



Instituto Superior de Gestão e Administração de Santarém

Informática de Gestão

Gestão de Frota – Veículos Exóticos

Gonçalo Dias | 22206831

Andreza Helen | 22208311

Pedro Varela | 22200848

Docente:

Professor Marco Tereso

Unidade Curricular: Tecnologias e Práticas Web

Santarém

Ano letivo 2022-2023

Índice

RESUMO.....	3
IDENTIFICADORES	4
TESTES SOFTWARE	5
DESENVOLVIMENTO DO TEMA.....	8
IMAGENS DO PROJETO	17
CONCLUSÃO	23

RESUMO

No âmbito da unidade curricular de Programação Web, foi proposto aos alunos o desenvolvimento de um projeto prático, de escolha livre entre os temas sugeridos ou uma ideia equivalente em termos de complexidade, nós decidimos seguir com o projeto 2.

O projeto constitui 40% da nota final e é essencial para a avaliação ao longo do semestre. Os grupos, compostos por até 3 alunos, foram incentivados a aplicar os conhecimentos adquiridos, utilizando tecnologias como PHP, HTML, Javascript, CSS e MySQL.

Adicionalmente, foi permitido o uso de frameworks como, o Laragon para facilitar o desenvolvimento. Ao final do semestre, os alunos são responsáveis pela apresentação e defesa do projeto perante a turma, destacando as funcionalidades implementadas e as decisões técnicas tomadas durante o processo de desenvolvimento.

IDENTIFICADORES

Versão 1.0: Base HTML e CSS

Versão 1.1: Adicionar ficheiro via URL (PHP)

Versão 1.2: Criação da BD

Versão 1.3: Inserção HTML restante

Versão 1.4: Recolha de dados

Versão 1.5: Conversão de imagens e modificação utilizadores

Versão 2.0: Eliminação de erros e criação logout, menu admin, menu utilizador

Versão 2.1: Animações e retoques no HTML

Versão 2.2 (Final): Eliminação de erros, retoques finais CSS e Testes

TESTES SOFTWARE

Os testes foram realizados para garantir que todas as funcionalidades principais do sistema estão a funcionar corretamente e que o sistema cumpre os requisitos especificados.

Tipos de Testes Realizados

- Testes de Unidade
- Testes de Integração
- Testes Funcionais
- Testes de Interface do Utilizador
- Testes de Regressão
- Testes de Desempenho

Testes de Unidade:

Função	Descrição	Entrada	Resultado Esperado
validarLogin	Verifica se o login é válido	username, password	Autenticação bem-sucedida
calcularPrecoAluguer	Calcula o preço do aluguer de um veículo	dias, preçoDiario	N/a – Não implementado
verificarDisponibilidade	Verifica se o veículo está disponível	veiculoID, dataInicio, dataFim	Disponibilidade confirmada

Testes de Integração:

Módulo	Descrição	Entrada	Resultado Esperado
Autenticação e Requisição	Verifica se um utilizador autenticado pode fazer uma requisição de veículo	username, veiculoID, dataInicio, dataFim	Requisição bem-sucedida
Requisição e Faturação	Verifica se a requisição gera uma fatura corretamente	requisicaoID	Fatura gerada corretamente
Devolução e Atualização de Estado	Verifica se a devolução do veículo atualiza o estado corretamente	requisicaoID	Estado atualizado para "devolvido"

Testes Funcionais:

Funcionalidade	Descrição	Entrada	Resultado Esperado
Login	Verifica se o utilizador pode iniciar sessão	username, password	Login bem-sucedido
Requisição de Veículo	Verifica se o utilizador pode requisitar um veículo	veiculoID, dataInicio, dataFim	Requisição registada
Consulta de Estado	Verifica se o utilizador pode consultar o estado das suas requisições	username	Estados das requisições exibidos

Testes de Interface do Utilizador:

Interface	Descrição	Ação	Resultado Esperado
Página de Login	Verifica se a página de login carrega corretamente	Aceder à página de login	Página carregada
Formulário de Requisição	Verifica se o formulário de requisição aceita entradas válidas	Preencher formulário	Formulário submetido
Tabela de Estado	Verifica se a tabela de estado exibe os dados corretamente	Consultar estado	Dados exibidos corretamente

Testes de Regressão:

Funcionalidade	Descrição	Ação	Resultado Esperado
Login	Verifica se o login continua a funcionar após atualização	username, password	Login bem-sucedido
Requisição de Veículo	Verifica se a requisição continua a funcionar após atualização	veiculoID, dataInicio, dataFim	Requisição registada
Consulta de Estado	Verifica se a consulta de estado continua a funcionar após atualização	username	Estados das requisições exibidos

Testes de Desempenho:

Funcionalidade	Descrição	Cenário	Resultado Esperado
Login	Verifica o tempo de resposta do login sob carga	1000 utilizadores simultâneos	Tempo de resposta aceitável
Requisição de Veículo	Verifica o desempenho da requisição sob carga	500 requisições simultâneas	Sistema responde corretamente
Consulta de Estado	Verifica o desempenho da consulta de estado sob carga	1000 consultas simultâneas	Dados exibidos corretamente

DESENVOLVIMENTO DO TEMA

No decorrer deste trabalho várias ideias foram-nos fornecidas por fontes externas, mas nós acabamos por seguir com a ideia relacionada a “Carros Exóticos”.

