epb
escola profissional
de braga

PAP - PROVA DE
APTIDÃO PROFISSIONAL


# CarMax

## CURSO PROFISSIONAL DE TECNICO DE GESTÃO E PROGRAMAÇÃO DE SISTEMAS INFORMÁTICOS

Pedro Brites Alves

Trabalho realizado sob a orientação de:

Prof. Manuel Ferreira

Escola Profissional de Braga

Braga, 10 de maio de 2024

PESSOAS 2030
PORTUGAL 2030
Cofinanciado pela União Europeia
SELO DE CONFORMIDADE EQAVET
uma escola Rumos education | Knowledge sharing
Os Fundos Europeus mais próximos de si.

PESSOAS 2030
PORTUGAL 2030
Cofinanciado pela União Europeia
SELO DE CONFORMIDADE EQAVET
uma escola Rumos education | Knowledge sharing
Os Fundos Europeus mais próximos de si.

## Nota Introdutória

A Prova de Aptidão Profissional consiste na apresentação e defesa, perante um júri, de um projeto, consubstanciado num produto, material ou intelectual, numa intervenção ou numa atuação, bem como do respetivo relatório final de realização e apreciação crítica, demonstrativo de saberes e competências profissionais, adquiridos ao longo da formação e estruturante do futuro profissional.

A sua realização constitui-se como um dos imperativos legais para a conclusão do curso profissional de (designação do curso), que confere uma dupla certificação: qualificação profissional de nível IV e o 12º ano como certificação escolar de nível secundário.

A presente prova foi realizada na Escola Profissional de Braga no presente ano letivo e a sua defesa realizada, perante um júri final, nas datas estabelecidas no calendário escolar.



Os Fundos Europeus mais próximos de si.

## Dedicatória

Quero dedicar esta PAP (Prova de Aptidão Profissional) a todos os meus familiares e colegas que me ajudaram e me apoiaram principalmente nestes últimos 3 anos



Os Fundos Europeus mais próximos de si.

## Agradecimentos

Queria agradecer a todos os professores por me ajudarem nas aulas principalmente o professor Manuel Ferreira o meu professor acompanhante também queria agradecer aos meus colegas



PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

# Índice





PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing

Os Fundos Europeus mais próximos de si.

# Índice de Figuras

# Introdução

Escolhi para o tema da minha pap uma oficina mecânica porque tenho uma paixão pelos carros de corrida desde pequeno e ao longo do tempo comecei a gostar mais ainda, quando eu era pequeno amava ver os carros de corrida na televisão depois que descobri o youtube comecei a pesquisar filmes e desenhos animados de carros, quando cresci descobri mais aplicações e browsers que podia pesquisar mais sobre os carros e como já sabia um bocado sobre os carros então comecei o meu projeto final.

Fundamentação do projeto

- No desenvolvimento da minha prova de aptidão profissional vou realizar um website para uma oficina mecânica.
- Será possível iniciar sessão numa página de login após o registo na aplicação.
- Será possível agendar reparações automóveis e visualizar o estado das mesmas, acompanhando assim a evolução da reparação do automóvel.
- Será possível ter acesso à ficha de reparação do veículo ficando a conhecer os custos.
- Será possível solicitar orçamentos para reparação dos veículos e ter uma previsão do tempo da reparação.
- O gestor terá acesso aos pedidos de agendamento assim como aos pedidos de orçamento.
- Como Linguagem de Programação irei usar o Html, CSS e JavaScript.
- Para a base de dados vou usar o SQL Server.

Objetivos

1. Reforçar a capacidade de superar situações inesperadas ou imprevistas nas diferentes fases do projeto.



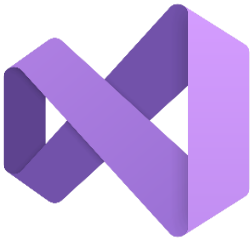
PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

2. Desenvolver o espírito crítico e sentido de autonomia;
3. Desenvolver competências de decisão, escolha e procura de informação;
4. Aplicar conhecimentos e competências adquiridos nas diferentes disciplinas;
5. Melhorar a disciplina mental, capacidade de organização e metodologia de trabalho;
6. Desenvolver trabalho em equipa;
7. Aumentar a minha capacidade de domínio sobre este tema:
8. Desenvolver competências de criação numa aplicação de CarMax.

| Cronograma | | |
|---|---|---|
| Fase | Calendarização | Elenco de atividades nas diversas fases |
| Conceção | Setembro 2023 | Conhecimento do regulamento |

| | | |
|---|---|---|
| | Setembro 2023 | Definição dos critérios da PAP pela direção |
| | Setembro 2023 | Período de negociação / reflexão / decisão sobre os temas / problemas a explorar |
| | Setembro 2023 | Normas para a elaboração de um trabalho científico |
| | Até 31 outubro 2023 | Entrega do projeto de acordo com as normas do regulamento |
| | | |
| Desenvolvimento | Outubro 2023 | Brainstorming para definição das funcionalidades do site |
| | Novembro 2023 | Desenho de mockups / Criação da base de dados |
| | Dezembro/Fevereiro | Inicio da codificação do site |
| | Março 2024 | Testes e melhoria da interface do site |
| | | |
| | | |
| Autoavaliação e Relatório | Janeiro 2024 | Defesa intermédia da PAP |
| | 6 maio 2024 | Conclusão do projeto e entrega do relatório |
| | Julho 2024 | Defesa final da PAP |
| | | |
| | | |
| | | |

Materiais utlizados



Microsoft Visual Studio é um ambiente de desenvolvimento integrado (IDE) da Microsoft para desenvolvimento de software especialmente dedicado ao .NET Framework e às linguagens Visual Basic (VB), C, C++, C# (C Sharp) e F# (F Sharp). Também é um produto de desenvolvimento na área web, usando a plataforma do ASP.NET, como websites, aplicativos web, serviços web e aplicativos móveis.

Linguagem utlizadas


HTML (HyperText Markup Language,) (abreviação para a expressão inglesa HyperText Markup Language, que significa: "Linguagem de Marcação de Hipertexto") é uma linguagem de marcação utilizada na construção de páginas na Web. Documentos HTML podem ser interpretados por navegadores. A tecnologia é fruto da junção entre os padrões HyTime e SGML.

HyTime é um padrão para a representação estruturada de hipermídia e conteúdo baseado em tempo. Um documento é visto como um conjunto de eventos concorrentes dependentes de tempo (como áudio, vídeo, etc.), conectados por hiperligações (em inglês: hyperlink e link). O padrão é independente de outros padrões de processamento de texto em geral.

SGML é um padrão de formatação de textos. Não foi desenvolvido para hipertexto, mas tornou-se conveniente para transformar documentos em hiper-objetos e para descrever as ligações.



PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

Cascading Style Sheets (abreviado CSS) é um mecanismo para adicionar estilos (cores, fontes, espaçamento, etc.) a uma página web,aplicado diretamente nas tags HTML ou ficar contido dentro das tags <style>. Também é possível, adicionar estilos adicionando um link para um arquivo CSS que contém os estilos. Assim, quando se quiser alterar a aparência dos documentos vinculados a este arquivo CSS, basta modificá-lo.

JavaScript (frequentemente abreviado como JS) é uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinâmica fraca e multiparadigma (protótipos, orientado a objeto, imperativo e funcional). Juntamente com HTML e CSS, o JavaScript é uma das três principais tecnologias da World Wide Web. JavaScript permite páginas da Web interativas e, portanto, é uma parte essencial dos aplicativos da web. A grande maioria dos sites usa, e todos os principais navegadores têm um mecanismo JavaScript dedicado para executá-lo. É atualmente a principal linguagem para programação client-side em navegadores web. É também bastante utilizada do lado do servidor através de ambientes como o node.js.



