

UNIVERSIDADE DE MOGI DAS CRUZES
ANÁLISE E DESINVOLVIMENTO DE SISTEMAS

HENRY WALLACE DE PAULA CHAGAS
RGM: 11241103570

Projeto Final: Atividade - M2
Controlador de Ordens de Serviços

SÃO PAULO

2025

SÚMARIO

1 .	Explicando a dor da empresa GrupoGPS	3
2 .	Caso de uso	3
3 .	Códigos e prints	4
4 .	Banco de dados	27
5 .	Link repositório GitHub	28

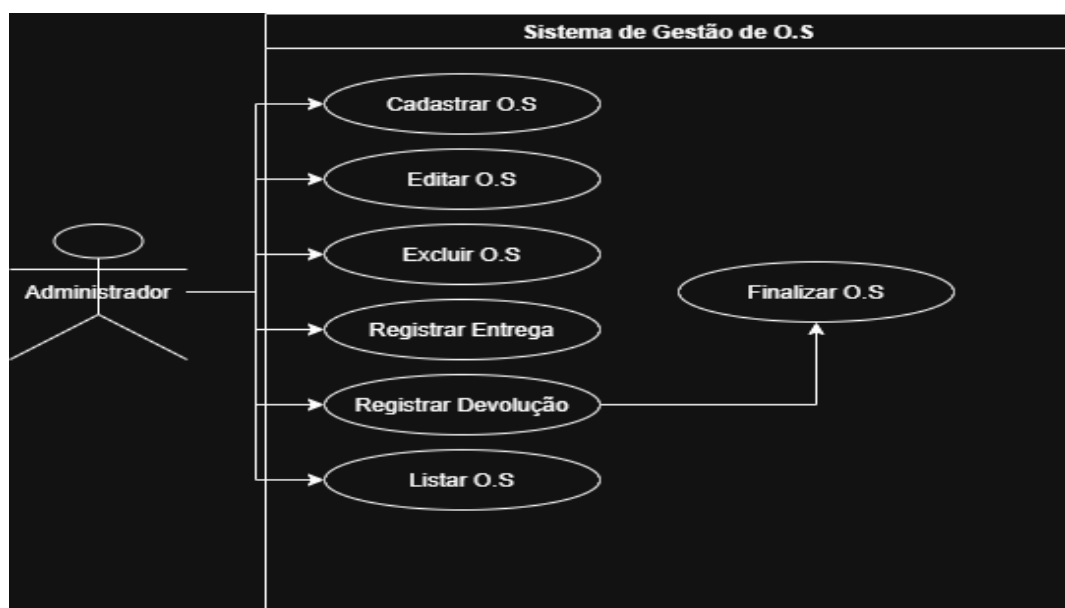
1. Explicando a dor da empresa GrupoGPS

A minha ideia parte de um problema dentro da empresa a qual trabalho. Para uma maior compreensão irei descrever a função da minha empresa que é contratada como terceirizada no hospital ICESP, a empresa é a Grupo GPS. Ela exerce algumas funções no hospital, as duas que envolve tal problema é a manutenção e a disque manutenção - que, querendo ou não faz parte da manutenção. Ao abrir um chamado de manutenção para algo (função do disque) é gerando o que chamamos de O.S (Ordem de serviço), então, é impressa uma folha com todas as informações necessárias para o atendimento do serviço, algumas das informações são: O número da ordem de serviço (o que seria equivalente a um id), o nome do solicitante, o local (Andar e numeração da porta), o grupo de trabalho (que seria os responsáveis por exercer tal serviço, então seria por exemplo: Hidráulica, elétrica, manutenção civil, serralheria, marcenaria, entre outros), o dano (que seria uma descrição do que ocorreu, ou o que está danificado), o ramal (que seria um conjunto numérico composto por 4 números, usado para entrar em contato com o solicitante através de um telefone local).

E o problema está justamente na hora da entregar a O.S para o prestador de serviço realizar, porque muitas das vezes não sabemos se de fato o colaborador pegou a folha impressa para assim realizar o chamado e não ficar pendente.

O sistema consiste em uma pessoa ficar responsável pela entrega das O.S mão a mão e ir atualizando no sistema aquilo que foi entregue e a sua devolutiva.

2. Caso de uso



3. Código no NetBeans (Fonte)

Index.jsp:

The image displays a web browser window and a code editor side-by-side. The browser window shows the 'Sistema GPS - Manutenção ICES' application running on 'localhost:8080/ProjGPS/'. The page features a blue header with the 'GRUPO GPS' logo and the title 'Bem-vindo ao Sistema de O.S. (GPS)'. Below the header, it prompts the user to 'Escolha uma opção:' and lists three menu items: 'Abrir Nova O.S.', 'Consultar e Finalizar O.S.', and 'Gerenciar Grupos de Trabalho (CRUD)'. The code editor shows the source code of 'index.jsp', which includes a document header with metadata (index, created on 20 de nov. de 2025, 17:18:46, author Henry Wallace) and an HTML structure with a title 'Sistema GPS - Manutenção ICES', a link to 'styleGeral.css', and a body containing the same header, logo, title, and menu items as seen in the browser.

Sistema GPS - Manutenção ICES

localhost:8080/ProjGPS/

GRUPO GPS Bem-vindo ao Sistema de O.S. (GPS)

Escolha uma opção:

- [Abrir Nova O.S.](#)
- [Consultar e Finalizar O.S.](#)
- [Gerenciar Grupos de Trabalho \(CRUD\)](#)

index.jsp

```
<!--
2
Document    : index
3
Created on  : 20 de nov. de 2025, 17:18:46
4
Author      : Henry Wallace
5
-->
6
7
<%@page contentType="text/html" pageEncoding="UTF-8"%>
8
<!DOCTYPE html>
9
<html>
10
  <head>
11
    <title>Sistema GPS - Manutenção ICES</title>
12
    <link rel="stylesheet" href="styleGeral.css">
13
  </head>
14
  <body>
15
    <div class="header">
16
      <div class="logo-area">
17
        
18
      </div>
19
      <h1>Bem-vindo ao Sistema de O.S. (GPS)</h1>
20
    </div>
21
    <p>Escolha uma opção:</p>
22
    <ul>
23
      <li><a href="cadastroOS.jsp">Abrir Nova O.S.</a></li>
24
      <li><a href="consultarOS.jsp">Consultar e Finalizar O.S.</a></li>
25
      <li><a href="gerenciarGrupos.jsp">Gerenciar Grupos de Trabalho (CRUD)</a></li>
26
    </ul>
27
  </body>
28
</html>
```

Cadastro.jsp:

GPS Cadastro de O.S.

