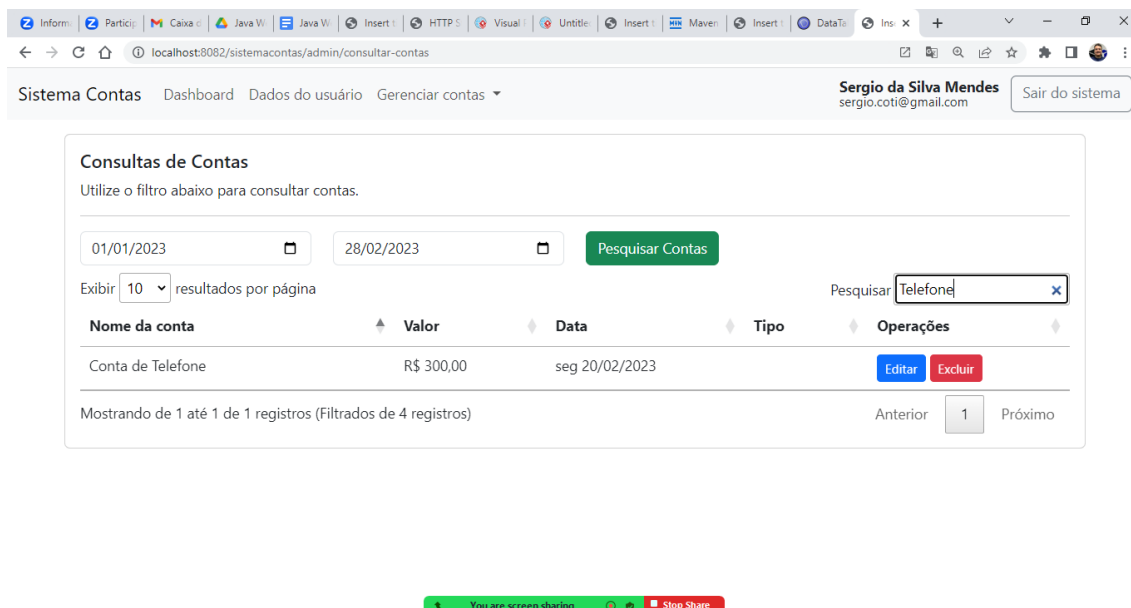
**Consultas de Contas**  
Utilize o filtro abaixo para consultar contas.

01/01/2023 28/02/2023 **Pesquisar Contas**

Exibir 10 resultados por página Pesquisar Buscar registros

Nome da conta	Valor	Data	Tipo	Operações
Conta de Luz	R\$ 400,00	qua 15/02/2023		<a href="#">Editar</a> <a href="#">Excluir</a>
Conta de Telefone	R\$ 300,00	seg 20/02/2023		<a href="#">Editar</a> <a href="#">Excluir</a>
Fatura de cartão de crédito	R\$ 1.000,00	qua 15/02/2023	RECEBER	<a href="#">Editar</a> <a href="#">Excluir</a>
Salário Mensal	R\$ 3.000,00	sex 10/02/2023	RECEBER	<a href="#">Editar</a> <a href="#">Excluir</a>

Mostrando de 1 até 4 de 4 registros Anterior 1 Próximo



**Consultas de Contas**  
Utilize o filtro abaixo para consultar contas.

01/01/2023 28/02/2023 **Pesquisar Contas**

Exibir 10 resultados por página Pesquisar Telefone

Nome da conta	Valor	Data	Tipo	Operações
Conta de Telefone	R\$ 300,00	seg 20/02/2023		<a href="#">Editar</a> <a href="#">Excluir</a>

Mostrando de 1 até 1 de 1 registros (Filtrados de 4 registros) Anterior 1 Próximo

### Excluindo contas:

Precisamos fazer uma requisição para o controlador, ao clicar no link de exclusão, que envie o ID da conta selecionada:

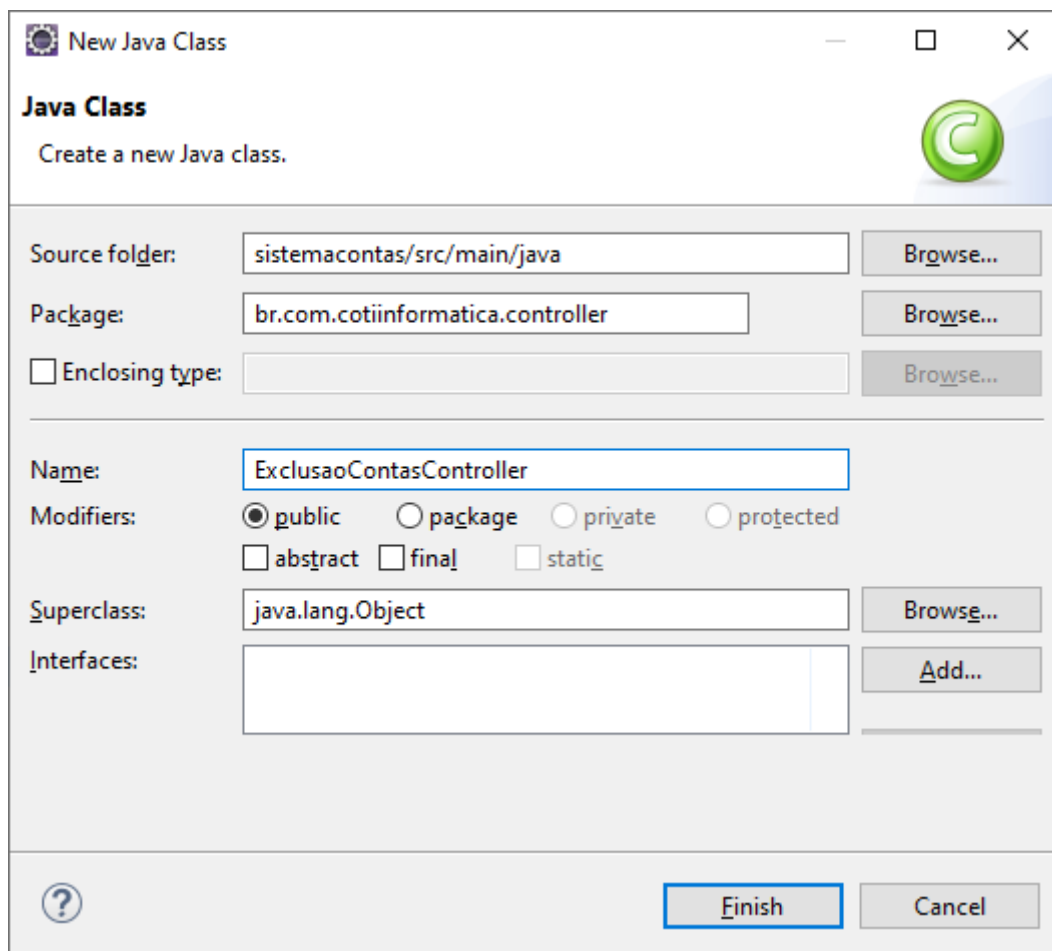
```
<a href="/sistemacontas/admin/exclusao-contas?id=${conta.idConta}"
  onclick="return confirm('Deseja realmente excluir a conta?
    \n ${conta.nome}');"
  class="btn btn-danger btn-sm">
  Excluir
</a>
```



