Desenvolvimento de Aplicações WEB

Frameworks

Profa. Joyce Miranda

Materiais de Referência: http://www.caelum.com.br/apostila-java-web

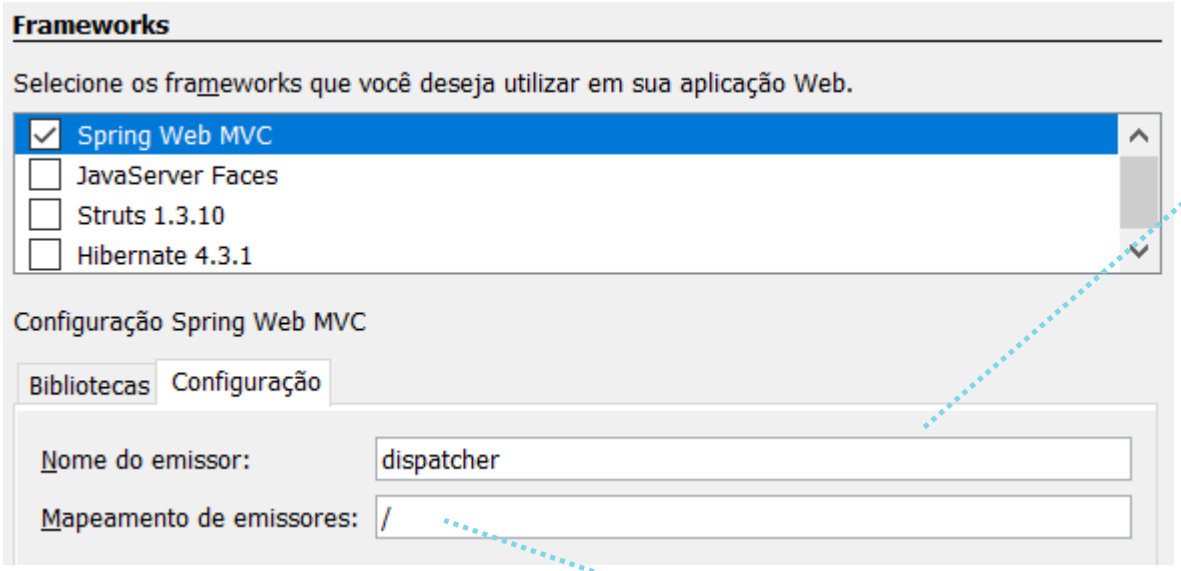
Frameworks

- Frameworks WEB
  - Evolução

  - Tornar o desenvolvimento web mais simples
    - Abstrair questões sobre arquitetura da aplicação
    - Manter foco na lógica de negócios

Frameworks

- Criando o primeiro projeto SPRING
  - Netbeans – JAVA Web – SPRING Web MVC – Configuração

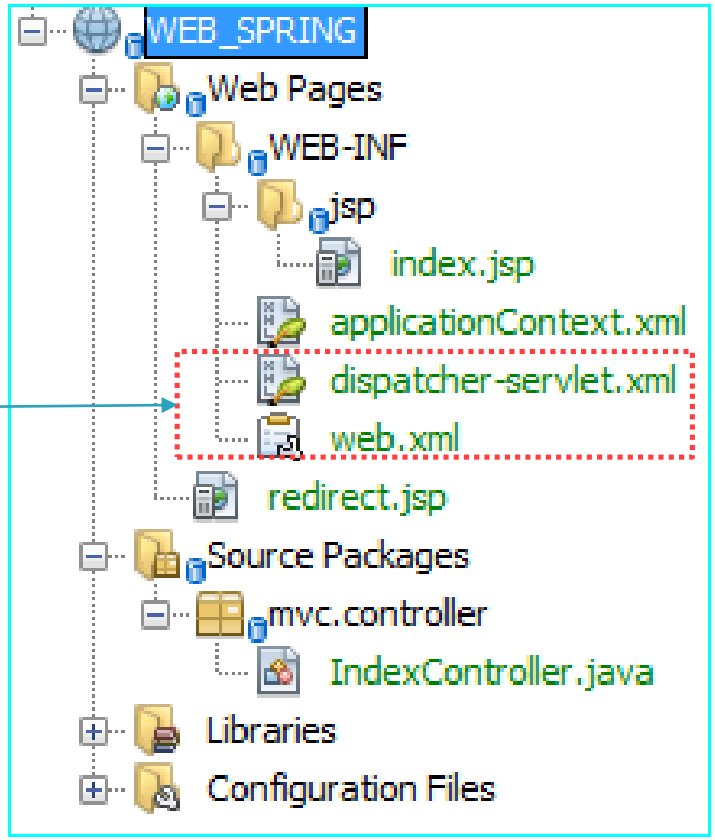


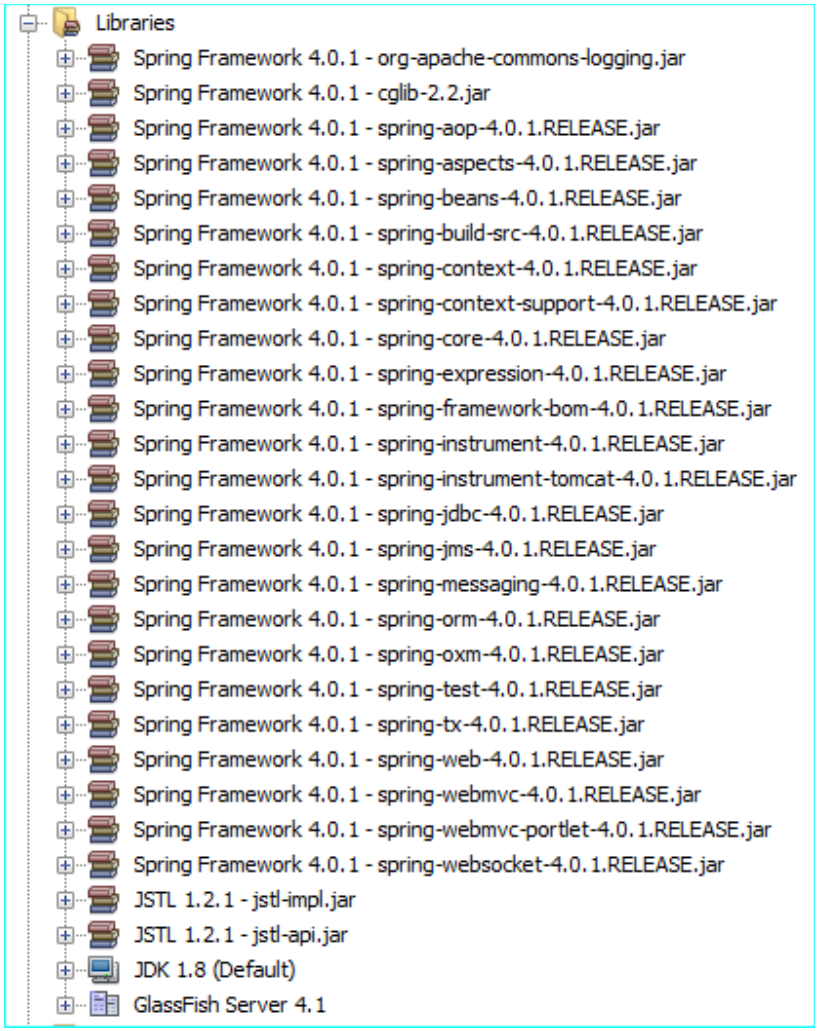
Front-Controller

Quais requisições serão atendidas pelo *Front-Controller*

Frameworks

- Projeto SPRING
  - Estrutura do Projeto





Frameworks

- Configurando Projeto SPRING
  - web.xml
    - XML de configuração do SPRING

```
10  <servlet>
11      <servlet-name>dispatcher</servlet-name>
12      <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
13      <load-on-startup>2</load-on-startup>
14  </servlet>
15  <servlet-mapping>
16      <servlet-name>dispatcher</servlet-name>
17      <url-pattern>/</url-pattern>
18  </servlet-mapping>
```

Indica qual classe fará o papel do *Front Controller*

Quais requisições serão atendidas pelo Front-Controller

Frameworks

- Configurando Projeto SPRING
  - dispatcher-servlet.xml

incluir

```
<mvc:annotation-driven />
<context:component-scan base-package="mvc.*" />
```

Define o pacote onde as suas classes Controller serão buscadas.

Define a pasta onde ficarão as Views

```
<bean id="viewResolver"
      class="org.springframework.web.servlet.view.InternalResourceViewResolver"
      p:prefix="/WEB-INF/jsp/"
      p:suffix=".jsp" />
```

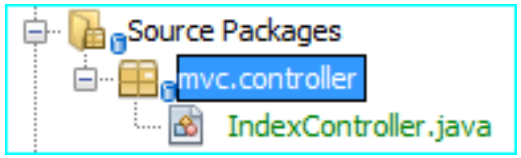
```
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:p="http://www.springframework.org/schema/p"
       xmlns:aop="http://www.springframework.org/schema/aop"
       xmlns:tx="http://www.springframework.org/schema/tx"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:mvc="http://www.springframework.org/schema/mvc"
       xsi:schemaLocation="
       http://www.springframework.org/schema/mvc
       http://www.springframework.org/schema/mvc/spring-mvc-4.0.xsd
       http://www.springframework.org/schema/context
       http://www.springframework.org/schema/context/spring-context-4.0.xsd
       http://www.springframework.org/schema/beans http://www.springframework.org/sc
       http://www.springframework.org/schema/aop http://www.springframework.org/sche
       http://www.springframework.org/schema/tx http://www.springframework.org/schem
       " >

    <mvc:annotation-driven />
    <context:component-scan base-package="mvc.*" />

    <bean id="viewResolver"
          class="org.springframework.web.servlet.view.InternalResourceViewResolver"
          p:prefix="/WEB-INF/jsp/"
          p:suffix=".jsp" />

</beans>
```

incluir

dispatcher-servlet.xml

Frameworks

- Configurando Projeto SPRING
  - 1ª Lógica
    - Crie a classe IndexController




```
6   package mvc.controller;
7
8   import org.springframework.stereotype.Controller;
9   import org.springframework.web.bind.annotation.RequestMapping;
10
11  @Controller
12  public class IndexController {
13
14      @RequestMapping("/")
15      public String index(){
16          return "index";
17      }
18  }
```

