

Prova de Aptidão Profissional

CURSO PROFISSIONAL DE NÍVEL SECUNDÁRIO

Técnico/a de Gestão e Programação de Sistemas Informáticos



“ARDUINO CODE MASTERS”

Bruno Miguel Gouveia Vieira (a32496)

Janeiro 2024

Agradecimentos

Estimados Professores Daniela Cervan, Rui Costa, Jorge Grante, João Campos e Rui Carvalho,

Gostaria de expressar o meu sincero agradecimento pela valiosa ajuda e orientação durante o desenvolvimento do site "ARDUINO CODE MASTERS". A vossa dedicação e conhecimento foram essenciais para o sucesso do projeto. Agradeço sinceramente pela inspiração, paciência e contribuições significativas.

Índice

Agradecimentos.....	2
Índice de ilustrações.....	4
Introdução	6
Fundamentação do Projeto:	7
Material e métodos.....	9
Ferramentas Utilizadas	9
Linguagens Utilizadas.....	13
Produto.....	16
Diagrama de casos de uso.....	16
Página inicial.....	17
Criação de Conteúdo para as páginas:	20
Página Criar conta.....	20
PHP Criar Conta.....	23
Página de login.....	27
PHP Login.....	30
Página sobre o Arduino em si.....	33
Página dos projetos.....	34
PHP Projetos.....	36
Propor Projetos.....	39
PHP Propor Projetos.....	42
Página dos top 5 componentes mais utilizados no sítio ‘web’	49
Página das lojas.....	53
Forum de perguntas e respostas	55
PHP Forum para submeter uma pergunta	57
PHP das respostas.....	59
Página de contacto	61
Página da conta de utilizador	63
Página para editar detalhes de utilizador.....	65
Dashboard	68
Página de gestão de utilizadores	71
Página adicionar componentes	73
Página Associar Loja a Componente.....	75
Página Adicionar Categoria	77
Página Associar Componente a Categoria.....	79
Página adicionar lojas.....	81

Página adicionar projetos do admin	83
Projetos pendentes para verificação.....	85
Base de dados.....	87
Ligação.....	87
Apresentarei então a estrutura das 11 tabelas.....	90
Modelo Relacional.....	95
Resultados do Desenvolvimento	96
Conclusão.....	97
Autoavaliação.....	98
Webgrafia.....	100

Índice de ilustrações

FIGURA 1	9
FIGURA 2	9
FIGURA 3	10
FIGURA 4	10
FIGURA 5	11
FIGURA 6	11
FIGURA 7	12
FIGURA 8	13
FIGURA 9	13
FIGURA 10	14
FIGURA 11	14
FIGURA 12	15
FIGURA 13	16
FIGURA 14	17
FIGURA 15	18
FIGURA 16	18
FIGURA 17	19
FIGURA 18	20
FIGURA 19 PHP 1	23
FIGURA 20 PHP 2	24
FIGURA 21 PHP 3	24
FIGURA 22 MAILHOG 1	25
FIGURA 23	27
FIGURA 24 PHP 4	30
FIGURA 25 PHP 5	30
FIGURA 26	33
FIGURA 27	34
FIGURA 28 PHP 6	36

FIGURA 29 PHP 7	37
FIGURA 30	39
FIGURA 31	39
FIGURA 32	39
FIGURA 33 PHP 8	42
FIGURA 34 PHP 9	45
FIGURA 35 PHP 10	46
FIGURA 36	49
FIGURA 37 PHP 11	51
FIGURA 38	53
FIGURA 39	55
FIGURA 40 PHP 12	57
FIGURA 41 PHP 13	59
FIGURA 42	61
FIGURA 43	63
FIGURA 44	65
FIGURA 45	68
FIGURA 46	71
FIGURA 47	73
FIGURA 48	75
FIGURA 49	77
FIGURA 50	79
FIGURA 51	81
FIGURA 52	83
FIGURA 53	83
FIGURA 54	85
FIGURA 55 PHP 14	87
FIGURA 56 BASE DE DADOS 1	90
FIGURA 57 BASE DE DADOS 2	90
FIGURA 58 BASE DE DADOS 3	91
FIGURA 59 BASE DE DADOS 4	91
FIGURA 60 BASE DE DADOS 5	92
FIGURA 61 BASE DE DADOS 6	92
FIGURA 62 BASE DE DADOS 7	92
FIGURA 63 BASE DE DADOS 8	93
FIGURA 64 BASE DE DADOS 9	93
FIGURA 65 BASE DE DADOS 10	93
FIGURA 66 BASE DE DADOS 11	93
FIGURA 67 BASE DE DADOS 12	95

