

UNIVERSIDADE DE MOGI DAS CRUZES
Ensino Superior com Habilitação Profissional de Analista de
Desenvolvimento de Sistemas

Guilherme Napolitano
Gustavo Figueiredo Oliveira
João Pedro Ventura Rovaris

PROJETO FINAL - M2

MOGI
2025

Guilherme Napolitano
Gustavo Figueiredo Oliveira
João Pedro Ventura Rovaris

PROJETO FINAL - M2

Pesquisa acadêmica apresentado ao
Curso Análise de Desenvolvimento de
Sistemas, orientado pelo Prof. Adilson
Lima da Silva.

MOGI
2025

AGRADECIMENTOS

Queremos agradecer primeiramente ao nosso orientador Adilson Lima da Silva pelo auxílio no desenvolvimento do projeto e pela paciência, apoio e conselhos.

Sumário

1. MER – Modelo Entidade-Relacionamento
2. MR – Modelo Relacional
3. Protótipos do Sistema
 - 3.1. Tela de Login (Desktop e Mobile)
 - 3.2. Tela Principal (Desktop e Mobile)
 - 3.3. Tela de Perfil (Desktop e Mobile)
 - 3.4. Tela de Listagem (Desktop e Mobile)
4. Código do Sistema
 - 4.1. Model
 - 4.1.1. Model Funcionário
 - 4.1.2. Model Ordem
 - 4.1.3. Model Peça
 - 4.1.4. Model Setor
 - 4.1.5. Model Usuário
 - 4.2. DAO
 - 4.2.1. Conexão Java
 - 4.2.2. DAO Funcionário
 - 4.2.3. DAO Ordem
 - 4.2.4. DAO Peça
 - 4.2.5. DAO Setor
 - 4.2.6. DAO Usuário
 - 4.3. Servlets
 - 4.3.1. Servlet Funcionário
 - 4.3.2. Servlet Login
 - 4.3.3. Servlet Logout
 - 4.3.4. Servlet Ordem
 - 4.3.5. Servlet Peça
 - 4.3.6. Servlet Setor
 - 4.3.7. Servlet Usuário
 - 4.4. JSP
 - 4.4.1. Dashboard
 - 4.4.2. Index

4.4.3. Login

4.5. JSPs Específicos

4.5.1. Usuário – Novo, Listar, Editar

4.5.2. Setor – Novo, Listar, Editar

4.5.3. Peça – Novo, Listar, Editar

4.5.4. Ordem – Novo, Listar, Editar

4.6. CSS

4.6.1. style.css

4.6.2. new.css

5. Operações no Sistema

5.1. Atualização de Dados

5.2. Exclusão de Dados

6. Criação das VIEWS

6.1. Relatório de vendas dos funcionários

6.2. Produtos vendidos

6.3. Relatório gerencial

6.4. Compras dos clientes

7. Criação de Índices para Otimização

7.1. Busca do ID do cliente no pedido

7.2. Busca pela data do pedido

7.3. Busca pelo funcionário responsável

8. Transação

9. Script do Banco de Dados

10. Evidências do Database

11. Evidências do Sistema

12. Link do GitHub

13. Referências

EPÍGRAFE

“Sábio é aquele que conhece os limites da própria ignorância.”

-Sócrates

Resumo

Este estudo descreve a criação de uma aplicação web em Java conectada a um banco de dados, centrada no registro de funcionários. A proposta foi elaborada para executar tarefas como inserção, remoção, modificação e busca de informações, garantindo a integridade e proteção dos dados. Optou-se pela linguagem Java devido à sua solidez e capacidade de expansão. A plataforma, acessível por meio de um navegador, conta com uma interface amigável e fácil de usar, o que acelera e fortalece as operações administrativas. O empreendimento destaca a importância do uso de ferramentas digitais para o gerenciamento empresarial, combinando conhecimentos conceituais e práticos em uma abordagem viável para variados ambientes de trabalho.

Palavras-chave: Sistema Web. Java. Banco de Dados. Cadastro de Colaboradores.

Abstract

This work presents the development of a Java web system integrated with a database, focusing on employee registration. The solution was designed to perform operations such as inclusion, deletion, updating, and querying of data, ensuring consistency and security of information. Java was selected for its robustness and scalability. The system, accessed via a browser, features a simple and intuitive interface, enabling greater agility and reliability in administrative processes. The project demonstrates the relevance of applying web technologies for corporate management, combining theoretical foundations and practical implementation in a model applicable to various organizational contexts.

Keywords: Web System. Java. Database. Employee Registration.

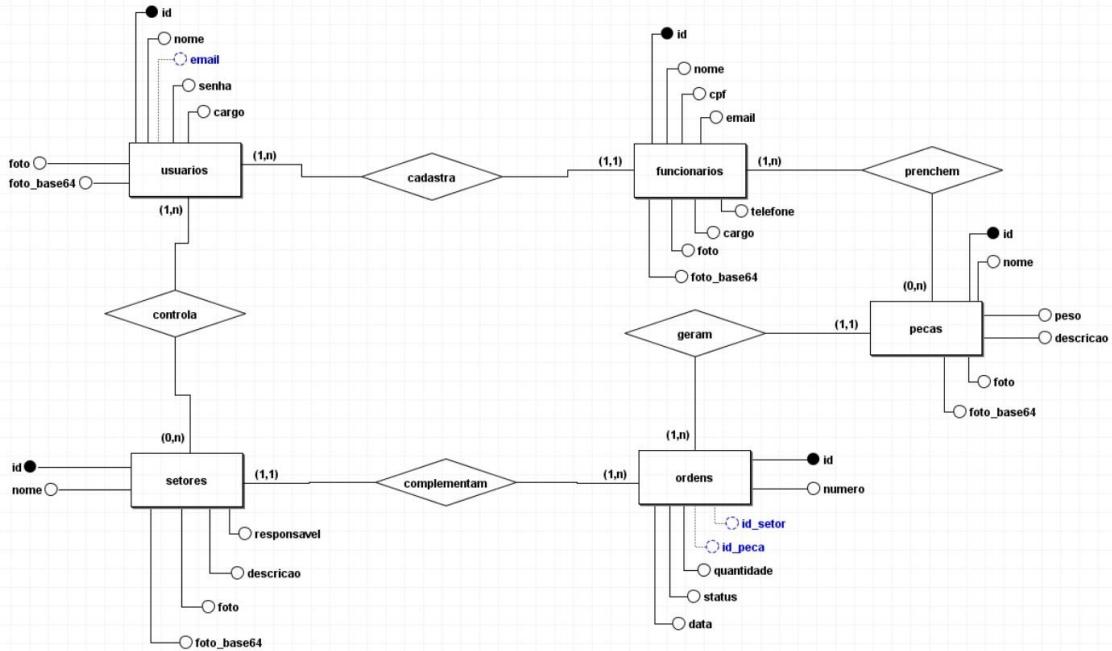
Sumário

| | | |
|------|--|----|
| 1. | MER | 4 |
| 2. | MR..... | 4 |
| 3. | Protótipos sistema | 5 |
| 4. | Código do sistema..... | 13 |
| 4.1. | Model..... | 13 |
| | | 14 |
| 5 | 18 | |
| 5.1. | Atualização de dados..... | 19 |
| 5.2. | Exclusão de dados..... | 22 |
| 6. | Criação das VIEWs..... | 23 |
| 6.1. | Relatório de vendas dos funcionários | 23 |
| 6.2. | Produtos vendidos | 24 |
| 6.3. | Relatório Gerencial | 24 |
| 6.4. | Compra dos clientes | 24 |

| | |
|--|----|
| 7. Criação de 3 índices para agilizar consultas frequentes | 25 |
| 7.1. Buscando o id do cliente no pedido | 25 |
| 7.2. Buscando pela data do pedido..... | 25 |
| 7.3. Buscando pelo funcionário que realizou o pedido | 25 |
| 8. Transação | 25 |