O código fornecido é uma mistura de HTML, CSS e PHP, que implementa um sistema de login e manipulação de dados guardados em base de dados. Contém várias funções que realizam diferentes tarefas, como verificação de login, registo de novos dados de veículos e trajetos, pesquisa por nome ou matrícula, consulta de dados, criação de backup de BD e atualização de informações.

Alguns elementos do projeto foram diretamente retirados do Moodle, Git Hub, BlackBox, etc.

Aqui está uma explicação mais detalhada:

Estrutura Geral do Código

O código é estruturado em diferentes secções que desempenham funções específicas:

- Conexão com a Base de Dados
- Estrutura HTML e CSS
- Gestão de Itinerários
- Adição de Itinerário
- Atualização de Itinerário
- Remoção de Itinerário
- Gestão de Veículos
- Adição de Veículo
- Atualização de Veículo
- Remoção de Veículo
- Gestão de Utilizadores
- Adição de Utilizador
- Atualização de Utilizador
- Remoção de Utilizador

1. Conexão com a Base de Dados:

A conexão com a base de dados é estabelecida através do ficheiro `conexaoDb.php`. Este ficheiro é incluído em várias partes do código para garantir a conexão com a base de dados. Abaixo está o trecho que inclui o ficheiro de conexão:

```
php
Copy code
<?php
include 'conexaoDb.php';
?>
```


2. Estrutura HTML e CSS:

A interface do utilizador é criada utilizando HTML e CSS. O exemplo abaixo mostra a estrutura de uma página HTML para adicionar um novo itinerário:

```
html
Copy code
<!DOCTYPE html>
<html lang="pt-PT">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Adicionar Itinerário</title>
  <link rel="stylesheet" href="styles.css">
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      margin: 0;
      padding: 0;
    }
    .form-container {
      max-width: 600px;
      margin: 50px auto;
      padding: 20px;
      background-color: white;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }
    .form-container form label {
      display: block;
      margin-bottom: 8px;
      font-weight: bold;
    }
    .form-container form input, .form-container form textarea {
      width: 100%;
      padding: 10px;
      margin-bottom: 20px;
      border: 1px solid #ccc;
      border-radius: 5px;
    }
    .form-container form button {
      padding: 10px 15px;
      background-color: #28a745;
      border: none;
      border-radius: 5px;
    }
```

```

        color: white;
        font-size: 16px;
    }
</style>
</head>
<body>
    <div class="form-container">
        <h2>Adicionar Itinerário</h2>
        <form action="adicionarItinerario.php" method="post">
            <label for="origem">Origem</label>
            <input type="text" id="origem" name="origem" required>
            <label for="destino">Destino</label>
            <input type="text" id="destino" name="destino" required>
            <label for="data_partida">Data de Partida</label>
            <input type="date" id="data_partida" name="data_partida" required>
            <label for="data_chegada">Data de Chegada</label>
            <input type="date" id="data_chegada" name="data_chegada" required>
            <label for="descricao">Descrição</label>
            <textarea id="descricao" name="descricao" required></textarea>
            <button type="submit">Adicionar Itinerário</button>
        </form>
    </div>
</body>
</html>

```

3. Gestão de Itinerários:

Adição de Itinerário -> O ficheiro adicionarItinerario.php contém o código necessário para adicionar um novo itinerário à base de dados:

```

php
Copy code
<?php
include 'conexaoDb.php';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $origem = $_POST['origem'];
    $destino = $_POST['destino'];
    $data_partida = $_POST['data_partida'];
    $data_chegada = $_POST['data_chegada'];
    $descricao = $_POST['descricao'];

    $sql = "INSERT INTO itinerarios (origem, destino, data_partida, data_chegada,
descricao) VALUES ('$origem', '$destino', '$data_partida', '$data_chegada',
'$descricao')";

    if ($conn->query($sql) === TRUE) {

```

```

        echo "Novo itinerário adicionado com sucesso";
    } else {
        echo "Erro: " . $sql . "<br>" . $conn->error;
    }
}

$conn->close();
?>

```

Atualização de Itinerário:

O ficheiro atualizarItinerario.php contém o código necessário para atualizar um itinerário existente na base de dados:

```

php
Copy code
<?php
include 'conexaoDb.php';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $id = $_POST['id'];
    $origem = $_POST['origem'];
    $destino = $_POST['destino'];
    $data_partida = $_POST['data_partida'];
    $data_chegada = $_POST['data_chegada'];
    $descricao = $_POST['descricao'];

    $sql = "UPDATE itinerarios SET origem='$origem', destino='$destino',
data_partida='$data_partida', data_chegada='$data_chegada', descricao='$descricao'
WHERE id=$id";

    if ($conn->query($sql) === TRUE) {
        echo "Itinerário atualizado com sucesso";
    } else {
        echo "Erro: " . $sql . "<br>" . $conn->error;
    }
}

$conn->close();
?>

```

Remoção de Itinerário:

O ficheiro removerItinerario.php contém o código necessário para remover um itinerário da base de dados:

```
php
Copy code
<?php
include 'conexaoDb.php';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $id = $_POST['id'];

    $sql = "DELETE FROM itinerarios WHERE id=$id";

    if ($conn->query($sql) === TRUE) {
        echo "Itinerário removido com sucesso";
    } else {
        echo "Erro: " . $sql . "<br>" . $conn->error;
    }

    $conn->close();
}
?>
```

4. Gestão de Veículos:

Adição de Veículo

O ficheiro adicionarVeiculo.php contém o código necessário para adicionar um novo veículo à base de dados:

```
php
Copy code
<?php
include 'conexaoDb.php';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $marca = $_POST['marca'];
    $modelo = $_POST['modelo'];
    $ano = $_POST['ano'];
    $matricula = $_POST['matricula'];

    $sql = "INSERT INTO veiculos (marca, modelo, ano, matricula) VALUES ('$marca', '$modelo', '$ano', '$matricula')";

    if ($conn->query($sql) === TRUE) {
```

```

        echo "Novo veículo adicionado com sucesso";
    } else {
        echo "Erro: " . $sql . "<br>" . $conn->error;
    }
}

$conn->close();
?>

```

Atualização de Veículo:

O ficheiro atualizarVeiculo.php contém o código necessário para atualizar um veículo existente na base de dados:

```

php
Copy code
<?php
include 'conexaoDb.php';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $id = $_POST['id'];
    $marca = $_POST['marca'];
    $modelo = $_POST['modelo'];
    $ano = $_POST['ano'];
    $matricula = $_POST['matricula'];

    $sql = "UPDATE veiculos SET marca='$marca', modelo='$modelo', ano='$ano',
matricula='$matricula' WHERE id=$id";

    if ($conn->query($sql) === TRUE) {
        echo "Veículo atualizado com sucesso";
    } else {
        echo "Erro: " . $sql . "<br>" . $conn->error;
    }
}

$conn->close();
?>

```

Remoção de Veículo:

O ficheiro `removerVeiculo.php` contém o código necessário para remover um veículo da base de dados:

```
php
Copy code
<?php
include 'conexaoDb.php';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $id = $_POST['id'];

    $sql = "DELETE FROM veiculos WHERE id=$id";

    if ($conn->query($sql) === TRUE) {
        echo "Veículo removido com sucesso";
    } else {
        echo "Erro: " . $sql . "<br>" . $conn->error;
    }

    $conn->close();
}
?>
```

5. Gestão de Utilizadores:

Adição de Utilizador -> O ficheiro `adicionarUtilizador.php` contém o código necessário para adicionar um novo utilizador à base de dados:

```
php
Copy code
<?php
include 'conexaoDb.php';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nome = $_POST['nome'];
    $email = $_POST['email'];
    $senha = password_hash($_POST['senha'], PASSWORD_BCRYPT);

    $sql = "INSERT INTO utilizadores (nome, email, senha) VALUES ('$nome', '$email', '$senha')";

    if ($conn->query($sql) === TRUE) {
        echo "Novo utilizador adicionado com sucesso";
    } else {
        echo "Erro: " . $sql . "<br>" . $conn->error;
    }
}
```

```
}
```

```
$conn->close();
```

```
}
```

```
?>
```

Atualização de Utilizador:

O ficheiro atualizarUtilizador.php contém o código necessário para atualizar um utilizador existente na base de dados:

```
php
```

```
Copy code
```

```
<?php
```

```
include 'conexaoDb.php';
```

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
```

```
    $id = $_POST['id'];
```

```
    $nome = $_POST['nome'];
```

```
    $email = $_POST['email'];
```

```
    $senha = password_hash($_POST['senha'], PASSWORD_BCRYPT);
```

```
    $sql = "UPDATE utilizadores SET nome='$nome', email='$email', senha='$senha'
    WHERE id=$id";
```