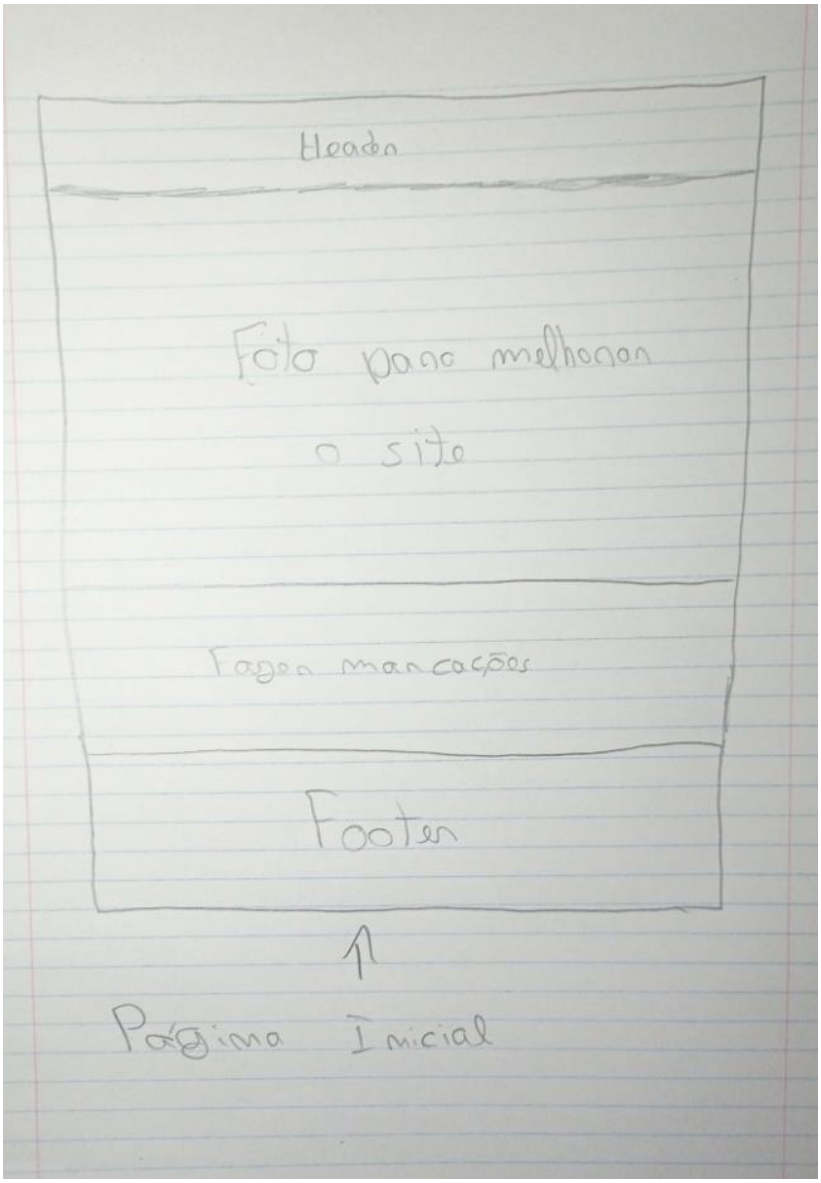
PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

# 1. Desenvolvimento

1- Comecei a pensar na estrutura do meu web site, a organização da estrutura,e as páginas que vai ter.
2-Comecei a fazer o que tinha pensado que era a página inicial,a página de login e registo, header, footer, a ficha cliente, a ficha de reparação e a base de dados.
3- A página de login e registo serve para logar ou registar a sua conta.
4-O header e o footer serve para a organização do site.
5-A página inicial vai ter muita coisa.
6-A Base de dados vou ter de ligar à página de login e registo.
7-A página inicial o header e o footer vão ter ligações a outras páginas para ajudar na organização do meu web site.

# 2. O meu projeto final

# Desenho da página inicial



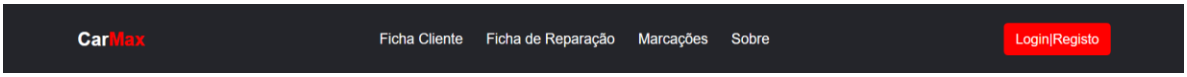
```
<header>
    <nav class="nav-bar">
        <div class="logo">
            <h1>Car</h1>
            <h1 style="color:red;"><p>Max</p></h1>
        </div>
        <div class="nav-list">
            <ul>
                <li class="nav-item"> <a href="fichcliente.aspx" class="nav-link">Ficha Cliente</a> </li>
                <li class="nav-item"><a href="fichreparacao.aspx" class="nav-link">Ficha de Reparação</a></li>
                <li class="nav-item"><a href="agenda.html" class="nav-link">Marcações</a></li>
                <li class="nav-item"> <a href="Sobre.aspx" class="nav-link">Sobre</a></li>
            </ul>
        </div>
        <div class="login-button">
            <button><a href="Login.aspx">Login|Registo</a></button>
        </div>

        <div class="mobile-menu-icon">
            <button onclick="menuShow()"><img class="icon" src="imagens/menu_white_36dp.svg" alt=""></button>
        </div>
    </nav>
    <div class="mobile-menu">
        <ul>
            <li class="nav-item"><a href="fichcliente.aspx " class="nav-link">Ficha Cliente</a></li>
            <li class="nav-item"><a href="fichreparacao.aspx" class="nav-link">Ficha de Reparação</a></li>
            <li class="nav-item"><a href="agenda.html" class="nav-link">Marcações</a></li>
            <li class="nav-item"><a href="Sobre.aspx" class="nav-link">Sobre</a></li>
        </ul>

        <div class="login-button">

            <button><a href="registo1.aspx">Entrar</a></button>
        </div>
    </div>
</header>
```

Este código é o header, o header é um menu responsivo que usei para organizar o meu site quando minimizamos o site tem um botão que vai abrir o menu hambúrguer para o site ficar organizado nos dispositivos móveis




Como o nome indica, um header refere-se à seção superior da página de um site. É uma faixa de conteúdo geralmente fixa no topo e que, portanto, aparece de forma consistente em todas as páginas



PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

Este é o menu responsivo quando minimizo a página para dispositivos moveis como por exemplo telemóveis, tablets e muito mais



Este é o menu depois de clicar nos 3 tracinhos que vai ter as mesmas coisas que o menu normal só que para os dispositivos móveis

```
function menuShow() {
    let menuMobile = document.querySelector('.mobile-menu');
    if (menuMobile.classList.contains('open')) {
        menuMobile.classList.remove('open');
        document.querySelector('.icon').src = "imagens/menu_white_36dp.svg";
    } else {
        menuMobile.classList.add('open');
        document.querySelector('.icon').src = "imagens/close_white_36dp.svg";
    }
}
```

**Seleção do elemento do menu móvel**

```
let menuMobile = document.querySelector('.mobile-menu');
```

Utiliza se **document.querySelector** para selecionar um elemento HTML com a classe CSS **.mobile-menu** e armazena-o na variável **menu Mobile**.

**Verificação da classe 'open'**

```
if (menuMobile.classList.contains('open')) {
```

Verifica se o elemento do menu móvel tem a classe CSS **open**.

**Se o menu já estiver aberto**

```
menuMobile.classList.remove('open');
document.querySelector('.icon').src = "imagens/menu_white_36dp.svg";
```

Se o menu já estiver aberto (ou seja, tem a classe open), esta classe é removida do elemento do menu móvel. Em seguida, altera a fonte da imagem de um ícone (possivelmente um ícone de menu) para a imagem de um ícone de fechar (provavelmente para indicar que o menu pode ser fechado).

**Se o menu não estiver aberto**:

```
    } else {
        menuMobile.classList.add('open');
        document.querySelector('.icon').src = "imagens/close_white_36dp.svg";
    }
```

Se o menu não estiver aberto, adiciona a classe **open** ao elemento do menu móvel, o que possivelmente fará com que seja exibido. Em seguida, altera a fonte da imagem do ícone para uma imagem de fechar (indicando que o menu pode ser fechado).