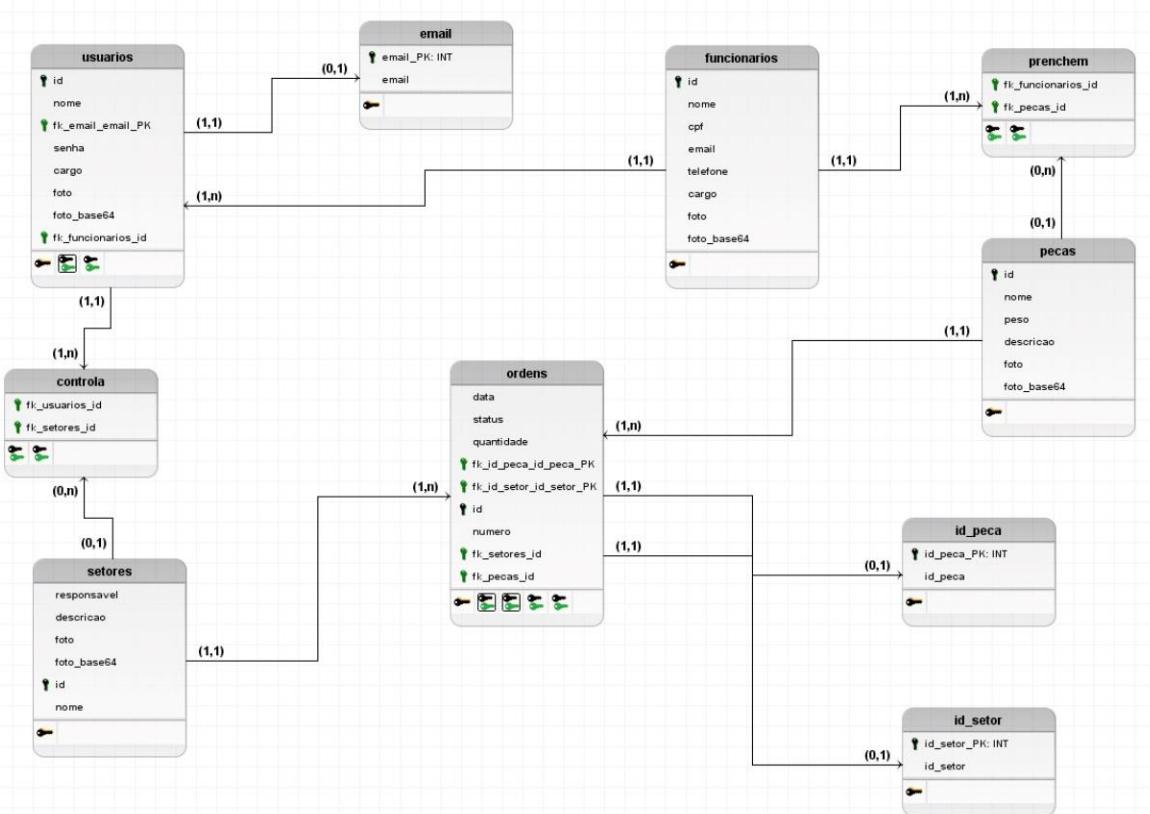
MER

Figura 1-Modelo de Entidade de Relacionamento



MR

Figura 2-Modelo Relacional



■ Protótipos sistema

Figura 3-Tela login desktop

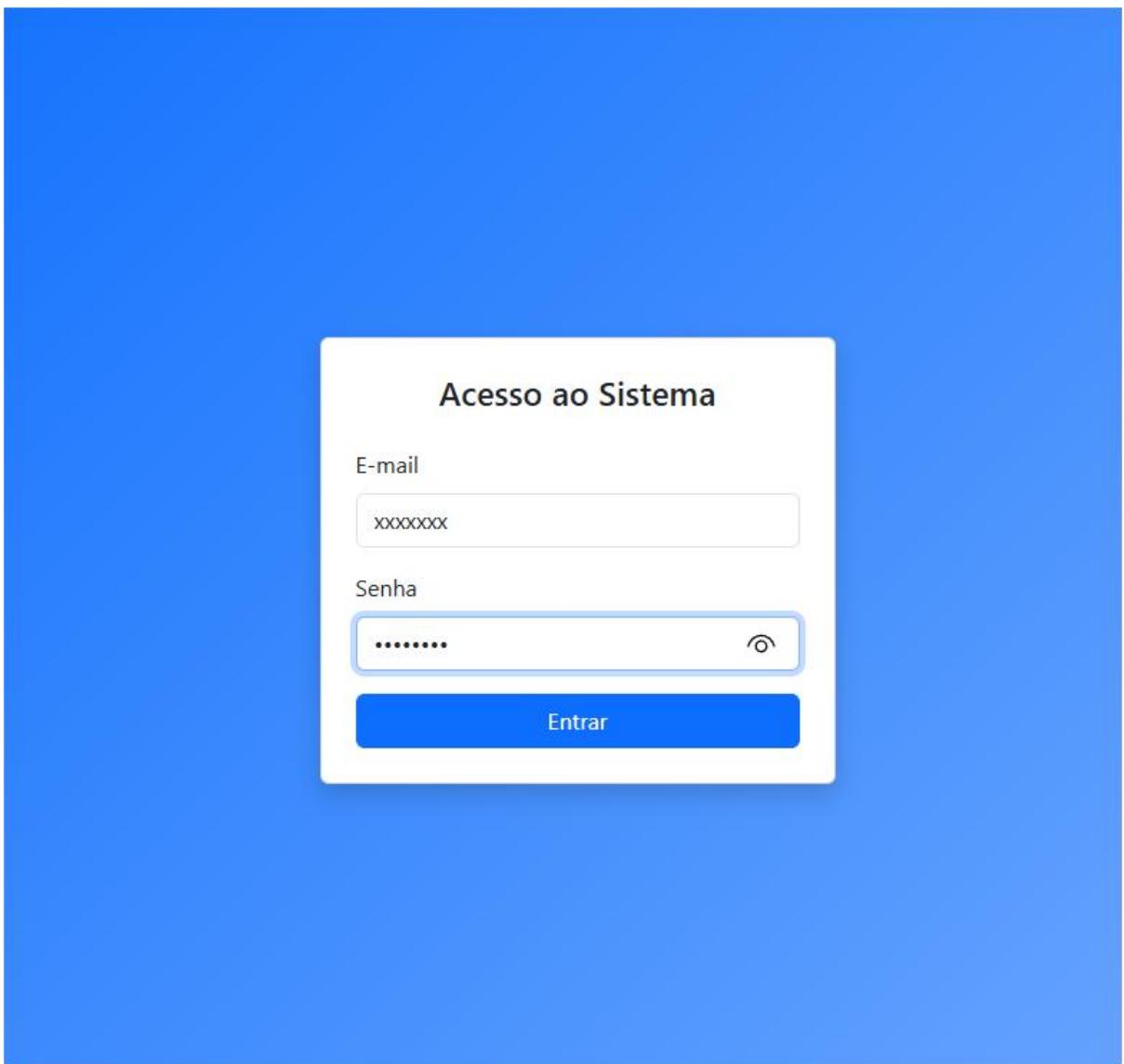


Figura 4-Tela login mobile

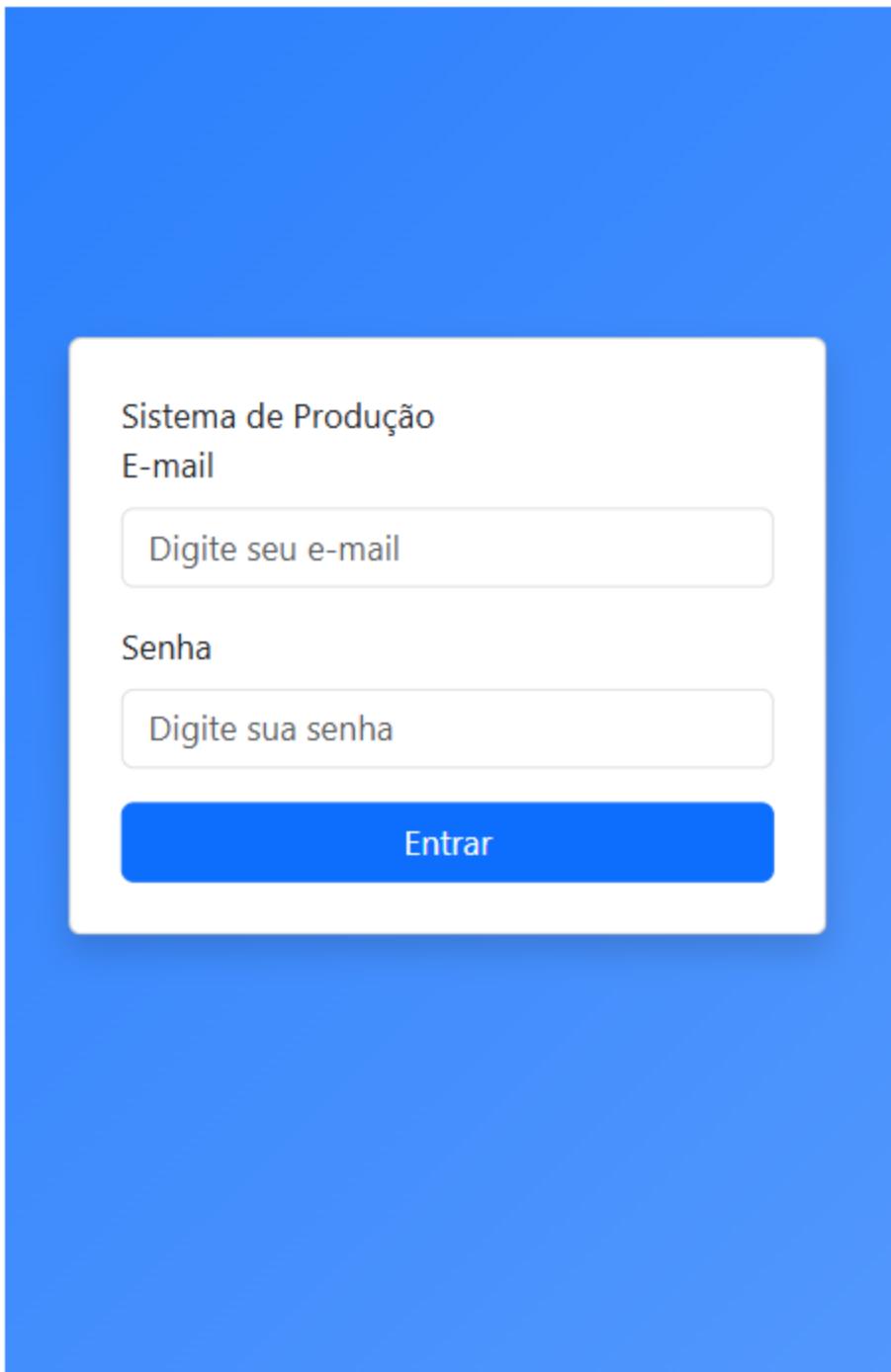


Figura 5-Tela principal desktop



Figura 6-Tela principal mobile

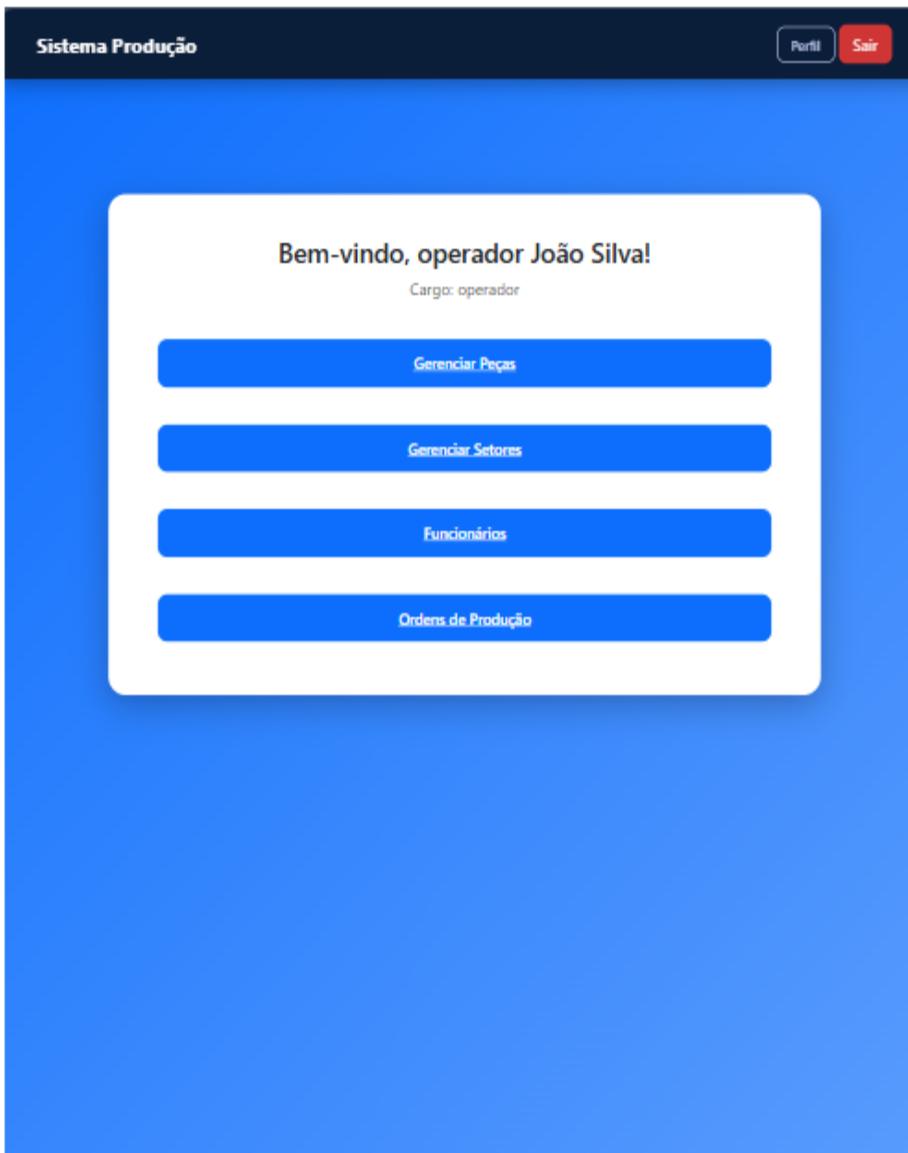


Figura 7-Tela perfil desktop

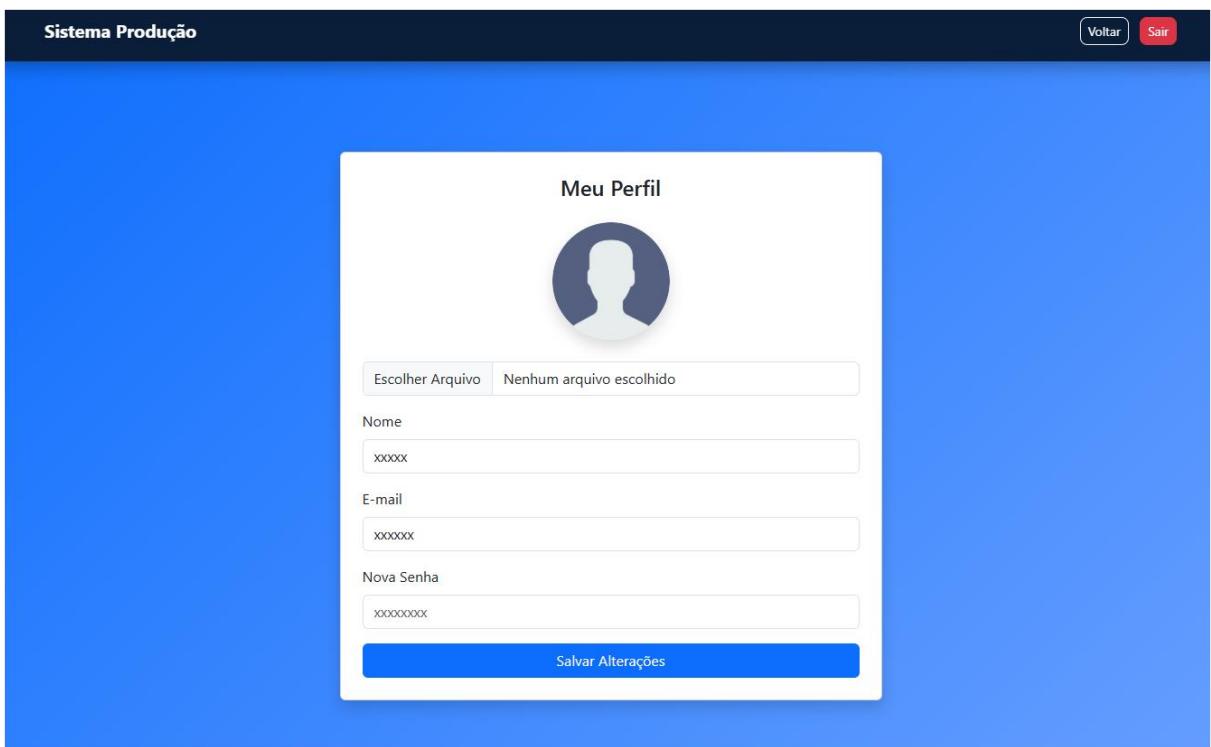


Figura 8-Tela perfil mobile

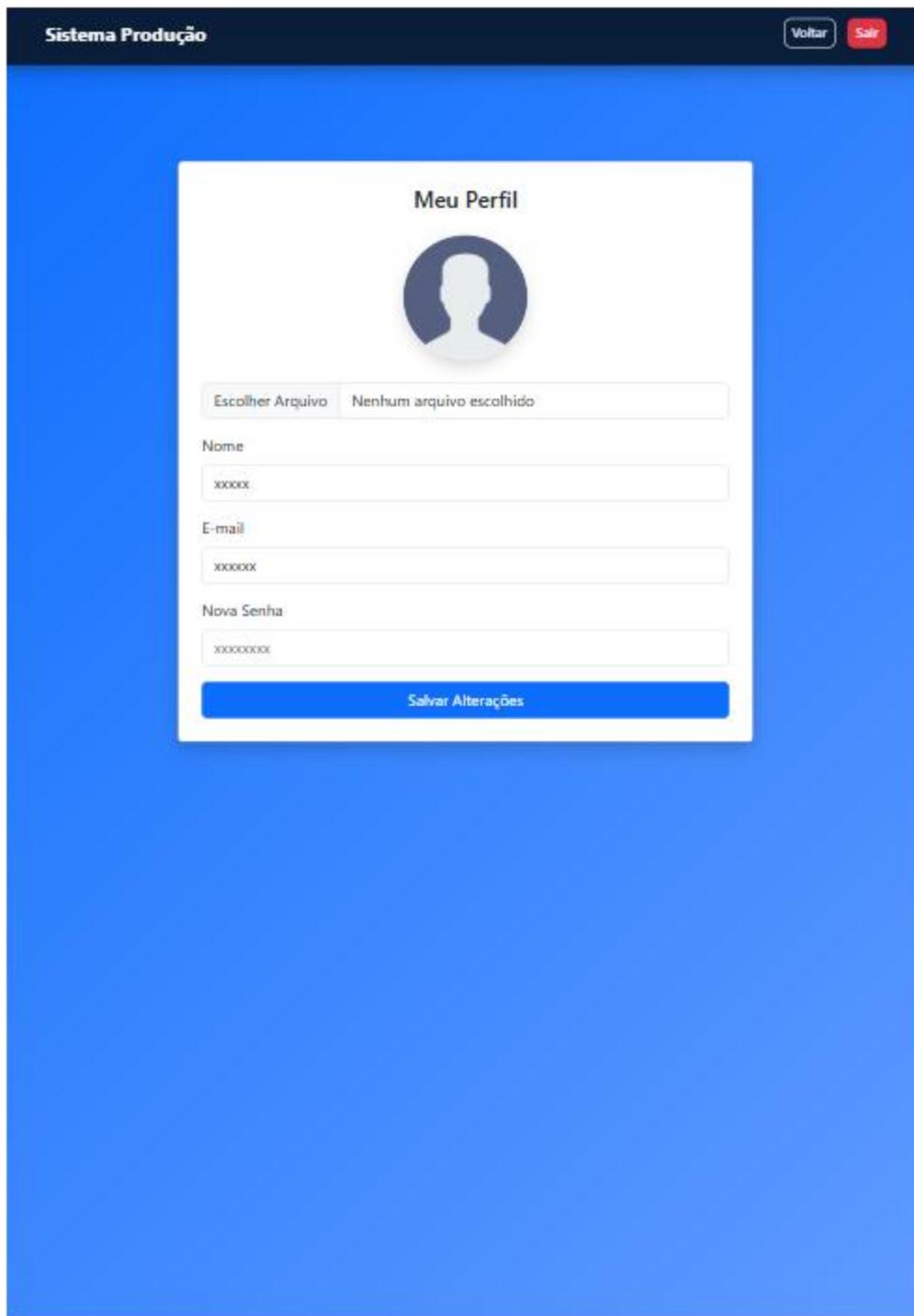


Figura 9-Tela de listagem desktop

| Lista de Peças | | | | | |
|----------------|----|------------|------|----------------|--|
| Foto | ID | Nome | Peso | Descrição | Ações |
| | 1 | xxxxxxxxxx | 0.0 | xxxxxxxxxxxx | <button>Editar</button> <button>Excluir</button> |
| | 2 | xxxxxxxxxx | 0.0 | xxxxxxxxxxxxxx | <button>Editar</button> <button>Excluir</button> |

Figura 10-Tela de listagem mobile

The image shows a mobile application interface for managing parts. At the top, there is a blue header bar with the title "Lista de Peças" and a green button labeled "+ Nova Peça". Below the header is a table with the following data:

| Foto | ID | Nome | Peso | Descrição | Ações |
|------|----|------------|------|--------------|--|
| | 1 | xxxxxxxxxx | 0.0 | xxxxxxxxxxxx | <button>Editar</button> <button>Excluir</button> |
| | 2 | xxxxxxxxxx | 0.0 | xxxxxxxxxxxx | <button>Editar</button> <button>Excluir</button> |

Código do sistema.

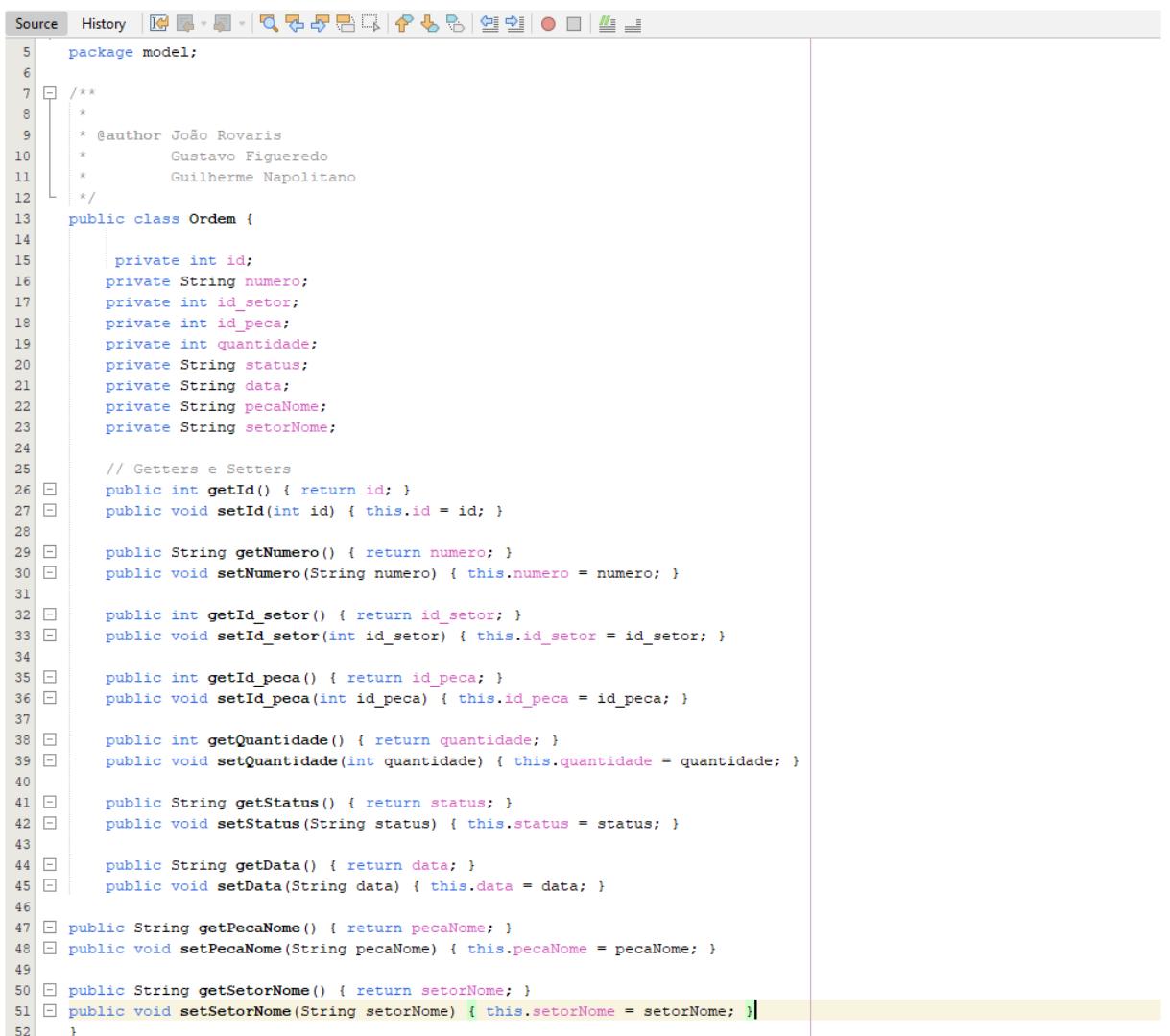
4.1. Model.

Figura 11-Model Funcionário

```

1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package model;
6   /**
7   *
8   * @author João Rovaris
9   *         Gustavo Figueiredo
10  *        Guilherme Napolitano
11  */
12
13  public class Funcionario {
14
15      private int id;
16      private String nome;
17      private String cpf;
18      private String email;
19      private String telefone;
20      private String cargo;
21      private String foto;
22      private String foto_base64;
23
24      // GETTERS E SETTERS
25      public int getId() { return id; }
26      public void setId(int id) { this.id = id; }
27
28      public String getNome() { return nome; }
29      public void setNome(String nome) { this.nome = nome; }
30
31      public String getCpf() { return cpf; }
32      public void setCpf(String cpf) { this.cpf = cpf; }
33
34      public String getEmail() { return email; }
35      public void setEmail(String email) { this.email = email; }
36
37      public String getTelefone() { return telefone; }
38      public void setTelefone(String telefone) { this.telefone = telefone; }
39
40      public String getCargo() { return cargo; }
41      public void setCargo(String cargo) { this.cargo = cargo; }
42
43      public String getFoto() { return foto; }
44      public void setFoto(String foto) { this.foto = foto; }
45
46      public String getFoto_base64() { return foto_base64; }
47      public void setFoto_base64(String foto_base64) { this.foto_base64 = foto_base64; }
48  }

```

Figura 12-Model Ordem


The screenshot shows a Java code editor window with the following code:

```

5 package model;
6
7 /**
8 * @author João Rovaris
9 *         Gustavo Figueiredo
10 *        Guilherme Napolitano
11 */
12 public class Ordem {
13
14     private int id;
15     private String numero;
16     private int id_setor;
17     private int id_peca;
18     private int quantidade;
19     private String status;
20     private String data;
21     private String pecaNome;
22     private String setorNome;
23
24     // Getters e Setters
25     public int getId() { return id; }
26     public void setId(int id) { this.id = id; }
27
28     public String getNumero() { return numero; }
29     public void setNumero(String numero) { this.numero = numero; }
30
31     public int getId_setor() { return id_setor; }
32     public void setId_setor(int id_setor) { this.id_setor = id_setor; }
33
34     public int getId_peca() { return id_peca; }
35     public void setId_peca(int id_peca) { this.id_peca = id_peca; }
36
37     public int getQuantidade() { return quantidade; }
38     public void setQuantidade(int quantidade) { this.quantidade = quantidade; }
39
40     public String getStatus() { return status; }
41     public void setStatus(String status) { this.status = status; }
42
43     public String getData() { return data; }
44     public void setData(String data) { this.data = data; }
45
46     public String getPecaNome() { return pecaNome; }
47     public void setPecaNome(String pecaNome) { this.pecaNome = pecaNome; }
48
49     public String getSetorNome() { return setorNome; }
50     public void setSetorNome(String setorNome) { this.setorNome = setorNome; }
51
52 }

```

The code defines a class named `Ordem` with various fields and their corresponding getters and setters. The class is annotated with a multi-line comment at the top. The code editor interface is visible at the top, showing tabs like "Source" and "History". The code is displayed in a monospaced font.

