



MAPA – Material de Avaliação Prática da Aprendizagem

Nome: EDGARD CEZAR DE SOUZA MORAES	R.A 23048022-5
Curso: SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS	
Disciplina: IMERSÃO PROFISSIONAL: FÁBRICA DE SOFTWARE	

Instruções para Realização da Atividade

1. Revise seu arquivo antes do envio. Certifique-se de que é o arquivo correto, formato correto, se contempla todas as demandas da atividade etc.
2. Após o envio não serão permitidas alterações.
3. Durante a disciplina, procure sanar suas dúvidas pontuais em relação ao conteúdo relacionado à atividade. Porém, não são permitidas correções parciais, ou seja, enviar para que o professor possa fazer uma avaliação prévia e retornar para que o aluno possa ajustar e enviar novamente. Isso não é permitido, pois descaracteriza o processo de avaliação.
4. Ao utilizar quaisquer materiais de pesquisa referencie conforme as normas da ABNT;

Em caso de dúvidas, entre em contato com seu Professor Mediador.

Bons estudos!



AGORA, É COM VOCÊ!

Sistema Web - Biblioteca Universidade Cesumar

A Biblioteca da Universidade Cesumar está passando por um processo de modernização de seus serviços. A direção da instituição identificou a necessidade de um sistema web simples, funcional e com interface amigável, que permita aos funcionários realizarem o cadastro, a listagem e a exclusão de livros no acervo interno. Atualmente, o controle é feito em planilhas manuais, o que aumenta os riscos de inconsistência e dificulta o acesso remoto às informações. Para resolver esse problema, a universidade contratou uma equipe júnior de desenvolvedores, representada por você, para projetar e implementar um sistema web em Java que atenda a essas demandas, com base nos seguintes critérios:

Objetivo

Desenvolver um sistema web em Java utilizando Servlets, JSP e JSF, aplicando os conceitos de orientação a objetos e o padrão MVC.

Requisitos Funcionais

O sistema deve oferecer:

1. Cadastro de livros, com os seguintes dados:

Título

Autor

Ano de publicação

ISBN

2. Listagem de livros cadastrados, exibindo as informações de forma clara e organizada.

3. Exclusão de livros da lista (por ISBN ou ID).

Requisitos Técnicos:

Utilize Servlets para controlar as requisições e processar a lógica do sistema.

Use JSP e/ou JSF para a construção da interface do usuário.

Estruture o projeto com base no padrão MVC (*Model-View-Controller*).

O código deve ser autoral, ou seja, desenvolvido do zero.

Não utilize construtores visuais automáticos ou frameworks externos além do escopo da atividade.

Trate erros comuns, como campos vazios ou entradas inválidas de ISBN.

Checklist de Entrega — Sistema Web com Servlet, JSP e JSF

Funcionalidades:

Cadastro de livros com Título, Autor, Ano e ISBN.



Listagem dos livros cadastrados.

Exclusão de livros.

Interfaces web desenvolvidas com JSP e/ou JSF.

Tratamento de entradas inválidas (ex: campos vazios, ISBN inválido).

Organização do Projeto:

Projeto estruturado em pacotes: *model*, *controller*, *view*.

Separação clara entre lógica e apresentação.

Código comentado e de fácil leitura.

Uso adequado de Servlets para controle das ações.

Utilização de JSF em pelo menos uma das interfaces.

Como entregar a atividade:

A atividade deverá ser produzida em um arquivo do tipo TEXTO, conforme *TEMPLATE* anexado no MATERIAL DA DISCIPLINA, disponibilizado no Studeo, e DEVE ser entregue com a extensão (.PDF). Depois, deve ser anexado no ambiente da Atividade no STUDEO.

Obs.: Inserir o print com o código e a execução do projeto.

Dicas para realizar a atividade:

1. Durante as aulas, o professor fornecerá dicas que podem ser utilizadas para a confecção das suas atividades. Assim, é de suma importância participar das aulas ao vivo ou assisti-las posteriormente.

2. Assista às aulas conceituais da disciplina.

Orientações:

Plágios e cópias indevidas serão penalizados com descontos na nota, podendo chegar a zero. Não são permitidas correções parciais no decorrer do módulo, pois a interpretação da atividade também faz parte da avaliação.

Atenção ao prazo de entrega da atividade. Sugerimos que envie sua atividade antes do prazo final para evitar transtornos e lentidão nos servidores. Evite o envio de atividade em cima do prazo.

Boa atividade!