Esta página é a inicial

```
<footer>
    <div class="container">
        <div class="footer-content">
            <h3>Contactos</h3>
            <p>Email: 0521182@alunos.epb.pt</p>
            <p>Telemóvel: +351 253 954 195</p>
            <p>Morada: R. Augusto Veloso 140, Braga</p>
        </div>
        <div class="footer-content">
            <h3>Menus</h3>
            <ul class="list">
                <li><a href="">Início</a></li>
                <li><a href="fichcliente.aspx">Ficha Cliente</a></li>
                <li><a href="fichreparacao.aspx">Ficha de Reparação</a></li>
                <li class="nav-item"><a href="agenda.html" class="nav-link">Marcações</a></li>
                <li><a href="Sobre.aspx">Sobre</a></li>
            </ul>
        </div>
        <div class="footer-content">
            <h3>Redes Sociais</h3>
            <ul class="social-icons">
                <li><a href="https://instagram.com/carmax_epb"><img class="img_po" src="imagens/face.png" width="37" height="37"></a></li>
                <li><a href="https://instagram.com/carmax_epb"><img class="img_po" src="imagens/insta.png" width="37" height="37"></a></li>
                <li><a href="https://www.youtube.com/@Carmaxepb"><img class="img_po" src="imagens/yt.png" width="50" height="50"></a></li>
            </ul>
        </div>
    </div>
    <div class="bottom-bar">
        <p>CarMax</p>
    </div>
</footer>
```


O footer é a área final das páginas de um site. Normalmente, é no footer que ficam informações como endereço, contatos, links para páginas importantes, políticas de privacidade, termos e condições de troca e devolução, selos de segurança, redes sociais e mais.



PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
    <style>
        * {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: Arial, Helvetica, sans-serif;
}

header {
    background-color: #24252a;
    box-shadow: 0px 3px 10px #464646;
}

footer {
    background: #24252A;
    padding-top: 20px;
    color: white;
}

.container {
    width: 100%;
    max-width: 1140px;
    margin: auto;
    display: flex;
    flex-wrap: wrap;
    justify-content: space-around;
}

.footer-content {
    width: 100%;
    box-sizing: border-box;
    padding: 0 10px;
    margin-bottom: 20px;
}

h3 {
    font-size: 24px;
    margin-bottom: 10px;
    text-align: center;
}

.footer-content p {
    width: 100%;
    margin: auto;
    padding: 7px;
    text-align: center;
}
```

```
.footer-content ul {
    text-align: center;
    padding: 0;
}

.list li {
    width: auto;
    text-align: center;
    list-style-type: none;
    padding: 7px;
    position: relative;
}

.list li::before {
    content: '';
    position: absolute;
    transform: translate(-50%, -50%);
    left: 50%;
    top: 100%;
    width: 0;
    height: 2px;
    background: #f62e2e;
    transition-duration: .5s;
}

.list li:hover::before {
    width: 70px;
}

.social-icons {
    text-align: center;
    padding: 0;
}

.social-icons li {
    display: inline-block;
    text-align: center;
    padding: 5px;
}

.social-icons i {
    color: white;
    font-size: 20px;
}

a {
    text-decoration: none;
    color: white;
}
```



PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
a:hover {
    color: #fff;
}

.social-icons i:hover {
    color: #f18930;
}

.bottom-bar {
    background: #f62e2e;
    text-align: center;
    padding: 10px 0;
    margin-top: 20px;
}

.bottom-bar p {
    color: #FFF;
    margin: 0;
    font-size: 20px;
    padding: 7px;
}

img {
    max-width: 100%;
    height: auto;
    margin-bottom: 1em;
}

.img_po {
    width: 35px;
}

@media screen and (min-width: 768px) {
    .footer-content {
        max-width: 33.3%;
    }
}

    </style>
```

Este código é feito em CSS ele serve para dar estilos ao meu site.

Este código foi feito para o footer que me ajudou a deixar tudo mais organizado, como por exemplo o endereço, contatos, links para páginas importantes, termos e condições de troca e devolução, redes sociais e mais.



PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```css
* {
    padding: 0;
    margin: 0;
    font-family: 'Inter', sans-serif;
}

.login {
    background: rgba(0,0,0,.5);
    border-radius: 10px;
    width: 400px;
    padding: 40px;
}

header {
    background-color: #24252a;
    box-shadow: 0px 3px 10px #464646;
}

.nav-bar {
    display: flex;
    justify-content: space-between;
    padding: 1.5rem 6rem;
}

.logo {
    display: flex;
    align-items: center;
}

    .logo h1 {
        color: #fff;
    }

.nav-list {
    display: flex;
    align-items: center;
}

    .nav-list ul {
        display: flex;
        justify-content: center;
        list-style: none;
    }

.nav-item {
    margin: 0 15px;
```

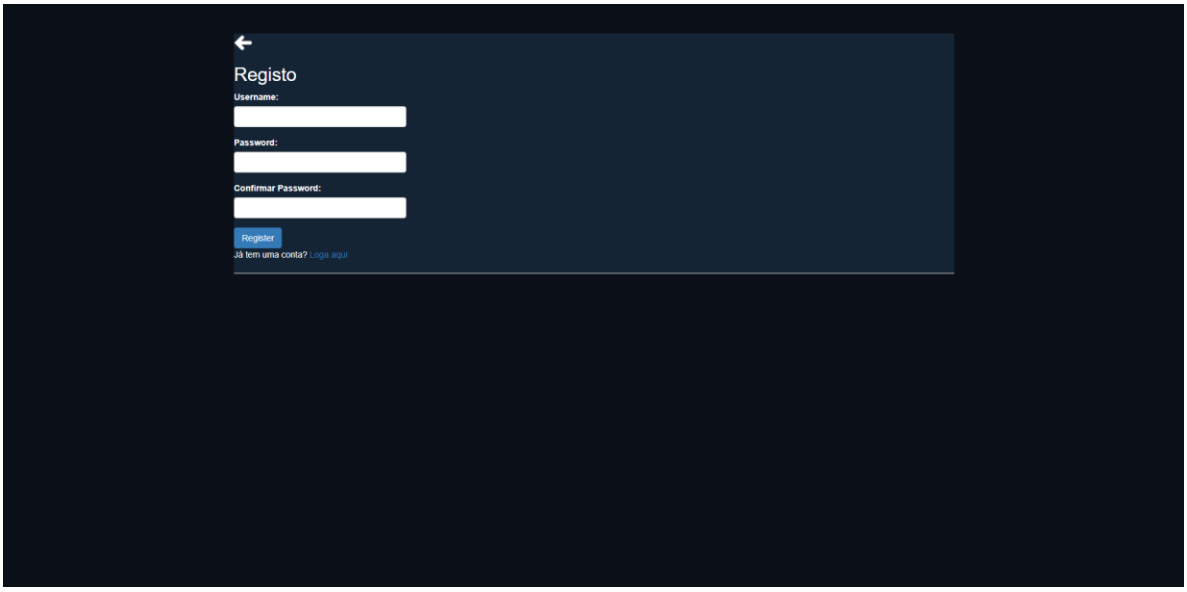
Este código também foi feito para dar estilos ao meu site e organizá-lo. Este código foi feito para dar estilos ao header e o menu responsivo que deixa o meu site muito mais organizado assim já sabemos onde podemos encontrar as páginas que queremos

## Login/Registo



PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

# Registo




Está é a página de Registo que serve para criar uma conta depois para logar na página de Login

Como a página mostra tem o username a password e confirmar a password, tem a seta para voltar para o menu inicial, mas depois de criar a conta e logar vai diretamente para a página inicial que foi mostrada acima

Se já tiver conta basta clicar em “Loga aqui” para logar a conta

```
<asp:Content ID="RegisterContent" ContentPlaceHolderID="MainContent" runat="server">
      <div style="background-color: #142434;">
    <style>
        body {
            background-color: #0B1018; /* Alterado para preto */
            color: #ffffff; /* Cor do texto branco */
        }
        .login-container {
            background-color: #142434; /* Cor de fundo azul claro */
            padding: 100px; /* Espaçamento interno */
        }
    </style>
    <div class="register-container">
        <a href="Home.aspx"><img src="imagens/seta.png" style="width:30px;height:30px;"></a>
        <h2>Registo</h2>
        <asp:Label ID="lblErrorMessage" runat="server" ForeColor="Red" Visible="false"></asp:Label>
        <asp:Label ID="lblSuccessMessage" runat="server" ForeColor="Green" Visible="false"></asp:Label>
        <div class="form-group">
            <label for="txtUsername">Username:</label>
            <asp:TextBox ID="txtUsername" runat="server" CssClass="form-control"></asp:TextBox>
        </div>
        <div class="form-group">
            <label for="txtPassword">Password:</label>
            <asp:TextBox ID="txtPassword" TextMode="Password" runat="server" CssClass="form-control"></asp:TextBox>
        </div>
        <div class="form-group">
            <label for="txtConfirmPassword">Confirm Password:</label>
            <asp:TextBox ID="txtConfirmPassword" TextMode="Password" runat="server" CssClass="form-control"></asp:TextBox>
        </div>
        <div>
            <asp:Button ID="btnRegister" runat="server" Text="Register" OnClick="Register_Click" CssClass="btn btn-primary" />
        </div>