Figura 13-Model Peca

```

1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package model;
6   /**
7   *
8   * @author João Rovaris
9   *         Gustavo Figueiredo
10  *        Guilherme Napolitano
11  */
12 public class Peca {
13
14     private int id;
15     private String nome;
16     private double peso;
17     private String descricao;
18     private String foto;
19     private String foto_base64; // campo novo
20
21     public int getId() {
22         return id;
23     }
24     public void setId(int id) {
25         this.id = id;
26     }
27
28     public String getNome() {
29         return nome;
30     }
31     public void setNome(String nome) {
32         this.nome = nome;
33     }
34
35     public double getPeso() {
36         return peso;
37     }
38     public void setPeso(double peso) {
39         this.peso = peso;
40     }
41
42     public String getDescricao() {
43         return descricao;
44     }
45     public void setDescricao(String descricao) {
46         this.descricao = descricao;
47     }
48     public String getFoto() { return foto; }

```

Figura 14-Model Peca pt 2

```

48     public String getFoto() { return foto; }
49     public void setFoto(String foto) { this.foto = foto; }
50
51     public String getFoto_base64() {
52         return foto_base64;
53     }
54     public void setFoto_base64(String foto_base64) {
55         this.foto_base64 = foto_base64;
56     }
57 }
58

```

Figura 15-Model Setor

```

1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package model;
6  /**
7  *
8  * @author João Rovaris
9  *         Gustavo Figueiredo
10 *        Guilherme Napolitano
11 */
12 public class Setor {
13
14     private int id;
15     private String nome;
16     private String descricao;
17     private String responsavel;
18     private String foto;
19     private String foto_base64;
20
21     // GETTERS E SETTERS
22     public int getId() {
23         return id;
24     }
25     public void setId(int id) {
26         this.id = id;
27     }
28
29     public String getNome() {
30         return nome;
31     }
32     public void setNome(String nome) {
33         this.nome = nome;
34     }
35
36     public String getDescricao() {
37         return descricao;
38     }
39     public void setDescricao(String descricao) {
40         this.descricao = descricao;
41     }
42
43     public String getResponsavel() {
44         return responsavel;
45     }
46     public void setResponsavel(String responsavel) {
47         this.responsavel = responsavel;
48     }

```

Figura 16-Model Setor pt2

```

47         this.responsavel = responsavel;
48     }
49
50     public String getFoto() {
51         return foto;
52     }
53     public void setFoto(String foto) {
54         this.foto = foto;
55     }
56
57     public String getFoto_base64() {
58         return foto_base64;
59     }
60     public void setFoto_base64(String foto_base64) {
61         this.foto_base64 = foto_base64;
62     }

```

Figura 17-Model Usuário

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5  package model;
6  /**
7   *
8   * @author João Rovaris
9   *         Gustavo Figueiredo
10  *        Guilherme Napolitano
11  */
12 public class Usuario {
13
14     private int id;
15     private String nome;
16     private String email;
17     private String senha;
18     private String cargo;
19     private String foto;           // nome do arquivo salvo
20     private String foto_base64;   // imagem convertida para Base64
21
22     public Usuario() {}
23
24     public Usuario(String nome, String email, String senha, String cargo) {
25         this.nome = nome;
26         this.email = email;
27         this.senha = senha;
28         this.cargo = cargo;
29     }
30
31     // Getters e Setters
32     public int getId() {
33         return id;
34     }
35     public void setId(int id) {
36         this.id = id;
37     }
38
39     public String getNome() {
40         return nome;
41     }
42     public void setNome(String nome) {
43         this.nome = nome;
44     }
45
46     public String getEmail() {
47         return email;
48     }
```

Figura 18-Model Usuário pt2

```

48     }
49     public void setEmail(String email) {
50         this.email = email;
51     }
52
53     public String getSenha() {
54         return senha;
55     }
56     public void setSenha(String senha) {
57         this.senha = senha;
58     }
59
60     public String getCargo() {
61         return cargo;
62     }
63     public void setCargo(String cargo) {
64         this.cargo = cargo;
65     }
66

```

4.2. DAO

Figura 19-conexão Java

```

1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package dao;
6  /*
7  * João Pedro Rovaris
8  * Gustavo Figueiredo
9  * Guilherme Napolitano
10 */
11 import java.sql.Connection;
12 import java.sql.DriverManager;
13 import java.sql.SQLException;
14
15 public class Conexao {
16     private static final String URL = "jdbc:mysql://localhost:3306/sistema_producao";
17     private static final String USER = "root";
18     private static final String PASSWORD = "";
19
20     public static Connection getConexao() throws SQLException {
21         try {
22             Class.forName("com.mysql.cj.jdbc.Driver");
23             return DriverManager.getConnection(URL, USER, PASSWORD);
24         } catch (ClassNotFoundException e) {
25             throw new SQLException("Driver do MySQL não encontrado!", e);
26         }
27     }
28 }

```

Figura 20-Funcionário

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5  package dao;
6  /*
7   * João Pedro Rovaris
8   * Gustavo Figueiredo
9   * Guilherme Napolitano
10  */
11 import model.Funcioario;
12 import java.sql.*;
13 import java.util.*;
14
15 public class FuncionarioDAO {
16
17     private Connection con;
18
19     public FuncionarioDAO() {
20         try {
21             con = Conexao.getConexao();
22         } catch (SQLException e) {
23             e.printStackTrace();
24         }
25     }
26
27     // CADASTRAR
28     public void cadastrar(Funcioario f) {
29         try {
30             String sql = "INSERT INTO funcionarios (nome, cpf, email, telefone, cargo, foto, foto_base64) VALUES (?, ?, ?, ?, ?, ?, ?)";
31             PreparedStatement stmt = con.prepareStatement(sql);
32
33             stmt.setString(1, f.getNome());
34             stmt.setString(2, f.getCpf());
35             stmt.setString(3, f.getEmail());
36             stmt.setString(4, f.getTelefone());
37             stmt.setString(5, f.getCargo());
38             stmt.setString(6, f.getFoto());
39             stmt.setString(7, f.getFoto_base64());
40
41             stmt.executeUpdate();
42         } catch (SQLException e) {
43             e.printStackTrace();
44         }
45     }
46
47     // LISTAR
48     public List<Funcioario> listar() {
```

Figura 21-Funcionário pt2

```
48 |     public List<Funcionario> listar() {
49 |         List<Funcionario> lista = new ArrayList<>();
50 |
51 |         try {
52 |             String sql = "SELECT * FROM funcionarios";
53 |             PreparedStatement stmt = con.prepareStatement(sql);
54 |             ResultSet rs = stmt.executeQuery();
55 |
56 |             while (rs.next()) {
57 |                 Funcionario f = new Funcionario();
58 |
59 |                 f.setId(rs.getInt("id"));
60 |                 f.setNome(rs.getString("nome"));
61 |                 f.setCpf(rs.getString("cpf"));
62 |                 f.setEmail(rs.getString("email"));
63 |                 f.setTelefone(rs.getString("telefone"));
64 |                 f.setCargo(rs.getString("cargo"));
65 |                 f.setFoto(rs.getString("foto"));
66 |                 f.setFoto_base64(rs.getString("foto_base64"));
67 |
68 |                 lista.add(f);
69 |             }
70 |         } catch (SQLException e) {
71 |             e.printStackTrace();
72 |         }
73 |
74 |         return lista;
75 |     }
76 |
77 // ATUALIZAR
78 |     public void atualizar(Funcionario f) {
79 |         try {
80 |             String sql = "UPDATE funcionarios SET nome=?, cpf=?, email=?, telefone=?, cargo=?, foto=? WHERE id=?";
81 |             PreparedStatement stmt = con.prepareStatement(sql);
82 |
83 |             stmt.setString(1, f.getNome());
84 |             stmt.setString(2, f.getCpf());
85 |             stmt.setString(3, f.getEmail());
86 |             stmt.setString(4, f.getTelefone());
87 |             stmt.setString(5, f.getCargo());
88 |             stmt.setString(6, f.getFoto());
89 |             stmt.setString(7, f.getFoto_base64());
90 |             stmt.setInt(8, f.getId());
91 |
92 |             stmt.executeUpdate();
93 |         } catch (SQLException e) {
```

Figura 22-Funcionário pt3

```

91
92
93     stmt.executeUpdate();
94 } catch (SQLException e) {
95     e.printStackTrace();
96 }
97
98 // EXCLUIR
99 public void excluir(int id) {
100    try {
101        String sql = "DELETE FROM funcionarios WHERE id=?";
102        PreparedStatement stmt = con.prepareStatement(sql);
103
104        stmt.setInt(1, id);
105        stmt.executeUpdate();
106
107    } catch (SQLException e) {
108        e.printStackTrace();
109    }
110 }
111
112 // BUSCAR POR ID
113 public Funcionario buscarPorId(int id) {
114    try {
115        String sql = "SELECT * FROM funcionarios WHERE id=?";
116        PreparedStatement stmt = con.prepareStatement(sql);
117
118        stmt.setInt(1, id);
119        ResultSet rs = stmt.executeQuery();
120
121        if (rs.next()) {
122            Funcionario f = new Funcionario();
123
124            f.setId(rs.getInt("id"));
125            f.setNome(rs.getString("nome"));
126            f.setCpf(rs.getString("cpf"));
127            f.setEmail(rs.getString("email"));
128            f.setTelefone(rs.getString("telefone"));
129            f.setCargo(rs.getString("cargo"));
130            f.setFoto(rs.getString("foto"));
131            f.setFoto_base64(rs.getString("foto_base64"));
132
133            return f;
134        }
135
136    } catch (SQLException e) {
137        e.printStackTrace();
138    }
139
140
141
142
}

```

Figura 23-Funcionário pt4

```

138
139
140
141
142
}
return null;
}

```

Figura 24-Ordem

```

1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5  package dao;
6  /*
7   * João Pedro Rovaris
8   * Gustavo Figueiredo
9   * Guilherme Napolitano
10  */
11 import model.Ordem;
12 import java.sql.*;
13 import java.util.*;
14
15 public class OrdemDAO {
16
17     private Connection con;
18
19     public OrdemDAO() {
20         try {
21             con = Conexao.getConexao();
22         } catch (SQLException e) {
23             e.printStackTrace();
24         }
25     }
26
27     // =====
28     // CADASTRAR
29     // =====
30     public void cadastrar(Ordem o) {
31         try {
32             String sql = "INSERT INTO ordens (numero, id_setor, id_peca, quantidade, status, data) VALUES (?, ?, ?, ?, ?, ?)";
33             PreparedStatement stmt = con.prepareStatement(sql);
34
35             stmt.setString(1, o.getNumero());
36             stmt.setInt(2, o.getId_setor());
37             stmt.setInt(3, o.getId_peca());
38             stmt.setInt(4, o.getQuantidade());
39             stmt.setString(5, o.getStatus());
40             stmt.setString(6, o.getData());
41
42             stmt.executeUpdate();
43         } catch (SQLException e) {
44             e.printStackTrace();
45         }
46     }
47
48     // =====

```

Figura 25-Ordem pt2

```

48 // =====
49 // LISTAR COM JOIN
50 // =====
51 public List<Ordem> listar() {
52     List<Ordem> lista = new ArrayList<>();
53
54     try {
55         String sql =
56             "SELECT o.*, s.nome AS setorNome, p.nome AS pecaNome " +
57             "FROM ordens o " +
58             "JOIN setores s ON o.id_setor = s.id " +      // <-- CORRIGIDO
59             "JOIN pecas p ON o.id_peca = p.id";           // <-- CORRIGIDO
60
61         PreparedStatement stmt = con.prepareStatement(sql);
62         ResultSet rs = stmt.executeQuery();
63
64         while (rs.next()) {
65             Ordem o = new Ordem();
66
67             o.setId(rs.getInt("id"));
68             o.setNumero(rs.getString("numero"));
69             o.setId_setor(rs.getInt("id_setor"));
70             o.setId_peca(rs.getInt("id_peca"));
71             o.setQuantidade(rs.getInt("quantidade"));
72             o.setStatus(rs.getString("status"));
73             o.setData(rs.getString("data"));
74
75             // nomes trazidos pelo JOIN
76             o.setSetorNome(rs.getString("setorNome"));
77             o.setPeceNome(rs.getString("pecaNome"));
78
79             lista.add(o);
80         }
81     } catch (SQLException e) {
82         e.printStackTrace();
83     }
84
85     return lista;
86 }
87
88 // =====
89 // ATUALIZAR
90 // =====
91 public void atualizar(Ordem o) {
92     try {
93         String sql = "UPDATE ordens SET numero=?, id_setor=?, id_peca=?, quantidade=?, status=?, data=? WHERE id=?";
94         PreparedStatement stmt = con.prepareStatement(sql);
95

```

Figura 26-Ordem pt3

```

88 // =====
89 // ATUALIZAR
90 // =====
91 public void atualizar(Ordem o) {
92     try {
93         String sql = "UPDATE ordens SET numero=?, id_setor=?, id_peca=?, quantidade=?, status=?, data=? WHERE id=?";
94         PreparedStatement stmt = con.prepareStatement(sql);
95
96         stmt.setString(1, o.getNumero());
97         stmt.setInt(2, o.getId_setor());
98         stmt.setInt(3, o.getId_peca());
99         stmt.setInt(4, o.getQuantidade());
100        stmt.setString(5, o.getStatus());
101        stmt.setString(6, o.getData());
102        stmt.setInt(7, o.getId());
103
104        stmt.executeUpdate();
105    } catch (SQLException e) {
106        e.printStackTrace();
107    }
108 }
109
110 // =====
111 // EXCLUIR
112 // =====
113 public void excluir(int id) {
114     try {
115         String sql = "DELETE FROM ordens WHERE id=?";
116         PreparedStatement stmt = con.prepareStatement(sql);
117
118         stmt.setInt(1, id);
119         stmt.executeUpdate();
120
121     } catch (SQLException e) {
122        e.printStackTrace();
123    }
124 }
125
126 // =====
127 // BUSCAR POR ID (para editar.jsp)
128 // =====
129 public Ordem buscarPorId(int id) {
130     try {
131         String sql =
132             "SELECT o.*, s.nome AS setorNome, p.nome AS pecaNome " +
133             "FROM ordens o " +
134             "JOIN setores s ON o.id_setor = s.id " + // <-- CORRIGIDO
135             "JOIN pecas p ON o.id_peca = p.id " + // <-- CORRIGIDO

```

Figura 27-Ordem pt4

```
131     String sql =
132         "SELECT o.*, s.nome AS setorNome, p.nome AS pecaNome " +
133         "FROM ordens o " +
134         "JOIN setores s ON o.id_setor = s.id " + // <-- CORRIGIDO
135         "JOIN pecas p ON o.id_peca = p.id " + // <-- CORRIGIDO
136         "WHERE o.id=?";
137
138     PreparedStatement stmt = con.prepareStatement(sql);
139     stmt.setInt(1, id);
140     ResultSet rs = stmt.executeQuery();
141
142     if (rs.next()) {
143         Ordem o = new Ordem();
144
145         o.setId(rs.getInt("id"));
146         o.setNumero(rs.getString("numero"));
147         o.setId_setor(rs.getInt("id_setor"));
148         o.setId_peca(rs.getInt("id_peca"));
149         o.setQuantidade(rs.getInt("quantidade"));
150         o.setStatus(rs.getString("status"));
151         o.setData(rs.getString("data"));
152
153         o.setSetorNome(rs.getString("setorNome"));
154         o.setPeceNome(rs.getString("pecaNome"));
155
156         return o;
157     }
158
159     } catch (SQLException e) {
160         e.printStackTrace();
161     }
162
163     return null;
164 }
165
166 }
```

Figura 28-Peça

```

1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package dao;
6   /*
7   * João Pedro Rovaris
8   * Gustavo Figueiredo
9   * Guilherme Napolitano
10  */
11  import model.Peca;
12  import java.sql.*;
13  import java.util.*;

14
15  public class PecaDAO {
16
17      private Connection con;
18
19      public PecaDAO() {
20          try {
21              con = Conexao.getConexao();
22          } catch (SQLException e) {
23              e.printStackTrace();
24          }
25      }
26
27      // CADASTRAR
28      public void cadastrar(Peca p) {
29          try {
30              String sql = "INSERT INTO pecas (nome, peso, descricao, foto_base64) VALUES (?, ?, ?, ?)";
31              PreparedStatement stmt = con.prepareStatement(sql);
32              stmt.setString(1, p.getNome());
33              stmt.setDouble(2, p.getPeso());
34              stmt.setString(3, p.getDescricao());
35              stmt.setString(4, p.getFoto_base64());
36              stmt.executeUpdate();
37          } catch (SQLException e) {
38              e.printStackTrace();
39          }
40      }
41
42      // LISTAR
43      public List<Peca> listar() {
44          List<Peca> lista = new ArrayList<>();
45          try {
46              String sql = "SELECT * FROM pecas";
47              PreparedStatement stmt = con.prepareStatement(sql);
48              ResultSet rs = stmt.executeQuery();

```

Figura 29-Peça pt2

```

49 |         while (rs.next()) {
50 |             Peca p = new Peca();
51 |             p.setId(rs.getInt("id"));
52 |             p.setNome(rs.getString("nome"));
53 |             p.setPeso(rs.getDouble("peso"));
54 |             p.setDescricao(rs.getString("descricao"));
55 |             p.setFoto_base64(rs.getString("foto_base64"));
56 |             lista.add(p);
57 |         }
58 |
59 |     } catch (SQLException e) {
60 |         e.printStackTrace();
61 |     }
62 |     return lista;
63 | }
64 |
65 // BUSCAR POR ID
66 public Peca buscarPorId(int id) {
67     try {
68         String sql = "SELECT * FROM pecas WHERE id=?";
69         PreparedStatement stmt = con.prepareStatement(sql);
70         stmt.setInt(1, id);
71         ResultSet rs = stmt.executeQuery();
72 |
73     if (rs.next()) {
74         Peca p = new Peca();
75         p.setId(rs.getInt("id"));
76         p.setNome(rs.getString("nome"));
77         p.setPeso(rs.getDouble("peso"));
78         p.setDescricao(rs.getString("descricao"));
79         p.setFoto_base64(rs.getString("foto_base64"));
80         return p;
81     }
82 } catch (SQLException e) {
83     e.printStackTrace();
84 }
85     return null;
86 }
87 |
88 // ATUALIZAR
89 public void atualizar(Peca p) {
90     try {
91         String sql = "UPDATE pecas SET nome=?, peso=?, descricao=?, foto_base64=? WHERE id=?";
92         PreparedStatement stmt = con.prepareStatement(sql);
93         stmt.setString(1, p.getNome());
94         stmt.setDouble(2, p.getPeso());

```

Figura 28-Peça pt3

```
94     stmt.setDouble(2, p.getPeso());
95     stmt.setString(3, p.getDescricao());
96     stmt.setString(4, p.getFoto_base64());
97     stmt.setInt(5, p.getId());
98     stmt.executeUpdate();
99 } catch (SQLException e) {
100     e.printStackTrace();
101 }
102 }

103 // EXCLUIR
104 public void excluir(int id) {
105     try {
106         String sql = "DELETE FROM pecas WHERE id=?";
107         PreparedStatement stmt = con.prepareStatement(sql);
108         stmt.setInt(1, id);
109         stmt.executeUpdate();
110     } catch (SQLException e) {
111         e.printStackTrace();
112     }
113 }
114 }
115 }
```