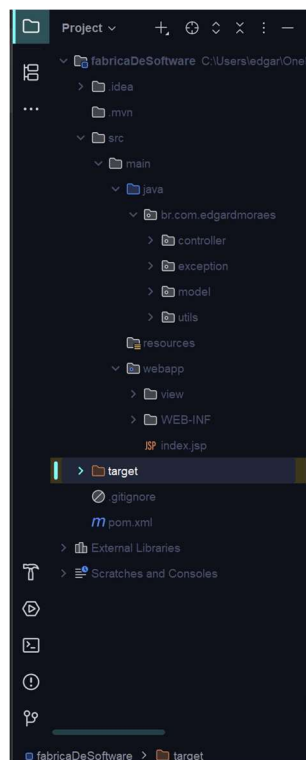


Coloque sua resposta no quadro abaixo.

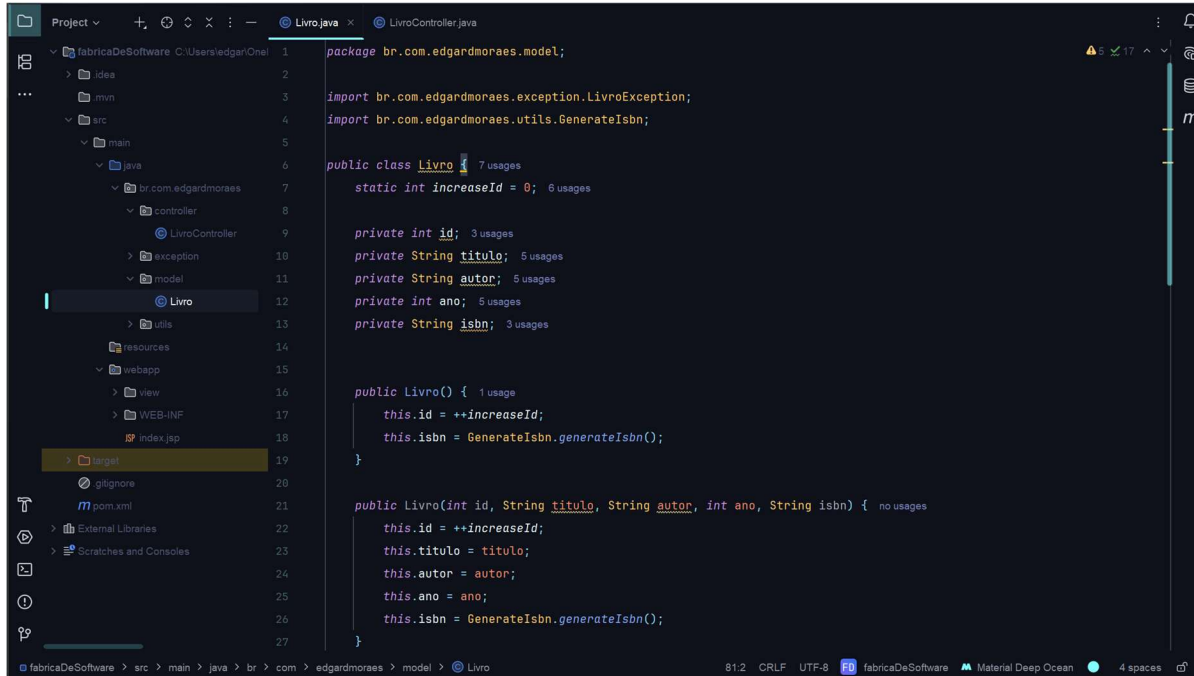
Olá Professor!

Abaixo segue meu projeto do Mapa, espero que aprecie.

Estrutura do projeto no padrão MVC, separando a parte lógica da visual.