```
    if ($conn->query($sql) === TRUE) {
```

```
        echo "Utilizador atualizado com sucesso";
```

```
    } else {
```

```
        echo "Erro: " . $sql . "<br>" . $conn->error;
```

```
    }
```

```
    $conn->close();
```

```
}
```

```
?>
```

Remoção de Utilizador

O ficheiro removerUtilizador.php contém o código necessário para remover um utilizador da base de dados:

```
php
```

```
Copy code
```

```
<?php
```

```
include 'conexaoDb.php';
```

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
```

```
    $id = $_POST['id'];
```

```
    $sql = "DELETE FROM utilizadores WHERE id=$id";
```

```
if ($conn->query($sql) === TRUE) {  
    echo "Utilizador removido com sucesso";  
} else {  
    echo "Erro: " . $sql . "<br>" . $conn->error;  
}  
  
$conn->close();  
}  
?>
```


IMAGENS DO PROJETO



Figura 1 – Index

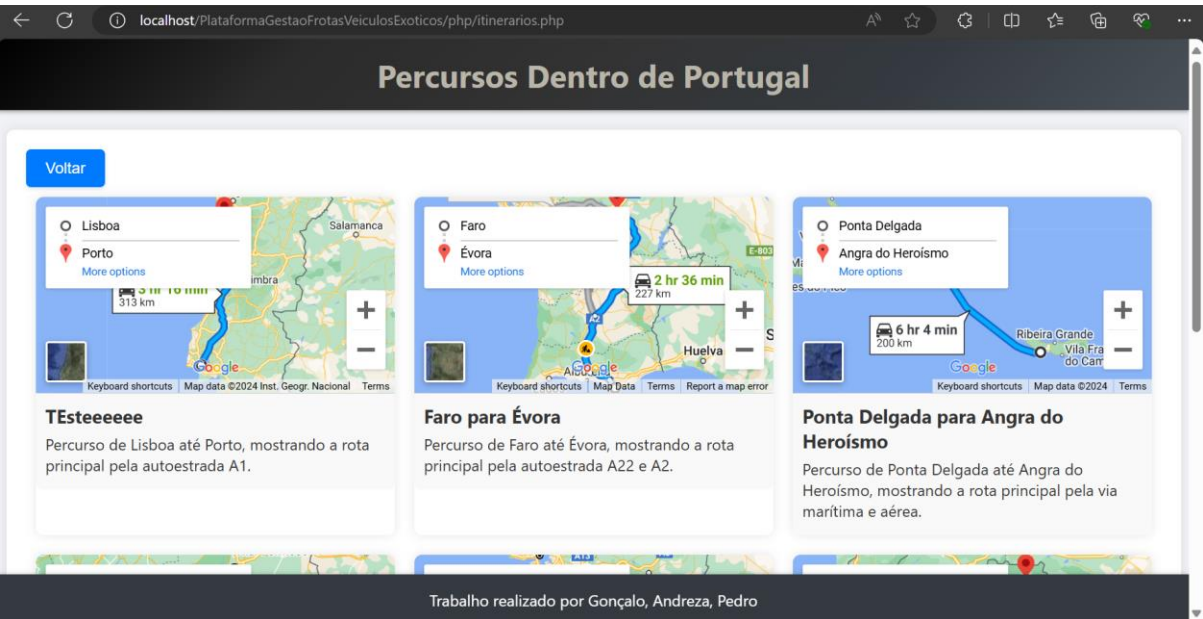


Figura 2 - Percursos Disponíveis

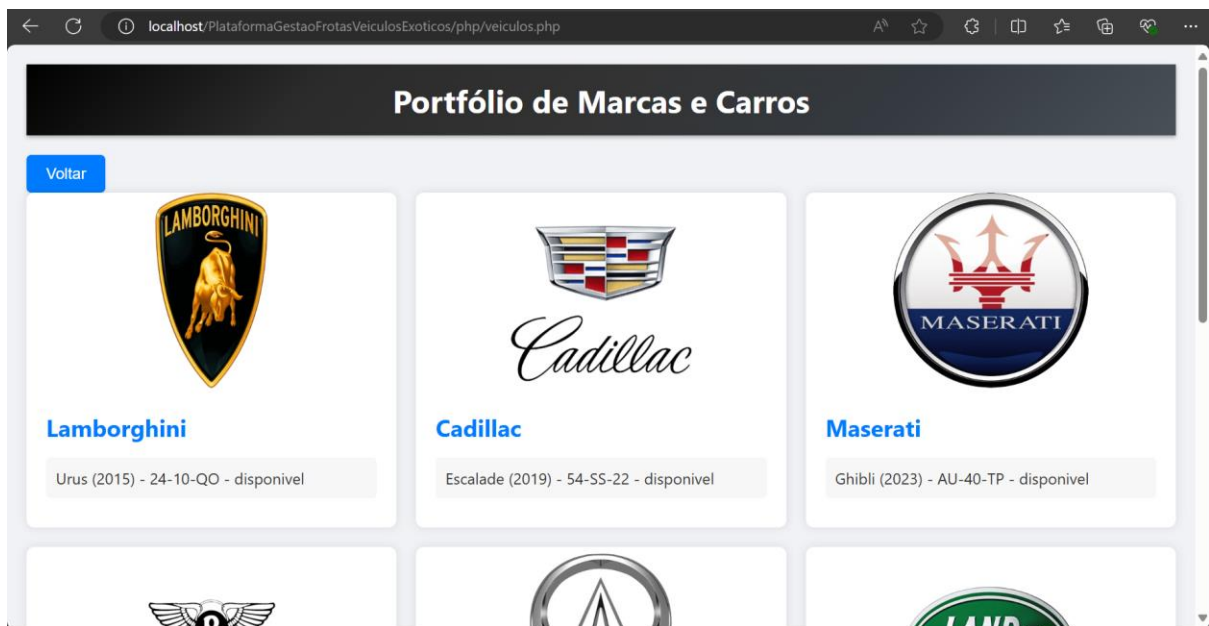


Figura 3 - Portfólio de Marcas de e carros

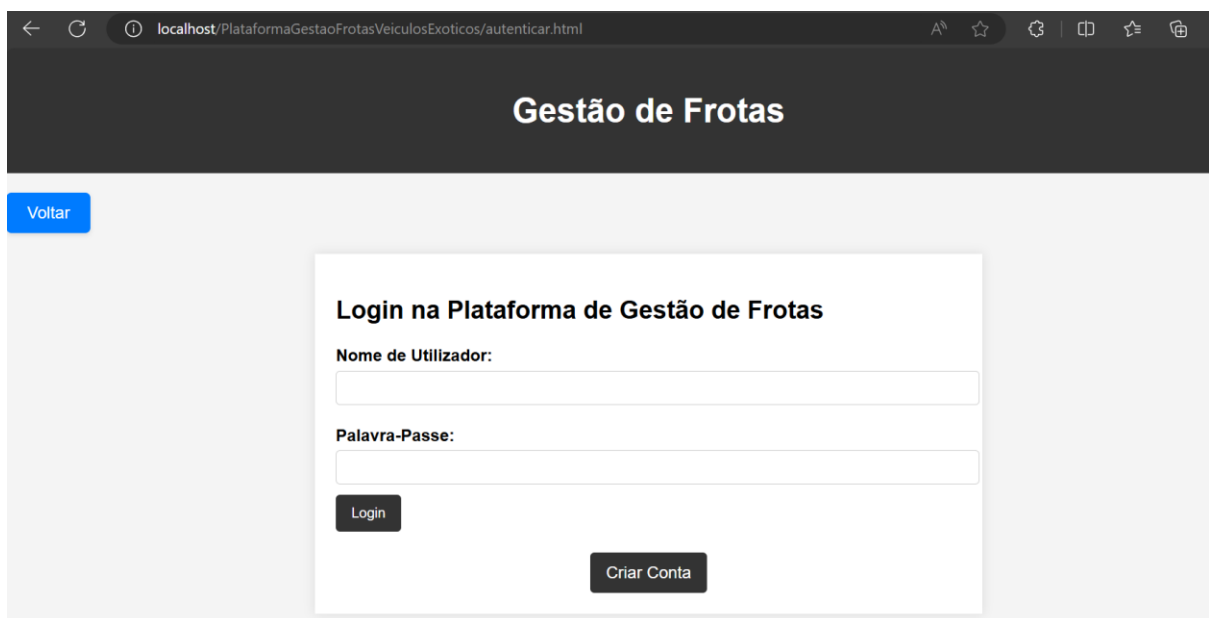


Figura 4 - Login | Criação de Conta



Figura 5 - Menu Admin

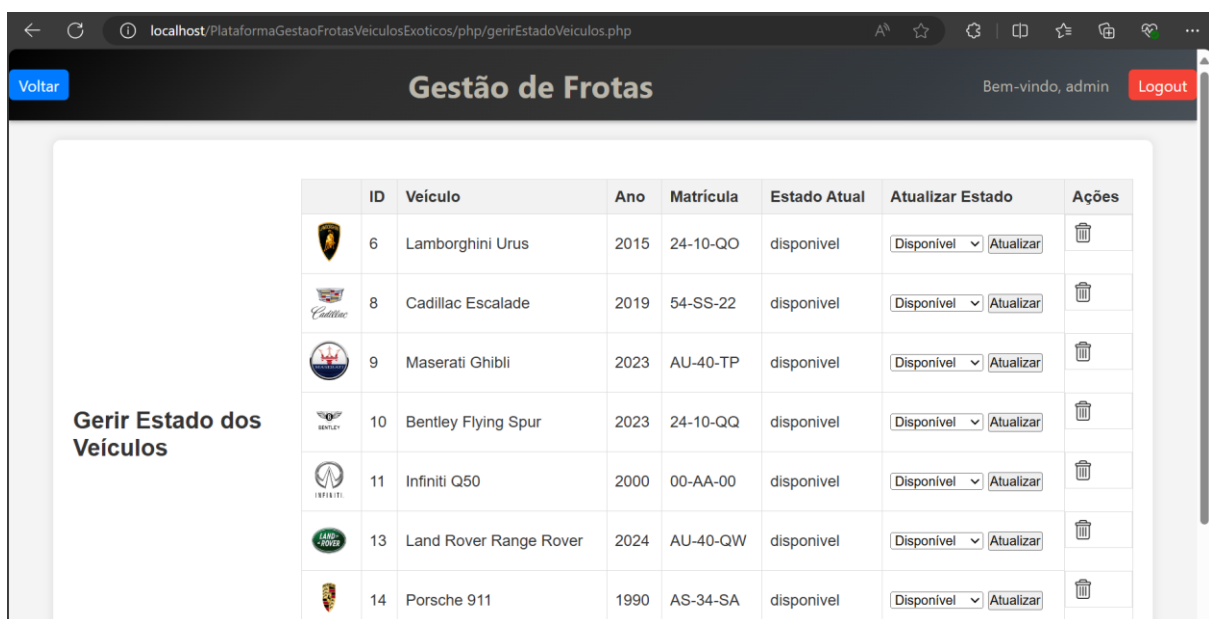


Figura 6 - Admin | Gerir Estado Veículos