Figura29-Setor

```

1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package dao;
6   /*
7   * João Pedro Rovaris
8   * Gustavo Figueiredo
9   * Guilherme Napolitano
10  */
11  import model.Setor;
12  import java.sql.*;
13  import java.util.*;
14
15  public class SetorDAO {
16
17      private Connection con;
18
19      public SetorDAO() {
20          try {
21              con = Conexao.getConexao();
22          } catch (SQLException e) {
23              e.printStackTrace();
24          }
25      }
26
27      // CADASTRAR
28      public void cadastrar(Setor s) {
29          try {
30              String sql = "INSERT INTO setores (nome, descricao, responsavel, foto, foto_base64) VALUES (?, ?, ?, ?, ?)";
31              PreparedStatement stmt = con.prepareStatement(sql);
32              stmt.setString(1, s.getNome());
33              stmt.setString(2, s.getDescricao());
34              stmt.setString(3, s.getResponsavel());
35              stmt.setString(4, s.getFoto());
36              stmt.setString(5, s.getFoto_base64());
37              stmt.executeUpdate();
38          } catch (SQLException e) {
39              e.printStackTrace();
40          }
41      }
42
43      // LISTAR
44      public List<Setor> listar() {
45          List<Setor> lista = new ArrayList<>();
46          try {
47              String sql = "SELECT * FROM setores ORDER BY id";
48              PreparedStatement stmt = con.prepareStatement(sql);

```

Figura30-Setor pt2

```
49
50
51     ResultSet rs = stmt.executeQuery();
52
53     while (rs.next()) {
54         Setor s = new Setor();
55         s.setId(rs.getInt("id"));
56         s.setNome(rs.getString("nome"));
57         s.setDescricao(rs.getString("descricao"));
58         s.setResponsavel(rs.getString("responsavel"));
59         s.setFoto(rs.getString("foto"));
60         s.setFoto_base64(rs.getString("foto_base64"));
61         lista.add(s);
62     }
63
64 } catch (SQLException e) {
65     e.printStackTrace();
66 }
67
68 // BUSCAR POR ID
69 public Setor buscarPorId(int id) {
70     try {
71         String sql = "SELECT * FROM setores WHERE id=?";
72         PreparedStatement stmt = con.prepareStatement(sql);
73         stmt.setInt(1, id);
74         ResultSet rs = stmt.executeQuery();
75
76         if (rs.next()) {
77             Setor s = new Setor();
78             s.setId(rs.getInt("id"));
79             s.setNome(rs.getString("nome"));
80             s.setDescricao(rs.getString("descricao"));
81             s.setResponsavel(rs.getString("responsavel"));
82             s.setFoto(rs.getString("foto"));
83             s.setFoto_base64(rs.getString("foto_base64"));
84             return s;
85         }
86
87     } catch (SQLException e) {
88         e.printStackTrace();
89     }
90
91     return null;
92 }
93
94 // ATUALIZAR
95 public void atualizar(Setor s) {
96     try {
97         String sql = "UPDATE setores SET nome=?, descricao=?, responsavel=?, foto=?, foto_base64=? WHERE id=?";
98         PreparedStatement stmt = con.prepareStatement(sql);
99         stmt.setString(1, s.getNome());
100        stmt.setString(2, s.getDescricao());
101        stmt.setString(3, s.getResponsavel());
102        stmt.setString(4, s.getFoto());
103        stmt.setString(5, s.getFoto_base64());
104        stmt.setInt(6, s.getId());
105        stmt.executeUpdate();
106    } catch (SQLException e) {
107        e.printStackTrace();
108    }
109 }
```

Figura31-Setor pt3

```
96     String sql = "UPDATE setores SET nome=?, descricao=?, responsavel=?,  
97     PreparedStatement stmt = con.prepareStatement(sql);  
98     stmt.setString(1, s.getNome());  
99     stmt.setString(2, s.getDescricao());  
100    stmt.setString(3, s.getResponsavel());  
101    stmt.setString(4, s.getFoto());  
102    stmt.setString(5, s.getFoto_base64());  
103    stmt.setInt(6, s.getId());  
104    stmt.executeUpdate();  
105  
106    } catch (SQLException e) {  
107        e.printStackTrace();  
108    }  
109}  
110  
111 // EXCLUIR  
112 public void excluir(int id) {  
113    try {  
114        String sql = "DELETE FROM setores WHERE id=?";  
115        PreparedStatement stmt = con.prepareStatement(sql);  
116        stmt.setInt(1, id);  
117        stmt.executeUpdate();  
118  
119    } catch (SQLException e) {  
120        e.printStackTrace();  
121    }  
122}  
123}  
124}
```

Figura32-Usuário

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package dao;
6   import dao.Conexao;
7   import model.Usuario;
8   import util.BCryptUtil;
9   import java.sql.*;
10  import java.util.*;
11  /*
12   * João Pedro Rovaris
13   * Gustavo Figueiredo
14   * Guilherme Napolitano
15   */
16  public class UsuarioDAO {
17      private Connection con;
18
19      public UsuarioDAO() {
20          try {
21              con = Conexao.getConexao();
22          } catch (SQLException e) {
23              e.printStackTrace();
24          }
25      }
26
27      // 🔑 LOGIN
28      public Usuario buscarPorEmail(String email) {
29          try {
30              String sql = "SELECT * FROM usuarios WHERE email=?";
31              PreparedStatement stmt = con.prepareStatement(sql);
32              stmt.setString(1, email);
33              ResultSet rs = stmt.executeQuery();
34              if (rs.next()) {
35                  Usuario u = new Usuario();
36                  u.setId(rs.getInt("id"));
37                  u.setNome(rs.getString("nome"));
38                  u.setEmail(rs.getString("email"));
39                  u.setSenha(rs.getString("senha"));
40                  u.setCargo(rs.getString("cargo"));
41                  u.setFoto(rs.getString("foto"));
42                  return u;
43              }
44          } catch (SQLException e) {
45              e.printStackTrace();
46          }
47      }
48      return null;
49  }
```

Figura33-Usuário pt 2

```

50 // 🔒 CADASTRAR (com senha criptografada)
51 public void cadastrar(Usuario u) {
52     try {
53         String sql = "INSERT INTO usuarios (nome, email, senha, cargo, foto) VALUES (?, ?, ?, ?, ?)";
54         PreparedStatement stmt = con.prepareStatement(sql);
55         stmt.setString(1, u.getNome());
56         stmt.setString(2, u.getEmail());
57         stmt.setString(3, BCryptUtil.hash(u.getSenha()));
58         stmt.setString(4, u.getCargo());
59         stmt.setString(5, u.getFoto());
60         stmt.executeUpdate();
61     } catch (SQLException e) {
62         e.printStackTrace();
63     }
64 }
65
66 // 🔎 LISTAR TODOS
67 public List<Usuario> listar() {
68     List<Usuario> lista = new ArrayList<>();
69     try {
70         String sql = "SELECT * FROM usuarios";
71         PreparedStatement stmt = con.prepareStatement(sql);
72         ResultSet rs = stmt.executeQuery();
73         while (rs.next()) {
74             Usuario u = new Usuario();
75             u.setId(rs.getInt("id"));
76             u.setNome(rs.getString("nome"));
77             u.setEmail(rs.getString("email"));
78             u.setCargo(rs.getString("cargo"));
79             u.setFoto(rs.getString("foto"));
80             lista.add(u);
81         }
82     } catch (SQLException e) {
83         e.printStackTrace();
84     }
85     return lista;
86 }
87
88 // 🔍 EDITAR PERFIL
89 public void atualizar(Usuario u) {
90     try {
91         String sql = "UPDATE usuarios SET nome=?, email=?, senha=?, cargo=?, foto=? WHERE id=?";
92         PreparedStatement stmt = con.prepareStatement(sql);
93         stmt.setString(1, u.getNome());
94         stmt.setString(2, u.getEmail());
95         stmt.setString(3, u.getSenha());
96         stmt.setString(4, u.getCargo());

```

Figura34-Usuário pt3

```

94         stmt.setString(2, u.getEmail());
95         stmt.setString(3, u.getSenha());
96         stmt.setString(4, u.getCargo());
97         stmt.setString(5, u.getFoto());
98         stmt.setInt(6, u.getId());
99         stmt.executeUpdate();
100        } catch (SQLException e) {
101            e.printStackTrace();
102        }
103    }
104
105    // ■ EXCLUIR
106    public void excluir(int id) {
107        try {
108            String sql = "DELETE FROM usuarios WHERE id=?";
109            PreparedStatement stmt = con.prepareStatement(sql);
110            stmt.setInt(1, id);
111            stmt.executeUpdate();
112        } catch (SQLException e) {
113            e.printStackTrace();
114        }
115    }
116
117    // ⚡ BUSCAR POR ID
118    public Usuario buscarPorId(int id) {
119        try {
120            String sql = "SELECT * FROM usuarios WHERE id=?";
121            PreparedStatement stmt = con.prepareStatement(sql);
122            stmt.setInt(1, id);
123            ResultSet rs = stmt.executeQuery();
124
125            if (rs.next()) {
126                Usuario u = new Usuario();
127                u.setId(rs.getInt("id"));
128                u.setNome(rs.getString("nome"));
129                u.setEmail(rs.getString("email"));
130                u.setSenha(rs.getString("senha"));
131                u.setCargo(rs.getString("cargo"));
132                u.setFoto(rs.getString("foto"));
133                return u;
134            }
135        } catch (SQLException e) {
136            e.printStackTrace();
137        }
138        return null;
139    }
140
141}

```

4.3. Servlet

Figura35-Funcionário

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5  package controller;
6
7  /*
8   * João Pedro Ventura Rovaris
9   * Gustavo Figueiredo
10  * Guilherma Napolitano
11  */
12
13 import dao.FuncionarioDAO;
14 import model.Funcionario;
15
16 import javax.servlet.*;
17 import javax.servlet.annotation.WebServlet;
18 import javax.servlet.http.*;
19 import java.io.IOException;
20 import java.nio.file.*;
21 import java.io.*;
22 import java.util.Base64;
23 import javax.servlet.annotation.MultipartConfig;
24
25 @MultipartConfig
26 @WebServlet("/FuncionarioServlet")
27 public class FuncionarioServlet extends HttpServlet {
28
29     @Override
30     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException, ServletException {
31         doPost(request, response); // para excluir via GET
32     }
33
34     @Override
35     protected void doPost(HttpServletRequest request, HttpServletResponse response)
36             throws ServletException, IOException {
37
38         request.setCharacterEncoding("UTF-8");
39
40         Stringacao = request.getParameter("acao");
41         FuncionarioDAO dao = new FuncionarioDAO();
42
43         // ===== CADASTRAR =====
44         if ("cadastrar".equals(acao)) {
45
46             Funcionario f = new Funcionario();
47
48             f.setNome(request.getParameter("nome"));
```

Figura36-Funcionário pt2

```

48     f.setNome(request.getParameter("nome"));
49     f.setCpf(request.getParameter("cpf"));
50     f.setEmail(request.getParameter("email"));
51     f.setTelefone(request.getParameter("telefone"));
52     f.setCargo(request.getParameter("cargo"));

53
54     // Foto
55     Part foto = request.getPart("foto");
56     processarFoto(f, foto, request);

57
58     dao.cadastrar(f);
59     response.sendRedirect("funcionario/listar.jsp");
60 }

61
62 // ===== ATUALIZAR =====
63 if ("atualizar".equals(acao)) {
64
65     Funcionario f = new Funcionario();

66
67     f.setId(Integer.parseInt(request.getParameter("id")));
68     f.setNome(request.getParameter("nome"));
69     f.setCpf(request.getParameter("cpf"));
70     f.setEmail(request.getParameter("email"));
71     f.setTelefone(request.getParameter("telefone"));
72     f.setCargo(request.getParameter("cargo"));

73
74     Part foto = request.getPart("foto");
75     processarFoto(f, foto, request);

76
77     dao.atualizar(f);
78     response.sendRedirect("funcionario/listar.jsp");
79 }

80
81 // ===== EXCLUIR =====
82 if ("excluir".equals(acao)) {
83     int id = Integer.parseInt(request.getParameter("id"));
84     dao.excluir(id);
85     response.sendRedirect("funcionario/listar.jsp");
86 }
87 }

88
89 // PROCESSAMENTO DE FOTO
90 private void processarFoto(Funcionario f, Part foto, HttpServletRequest request) throws IOException {
91
92     if (foto != null && foto.getSize() > 0) {
93
94         String nomeArquivo = Paths.get(foto.getSubmittedFileName()).getFileName().toString();
95
96
97         // salva no servidor
98         String caminho = request.getServletContext().getRealPath("/assets/uploads/");
99         Files.createDirectories(Paths.get(caminho));
100        foto.write(caminho + File.separator + nomeArquivo);
101
102        f.setFoto(nomeArquivo);
103
104        // Converte para Base64
105        byte[] bytes = Files.readAllBytes(Paths.get(caminho + File.separator + nomeArquivo));
106        String base64 = Base64.getEncoder().encodeToString(bytes);
107
108        f.setFoto_base64(base64);
109    }
110}

```

Figura37-Funcionário pt3

```

94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110

```

```

String nomeArquivo = Paths.get(foto.getSubmittedFileName()).getFileName().toString();

// salva no servidor
String caminho = request.getServletContext().getRealPath("/assets/uploads/");
Files.createDirectories(Paths.get(caminho));
foto.write(caminho + File.separator + nomeArquivo);

f.setFoto(nomeArquivo);

// Converte para Base64
byte[] bytes = Files.readAllBytes(Paths.get(caminho + File.separator + nomeArquivo));
String base64 = Base64.getEncoder().encodeToString(bytes);

f.setFoto_base64(base64);
}
}

```

Figura38-Login

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5  package controller;
6  /*
7   João Pedro Ventura Rovaris
8   Gustavo Figueiredo
9   Guilherma Napolitano
10  */
11 
12 import dao.UsuarioDAO;
13 import model.Usuario;
14 import javax.servlet.annotation.WebServlet;
15 import javax.servlet.http.*;
16 import javax.servlet.*;
17 import java.io.IOException;
18 @WebServlet("/LoginServlet")
19 public class LoginServlet extends HttpServlet {
20     @Override
21     protected void doPost(HttpServletRequest request, HttpServletResponse response)
22             throws ServletException, IOException {
23         request.setCharacterEncoding("UTF-8");
24         String email = request.getParameter("email");
25         String senha = request.getParameter("senha");
26         UsuarioDAO dao = new UsuarioDAO();
27         Usuario u = dao.buscarPorEmail(email);
28         // LOGIN SEM CRIPTOGRAFIA
29         if (u != null && senha.equals(u.getSenha())) {
30             request.getSession().setAttribute("usuario", u);
31             response.sendRedirect("dashboard.jsp");
32         } else {
33             response.sendRedirect("login.jsp?erro=1");
34         }
35     }
36 }
37 }
```

Figura39-Logout

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5  package controller;
6  /*
7   * João Pedro Ventura Rovaris
8   * Gustavo Figueiredo
9   * Guilherma Napolitano
10  */
11
12
13 import javax.servlet.*;
14 import javax.servlet.annotation.WebServlet;
15 import javax.servlet.http.*;
16 import java.io.IOException;
17
18 @WebServlet("/LogoutServlet")
19 public class LogoutServlet extends HttpServlet {
20
21     @Override
22     protected void doGet(HttpServletRequest request, HttpServletResponse response)
23     throws ServletException, IOException {
24
25         request.getSession().invalidate();
26         response.sendRedirect("login.jsp");
27     }
28 }
29 ]
```

Figura40-Ordem

```

1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package controller;
6   /*
7    * João Pedro Ventura Rovaris
8    * Gustavo Figueiredo
9    * Guilherma Napolitano
10   */
11
12  import dao.OrdemDAO;
13  import model.Ordem;
14  import javax.servlet.annotation.WebServlet;
15  import javax.servlet.http.*;
16  import javax.servlet.*;
17  import java.io.IOException;
18
19  @WebServlet("/OrdemServlet")
20  public class OrdemServlet extends HttpServlet {
21
22      // --- Permite usar GET (excluir pelo link)
23      @Override
24      protected void doGet(HttpServletRequest request, HttpServletResponse response)
25          throws ServletException, IOException {
26          doPost(request, response); // redireciona para o doPost
27      }
28
29      // --- Trata todas as ações
30      @Override
31      protected void doPost(HttpServletRequest request, HttpServletResponse response)
32          throws ServletException, IOException {
33
34          request.setCharacterEncoding("UTF-8");
35
36          String acao = request.getParameter("acao");
37          OrdemDAO dao = new OrdemDAO();
38
39          try {
40
41              // --- CADASTRAR NOVA ORDEM ---
42              if ("cadastrar".equals(acao)) {
43
44                  Ordem o = new Ordem();
45                  o.setNumero(request.getParameter("numero"));
46                  o.setId_setor(Integer.parseInt(request.getParameter("id_setor")));
47                  o.setId_peca(Integer.parseInt(request.getParameter("id_peca")));
48                  o.setQuantidade(Integer.parseInt(request.getParameter("quantidade")));

```

Figura41-Ordem pt2

```
49     o.setStatus(request.getParameter("status"));
50     o.setData(request.getParameter("data"));

51     dao.cadastrar(o);
52     response.sendRedirect("ordem/listar.jsp");
53     return;
54 }

55

56

57
58 // --- EXCLUIR ---
59 if ("excluir".equals(acao)) {
60
61     int id = Integer.parseInt(request.getParameter("id"));

62     dao.excluir(id);

63     response.sendRedirect("ordem/listar.jsp");
64     return;
65 }

66

67

68
69 // --- ATUALIZAR ---
70 if ("atualizar".equals(acao)) {
71
72     Ordem o = new Ordem();

73
74     o.setId(Integer.parseInt(request.getParameter("id")));
75     o.setNumero(request.getParameter("numero"));
76     o.setId_setor(Integer.parseInt(request.getParameter("id_setor")));
77     o.setId_peca(Integer.parseInt(request.getParameter("id_peca")));
78     o.setQuantidade(Integer.parseInt(request.getParameter("quantidade")));
79     o.setStatus(request.getParameter("status"));
80     o.setData(request.getParameter("data"));

81
82     dao.atualizar(o);

83
84     response.sendRedirect("ordem/listar.jsp");
85     return;
86 }

87

88
89
90 } catch (Exception e) {
91     e.printStackTrace();
92     response.getWriter().println("<h2>Erro no OrdemServlet:</h2>");
93     response.getWriter().println(e.getMessage());
94 }
95 }
96 }
```

Figura42-Peca

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5  package controller;
6  /*
7   João Pedro Ventura Rovaris
8   Gustavo Figueiredo
9   Guilherma Napolitano
10  */
11
12 import dao.PecaDAO;
13 import model.Peca;
14
15 import javax.servlet.*;
16 import javax.servlet.annotation.MultipartConfig;
17 import javax.servlet.annotation.WebServlet;
18 import javax.servlet.http.*;
19 import java.io.*;
20 import java.nio.file.*;
21 import java.util.Base64;
22
23 @WebServlet("/PecaServlet")
24 @MultipartConfig
25 public class PecaServlet extends HttpServlet {
26
27     @Override
28     protected void doPost(HttpServletRequest request, HttpServletResponse response)
29             throws ServletException, IOException {
30
31         request.setCharacterEncoding("UTF-8");
32         String acao = request.getParameter("acao");
33
34         PecaDAO dao = new PecaDAO();
35
36         try {
37
38             // ===== CADASTRAR =====
39             if ("cadastrar".equals(acao)) {
40
41                 Peca p = new Peca();
42                 p.setNome(request.getParameter("nome"));
43                 p.setPeso(Double.parseDouble(request.getParameter("peso")));
44                 p.setDescricao(request.getParameter("descricao"));
45
46                 Part foto = request.getPart("foto");
47                 processarFoto(p, foto, request);
48
49             }
50
51         } catch (Exception e) {
52             e.printStackTrace();
53         }
54     }
55
56 }
```

Figura43-Peca pt2