Número O.S. (Impressa):

Solicitante:

Local (Andar):

Local (Porta):

Ramal:

Grupo de Trabalho:
Elétrica ▼

Descrição do Dano:

O.S. Impressa Entregue Agora?
☐

Salvar Ordem de Serviço

Cadastro de Ordem de Serviço

localhost:8080/ProjGPS/ca... Resumir

GRUPO GPS

Cadastro de O.S.

Número O.S. (Impressa):

1234

Solicitante:

Henry Wallace

Local (Andar):

10

Local (Porta):

23

Ramal:

3452

Grupo de Trabalho:

Elétrica

Descrição do Dano:

Lâmpada queimada

O.S. Impressa Entregue Agora?

☐

Salvar Ordem de Serviço

```
index.jsp x cadastroOS.jsp x
Source History
2 <%--
3     Document : cadastroOS
4     Created on : 20 de nov. de 2025, 16:20:45
5     Author : Henry Wallace
6 --%>
7 <%@page import="com.gps.dao.GrupoTrabalhoDAO"%>
8 <%@page import="com.gps.model.GrupoTrabalho"%>
9 <%@page import="com.gps.dao.OrdemServicoDAO"%>
10 <%@page import="com.gps.model.OrdemServico"%>
11 <%@page import="java.util.Date"%>
12 <%@page import="java.util.List"%>
13 <%@page contentType="text/html" pageEncoding="UTF-8"%>
14 <%
15     String acao = request.getParameter("acao");
16     String msg = "";
17
18     if ("cadastrar".equals(acao)) {
19         try {
20             OrdemServico os = new OrdemServico();
21             OrdemServicoDAO osDAO = new OrdemServicoDAO();
22
23             // Recebe os parâmetros do formulário
24             os.setSolicitante(request.getParameter("solicitante"));
25             os.setLocalAndar(Integer.parseInt(request.getParameter("localAndar")));
26             os.setLocalPorta(request.getParameter("localPorta"));
27             os.setDanoDescricao(request.getParameter("danoDescricao"));
28             os.setRamal(request.getParameter("ramal"));
29             os.setNumeroOSImpressa(request.getParameter("numeroOSImpressa"));
30
31             // Trata o Grupo (Chave Estrangeira)
32             int idGrupo = Integer.parseInt(request.getParameter("idGrupo_fk"));
33             GrupoTrabalho grupo = new GrupoTrabalho();
34             grupo.setIdGrupo(idGrupo);
35             os.setGrupo(grupo);
36
37             // Controle de Entrega (Preenchimento inicial)
38             boolean entregue = request.getParameter("entregue") != null;
39             os.setEntregue(entregue);
40
41             if (entregue) {
42                 os.setHoraEntrega(new Date()); // Define a hora de entrega como a hora atual
43             } else {
44                 os.setHoraEntrega(null); // Se não foi entregue, fica null
45             }
46
47             // Persiste no banco de dados
48             osDAO.create(os);
```

```

        response.sendRedirect("consultarOS.jsp?sucesso=1");
        return; // Interrompe a execução do restante do JSP para fazer o redirecionamento

    } catch (Exception e) {
        msg = "Erro ao cadastrar O.S.: " + e.getMessage();
    }
}

// Dados necessários para popular Grupos de Trabalho
GrupoTrabalhoDAO grupoDAO = new GrupoTrabalhoDAO();
List<GrupoTrabalho> listaGrupos = grupoDAO.readAll();
%>
<!DOCTYPE html>
<html>
    <head>
        <title>Cadastro de Ordem de Serviço</title>
        <link rel="stylesheet" href="styleGeral.css">
    </head>
    <body>
        <div class="header">
            <div class="logo-area">
                
            </div>
            <h1>Cadastro de O.S.</h1>
        </div>
        <% if (!"".equals(msg)) { %> <p style="color: red;"><%= msg %></p> <% } %>

        <form action="cadastroOS.jsp?acao=cadastrar" method="POST">

            <label>Número O.S. (Impressa):</label>
            <input type="text" name="numeroOSImpressa" required><br>

            <label>Solicitante:</label>
            <input type="text" name="solicitante" required><br>

            <label>Local (Andar):</label>
            <input type="number" name="localAndar" required><br>

            <label>Local (Porta):</label>
            <input type="text" name="localPorta" required><br>

            <label>Ramal:</label>
            <input type="text" name="ramal" maxlength="4"><br>

            <label>Grupo de Trabalho:</label>
            <select name="idGrupo_fk" required>
                <% for (GrupoTrabalho g : listaGrupos) { %>
                    <option value="<%= g.getIdGrupo() %>"><%= g.getNomeGrupo() %></option>
                <% } %>
            </select><br>

            <label>Descrição do Dano:</label>
            <textarea name="danoDescricao"></textarea><br>

            <label>O.S. Impressa Entregue Agora?</label>
            <input type="checkbox" name="entregue" value="true"><br>

            <button type="submit">Salvar Ordem de Serviço</button>
        </form>
    </body>
</html>