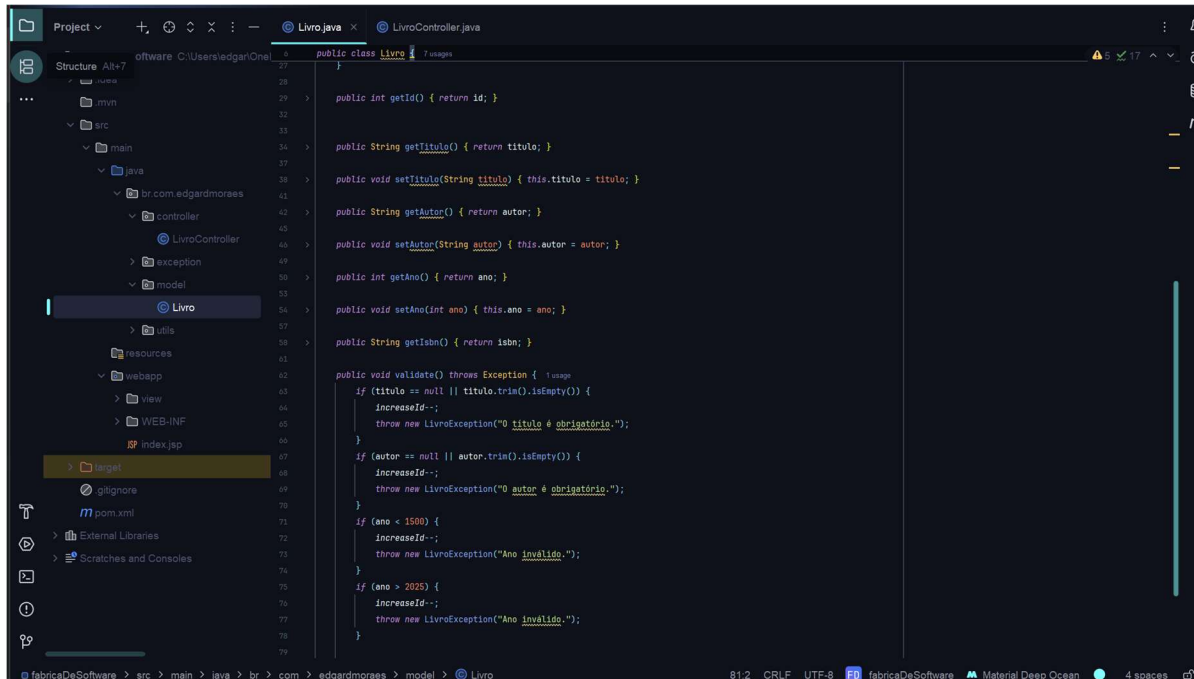


Abaixo segue a classe Livro que é o objeto do projeto:



```
1 package br.com.edgardmoraes.model;
2
3 import br.com.edgardmoraes.exception.LivroException;
4 import br.com.edgardmoraes.utils.GenerateIsbn;
5
6 public class Livro {
7     static int increaseId = 0;
8
9     private int id;
10    private String titulo;
11    private String autor;
12    private int ano;
13    private String isbn;
14
15    public Livro() {
16        this.id = ++increaseId;
17        this.isbn = GenerateIsbn.generateIsbn();
18    }
19
20    public Livro(int id, String titulo, String autor, int ano, String isbn) {
21        this.id = ++increaseId;
22        this.titulo = titulo;
23        this.autor = autor;
24        this.ano = ano;
25        this.isbn = GenerateIsbn.generateIsbn();
26    }
27 }
```

Abaixo seguem os getter e setter e coloquei também algumas validações



```
27 public class Livro {
28
29     public int getId() { return id; }
30
31     public String getTitulo() { return titulo; }
32
33     public void setTitulo(String titulo) { this.titulo = titulo; }
34
35     public String getAutor() { return autor; }
36
37     public void setAutor(String autor) { this.autor = autor; }
38
39     public int getAno() { return ano; }
40
41     public void setAno(int ano) { this.ano = ano; }
42
43     public String getIsbn() { return isbn; }
44
45     public void validate() throws Exception {
46         if (titulo == null || titulo.trim().isEmpty()) {
47             increaseId--;
48             throw new LivroException("O titulo é obrigatório.");
49         }
50         if (autor == null || autor.trim().isEmpty()) {
51             increaseId--;
52             throw new LivroException("O autor é obrigatório.");
53         }
54         if (ano < 1900) {
55             increaseId--;
56             throw new LivroException("Ano inválido.");
57         }
58         if (ano > 2025) {
59             increaseId--;
60             throw new LivroException("Ano inválido.");
61         }
62     }
63 }
```