**Criando um controlador capaz de capturar a requisição de exclusão de contas:**

`/sistemacontas/admin/exclusao-contas?id=${conta.idConta}`

`/controllers/ExclusaoContasController.java`



```
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.servlet.ModelAndView;
```

```
import br.com.cotiinformatica.dtos.UsuarioDTO;
```

```
import br.com.cotiinformatica.entities.Conta;
```

```
import br.com.cotiinformatica.models.ConsultaContasModel;
```

```
import br.com.cotiinformatica.repositories.ContaRepository;
```

```
@Controller
public class ExclusaoContasController {

    @Autowired
    private ContaRepository contaRepository;

    @RequestMapping(value = "/admin/exclusao-contas")
    public ModelAndView exclusaoContas
        (Integer id, HttpServletRequest request) {

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

        try {

            //capturando o usuário autenticado na sessão
            UsuarioDTO usuarioDto = (UsuarioDTO)
                request.getSession().getAttribute("usuario");

            //buscando o registro da conta no banco de dados
            Conta conta = contaRepository.findById
                (id, usuarioDto.getIdUsuario());

            //verificando se a conta foi encontrada
            if(conta != null) {

                //excluindo a conta no banco de dados
                contaRepository.delete(conta);

                modelAndView.addObject("mensagem", "Conta '"
                    + conta.getNome()
                    + "' excluída com sucesso.");
            }
        }
        catch(Exception e) {

            modelAndView.addObject("mensagem",
                "Falha ao excluir conta: " + e.getMessage());
        }

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

        return modelAndView;
    }
}
```

### Resultado:

localhost:8082/sistemacontas/admin/consultar-contas

Sistema Contas Dashboard Dados do usuário G

localhost:8082 diz  
Deseja realmente excluir a conta?  
Conta de Luz

Sergio da Silva Mendes  
sergio.coti@gmail.com

Sair do sistema

**Consultas de Contas**  
Utilize o filtro abaixo para consultar contas.

01/01/2023 28/02/2023 **Pesquisar Contas**

Exibir 10 resultados por página Pesquisar Buscar registros

Nome da conta	Valor	Data	Tipo	Operações
Conta de Luz	R\$ 400,00	qua 15/02/2023		<a href="#">Editar</a> <a href="#">Excluir</a>
Conta de Telefone	R\$ 300,00	seg 20/02/2023		<a href="#">Editar</a> <a href="#">Excluir</a>
Fatura de cartão de crédito	R\$ 1.000,00	qua 15/02/2023	RECEBER	<a href="#">Editar</a> <a href="#">Excluir</a>
Salário Mensal	R\$ 3.000,00	sex 10/02/2023	RECEBER	<a href="#">Editar</a> <a href="#">Excluir</a>

Mostrando de 1 até 4 de 4 registros

Anterior 1 Próximo

localhost:8082/sistemacontas/admin/exclusao-contas?id=1

You are screen sharing

Stop Share

localhost:8082/sistemacontas/admin/exclusao-contas?id=1

Sistema Contas Dashboard Dados do usuário Gerenciar contas

Sergio da Silva Mendes  
sergio.coti@gmail.com

Sair do sistema

**Consultas de Contas**  
Utilize o filtro abaixo para consultar contas.

Conta 'Conta de Luz' excluída com sucesso.

dd/mm/aaaa dd/mm/aaaa **Pesquisar Contas**

localhost:8082/sistemacontas/admin/consultar-contas

Sistema Contas Dashboard Dados do usuário Gerenciar contas

Sergio da Silva Mendes  
sergio.coti@gmail.com

Sair do sistema

**Consultas de Contas**  
Utilize o filtro abaixo para consultar contas.

01/01/2023 28/02/2023 **Pesquisar Contas**

Exibir 10 resultados por página Pesquisar Buscar registros

Nome da conta	Valor	Data	Tipo	Operações
Fatura de cartão de crédito	R\$ 1.000,00	qua 15/02/2023	RECEBER	<a href="#">Editar</a> <a href="#">Excluir</a>
Salário Mensal	R\$ 3.000,00	sex 10/02/2023	RECEBER	<a href="#">Editar</a> <a href="#">Excluir</a>