Introdução

A Prova de Aptidão Profissional (PAP) é um projeto inserido no âmbito do curso profissional de Técnico de Gestão e Programação de Sistemas Informáticos. O tema do presente projeto é um sítio 'web' para pessoas que querem aprender a utilizar o Arduino. O objetivo principal deste projeto é fazer um sítio 'web' sobre como utilizar o Arduino. O tema escolhido planeia ajudar pessoas que começam a fazer projetos nesta área, dando sugestões, esclarecendo dúvidas e corrigindo erros que advêm da utilização do mesmo. Quando terminar este trabalho, o formando espera ter conseguido atingir o objetivo pretendido e ter ajudado muita gente.

Fundamentação do Projeto:

O presente projeto fundamenta-se na necessidade premente de criar uma plataforma online dedicada ao ensino e colaboração em Arduino. O Arduino, reconhecido como uma plataforma de prototipagem eletrónica e programação, desempenha um papel crucial no desenvolvimento de competências práticas em eletrónica. A proposta de criar esta plataforma justifica-se pelos seguintes pontos:

Contextualização do Arduino:

O Arduino surgiu como uma ferramenta de código aberto, tornando-se imprescindível para o ensino prático de eletrónica e programação. A sua versatilidade e acessibilidade fizeram dele uma escolha popular em ambientes educacionais e projetos DIY (Do It Yourself).

Relevância do Projeto:

A plataforma proposta atende à crescente procura por recursos educacionais práticos e interativos. Num cenário de ensino à distância, torna-se essencial proporcionar um espaço centralizado para estudantes, professores e especialistas partilharem conhecimentos, experiências e projetos relacionados ao Arduino.

Utilidade da Aplicação a Desenvolver:

A aplicação a ser desenvolvida facilitará a criação, documentação e partilha de projetos Arduino. A inclusão de um fórum de discussão proporcionará um ambiente dinâmico para a troca de ideias, resolução de dúvidas e colaboração ativa entre os utilizadores. O objetivo é proporcionar uma experiência de aprendizagem envolvente e prática.

Abrangência (Público-Alvo):

O público-alvo da plataforma abrange estudantes em busca de aprimoramento em eletrónica e programação, professores interessados em enriquecer os seus métodos de ensino com recursos práticos, e especialistas que desejam partilhar conhecimentos. A abordagem adotada visa atender às diversas necessidades e níveis de experiência desse público, promovendo um ambiente inclusivo e colaborativo.

Desta forma, a proposta de criar esta plataforma visa preencher uma lacuna educacional, proporcionando um espaço online dedicado ao ensino prático e à colaboração efetiva no universo do Arduino.

Material e métodos

Ferramentas Utilizadas

Visual Studio Code - Editor de Código: o Visual Studio Code é como o gabinete do arquiteto onde uma casa é planeada e desenhada. De facto, é uma aplicação onde é concebido o "plano" do sítio 'web', para criar a estrutura, o 'design' e a interatividade do projeto, usando linguagens de programação: HTML, CSS e JavaScript.



Figura 1

XAMPP - Servidor Local: o XAMPP é uma base de dados que permite que tudo realmente funcione (utilizadores, projetos, mensagens, ...). É um "local" especial onde se coloca o sítio 'web', para ser testado antes de ser tornado público. É como verificar se todas as luzes, a água e os aparelhos da casa estão a funcionar corretamente antes de convidares visitantes.