```

        dao.cadastrar(p);
        response.sendRedirect("peca/listar.jsp");
        return;
    }

    // ===== ATUALIZAR =====
    if ("atualizar".equals(acao)) {

        Peca p = new Peca();
        p.setId(Integer.parseInt(request.getParameter("id")));
        p.setNome(request.getParameter("nome"));
        p.setPeso(Double.parseDouble(request.getParameter("peso")));
        p.setDescricao(request.getParameter("descricao"));

        Part foto = request.getPart("foto");
        processarFoto(p, foto, request);

        dao.atualizar(p);
        response.sendRedirect("peca/listar.jsp");
        return;
    }

    // ===== EXCLUIR =====
    if ("excluir".equals(acao)) {
        int id = Integer.parseInt(request.getParameter("id"));
        dao.excluir(id);
        response.sendRedirect("peca/listar.jsp");
        return;
    }

} catch (Exception e) {
    e.printStackTrace();
    response.getWriter().println("<h2>Erro no PecaServlet:</h2><pre>" + e + "</pre>");
}
}

private void processarFoto(Peca p, Part foto, HttpServletRequest request) throws IOException {
    if (foto == null || foto.getSize() == 0) return;

    String nomeArquivo = Paths.get(foto.getSubmittedFileName()).getFileName().toString();
    String caminho = request.getServletContext().getRealPath("/assets/uploads/");
    Files.createDirectories(Paths.get(caminho));

    foto.write(caminho + File.separator + nomeArquivo);
    p.setFoto(nomeArquivo);
}

```

Figura44-Setor

```

1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package controller;
6   /*
7   João Pedro Ventura Rovaris
8   Gustavo Figueiredo
9   Guilherma Napolitano
10  */
11
12 import dao.SetorDAO;
13 import model.Setor;
14
15 import javax.servlet.*;
16 import javax.servlet.annotation.MultipartConfig;
17 import javax.servlet.annotation.WebServlet;
18 import javax.servlet.http.*;
19 import java.io.*;
20 import java.nio.file.*;
21 import java.util.Base64;
22
23 @WebServlet("/SetorServlet")
24 @MultipartConfig
25 public class SetorServlet extends HttpServlet {
26
27     @Override
28     protected void doPost(HttpServletRequest request, HttpServletResponse response)
29             throws ServletException, IOException {
30
31         request.setCharacterEncoding("UTF-8");
32         String acao = request.getParameter("acao");
33
34         SetorDAO dao = new SetorDAO();
35
36         try {
37
38             // ===== CADASTRAR =====
39             if ("cadastrar".equals(acao)) {
40
41                 Setor s = new Setor();
42                 s.setNome(request.getParameter("nome"));
43                 s.setDescricao(request.getParameter("descricao"));
44                 s.setResponsavel(request.getParameter("responsavel"));
45
46                 Part foto = request.getPart("foto");
47                 processarFoto(s, foto, request);
48

```

Figura45-Setor pt2

```

49    dao.cadastrar(s);
50    response.sendRedirect("setor/listar.jsp");
51    return;
52  }

53
54  // ===== ATUALIZAR =====
55  if ("atualizar".equals(acao)) {
56
57    Setor s = new Setor();
58    s.setId(Integer.parseInt(request.getParameter("id")));
59    s.setNome(request.getParameter("nome"));
60    s.setDescricao(request.getParameter("descricao"));
61    s.setResponsavel(request.getParameter("responsavel"));

62
63    Part foto = request.getPart("foto");
64    processarFoto(s, foto, request);

65
66    dao.atualizar(s);
67    response.sendRedirect("setor/listar.jsp");
68    return;
69  }

70
71  // ===== EXCLUIR =====
72  if ("excluir".equals(acao)) {
73    int id = Integer.parseInt(request.getParameter("id"));
74    dao.excluir(id);
75    response.sendRedirect("setor/listar.jsp");
76    return;
77  }

78  } catch (Exception e) {
79    e.printStackTrace();
80    response.getWriter().println("<h2>Erro no SetorServlet:</h2><pre>" + e + "</pre>");
81  }
82}

83
84 private void processarFoto(Setor s, Part foto, HttpServletRequest request) throws IOException {
85
86  if (foto == null || foto.getSize() == 0) return;

87  String nomeArquivo = Paths.get(foto.getSubmittedFileName()).getFileName().toString();
88  String caminho = request.getServletContext().getRealPath("/assets/uploads/");
89
90  Files.createDirectories(Paths.get(caminho));
91
92  foto.write(caminho + File.separator + nomeArquivo);
93  s.setFoto(nomeArquivo);
94
95
96}

```

Figura46-Setor pt3

```

    s.setFoto(nomeArquivo);

    byte[] bytes = Files.readAllBytes(Paths.get(caminho + File.separator + nomeArquivo));
    s.setFoto_base64(Base64.getEncoder().encodeToString(bytes));
}
}

```

Figura47-Usuário

```

1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package controller;
6   /*
7   João Pedro Ventura Rovaris
8   Gustavo Figueiredo
9   Guilherma Napolitano
10  */
11 
12  import dao.UsuarioDAO;
13  import java.io.File;
14  import model.Usuario;
15 
16  import javax.servlet.annotation.MultipartConfig;
17  import javax.servlet.annotation.WebServlet;
18  import javax.servlet.http.*;
19  import javax.servlet.*;
20  import java.io.IOException;
21  import java.nio.file.*;
22  import java.util.Base64;
23 
24  @WebServlet("/UsuarioServlet")
25  @MultipartConfig
26  public class UsuarioServlet extends HttpServlet {
27 
28      @Override
29      protected void doPost(HttpServletRequest request, HttpServletResponse response)
30          throws ServletException, IOException {
31 
32          HttpSession sessao = request.getSession();
33          Usuario logado = (Usuario) sessao.getAttribute("usuario");
34 
35          String nome = request.getParameter("nome");
36          String senha = request.getParameter("senha");
37 
38          Part foto = request.getPart("foto");
39          String nomeArquivo = null;
40          String base64 = null;
41 
42          if (foto != null && foto.getSize() > 0) {
43 
44              nomeArquivo = Paths.get(foto.getSubmittedFileName()).getFileName().toString();
45              String caminhoUpload = getServletContext().getRealPath("/assets/uploads/");
46              Files.createDirectories(Paths.get(caminhoUpload));
47              Path destino = Paths.get(caminhoUpload + File.separator + nomeArquivo);
48              foto.write(destino.toString());
49 
50              byte[] bytes = Files.readAllBytes(destino);
51              base64 = Base64.getEncoder().encodeToString(bytes);
52 
53              logado.setFoto(nomeArquivo);
54              logado.setFoto_base64(base64);
55          }
56 
57          logado.setNome(nome);
58 
59          if (senha != null && !senha.isEmpty()) {
60              logado.setSenha(senha);
61          }
62 
63          UsuarioDAO dao = new UsuarioDAO();
64          dao.atualizar(logado);
65 
66          sessao.setAttribute("usuario", logado);
67 
68          response.sendRedirect("perfil.jsp?ok=1");
69      }
70  }

```

Figura48-Usuário pt2

```

48          foto.write(destino.toString());
49 
50          byte[] bytes = Files.readAllBytes(destino);
51          base64 = Base64.getEncoder().encodeToString(bytes);
52 
53          logado.setFoto(nomeArquivo);
54          logado.setFoto_base64(base64);
55      }
56 
57      logado.setNome(nome);
58 
59      if (senha != null && !senha.isEmpty()) {
60          logado.setSenha(senha);
61      }
62 
63      UsuarioDAO dao = new UsuarioDAO();
64      dao.atualizar(logado);
65 
66      sessao.setAttribute("usuario", logado);
67 
68      response.sendRedirect("perfil.jsp?ok=1");
69  }
70 }

```

4.4. JSP

4.1.1 JSP Principal

Figura49-Dashboard

```

1  <%-- Document : dashboard
2   Created on : 11 de nov. de 2025, 22:26:17
3   Author    : João Rovaris
4   |          Gustavo Figueiredo
5   |          Guilherme Napolitano
6   --%>
7
8
9  <%@ page import="model.Usuario" %>
10 <%
11   Usuario usuario = (Usuario) session.getAttribute("usuario");
12   if (usuario == null) {
13     response.sendRedirect("index.jsp");
14     return;
15   }
16 <%
17 <!DOCTYPE html>
18 <html lang="pt-br">
19 <head>
20   <meta charset="UTF-8">
21   <title>Painel - Sistema de Produção</title>
22
23   <!-- Bootstrap -->
24   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
25
26   <!-- CSS Externo -->
27   <link rel="stylesheet" href="assets/css/style.css">
28 </head>
29 <body class="index-bg">
30
31
32 <nav class="navbar custom-navbar">
33   <span class="navbar-brand">Sistema Produção</span>
34 <div>
35   <a href="perfil.jsp" class="btn nav-btn-outline">Perfil</a>
36   <a href="LogoutServlet" class="btn nav-btn-danger">Sair</a>
37 </div>
38 </nav>
39
40 <div class="dashboard-wrapper">
41 <div class="dashboard-card">
42   <h3>Bem-vindo, <%= usuario.getCargo() %> <%= usuario.getNome() %>!</h3>
43   <p class="cargo-text">Cargo: <%= usuario.getCargo() %></p>
44
45   <div class="dashboard-buttons">
46     <a href="peca/listar.jsp" class="dashboard-btn">Gerenciar Peças</a>
47     <a href="setor/listar.jsp" class="dashboard-btn">Gerenciar Setores</a>
48     <a href="funcionario/listar.jsp" class="dashboard-btn">Funcionários</a>

```

Figura50-Dashboard pt2

```

49   <a href="ordem/listar.jsp" class="dashboard-btn">Ordens de Produção</a>
50   </div>
51 </div>
52 </div>
53
54
55 </body>
56 </html>
57

```

Figura51-Index

```
<!--
Document : index
Created on : 11 de nov. de 2015, 22:21:31
Author   : João Rovaris
           | Gustavo Figueiredo
           | Guilherme Napolitano
-->

<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html lang="pt-br">
    <head>
        <meta charset="UTF-8">
        <title>Login - Sistema de Produção</title>
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
        <link rel="stylesheet" href="assets/css/style.css">
    </head>
    <body class="index-bg">
        <div class="container d-flex justify-content-center align-items-center vh-100">
            <div class="card p-4 shadow" style="width: 360px;">
                <h4 class="text-center mb-4">Acesso ao Sistema</h4>
                <form method="post" action="LoginServlet">
                    <div class="mb-3">
                        <label for="email" class="form-label">E-mail</label>
                        <input type="text" class="form-control" id="email" name="email" required>
                    </div>
                    <div class="mb-3">
                        <label for="senha" class="form-label">Senha</label>
                        <input type="password" class="form-control" id="senha" name="senha" required>
                    </div>
                    <button type="submit" class="btn btn-primary w-100">Entrar</button>
                </form>
                <% if (request.getParameter("erro") != null) { %>
                <div class="alert alert-danger mt-3 text-center">
                    Usuário ou senha inválidos!
                </div>
                <% } %>
            </div>
        </div>
    </body>
</html>
```

Figura52-Login

```

<%-- Document : login
   Created on : 24 de nov. de 2025, 21:19:59
   Author     : PICHAU
--%>

<%-- Document : login
   Created on : 24 de nov. de 2025, 21:19:59
   Author     : João Rovaris
               Gustavo Figueiredo
               Guilherme Napolitano
--%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Login - Sistema de Produção</title>

        <!-- Bootstrap CSS -->
        <link rel="stylesheet"
              href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css">
        <!-- CSS Externo-->
        <link rel="stylesheet" href="assets/css/style.css">
    </head>
    <body class="index-bg">
        <div class="container">
            <div class="container d-flex justify-content-center align-items-center vh-100">
                <div class="card p-4 shadow" style="width: 360px;">
                    <div class="logo">Sistema de Produção</div>
                    <%
                        if (request.getParameter("erro") != null) {
                    %>
                        <div class="alert alert-danger text-center">
                            Usuário ou senha incorretos!
                        </div>
                    <%
                        }
                    %>
                    <form action="LoginServlet" method="post">
                        <div class="mb-3">
                            <label class="form-label">E-mail</label>
                            <input type="email" name="email" class="form-control"
                                   placeholder="Digite seu e-mail" required>
                        </div>
                        <div class="mb-3">
                            <label class="form-label">Senha</label>
                            <input type="password" name="senha" class="form-control"
                                   placeholder="Digite sua senha" required>
                        </div>
                        <button type="submit" class="btn btn-primary w-100">
                            Entrar
                        </button>
                    </form>
                </div>
            </div>
        </div>
        <!-- Bootstrap JS -->
        <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
    </body>
</html>

```

Figura53-Login pt2

```

48         <input type="email" name="email" class="form-control"
49                               placeholder="Digite seu e-mail" required>
50
51         <div class="mb-3">
52             <label class="form-label">Senha</label>
53             <input type="password" name="senha" class="form-control"
54                               placeholder="Digite sua senha" required>
55
56         <button type="submit" class="btn btn-primary w-100">
57             Entrar
58         </button>
59     </form>
60   </div>
61 </div>
62
63
64 <!-- Bootstrap JS -->
65 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
66
67 </body>
</html>

```

Figura54-Perfil

```

<%-->
    Document      : perfil
    Created on   : 11 de nov. de 2025, 22:44:15
    Author        : João Rovaris
                    Gustavo Figueiredo
                    Guilherme Napolitano
--%>
<%@ page import="model.Usuario" %>
<%
    Usuario usuario = (Usuario) session.getAttribute("usuario");
    if (usuario == null) {
        response.sendRedirect("index.jsp");
        return;
    }
%>
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <title>Meu Perfil - Sistema de Produção</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" href="assets/css/style.css">
</head>
<body class="gradient-bg">

<nav class="navbar custom-navbar">
    <div class="container-fluid">
        <span class="navbar-brand" href="dashboard.jsp">Sistema Produção</span>
        <div>
            <a href="dashboard.jsp" class="btn btn-outline-light btn-sm me-2">Voltar</a>
            <a href="LogoutServlet" class="btn btn-danger btn-sm">Sair</a>
        </div>
    </div>
</nav>

<div class="container mt-5">
    <div class="card shadow p-4 mx-auto" style="max-width: 600px;">
        <h4 class="mb-4 text-center">Meu Perfil</h4>

        <form method="post" action="UsuarioServlet" enctype="multipart/form-data">
            <input type="hidden" name="acao" value="atualizar">
            <input type="hidden" name="id" value="<%= usuario.getId() %>">

            <div class="text-center mb-3">
                <%
                    String fotoURL;
                    if (usuario.getFoto_base64() != null && !usuario.getFoto_base64().isEmpty()) {
                        fotoURL = "data:image/*;base64," + usuario.getFoto_base64();
                %>
            </div>
        </form>
    </div>
</div>

```

Figura55-Perfil pt2

```

49 } else if (usuario.getFoto() != null && !usuario.getFoto().isEmpty()) {
50     fotoURL = "assets/uploads/" + usuario.getFoto();
51 } else {
52     fotoURL = "https://cdn-icons-png.flaticon.com/512/149/149071.png";
53 }
54 %>
55
56     
57
58     <div>
59         <input type="file" class="form-control mt-2" name="foto">
60     </div>
61 </div>
62
63
64
65     <div class="mb-3">
66         <label class="form-label">Nome</label>
67         <input type="text" class="form-control" name="nome" value="<%= usuario.getNome() %>" required>
68     </div>
69
70     <div class="mb-3">
71         <label class="form-label">E-mail</label>
72         <input type="email" class="form-control" value="<%= usuario.getEmail() %>" readonly>
73     </div>
74
75     <div class="mb-3">
76         <label class="form-label">Nova Senha</label>
77         <input type="password" class="form-control" name="senha" placeholder="Digite uma nova senha (opcional)">
78     </div>
79
80
81     <button type="submit" class="btn btn-primary w-100">Salvar Alterações</button>
82
83     <% if (request.getParameter("ok") != null) { %>
84         <div class="alert alert-success mt-3 text-center">
85             Perfil atualizado com sucesso!
86         </div>
87     <% } %>
88
89     </form>
90 </div>
91
92 </body>
93 </html>
...

```

4.1.2 JSP de Usuário

Figura56-Novo

```

<%-->
Document      : novo
Created on   : 24 de nov. de 2025, 21:15:04
Author        : João Rovaris
               : Gustavo Figueiredo
               : Guilherme Napolitano
--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>
<html>
<head>
    <title>Novo Usuário</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
    <link rel="stylesheet" href="../assets/css/style.css">
    <link rel="stylesheet" href="../assets/css/new.css">
</head>

<body class="gradient-bg">
    <div class="container">
        <h2>Cadastrar Usuário</h2>
        <form action="/UsuarioServlet" method="post" enctype="multipart/form-data">
            <input type="hidden" name="acao" value="cadastrar">

            <div class="form-group">
                Nome:<br>
                <input type="text" name="nome" class="form-control" required><br>
            </div>

            <div class="form-group">
                Email:<br>
                <input type="email" name="email" class="form-control" required><br>
            </div>

            <div class="form-group">
                Senha:<br>
                <input type="password" name="senha" class="form-control" required><br>
            </div>

            <div class="form-group">
                Cargo:<br>
                <input type="text" name="cargo" class="form-control" required><br>
            </div>

            <div class="form-group">
                Foto de Perfil:<br>
                <input type="file" name="foto" class="form-control"><br><br>
            </div>
        </form>
    </div>
</body>

```

Figura57-Novo pt2

```

48           <input type="file" name="foto" class="form-control"><br><br>
49           </div>
50           <button class="btn-add">Cadastrar</button>
51           <a href="listar.jsp" class="btn btn-secondary">Cancelar</a>
52       </form>
53   </div>
54 </body>
55 </html>
56

```

Figura58-Listar