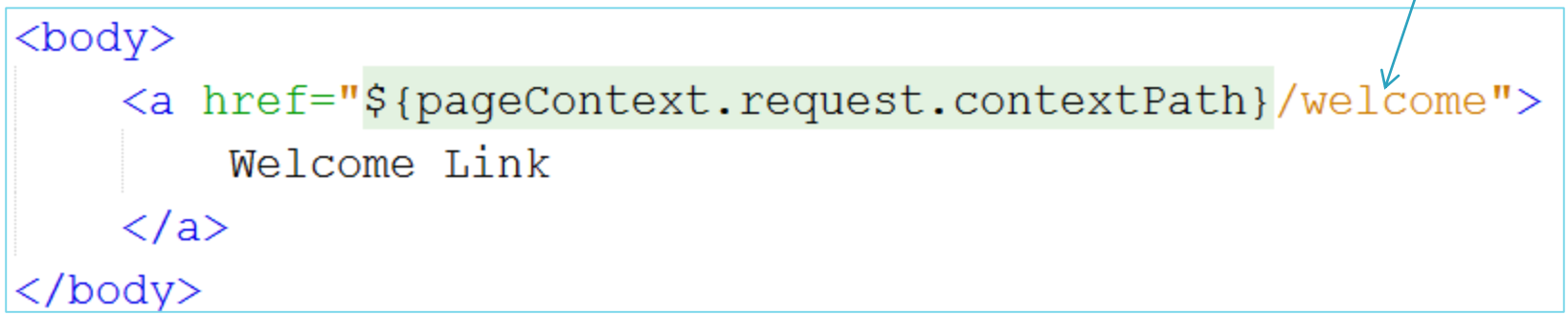
Indica que os métodos dessa classe são ações'

Indica qual será a URL utilizada para invocar o método

Indica a *View* que será oferecida como resposta: *index.jsp*

Frameworks

- Projeto SPRING
  - Incluindo uma nova lógica...
    - Modifique sua página index.jsp



index.jsp

Frameworks

- Projeto SPRING
  - Adicionando uma nova ação na classe IndexController

```
@RequestMapping("/welcome")
public ModelAndView wellcome(){
    String viewName = "welcome";
    String var = "message";
    String content = "WELCOME TO SPRING WORLD!";
    return new ModelAndView(viewName, var, content) ;
}
```

IndexController

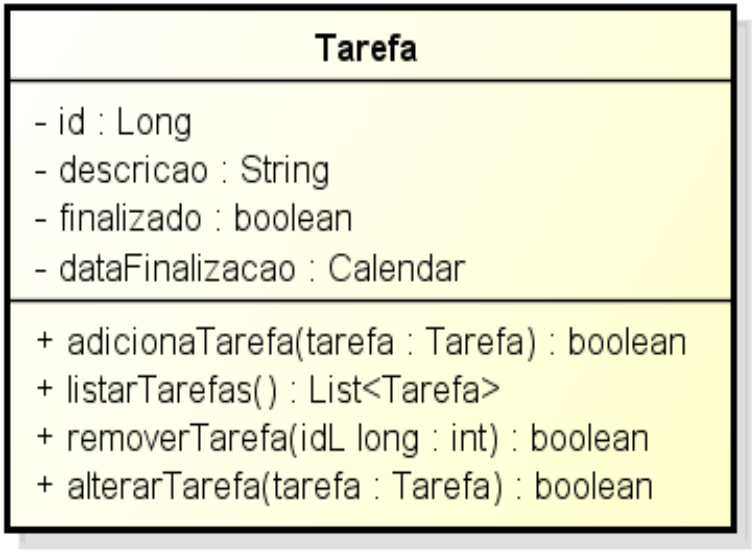
  - Criar *View* welcome.jsp

```
<html>
    <body>
        <h1>${message}</h1>
    </body>
</html>
```

welcome.jsp

Frameworks

- Projeto SPRING – Conexão com BD
  - Estudo de Caso – Tarefas
    - Lembre-se de adicionar ao projeto o driver de conexão JDBC MySQL.



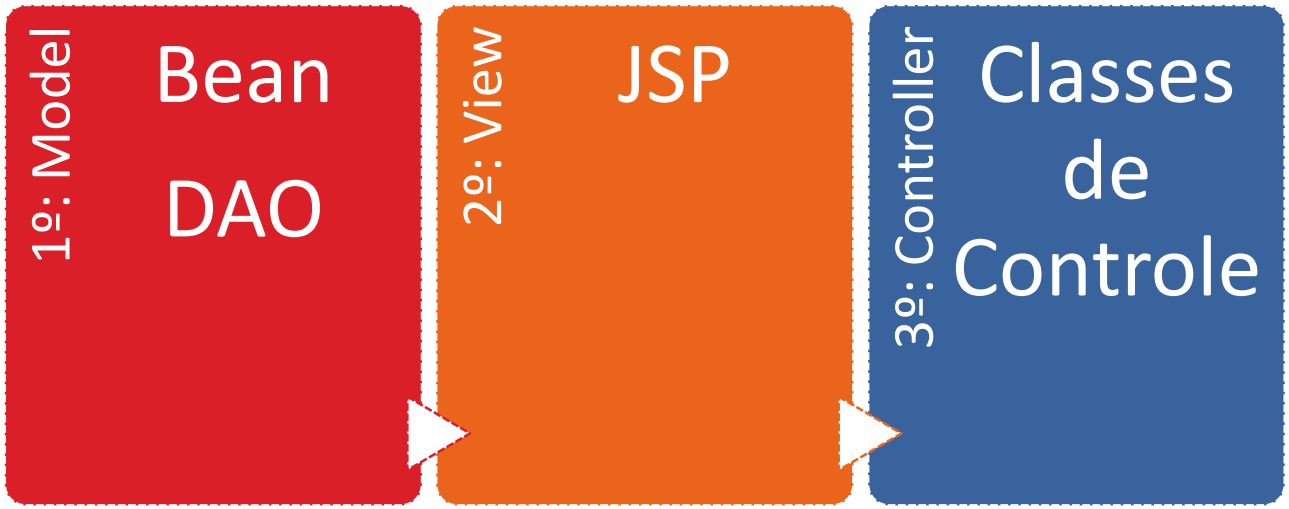

```sql
create database gestaoTarefas;

use gestaoTarefas;

create table tarefas (
id BIGINT NOT NULL AUTO_INCREMENT,
descricao VARCHAR(255),
finalizado BOOLEAN,
dataFinalizacao DATE,
primary key (id)
);
```

BD

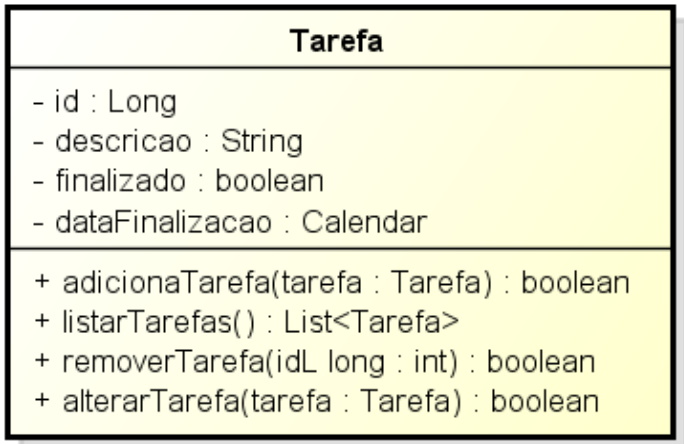
Código disponível em:
https://github.com/joyceMiranda/classCodes/tree/master/WEB_SPRING_TAREFAS

Frameworks

- Projeto SPRING – Conexão com BD
  - Estudo de Caso – Tarefas
    - Passos para solução




```java
package mvc.bean;

import java.util.Calendar;

public class Tarefa {

    private Long id;
    private String descricao;
    private boolean finalizado;
    private Calendar dataFinalizacao;

    public Tarefa() {}

    public Long getId() {...3 lines }

    public void setId(Long id) {...3 lines }

    public String getDescricao() {...3 lines }

    public void setDescricao(String descricao) {...3 lines }

    public boolean isFinalizado() {...3 lines }

    public void setFinalizado(boolean finalizado) {...3 lines }

    public Calendar getDataFinalizacao() {...3 lines }

    public void setDataFinalizacao(Calendar dataFinalizacao) {...3 lines }

}
```



Frameworks

- Projeto SPRING
  - @Repository

Permite que o SPRING instancie a classe

  - @Autowired

Injeção de dependência

  - DataSource

Provedor de Conexão


```java
@Repository
public class TarefaDAO {

    private final Connection connection;

    @Autowired
    public TarefaDAO(DataSource dataSource){
        try {
            this.connection = dataSource.getConnection();
        } catch (SQLException ex) {
            throw new RuntimeException(ex);
        }
    }

    public boolean adicionaTarefa(Tarefa tarefa){
       String sql = "insert into tarefas (descricao) values (?)";
       try (
        // prepared statement para inserção
        PreparedStatement stmt = connection.prepareStatement(sql)) {
        // seta os valores
        stmt.setString(1,tarefa.getDescricao());
        // executa
        stmt.execute();
       } catch (SQLException e) {
         throw new RuntimeException(e);
       }
       return true;
    }
}
```