Abaixo segue a classe controller que é a responsável por controlar o fluxo de dados com a interface View do projeto, contém também algumas regras de negócios.

```

package br.com.edgardmoraes.controller;

import br.com.edgardmoraes.exception.LivroException;
import br.com.edgardmoraes.model.Livro;
import br.com.edgardmoraes.utils.GenerateBookPage;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

@WebServlet(name = "LivroController", value = "/livros")
public class LivroController extends HttpServlet {

    private static final long serialVersionUID = 1L;

    private List<Livro> livros;

    public LivroController() { super(); }

    @Override
    public void init() { livros = new ArrayList<>(); }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        String bookPages = GenerateBookPage.generatePage(request, livros);
        request.setAttribute("bookPages", bookPages);
        request.getRequestDispatcher("view/index.jsp").forward(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        String action = request.getParameter("action");

        if ("delete".equals(action)) {
            String id = request.getParameter("id");
            if (id != null && !id.isEmpty()) {
                int idParse = Integer.parseInt(id);
                livros.removeIf(livro -> livro.getId() == idParse);
                response.sendRedirect(request.getContextPath() + "/livros");
            }
        }

        try {
            Livro livro = new Livro();
            livro.setTitulo(request.getParameter("titulo"));
            livro.setAutor(request.getParameter("autor"));
            livro.setAno(Integer.parseInt(request.getParameter("ano")));
            livro.validate();
            livros.add(livro);
            response.sendRedirect(request.getContextPath() + "/livros");
        } catch (NumberFormatException e) {
            request.setAttribute("mensagemErro", "Ano deve ser um número.");
            request.setAttribute("bookPages", GenerateBookPage.generatePage(request, livros));
            request.getRequestDispatcher("view/index.jsp").forward(request, response);
        } catch (LivroException e) {
            request.setAttribute("mensagemErro", e.getMessage());
            request.setAttribute("bookPages", GenerateBookPage.generatePage(request, livros));
            request.getRequestDispatcher("view/index.jsp").forward(request, response);
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }
}