```

Consultar.jsp

Consulta de Ordens de Serviço

localhost:8080/ProjGPS/consultarOS.jsp

Chat

<div><div>GRUPO GPS</div><div>Ordens de Serviço em Aberto e Finalizadas</div><div>Cadastrar Nova O.S.</div></div>									
ID	Número O.S.	Grupo	Solicitante	Local	Ramal	Entregue?	Hora Entrega	Hora Devolução	Ação
2	1234	Elétrica	Henry Wallace	10/23	3452	Sim	Fri Nov 28 23:19:31 BRT 2025	Pendente	Finalizar

```
index.jsp x cadastroOS.jsp x consultarOS.jsp x
Source History
<!--
2 Document : consultarOS
3 Created on : 20 de nov. de 2025, 16:29:23
4 Author : Henry Wallace
5 -->
6
7 <%@page import="com.gps.dao.GrupoTrabalhoDAO"%>
8 <%@page import="com.gps.model.OrdemServico"%>
9 <%@page import="com.gps.dao.OrdemServicoDAO"%>
10 <%@page import="java.util.List"%>
11 <%@page contentType="text/html" pageEncoding="UTF-8"%>
12 <%
13     OrdemServicoDAO osDAO = new OrdemServicoDAO();
14     // Método para buscar a lista de OS com o nome do Grupo
15     List<OrdemServico> listaOS = osDAO.readAll(); // Implementar com JOIN
16
17     String sucesso = request.getParameter("sucesso");
18     if (sucesso != null && sucesso.equals("1")) {
19         out.println("<p style='color: green;'>Operação realizada com sucesso!</p>");
20     }
21 %>
22 <!DOCTYPE html>
23 <html>
24 <head>
25 <title>Consulta de Ordens de Serviço</title>
26 <link rel="stylesheet" href="styleGeral.css">
27 </head>
28 <body>
29 <div class="header">
30 <div class="logo-area">
31 
32 </div>
33 <h1>Ordens de Serviço em Aberto e Finalizadas</h1>
34 </div>
35 <a href="cadastroOS.jsp">Cadastrar Nova O.S.</a>
36 <br><br>
37
```

```

37
38
39
40
41
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
69
70
71
72
73
74
75
76
77
78
79
<table border="1">
  <thead>
    <tr>
      <th>ID</th>
      <th>Número O.S.</th>
      <th>Grupo</th>
      <th>Solicitante</th>
      <th>Local</th>
      <th>Ramal</th>
      <th>Entregue?</th>
      <th>Hora Entrega</th>
      <th>Hora Devolução</th>
      <th>Ação</th>
    </tr>
  </thead>
  <tbody>
    <% for (OrdemServico os : listaOS) { %>
      <tr>
        <td><%= os.getIdOS() %></td>
        <td><%= os.getNumeroOSImpressa() %></td>
        <td><%= os.getGrupo().getNomeGrupo() %></td>
        <td><%= os.getSolicitante() %></td>
        <td><%= os.getLocalAndar() %>/<%= os.getLocalPorta() %></td>
        <td><%= os.getRamal() %></td>
        <td><%= os.isEntregue() ? "Sim" : "Não" %></td>
        <td><%= os.getHoraEntrega() != null ? os.getHoraEntrega() : "Pendente" %></td>
        <td style="color: <%= os.getHoraDevolucao() != null ? "blue" : "red" %>;">
          <%= os.getHoraDevolucao() != null ? os.getHoraDevolucao() : "Pendente" %>
        </td>
        <td>
          <% if (os.getHoraDevolucao() == null) { %>
            <a href="finalizarOS.jsp?id=<%= os.getIdOS() %>">Finalizar</a>
          <% } else { %>
            Finalizado
          <% } %>
        </td>
      </tr>
    <% } %>
  </tbody>
</table>
</body>
</html>

```

FinalizarOS.jsp:

Finalizar Ordem de Serviço #2

Resumir

Grupo GPS Finalizar O.S. #2

Grupo: Elétrica

Local: 10 / 23

Descrição: Lâmpada queimada

Esta O.S. ainda está **PENDENTE** de devolução.

[Confirmar Devolução e Finalizar Serviço Agora](#)

[Voltar para a Consulta](#)

```
index.jsp x cadastroOS.jsp x consultarOS.jsp x finalizarOS.jsp x
Source History
Document : finalizarOS
Created on : 20 de nov. de 2025, 16:34:55
Author : Henry Wallace

<!--
-->

<%@page import="com.gps.dao.OrdemServicoDAO"%>
<%@page import="com.gps.model.OrdemServico"%>
<%@page import="java.util.Date"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%
// Recebe o ID da OS a ser finalizada
int idOS = Integer.parseInt(request.getParameter("id"));
OrdemServicoDAO osDAO = new OrdemServicoDAO();
OrdemServico os = osDAO.readById(idOS); // Você deve implementar este método

// Lógica de Ação (POST)
String acao = request.getParameter("acao");
if ("devolver".equals(acao)) {

    // Se ainda não foi entregue, precisamos registrar a entrega antes de devolver
    if (!os.isEntregue()) {
        os.setEntregue(true);
        os.setHoraEntrega(new Date());
    }

    os.setHoraDevolucao(new Date()); // Define a hora da devolução como a hora atual

    osDAO.updateDevolucao(os);

    response.sendRedirect("consultarOS.jsp?sucesso=1");
    return;
}
%>
```

```
<!DOCTYPE html>
<html>
<head>
<title>Finalizar Ordem de Serviço #<%= idOS %></title>
<link rel="stylesheet" href="styleGeral.css">
</head>
<body>
<div class="header">
<div class="logo-area">

</div>
<h1>Finalizar O.S. #<%= idOS %></h1>
</div>

<p> **Grupo:** <%= os.getGrupo().getNomeGrupo() %></p>
<p> **Local:** <%= os.getLocalAndar() %> / <%= os.getLocalPorta() %></p>
<p> **Descrição:** <%= os.getDanoDescricao() %></p>

<% if (os.getHoraDevolucao() == null) { %>
<p style="color: red;">Esta O.S. ainda está **PENDENTE** de devolução.</p>

<form action="finalizarOS.jsp?id=<%= idOS %>&acao=devolver" method="POST">
<button type="submit">Confirmar Devolução e Finalizar Serviço Agora</button>
</form>
<% } else { %>
<p style="color: blue;">Esta O.S. foi finalizada em: <%= os.getHoraDevolucao() %></p>
<% } %>

<br><a href="consultarOS.jsp">Voltar para a Consulta</a>
</body>
</html>
```

Consulta de Ordens de Serviço

localhost:8080/ProjGPS/consultarOS.jsp?sucess...

Operação realizada com sucesso!