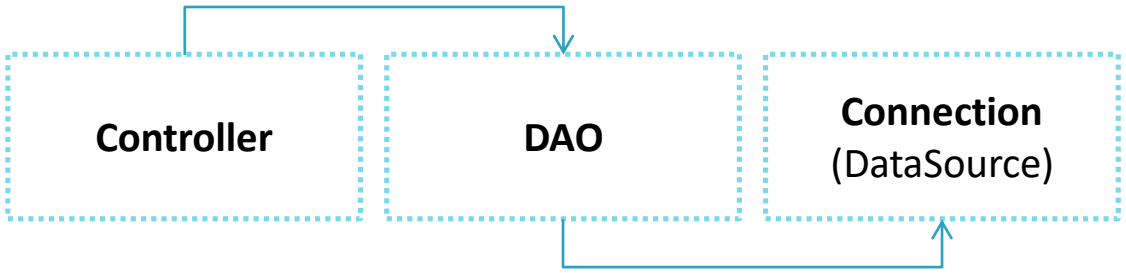
Frameworks

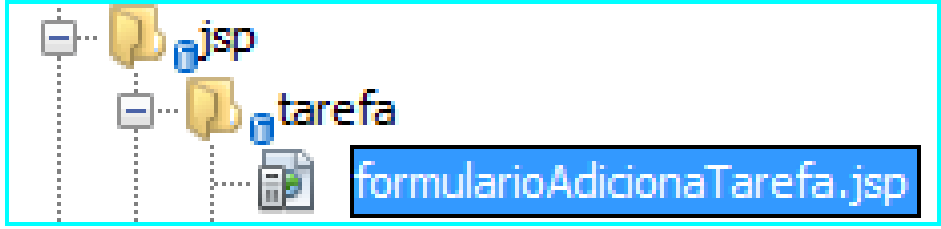
- Projeto SPRING
  - DataSource
    - XML do SPRING

Configurações de acesso ao BD

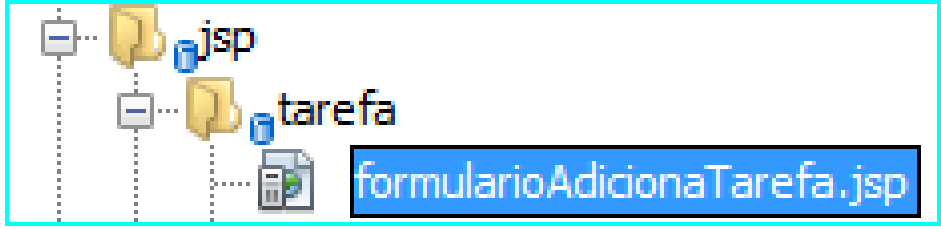
```
<bean id="dataSource"
   class="org.springframework.jdbc.datasource.DriverManagerDataSource"
   p:driverClassName="com.mysql.jdbc.Driver"
   p:url="jdbc:mysql://localhost/gestaotarefas"
   p:username="root"
   p:password=""/>
```

dispatcher-servlet.xml





- Projeto SPRING
  - Estudo de Caso – Tarefas
    - Adicionando



- Projeto SPRING
  - Estudo de Caso – Tarefas

```
6  <html>
7      <body>
8          <h1>Formulário de Adição de Tarefas</h1>
9          <form action="<c:url value='/adicionaTarefa'/>" method="post">
10             Descrição
11             <br>
12             <textarea name="descricao" rows="5" cols="100" ></textarea>
13             <br>
14             <input type="submit" value="Adicionar">
15         </form>
16     </body>
17 </html>
```

Frameworks

- Projeto SPRING


```
14  @Controller
15  public class TarefaController {
16
17      private final TarefaDAO dao;
18
19      @Autowired
20      public TarefaController(TarefaDAO dao) {
21          this.dao = dao;
22      }
23
24      @RequestMapping("/formAdicionaTarefa")
25      public String form(){
26          return "tarefa/formularioAdicionaTarefa";
27      }
28
29      @RequestMapping("/adicionaTarefa")
30      public String adiciona(Tarefa tarefa){
31          dao.adicionaTarefa(tarefa);
32  [1]     return "tarefa/tarefa-adicionada";
33      }
34
35  }
```

Frameworks

- Projeto SPRING
  - @Autowired

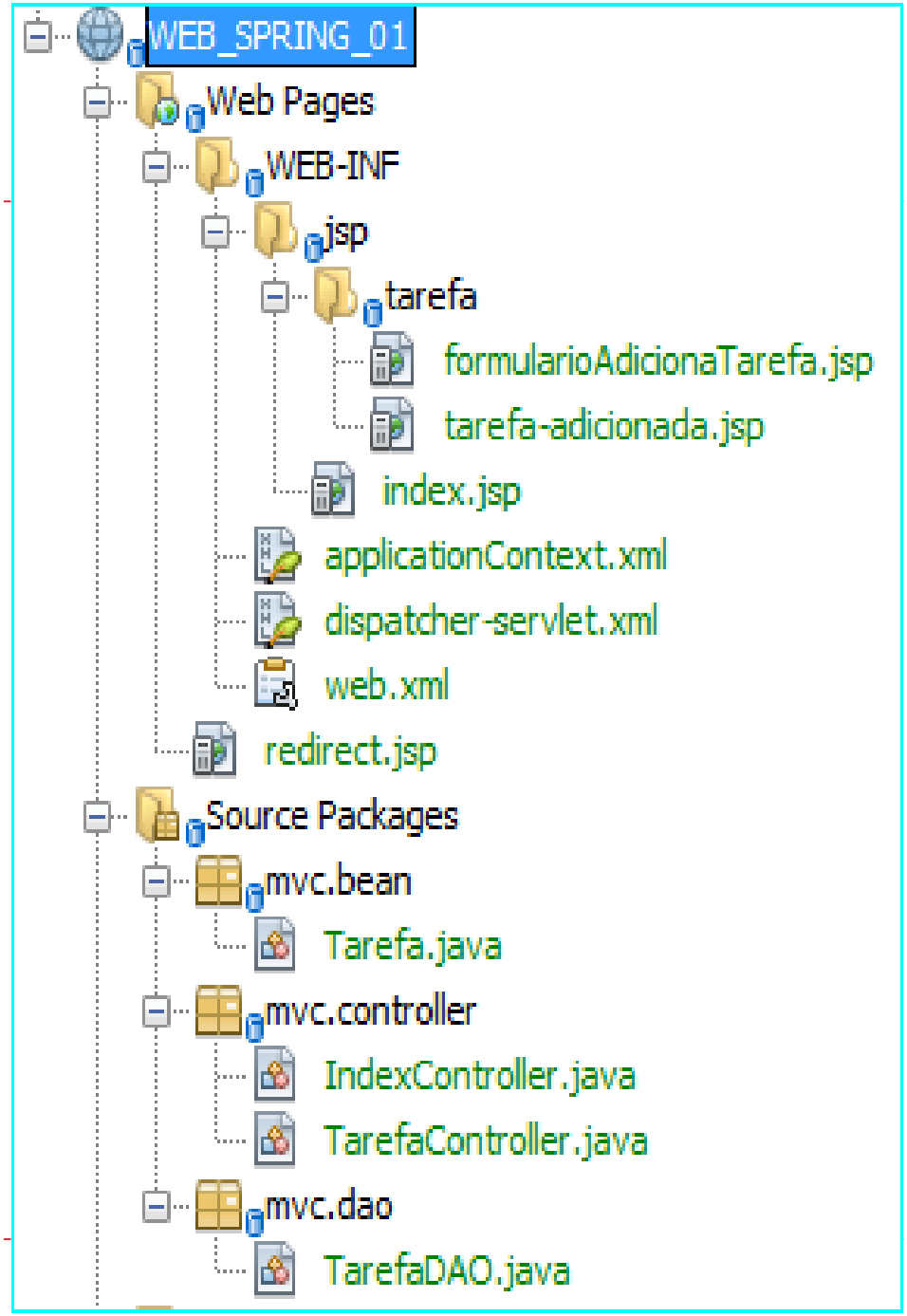
    Injeção de dependência

Avisa ao SPRING que ele precisa resolver e injetar a dependência


```
14  @Controller
15  public class TarefaController {
16
17      private final TarefaDAO dao;
18
19      @Autowired
20      public TarefaController(TarefaDAO dao) {
21          this.dao = dao;
22      }
23
24      @RequestMapping("/formAdicionaTarefa")
25      public String form(){
26          return "tarefa/formularioAdicionaTarefa";
27      }
28
29      @RequestMapping("/adicionaTarefa")
30      public String adiciona(Tarefa tarefa){
31          dao.adicionaTarefa(tarefa);
32  [1]     return "tarefa/tarefa-adicionada";
33      }
34
35  }
```

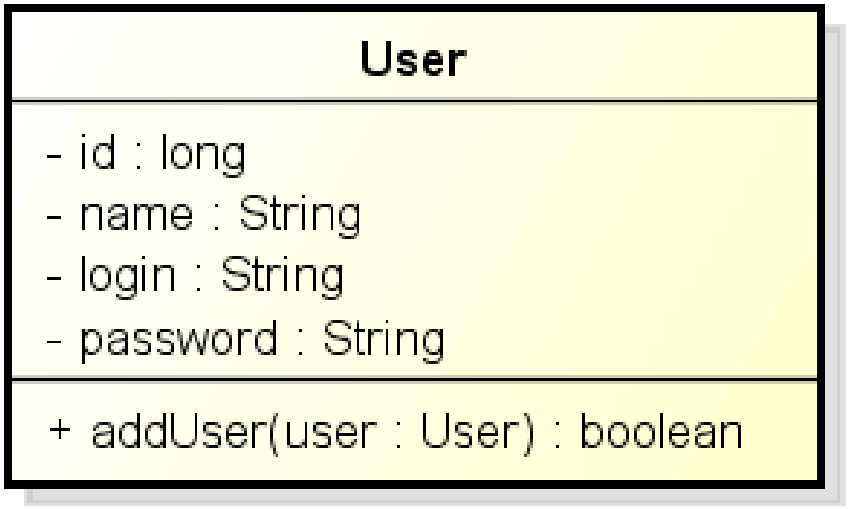
Frameworks

- Projeto SPRING
  - Estudo de Caso – Tarefas
    - Estrutura



Frameworks

- Prática
  - Utilizando o *Framework* SPRING
    - Crie uma página responsável por receber informações referentes a um usuário e inserir no banco de dados.