```

```

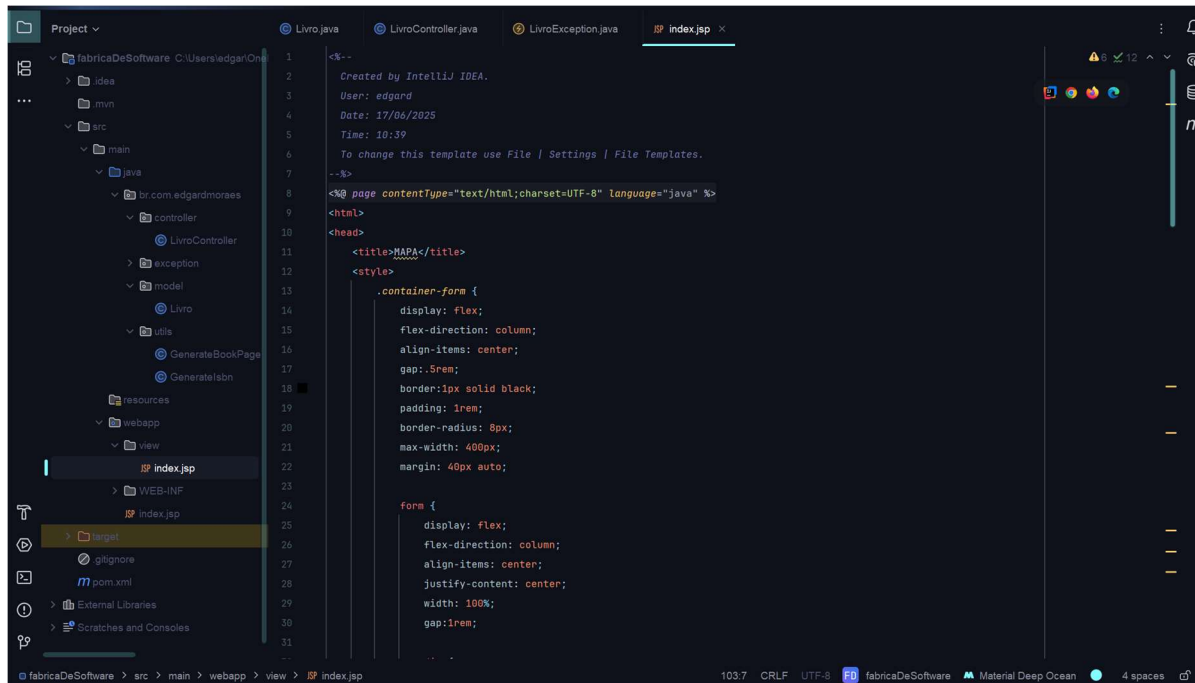
        String action = request.getParameter("action");

        if ("delete".equals(action)) {
            String id = request.getParameter("id");
            if (id != null && !id.isEmpty()) {
                int idParse = Integer.parseInt(id);
                livros.removeIf(livro -> livro.getId() == idParse);
                response.sendRedirect(request.getContextPath() + "/livros");
            }
        }

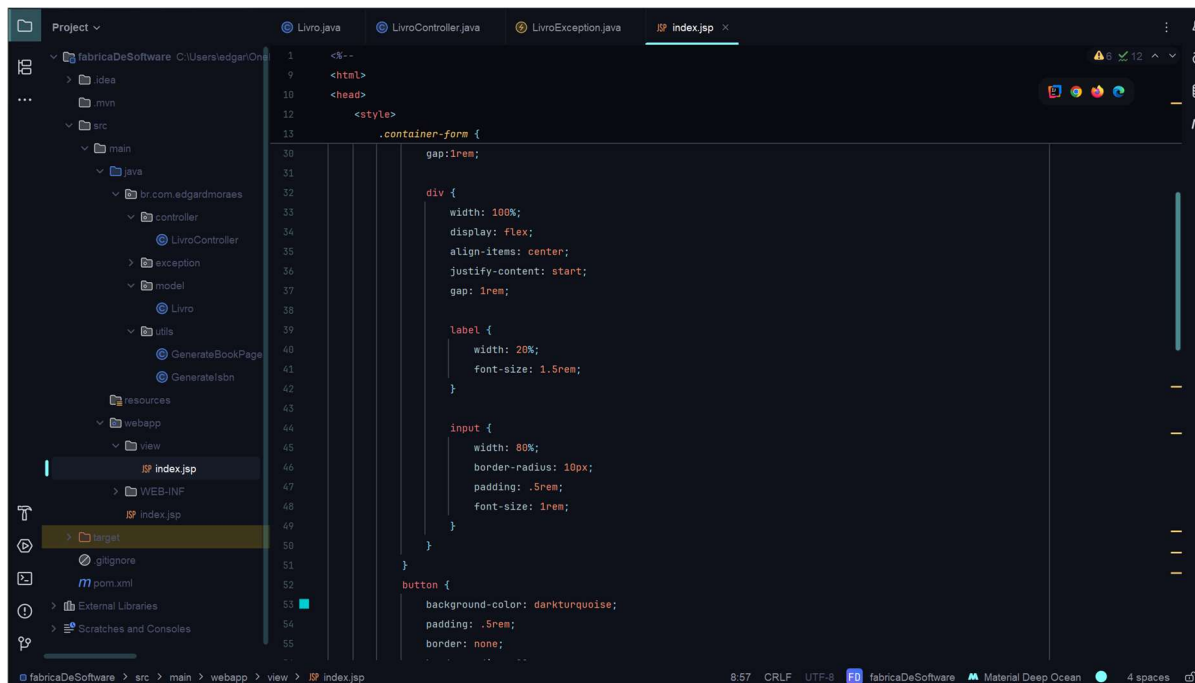
        try {
            Livro livro = new Livro();
            livro.setTitulo(request.getParameter("titulo"));
            livro.setAutor(request.getParameter("autor"));
            livro.setAno(Integer.parseInt(request.getParameter("ano")));
            livro.validate();
            livros.add(livro);
            response.sendRedirect(request.getContextPath() + "/livros");
        } catch (NumberFormatException e) {
            request.setAttribute("mensagemErro", "Ano deve ser um número.");
            request.setAttribute("bookPages", GenerateBookPage.generatePage(request, livros));
            request.getRequestDispatcher("view/index.jsp").forward(request, response);
        } catch (LivroException e) {
            request.setAttribute("mensagemErro", e.getMessage());
            request.setAttribute("bookPages", GenerateBookPage.generatePage(request, livros));
            request.getRequestDispatcher("view/index.jsp").forward(request, response);
        } catch (Exception e) {
            throw new RuntimeException(e);
        }
    }
}