GRUPO GPS

Ordens de Serviço em Aberto e Finalizadas

[Cadastrar Nova O.S.](#)

ID	Número O.S.	Grupo	Solicitante	Local	Ramal	Entregue?	Hora Entrega	Hora Devolução	Ação
2	1234	Elétrica	Henry Wallace	10/23	3452	Sim	Fri Nov 28 23:19:31 BRT 2025	Fri Nov 28 23:25:05 BRT 2025	Finalizado

GerenciarGrupos.jsp:

Gerenciamento de Grupos de Trabalho

localhost:8080/ProjGPS/gerencia... Resumir

[← Voltar para a Consulta de O.S.](#)

GRUPO GPS

Gerenciamento de Grupos de Trabalho

[Cadastrar Novo Grupo](#)

Nome do Grupo (Ex: Elétrica, Hidráulica):

[Adicionar Grupo](#)

[Grupos Cadastrados \(5\)](#)

ID	Nome do Grupo	Ações
1	Elétrica	Excluir
2	Hidráulica	Excluir
3	Manutenção Civil	Excluir
5	Marcenaria	Excluir
4	Serralheria	Excluir

```
index.jsp x cadastroOS.jsp x consultarOS.jsp x finalizarOS.jsp x gerenciarGrupos.jsp x
Source History
Document : gerenciarGrupos
Created on : 20 de nov. de 2025, 17:06:34
Author : Henry Wallace
<!-->
<%@page import="com.gps.dao.GrupoTrabalhoDAO"%>
<%@page import="com.gps.model.GrupoTrabalho"%>
<%@page import="java.util.List"%>
<%@page import="java.sql.SQLException"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%
// Controller: Lógica para Inserir e Deletar Grupos

GrupoTrabalhoDAO grupoDAO = new GrupoTrabalhoDAO();
String acao = request.getParameter("acao");
String msg = null;

try {
    if ("cadastrar".equals(acao)) {
        String nomeGrupo = request.getParameter("nomeGrupo");
        if (nomeGrupo != null && !nomeGrupo.trim().isEmpty()) {
            GrupoTrabalho novoGrupo = new GrupoTrabalho();
            novoGrupo.setNomeGrupo(nomeGrupo.trim());
            grupoDAO.create(novoGrupo); // Chama o método de inserção
            msg = "Grupo '" + nomeGrupo + "' cadastrado com sucesso!";
        } else {
            msg = "O nome do grupo não pode ser vazio.";
        }
    } else if ("deletar".equals(acao)) {
        int idGrupo = Integer.parseInt(request.getParameter("id"));
        grupoDAO.delete(idGrupo); // Chama o método de exclusão
        msg = "Grupo excluído com sucesso!";
    }
} catch (SQLException ex) {
    // Trata exceções do banco de dados
    if (ex.getMessage().contains("a foreign key constraint fails")) {
        msg = "ERRO: Não é possível excluir o grupo, pois há Ordens de Serviço (O.S.) vinculadas a ele.";
    } else {
        msg = "ERRO no Banco de Dados: " + ex.getMessage();
    }
} catch (Exception ex) {
    msg = "ERRO: " + ex.getMessage();
}
}
```

```

// Lista todos os grupos após as ações
List<GrupoTrabalho> listaGrupos = null;
try {
    listaGrupos = grupoDAO.readAll(); // Busca todos os grupos para exibição
} catch (SQLException ex) {
    // Se houver erro ao listar, exibe a mensagem, mas não interrompe
    msg = (msg != null) ? msg + " | Erro ao listar: " + ex.getMessage() : "Erro ao listar grupos: " + ex.getMessage();
}
<%>
<!DOCTYPE html>
<html>
<head>
<title>Gerenciamento de Grupos de Trabalho</title>
<link rel="stylesheet" href="styleGeral.css">
<style>
    .mensagem { padding: 10px; margin-bottom: 15px; border-radius: 4px; }
    .sucesso { background-color: #d4edda; color: #155724; border: 1px solid #c3e6cb; }
    .erro { background-color: #f8d7da; color: #721c24; border: 1px solid #f5c6cb; }
    table { border-collapse: collapse; width: 50%; }
    th, td { border: 1px solid #ddd; padding: 8px; text-align: left; }
    th { background-color: #f2f2f2; }
</style>
</head>
<body>
<div class="header">
<div class="logo-area">

</div>
<h1>Gerenciamento de Grupos de Trabalho</h1>
</div>

<p><a href="consultarOS.jsp">- Voltar para a Consulta de O.S.</a></p>

<%
// Exibe a mensagem de feedback
if (msg != null) {
    String tipo = msg.startsWith("ERRO") ? "erro" : "sucesso";
%>
<div class="mensagem" <%= tipo %>>
<%= msg %>
</div>
<%
}
%>
```

```

92         <hr>
93
94         <h2>Cadastrar Novo Grupo</h2>
95         <form action="gerenciarGrupos.jsp?acao=cadastrar" method="POST">
96             <label for="nomeGrupo">Nome do Grupo (Ex: Elétrica, Hidráulica):</label><br>
97             <input type="text" id="nomeGrupo" name="nomeGrupo" required style="width: 300px;"><br><br>
98             <button type="submit">Adicionar Grupo</button>
99         </form>
100
101         <hr>
102
103         <h2>Grupos Cadastrados (<%= (listaGrupos != null ? listaGrupos.size() : 0) %>)</h2>
104         <% if (listaGrupos != null && !listaGrupos.isEmpty()) { %>
105             <table>
106                 <thead>
107                     <tr>
108                         <th>ID</th>
109                         <th>Nome do Grupo</th>
110                         <th>Ações</th>
111                     </tr>
112                 </thead>
113                 <tbody>
114                     <% for (GrupoTrabalho grupo : listaGrupos) { %>
115                         <tr>
116                             <td><%= grupo.getIdGrupo() %></td>
117                             <td><%= grupo.getNomeGrupo() %></td>
118                             <td>
119                                 <a href="gerenciarGrupos.jsp?acao=deletar&id=<%= grupo.getIdGrupo() %>"
120                                   onclick="return confirm('Tem certeza que deseja excluir o grupo <%= grupo.getNomeGrupo() %>? Se houver O.S. vinculada, a exclusão falhará.')"
121                                   style="color: red;">
122                                     Excluir
123                                 </a>
124                             </td>
125                         </tr>
126                     <% } %>
127                 </tbody>
128             </table>
129         <% } else if (listaGrupos != null) { %>
130             <p>Não há grupos de trabalho cadastrados.</p>
131         <% } %>
132     </body>
133 </html>