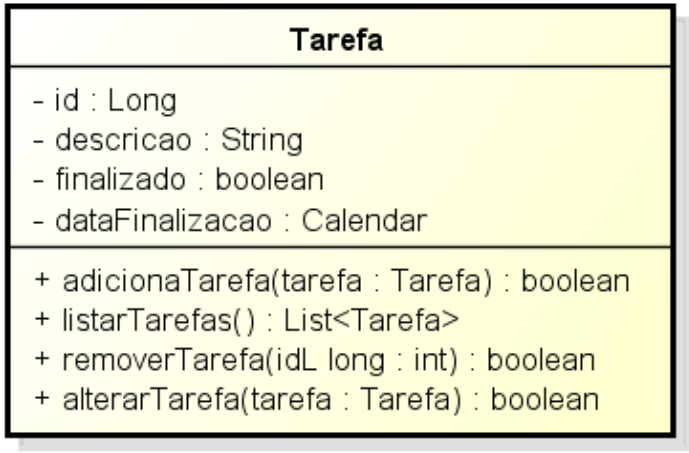


Frameworks

- Projeto SPRING
  - Listar Tarefas

Id	Descrição	Finalizada?	Data de Finalização	Ação
3	Estudar SPRING	Finalizado	08/05/2017	Alterar Remover
1	Estudar SPRING!	Finalizada	24/04/2017	Alterar Remover

Frameworks



- Projeto SPRING
  - Listar Tarefas

TarefaController

```
@RequestMapping("/listaTarefas")
public String lista(Model model){
    model.addAttribute("listaTarefas", dao.listarTarefas());
    return "tarefa/listagem-tarefas";
}
```

TarefaDAO

```
50  public List<Tarefa> listarTarefas(){
        List<Tarefa> listaTarefas = new ArrayList<Tarefa>();
52      String sql = "select * from tarefas order by descricao";
53      try (
54       PreparedStatement stmt = connection.prepareStatement(sql)) {
55       ResultSet rs = stmt.executeQuery();
56       while (rs.next()) {
57          Tarefa tarefa = new Tarefa();
58          tarefa.setId(rs.getLong("id"));
59          tarefa.setDescricao(rs.getString("descricao"));
60          tarefa.setFinalizado(rs.getBoolean("finalizado"));
61          //montando data
62          Calendar data = Calendar.getInstance();
63          if(rs.getDate("dataFinalizacao") != null)
64          {
65              data.setTime(rs.getDate("dataFinalizacao"));
66              tarefa.setDataFinalizacao(data);
67          }
68          listaTarefas.add(tarefa);
69       }
70      } catch (SQLException e) {
71        throw new RuntimeException(e);
72      }
73      return listaTarefas;
74  }
```

TarefaDAO

```
50  public List<Tarefa> listarTarefas(){
    List<Tarefa> listaTarefas = new ArrayList<Tarefa>();
52  String sql = "select * from tarefas order by descricao";
53  try (
54   PreparedStatement stmt = connection.prepareStatement(sql)) {
55   ResultSet rs = stmt.executeQuery();
56   while (rs.next()) {
57      Tarefa tarefa = new Tarefa();
58      tarefa.setId(rs.getLong("id"));
59      tarefa.setDescricao(rs.getString("descricao"));
60      tarefa.setFinalizado(rs.getBoolean("finalizado"));
61      //montando data
62      Calendar data = Calendar.getInstance();
63      if(rs.getDate("dataFinalizacao") != null)
64      {
65          data.setTime(rs.getDate("dataFinalizacao"));
66          tarefa.setDataFinalizacao(data);
67      }
68      listaTarefas.add(tarefa);
69   }
70  } catch (SQLException e) {
71    throw new RuntimeException(e);
72  }
73  return listaTarefas;
74  }
```

```
50      public List<Tarefa> listarTarefas(){
            List<Tarefa> listaTarefas = new ArrayList<Tarefa>();
52          String sql = "select * from tarefas order by descricao";
53          try (
54           PreparedStatement stmt = connection.prepareStatement(sql)) {
55           ResultSet rs = stmt.executeQuery();
56           while (rs.next()) {
57              Tarefa tarefa = new Tarefa();
58              tarefa.setId(rs.getLong("id"));
59              tarefa.setDescricao(rs.getString("descricao"));
60              tarefa.setFinalizado(rs.getBoolean("finalizado"));
61              //montando data
62              Calendar data = Calendar.getInstance();
63              if(rs.getDate("dataFinalizacao") != null)
64              {
65                  data.setTime(rs.getDate("dataFinalizacao"));
66                  tarefa.setDataFinalizacao(data);
67              }
68              listaTarefas.add(tarefa);
69           }
70          } catch (SQLException e) {
71            throw new RuntimeException(e);
72          }
73          return listaTarefas;
74      }
```

TarefaDAO

TarefaDAO

```
50  public List<Tarefa> listarTarefas(){
    List<Tarefa> listaTarefas = new ArrayList<Tarefa>();
52  String sql = "select * from tarefas order by descricao";
53  try (
54   PreparedStatement stmt = connection.prepareStatement(sql)) {
55   ResultSet rs = stmt.executeQuery();
56  while (rs.next()) {
57     Tarefa tarefa = new Tarefa();
58     tarefa.setId(rs.getLong("id"));
59     tarefa.setDescricao(rs.getString("descricao"));
60     tarefa.setFinalizado(rs.getBoolean("finalizado"));
61     //montando data
62     Calendar data = Calendar.getInstance();
63     if(rs.getDate("dataFinalizacao") != null)
64     {
65        data.setTime(rs.getDate("dataFinalizacao"));
66        tarefa.setDataFinalizacao(data);
67     }
68     listaTarefas.add(tarefa);
69  }
70  } catch (SQLException e) {
71    throw new RuntimeException(e);
72  }
73  return listaTarefas;
74  }
```

TarefaDAO

```
50      public List<Tarefa> listarTarefas(){
        List<Tarefa> listaTarefas = new ArrayList<Tarefa>();
52      String sql = "select * from tarefas order by descricao";
53      try (
54       PreparedStatement stmt = connection.prepareStatement(sql)) {
55       ResultSet rs = stmt.executeQuery();
56       while (rs.next()) {
57          Tarefa tarefa = new Tarefa();
58          tarefa.setId(rs.getLong("id"));
59          tarefa.setDescricao(rs.getString("descricao"));
60          tarefa.setFinalizado(rs.getBoolean("finalizado"));
61          //montando data
62          Calendar data = Calendar.getInstance();
63          if(rs.getDate("dataFinalizacao") != null)
64          {
65              data.setTime(rs.getDate("dataFinalizacao"));
66              tarefa.setDataFinalizacao(data);
67          }
68          listaTarefas.add(tarefa);
69       }
70      } catch (SQLException e) {
71       throw new RuntimeException(e);
72      }
73      return listaTarefas;
74      }
```

listagem-tarefas.jsp

```
7   <body>
8       <a href="<c:url value='/'/>">Voltar</a> <br><br>
9       <table>
10          <tr>
11              <th>Id</th>
12              <th>Descrição</th>
13              <th>Finalizada?</th>
14              <th>Data de Finalização</th>
15          </tr>
16          <c:forEach items="${listaTarefas}" var="tarefa">
17          <tr>
18              <td>${tarefa.id}</td>
19              <td>${tarefa.descricao}</td>
20              <c:if test="${tarefa.finalizado eq false}">
21               <td>Não finalizada</td>
22              </c:if>
23              <c:if test="${tarefa.finalizado eq true}">
24               <td>Finalizada</td>
25              </c:if>
26              <td>
27              <fmt:formatDate value="${tarefa.dataFinalizacao.time}" pattern="dd/MM/yyyy"/>
                </td>
29          </tr>
30          </c:forEach>
31      </table>
32  </body>
```

Frameworks

- Projeto SPRING
  - Redirecionando Requisições
    - Remover Tarefa

Id	Descrição	Finalizada?	Data de Finalização	Ação
37	Estudar Framework Spring.	Não finalizada		Alterar Remover
38	Estudar JavaScript.	Finalizada	18/12/2015	Alterar Remover
36	Estudar Modelo MVC.	Não finalizada		Alterar Remover

```
<a href="removeTarefa?id=${tarefa.id}"
   onclick="return confirm('Deseja realmente excluir?')" > Remover </a>
```

Frameworks

- Projeto SPRING
  - Redirecionando Requisições
    - Remover Tarefa
      - redirect

```
<a href="removeTarefa?id=${tarefa.id}"
   onclick="return confirm('Deseja realmente excluir?')" > Remover </a>
```

```
@RequestMapping("/removeTarefa")
public String remove(Long id){
    dao.removerTarefa(id);
    return "redirect:/listaTarefas";
}
```

TarefaController

Frameworks

- Projeto SPRING
  - Redirecionando Requisições
    - Remover Tarefa

```
public boolean removerTarefa(Long id){
    String sql = "delete from tarefas where id = ? ";
    try (
     PreparedStatement stmt = connection.prepareStatement(sql)) {
     stmt.setLong(1,id);
     stmt.execute();
    } catch (SQLException e) {
      throw new RuntimeException(e);
    }
    return true;
}
```

TarefaDao

Frameworks

- Projeto SPRING
  - Redirecionando Requisições
    - Alterar Tarefa

Id	Descrição	Finalizada?	Data de Finalização	Ação
37	Estudar Framework Spring.	Não finalizada		Alterar Remover
38	Estudar JavaScript.	Finalizada	18/12/2015	Alterar Remover
36	Estudar Modelo MVC.	Não finalizada		Alterar Remover

```
<a href="exibeTarefa?id=${tarefa.id}"> Alterar </a>
```