        <p>
            Já tem uma conta? <a href="Login.aspx">Loga aqui</a>
        </p>
    </div>
</asp:Content>
```

```
public partial class Register : System.Web.UI.Page
{
    protected void Register_Click(object sender, EventArgs e)
    {
        try {
         string connectionString = ConfigurationManager.ConnectionStrings["MyDBConnectionString"].ConnectionString;

         using (SqlConnection con = new SqlConnection(connectionString))
         {
             con.Open();
             string checkUser = "SELECT COUNT(*) FROM Users WHERE Username=@username";
             using (SqlCommand checkCmd = new SqlCommand(checkUser, con))
             {
                 checkCmd.Parameters.AddWithValue("@username", txtUsername.Text.Trim());

                 int userExists = (int)checkCmd.ExecuteScalar();
                 if (userExists > 0)
                 {
                     lblErrorMessage.Visible = true;
                     lblErrorMessage.Text = "Username already exists.";
                 }
                 else
                 {
                     string insertQuery = "INSERT INTO Users (Username, Password) VALUES (@username, @password)";
                     using (SqlCommand insertCmd = new SqlCommand(insertQuery, con))
                     {
                         insertCmd.Parameters.AddWithValue("@username", txtUsername.Text.Trim());
                         insertCmd.Parameters.AddWithValue("@password", txtPassword.Text.Trim());

                         int result = insertCmd.ExecuteNonQuery();
                         if (result > 0)
                         {
                             lblSuccessMessage.Visible = true;
                             lblSuccessMessage.Text = "Registration successful.";
                             Response.Redirect("Login.aspx"); // Optionally redirect to the login page
                         }
                         else
                         {
                             lblErrorMessage.Visible = true;
                             lblErrorMessage.Text = "Registration failed.";
                         }
                     }
                 }
             }
         }
         } catch (Exception ex)
             {
                 lblErrorMessage.Visible = true;
                 lblErrorMessage.Text = "Error: " + ex.Message;
                 Console.WriteLine(ex.ToString());
             }
    }
}
}
```

Este código em C# é parte de uma página web ASP.NET que trata do registo de utilizadores

- O método **Register_Click** é invocado quando o botão de registo é clicado na página.



PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

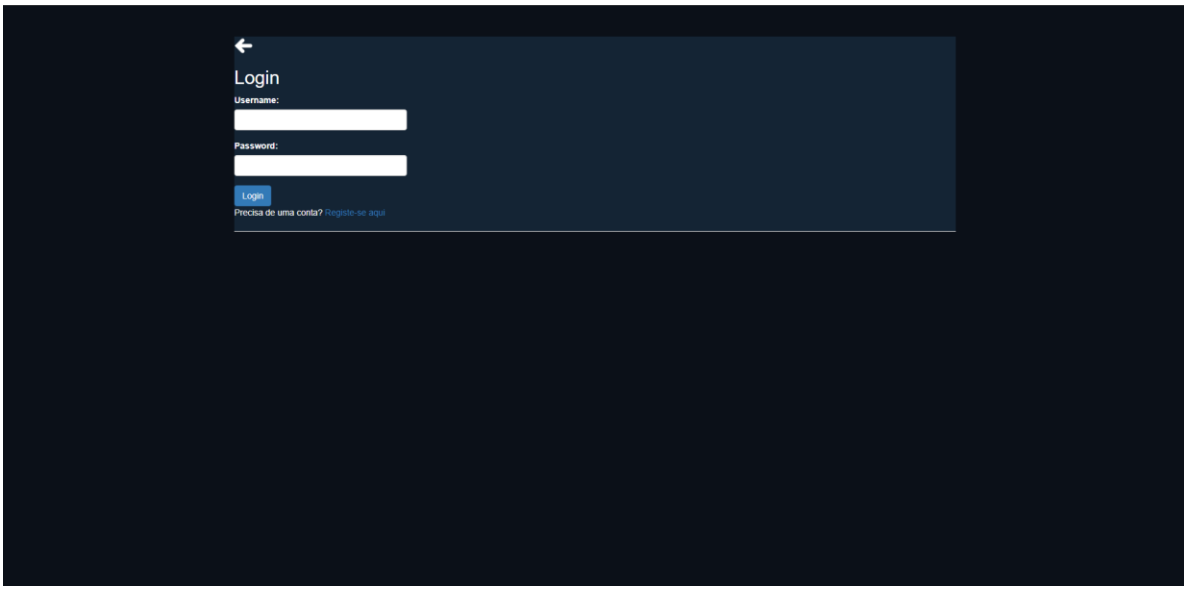
- Primeiro, ele tenta estabelecer uma conexão com a base de dados usando a string de conexão definida no arquivo de configuração.
- Em seguida, verifica se o nome de utilizador já existe na base de dados.
- Se o utilizador já existir, exibe uma mensagem de erro.
- Caso contrário, insere o novo utilizador na base de dados.
- Após a inserção bem-sucedida, exibe uma mensagem de sucesso e redireciona o utilizador para a página de login.
- Se ocorrer algum erro durante o processo, exibe uma mensagem de erro detalhada.

Este código gerência o registo de utilizadores em uma aplicação web ASP.NET, assegurando que não haja duplicatas de nomes de utilizador e fornecendo feedback ao utilizador sobre o resultado do registo.

## Login



PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.




Está é a página de Login que serve para logar a conta depois de termos criado o nosso registo.

Como a página mostra tem o username e a password que o usuário vai ter de preencher tem a seta para voltar para o menu inicial, mas se já tiver uma conta e logar vai diretamente para a página inicial que foi mostrada acima

Se não tiver conta basta clicar em "Registe-se aqui" para criar a conta

```
<asp:Content ID="LoginContent" ContentPlaceHolderID="MainContent" runat="server">
        <div style="background-color: #142434;">
    <style>
        body {
            background-color: #0B1018; /* Alterado para preto */
            color: #ffffff; /* Cor do texto branco */
        }
        .login-container {
            background-color: #142434; /* Cor de fundo azul claro */
            padding: 0px; /* Espaçamento interno */
        }
    </style>
    <div class="login-container">
        <a href="Home.aspx"><img src="imagens/seta.png" style="width:30px;height:30px;"></a>
        <h2>Login</h2>
        <asp:Label ID="lblErrorMessage" runat="server" ForeColor="Red" Visible="false"></asp:Label>
        <div class="form-group">
            <label for="txtUsername">Username:</label>
            <asp:TextBox ID="txtUsername" runat="server" CssClass="form-control"></asp:TextBox>
        </div>
        <div class="form-group">
            <label for="txtPassword">Password:</label>
            <asp:TextBox ID="txtPassword" TextMode="Password" runat="server" CssClass="form-control"></asp:TextBox>
        </div>
        <div>
            <asp:Button ID="btnLogin" runat="server" Text="Login" OnClick="Login_Click" CssClass="btn btn-primary" />
        </div>
        <p>
            Precisa de uma conta? <a href="Register.aspx">Registe-se aqui</a>
        </p>
    </div>
</asp:Content>
```

Este código faz parte de uma página de login desenvolvida usando ASP.NET. Inclui um formulário de login com campos para nome de usuário e senha, um botão para enviar informações de login e um link para a página de registo. O estilo de página é definido com cores de plano de fundo e texto específicos, e há uma seção de estilo embutido para estilizar elementos específico.