```

StyleGeral:

```

index.jsp x cadastroOS.jsp x consultarOS.jsp x finalizarOS.jsp x gerenc
Source History
1  /*
2      Created on : 20 de nov. de 2025, 20:59:03
3      Author      : Henry Wallace
4  */
5
6      /* Estilo do corpo da página */
7  body {
8      font-family: 'Arial', sans-serif;
9      background-color: #f4f7f6; /* Cinza bem claro */
10     color: #333;
11     margin: 0;
12     padding: 0;
13     display: flex;
14     flex-direction: column;
15     align-items: center;
16 }
17
18 /* Container principal para centralizar o conteúdo */
19 .container {
20     width: 80%;
21     max-width: 1000px;
22     background-color: #fff;
23     padding: 30px;
24     margin-top: 20px;
25     box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
26     border-radius: 8px;
27 }
28
29 /* Estilo do cabeçalho com espaço para a logo */
30 .header {
31     display: flex;
32     justify-content: space-between;
33     align-items: center;
34     margin-bottom: 20px;
35     border-bottom: 2px solid #ddd;
36     padding-bottom: 10px;
37 }

```

```

39      /* Área da logo (canto superior esquerdo) */
40      .logo-area {
41          width: 150px;
42          height: 50px;
43          background-color: #007bff; /* Cor para identificar o espaço */
44          color: white;
45          text-align: center;
46          line-height: 50px;
47          font-weight: bold;
48          border-radius: 4px;
49      }
50
51      .logo-area img {
52          max-width: 100%;
53          max-height: 100%;
54          height: auto;
55      }
56
57      /* Estilo de títulos */
58      h1 {
59          color: #007bff;
60          font-size: 2em;
61          margin: 0;
62      }
63
64      h2 {
65          color: #0056b3;
66          border-bottom: 1px dashed #eee;
67          padding-bottom: 5px;
68          margin-top: 30px;
69      }
70
71      /* Estilo de formulários */
72      form label {
73          display: block;
74          margin-top: 15px;
75          font-weight: bold;
76      }
77

```

```

78      form input[type="text"],
79      form input[type="number"],
80      form select,
81      form textarea {
82          width: 100%;
83          padding: 10px;
84          margin-top: 5px;
85          border: 1px solid #ccc;
86          border-radius: 4px;
87          box-sizing: border-box; /* Garante que padding não estoure a largura */
88      }
89
90      form textarea {
91          resize: vertical;
92          min-height: 80px;
93      }
94
95      /* Estilo de botões */
96      button, .button-link {
97          background-color: #007bff;
98          color: white;
99          padding: 10px 20px;
100         border: none;
101         border-radius: 4px;
102         cursor: pointer;
103         font-size: 1em;
104         margin-top: 20px;
105         text-decoration: none;
106         display: inline-block;
107         text-align: center;
108     }
109
110     button:hover, .button-link:hover {
111         background-color: #0056b3;
112     }
113
114     /* Estilo de links de ação (dentro da tabela, por exemplo) */
115     a {
116         color: #007bff;
117         text-decoration: none;
118     }
119
120     a:hover {
121         text-decoration: underline;
122     }
123

```



```

124      /* Estilo de tabelas */
125      table {
126          width: 100%;
127          margin-top: 20px;
128          border-collapse: collapse;
129      }
130
131      th, td {
132          border: 1px solid #ddd;
133          padding: 12px;
134          text-align: left;
135      }
136
137      th {
138          background-color: #f2f2f2;
139          color: #333;
140          font-weight: bold;
141      }
142
143      tr:nth-child(even) {
144          background-color: #f9f9f9; /* Linhas zebradas */
145      }
146
147      /* Estilo para mensagens de feedback */
148      .mensagem { padding: 10px; margin: 15px 0; border-radius: 4px; }
149      .sucesso { background-color: #d4edda; color: #155724; border: 1px solid #c3e6cb; }
150      .erro { background-color: #f8d7da; color: #721c24; border: 1px solid #f5c6cb; }

```

Conexão.java:

```

1      package com.gps.dao;
2
3      /**
4       *
5       * @author Henry Wallace
6       */
7
8      import java.sql.Connection;
9      import java.sql.DriverManager;
10     import java.sql.PreparedStatement;
11     import java.sql.ResultSet;
12     import java.sql.SQLException;
13     import java.sql.Statement;
14
15     public class Conexao {
16         // Configurações do seu banco de dados
17         private static final String DRIVER = "com.mysql.cj.jdbc.Driver";
18         private static final String URL = "jdbc:mysql://localhost:3307/ordensdeservico";
19         private static final String USER = "root";
20         private static final String PASS = "";
21
22         // Método principal para obter a conexão
23         public static Connection getConnection() {
24             try {
25                 Class.forName(DRIVER);
26                 return DriverManager.getConnection(URL, USER, PASS);
27             } catch (ClassNotFoundException ex) {
28                 throw new RuntimeException("Driver do banco de dados não encontrado: ", ex);
29             } catch (SQLException ex) {
30                 throw new RuntimeException("Erro na conexão com o BD: ", ex);
31             }
32         }
33
34         // Métodos de fechamento de recursos
35
36         public static void closeConnection(Connection con) {
37             if (con != null) {
38                 try {
39                     con.close();
40                 } catch (SQLException ex) {
41                     System.err.println("Erro ao fechar conexão: " + ex);
42                 }
43             }
44         }
45     }

```

```

46     public static void closeConnection(Connection con, PreparedStatement stmt) {
47         closeConnection(con);
48         if (stmt != null) {
49             try {
50                 stmt.close();
51             } catch (SQLException ex) {
52                 System.err.println("Erro ao fechar Statement: " + ex);
53             }
54         }
55     }
56
57     public static void closeConnection(Connection con, PreparedStatement stmt, ResultSet rs) {
58         closeConnection(con, stmt);
59         if (rs != null) {
60             try {
61                 rs.close();
62             } catch (SQLException ex) {
63                 System.err.println("Erro ao fechar ResultSet: " + ex);
64             }
65         }
66     }
67 }