Frameworks

- Projeto SPRING
  - Redirecionando Requisições
    - Alterar Tarefa

```
<a href="exibeTarefa?id=${tarefa.id}"> Alterar </a>
```

```
@RequestMapping("/exibeTarefa")
public String exibe(Long id, Model model){
    model.addAttribute("tarefa", dao.buscarTarefaPorId(id));
    return "tarefa/exibe-tarefa";
}
```

TarefaController

```
76      public Tarefa buscarTarefaPorId(Long id){
77          String sql = "select * from tarefas where id = ? ";
78          try (
79           PreparedStatement stmt = connection.prepareStatement(sql)) {
80           stmt.setLong(1,id);
81           ResultSet rs = stmt.executeQuery();
82           Tarefa tarefa = new Tarefa();
83           if(rs.next()) {
84              tarefa.setId(rs.getLong("id"));
85              tarefa.setDescricao(rs.getString("descricao"));
86              tarefa.setFinalizado(rs.getBoolean("finalizado"));
87              //montando data
88              Calendar data = Calendar.getInstance();
89              if(rs.getDate("dataFinalizacao") != null)
90              {
91                  data.setTime(rs.getDate("dataFinalizacao"));
92                  tarefa.setDataFinalizacao(data);
93              }
94           }
95           return tarefa;
96          } catch (SQLException e) {
97           throw new RuntimeException(e);
98          }
99      }
```

TarefaDao

Frameworks

- Projeto SPRING
  - Redirecionando Requisições
    - Alterar Tarefa

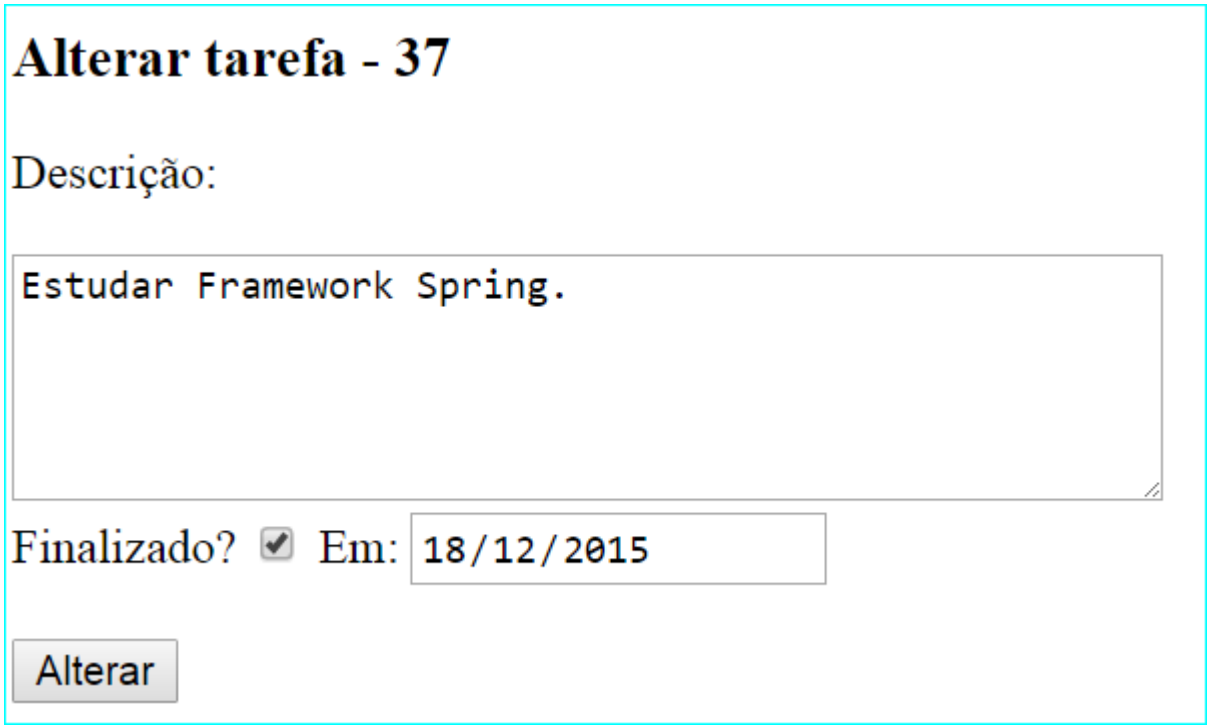
```
<a href="exibeTarefa?id=${tarefa.id}"> Alterar </a>
```

```
@RequestMapping("/exibeTarefa")
public String exibe(Long id, Model model){
    model.addAttribute("tarefa", dao.buscarTarefaPorId(id));
    return "tarefa/exibe-tarefa";
}
```

TarefaController

Frameworks

- Projeto SPRING
  - Redirecionando Requisições
    - Exibindo Tarefa



exibe-tarefa.jsp

```
<body>
    <h3>Alterar tarefa - ${tarefa.id}</h3>
    <form action="<c:url value='/alteraTarefa'/>" method="post">
        <input type="hidden" name="id" value="${tarefa.id}" />
        Descrição:
        <br><form:errors path="tarefa.descricao" cssStyle="color:red"/> <br>
        <textarea name="descricao" cols="50" rows="5">${tarefa.descricao}</textarea>
        <br>
        Finalizado?
        <input type="checkbox" name="finalizado"
        value="true" ${tarefa.finalizado?'checked':''}/>
        Em:
        <form:errors path="tarefa.dataFinalizacao" cssStyle="color:red"/>
        <fmt:formatDate var="fmtDate" value="${tarefa.dataFinalizacao.time}"
                        pattern="yyyy-MM-dd"/>
        <input type="date" name="dataFinalizacao" value="${fmtDate}" >
        <br><br>
        <input type="submit" value="Alterar"/>
        <br><br>
    </form>
    <a href="<c:url value='/'/>">Voltar</a>
</body>
```

exibe-tarefa.jsp

```
<body>
    <h3>Alterar tarefa - ${tarefa.id}</h3>
    <form action="<c:url value='/alteraTarefa'/>" method="post">
        <input type="hidden" name="id" value="${tarefa.id}" />
        Descrição:
        <br><form:errors path="tarefa.descricao" cssStyle="color:red"/> <br>
        <textarea name="descricao" cols="50" rows="5">${tarefa.descricao}</textarea>
        <br>
        Finalizado?
        <input type="checkbox" name="finalizado"
        value="true" ${tarefa.finalizado?'checked':''}/>
        Em:
        <form:errors path="tarefa.dataFinalizacao" cssStyle="color:red"/>
        <fmt:formatDate var="fmtDate" value="${tarefa.dataFinalizacao.time}"
                        pattern="yyyy-MM-dd"/>
        <input type="date" name="dataFinalizacao" value="${fmtDate}" >
        <br><br>
        <input type="submit" value="Alterar"/>
        <br><br>
    </form>
    <a href="<c:url value='/'/>">Voltar</a>
</body>
```


exibe-tarefa.jsp

```
<body>
    <h3>Alterar tarefa - ${tarefa.id}</h3>
    <form action="<c:url value='/alteraTarefa'/>" method="post">
        <input type="hidden" name="id" value="${tarefa.id}" />
        Descrição:
        <br><form:errors path="tarefa.descricao" cssStyle="color:red"/> <br>
        <textarea name="descricao" cols="50" rows="5">${tarefa.descricao}</textarea>
        <br>
        Finalizado?
        <input type="checkbox" name="finalizado"
        value="true" ${tarefa.finalizado?'checked':''}/>
        Em:
        <form:errors path="tarefa.dataFinalizacao" cssStyle="color:red"/>
        <fmt:formatDate var="fmtDate" value="${tarefa.dataFinalizacao.time}"
                        pattern="yyyy-MM-dd"/>
        <input type="date" name="dataFinalizacao" value="${fmtDate}" >
        <br><br>
        <input type="submit" value="Alterar"/>
        <br><br>
    </form>
    <a href="<c:url value='/'/>">Voltar</a>
</body>
```


exibe-tarefa.jsp

```
<body>
    <h3>Alterar tarefa - ${tarefa.id}</h3>
    <form action="<c:url value='/alteraTarefa'/>" method="post">
        <input type="hidden" name="id" value="${tarefa.id}" />
        Descrição:
        <br><form:errors path="tarefa.descricao" cssStyle="color:red"/> <br>
        <textarea name="descricao" cols="50" rows="5">${tarefa.descricao}</textarea>
        <br>
        Finalizado?
        <input type="checkbox" name="finalizado"
        value="true" ${tarefa.finalizado?'checked':''}/>
        Em:
        <form:errors path="tarefa.dataFinalizacao" cssStyle="color:red"/>
        <fmt:formatDate var="fmtDate" value="${tarefa.dataFinalizacao.time}"
                        pattern="yyyy-MM-dd"/>
        <input type="date" name="dataFinalizacao" value="${fmtDate}" >
        <br><br>
        <input type="submit" value="Alterar"/>
        <br><br>
    </form>
    <a href="<c:url value='/'/>">Voltar</a>
</body>
```


exibe-tarefa.jsp

Frameworks

- Projeto SPRING
  - Redirecionando Requisições
    - Alterando Tarefas

```
@RequestMapping("/alteraTarefa")
public String altera(@Valid Tarefa tarefa, BindingResult result){
    if(result.hasErrors()){
        return "tarefa/exibe-tarefa";
    }
    dao.alterarTarefa(tarefa);
    return "redirect:/listaTarefas";
}
```