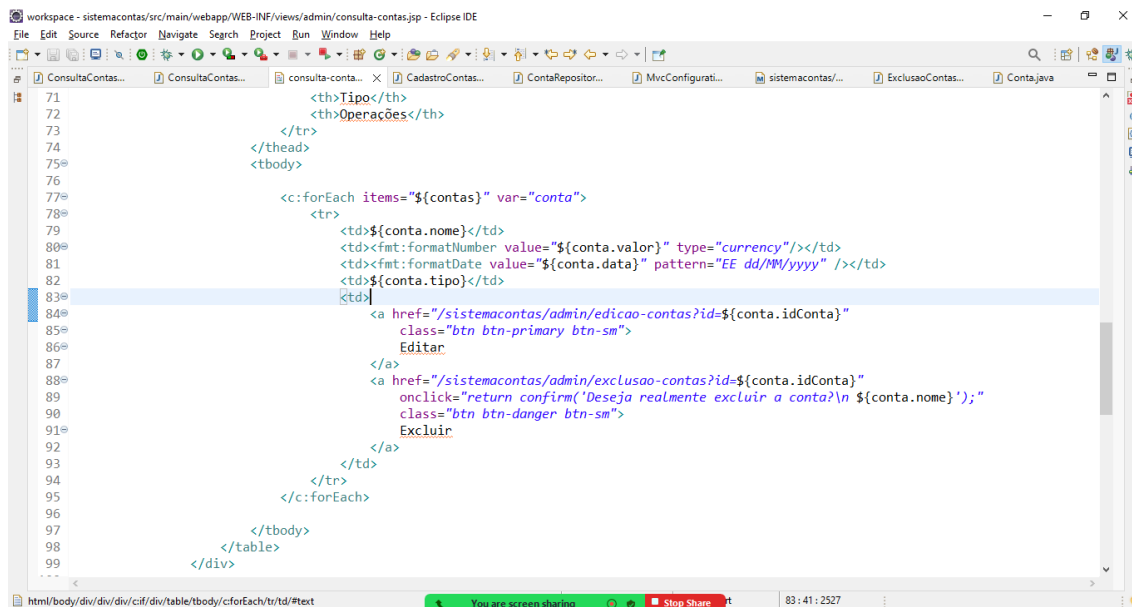
Mostrando de 1 até 2 de 2 registros

Anterior 1 Próximo

You are screen sharing

Stop Share

### Criando a edição de contas:



```

71      <th>Tipo</th>
72      <th>Operações</th>
73    </tr>
74  </thead>
75  <tbody>
76
77    <c:forEach items="${contas}" var="conta">
78      <tr>
79        <td>${conta.nome}</td>
80        <td><fmt:formatNumber value="${conta.valor}" type="currency"/></td>
81        <td><fmt:formatDate value="${conta.data}" pattern="EE dd/MM/yyyy" /></td>
82        <td>${conta.tipo}</td>
83        <td>
84          <a href="/sistemacontas/admin/edicao-contas?id=${conta.idConta}"
85            class="btn btn-primary btn-sm">
86            Editar
87          </a>
88          <a href="/sistemacontas/admin/exclusao-contas?id=${conta.idConta}"
89            onclick="return confirm('Deseja realmente excluir a conta?n ${conta.nome}');"
90            class="btn btn-danger btn-sm">
91            Excluir
92          </a>
93        </td>
94      </tr>
95    </c:forEach>
96  </tbody>
97</table>
98</div>
99

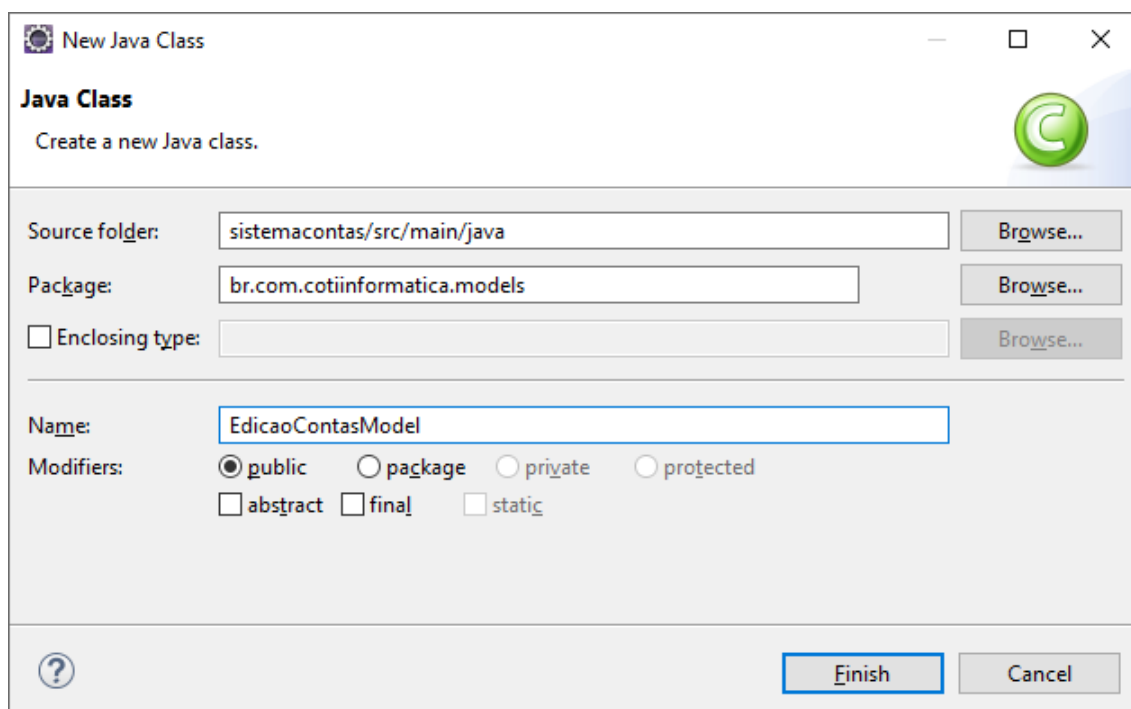
```

```

<a href="/sistemacontas/admin/edicao-contas?id=${conta.idConta}"
  class="btn btn-primary btn-sm">
  Editar
</a>