```

GrupoDeTrabalhoDAO:

```

1  package com.gps.dao;
2
3  /**
4   *
5   * @author Henry Wallace
6   */
7
8  import com.gps.model.GrupoTrabalho;
9  import java.sql.Connection;
10 import java.sql.PreparedStatement;
11 import java.sql.ResultSet;
12 import java.sql.SQLException;
13 import java.util.ArrayList;
14 import java.util.List;
15
16 public class GrupoTrabalhoDAO {
17
18     // Create
19     public void create(GrupoTrabalho grupo) throws SQLException {
20         String sql = "INSERT INTO GrupoTrabalho (nomeGrupo) VALUES (?)";
21         Connection con = null;
22         PreparedStatement stmt = null;
23
24         try {
25             con = Conexao.getConnection();
26             stmt = con.prepareStatement(sql);
27             stmt.setString(1, grupo.getNomeGrupo());
28
29             stmt.executeUpdate();
30
31         } catch (SQLException ex) {
32             throw new SQLException("Erro ao cadastrar Grupo de Trabalho: " + ex.getMessage(), ex);
33         } finally {
34             Conexao.closeConnection(con, stmt);
35         }
36     }
37
38     // ReadALL
39     public List<GrupoTrabalho> readAll() throws SQLException {
40         String sql = "SELECT * FROM GrupoTrabalho ORDER BY nomeGrupo";
41         List<GrupoTrabalho> grupos = new ArrayList<>();
42         Connection con = null;
43         PreparedStatement stmt = null;
44         ResultSet rs = null;
45     }

```

```

6   try {
7       con = Conexao.getConnection();
8       stmt = con.prepareStatement(sql);
9       rs = stmt.executeQuery();
10
11       while (rs.next()) {
12           GrupoTrabalho grupo = new GrupoTrabalho();
13           grupo.setIdGrupo(rs.getInt("idGrupo"));
14           grupo.setNomeGrupo(rs.getString("nomeGrupo"));
15           grupos.add(grupo);
16       }
17   } catch (SQLException ex) {
18       throw new SQLException("Erro ao listar grupos: " + ex.getMessage(), ex);
19   } finally {
20       Conexao.closeConnection(con, stmt, rs);
21   }
22   return grupos;
23 }
24
25 // Delete
26 public void delete(int idGrupo) throws SQLException {
27     String sql = "DELETE FROM GrupoTrabalho WHERE idGrupo = ?";
28     Connection con = null;
29     PreparedStatement stmt = null;
30
31     try {
32         con = Conexao.getConnection();
33         stmt = con.prepareStatement(sql);
34         stmt.setInt(1, idGrupo);
35
36         stmt.executeUpdate();
37
38     } catch (SQLException ex) {
39         throw new SQLException("Erro ao deletar Grupo de Trabalho: " + ex.getMessage(), ex);
40     } finally {
41         Conexao.closeConnection(con, stmt);
42     }
43 }
44
45

```

OrdemServiçoDAO:

```

1 package com.gps.dao;
2
3 /**
4  *
5  * @author Henry Wallace
6  */
7
8 import com.gps.model.OrderService;
9 import com.gps.model.GrupoTrabalho;
10 import java.sql.Connection;
11 import java.sql.PreparedStatement;
12 import java.sql.ResultSet;
13 import java.sql.SQLException;
14 import java.sql.Timestamp;
15 import java.util.ArrayList;
16 import java.util.Date;
17 import java.util.List;
18
19 public class OrdemServicoDAO {
20
21     // Create (Cadastro de Nova O.S.)
22     public void create(OrdemServico os) throws SQLException {
23         String sql = "INSERT INTO OrdemServico (idGrupo_fk, solicitante, localAndar, localPorta, danoDescricao, ramal, entrega, horaEntrega, numeroOSImpressa) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)";
24         Connection con = null;
25         PreparedStatement stmt = null;
26
27         try {
28             con = Conexao.getConnection();
29             stmt = con.prepareStatement(sql);
30
31             stmt.setInt(1, os.getGrupo().getIdGrupo());
32             stmt.setString(2, os.getSolicitante());
33             stmt.setInt(3, os.getLocalAndar());
34             stmt.setString(4, os.getLocalPorta());
35             stmt.setString(5, os.getDanoDescricao());
36             stmt.setString(6, os.getRamal());
37             stmt.setBoolean(7, os.isEntrega());
38             // Converte java.util.Date para java.sql.Timestamp (para compatibilidade com DATETIME do MySQL)
39             if (os.getHoraEntrega() != null) {
40                 stmt.setTimestamp(8, new Timestamp(os.getHoraEntrega().getTime()));
41             } else {
42                 stmt.setTimestamp(8, null);
43             }
44             stmt.setString(9, os.getNumeroOSImpressa());
45
46             stmt.executeUpdate();
47
48         } catch (SQLException ex) {
49             throw new SQLException("Erro ao salvar a Ordem de Serviço: " + ex.getMessage(), ex);
50         } finally {
51             Conexao.closeConnection(con, stmt);
52         }
53     }
54
55     // ReadAll (Consulta Completa com JOIN)
56     public List<OrdemServico> readAll() throws SQLException {
57         // SQL com JOIN para buscar o nome do grupo
58         String sql = "SELECT OS.*, GT.nomeGrupo " +
59             "FROM OrdemServico OS " +
60             "INNER JOIN GrupoTrabalho GT ON OS.idGrupo_fk = GT.idGrupo " +
61             "ORDER BY OS.idOS DESC";
62
63         List<OrdemServico> ordens = new ArrayList<>();
64         Connection con = null;
65         PreparedStatement stmt = null;
66         ResultSet rs = null;
67
68         try {
69             con = Conexao.getConnection();
70             stmt = con.prepareStatement(sql);
71             rs = stmt.executeQuery();
72
73             while (rs.next()) {
74                 OrdemServico os = new OrdemServico();
75
76                 // Popula os dados da O.S.
77                 os.setIdOS(rs.getInt("idOS"));
78                 os.setSolicitante(rs.getString("solicitante"));
79                 os.setLocalAndar(rs.getInt("localAndar"));
80                 os.setLocalPorta(rs.getString("localPorta"));
81                 os.setDanoDescricao(rs.getString("danoDescricao"));
82                 os.setRamal(rs.getString("ramal"));
83                 os.setEntrega(rs.getBoolean("entrega"));
84                 os.setNumeroOSImpressa(rs.getString("numeroOSImpressa"));
85
86                 // Trata as datas de Entrega
87                 Timestamp tsEntrega = rs.getTimestamp("horaEntrega");
88                 if (tsEntrega != null) {
89                     os.setHoraEntrega(new Date(tsEntrega.getTime()));
90                 }
91
92                 // Trata as datas de Devolução
93                 Timestamp tsDevolucao = rs.getTimestamp("horaDevolucao");
94                 if (tsDevolucao != null) {
95                     os.setHoraDevolucao(new Date(tsDevolucao.getTime()));
96                 }
97             }
98         }
99     }
100 }