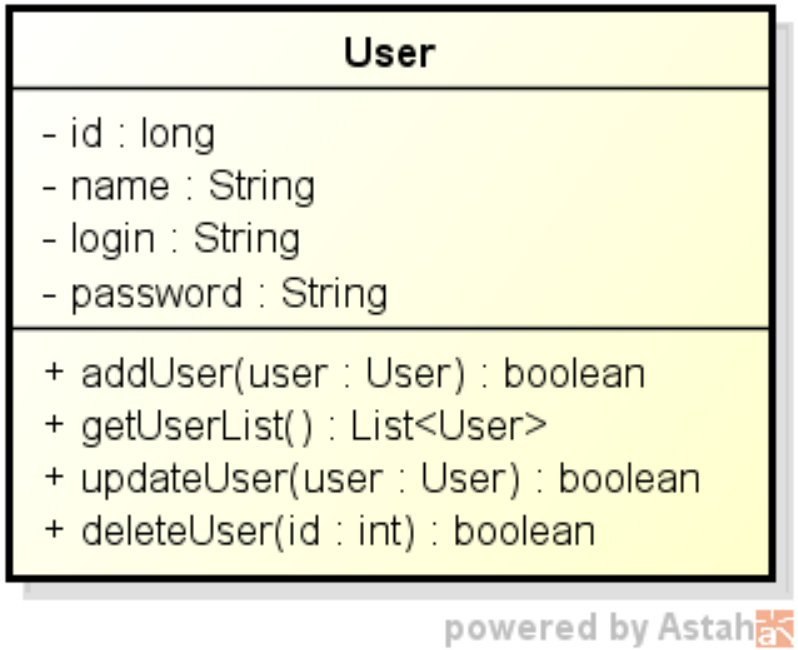
TarefaController

```
114     public boolean alterarTarefa(Tarefa tarefa){
115         String sql = "update tarefas set descricao = ?,"
116                 + " finalizado=?, dataFinalizacao=? where id = ? ";
117         try (
118          PreparedStatement stmt = connection.prepareStatement(sql)) {
119          stmt.setString(1,tarefa.getDescricao());
120          stmt.setBoolean(2,tarefa.isFinalizado());
121          if(tarefa.getDataFinalizacao() != null){
122              stmt.setDate(3, new Date(tarefa.getDataFinalizacao().getTimeInMillis()));
123          }else{
124              stmt.setNull(3, java.sql.Types.DATE);
125          }
126          stmt.setLong(4,tarefa.getId());
127          stmt.execute();
128         } catch (SQLException e) {
129          throw new RuntimeException(e);
130         }
131         return true;
132     }
```

TarefaDao

Frameworks

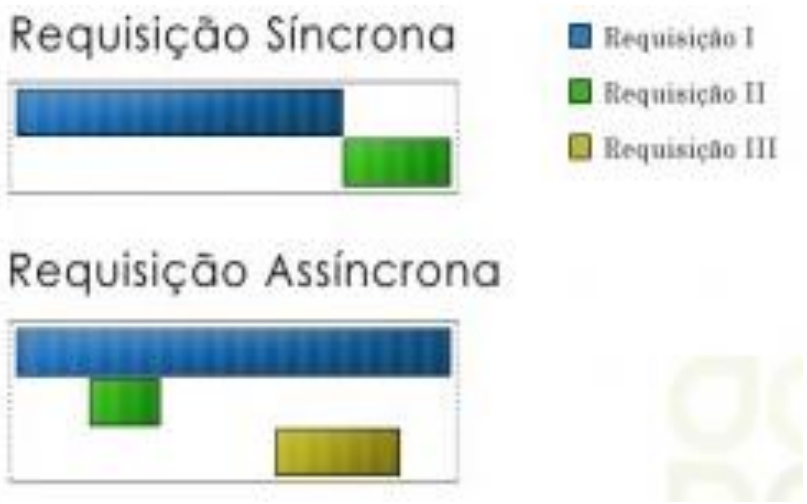
- Prática
  - Utilizando o *Framework* SPRING
    - Implemente as funcionalidades de adicionar, listar, alterar e excluir usuários.



Frameworks

- Projeto SPRING
  - Melhorando a Usabilidade
    - AJAX - *Asynchronous Javascript and XML*
      - Requisições assíncronas
      - Utiliza árvore DOM para modificar o conteúdo sem recarregar a página
        - Economia de tráfego de internet e melhora da experiência do usuário
      - Faz requisições utilizando um objeto Javascript e não um formulário

Extensible Markup Language
Linguagem Extensível de Marcação Genérica



Frameworks

- Projeto SPRING
  - Melhorando a Usabilidade
    - Carregar apenas o conteúdo que mudou com a nova requisição?

Id	Descrição	Finalizada?	Data de Finalização	Ação
37	Estudar Framework Spring.	Não finalizada		Alterar Remover

Id	Descrição	Finalizada?	Data de Finalização	Ação
37	Estudar Framework Spring.	Não finalizada Finalizar agora?		Alterar Remover

```
<a href="#" onclick="finalizaAgora(${tarefa.id})"> Finalizar agora? </a>
```

Frameworks

- Projeto SPRING
  - Melhorando a Usabilidade
    - AJAX
      - Marcação das regiões específicas que serão atualizadas

```
<td id="tarefa_${tarefa.id}">
     Não finalizada <br>
     <a href="#" onclick="finalizaAgora(${tarefa.id})"> Finalizar agora? </a>
</td>
```

```
<td id="tarefa_data_${tarefa.id}">
     <fmt:formatDate value="${tarefa.dataFinalizacao.time}" pattern="dd/MM/yyyy"/>
</td>
```

Id	Descrição	Finalizada?	Data de Finalização	Ação
37	Estudar Framework Spring.	Não finalizada Finalizar agora?		Alterar Remover

listagem-tarefa.jsp

Frameworks

- Projeto SPRING
  - Melhorando a Usabilidade
    - AJAX
      - Suporte *JQuery* para trabalhar com AJAX

```
function finalizaAgora(id) {
    $.post("finalizaTarefa", {'id' : id}, function(dadosDeResposta) {
        // selecionando o elemento html através da
        // ID e alterando o HTML dele
        $("#tarefa_"+id).html("Finalizado");
        $("#tarefa_data_"+id).html(dadosDeResposta);
    });
}
```

default.js

Frameworks

- Projeto SPRING
  - Melhorando a Usabilidade
    - AJAX

TarefaController

```
@RequestMapping("/finalizaTarefa")
public String finaliza(Long id, Model model) {
    dao.finalizarTarefa(id);
    model.addAttribute("tarefa", dao.buscarTarefaPorId(id));
    return "tarefa/dataFinalizada";
}
```

Frameworks

- Projeto SPRING
  - Melhorando a Usabilidade
    - AJAX

TarefaController

```
@RequestMapping("/finalizaTarefa")
public String finaliza(Long id, Model model) {
    dao.finalizarTarefa(id);
    model.addAttribute("tarefa", dao.buscarTarefaPorId(id));
    return "tarefa/dataFinalizada";
}
```

Frameworks

- Projeto SPRING
  - Melhorando a Usabilidade
    - AJAX

TarefaController

```
@RequestMapping("/finalizaTarefa")
public String finaliza(Long id, Model model) {
    dao.finalizarTarefa(id);
    model.addAttribute("tarefa", dao.buscarTarefaPorId(id));
    return "tarefa/dataFinalizada";
}
```

Frameworks

- Projeto SPRING
  - Melhorando a Usabilidade
    - AJAX - jquery

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

<fmt:formatDate value="${tarefa.dataFinalizacao.time}" pattern="dd/MM/yyyy" />
```

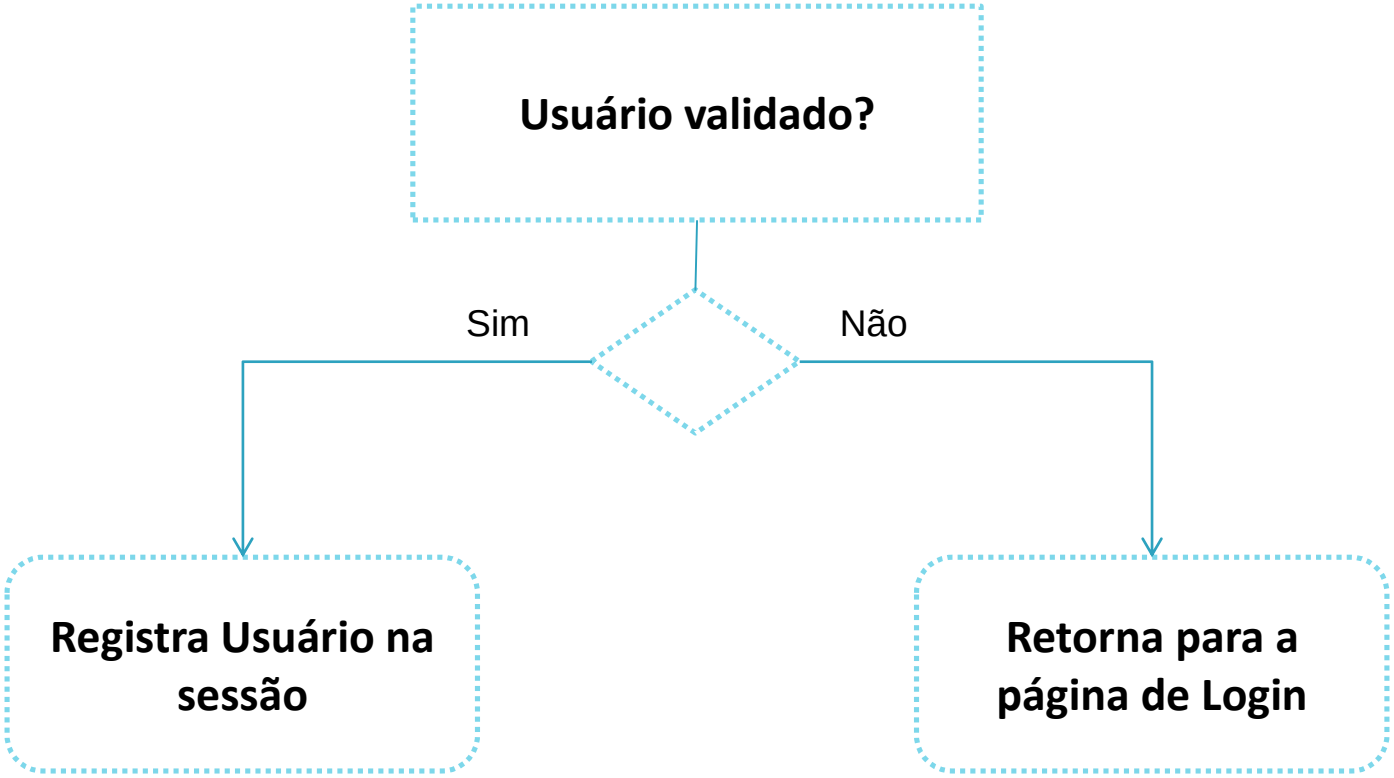
dataFinalizada.jsp

Frameworks

- Autenticação e Autorização
  - Acesso a conteúdo restrito
  - Protocolo HTTP não mantém informações sobre quem é quem entre uma conexão e outra

Frameworks

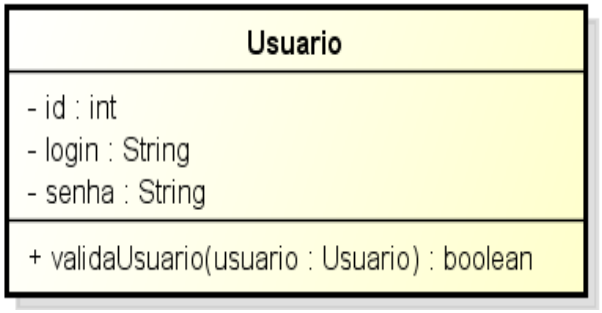
- Projeto SPRING – Login da Aplicação

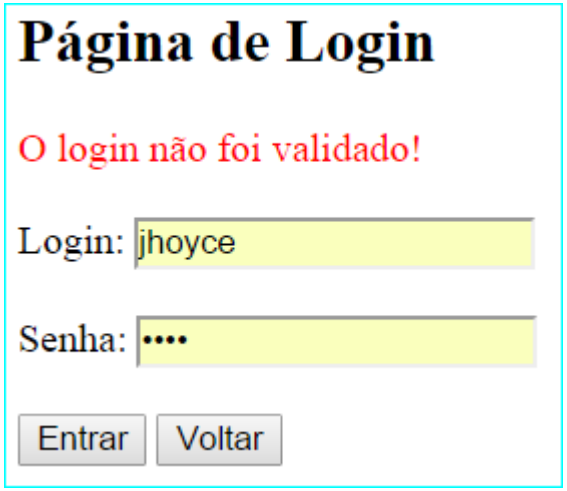


Frameworks

- Projeto SPRING – Login da Aplicação
  - Validação de Usuário