Figura 7 - Admin | Gerir Requisições

The screenshot shows a web browser at localhost/PlataformaGestaoFrotasVeiculosExoticos/php/adicionarVeiculo.php. The page has a dark header with a 'Voltar' button, the title 'Gestão de Frotas', and a user greeting 'Bem-vindo, admin' with a 'Logout' button. The main content area features a white form titled 'Adicionar Novo Veículo'. The form includes four fields: 'Marca' (a dropdown menu with 'Selecione uma marca'), 'Modelo' (a dropdown menu with 'Selecione um modelo'), 'Ano' (a text input), and 'Matrícula' (a text input). A green 'Adicionar Veículo' button is at the bottom of the form.

Figura 8 - Admin | Inserir novo veiculo

The screenshot shows a web browser at localhost/PlataformaGestaoFrotasVeiculosExoticos/php/adicionarItinerario.php. The page has a dark header with a 'Voltar' button, the title 'Gestão de Frotas', and a user greeting 'Bem-vindo, admin' with a 'Logout' button. The main content area features a white form titled 'Adicionar Novo Itinerário'. The form includes three fields: 'Título' (a text input), 'Descrição' (a text area), and 'URL do Mapa' (a text input). A green 'Adicionar Itinerário' button is at the bottom of the form.

Figura 9 - Admin | Inserir novo Itinerário

The screenshot shows a web browser at localhost/PlataformaGestaoFrotasVeiculosExoticos/php/verHistoricoRequisicoes.php. The page has a dark header with a 'Voltar' button, the title 'Gestão de Frotas', and a user greeting 'Bem-vindo, admin' with a 'Logout' button. The main content area features a white box titled 'Histórico de Requisições'. Inside the box, there are filters for 'Utilizador:', 'Veículo:', 'Estado:' (a dropdown menu with 'Todos'), 'Data Início:' (a date input with 'dd/mm/yyyy' format), and 'Data Fim:' (a date input with 'dd/mm/yyyy' format). A 'Filtrar' button is to the right of the filters. Below the filters is a table with 6 columns: ID, Utilizador, Veículo, Data de Requisição, Data de Devolução, and Estado. The table contains 8 rows of data.

ID	Utilizador	Veículo	Data de Requisição	Data de Devolução	Estado
1	pedro	Cadillac Escalade	2024-06-28		aprovado
2	pedro	Lamborghini Urus	2024-06-18		rejeitado
3	pedro	Lamborghini Urus	2000-12-20		aprovado
5	pedro	Maserati Ghibli	2024-06-27		aprovado
7	pedro	Infiniti Q50	2024-06-29		aprovado
8	pedro	Lamborghini Urus	2024-06-26		rejeitado