```

```

96
97         // Popula o objeto Grupo de Trabalho (Chave Estrangeira)
98         GrupoTrabalho grupo = new GrupoTrabalho();
99         grupo.setIdGrupo(rs.getInt("idGrupo_fk"));
100         grupo.setNomeGrupo(rs.getString("nomeGrupo"));
101         os.setGrupo(grupo);
102
103         ordens.add(os);
104     }
105 } catch (SQLException ex) {
106     throw new SQLException("Erro ao listar Ordens de Serviço: " + ex.getMessage(), ex);
107 } finally {
108     Conexao.closeConnection(con, stmt, rs); // Fechamento seguro
109 }
110 return ordens;
111 }
112
113 // ReadById (Busca uma única O.S. para Finalização)
114 public OrdemServico readById(int idOS) throws SQLException {
115     String sql = "SELECT OS.*, GT.nomeGrupo " +
116                 "FROM OrdemServico OS " +
117                 "INNER JOIN GrupoTrabalho GT ON OS.idGrupo_fk = GT.idGrupo " +
118                 "WHERE OS.idOS = ?";
119
120     Connection con = null;
121     PreparedStatement stmt = null;
122     ResultSet rs = null;
123     OrdemServico os = null;
124
125     try {
126         con = Conexao.getConnection();
127         stmt = con.prepareStatement(sql);
128         stmt.setInt(1, idOS);
129         rs = stmt.executeQuery();
130
131         if (rs.next()) {
132             os = new OrdemServico();
133             // Popule o objeto 'os' (mesmo padrão do readAll)
134             os.setIdOS(rs.getInt("idOS"));
135             os.setSolicitante(rs.getString("solicitante"));
136             os.setLocalAndar(rs.getInt("localAndar"));
137             os.setLocalPorta(rs.getString("localPorta"));
138             os.setDanoDescricao(rs.getString("danoDescricao"));
139             os.setRamal(rs.getString("ramal"));
140             os.setEntregue(rs.getBoolean("entregue"));
141

```

```

142 // Trata as datas
143 Timestamp tsEntrega = rs.getTimestamp("horaEntrega");
144 if (tsEntrega != null) { os.setHoraEntrega(new Date(tsEntrega.getTime())); }
145 Timestamp tsDevolucao = rs.getTimestamp("horaDevolucao");
146 if (tsDevolucao != null) { os.setHoraDevolucao(new Date(tsDevolucao.getTime())); }
147
148 // Popula o objeto Grupo
149 GrupoTrabalho grupo = new GrupoTrabalho();
150 grupo.setIdGrupo(rs.getInt("idGrupo_fk"));
151 grupo.setNomeGrupo(rs.getString("nomeGrupo"));
152 os.setGrupo(grupo);
153 }
154 } catch (SQLException ex) {
155     throw new SQLException("Erro ao buscar a O.S. por ID: " + ex.getMessage(), ex);
156 } finally {
157     Conexao.closeConnection(con, stmt, rs); // Fechamento seguro
158 }
159 return os;
160 }
161
162 // Update (Finalização/Devolução da O.S.)
163 public void updateDevolucao(OrdemServico os) throws SQLException {
164     // Atualiza três campos: entregue, horaEntrega e horaDevolucao
165     String sql = "UPDATE OrdemServico SET entregue = ?, horaEntrega = ?, horaDevolucao = ? WHERE idOS = ?";
166     Connection con = null;
167     PreparedStatement stmt = null;
168
169     try {
170         con = Conexao.getConnection();
171         stmt = con.prepareStatement(sql);
172
173         // Garante que entregue seja TRUE
174         stmt.setBoolean(1, true);
175
176         // Seta a hora de Entrega (se for nula no BD, usamos a data/hora atual para preencher)
177         Date horaEntrega = (os.getHoraEntrega() != null) ? os.getHoraEntrega() : new Date();
178         stmt.setTimestamp(2, new Timestamp(horaEntrega.getTime()));
179
180         // Seta a hora de Devolução
181         stmt.setTimestamp(3, new Timestamp(os.getHoraDevolucao().getTime()));
182
183         // Condição WHERE
184         stmt.setInt(4, os.getIdOS());
185
186         stmt.executeUpdate();
187
188     } catch (SQLException ex) {
189         throw new SQLException("Erro ao finalizar a Ordem de Serviço: " + ex.getMessage(), ex);
190     } finally {
191         Conexao.closeConnection(con, stmt); // Fechamento seguro
192     }
193 }

```

GrupoTrabalho.java:

```

1  package com.gps.model;
2
3  /**
4   *
5   * @author Henry Wallace
6   */
7
8  public class GrupoTrabalho {
9      private int idGrupo;
10     private String nomeGrupo;
11
12     // Construtor vazio
13     public GrupoTrabalho() {}
14
15     // Construtor completo
16     public GrupoTrabalho(int idGrupo, String nomeGrupo) {
17         this.idGrupo = idGrupo;
18         this.nomeGrupo = nomeGrupo;
19     }
20
21     // Getters e Setters
22     public int getIdGrupo() {
23         return idGrupo;
24     }
25
26     public void setIdGrupo(int idGrupo) {
27         this.idGrupo = idGrupo;
28     }
29
30     public String getNomeGrupo() {
31         return nomeGrupo;
32     }
33
34     public void setNomeGrupo(String nomeGrupo) {
35         this.nomeGrupo = nomeGrupo;
36     }
37
38 }