```
1  <%--  
2   Document  : listar  
3   Created on : 24 de nov. de 2025, 21:14:51  
4   Author    : João Rovaris  
5           | Gustavo Figueiredo  
6           | Guilherme Napolitano  
7  --%>  
8  
9  <%@page import="java.util.List"%>  
10 <%@page import="dao.UsuarioDAO"%>  
11 <%@page import="model.Usuario"%>  
12  
13 <%  
14   UsuarioDAO dao = new UsuarioDAO();  
15   List<Usuario> lista = dao.listar();  
16 %>  
17  
18  <!DOCTYPE html>  
19 <html lang="pt-br">  
20  <head>  
21    <meta charset="UTF-8">  
22    <title>Usuários - Sistema Produção</title>  
23  
24    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">  
25    <link rel="stylesheet" href="../assets/css/style.css">  
26  </head>  
27  
28 <body class="gradient-bg">  
29  
30  <!-- NAVBAR PADRÃO DO TEMA -->  
31 <nav class="navbar custom-navbar">  
32   <span class="navbar-brand">Sistema Produção</span>  
33   <div>  
34     <a href="../dashboard.jsp" class="btn nav-btn-outline me-2">Dashboard</a>  
35     <a href="../LogoutServlet" class="btn nav-btn-danger">Sair</a>  
36   </div>  
37 </nav>  
38  
39 <div class="container listar-container">  
40  
41   <div class="table-card">  
42     <h2 class="table-title">Lista de Usuários</h2>  
43  
44   <table class="table custom-table table-hover align-middle">  
45     <thead>  
46       <tr>  
47         <th>ID</th>  
48         <th>Nome</th>
```

Figura59-Listar pt2

```
48 |           <th>Nome</th>
49 |           <th>Email</th>
50 |           <th>Cargo</th>
51 |           <th>Foto</th>
52 |       </tr>
53 |   </thead>
54 |
55 |   <tbody>
56 |       <% for (Usuario u : lista) { %>
57 |       <tr>
58 |           <td><%= u.getId() %></td>
59 |           <td><%= u.getNome() %></td>
60 |           <td><%= u.getEmail() %></td>
61 |           <td><%= u.getCargo() %></td>
62 |
63 |           <td>
64 |               <% if (u.getFoto() != null) { %>
65 |                   
66 |               <% } else { %>
67 |                   <span class="text-muted">?</span>
68 |               <% } %>
69 |           </td>
70 |       </tr>
71 |   <% } %>
72 | </tbody>
73 | </table>
74 | </div>
75 |
76 | </div>
77 |
78 | </body>
79 | </html>
80 |
```

Figura60-Editar

```

1 <%-- Document : editar
2   Created on : 24 de nov. de 2025, 21:15:15
3   Author : João Rovaris
4   Gustavo Figueiredo
5   Guilherme Napolitano
6 --%>
7
8
9  <%@page import="model.Usuario"%>
10 <%@page import="dao.UsuarioDAO"%>
11
12 <%
13   Usuario u = (Usuario) request.getSession().getAttribute("usuario");
14 %>
15
16 <!DOCTYPE html>
17 <html>
18 <head>
19   <title>Editar Perfil</title>
20   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
21   <link rel="stylesheet" href="../../assets/css/style.css">
22   <link rel="stylesheet" href="../../assets/css/new.css">
23 </head>
24 <body class="gradient-bg">
25
26 <div class="container">
27   <h2>Editar Perfil</h2>
28   <form action="../UsuarioServlet" method="post" enctype="multipart/form-data">
29     <input type="hidden" name="id" value="<%= u.getId() %>">
30
31     <div class="form-group">
32       Nome:<br>
33       <input type="text" class="form-control" name="nome" value="<%= u.getNome() %>" required><br>
34     </div>
35
36     <div class="form-group">
37       Email:<br>
38       <input type="email" class="form-control" value="<%= u.getEmail() %>" disabled><br>
39     </div>
40
41     <div class="form-group">
42       Nova senha (opcional):<br>
43       <input type="password" name="senha" class="form-control"><br>
44     </div>
45
46     <div class="form-group">
47       Cargo:<br>
48       <input type="text" name="cargo" class="form-control" value="<%= u.getCargo() %>" required><br>

```

Figura61-Editar pt2

```

48   <input type="text" name="cargo" class="form-control" value="<%= u.getCargo() %>" required><br>
49   </div>
50
51   <div class="form-group">
52     Foto atual:<br>
53     <% if (u.getFoto() != null) { %>
54       <br><br>
55     <% } %>
56   </div>
57
58   <div class="form-group">
59     Alterar foto:<br>
60     <input type="file" name="foto" class="form-control"><br><br>
61   </div>
62   <button class="btn-add">Salvar</button>
63   <a href="listar.jsp" class="btn btn-secondary">Voltar</a>
64 </form>
65 </div>
66 </body>
67 </html>
68

```

4.1.3 JSP de Setor

Figura62-Novo

```
3      Created on : 24 de nov. de 2025, 20:56:46
4      Author     : João Rovaris
5                  Gustavo Figueiredo
6                  Guilherme Napolitano
7      --%>
8
9      <%@page contentType="text/html" pageEncoding="UTF-8"%>
10
11     <!DOCTYPE html>
12     <html>
13     <head>
14         <title>Novo Setor</title>
15         <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
16         <link rel="stylesheet" href="../assets/css/style.css">
17         <link rel="stylesheet" href="../assets/css/new.css">
18     </head>
19
20     <body class="gradient-bg">
21     <div class="container">
22         <h2>Novo Setor</h2>
23         <form action="../SetorServlet" method="post" enctype="multipart/form-data">
24             <input type="hidden" name="acao" value="cadastrar">
25
26             <div class="form-group">
27                 Nome:<br>
28                 <input type="text" name="nome" class="form-control" required><br>
29             </div>
30
31             <div class="form-group">
32                 Descrição:<br>
33                 <textarea name="descricao" class="form-control"></textarea><br>
34             </div>
35
36             <div class="form-group">
37                 Responsável:<br>
38                 <input type="text" name="responsavel" class="form-control"><br>
39             </div>
40
41             <div class="form-group">
42                 Foto:<br>
43                 <input type="file" name="foto" class="form-control"><br>
44             </div>
45             <button class="btn-add">Salvar</button>
46             <a href="listar.jsp" class="btn btn-secondary">Cancelar</a>
47         </form>
48     </div>
49     </body>
50     </html>
```

Figura64-Listar

```

1  <%--  
2   Document      : listar  
3   Created on : 24 de nov. de 2025, 20:52:53  
4   Author       : João Rovaris  
5           | Gustavo Figueiredo  
6           | Guilherme Napolita  
7  --%>  
8  
9  <%@page import="dao.SetorDAO"%>  
10 <%@page import="model.Setor"%>  
11 <%@page import="java.util.List"%>  
12 <%@page contentType="text/html" pageEncoding="UTF-8"%>  
13  
14 <%  
15   SetorDAO dao = new SetorDAO();  
16   List<Setor> lista = dao.listar();  
17 %>  
18  
19 <!DOCTYPE html>  
20  
21 <html>  
22 <head>  
23   <title>Lista de Setores</title>  
24   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">  
25   <link rel="stylesheet" href="../assets/css/style.css">  
26 </head>  
27  
28 <body class="gradient-bg">  
29  
30   <h2 class="mb-4 text-white">Gerenciar Setores</h2>  
31  
32   <a href="novo.jsp" class="btn-add">+ Novo Setor</a>  
33  
34   <div class="table-card">  
35     <table class="table custom-table table-striped align-middle">  
36       <thead>  
37         <tr>  
38           <th>Foto</th>  
39           <th>ID</th>  
40           <th>Nome</th>  
41           <th>Descrição</th>  
42           <th>Responsável</th>  
43           <th>Ações</th>  
44         </tr>  
45       </thead>  
46       <tbody>  
47   </tbody>  
48

```

Figura65-Listar pt2

```
17 |         <tbody>
18 |             |
19 |             <% for (Setor s : lista) { %>
20 |                 <tr>
21 |                     <td>
22 |                         <% if (s.getFoto_base64() != null) { %>
23 |                             
24 |                         <% } else { %>
25 |                             
26 |                         <% } %>
27 |                     </td>
28 |
29 |                     <td><%= s.getId() %></td>
30 |                     <td><%= s.getNome() %></td>
31 |                     <td><%= s.getDescricao() %></td>
32 |                     <td><%= s.getResponsavel() %></td>
33 |
34 |                     <td>
35 |                         <a href="editar.jsp?id=<%= s.getId() %>" class="btn btn-primary btn-sm">Editar</a>
36 |                         <a href="../SetorServlet?acao=excluir&id=<%= s.getId() %>" class="btn btn-danger btn-sm"
37 |                             accesskey="" onclick="return confirm('Deseja excluir este setor?')">
38 |                             Excluir
39 |                         </a>
40 |                     </td>
41 |                 </tr>
42 |             <% } %>
43 |
44 |         </tbody>
45 |     </table>
46 | 
```

Figura66-Editar

```

<%-->
    Document : editar
    Created on : 24 de nov. de 2025, 20:57:21
    Author   : João Rovaris
               Gustavo Figueiredo
               Guilherme Napolito
--%>

<%@page import="dao.SetorDAO"%>
<%@page import="model.Setor"%>
<%
    int id = Integer.parseInt(request.getParameter("id"));
    SetorDAO dao = new SetorDAO();
    Setor s = dao.buscarPorId(id);
%>

<!DOCTYPE html>
<html>
<head>
    <title>Editar Setor</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
    <link rel="stylesheet" href="../assets/css/style.css">
    <link rel="stylesheet" href="../assets/css/new.css">
</head>
<body class="gradient-bg">

    <div class="container">
        <h2>Editar Setor</h2>
        <form action="../SetorServlet" method="post" enctype="multipart/form-data">
            <input type="hidden" name="acao" value="atualizar">
            <input type="hidden" name="id" value="<%= s.getId() %>">

            <div class="form-group">
                Nome:<br>
                <input type="text" name="nome" class="form-control" value="<%= s.getNome() %>" required><br>
            </div>

            <div class="form-group">
                Descrição:<br>
                <textarea name="descricao" class="form-control" value="<%= s.getDescricao() %>"></textarea><br>
            </div>

            <div class="form-group">
                Responsável:<br>
                <input type="text" name="responsavel" class="form-control" value="<%= s.getResponsavel() %>"><br>
            </div>

            <div class="form-group">

```

Figura67-Editar pt2

```

42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
<div class="form-group">
    Responsável:<br>
    <input type="text" name="responsavel" class="form-control" value="<%= s.getResponsavel() %>"><br>
</div>

<div class="form-group">
    Foto Atual:<br>
    <% if (s.getFoto_base64() != null) { %>
        
    <% } else { %>
        
    <% } %>
    <br><br>
</div>

<div class="form-group">
    Nova Foto:<br>
    <input type="file" name="foto" class="form-control"><br>
</div>
<button class="btn-add">Salvar</button>
<a href="listar.jsp" class="btn btn-secondary">Cancelar</a>
</form>
</div>
</body>
</html>

```

4.1.4 JSP de Peça

Figura68-Novo

```
<%--  
    Document : novo  
    Created on : 24 de nov. de 2025, 20:39:27  
    Author : João Pedro Rovaris  
              Gustavo Figueiredo  
              Guilherme Napolitano  
--%>  
  
<%@page contentType="text/html" pageEncoding="UTF-8"%>  
<!DOCTYPE html>  
<html>  
<head>  
    <title>Nova Peça</title>  
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">  
    <link rel="stylesheet" href="../assets/css/style.css">  
    <link rel="stylesheet" href="../assets/css/new.css">  
</head>  
  
<body class="gradient-bg">  
<div class="container">  
    <h2>Nova Peça</h2>  
    <form action="../PecaServlet" method="post" enctype="multipart/form-data">  
        <input type="hidden" name="acao" value="cadastrar">  
        <div class="form-group">  
            Nome:<br>  
            <input type="text" name="nome" class="form-control" required><br>  
        </div>  
  
        <div class="form-group">  
            Peso (KG):<br>  
            <input type="number" step="0.01" name="peso" class="form-control" required><br>  
        </div>  
  
        <div class="form-group">  
            Descrição: <br>  
            <textarea name="descricao" class="form-control"></textarea><br>  
        </div>  
  
        <div class="form-group">  
            Foto:<br>  
            <input type="file" name="foto" class="form-control" accept="image/*"><br>  
        </div>  
        <button class="btn-add">Salvar</button>  
        <a href="listar.jsp" class="btn btn-secondary">Cancelar</a>  
    </form>  
</div>  
</body>  
</html>
```

Figura69-Listar

```
1  <%--  
2   Document  : listar  
3   Created on : 24 de nov. de 2025, 20:39:56  
4   Author     : João Pedro Rovaris  
5           | Gustavo Figueiredo  
6           | Guilherme Napolitano  
7  --%>  
8  
9  <%@page import="dao.PecaDAO"%>  
10 <%@page import="model.Peca"%>  
11 <%@page import="java.util.List"%>  
12 <%@page contentType="text/html" pageEncoding="UTF-8"%>  
13  
14 <%  
15   PecaDAO dao = new PecaDAO();  
16   List<Peca> lista = dao.listar();  
17 %>  
18  
19  <!DOCTYPE html>  
20 <html>  
21 <head>  
22   <title>Lista de Peças</title>  
23   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">  
24   <link rel="stylesheet" href="../assets/css/style.css">  
25 </head>  
26  
27 <body class="gradient-bg">  
28  
29  <h2 class="mb-4 text-white">Lista de Peças</h2>  
30  
31  <a href="novo.jsp" class="btn-add">+ Nova Peça</a>  
32  
33  
34 <div class="table-card">  
35 <table class="table custom-table table-striped align-middle">  
36 <thead>  
37 <tr>  
38   <th>Foto</th>  
39   <th>ID</th>  
40   <th>Nome</th>  
41   <th>Peso</th>  
42   <th>Descrição</th>  
43   <th>Ações</th>  
44 </tr>  
45 </thead>  
46  
47 <tbody>  
48   <% for (Peca p : lista) { %>
```

Figura70-Listar pt2

```
16
17 <tbody>
18     <% for (Peca p : lista) { %>
19         <tr>
20             <td>
21                 <% if (p.getFoto_base64() != null) { %>
22                     
23                 <% } else { %>
24                     
25                 <% } %>
26             </td>
27
28             <td><%= p.getId() %></td>
29             <td><%= p.getNome() %></td>
30             <td><%= p.getPeso() %></td>
31             <td><%= p.getDescricao() %></td>
32
33             <td>
34                 <a href="editar.jsp?id=<%= p.getId() %>" class="btn btn-primary btn-sm">Editar</a>
35                 <a href="../PecaServlet?acao=excluir&id=<%= p.getId() %>" class="btn btn-danger btn-sm">Excluir</a>
36             </td>
37         </tr>
38     <% } %>
39 </tbody>
40 </table>
41 </div>
42
43 </body>
44 </html>
```

Figura71-Editar

```
1 <%--  
2 Document : editar  
3 Created on : 24 de nov. de 2025, 20:39:46  
4 Author : João Pedro Rovaris  
5 | Gustavo Figueiredo  
6 | Guilherme Napolitano  
7 --%>  
8 <%@page import="dao.PecaDAO"%>  
9 <%@page import="model.Peca"%>  
10 <%  
11 int id = Integer.parseInt(request.getParameter("id"));  
12 PecaDAO dao = new PecaDAO();  
13 Peca p = dao.buscarPorId(id);  
14 %>  
15  
16 <!DOCTYPE html>  
17 <html>  
18 <head>  
19 <title>Editar Peça</title>  
20 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">  
21 <link rel="stylesheet" href="../assets/css/style.css">  
22 <link rel="stylesheet" href="../assets/css/new.css">  
23 </head>  
24 <body class="gradient-bg">  
25  
26 <div class="container">  
27 <h2>Editar Peça</h2>  
28 <form action="./PecaServlet" method="post" enctype="multipart/form-data">  
29 <input type="hidden" name="acao" value="atualizar">  
30 <input type="hidden" name="id" value="<%= p.getId() %>">  
31  
32 <div class="form-group">  
33 <label>Nome:</label>  
34 <input type="text" name="nome" class="form-control" value="<%= p.getNome() %>" required><br>  
35 </div>  
36  
37 <div class="form-group">  
38 <label>Peso (KG):</label>  
39 <input type="number" step="0.01" name="peso" class="form-control" value="<%= p.getPeso() %>" required><br>  
40 </div>  
41  
42 <div class="form-group">  
43 <label>Descrição:<br></label>  
44 <textarea name="descricao" class="form-control">  
45 <%= p.getDescricao() %></textarea><br>  
46 </div>  
47  
48 <div class="form-group">  
49 <label>Foto atual:<br></label>
```

Figura72-Editar pt2

```
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
```

```
<div class="form-group">
    Foto atual:<br>
    <% if (p.getFoto_base64() != null) { %>
        
    <% } else { %>
        
    <% } %>
    <br><br>
</div>

<div class="form-group">
    Alterar foto:
    <input type="file" name="foto" class="form-control" accept="image/*"><br>
</div>
<button class="btn-add">Salvar Alterações</button>
<a href="listar.jsp" class="btn btn-secondary">Cancelar</a>
</form>
</div>
</body>
</html>
```

4.1.5 JSP de ordem

Figura73-Novo

```

1   <%-- 
2       Document      : novo
3       Created on   : 24 de nov. de 2025, 20:59:16
4       Author        : João Pedro Rovaris
5               | Gustavo Figueiredo
6               | Guilherme Napolitano
7   --%>
8   <%@page import="dao.SetorDAO"%>
9   <%@page import="dao.PecaDAO"%>
10  <%@page import="model.Setor"%>
11  <%@page import="model.Peca"%>
12  <%@page contentType="text/html" pageEncoding="UTF-8"%>
13
14  <%
15      SetorDAO sdao = new SetorDAO();
16      PecaDAO pdao = new PecaDAO();
17  %>
18
19  <!DOCTYPE html>
20  <html>
21  <head>
22      <title>Nova Ordem</title>
23      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
24      <link rel="stylesheet" href="../assets/css/style.css">
25      <link rel="stylesheet" href="../assets/css/new.css">
26  </head>
27
28  <body class="gradient-bg">
29      <div class="container">
30          <h2>Criar Nova Ordem de Produção</h2>
31          <form action="../OrdemServlet" method="post">
32              <input type="hidden" name="acao" value="cadastrar">
33
34          <div class="form-group">
35              Número:<br>
36              <input type="text" name="numero" class="form-control" required><br>
37          </div>
38
39          <div class="form-group">
40              Setor:<br>
41              <select name="id_setor" class="form-control">
42                  <% for (Setor s : sdao.listar()) { %>
43                      <option value="<%= s.getId() %>"><%= s.getNome() %</option>
44                  <% } %>
45              </select><br>
46          </div>
47
48          <div class="form-group">

```

Figura74-Novo pt2

```
48 <div class="form-group">
49     Peça:<br>
50     <select name="id_peca" class="form-control">
51         <% for (Peca p : pdao.listar()) { %>
52             <option value="<%= p.getId() %>"><%= p.getNome() %></option>
53         <% } %>
54     </select><br>
55 </div>
56
57 <div class="form-group">
58     Quantidade:<br>
59     <input type="number" name="quantidade" class="form-control" required><br>
60 </div>
61
62 <div class="form-group">
63     Data:<br>
64     <input type="date" name="data" class="form-control" required><br>
65 </div>
66
67 <div class="form-group">
68     Status:<br>
69     <select name="status" class="form-control">
70         <option>Aguardando</option>
71         <option>Produção</option>
72         <option>Finalizado</option>
73     </select>
74     <br>
75 </div>
76     <button class="btn-add">Salvar</button>
77     <a href="listar.jsp" class="btn btn-secondary">Cancelar</a>
78 </form>
79 </div>
80 </body>
81 </html>
```

Figura75-Listar