```

Criando uma classe de modelo de dados para a página de edição:  
/models/EdicaoContasModel.java



**New Java Class**

Java Class  
Create a new Java class.

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

Package: br.com.cotiinformatica.models Browse...

☐ Enclosing type: Browse...

Name: EdicaoContasModel

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

Finish Cancel

```
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 EdicaoContasModel {
```

```
    private Integer idConta;
    private String nome;
    private String valor;
    private String data;
    private String tipo;
    private String observacoes;
```

```
}
```

Em seguida, precisamos fazer com o que o controlador, ao abrir a página de edição já exiba um formulário preenchido com os dados da conta selecionada:

/controller/**EdicaoContasController.java**

```
package br.com.cotiinformatica.controller;
```

```
import java.text.SimpleDateFormat;
```

```
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.servlet.ModelAndView;
```

```
import br.com.cotiinformatica.dtos.UsuarioDTO;
```

```
import br.com.cotiinformatica.entities.Conta;
```

```
import br.com.cotiinformatica.enums.TipoConta;
```

```
import br.com.cotiinformatica.models.EdicaoContasModel;
```

```
import br.com.cotiinformatica.repositories.ContaRepository;
```

```
@Controller
```

```
public class EdicaoContasController {
```

```
    @Autowired
```

```
    private ContaRepository contaRepository;
```

```
@RequestMapping(value = "/admin/edicao-contas")
public ModelAndView edicaoContas
(Integer id, HttpServletRequest request) {

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

    try {

        //capturar o usuário autenticado
        UsuarioDTO usuarioDto = (UsuarioDTO) request.getSession()
            .getAttribute("usuario");

        //consultar a conta no banco de dados
        Conta conta = contaRepository.findById
            (id, usuarioDto.getIdUsuario());

        //verificando se a conta foi encontrada
        if(conta != null) {

            EdicaoContasModel model = new EdicaoContasModel();

            model.setIdConta(conta.getIdConta());
            model.setNome(conta.getNome());
            model.setValor(conta.getValor().toString());
            model.setData(new SimpleDateFormat
                ("yyyy-MM-dd").format(conta.getData()));
            model.setTipo(conta.getTipo() == TipoConta.RECEBER
                ? "1" : conta.getTipo() == TipoConta.PAGAR
                ? "2" : null);
            model.setObservacoes(conta.getObservacoes());

            modelAndView.addObject("model", model);
        }
    }
    catch(Exception e) {
        modelAndView.addObject
            ("mensagem", "Falha ao obter conta: " + e.getMessage());
    }

    return modelAndView;
}
```