```
public partial class Login : System.Web.UI.Page
{
    protected void Login_Click(object sender, EventArgs e)
    {
        // Get the connection string from web.config
        string connectionString = ConfigurationManager.ConnectionStrings["MyDBConnectionString"].ConnectionString;

        // Use a 'using' block to ensure the connection is closed properly
        using (SqlConnection con = new SqlConnection(connectionString))
        {
            try
            {
                // Open the SQL connection
                con.Open();
                string query = "SELECT COUNT(1) FROM Users WHERE Username=@username AND Password=@password";

                // Create a SqlCommand object
                using (SqlCommand cmd = new SqlCommand(query, con))
                {
                    // Add parameters to prevent SQL injection
                    cmd.Parameters.AddWithValue("@username", txtUsername.Text.Trim());
                    cmd.Parameters.AddWithValue("@password", txtPassword.Text.Trim());

                    // Execute the query and get the result
                    int count = Convert.ToInt32(cmd.ExecuteScalar());

                    if (count == 1)
                    {
                        // User is found
                        Session["username"] = txtUsername.Text;
                        Response.Redirect("Home.aspx"); // Redirect to the homepage
                    }
                    else
                    {
                        // User is not found
                        lblErrorMessage.Visible = true;
                        lblErrorMessage.Text = "Username or password is incorrect.";
                    }
                }
            }
            catch (SqlException ex)
            {
                // Log the SQL exception
                lblErrorMessage.Visible = true;
                lblErrorMessage.Text = "SQL Error: " + ex.Message;
            }
            catch (Exception ex)
            {
                // Log other exceptions
                lblErrorMessage.Visible = true;
                lblErrorMessage.Text = "Error: " + ex.Message;
            }
        }
    }
}
}
```

Este código é uma parte de uma página de login em uma aplicação web ASP.NET. Ele verifica as credenciais do usuário em uma base de dados.

2. Conexão com o Banco de Dados: Obtém a string de conexão do arquivo web.config e abre uma conexão com o banco de dados usando a classe SqlConnection.

3. Consulta SQL: Constrói uma consulta SQL para verificar se o usuário existe na tabela Users com o nome de usuário e senha fornecidos.

4. Prevenção de Injeção SQL: Usa parâmetros na consulta para evitar ataques de injeção SQL.

5. Execução da Consulta: Executa a consulta usando ExecuteScalar() e verifica o resultado.

6. Redirecionamento: Se o usuário for encontrado, armazena o nome de usuário na sessão e redireciona para a página inicial. Se não for encontrado, exibe uma mensagem de erro.

7. Tratamento de Exceções: Captura exceções SQL e outras exceções, exibindo mensagens de erro adequadas.

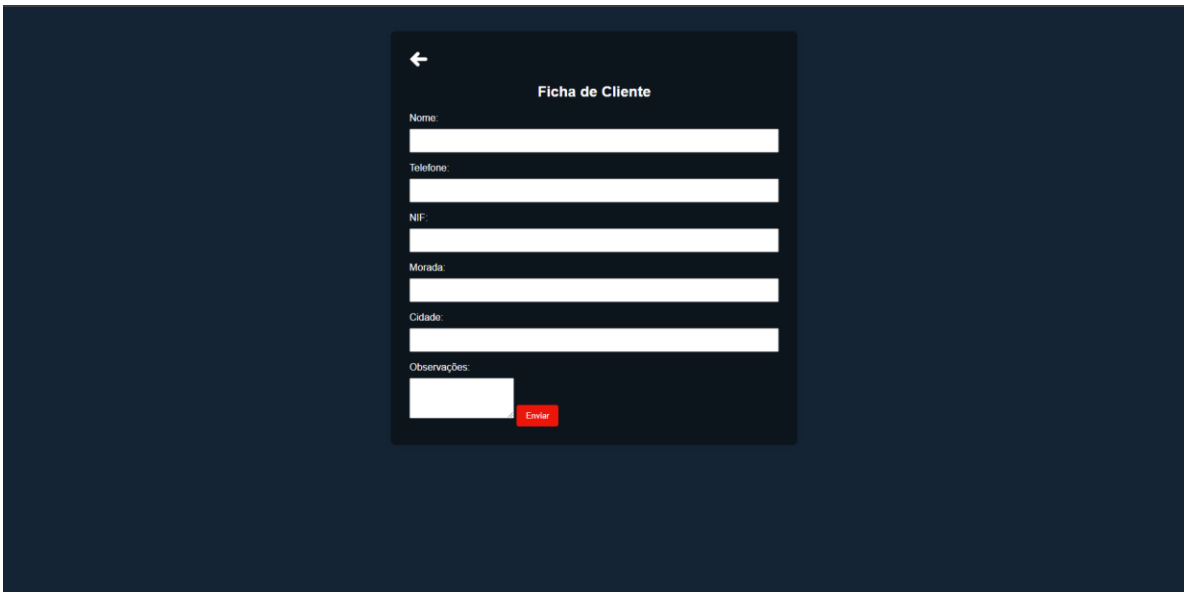
8. Fechamento de Conexão: Usa um bloco using para garantir que a conexão com o banco de dados seja fechada corretamente, independentemente de ocorrerem erros.

# Ficha Cliente e Ficha de Reparação

# Ficha Cliente



PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.




Está é a página ficha cliente que serve para o cliente meter os seus dados



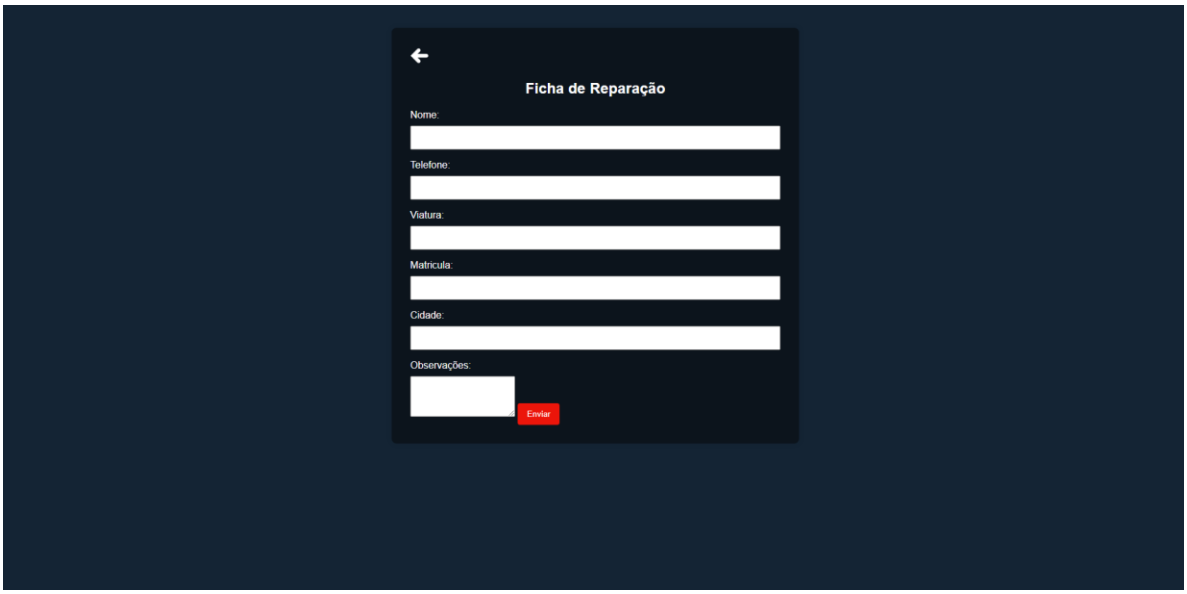
PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

```
<!DOCTYPE html>
<html lang="pt">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Ficha de Cliente</title>
<style>
        body {
            font-family: Arial, sans-serif;
            background-color: #142434;
            margin: 0;
            padding: 0;
        }
        form {
            max-width: 600px;
            margin: 40px auto;
            padding: 30px;
            background-color: #0c141c;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            color: #fff;
        }
        label {
            display: block;
            margin-bottom: 8px;
            color: #fff;
        }
        input, select {
            width: 100%;
            padding: 10px;
            margin-bottom: 16px;
            box-sizing: border-box;
        }
```