```

OrdemServico.Java:

```
1 package com.gps.model;
2
3 /**
4  *
5  * @author Henry Wallace
6  */
7
8 import java.util.Date;
9
10 public class OrdemServico {
11     private int idOS;
12     private String numeroOSImpressa;
13     private GrupoTrabalho grupo; // Relacionamento com GrupoTrabalho
14     private String solicitante;
15     private int localAndar;
16     private String localPorta;
17     private String danoDescricao;
18     private String ramal;
19     private boolean entregue;
20     private Date horaEntrega;
21     private Date horaDevolucao;
22
23     // Construtor Vazio
24     public OrdemServico() {
25         this.grupo = new GrupoTrabalho(); // Inicializa o objeto
26     }
27
28     // Getters e Setters
29     public int getIdOS() {
30         return idOS;
31     }
32
33     public void setIdOS(int idOS) {
34         this.idOS = idOS;
35     }
36
37     public String getNumeroOSImpressa() {
38         return numeroOSImpressa;
39     }
40
41     public void setNumeroOSImpressa(String numeroOSImpressa) {
42         this.numeroOSImpressa = numeroOSImpressa;
43     }
44
45     public GrupoTrabalho getGrupo() {
46         return grupo;
47     }
```



```

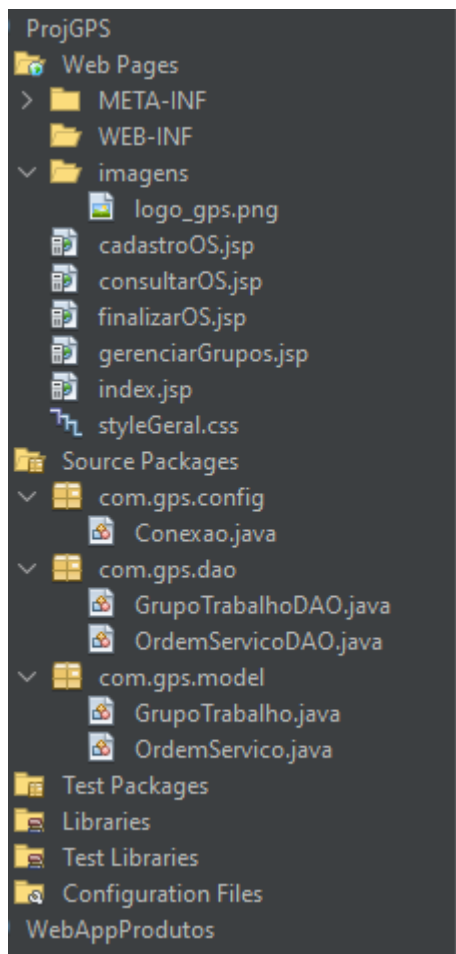
49  [ public void setGrupo(GrupoTrabalho grupo) {
50      this.grupo = grupo;
51  }
52
53  [ public String getSolicitante() {
54      return solicitante;
55  }
56
57  [ public void setSolicitante(String solicitante) {
58      this.solicitante = solicitante;
59  }
60
61  [ public int getLocalAndar() {
62      return localAndar;
63  }
64
65  [ public void setLocalAndar(int localAndar) {
66      this.localAndar = localAndar;
67  }
68
69  [ public String getLocalPorta() {
70      return localPorta;
71  }
72
73  [ public void setLocalPorta(String localPorta) {
74      this.localPorta = localPorta;
75  }
76
77  [ public String getDanoDescricao() {
78      return danoDescricao;
79  }
80
81  [ public void setDanoDescricao(String danoDescricao) {
82      this.danoDescricao = danoDescricao;
83  }
84
85  [ public String getRamal() {
86      return ramal;
87  }
88
89  [ public void setRamal(String ramal) {
90      this.ramal = ramal;
91  }
92
93  [ public boolean isEntregue() {
94      return entregue;
95  }

```

```

96
97  [ public void setEntregue(boolean entregue) {
98      this.entregue = entregue;
99  }
100
101  [ public Date getHoraEntrega() {
102      return horaEntrega;
103  }
104
105  [ public void setHoraEntrega(Date horaEntrega) {
106      this.horaEntrega = horaEntrega;
107  }
108
109  [ public Date getHoraDevolucao() {
110      return horaDevolucao;
111  }
112
113  [ public void setHoraDevolucao(Date horaDevolucao) {
114      this.horaDevolucao = horaDevolucao;
115  }
116
117  }

```



4. Banco de Dados;

Opções extras

					idGrupo	nomeGrupo
<input type="checkbox"/>	Editar	Copiar	Remover		1	Elétrica
<input type="checkbox"/>	Editar	Copiar	Remover		2	Hidráulica
<input type="checkbox"/>	Editar	Copiar	Remover		3	Manutenção Civil
<input type="checkbox"/>	Editar	Copiar	Remover		5	Marcenaria
<input type="checkbox"/>	Editar	Copiar	Remover		4	Serralheria

					idOS	idGrupo_fk	solicitante	localAndar	localPorta	danoDescricao	ramal	entregue	horaEntrega	horaDevolucao	numeroOSImpressa
<input type="checkbox"/>	Editar	Copiar	Remover		1	1	Henry Wallace	10	23	Lâmpada queimada	3452	1	2025-11-28 14:08:32	2025-11-28 14:08:30	NULL
<input type="checkbox"/>	Editar	Copiar	Remover		2	1	Henry Wallace	10	23	Lâmpada queimada	3452	1	2025-11-28 23:19:31	2025-11-28 23:25:05	1234

Foi utilizado chave estrangeira para interligação das tabelas.

5. Link repositório GitHub:

<https://github.com/HenryChagas/ProjetoFinal>