Desenhando a página de edição de contas:

/WEB-INF/views/admin/**edicao-contas.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, textarea.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>Edição de Contas</h5>
                <p>Utilize o formulário para alterar
                    os dados da conta.</p>
                <hr/>

                <div class="mb-2">
                    <strong>${mensagem}</strong>
                </div>

                <form id="form_edicao"
                    action="atualizar-conta" method="post">

                    <!-- campo oculto -->
                    <form:hidden path="model.idConta"/>

                    <div class="row mb-2">
                        <div class="col-md-6">
                            <label>Nome da conta:</label>
                            <form:input path="model.nome"
                                type="text" name="nome"
                                id="nome" class="form-control"/>
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>
```

```

        </div>
        <div class="col-md-3">
            <label>Valor da conta:</label>
            <form:input path="model.valor"
                type="text" name="valor"
                id="valor" class="form-
                control"/>
        </div>
        <div class="col-md-3">
            <label>Data da conta:</label>
            <form:input path="model.data"
                type="date" name="data"
                id="data" class="form-control"/>
        </div>
    </div>
    <div class="row mb-2">
        <div class="col-md-3">
            <label>Selecione o tipo da
                conta:</label>
            <div>
                <form:radio button
                    path="model.tipo"
                    name="tipo" value="1"
                    checked="checked"/> Conta
                    a receber
            </div>
            <div>
                <form:radio button
                    path="model.tipo"
                    name="tipo" value="2"/>
                    Conta a pagar
            </div>
        </div>
        <div class="col-md-9">
            <label>Observações da
                conta:</label>
            <form:textarea
                path="model.observacoes"
                name="observacoes"
                id="observacoes" class="form-
                control"></form:textarea>
        </div>
    </div>
    <div class="row mb-2">
        <div class="col-md-12">
            <input type="submit" class="btn
                btn-primary"
                value="Salvar
                Alterações"/>
        </div>
    </div>
</form>
</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>

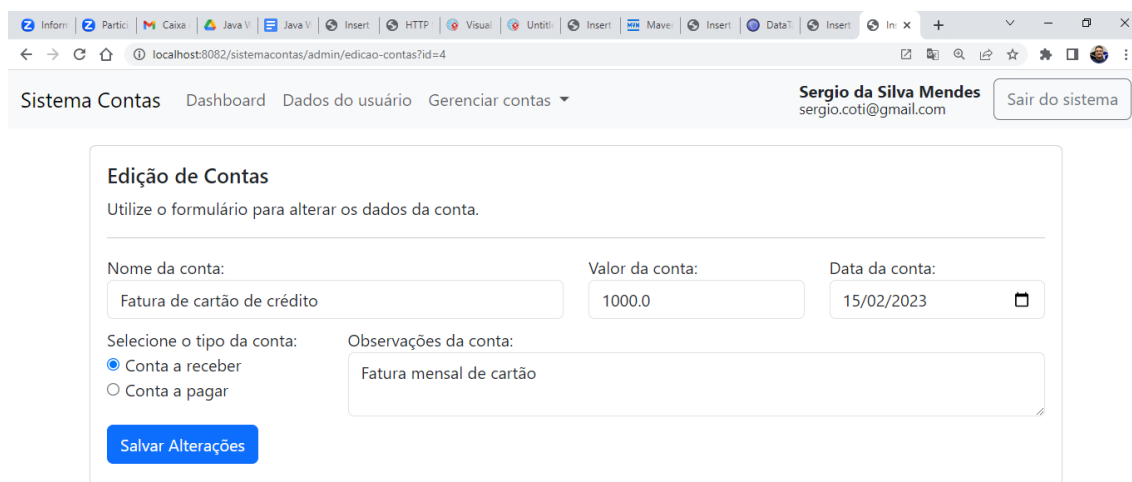
<!-- CDN do arquivo javascript do JQuery -->
<script src="https://code.jquery.com/jquery-3.6.3.min.js"></script>