```
        button {
            background-color: #EC150A;
            color: #fff;
            padding: 10px 15px;
            border: none;
            border-radius: 4px;
            cursor: pointer;
        }
        button:hover {
            background-color: #9E150A;
        }
</style>
</head>
<body>
<form>
        <a href="Home.aspx"><img src="imagens/seta.png" " style="width:30px;height:30px;"></a>
    <center>
<h2>Ficha de Cliente</h2>
        </center>
<label for="nome">Nome:</label>
<input type="text" id="nome" name="nome" required>
<label for="telefone">Telefone:</label>
<input type="tel" id="telefone" name="telefone">
<label for="nif">NIF:</label>
<input type="nif" id="nif" name=nif" required>
<label for="morada">Morada:</label>
<input type="text" id="morada" name="morada">
<label for="cidade">Cidade:</label>
<input type="text" id="cidade" name="cidade">
<!-- Adicione mais opções conforme necessário -->
</select>
<label for="observacoes">Observações:</label>
<textarea id="observacoes" name="observacoes" rows="4"></textarea>
<button type="submit">Enviar</button>
</form>
</body>
</html>
```

O código HTML abaixo cria um formulário de Ficha de Cliente com campos de entrada nome, telefone, NIF, Morada, Cidade e Observações. O código do estilo CSS customizado estabelece a aparência do formulário, incluindo cores de fundo e de texto e o estilo de borda arredondada.

## Ficha de reparação



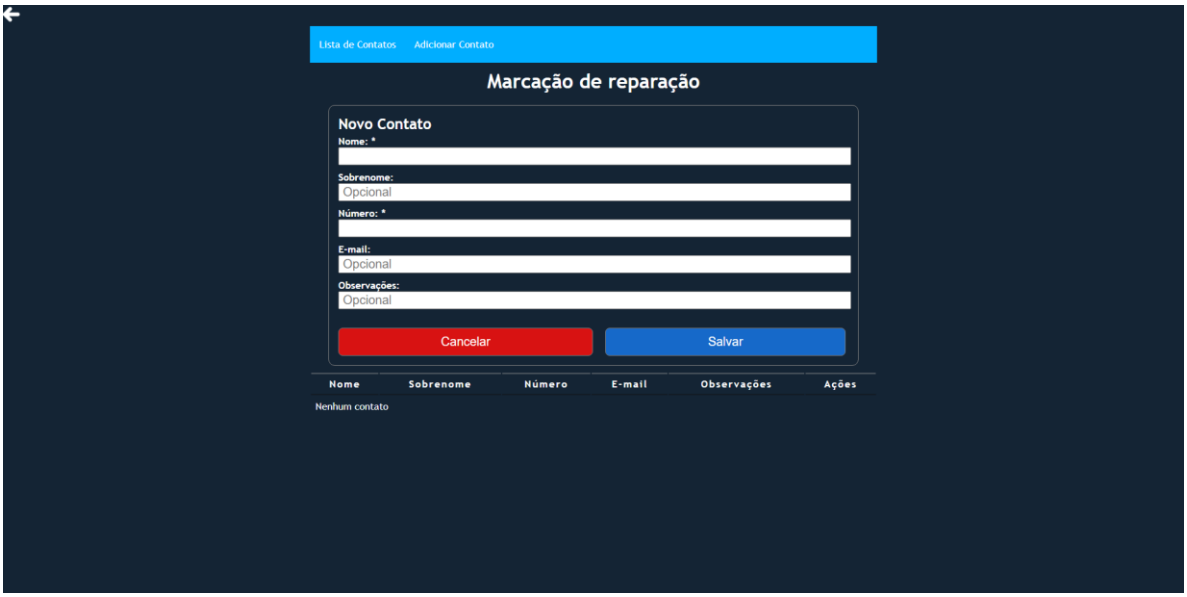

Está é a página da ficha de reparação que serve para o cliente meter os dados do seu automóvel.

```
<!DOCTYPE html>
<html lang="pt">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Ficha de Reparação</title>
<style>
        body {
            font-family: Arial, sans-serif;
            background-color: #142434;
            margin: 0;
            padding: 0;
        }
        form {
            max-width: 600px;
            margin: 40px auto;
            padding: 30px;
            background-color: #0c141c;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            color: #fff;
        }
        label {
            display: block;
            margin-bottom: 8px;
            color: #fff;
        }
        input, select {
            width: 100%;
            padding: 10px;
            margin-bottom: 16px;
            box-sizing: border-box;
        }
        button {
            background-color: #EC150A;
            color: #fff;
            padding: 10px 15px;
            border: none;
            border-radius: 4px;
            cursor: pointer;
        }
```

```
        button:hover {
            background-color: #9E150A;
        }
</style>
</head>
<body>
<form>
        <a href="pagoficial.aspx"><img src="imagens/seta.png" " style="width:30px;height:30px;"></a>
    <center>
<h2>Ficha de Reparação</h2>
        </center>
<label for="nome">Nome:</label>
<input type="text" id="nome" name="nome" required>
<label for="telefone">Telefone:</label>
<input type="tel" id="telefone" name="telefone">
<label for="email">Viatura:</label>
<input type="email" id="nif" name=nif" required>
<label for="morada">Matricula:</label>
<input type="text" id="morada" name="morada">
<label for="cidade">Cidade:</label>
<input type="text" id="cidade" name="cidade">
<!-- Adicione mais opções conforme necessário -->
</select>
<label for="observacoes">Observações:</label>
<textarea id="observacoes" name="observacoes" rows="4"></textarea>
<button type="submit">Enviar</button>
</form>
</body>
</html>
```

O código HTML abaixo cria um formulário de Ficha de Reparação com campos de entrada nome, telefone, viatura, matrícula, Cidade e Observações. O código do estilo CSS customizado estabelece a aparência do formulário, incluindo cores de fundo e de texto e o estilo de borda arredondada.

# Marcações





A ficha de marcações serve para agendar uma reparação ao seu automóvel.

Nela podemos adicionar os nossos dados tem os dados obrigatórios e os opcionais que não é preciso colocar para efetuar a marcação depois de adicionar os seus dados vão aparecer

na primeira página que esta na foto acima.

## HTML

```
<!DOCTYPE html>
<html lang="en">
<a href="Home.aspx"><img src="imagens/seta.png" style="width:30px;height:30px;"></a>
<head>
    <meta charset="UTF-8">
    <link rel="stylesheet" href="agenda.css">
    <title>Marcação de reparação</title>
</head>

<body>
    <div id="principal">
        <div class="cabecalho">
            <ul>
                <li><a id="telaInicio">Lista de Contatos</a></li>
                <li><a id="buttonNovoContato">Adicionar Contato</a></li>
            </ul>
        </div>
        <h1 class="titulo">Marcação de reparação</h1>
        <div id="add-contato" class="display">
            <form action="" id="formContato">
                <h2>Novo Contato</h2>
                <div>
                    <label for="nomeContato" class="nomeContato"><b>Nome: *</b><br></label>
                    <input type="text" name="novoContato" id="nomeContato">
                </div>
                <div>
                    <label for="sobrenomeContato" class="sobrenomeContato"><b>Sobrenome: </b><br></label>
                    <input type="text" name="sobrenomeContato" id="sobrenomeContato" placeholder=" Opcional">
                </div>
                <div>
                    <label for="numeroContato" class="numeroContato"><b>Número: *</b><br></label>
                    <input type="tel" name="numeroContato" id="numeroContato">
                </div>
                <div>
                    <label for="novoEmail" class="novoEmail"><b>E-mail: </b> <br></label>
                    <input type="email" name="novoEmail" id="emailContato" placeholder=" Opcional">
                </div>
                <div>
                    <label for="novaObservacao" class="novaObservacao"><b> Observações: </b><br></label>
                    <input type="text" name="novaObservacao" id="obsContato" placeholder=" Opcional">
                </div>
                <div id="mensagemErro" class="display">
```



PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET
Os Fundos Europeus mais próximos de si.
uma escola Rumos education | Knowledge sharing

```
                    Mensagem de erro!
                </div>
                <div>
                    <button type="button" class="cancelar" id="buttonCancelar">Cancelar</button>
                    <button type="submit" class="salvar">Salvar</button>
                </div>
            </form>
        </div>

        <table id="lista">
            <thead>
                <tr>
                    <th>Nome</th>
                    <th>Sobrenome</th>
                    <th>Número</th>
                    <th>E-mail</th>
                    <th>Observações</th>
                    <th>Ações</th>
                </tr>
            </thead>
            <tbody id="tabelaContatos">
            </tbody>
        </table>
    </div>
    <script src="agenda.js"></script>
    <style>
        body {
            background-color: #142434;
            color: white; /* Adicionando esta linha para definir a cor do texto como branco */
        }
    </style>
</body>

</html>
```

Este código HTML representa uma página para um sistema de marcação de reparação. Ele inclui um formulário para adicionar novos contatos, uma tabela para exibir contatos existentes e links para outras partes do sistema. O código também importa estilos CSS e scripts JavaScript para adicionar funcionalidades interativas. A página tem um esquema de cores escuro, com fundo azul escuro e texto branco.

## CSS