Figura 10 - Admin | Histórico Requisições

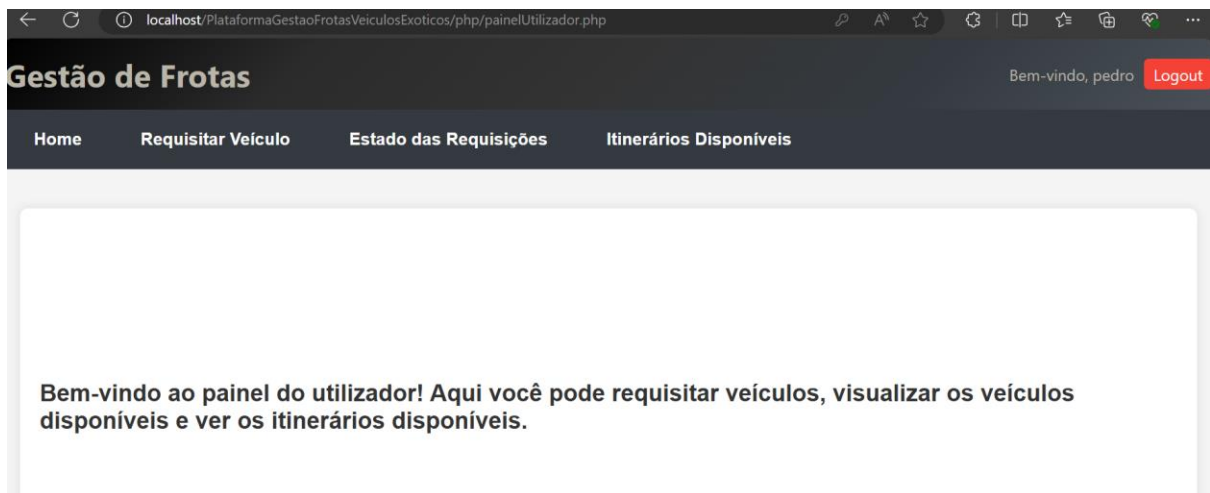


Figura 11 - Login Utilizador

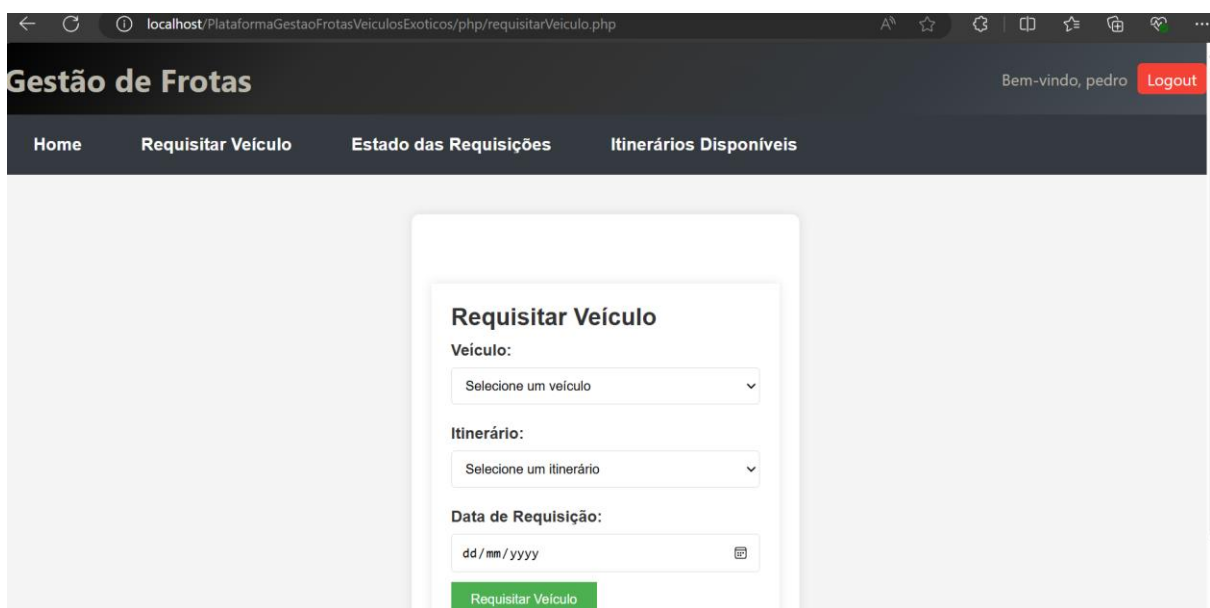


Figura 12 - Utilizador | Requisitar veículo

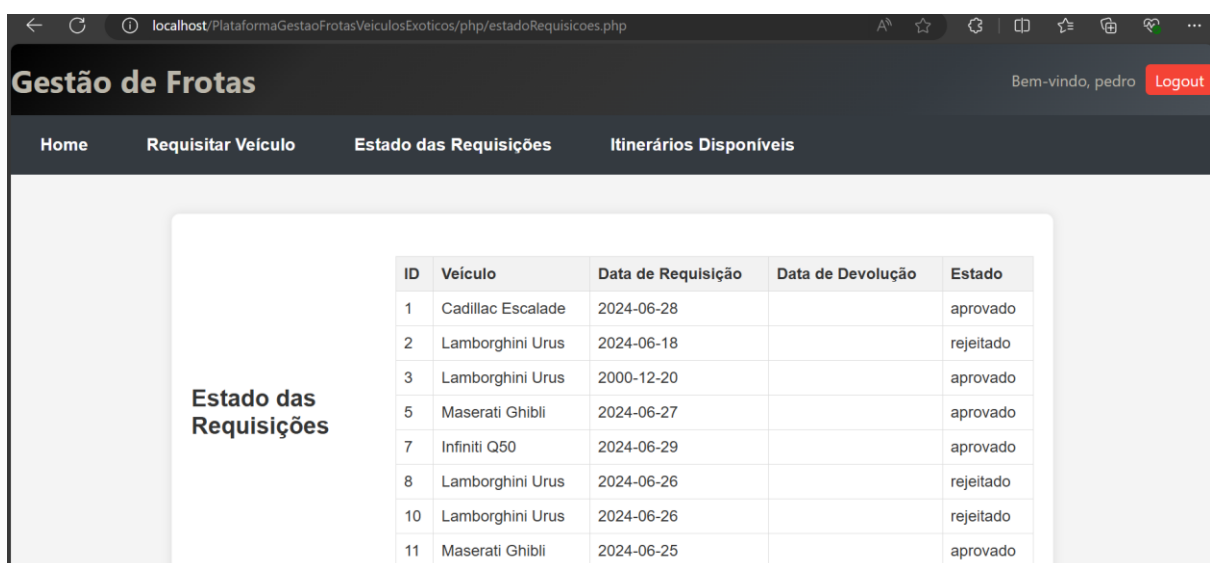


Figura 13 - Utilizador | Estado das Requisições

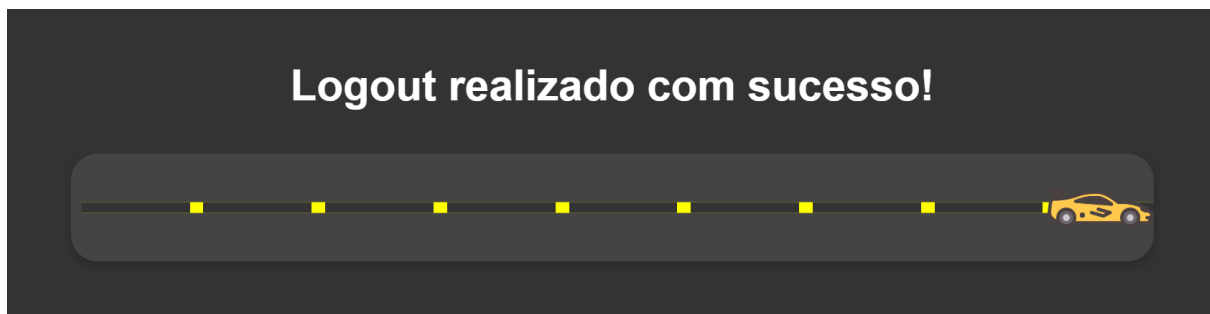


Figura 14 - Loading Screen

CONCLUSÃO

O sistema de gestão de itinerários, veículos e utilizadores desenvolvido em PHP fornece uma maneira eficiente e organizada de gerir os dados relacionados a viagens, frota de veículos e utilizadores. As funcionalidades de adição, atualização e remoção de dados são implementadas de forma a garantir a integridade e segurança dos dados através da utilização de conexões seguras e da sanitização das entradas dos utilizadores. Este relatório detalhou cada parte do código, proporcionando uma visão abrangente do funcionamento do sistema.

Em síntese, o projeto desenvolvido na disciplina de Programação Web representou uma aplicação prática dos conhecimentos teóricos adquiridos ao longo do semestre. Criamos uma plataforma completa para gestão de frotas empresariais, abrangendo desde a inserção de veículos até à gestão de itinerários. A colaboração eficaz entre os membros do grupo permitiu superar desafios técnicos, como a integração de uma base de dados MySQL e o desenvolvimento de uma interface dinâmica utilizando HTML, CSS, PHP e Javascript.

A apresentação final do projeto não só demonstrou o produto final desenvolvido, mas também refletiu as escolhas técnicas e arquiteturais realizadas ao longo do processo. Esta experiência não apenas contribuiu para a avaliação académica, mas também fortaleceu competências em trabalho em equipe, gestão de projetos e resolução de problemas práticos. Este projeto foi uma preparação valiosa para futuras oportunidades e desafios na área de desenvolvimento web.