<!-- 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_edicao").validate({
        rules: {
            'nome' : { required: true, minlength : 6,
                        maxlength: 150 },
            'valor' : { required: true },
            'data' : { required: true },
            'tipo' : { required: true },
            'observacoes' : { required: true,
                              minlength : 6, maxlength: 500 },
        }
    });
});
</script>
</body>
</html>

```

<http://localhost:8082/sistemacontas/admin/edicao-contas?id=4>



**No controlador, precisamos criar o método POST para fazer a atualização da conta no banco de dados:**

```
package br.com.cotiinformatica.controller;

import java.text.SimpleDateFormat;

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.dtos.UsuarioDTO;
import br.com.cotiinformatica.entities.Conta;
import br.com.cotiinformatica.enums.TipoConta;
import br.com.cotiinformatica.models.EdicaoContasModel;
import br.com.cotiinformatica.repositories.ContaRepository;

@Controller
public class EdicaoContasController {

    @Autowired
    private ContaRepository contaRepository;

    @RequestMapping(value = "/admin/edicao-contas")
    public ModelAndView edicaoContas
        (Integer id, HttpServletRequest request) {

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

        try {

            // capturar o usuário autenticado
            UsuarioDTO usuarioDto = (UsuarioDTO)
                request.getSession().getAttribute("usuario");

            // consultar a conta no banco de dados
            Conta conta = contaRepository.findById
                (id, usuarioDto.getIdUsuario());

            // verificando se a conta foi encontrada
            if (conta != null) {

                EdicaoContasModel model
                    = new EdicaoContasModel();
```

```

        model.setIdConta(conta.getIdConta());
        model.setNome(conta.getNome());
        model.setValor(conta.getValor().toString());
        model.setData(new SimpleDateFormat
            ("yyyy-MM-dd").format(conta.getData()));
        model.setTipo(
            conta.getTipo() ==
            TipoConta.RECEBER ? "1" :
            conta.getTipo() == TipoConta.PAGAR
            ? "2" : null);
        model.setObservacoes(conta.getObservacoes());

        modelAndView.addObject("model", model);
    }
} catch (Exception e) {
    modelAndView.addObject
        ("mensagem", "Falha ao obter conta: "
            + e.getMessage());
}

return modelAndView;
}

@RequestMapping(value = "/admin/atualizar-conta",
    method = RequestMethod.POST)
public ModelAndView atualizarConta(EdicaoContasModel model,
    HttpServletRequest request) {

    ModelAndView modelAndView
        = new ModelAndView("admin/edicao-contas");

    try {

        Conta conta = new Conta();

        conta.setIdConta(model.getIdConta());
        conta.setNome(model.getNome());
        conta.setValor
            (Double.parseDouble(model.getValor()));
        conta.setData(new SimpleDateFormat
            ("yyyy-MM-dd").parse(model.getData()));
        conta.setTipo(model.getTipo().equals("1") ?
            TipoConta.RECEBER :
            model.getTipo().equals("2") ?
            TipoConta.PAGAR : null);
        conta.setObservacoes(model.getObservacoes());

        contaRepository.update(conta);

        modelAndView.addObject("mensagem",
            "Conta atualizada com sucesso.");
    }
}