```css
* {
    margin: 0;
    padding: 0;
}

#principal {
    max-width: 920px;
    margin: 0px auto;
    font-family: "Trebuchet MS", Helvetica, 'Sans-serif';
}

.cabecalho {
    width: 100%;
    background: #00aeff;
    float: left;
}

.cabecalho ul {
    list-style: none;
    display: block;
}

.cabecalho ul li {
    padding: 5px 15px;
    float: left;
}

.cabecalho ul li a {
    cursor: pointer;
    display: block;
    padding: 15px 0px;
    background: url(./img/separador.png) no-repeat left;
    text-decoration: none;
    color: white;
}

.cabecalho a:hover {
    color: rgb(1, 85, 182);
    transition: 1s;
}
```



PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET
Os Fundos Europeus mais próximos de si.
uma escola Rumos education | Knowledge sharing

```
41
42  .titulo {
43      text-align: center;
44      padding: 70px 0px 20px 0px;
45  }
46
47  #add-contato {
48      clear: both;
49      padding: 15px;
50      border: 1px solid grey;
51      border-radius: 10px;
52      width: 90%;
53      margin: 0px auto;
54  }
55
56  .display {
57      display: none;
58  }
59
60  #add-contato h2 {
61      margin-bottom: 5px;
62  }
63
64  #add-contato input {
65      line-height: 25px;
66      margin-bottom: 10px;
67      font-size: 20px;
68      width: 100%;
69  }
70
71  #mensagemErro {
72      margin: 5px auto;
73      border: 1px solid rgb(255, 72, 72);
74      background: rgba(243, 0, 20, 0.363);
75      color: rgb(80, 17, 17);
76      border-radius: 10px;
77      width: 96%;
78      padding: 15px;
79  }
80
```

```
80
81  .cancelar {
82      border: 1px solid grey;
83      border-radius: 7px;
84      margin-top: 20px;
85      display: inline-block;
86      font-size: 20px;
87      padding: 10px 20%;
88      background: rgb(216, 18, 18);
89      color: white;
90      cursor: pointer;
91  }
92
93  .cancelar:hover {
94      background: rgb(243, 40, 40);
95      transition: 0.3s;
96      color: rgb(102, 15, 15);
97  }
98
99  .salvar {
100     margin-left: 15px;
101     border: 1px solid grey;
102     border-radius: 7px;
103     margin-top: 20px;
104     display: inline-block;
105     font-size: 20px;
106     padding: 10px 20%;
107     background: rgb(22, 105, 201);
108     color: white;
109     cursor: pointer;
110 }
111
112 .salvar:hover {
113     transition: 0.3s;
114     background: rgb(43, 142, 255);
115     color: rgb(16, 58, 105);
116 }
117
118 #lista {
119     width: 100%;
120     margin-top: 10px;
121 }
```

```
122
123  #lista th, td {
124      padding: 7px;
125  }
126
127  tbody tr:hover {
128      background: rgb(187, 187, 187);
129  }
130
131  #lista th {
132      letter-spacing: 2px;
133      border-top: 1px solid #999;
134      border-bottom: 1px solid #111;
135  }
136
137  .campoInvalido {
138      border: 1.5px solid rgb(219, 46, 46);
139  }
140
141  .botaoDelete {
142      color: white;
143      background: red;
144      border: 1px solid grey;
145      border-radius: 5px;
146      padding: 5px;
147      cursor: pointer;
148  }
149
150  .botaoEditar {
151      margin-left: 5px;
152      color: white;
153      background: blue;
154      border: 1px solid grey;
155      border-radius: 5px;
156      padding: 5px;
157      cursor: pointer;
158  }
```

Este código CSS define estilos para uma página web. Ele faz o reset de margens e preenchimentos padrão, define o estilo do elemento principal da página, estiliza o cabeçalho, cria estilos para links dentro do cabeçalho, define estilos para títulos,

formulários de adição de contatos, mensagens de erro, botões de cancelar e salvar, uma tabela e seus elementos, campos inválidos em formulários e botões de deletar e editar. Cada parte do código contribui para uma apresentação visual consistente e agradável da página.

# Javascript

```javascript
//Declaração de variáveis globais
let buttonAddContato = document.querySelector('#buttonNovoContato')
let buttonCancelaContato = document.querySelector('#buttonCancelar')
let telaInicio = document.querySelector('#telaInicio')
let addContato = document.querySelector('#add-contato')
let formContato = document.querySelector('#formContato')
let InputNomeContato = document.querySelector('#nomeContato')
let InputsobrenomeContato = document.querySelector('#sobrenomeContato')
let InputNumeroContato = document.querySelector('#numeroContato')
let InputEmailContato = document.querySelector('#emailContato')
let InputObsContato = document.querySelector('#obsContato')
let divMensagemErro = document.querySelector('#mensagemErro')
let tabelaContatos = document.querySelector('#tabelaContatos')

//Criação da lista de contatos
var listaContatos = []

function removerContato(event) {
    var posicao = event.target.getAttribute('data-contato')
    listaContatos.splice(posicao, 1)
    atualizarTabelaContatos()
}

function atualizarTabelaContatos() {
    if (listaContatos.length === 0) {
        tabelaContatos.innerHTML = '<tr><td colspan="6">Nenhum contato</td></tr>'
        return
    }
    tabelaContatos.innerHTML = ''
    for (var i = 0; i < listaContatos.length; i++) {
        var contato = listaContatos[i]
        var linha = document.createElement('tr')
        var celulaNome = document.createElement('td')
        var celulaSobrenome = document.createElement('td')
        var celulaNumero = document.createElement('td')
        var celulaEmail = document.createElement('td')
        var celulaObs = document.createElement('td')
        var celulaAcoes = document.createElement('td')
        var botaoRemover = document.createElement('button')

        botaoRemover.setAttribute('data-contato', i)
        botaoRemover.classList.add('botaoDelete')
        botaoRemover.addEventListener('click', removerContato)
```

```
44
45              celulaNome.innerText = contato.nome
46              celulaSobrenome.innerText = contato.sobrenome
47              celulaNumero.innerText = contato.numero
48              celulaEmail.innerText = contato.email
49              celulaObs.innerText = contato.observacao
50              botaoRemover.innerText = 'Deletar'
51
52              celulaAcoes.appendChild(botaoRemover)
53              linha.appendChild(celulaNome)
54              linha.appendChild(celulaSobrenome)
55              linha.appendChild(celulaNumero)
56              linha.appendChild(celulaEmail)
57              linha.appendChild(celulaObs)
58              linha.appendChild(celulaAcoes)
59              tabelaContatos.appendChild(linha)
60      }
61  }
62
63  //Limpando os campos de escrita e removendo a tela de Novo Contato
64  function limpaContato() {
65      InputNomeContato.classList.remove('campoInvalido')
66      InputNumeroContato.classList.remove('campoInvalido')
67      divMensagemErro.classList.add('display')
68      InputNomeContato.value = ''
69      InputsobrenomeContato.value = ''
70      InputNumeroContato.value = ''
71      InputEmailContato.value = ''
72      InputObsContato.value = ''
73  }
74
75  //Botões "Salvar", "Cancelar" e "Lista de Contatos"
76  buttonAddContato.addEventListener('click', function novoContato() {
77      addContato.classList.remove('display')
78  })
79
80  function cancelaContato() {
81      addContato.classList.add('display')
82      limpaContato()
83  }
84  buttonCancelaContato.addEventListener('click', cancelaContato)
85  telaInicio.addEventListener('click', cancelaContato)
```

```
86
87    //Checando os campos "Nome" e "Email"
88    function validaContato(nomeContato, numeroContato) {
89        var validaCampo = true
90        var erro = ''
91        if (nomeContato.trim().length === 0) {
92            erro = 'Insira o Nome do contato!'
93            InputNomeContato.classList.add('campoInvalido')
94            validaCampo = false
95        } else {
96            InputNomeContato.classList.remove('campoInvalido')
97        }
98        var numeroContatoLimpo = numeroContato.trim().length
99        if (numeroContatoLimpo === 0) {
100           if (erro.length > 0) {
101               erro += '<br>'
102           }
103           erro += 'Insira o número do contato!'
104           InputNumeroContato.classList.add('campoInvalido')
105           validaCampo = false
106       } else {
107           InputNumeroContato.classList.remove('campoInvalido')
108       }
109
110       if (!validaCampo) {
111           divMensagemErro.innerHTML = erro
112           divMensagemErro.classList.remove('display')
113       } else {
114           divMensagemErro.classList.add('display')
115       }
116
117       return validaCampo
118   }
119
```