Figura 2

Ferramenta de Recorte Windows — Recorte de Imagens: imagine-se uma coleção de fotos que se planeia exibir em diferentes partes da casa. A Ferramenta de Recorte Windows é como uma tesoura que ajuda a moldar essas fotos como desejado. Podem-se, assim, ajustar as imagens para encaixarem perfeitamente nas diferentes áreas do projeto.



Figura 3

RemoveBG - Remove-dor de Fundo de Imagens: o RemoveBG é como uma ferramenta mágica que ajuda a destacar objetos das fotos, permitindo recortar e remover o fundo de uma imagem.



Figura 4

Splendid Logo Maker - Criador de Logotipos: o Splendid Logo Maker é como um artista que cria o cartão de visitas da casa. Ajuda a desenhar um logótipo único que representa o projeto, criando um símbolo especial, que as pessoas vão reconhecer sempre que visitarem aquele espaço virtual.



Figura 5

Tinkercad - o Tinkercad é uma plataforma 'online' que permite criar e simular circuitos eletrónicos virtualmente. É como uma oficina virtual onde se podem montar projetos eletrónicos usando peças virtuais, como resistências, 'Leds' e sensores. Não são necessários componentes reais, pois tudo é feito digitalmente. Podem testar-se os circuitos e ver como funcionam antes de os construir fisicamente. É uma ferramenta muito útil para aprender eletrónica e programação, como no projeto "Arduino Code Masters", onde é possível criar e testar códigos de Arduino antes de implementá-los no mundo real.



Figura 6

MailHog - o MailHog é uma ferramenta que permite aos desenvolvedores testar o envio de correios eletrónicos em ambientes de desenvolvimento, capturando e exibindo os correios eletrónicos enviados sem realmente os enviar para destinatários reais. Isso ajuda a verificar o funcionamento de funcionalidades de correio eletrónico em aplicativos, sem riscos, durante o desenvolvimento e teste de 'software'.



Figura 7

Em síntese, cada uma das ferramentas usadas para construir um projeto no “Arduino Code Masters” tem um papel específico na criação e no aprimoramento de um sítio ‘web’, assegurando que seja atrativo, interativo e informativo para os visitantes.

Linguagens Utilizadas

HTML (Linguagem de Marcação de Hipertexto): Imagina que estás a construir uma casa e precisas de organizar as peças. O HTML é como as instruções que dizes aos construtores. Eles sabem onde colocar as paredes, as portas e as janelas. No mundo 'online', o HTML é usado para dizer ao navegador como mostrar uma página. Usamos comandos chamados "etiquetas" para dizer onde colocar coisas como títulos, imagens e botões.



Figura 8

CSS (Folhas de Estilo em Cascata): Agora, pensa em como queres que a tua casa pareça por fora e por dentro. O CSS é como as instruções para a decoração. Dizes qual cor pintar as paredes, que tipo de letras usar e como as coisas devem ficar arrumadas. No mundo 'online', o CSS é usado para fazer com que as páginas 'Web' pareçam bonitas e organizadas.



Figura 9

JavaScript: Imagina que queres que as luzes da tua casa acendam quando alguém entra. O JavaScript é como o interruptor que controla essas luzes. Ele torna as páginas 'Web' mais interativas, permitindo que coisas aconteçam quando alguém clica num botão, insere informações ou move o rato.



Figura 10

SQL (Linguagem de Consulta Estruturada): Pensa nas prateleiras de uma biblioteca com diferentes tipos de livros. O SQL é como a forma de pedir os livros que queres. Ele é usado para falar com uma "biblioteca de dados" que guarda informações. Podes dizer para procurar, adicionar, atualizar ou apagar informações nessa "biblioteca".



Figura 11

PHP (Pré-processador de Hipertexto): Imagina que queres que a tua casa saiba quem está a visitar. O PHP é como o assistente que faz isso acontecer. Ele é usado para conectar a tua página 'Web' a um lugar onde podes guardar informações, como um armário. Assim, podes mostrar coisas diferentes às pessoas com base no que elas fazem.