```

1  <%-- 
2   Document : listar
3   Created on : 24 de nov. de 2025, 20:59:07
4   Author : João Pedro Rovaris
5   | Gustavo Figueiredo
6   | Guilherme Napolitano
7   --%>
8
9  <%@page import="dao.OrdemDAO"%>
10 <%@page import="model.Ordem"%>
11 <%@page import="java.util.List"%>
12 <%@page contentType="text/html" pageEncoding="UTF-8"%>
13
14 <%
15   OrdemDAO dao = new OrdemDAO();
16   List<Ordem> lista = dao.listar();
17 %>
18
19 <!DOCTYPE html>
20 <html>
21 <head>
22   <title>Ordens de Produção</title>
23   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
24   <link rel="stylesheet" href="../assets/css/style.css">
25 </head>
26
27 <body class="gradient-bg">
28
29 <h2 class="mb-4 text-white">Ordens de Produção</h2>
30
31 <a href="novo.jsp" class="btn-add">+ Nova Ordem</a>
32
33 <div class="table-card">
34   <table class="table custom-table table-striped align-middle">
35     <thead>
36       <tr>
37         <th>ID</th>
38         <th>Número</th>
39         <th>Setor</th>
40         <th>Peça</th>
41         <th>Qtd</th>
42         <th>Status</th>
43         <th>Data</th>
44         <th>Ações</th>
45       </tr>
46     </thead>
47     <tbody>
48       <% for (Ordem o : lista) { %>

```

Figura76-Listar pt2

```

1  <!-- 
2   <th>ID</th>
3   <th>Número</th>
4   <th>Setor</th>
5   <th>Peça</th>
6   <th>Qtd</th>
7   <th>Status</th>
8   <th>Data</th>
9   <th>Ações</th>
10  </tr>
11  </thead>
12  <tbody>
13    <% for (Ordem o : lista) { %>
14      <tr>
15        <td><%= o.getId() %></td>
16        <td><%= o.getNumero() %></td>
17        <td><%= o.getSetorNome() %></td>  <!-- AGORA MOSTRA O NOME DO SETOR --&gt;
18        &lt;td&gt;&lt;%= o.getPecaNome() %&gt;&lt;/td&gt;  <!-- AGORA MOSTRA O NOME DA PEÇA --&gt;
19        &lt;td&gt;&lt;%= o.getQuantidade() %&gt;&lt;/td&gt;
20        &lt;td&gt;&lt;%= o.getStatus() %&gt;&lt;/td&gt;
21        &lt;td&gt;&lt;%= o.getData() %&gt;&lt;/td&gt;
22        &lt;td&gt;
23          &lt;a href="editar.jsp?id=&lt;%= o.getId() %&gt;" class="btn btn-primary btn-sm"&gt;Editar&lt;/a&gt;
24          &lt;a href="../OrdemServlet?acao=excluir&amp;id=&lt;%= o.getId() %&gt;" class="btn btn-danger btn-sm"&gt;Excluir&lt;/a&gt;
25        &lt;/td&gt;
26      &lt;/tr&gt;
27    &lt;% } %&gt;
28  &lt;/tbody&gt;
29 &lt;/table&gt;
30 &lt;/div&gt;
31
32 &lt;/body&gt;
33 &lt;/html&gt;
</pre>

```

Figura78-Editar

```

1  <%--  
2   Document : editar  
3   Created on : 24 de nov. de 2025, 20:59:23  
4   Author : João Pedro Rovaris  
5   Gustavo Figueiredo  
6   Guilherme Napolitano  
7 --%>  
8  
9  <%@page import="dao.OrdemDAO"%>  
10 <%@page import="dao.SetorDAO"%>  
11 <%@page import="dao.PecaDAO"%>  
12 <%@page import="model.Ordem"%>  
13 <%@page import="model.Setor"%>  
14 <%@page import="model.Peca"%>  
15 <%@page contentType="text/html" pageEncoding="UTF-8"%>  
16  
17  <%  
18  int id = Integer.parseInt(request.getParameter("id"));  
19  OrdemDAO odao = new OrdemDAO();  
20  SetorDAO sdao = new SetorDAO();  
21  PecaDAO pdao = new PecaDAO();  
22  
23  Ordem ordem = odao.buscarPorId(id);  
24  %>  
25  
26  <!DOCTYPE html>  
27  <html>  
28  <head>  
29      <title>Editar Ordem</title>  
30      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">  
31      <link rel="stylesheet" href="../assets/css/style.css">  
32      <link rel="stylesheet" href="../assets/css/new.css">  
33  </head>  
34  <body class="gradient-bg">  
35  
36  <div class="container">  
37      <h2>Editar Ordem de Produção</h2>  
38      <form method="post" action="../OrdemServlet">  
39          <input type="hidden" name="acao" value="atualizar">  
40          <input type="hidden" name="id" value="<%= ordem.getId() %>">  
41  
42          <div class="form-group">  
43              Número:<br>  
44              <input type="text" class="form-control" name="numero" value="<%= ordem.getNumero() %>" required><br>  
45          </div>  
46  
47          <div class="form-group">  
48              Setor:<br>

```

Figura79-Editar pt2

```

48
49 Setor:<br>
50 <select name="id_setor" class="form-control">
51   <% for (Setor s : sdao.listar()) { %>
52     <option value="<%= s.getId() %>" 
53       <%= (s.getId() == ordem.getId_setor()) ? "selected" : "" %>>
54     <%= s.getNome() %>
55   <% } %>
56 </option>
57 </select><br>
58 </div>
59
60 <div class="form-group">
61   Peca:<br>
62   <select name="id_peca" class="form-control">
63     <% for (Peca p : pdao.listar()) { %>
64       <option value="<%= p.getId() %>" 
65         <%= (p.getId() == ordem.getId_peca()) ? "selected" : "" %>>
66       <%= p.getNome() %>
67     <% } %>
68   </select><br>
69 </div>
70
71 <div class="form-group">
72   Quantidade:<br>
73   <input type="number" name="quantidade" class="form-control" value="<%= ordem.getQuantidade() %>" required><br>
74 </div>
75
76 <div class="form-group">
77   Status:<br>
78   <select name="status" class="form-control">
79     <option <%= ordem.getStatus().equals("Aguardando") ? "selected" : "" %>>Aguardando</option>
80     <option <%= ordem.getStatus().equals("Produção") ? "selected" : "" %>>Produção</option>
81     <option <%= ordem.getStatus().equals("Finalizado") ? "selected" : "" %>>Finalizado</option>
82   </select><br>
83 </div>
84
85 <div class="form-group">
86   Data:<br>
87   <input type="date" class="form-control" name="data" value="<%= ordem.getData() %>" required><br><br>
88 </div>
89   <button class="btn-add">Salvar Alterações</button>
90   <a href="listar.jsp" class="btn btn-secondary">Cancelar</a>
91 </form>
92 </div>
93 </body>
94 </html>
95

```

Ativ
Acess

4.1.6 JSP Funcionário

Figura80-Novo

```

1  <!--
2   Document : novo
3   Created on : 26 de nov. de 2025, 22:58:23
4   Author : João Rovaris
5   |           Gustavo Figueiredo
6   |           Guilherme Napolitano
7   -->
8
9  <%@page contentType="text/html" pageEncoding="UTF-8"%>
10 <!DOCTYPE html>
11 <html>
12 <head>
13   <title>Novo Funcionário</title>
14   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
15   <link rel="stylesheet" href="../assets/css/style.css">
16   <link rel="stylesheet" href="../assets/css/new.css">
17 </head>
18
19 <body class="gradient-bg">
20 <div class="container">
21   <h2>Cadastro de Funcionário</h2>
22   <form action="../FuncionarioServlet" method="post" enctype="multipart/form-data">
23     <input type="hidden" name="acao" value="cadastrar">
24
25     <div class="form-group">
26       Nome:<br>
27       <input type="text" name="nome" class="form-control" required><br>
28     </div>
29
30     <div class="form-group">
31       CPF:<br>
32       <input type="text" name="cpf" class="form-control" required>
33     </div>
34
35     <div class="form-group">
36       Email:<br>
37       <input type="email" name="email" class="form-control" required>
38     </div>
39
40     <div class="form-group">
41       Telefone:<br>
42       <input type="text" name="telefone" class="form-control">
43     </div>
44
45     <div class="form-group">
46       Cargo:<br>
47       <input type="text" name="cargo" class="form-control">
48     </div>

```

Figura81-Novo pt2

```

48   </div>
49
50   <div class="form-group">
51     Foto:<br>
52     <input type="file" name="foto" class="form-control">
53   </div>
54
55   <button class="btn-add">Salvar</button>
56   <a href="listar.jsp" class="btn btn-secondary">Cancelar</a>
57 </form>
58 </div>
59 </body>
60 </html>

```

Figura82-Listar

```
1  <!--
2   Document    : listar
3   Created on : 26 de nov. de 2025, 22:58:17
4   Author      : João Rovaris
5           | Gustavo Figueiredo
6           | Guilherme Napolitano
7   --%>
8
9  <%@page import="dao.FuncionarioDAO"%>
10 <%@page import="model.Funcionario"%>
11 <%@page import="java.util.List"%>
12 <%@page contentType="text/html" pageEncoding="UTF-8"%>
13
14 <%
15   FuncionarioDAO dao = new FuncionarioDAO();
16   List<Funcionario> lista = dao.listar();
17 %>
18
19 <!DOCTYPE html>
20 <html>
21 <head>
22   <title>Funcionários</title>
23   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
24   <link rel="stylesheet" href="../assets/css/style.css">
25 </head>
26
27 <body class="gradient-bg">
28
29 <h2 class="mb-4 text-white">Gerenciar Funcionários</h2>
30
31 <a href="novo.jsp" class="btn-add">+ Novo Funcionário</a>
32
33
34 <div class="table-card">
35 <table class="table custom-table table-striped align-middle">
36 <thead>
37 <tr>
38   <th>Foto</th>
39   <th>ID</th>
40   <th>Nome</th>
41   <th>CPF</th>
42   <th>Email</th>
43   <th>Telefone</th>
44   <th>Cargo</th>
45   <th>Ações</th>
46 </tr>
47 </thead>
```

Figura83-Listar pt2

```
47 |         </thead>
48 |
49 |         <tbody>
50 |             <% for (Funcionario f : lista) { %>
51 |                 <tr>
52 |                     <td>
53 |                         <% if (f.getFoto() != null) { %>
54 |                             
55 |                         <% } else { %>
56 |                             
57 |                         <% } %>
58 |                     </td>
59 |
60 |                     <td><%= f.getId() %></td>
61 |                     <td><%= f.getNome() %></td>
62 |                     <td><%= f.getCpf() %></td>
63 |                     <td><%= f.getEmail() %></td>
64 |                     <td><%= f.getTelefone() %></td>
65 |                     <td><%= f.getCargo() %></td>
66 |
67 |                     <td>
68 |                         <a href="editar.jsp?id=<%= f.getId() %>" class="btn btn-primary btn-sm">Editar</a>
69 |                         <a href="../FuncionarioServlet?acao=excluir&id=<%= f.getId() %>" 
70 |                             class="btn btn-danger btn-sm"
71 |                             onclick="return confirm('Excluir funcionário?')">
72 |                             Excluir
73 |                         </a>
74 |                     </td>
75 |                 </tr>
76 |             <% } %>
77 |         </tbody>
78 |     </table>
79 | 
```

Figura84-Editar

```

1   <%--  
2       Document : editar  
3       Created on : 26 de nov. de 2025, 22:58:39  
4       Author : João Rovaris  
5               Gustavo Figueiredo  
6               Guilherme Napolitano  
7   --%>  
8   <%@page import="dao.FuncionarioDAO"%>  
9   <%@page import="model.Funcionario"%>  
10  
11  <%  
12      int id = Integer.parseInt(request.getParameter("id"));  
13      FuncionarioDAO dao = new FuncionarioDAO();  
14      Funcionario f = dao.buscarPorId(id);  
15  %>  
16  
17  <!DOCTYPE html>  
18  <html>  
19  <head>  
20      <title>Editar Funcionário</title>  
21      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">  
22      <link rel="stylesheet" href="../assets/css/style.css">  
23      <link rel="stylesheet" href="../assets/css/new.css">  
24  </head>  
25  <body class="gradient-bg">  
26  
27  <div class="container">  
28      <h2>Editar Funcionário</h2>  
29  <form action="../FuncionarioServlet" method="post" enctype="multipart/form-data">  
30      <input type="hidden" name="acao" value="atualizar">  
31      <input type="hidden" name="id" value="<%= f.getId() %>">  
32  
33  <div class="form-group">  
34      <label>Nome</label>  
35      <input type="text" name="nome" class="form-control" value="<%= f.getNome() %>" required>  
36  </div>  
37  
38  <div class="form-group">  
39      <label>CPF</label>  
40      <input type="text" name="cpf" class="form-control" value="<%= f.getCpf() %>" required>  
41  </div>  
42  
43  <div class="form-group">  
44      <label>Email</label>  
45      <input type="email" name="email" class="form-control" value="<%= f.getEmail() %>" required>  
46  </div>  
47  
48  <div class="form-group">

```

Figura85-Editar pt2

```
48 <div class="form-group">
49   <label>Telefone</label>
50   <input type="text" name="telefone" class="form-control" value="<%>= f.getTelefone() %>">
51 </div>
52
53 <div class="form-group">
54   <label>Cargo</label>
55   <input type="text" name="cargo" class="form-control" value="<%>= f.getCargo() %>">
56 </div>
57
58 <div class="form-group">
59   <label>Foto Atual</label><br>
60   <% if (f.getFoto_base64() != null) { %>
61     
62   <% } else { %>
63     <span>Sem foto</span>
64   <% } %>
65 </div>
66
67 <div class="form-group">
68   <label>Nova Foto</label>
69   <input type="file" name="foto" class="form-control">
70 </div>
71 <button class="btn-add">Salvar</button>
72 <a href="listar.jsp" class="btn btn-secondary">Cancelar</a>
73 </form>
74 </div>
75 </body>
76 </html>
```

4.5. CSS

Figura86-Style.css

```
1  /*
2   *          Created on : 27 de nov. de 2025, 21:25:57
3   *          Author      : João Pedro Rovaris
4   *                      Gustavo Figueiredo
5   *                      Guilherme Napolitano
6   */
7  /*
8   * =====
9   * DASHBOARD - FUNDO AZUL
10  ===== */
11
12 .index-bg {
13     background: linear-gradient(135deg, #0d6efd, #6ea8fe) !important;
14     min-height: 100vh;
15     padding-top: 40px;
16 }
17
18 /*
19  * =====
20  * CARD DO DASHBOARD
21  * ===== */
22 .dashboard-card {
23     max-width: 750px;
24     background: #ffffff;
25     border-radius: 18px;
26     padding: 40px;
27     margin: auto;
28     box-shadow: 0 6px 28px rgba(0,0,0,0.20);
29     animation: fadeIn .4s ease-in-out;
30     text-align: center;
31 }
32
33 /*
34  * =====
35  * BOTÕES DO DASHBOARD
36  * ===== */
37 .dashboard-btn {
38     display: inline-block;
39     background: #ffffff;
40     color: #0d6efd !important;
41     padding: 14px 26px;
42     margin: 10px;
43     border-radius: 12px;
44     border: 2px solid #ffffff;
45     font-weight: 600;
46     font-size: 15px;
47     transition: 0.25s ease;
48     box-shadow: 0 4px 20px rgba(255,255,255,0.2);
```

Figura87-Style.css pt2

```
49 L }
50
51 .dashboard-btn:hover {
52     background: #0d6efd;
53     color: #fffffd !important;
54     border-color: #0d6efd;
55     transform: translateY(-3px);
56 }
57
58 /* =====
59    NAVBAR PADRONIZADA
60 ===== */
61
62 .navbar {
63     box-shadow: 0 3px 20px rgba(0,0,0,0.25);
64 }
65
66 .navbar-brand {
67     font-size: 20px;
68     font-weight: 700;
69 }
70
71 /* Botões do topo */
72 .navbar .btn-outline-light {
73     border-radius: 8px;
74 }
75
76 .navbar .btn-danger {
77     border-radius: 8px;
78 }
79
80 .custom-navbar {
81     width: 100%;
82     background: #0ble39; /* azul petróleo escuro */
83     padding: 14px 40px;
84     display: flex;
85     justify-content: space-between;
86     align-items: center;
87     box-shadow: 0 4px 20px rgba(0,0,0,0.35);
88     position: fixed;
89     top: 0;
90     left: 0;
91     z-index: 20;
92 }
93
94 .custom-navbar .navbar-brand {
95     color: #fff;
96     font-weight: 700;
```

Figura88-Style.css pt3

```
97 } font-size: 20px;
98 }
99
100 .nav-btn-outline {
101 border: 1px solid #cf9ff;
102 color: #cf9ff;
103 padding: 6px 14px;
104 border-radius: 8px;
105 font-size: 14px;
106 transition: 0.3s;
107 }
108
109 .nav-btn-outline:hover {
110 background: #cf9ff;
111 color: #0ble39 !important;
112 }
113
114 .nav-btn-danger {
115 background: #cf3535;
116 color: white;
117 padding: 6px 14px;
118 border-radius: 8px;
119 transition: 0.3s;
120 }
121
122 .nav-btn-danger:hover {
123 background: #a82828;
124 }
125
126 .dashboard-wrapper {
127 margin-top: 140px; /* empurra para baixo da navbar */
128 display: flex;
129 justify-content: center;
130 }
131
132 .dashboard-card {
133 background: #ffffff;
134 padding: 40px;
135 border-radius: 18px;
136 width: 750px;
137 box-shadow: 0 10px 35px rgba(0,0,0,0.25);
138 text-align: center;
139 animation: fadeIn 0.4s ease-in-out;
140 }
141
142 .cargo-text {
143 color: #555;
144 margin-bottom: 25px;
```

Figura89-Style.css pt4

```
144     margin-bottom: 25px;
145 }
146
147 /* Agora os botões aparecem */
148 .dashboard-buttons {
149   display: flex;
150   flex-direction: column;
151   gap: 12px;
152 }
153
154 .dashboard-btn {
155   background: #0d6efd;
156   color: white !important;
157   padding: 12px;
158   border-radius: 12px;
159   font-weight: 600;
160   transition: 0.25s;
161 }
162
163 .dashboard-btn:hover {
164   background: #084fc7;
165   transform: translateY(-2px);
166 }
167
168 .gradient-bg {
169   background: linear-gradient(135deg, #0d6efd, #6aa0ff);
170   min-height: 100vh;
171   padding-top: 120px;
172 }
173
174 .listar-container {
175   margin-top: 40px;
176   max-width: 1000px;
177 }
178
179 .table-card {
180   background: #fff;
181   padding: 30px;
182   border-radius: 16px;
183   box-shadow: 0 8px 30px rgba(0,0,0,0.25);
184 }
185
186 .table-title {
187   text-align: center;
188   font-weight: 700;
189   margin-bottom: 25px;
190   color: #0ble39;
191 }
```

Figura90-Style.css pt5