```
<div class="msgErro">${msgLoginInvalido}</div><br>
```





formularioLogin.jsp

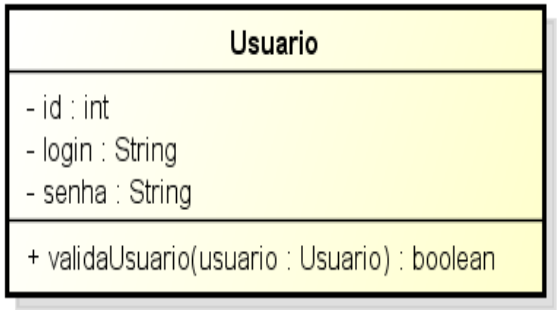
Frameworks

- Projeto SPRING – Login da Aplicação
  - Formulário de Login

```
7  <html>
8      <head>
9          <link rel="stylesheet" href="<c:url value='/resources/css/default.css'/>"
10     </head>
11     <body>
12         <h2>Página de Login</h2>
13         <div class="msgErro">${msgLoginInvalido}</div><br>
14         <form action="<c:url value='/efetuaLogin'/>" method="post">
15             Login: <input type="text" name="login" /> <br><br>
16             Senha: <input type="password" name="senha" /> <br><br>
17             <input type="submit" value="Entrar" />
18             <input type="button" value="Voltar" onclick="history.back()" />
19         </form>
20     </body>
21 </html>
```

formularioLogin.jsp

Frameworks



- Projeto SPRING – Login da Aplicação
  - Validação de Usuário

```
34      @RequestMapping("/efetuaLogin")
35      public String efetuaLogin(Usuario user, HttpSession session){
36          if(dao.validaUsuario(user)){
37              session.setAttribute("usuarioLogado", user);
38              session.removeAttribute("msgLoginInvalido");
39              return "menuAdm";
40          }else{
41              session.setAttribute("msgLoginInvalido", "O login não foi validado!");
42              return "redirect:formLogin";
43          }
44
45      }
```

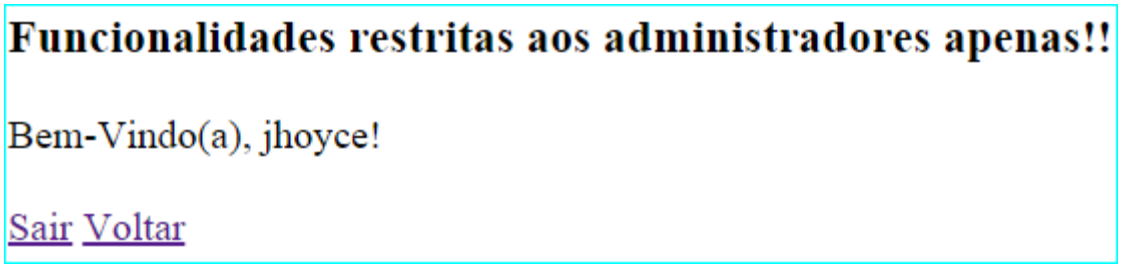
LoginController

Frameworks

- Projeto SPRING – Login da Aplicação
  - Validação de Usuário

```
10  <html>
11      <body>
12          <h1>Funcionalidades restritas aos administradores apenas!!</h1>
13          Bem-Vindo(a), ${usuarioLogado.login}!
14          <br><br>
15          <a href="<c:url value='/logout'/>">Sair</a>
16          <a href="<c:url value='/'/>">Voltar</a>
17      </body>
18  </html>
```

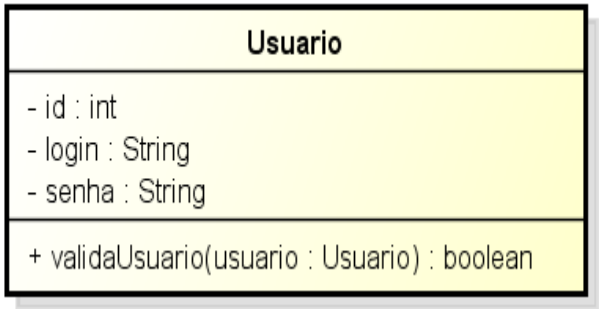
menuAdm.jsp



Frameworks

- Projeto SPRING – Login da Aplicação
  - Serialização
    - Salvar, Gravar e Capturar o estado atual de um objeto
    - Alguns *servlet contatiners* como o Glassfish só permitem gravar na sessão objetos serializados

```
public class Usuario implements Serializable{

    private int id;
    private String login;
    private String senha;

    public Usuario() {}
```



Bean Usuario

Frameworks

- Projeto SPRING – Login da Aplicação
  - Bloqueando acesso indevido
    - Em algumas situações precisamos garantir que antes de executarmos alguma ação o usuário esteja autenticado, ou seja, armazenado na sessão.
    - Interceptadores
      - Funcionam como Filtros
        - Classes que permitem executar códigos antes da requisição e depois da resposta
      - Recuperando sessão

```
HttpSession session = request.getSession();
```

```
session.getAttribute("usuarioLogado")
```

Frameworks

- Projeto SPRING – Login da Aplicação
  - Bloquando acesso indevido
    - Interceptadores
      - Registrar o interceptador
      - O Spring MVC não permite esse registro via anotações, então usaremos a configuração via XML.

```
<!-- Filtro de autorização e autenticação-->
<mvc:interceptors>
    <bean class="mvc.interceptor.AutorizadorInterceptor" />
</mvc:interceptors>
```

dispatcher_servlet.xml

```java
public class AutorizadorInterceptor extends HandlerInterceptorAdapter {

    @Override
    public boolean preHandle(HttpServletRequest request,
            HttpServletResponse response,
            Object controller) throws Exception {

        String uri = request.getRequestURI();

        if( uri.endsWith("/") ||
            uri.endsWith("formLogin") ||
            uri.endsWith("efetuaLogin") ||
            uri.endsWith("listaTarefas") ||
            uri.contains("resources")){
            return true;
        }

        HttpSession session = request.getSession();

        if(session.getAttribute("usuarioLogado") != null) {
            return true;
        }

        response.sendRedirect("formLogin");

        return false;
    }
}
```

Retorna um *boolean* que indica se é para continuar ou não com a requisição

AutorizadorInterceptor

```java
public class AutorizadorInterceptor extends HandlerInterceptorAdapter {

    @Override
    public boolean preHandle(HttpServletRequest request,
            HttpServletResponse response,
            Object controller) throws Exception {

        String uri = request.getRequestURI();

        if( uri.endsWith("/") ||
            uri.endsWith("formLogin") ||
            uri.endsWith("efetuaLogin") ||
            uri.endsWith("listaTarefas") ||
            uri.contains("resources")){
            return true;
        }

        HttpSession session = request.getSession();

        if(session.getAttribute("usuarioLogado") != null) {
            return true;
        }

        response.sendRedirect("formLogin");

        return false;
    }
}
```

Ignorando requisições que não precisam de validação

AutorizadorInterceptor

```java
public class AutorizadorInterceptor extends HandlerInterceptorAdapter {

    @Override
    public boolean preHandle(HttpServletRequest request,
            HttpServletResponse response,
            Object controller) throws Exception {

        String uri = request.getRequestURI();

        if( uri.endsWith("/") ||
            uri.endsWith("formLogin") ||
            uri.endsWith("efetuaLogin") ||
            uri.endsWith("listaTarefas") ||
            uri.contains("resources")){
            return true;
        }