Figura 12

Produto

Diagrama de casos de uso

https://lucid.app/lucidchart/128c6b32-66f6-41ab-bb27-37e3368b0ae1/edit?viewport_loc=-26961%2C-33372%2C12564%2C4637%2C0_0&invitationId=inv_66297d07-aa27-4af0-8c08-103a29daa4f6

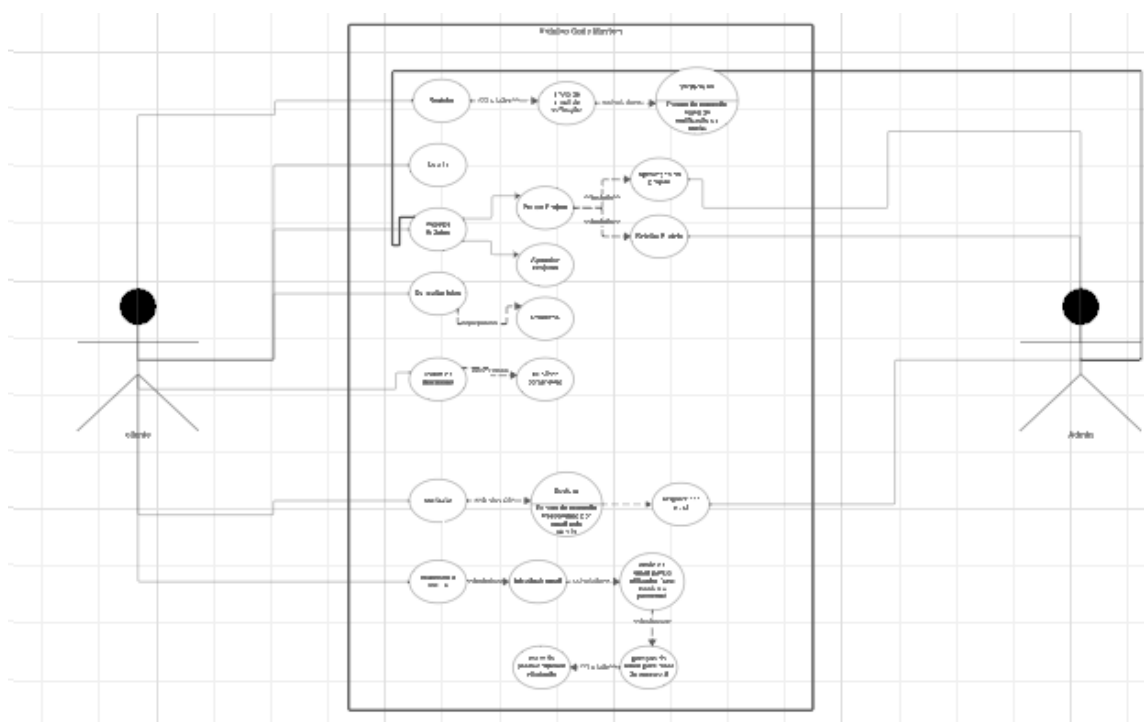


Figura 13

Página inicial

O processo de construir um sítio 'web' a partir do nada é como erguer uma estrutura a partir do solo. Nesta jornada, criei, desenvolvi e personalizei o projeto "Arduino Code Masters" desde o início. Cada linha de código, cada elemento visual e cada funcionalidade foram cuidadosamente planeados para criar uma plataforma envolvente e educacional.