```

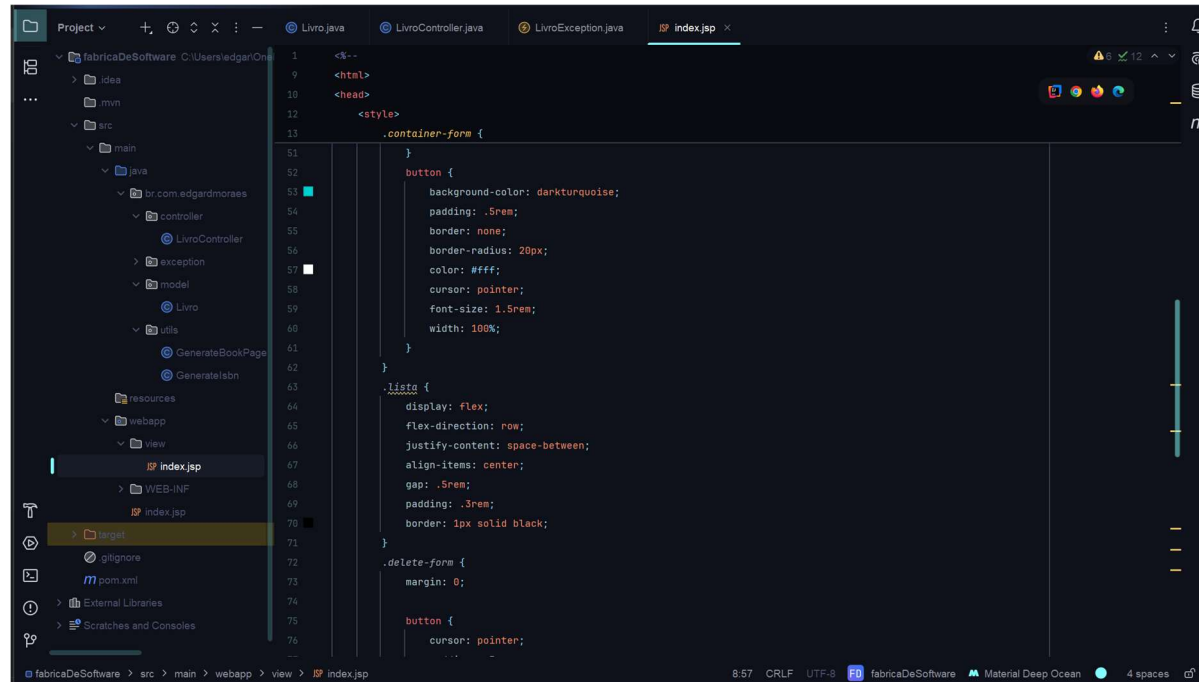
Abaixo segue a classe de visualização junto com o style.



```
1 <!--
2 Created by IntelliJ IDEA.
3 User: edgard
4 Date: 17/06/2025
5 Time: 10:39
6 To change this template use File | Settings | File Templates.
7 -->
8 <%@ page contentType="text/html; charset=UTF-8" language="java" %>
9 <html>
10 <head>
11 <title>MAPA</title>
12 <style>
13     .container-form {
14         display: flex;
15         flex-direction: column;
16         align-items: center;
17         gap: 5rem;
18         border: 1px solid black;
19         padding: 1rem;
20         border-radius: 8px;
21         max-width: 400px;
22         margin: 40px auto;
23     }
24
25     form {
26         display: flex;
27         flex-direction: column;
28         align-items: center;
29         justify-content: center;
30         width: 100%;
31         gap: 1rem;
32     }
```



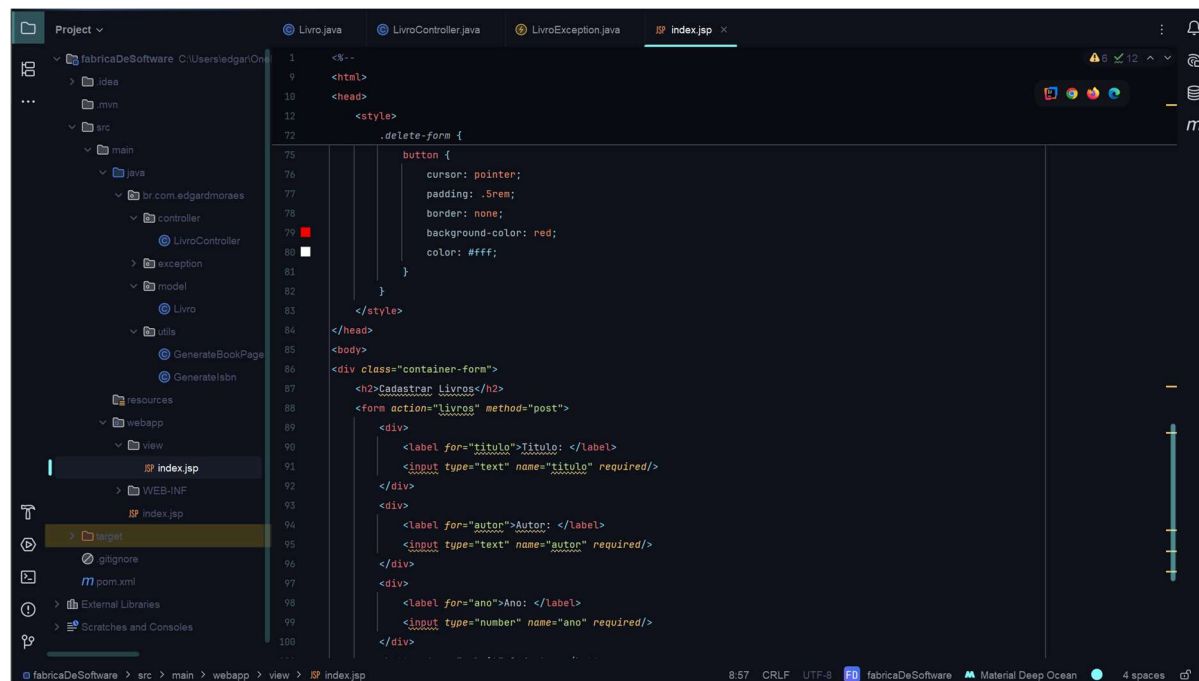
```
1 <!--
2 <html>
3 <head>
4 <style>
5     .container-form {
6         gap: 1rem;
7     }
8
9     div {
10         width: 100%;
11         display: flex;
12         align-items: center;
13         justify-content: start;
14         gap: 1rem;
15     }
16
17     label {
18         width: 20%;
19         font-size: 1.5rem;
20     }
21
22     input {
23         width: 80%;
24         border-radius: 10px;
25         padding: .5rem;
26         font-size: 1rem;
27     }
28 }
29
30 button {
31     background-color: darkturquoise;
32     padding: .5rem;
33     border: none;
34 }
```



