

**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

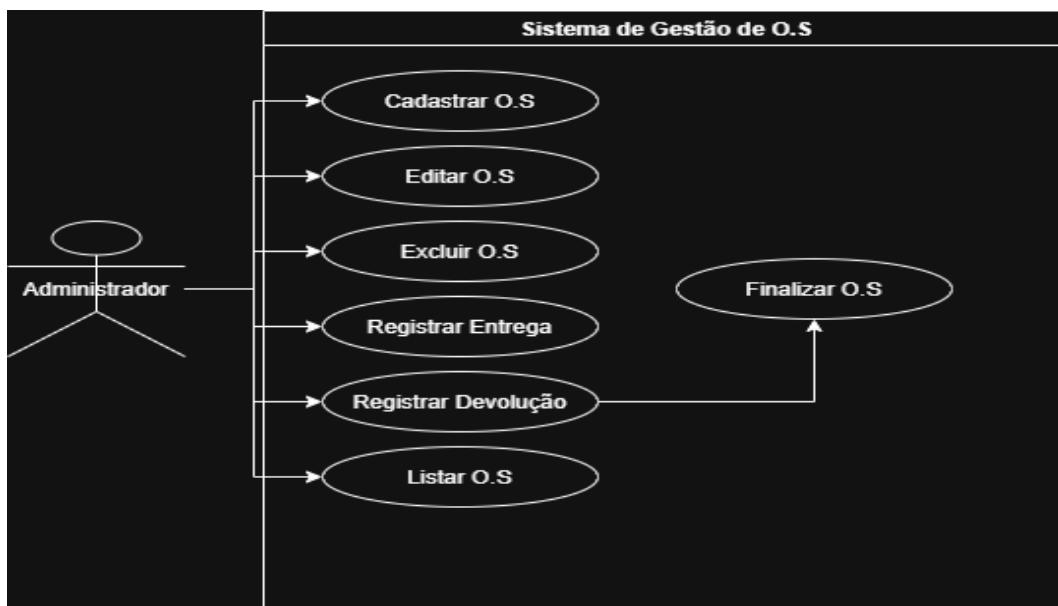
## **1. Explicando a dor da empresa GrupoGPS**

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 screenshot shows a web browser window titled "Sistema GPS - Manutenção ICES" at the URL "localhost:8080/ProjGPS/". The page content is displayed in a blue header bar with the text "GRUPO GPS" and "Bem-vindo ao Sistema de O.S. (GPS)". Below this, a message "Escolha uma opção:" is followed by a bulleted list of three options: "Abrir Nova O.S.", "Consultar e Finalizar O.S.", and "Gerenciar Grupos de Trabalho (CRUD)".

The bottom half of the screenshot shows the source code for "index.jsp" in a code editor. The code is a Java Server Page (JSP) with the following content:

```
<%--  
    Document : index  
    Created on : 20 de nov. de 2025, 17:18:46  
    Author : Henry Wallace  
--%>  
  
<%@page contentType="text/html" pageEncoding="UTF-8"%>  
<!DOCTYPE html>  
<html>  
    <head>  
        <title>Sistema GPS - Manutenção ICES</title>  
        <link rel="stylesheet" href="styleGeneral.css">  
    </head>  
    <body>  
        <div class="header">  
            <div class="logo-area">  
                  
            </div>  
            <h1>Bem-vindo ao Sistema de O.S. (GPS)</h1>  
        </div>  
        <p>Escolha uma opção:</p>  
        <ul>  
            <li><a href="cadastroOS.jsp">Abrir Nova O.S.</a></li>  
            <li><a href="consultarOS.jsp">Consultar e Finalizar O.S.</a></li>  
            <li><a href="gerenciarGrupos.jsp">Gerenciar Grupos de Trabalho (CRUD)</a></li>  
        </ul>  
    </body>  
</html>
```

Cadastro.jsp:

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 🔍 ⭐ ... Ch

## 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**

The screenshot shows a Java IDE interface with the 'cadastroOS.jsp' file open in the 'Source' tab. The code is a JSP page that handles the creation of an 'OrdemServico' object. It imports various DAO and model classes from the 'com.gps' package. The page starts with a header block and then defines an 'OrdemServico' object 'os'. It sets parameters from the request, including 'solicitante', 'localAndar', 'localPorta', 'donoDescricao', 'ramal', and 'numeroOSImpressa'. It then handles the 'idGrupo\_fk' parameter to set the 'GrupoTrabalho' object. The page also initializes the 'entregue' field and its corresponding date. Finally, it persistes the 'os' object using the 'osDAO.create(os)' method.

```
<%--  
    Document      : cadastroOS  
    Created on   : 20 de nov. de 2025, 16:20:45  
    Author       : Henry Wallace  
--%>  
  
<%@page import="com.gps.dao.GrupoTrabalhoDAO"%>  
<%@page import="com.gps.model.GrupoTrabalho"%>  
<%@page import="com.gps.dao.OrdemServicoDAO"%>  
<%@page import="com.gps.model.OrdemServico"%>  
<%@page import="java.util.Date"%>  
<%@page import="java.util.List"%>  
<%@page contentType="text/html" pageEncoding="UTF-8"%>  
<%  
  
String acao = request.getParameter("acao");  
String msg = "";  
  
if ("cadastrar".equals(acao)) {  
    try {  
        OrdemServico os = new OrdemServico();  
        OrdemServicoDAO osDAO = new OrdemServicoDAO();  
  
        // Recebe os parâmetros do formulário  
        os.setSolicitante(request.getParameter("solicitante"));  
        os.setLocalAndar(Integer.parseInt(request.getParameter("localAndar")));  
        os.setLocalPorta(request.getParameter("localPorta"));  
        os.setDonoDescricao(request.getParameter("donoDescricao"));  
        os.setRamal(request.getParameter("ramal"));  
        os.setNumeroOSImpressa(request.getParameter("numeroOSImpressa"));  
  
        // Trata o Grupo (Chave Estrangeira)  
        int idGrupo = Integer.parseInt(request.getParameter("idGrupo_fk"));  
        GrupoTrabalho grupo = new GrupoTrabalho();  
        grupo.setIdGrupo(idGrupo);  
        os.setGrupo(grupo);  
  
        // Controle de Entrega (Preenchimento inicial)  
        boolean entregue = request.getParameter("entregue") != null;  
        os.setEntregue(entregue);  
  
        if (entregue) {  
            os.setHoraEntrega(new Date()); // Define a hora de entrega como a hora atual  
        } else {  
            os.setHoraEntrega(null); // Se não foi entregue, fica null  
        }  
  
        // Persiste no banco de dados  
        osDAO.create(os);  
    } catch (Exception e) {  
        msg = "Ocorreu um erro ao cadastrar a OS: " + e.getMessage();  
    }  
}  
  
%>
```

```

        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="donoDescricao"></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

## 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	Pendente	Finalizar

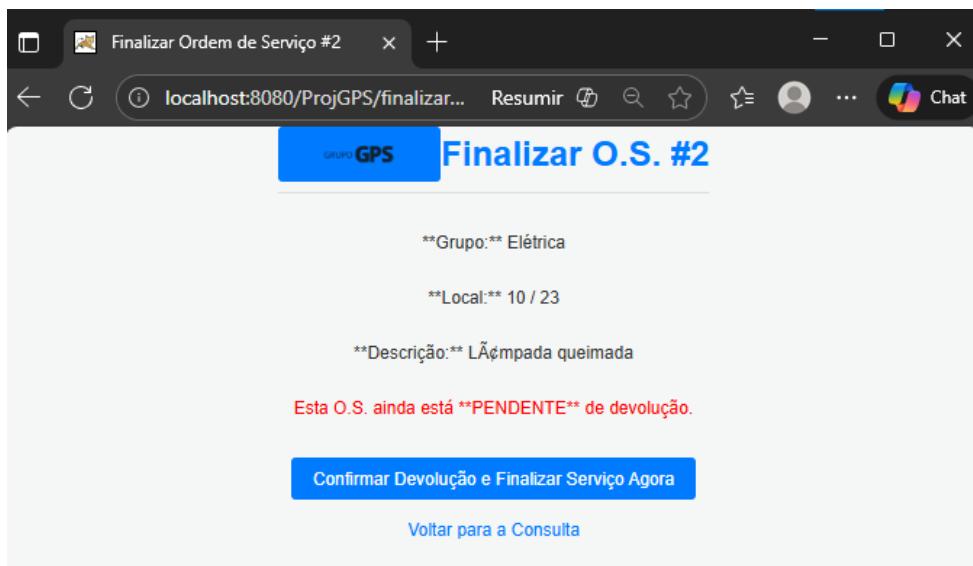