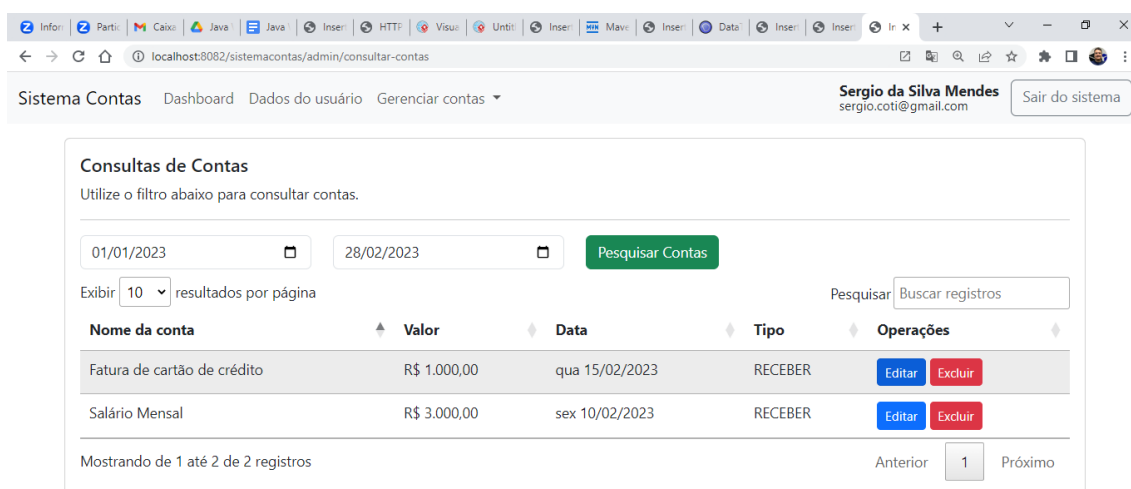
```

```

    } catch (Exception e) {
        modelAndView.addObject(
            "mensagem", "Falha ao atualizar conta: "
            + e.getMessage());
    }

    modelAndView.addObject("model", model);
    return modelAndView;
}
}

```



**Consultas de Contas**  
Utilize o filtro abaixo para consultar contas.

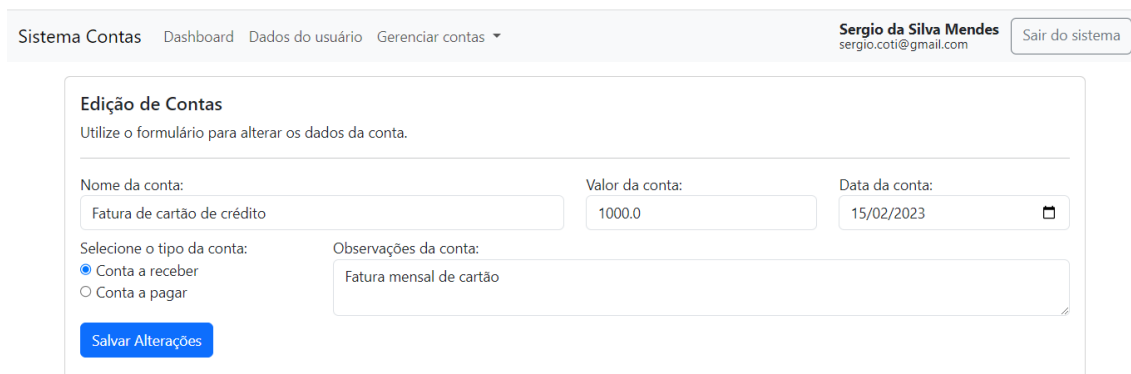
01/01/2023 28/02/2023 **Pesquisar Contas**

Exibir 10 resultados por página

Nome da conta	Valor	Data	Tipo	Operações
Fatura de cartão de crédito	R\$ 1.000,00	qua 15/02/2023	RECEBER	<a href="#">Editar</a> <a href="#">Excluir</a>
Salário Mensal	R\$ 3.000,00	sex 10/02/2023	RECEBER	<a href="#">Editar</a> <a href="#">Excluir</a>

Mostrando de 1 até 2 de 2 registros

Anterior 1 Próximo



**Edição de Contas**  
Utilize o formulário para alterar os dados da conta.

Nome da conta: Fatura de cartão de crédito

Valor da conta: 1000.0

Data da conta: 15/02/2023

Selecione o tipo da conta:  
☒ Conta a receber  
☐ Conta a pagar

Observações da conta:  
 Fatura mensal de cartão

**Salvar Alterações**



**Edição de Contas**  
Utilize o formulário para alterar os dados da conta.

**Conta atualizada com sucesso.**

Nome da conta: Fatura de cartão de crédito

Valor da conta: 1400

Data da conta: 15/02/2023

Selecione o tipo da conta:  
☐ Conta a receber  
☒ Conta a pagar

Observações da conta:  
 Fatura mensal de cartão

**Salvar Alterações**