```
191 }  
192  
193 .custom-table thead {  
194     background: #0ble39;  
195     color: #fff;  
196 }  
197  
198 .custom-table tbody tr:hover {  
199     background: rgba(0,0,0,0.05);  
200 }  
201  
202 .btn-add {  
203     display: inline-block;  
204     margin: 10px 0 20px 0;  
205     color: #fff;  
206     text-decoration: none;  
207     font-weight: 500;  
208     font-size: 16px;  
209     background: #086952;  
210     padding: 10px 18px;  
211     border-radius: 8px;  
212      backdrop-filter: blur(4px);  
213     transition: 0.3s ease;  
214 }  
215  
216 .btn-add:hover {  
217     background: rgba(255, 255, 255, 0.25);  
218     padding-left: 22px;  
219 }  
220
```

Figura91-New.css

```
1  /*
2   * Created on : 27 de nov. de 2025, 21:25:57
3   * Author     : João Pedro Rovaris
4   *             | Gustavo Figueiredo
5   *             | Guilherme Napolitano
6   */
7
8  body {
9      font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, sans-serif;
10     background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
11     min-height: 100vh;
12     display: flex;
13     justify-content: center;
14     align-items: center;
15     padding: 20px;
16 }
17
18 .container {
19     background: white;
20     border-radius: 16px;
21     box-shadow: 0 20px 60px rgba(0, 0, 0, 0.3);
22     max-width: 500px;
23     width: 100%;
24     padding: 40px;
25 }
26
27 h1 {
28     color: #2d3748;
29     font-size: 28px;
30     margin-bottom: 32px;
31     text-align: center;
32     font-weight: 600;
33 }
34
35 .form-group {
36     margin-bottom: 24px;
37 }
38
39 label {
40     display: block;
41     color: #4a5568;
42     font-size: 14px;
43     font-weight: 500;
44     margin-bottom: 8px;
45 }
46
47 input[type="text"],
48 input[type="number"],
```

Figura92-New.css pt2

```
49  textarea {
50      width: 100%;
51      padding: 12px 16px;
52      border: 2px solid #e2e8f0;
53      border-radius: 8px;
54      font-size: 15px;
55      color: #2d3748;
56      transition: all 0.3s ease;
57      font-family: inherit;
58  }
59
60      input[type="text"]:focus,
61      input[type="number"]:focus,
62  textarea:focus {
63      outline: none;
64      border-color: #667eea;
65      box-shadow: 0 0 0 3px rgba(102, 126, 234, 0.1);
66  }
67
68  textarea {
69      min-height: 100px;
70      resize: vertical;
71  }
72
73  .file-input-wrapper {
74      position: relative;
75      display: inline-block;
76      width: 100%;
77  }
78
79  input[type="file"] {
80      width: 100%;
81      padding: 12px 16px;
82      border: 2px dashed #e2e8f0;
83      border-radius: 8px;
84      font-size: 14px;
85      color: #4a5568;
86      cursor: pointer;
87      transition: all 0.3s ease;
88  }
89
90  input[type="file"]:hover {
91      border-color: #667eea;
92      background-color: #f7fafc;
93  }
94
95  button {
96      width: 100%;
```

Figura93-New.css pt3

```
96      width: 100%;  
97      padding: 14px 24px;  
98      background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);  
99      color: white;  
100     border: none;  
101     border-radius: 8px;  
102     font-size: 16px;  
103     font-weight: 600;  
104     cursor: pointer;  
105     transition: all 0.3s ease;  
106     margin-top: 8px;  
107 }  
108  
109 button:hover {  
110     transform: translateY(-2px);  
111     box-shadow: 0 10px 20px rgba(102, 126, 234, 0.3);  
112 }  
113  
114 button:active {  
115     transform: translateY(0);  
116 }  
117
```

■ Script Banco de Dados

Figura94- Criação Database e Tabelas

```
1 • CREATE DATABASE IF NOT EXISTS sistema_producao;
2 • USE sistema_producao;
3
4 -- -----
5 -- TABELA USUÁRIOS
6 --
7 • CREATE TABLE usuarios (
8     id INT AUTO_INCREMENT PRIMARY KEY,
9     nome VARCHAR(100) NOT NULL,
10    email VARCHAR(100) UNIQUE NOT NULL,
11    senha VARCHAR(255) NOT NULL,
12    cargo VARCHAR(50) NOT NULL,
13    foto VARCHAR(255),
14    foto_base64 LONGTEXT
15 );
16
17 -- -----
18 -- TABELA SETORES
19 --
20 • CREATE TABLE setores (
21     id INT AUTO_INCREMENT PRIMARY KEY,
22     nome VARCHAR(100) NOT NULL,
23     descricao TEXT,
24     foto VARCHAR(255),
25     foto_base64 LONGTEXT
26 );
27
```

Figura95- Criação Database e Tabelas pt-2

```

27
28      -- -----
29      -- TABELA PEÇAS
30      -- -----
31 • CREATE TABLE pecas (
32         id INT AUTO_INCREMENT PRIMARY KEY,
33         nome VARCHAR(100) NOT NULL,
34         peso DOUBLE NOT NULL,
35         descricao TEXT,
36         foto VARCHAR(255),
37         foto_base64 LONGTEXT
38     );
39
40      -- -----
41      -- TABELA ORDENS DE PRODUÇÃO
42      -- -----
43 • CREATE TABLE ordens (
44         id INT AUTO_INCREMENT PRIMARY KEY,
45         numero VARCHAR(50) NOT NULL,
46         id_setor INT NOT NULL,
47         id_peca INT NOT NULL,
48         quantidade INT NOT NULL,
49         status VARCHAR(50),
50         data DATE,
51         FOREIGN KEY (id_setor) REFERENCES setores(id) ON DELETE CASCADE,
52         FOREIGN KEY (id_peca) REFERENCES pecas(id) ON DELETE CASCADE
53     );
54

```

■ Evidências do Database

Figura96- Usuários

| Showing rows 0 - 1 (2 total, Query took 0.0001 seconds.) | | | | | | | | |
|---|------------------------------------|-------------------------------|---------------------------------|-------------------------------|---------------------------------|---------------------------------|----------|-------------|
| SELECT * FROM `usuarios` | | | | | | | | |
| <input type="checkbox"/> Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh] | | | | | | | | |
| | Show all | Number of rows: | 25 | Filter rows: | Search this table | Sort by key: | None | ▼ |
| Extra options | | | | | | | | |
| ← → | ▼ | id | nome | email | senha | cargo | foto | foto_base64 |
| <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 1 Administrador | admin@empresa.com | 123 | admin | NULL NULL |
| <input type="checkbox"/> | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | 2 João Silva | joao@empresa.com | 123 | operador | NULL NULL |
| ↑ | <input type="checkbox"/> Check all | With selected: | <input type="checkbox"/> Edit | <input type="checkbox"/> Copy | <input type="checkbox"/> Delete | <input type="checkbox"/> Export | | |

Figura97- Setores

| SELECT * FROM `setores` | | | | | | |
|---|--|-----------------|----|--------------|-------------------|---|
| <input type="checkbox"/> Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh] | | | | | | |
| <input type="checkbox"/> Show all | | Number of rows: | 25 | Filter rows: | Search this table | Sort by key: |
| Extra options | | | | | | |
| <input type="checkbox"/> | | | | 1 | Corte | Setor responsável pelo corte inicial das peças NULL NULL |
| <input type="checkbox"/> | | | | 2 | Montagem | Setor onde é feita a montagem final dos componentes... NULL NULL |
| <input type="checkbox"/> | | | | | | |
| <input type="checkbox"/> Show all | | Number of rows: | 25 | Filter rows: | Search this table | Sort by key: |
| Query results operations | | | | | | |
| | | | | | | |

Figura98- Peças

| SELECT * FROM `pecas` | | | | | | |
|---|--|-----------------|----|--------------|-------------------|--|
| <input type="checkbox"/> Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh] | | | | | | |
| <input type="checkbox"/> Show all | | Number of rows: | 25 | Filter rows: | Search this table | Sort by key: |
| Extra options | | | | | | |
| <input type="checkbox"/> | | | | 1 | Parafuso m10 | 0.5 Parafuso m10 NULL IVBORw0KGgoAAAANSUhEUgAAJ0AAAHDCAYAAADP4av1AAAAAX... |
| <input type="checkbox"/> | | | | 2 | Chapa A36 | 2.5 Chapa de aço carbono A36 NULL IVBORw0KGgoAAAANSUhEUgAAAtQAAAFwCAYAACYSOyAAAAAX... |
| <input type="checkbox"/> | | | | | | |
| <input type="checkbox"/> Show all | | Number of rows: | 25 | Filter rows: | Search this table | Sort by key: |

■ Evidências Sistema

Figura98- Erro Tela de Login

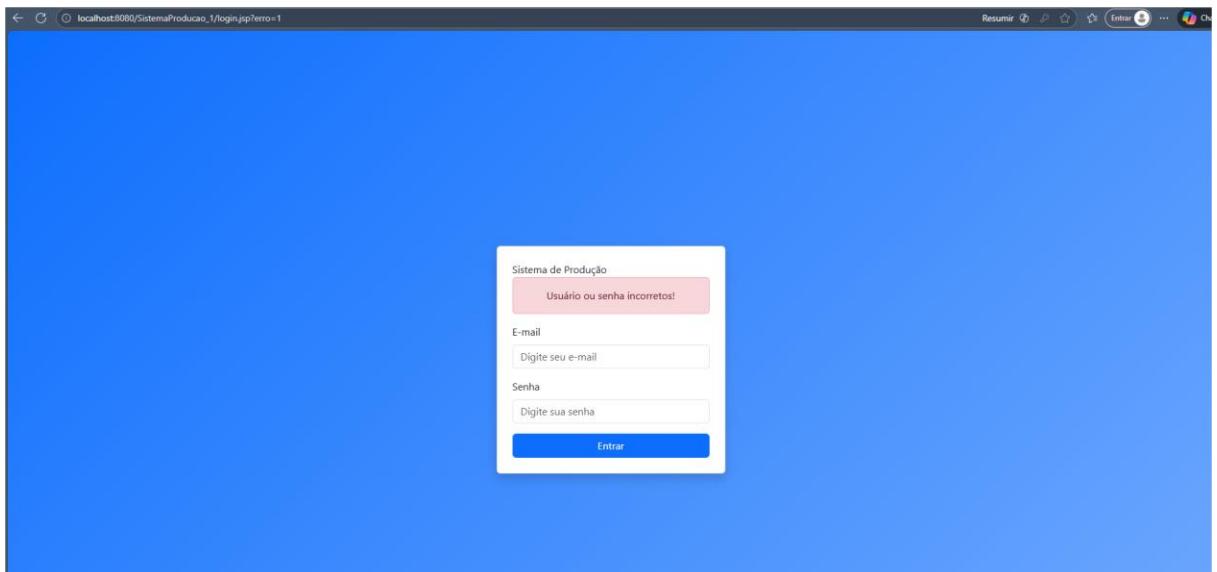


Figura99- Login ok, Dashboard inicial

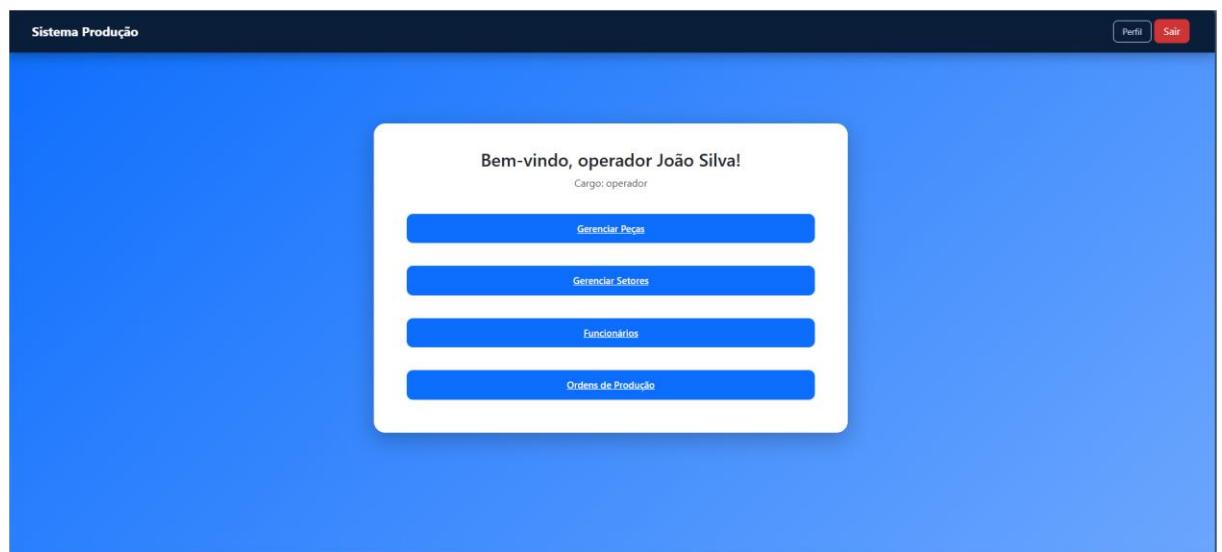


Figura100- Atualização Perfil do Usuário

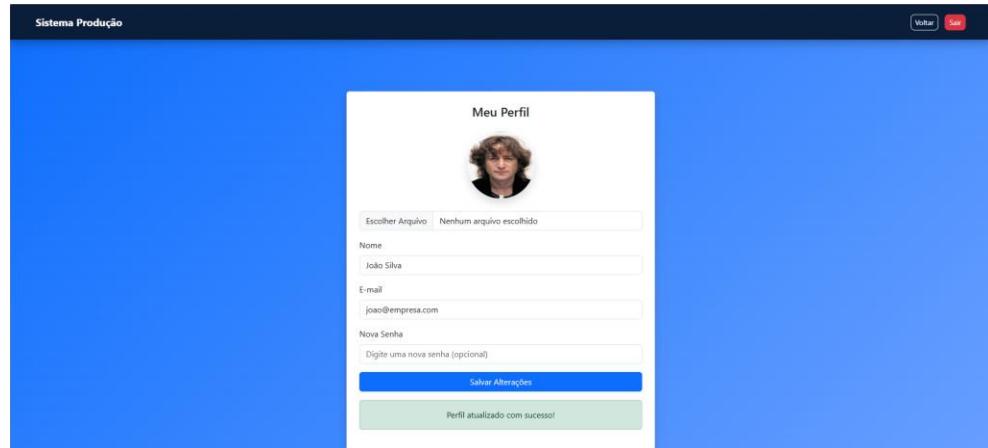


Figura101- Inserção Peça Nova

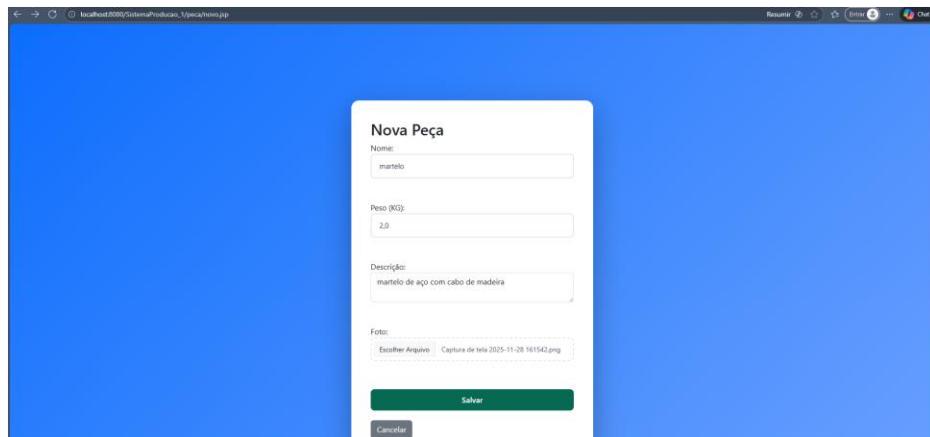


Figura102- Peça nova inserida ok

| Lista de Peças | | | | | |
|---|----|--------------|------|------------------------------------|--|
| Foto | ID | Nome | Peso | Descrição | Ações |
|  | 1 | Parafuso m10 | 0.5 | Parafuso m10 | <button>Editar</button> <button>Excluir</button> |
|  | 2 | Chapa A36 | 2.5 | Chapa de aço carbono A36 | <button>Editar</button> <button>Excluir</button> |
|  | 3 | martelo | 2.0 | martelo de aço com cabo de madeira | <button>Editar</button> <button>Excluir</button> |

Figura103- Edição de peça existente

localhost:8080/SistemaProducao_1/peca/editar.jsp?id=1

Resumir Entrar

Editar Peça

Nome:

Peso (KG):

Descrição:

Foto atual: 

Alterar foto: Nenhum arquivo escolhido

Salvar Alterações Cancelar

Ativar o Windows
Windows Configuration and Control Center

Figura104- Edição ok

| Foto | ID | Nome | Peso | Descrição | Ações |
|------|----|---------------|------|------------------------------------|--|
| | 1 | Parafuso m150 | 0.2 | Parafuso m10, atualiza??o de teste | <button>Editar</button> <button>Excluir</button> |
| | 2 | Chapa A36 | 2.5 | Chapa de aço carbono A36 | <button>Editar</button> <button>Excluir</button> |
| | 3 | martelo | 2.0 | martelo de aço com cabo de madeira | <button>Editar</button> <button>Excluir</button> |

■ Link GitHub

<https://github.com/Rovaris-j/Atividade-Adilson-M2>

■ Referências

ALDRICH, Jonathan; CHAMBERS, Craig; NOTKIN, David. Architectural reasoning in ArchJava. In: EUROPEAN CONFERENCE ON OBJECT ORIENTED PROGRAMMING (ECOOP), 16., 2002. Anais [...]. Disponível em: <https://dl.acm.org/doi/10.5555/646159.680031>. Acesso em: 12 set. 2025. KUCHANA, Partha. Software architecture design patterns in Java. Boca Raton: Auerbach Publications, 2004. Disponível <https://www.routledge.com/Software-Architecture-Design-Patterns-in-Java/Kuchana/p/book/9780849321429>. Acesso em: 12 set. 2025. em: 22 in WANG, Ying et al. An empirical study of usages, updates and risks of third-party libraries Java projects. 2020. Disponível <https://arxiv.org/abs/2002.11028>. Acesso em: 12 set. 2025.