```
120  //Transferindo os dados para a tabela em caso de sucesso
121  function salvarContato(event) {
122      event.preventDefault()
123      var nomeContato = InputNomeContato.value
124      var numeroContato = InputNumeroContato.value
125      if (validaContato(nomeContato, numeroContato)) {
126          console.log('Contato válido')
127          listaContatos.push({
128              nome: nomeContato,
129              numero: numeroContato,
130              sobrenome: sobrenomeContato.value,
131              email: emailContato.value,
132              observacao: obsContato.value,
133          })
134
135          atualizarTabelaContatos()
136          cancelaContato()
137          alert('Contato adicionado com sucesso!')
138      } else {
139
140      }
141  }
142
143  formContato.addEventListener('submit', salvarContato)
144  window.addEventListener('load', atualizarTabelaContatos)
```

Este código cria uma aplicação simples de lista de contatos em uma página web. Ele permite adicionar novos contatos, validando os campos de nome e número, e exibe-os em uma tabela. Os contatos podem ser removidos individualmente. O código organiza as funcionalidades em funções e usa event listeners para interações do usuário, como adicionar e cancelar contatos.



PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

# 3. Conclusão

**Objetivos Alcançados:**

O meu maior objetivo alcançado foi realizar este projeto ao longo destes meses.

**Dificuldades / Facilidades Sentidas:**

Muitas dificuldades a conseguir por o site responsivo e conseguir criar a base de dados.

A maior facilidade que eu senti foi pesquisar os vídeos que eu queria.

**Benefícios / Vantagens / Interesse do Projeto Desenvolvido:**

Com este projeto desenvolvido aprendi muita coisa, aprendi que não devo desistir dos meus objetivos, a minha maior vantagem foi a aprendizagem que tive nestes últimos 3 anos que ajudou imenso ao fazer o meu projeto final.



PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

# Bibliografia

Eu retirei várias coisas do youtube vi tutoriais para aprender como fazer o que queria como o header o footer, ficha cliente, ficha de reparação, a página sobre o meu site.

Os vídeos do youtube ajudaram me imenso principalmente na parte de conseguir meter o meu site responsivo.

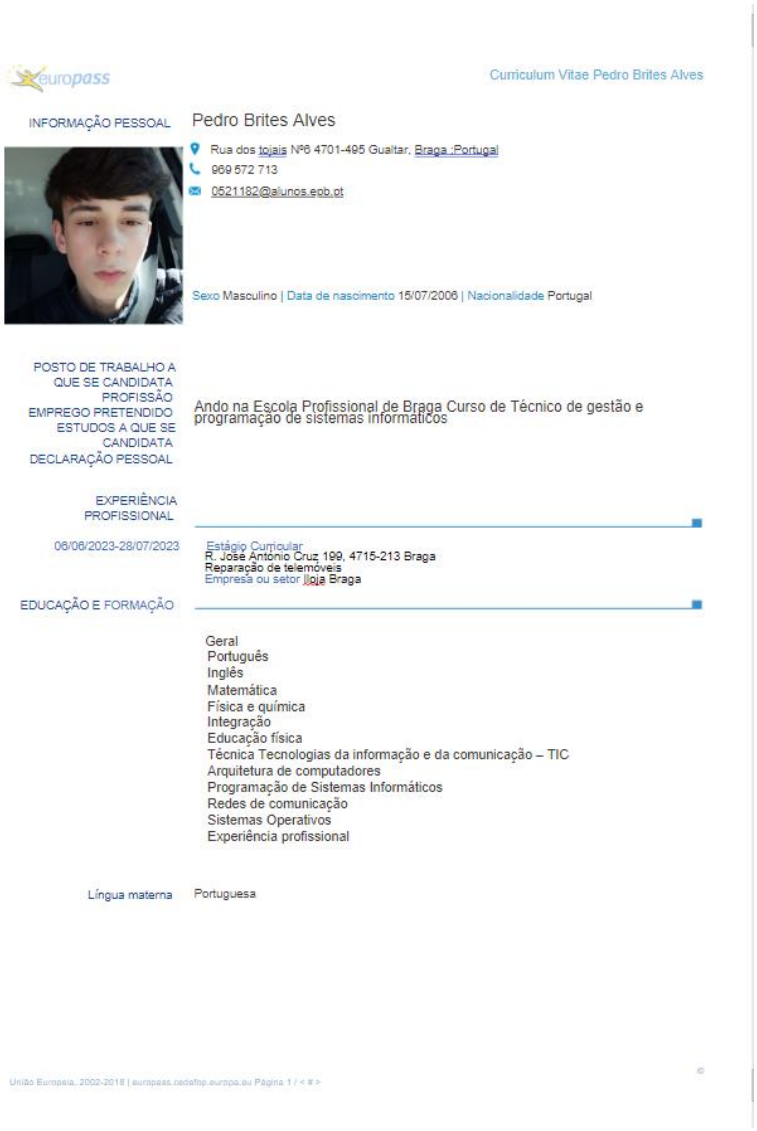
Também usei o ChatGPT para correção dos meus erros e os erros dos códigos.



PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing
Os Fundos Europeus mais próximos de si.

# Anexos

## Curriculum Vitae



europass Curriculum Vitae Pedro Brites Alves

INFORMAÇÃO PESSOAL **Pedro Brites Alves**

Rua dos tojais Nº6 4701-495 Gualtar, Braga ,Portugal
969 572 713
0521182@alunos.epb.pt

Sexo Masculino | Data de nascimento 15/07/2006 | Nacionalidade Portugal

POSTO DE TRABALHO A QUE SE CANDIDATA
PROFISSÃO
EMPREGO PRETENDIDO
ESTUDOS A QUE SE CANDIDATA
DECLARAÇÃO PESSOAL

Ando na Escola Profissional de Braga Curso de Técnico de gestão e programação de sistemas informáticos

EXPERIÊNCIA PROFISSIONAL

06/06/2023-28/07/2023 Estágio Curricular
R. José António Cruz 199, 4715-213 Braga
Reparação de telemóveis
Empresa ou setor loja Braga

EDUCAÇÃO E FORMAÇÃO

Geral
Português
Inglês
Matemática
Física e química
Integração
Educação física
Técnica Tecnologias da informação e da comunicação – TIC
Arquitetura de computadores
Programação de Sistemas Informáticos
Redes de comunicação
Sistemas Operativos
Experiência profissional

Língua materna Portuguesa

europass Curriculum Vitae Pedro Brites Alves

Competências digitais

| AUTOAVALIAÇÃO | | | | |
|---|---|---|---|---|
| Processamento de informação | Comunicação | Criação de conteúdos | Segurança | Resolução de problemas |
| Utilizador avançado | Utilizador independente | Utilizador independente | Utilizador avançado | Utilizador independente |

Níveis: utilizador básico - utilizador independente - utilizador avançado
Competências digitais - Grelha de auto-avaliação

Indique o(s) certificado(s) TIC

Descreva outras competências informáticas. Indique o contexto em que foram adquiridas. Exemplos:
- Conhecimentos Windows Desktop, Server e família Linux [Sistemas Operativos]
- Python, C#, Html, JavaScript e PHP [Linguagem de Programação]
- Programação Mobile
- Access, Mysql e SQL [Base de Dados]
- Office e Open Source [Software de Aplicação]

ANEXOS




PESSOAS 2030 PORTUGAL 2030 Cofinanciado pela União Europeia SELO DE CONFORMIDADE EQAVET uma escola Rumos education Knowledge sharing

Os Fundos Europeus mais próximos de si.

EUSOUDIGITAL
PROGRAMA DE CAPACITAÇÃO DIGITAL DE ADULTOS

CERTIFICADO

Pedro Brites
Mentor Nº 110639

Concluiu em março de 2023 com sucesso a Sessão de Formação de Mentores EUSOUDIGITAL que lhe permite desenvolver Sessões de Capacitação Digital de adultos.