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>Arduino Code Masters</title>
6   <link rel="stylesheet" href="home.css">
7   <link rel="icon" href="/imagens/favicon-32x32.png">
8
9 </head>
10
11 <body>
12   <header>
13     <h2 class="logo"> </h2>
14     <nav class="navigation">
15       <a href="about.php">About</a>
16
17
18
19
20       <button onclick="window.location.href= './login.php';"
21         class="btnlogin-popup">Login</button>
22     </nav>
23   </header>
24   <h2 class="logo">
25     
27 </body>
28
29 </html>
```

Figura 14

Começando com uma página em branco, mergulhei em linguagens de programação fundamentais, como HTML, CSS e JavaScript. Através delas, moldei a espinha dorsal do sítio 'web', delineando a estrutura da página, os arranjos visuais e as interações dinâmicas. Como um arquiteto digital, estabeleci os alicerces, erguendo as paredes e criando os espaços interativos.



Figura 15

Movido pela estética, utilizei ferramentas como a Ferramenta de Recorte Windows e o RemoveBG para refinar e otimizar as imagens, assegurando um ambiente visualmente agradável e coeso. Foi como escolher a paleta de cores perfeita para cada canto da casa digital, mantendo a harmonia e a aparência visual.

Para criar uma identidade única, recorri ao Splendid Logo Maker para desenvolver um logótipo personalizado. Este símbolo representa o cerne e o propósito do projeto, tornando-se uma marca que reflete a sua essência. Assim como uma casa é reconhecida pela sua fachada, o logótipo destaca e diferencia o “Arduino Code Masters” no vasto mundo ‘online’.



Figura 16

Contudo, a construção não se limitou à aparência. Tal como uma casa exige uma infraestrutura funcional, utilizei tecnologias como PHP e SQL para dotar o projeto de interatividade dinâmica. Estas ferramentas permitiram estabelecer ligação com uma base de dados, possibilitando a recolha, armazenamento e apresentação de informações de forma estruturada e personalizada.

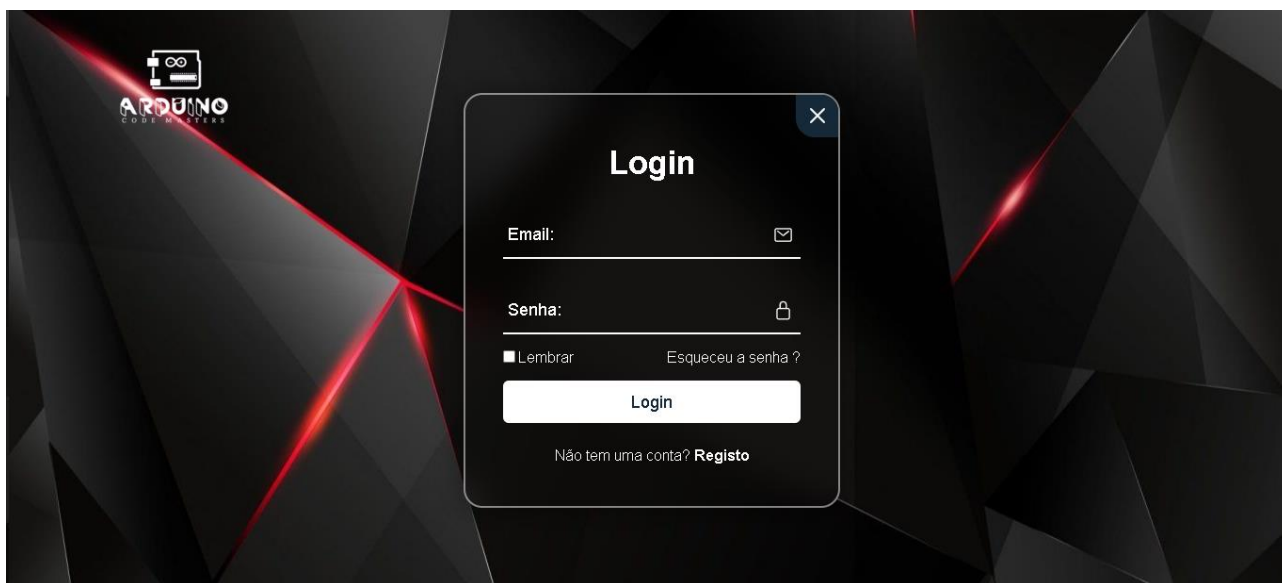


Figura 17