        HttpSession session = request.getSession();

        if(session.getAttribute("usuarioLogado") != null) {
            return true;
        }

        response.sendRedirect("formLogin");

        return false;
    }
}
```

Recuperando e Validando Sessão

AutorizadorInterceptor

EXTRAS

Frameworks

- Projeto SPRING
  - Internacionalização

```
<fmt:message key="tarefa.descricao" />
```

```
<!-- path dos properties de mensagens -->
<bean id="messageSource"
    class="org.springframework.context.support.ReloadableResourceBundleMessageSource">
    <property name="basename" value="/WEB-INF/mensagens" />
</bean>
```

dispatcher-servlet.xml

```
tarefa.descricao=Descrição
```

mensagens_pt.properties

```
tarefa.descricao=Description
```

mensagens_en.properties

Frameworks

- Projeto SPRING
  - Internacionalização

```
<!-- linguagem padrão -->
<bean id="localeResolver" class="org.springframework.web.servlet.i18n.CookieLocaleResolver">
    <property name="defaultLocale" value="pt"/>
</bean>

<!-- Changes the locale when a 'lang' request parameter is sent; e.g. /?lang=pt -->
<mvc:interceptors>
    <bean class="org.springframework.web.servlet.i18n.LocaleChangeInterceptor"
          p:paramName="lang" />
</mvc:interceptors>
```

dispatcher-servlet.xml

Frameworks

- Projeto SPRING
  - Internacionalização
    - Validação da Camada de Visão : **form:errors**

```
5   <%@taglib uri="http://www.springframework.org/tags/form" prefix="form" %>
6   <html>
7       <body>
8           <h1>Formulário de Adição de Tarefas</h1>
9           <form action="<c:url value='/adicionaTarefa'/>" method="post">
10              <fmt:message key="tarefa.descricao"/>
11              <br>
12              <textarea name="descricao" rows="5" cols="100" ></textarea>
13              <br>
14              <form:errors path="tarefa.descricao" cssStyle="color:red"/>
15              <br>
16              <input type="submit" value="Adicionar">
17          </form>
18      </body>
19  </html>
```

Frameworks

- Projeto SPRING
  - Listar Tarefas

Id	Descrição	Finalizada?	Data de Finalização
19	Ser dedicado!	Não finalizada	
21	Ser determinado!	Não finalizada	

Frameworks

- EXTRAS
  - Incluindo Recursos Externos [css, js, imgs]

```
<head>
    <link rel="stylesheet" href="<c:url value='/resources/css/default.css'/>" >
</head>
```

```
<!-- caminho de recursos [css, js, imgs] -->
<mvc:resources mapping="/resources/**" location="/resources/" />
```

dispatcher-servlet.xml

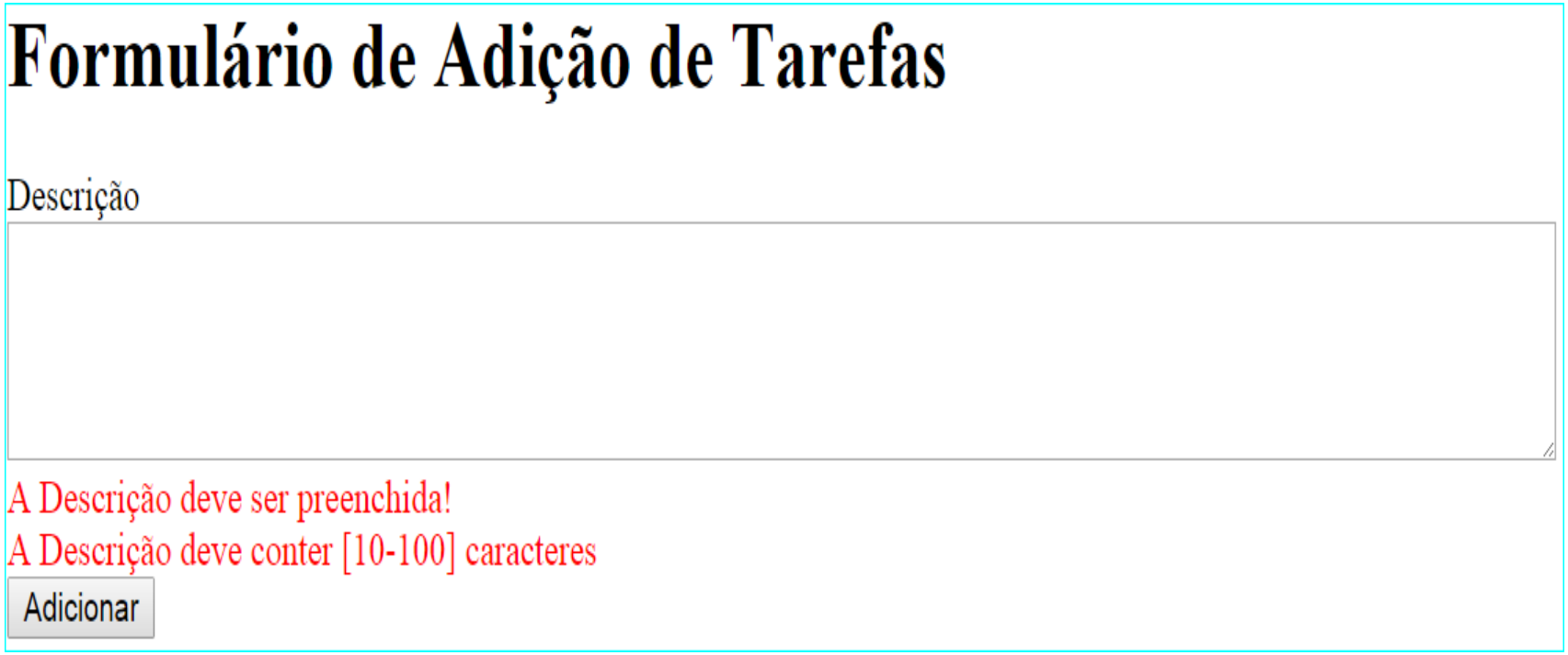
Frameworks

- Autenticação e Autorização
  - **Sessão**
    - Configurando o tempo limite – *web.xml*

```
<session-config>
    <session-timeout>
        30
    </session-timeout>
</session-config>
```

Frameworks

- Projeto SPRING
  - Validando formulários no lado do servidor



Frameworks

- Projeto SPRING
  - Validando Formulários – Jeito Não Recomendado

```
@RequestMapping("adicionaTarefa")
public String adiciona(Tarefa tarefa) {

    if(tarefa.getDescricao() == null || tarefa.getDescricao().equals("")) {
        return "tarefa/formulario";
    }

    JdbcTarefaDao dao = new JdbcTarefaDao();
    dao.adiciona(tarefa);
    return "tarefa/adicionada";
}
```

Frameworks

- Projeto SPRING
  - Validando Formulários – Bean Validation
    - Validação da Camada de Modelo
      - Regras de validação por meio de anotações

```
 8   public class Tarefa {
 9
10       private Long id;
11
12       @NotEmpty(message="{tarefa.descricao.vazia}")
13       @Size(min=10, max = 100, message="{tarefa.descricao.limite}")
14       private String descricao;
15
16       private boolean finalizado;
17
18       @DateTimeFormat(pattern = "yyyy-MM-dd")
19       private Calendar dataFinalizacao;
20
21       //...
```

bean: Tarefa

Annotation	Example	Description
@NotEmpty	@NotEmpty private String firstName;	Checks if: object != null
@NotNull	@NotNull private Boolean isStudent;	Checks if: object != null and object size >= 1
@Size	@Size(min=1,max=20) private String lastName;	Checks for min and max size
@DateTimeFormat	@DateTimeFormat(pattern = "MM/dd/yyyy") private Date dob;	The custom patternto use to format the field. This is used for display purpose on the form too.
@NumberFormat	@NumberFormat(pattern="#.##") private Double weight;	The custom patternto format the Double value.
@Pattern	@Pattern(regexp="^[A-Za-z]*$") private String lastName;	Support to Regular expressions. In this example restricts to lower and upper case characters only.
@NotBlank	@NotEmpty private String firstName;	Checks the size after trimming the String. Do not use to check for null.

Frameworks

- Projeto SPRING
  - Validando Formulários – Bean Validation
    - ValidationMessages.properties
      - Deve ser salvo dentro da pasta src

```
tarefa.descricao.vazia  = A Descrição deve ser preenchida!
tarefa.descricao.limite = A Descrição deve conter [{min}-{max}] caracteres
```

ValidationMessages.properties

Frameworks

- Projeto SPRING
  - Validando Formulários – Bean Validation
    - Validação da Camada de Controle
      - *@Valid*
      - *BindingResult*

ConstraintViolationException

```
@RequestMapping("/adicionaTarefa")
public String adiciona(@Valid Tarefa tarefa, BindingResult result){

    if(result.hasErrors()){
        return "tarefa/formularioAdicionaTarefa";
    }

    dao.adicionaTarefa(tarefa);
    return "tarefa/tarefa-adicionada";
}
```

TarefaController

Frameworks

- Projeto SPRING
  - Validando Formulários – Bean Validation
    - Validação da Camada de Visão : **form:errors**

```
5   <%@taglib uri="http://www.springframework.org/tags/form" prefix="form" %>
6   <html>
7       <body>
8           <h1>Formulário de Adição de Tarefas</h1>
9           <form action="<c:url value='/adicionaTarefa'/>" method="post">
10              <fmt:message key="tarefa.descricao"/>
11              <br>
12              <textarea name="descricao" rows="5" cols="100" ></textarea>
13              <br>
14              <form:errors path="tarefa.descricao" cssStyle="color:red"/>
15              <br>
16              <input type="submit" value="Adicionar">
17          </form>
18      </body>
19  </html>
```

Frameworks

- Projeto SPRING
  - Validando Formulários