The screenshot shows an IDE with the following structure:

- Project: fabricaDeSoftware - C:\Users\edgar\One...
- src/main/webapp/view/index.jsp

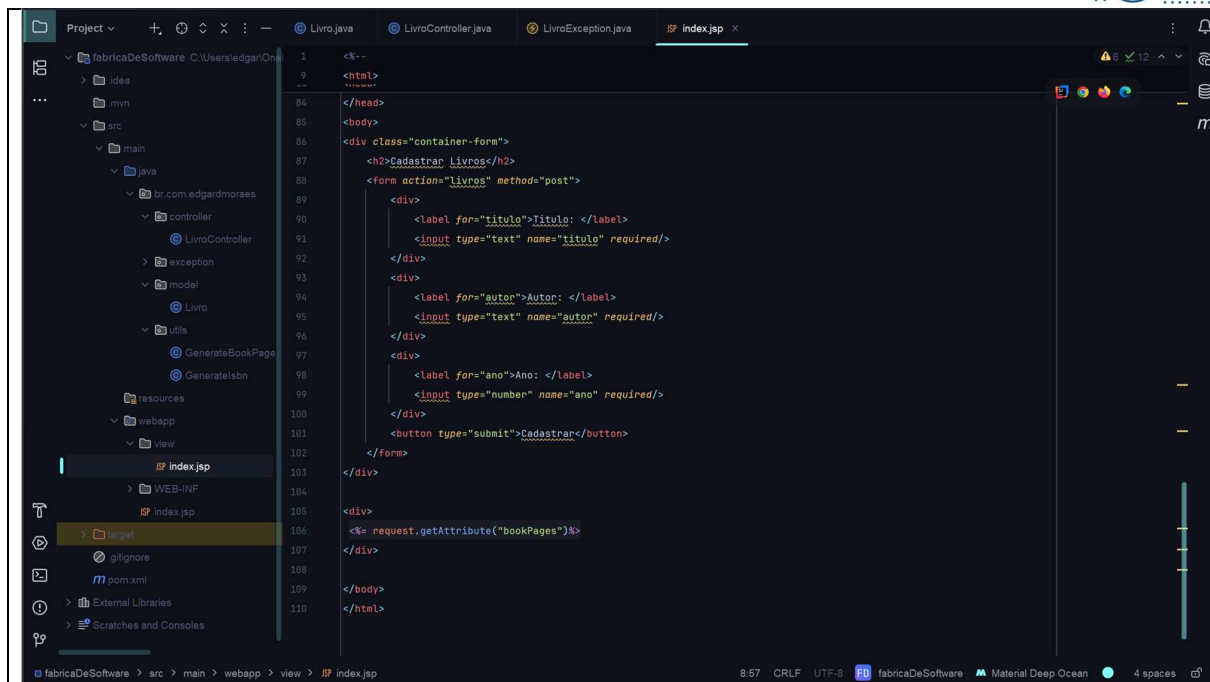
```
1 <!--
9 <html>
10 <head>
12 <style>
13 .container-form {
51 }
52 button {
53 background-color: darkturquoise;
54 padding: .5rem;
55 border: none;
56 border-radius: 20px;
57 color: #fff;
58 cursor: pointer;
59 font-size: 1.5rem;
60 width: 100%;
61 }
62 }
63 .lista {
64 display: flex;
65 flex-direction: row;
66 justify-content: space-between;
67 align-items: center;
68 gap: .5rem;
69 padding: .3rem;
70 border: 1px solid black;
71 }
72 .delete-form {
73 margin: 0;
74 }
75 button {
76 cursor: pointer;
77 ...
```