```

37
38         <table border="1">
39             <thead>
40                 <tr>
41                     <th>ID</th>
42                     <th>Número O.S.</th>
43                     <th>Grupo</th>
44                     <th>Solicitante</th>
45                     <th>Local</th>
46                     <th>Ramal</th>
47                     <th>Entregue?</th>
48                     <th>Hora Entrega</th>
49                     <th>Hora Devolução</th>
50                     <th>Ação</th>
51             </tr>
52         </thead>
53         <tbody>
54             <% for (OrdemServico os : listaOS) { %>
55                 <tr>
56                     <td><%= os.getIdOS() %></td>
57                     <td><%= os.getNumeroOSImpressa() %></td>
58                     <td><%= os.getGrupo().getNomeGrupo() %></td>
59                     <td><%= os.getSolicitante() %></td>
60                     <td><%= os.getLocalAndar() %>/<%= os.getLocalPorta() %></td>
61                     <td><%= os.getRamal() %></td>
62                     <td><%= os.isEntregue() ? "Sim" : "Não" %></td>
63                     <td><%= os.getHoraEntrega() != null ? os.getHoraEntrega() : "Pendente" %></td>
64                     <td style="color: <%= os.getHoraDevolucao() != null ? "blue" : "red" %>;">
65                         <%= os.getHoraDevolucao() != null ? os.getHoraDevolucao() : "Pendente" %>
66                     </td>
67                     <td>
68                         <% if (os.getHoraDevolucao() == null) { %>
69                             <a href="finalizarOS.jsp?id=<%= os.getIdOS() %>">Finalizar</a>
70                         <% } else { %>
71                             Finalizado
72                         <% } %>
73                     </td>
74                 </tr>
75             <% } %>
76         </tbody>
77     </table>
78 </body>
79 </html>

```

## FinalizarOS.jsp:



```
index.jsp X cadastroOS.jsp X consultarOS.jsp X finalizarOS.jsp X
Source History |        
<%-->
2     Document      : finalizarOS
3     Created on   : 20 de nov. de 2025, 16:34:55
4     Author        : Henry Wallace
5 --%>
6
7 <%@page import="com.gps.dao.OrdemServicoDAO"%>
8 <%@page import="com.gps.model.OrdemServico"%>
9 <%@page import="java.util.Date"%>
10 <%@page contentType="text/html" pageEncoding="UTF-8"%>
11 <%
12     // Recebe o ID da OS a ser finalizada
13     int idOS = Integer.parseInt(request.getParameter("id"));
14     OrdemServicoDAO osDAO = new OrdemServicoDAO();
15     OrdemServico os = osDAO.readById(idOS); // Você deve implementar este método
16
17     // Lógica de Ação (POST)
18     String acao = request.getParameter("acao");
19     if ("devolver".equals(acao)) {
20
21         // Se ainda não foi entregue, precisamos registrar a entrega antes de devolver
22         if (!os.isEntregue()) {
23             os.setEntregue(true);
24             os.setHoraEntrega(new Date());
25         }
26
27         os.setHoraDevolucao(new Date()); // Define a hora da devolução como a hora atual
28
29         osDAO.updateDevolucao(os);
30
31         response.sendRedirect("consultarOS.jsp?sucesso=1");
32         return;
33     }
34 %>
35
36 <!DOCTYPE html>
37 <html>
38     <head>
39         <title>Finalizar Ordem de Serviço #<%= idOS %></title>
40         <link rel="stylesheet" href="styleGeral.css">
41     </head>
42     <body>
43         <div class="header">
44             <div class="logo-area">
45                 
46             </div>
47             <h1>Finalizar O.S. #<%= idOS %></h1>
48         </div>
49
50         <p> **Grupo:** <%= os.getGrupo().getNomeGrupo() %></p>
51         <p> **Local:** <%= os.getLocalAndar() %> / <%= os.getLocalPorta() %></p>
52         <p> **Descrição:** <%= os.getDanoDescricao() %></p>
53
54         <% if (os.getHoraDevolucao() == null) { %>
55             <p style="color: red;">Esta O.S. ainda está **PENDENTE** de devolução.</p>
56
57             <form action="finalizarOS.jsp?id=<%= idOS %>&acao=devolver" method="POST">
58                 <button type="submit">Confirmar Devolução e Finalizar Serviço Agora</button>
59             </form>
60         <% } else { %>
61             <p style="color: blue;">Esta O.S. foi finalizada em: <%= os.getHoraDevolucao() %></p>
62         <% } %>
63
64         <br><a href="consultarOS.jsp">Voltar para a Consulta</a>
65     </body>
66 </html>
```

Consulta de Ordens de Serviço

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

Operação realizada com sucesso!

## 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

← Voltar para a Consulta de O.S.

### Cadastrar Novo Grupo

Nome do Grupo (Ex: Elétrica, Hidráulica):

**Adicionar Grupo**

### Grupos Cadastrados (5)

ID	Nome do Grupo	Ações
1	Elétrica	<a href="#">Excluir</a>
2	Hidráulica	<a href="#">Excluir</a>
3	Manutenção Civil	<a href="#">Excluir</a>
5	Marcenaria	<a href="#">Excluir</a>
4	Serralheria	<a href="#">Excluir</a>

```
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 à 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: #4CAF50; color: white; border: 1px solid #3EAE3E; }
            .erro { background-color: #F44336; color: white; border: 1px solid #D32F2F; }
            table { border-collapse: collapse; width: 100%; }
            th, td { border: 1px solid #3EAE3E; padding: 8px; text-align: left; }
            th { background-color: #4CAF50; color: white; }
        </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 |
100 |
101 |         <h2>
102 |             Grupos Cadastrados (<%=( listaGrupos != null ? listaGrupos.size() : 0 ) %>)
103 |         <% if (listaGrupos != null && !listaGrupos.isEmpty()) { %>
104 |             <table>
105 |                 <thead>
106 |                     <tr>
107 |                         <th>ID</th>
108 |                         <th>Nome do Grupo</th>
109 |                         <th>Ações</th>
110 |                     </tr>
111 |                 </thead>
112 |                 <tbody>
113 |                     <% for (GrupoOTrabalho grupo : listaGrupos) { %>
114 |                         <tr>
115 |                             <td><%= grupo.getIdGrupo() %></td>
116 |                             <td><%= grupo.getNomeGrupo() %></td>
117 |                             <td>
118 |                                 <a href="gerenciarGrupos.jsp?acao=deletar?id=<%= grupo.getIdGrupo\(\) %>">
119 |                                     onclick="return confirm('Tem certeza que deseja excluir o grupo <%= grupo.getNomeGrupo() %>? Se houver O.S. vinculada, a exclusão falhará.')">
120 |                                         Excluir
121 |                                     </a>
122 |                                 </td>
123 |                             </tr>
124 |                         <% } %>
125 |                     </tbody>
126 |                 </table>
127 |             <% } else if (listaGrupos != null) { %>
128 |                 <p>Não há grupos de trabalho cadastrados.</p>
129 |             <% } %>
130 |         </body>
131 |     </html>
132 |
133 |

```

### StyleGeral:

```

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 /* Contêiner 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 zebreadas */
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 * @author Henry Wallace
5 */
6
7 import java.sql.Connection;
8 import java.sql.DriverManager;
9 import java.sql.PreparedStatement;
10 import java.sql.ResultSet;
11 import java.sql.SQLException;
12 import java.sql.Statement;
13
14 public class Conexao {
15     // Configurações do seu banco de dados
16     private static final String DRIVER = "com.mysql.cj.jdbc.Driver";
17     private static final String URL = "jdbc:mysql://localhost:3307/ordensdeservico";
18     private static final String USER = "root";
19     private static final String PASS = "";
20
21     // Método principal para obter a conexão
22     public static Connection getConnection() {
23         try {
24             Class.forName(DRIVER);
25             return DriverManager.getConnection(URL, USER, PASS);
26         } catch (ClassNotFoundException ex) {
27             throw new RuntimeException("Driver do banco de dados não encontrado: ", ex);
28         } catch (SQLException ex) {
29             throw new RuntimeException("Erro na conexão com o BD: ", ex);
30         }
31     }
32
33     // Métodos de fechamento de recursos
34
35     public static void closeConnection(Connection con) {
36         if (con != null) {
37             try {
38                 con.close();
39             } catch (SQLException ex) {
40                 System.err.println("Erro ao fechar conexão: " + ex);
41             }
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 * @author Henry Wallace
5 */
6
7 import com.gps.model.GrupoTrabalho;
8 import java.sql.Connection;
9 import java.sql.PreparedStatement;
10 import java.sql.ResultSet;
11 import java.sql.SQLException;
12 import java.util.ArrayList;
13 import java.util.List;
14
15 public class GrupoTrabalhoDAO {
16
17     // Create
18     public void create(GrupoTrabalho grupo) throws SQLException {
19         String sql = "INSERT INTO GrupoTrabalho (nomeGrupo) VALUES (?)";
20         Connection con = null;
21         PreparedStatement stmt = null;
22
23         try {
24             con = Conexao.getConnection();
25             stmt = con.prepareStatement(sql);
26             stmt.setString(1, grupo.getNomeGrupo());
27
28             stmt.executeUpdate();
29
30         } catch (SQLException ex) {
31             throw new SQLException("Erro ao cadastrar Grupo de Trabalho: " + ex.getMessage(), ex);
32         } finally {
33             Conexao.closeConnection(con, stmt);
34         }
35     }
36
37     // ReadALL
38     public List<GrupoTrabalho> readAll() throws SQLException {
39         String sql = "SELECT * FROM GrupoTrabalho ORDER BY nomeGrupo";
40         List<GrupoTrabalho> grupos = new ArrayList<>();
41         Connection con = null;
42         PreparedStatement stmt = null;
43         ResultSet rs = null;
44
45     }
```

```
16     try {
17         con = Conexao.getConnection();
18         stmt = con.prepareStatement(sql);
19         rs = stmt.executeQuery();
20
21         while (rs.next()) {
22             GrupoTrabalho grupo = new GrupoTrabalho();
23             grupo.setIdGrupo(rs.getInt("idGrupo"));
24             grupo.setNomeGrupo(rs.getString("nomeGrupo"));
25             grupos.add(grupo);
26         }
27     } catch (SQLException ex) {
28         throw new SQLException("Erro ao listar grupos: " + ex.getMessage(), ex);
29     } finally {
30         Conexao.closeConnection(con, stmt, rs);
31     }
32     return grupos;
33 }
34
35 // Delete
36 public void delete(int idGrupo) throws SQLException {
37     String sql = "DELETE FROM GrupoTrabalho WHERE idGrupo = ?";
38     Connection con = null;
39     PreparedStatement stmt = null;
40
41     try {
42         con = Conexao.getConnection();
43         stmt = con.prepareStatement(sql);
44         stmt.setInt(1, idGrupo);
45
46         stmt.executeUpdate();
47
48     } catch (SQLException ex) {
49         throw new SQLException("Erro ao deletar Grupo de Trabalho: " + ex.getMessage(), ex);
50     } finally {
51         Conexao.closeConnection(con, stmt);
52     }
53 }
54
55 }
```

OrdemServiçoDAO:

```

1  package com.gps.dao;
2
3  /**
4   * 
5   * @author Henry Wallace
6   */
7
8  import com.gps.model.OrdemServico;
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, entregue, 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.isEntregue());
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.setEntregue(rs.getBoolean("entregue"));
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                 // Trata as datas de Devolução
92                 Timestamp tsDevolucao = rs.getTimestamp("horaDevolucao");
93                 if (tsDevolucao != null) {
94                     os.setHoraDevolucao(new Date(tsDevolucao.getTime()));
95                 }
96             }
97         } finally {
98             Conexao.closeConnection(con, stmt);
99         }
100    }
101}

```