The screenshot shows the same IDE with the following structure:

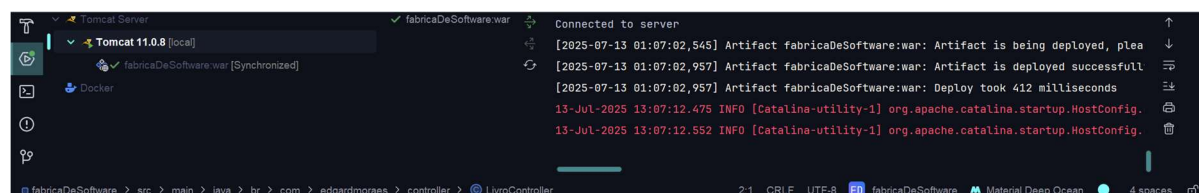
- Project: fabricaDeSoftware - C:\Users\edgar\One...
- src/main/webapp/view/index.jsp

```
1 <!--
9 <html>
10 <head>
12 <style>
72 .delete-form {
75 button {
76 cursor: pointer;
77 padding: .5rem;
78 border: none;
79 background-color: red;
80 color: #fff;
81 }
82 }
83 </style>
84 </head>
85 <body>
86 <div class="container-form">
87 <h2>Cadastrar Livros</h2>
88 <form action="livros" method="post">
89 <div>
90 <label for="titulo">Titulo: </label>
91 <input type="text" name="titulo" required/>
92 </div>
93 <div>
94 <label for="autor">Autor: </label>
95 <input type="text" name="autor" required/>
96 </div>
97 <div>
98 <label for="ano">Ano: </label>
99 <input type="number" name="ano" required/>
100 </div>
```

```
1 <!--
9 <html>
84 </head>
85 <body>
86 <div class="container-form">
87   <h2>Cadastrar Livros</h2>
88   <form action="/livros" method="post">
89     <div>
90       <label for="titulo">Titulo: </label>
91       <input type="text" name="titulo" required/>
92     </div>
93     <div>
94       <label for="autor">Autor: </label>
95       <input type="text" name="autor" required/>
96     </div>
97     <div>
98       <label for="ano">Ano: </label>
99       <input type="number" name="ano" required/>
100     </div>
101     <button type="submit">Cadastrar</button>
102   </form>
103 </div>
104
105 <div>
106   <%= request.getAttribute("bookPages")%>
107 </div>
108
109 </body>
110 </html>
```

Abaixo segue os prints do servidor rodando com o projeto



```
fabricaDeSoftware:war
Connected to server
[2025-07-13 01:07:02,545] Artifact fabricaDeSoftware:war: Artifact is being deployed, please
[2025-07-13 01:07:02,957] Artifact fabricaDeSoftware:war: Artifact is deployed successfully
[2025-07-13 01:07:02,957] Artifact fabricaDeSoftware:war: Deploy took 412 milliseconds
13-Jul-2025 13:07:12.475 INFO [Catalina-utility-1] org.apache.catalina.startup.HostConfig.
13-Jul-2025 13:07:12.552 INFO [Catalina-utility-1] org.apache.catalina.startup.HostConfig.
```



Cadastrar Livros

Titulo:

Autor:

Ano:

Cadastrar

Livros cadastrados



Cadastrar Livros

Título:

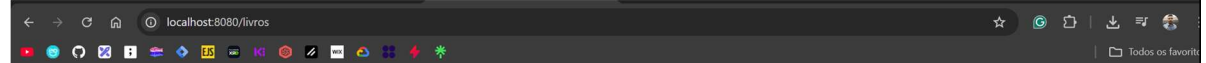
Autor:

Ano:

Livros cadastrados

Título: CREIA NO INVISIVEL	Autor: NATAN RUFINO	Ano: 2012	ISBN: 123122237756	id: 1	<input type="button" value="EXCLUIR"/>
----------------------------	---------------------	-----------	--------------------	-------	--

Abaixo foram excluídos alguns livros clicando no botão excluir, a notar se pelo ID 7



Cadastrar Livros

Título:

Autor:

Ano:

Livros cadastrados

Título: DESIGN PATTERNS JAVA	Autor: JIM WEBBER	Ano: 2014	ISBN: 123867391370	id: 7	<input type="button" value="EXCLUIR"/>
------------------------------	-------------------	-----------	--------------------	-------	--