```
96          // Popula o objeto Grupo de Trabalho (Chave Estrangeira)
97          GrupoTrabalho grupo = new GrupoTrabalho();
98          grupo.setIdGrupo(rs.getInt("idGrupo_fk"));
99          grupo.setNomeGrupo(rs.getString("nomeGrupo"));
100         os.setGrupo(grupo);
101
102         ordens.add(os);
103     }
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
111 return ordens;
112 }
113
114 // ReadById (Busca uma única O.S. para Finalização)
115 public OrdemServico readById(int idOS) throws SQLException {
116     String sql = "SELECT OS.* , GT.nomeGrupo " +
117                 "FROM OrdemServico OS " +
118                 "INNER JOIN GrupoTrabalho GT ON OS.idGrupo_fk = GT.idGrupo " +
119                 "WHERE OS.idOS = ?";
120
121     Connection con = null;
122     PreparedStatement stmt = null;
123     ResultSet rs = null;
124     OrdemServico os = null;
125
126     try {
127         con = Conexao.getConnection();
128         stmt = con.prepareStatement(sql);
129         stmt.setInt(1, idOS);
130         rs = stmt.executeQuery();
131
132         if (rs.next()) {
133             os = new OrdemServico();
134             // Popule o objeto 'os' (mesmo padrão do readAll)
135             os.setIdOS(rs.getInt("idOS"));
136             os.setSolicitante(rs.getString("solicitante"));
137             os.setLocalAndar(rs.getInt("localAndar"));
138             os.setLocalPorta(rs.getString("localPorta"));
139             os.setDanoDescricao(rs.getString("danoDescricao"));
140             os.setRamal(rs.getString("ramal"));
141             os.setEntregue(rs.getBoolean("entregue"));
142         }
143     } catch (SQLException ex) {
144         throw new SQLException("Erro ao buscar Ordem de Serviço: " + ex.getMessage(), ex);
145     } finally {
146         Conexao.closeConnection(con, stmt, rs); // Fechamento seguro
147     }
148 }
```

```

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 }
194 }

```

GrupoTrabalho.java:

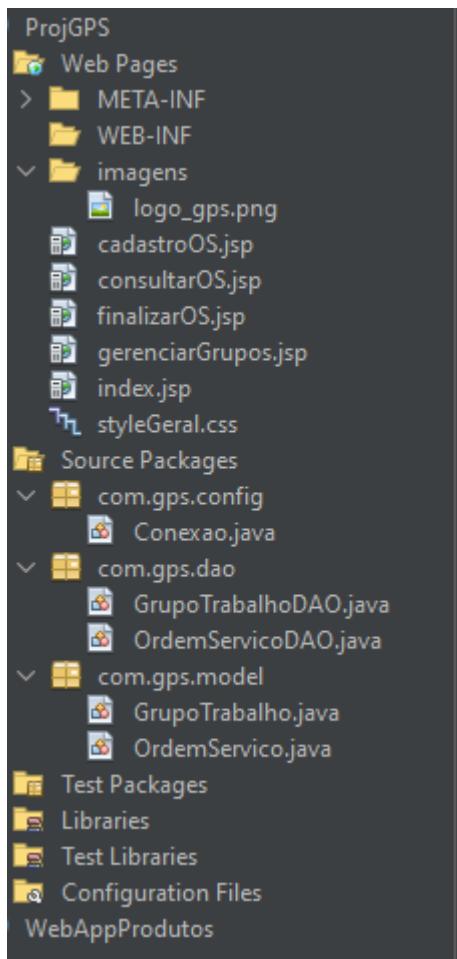
```
1 package com.gps.model;
2
3 /**
4 * 
5 * @author Henry Wallace
6 */
7
8 public class GrupoTrabalho {
9     private int idGrupo;
10    private String nomeGrupo;
11
12    // Construtor vazio
13    public GrupoTrabalho() {}
14
15    // Construtor completo
16    public GrupoTrabalho(int idGrupo, String 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;

Opcões 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 14:06:32	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>