O resultado é um projeto que, partindo do zero, evoluiu para uma plataforma pronta para receber entusiastas do Arduino. Cada elemento, cada funcionalidade e cada linha de código contribuíram para criar uma experiência única, repleta de conhecimento, interatividade e inspiração.

Assim como a construção de uma casa exige esforço, habilidade e dedicação, a criação deste sítio 'web' desde o princípio requer competência, paciência e compromisso. O "Arduino Code Masters" é o resultado palpável desta jornada de desenvolvimento, personalização e criação digital. É um testemunho da habilidade em traduzir uma visão em realidade 'online' e um convite para que outros se aventurem na construção dos seus próprios projetos digitais.

Criação de Conteúdo para as páginas:

Página Criar conta

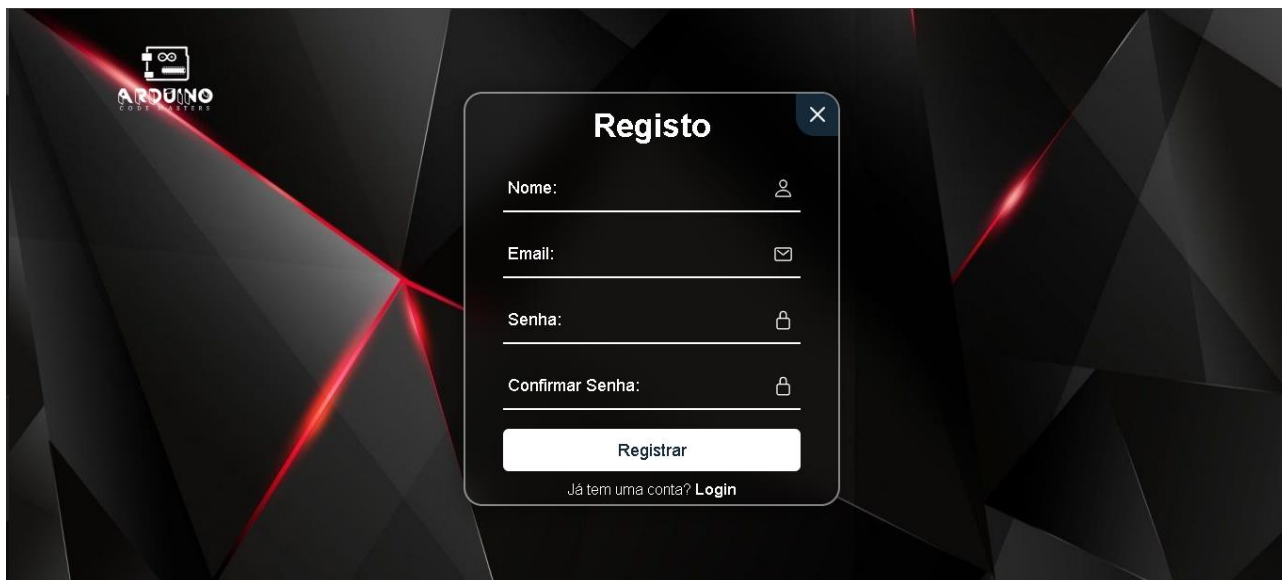


Figura 18

`<!DOCTYPE html>`: declaração do tipo de documento, que especifica que este é um documento HTML5.

A secção `<head>` contém informações sobre a página, como o título da página, hiperligações para folhas de estilo CSS, 'scripts' e outros metadados.

`<link rel="icon">`: define o ícone exibido na aba do navegador, referenciando um arquivo de imagem.

`<script>`: inclui um bloco de 'script' JavaScript que será executado quando a página for carregada. Neste caso, a função `exibirMensagem()` é definida para exibir uma mensagem de sucesso de 'login' num pop-up se houver uma mensagem passada pela URL.

A secção `<style>` contém regras de estilo CSS que são aplicadas à página. Isso inclui estilos para o fundo, cabeçalho, formulário de registo e outros elementos da página. É possível observar formatação de texto, cores, tamanhos, posicionamento e outros estilos.

*: Isso se refere a todos os elementos na página e define várias propriedades de estilo comuns, como margens, preenchimento e a fonte usada.

`<body>`: inicia o corpo da página, onde o conteúdo visível para os utilizadores é exibido.

`<header>`: cria um cabeçalho na parte superior da página que contém o logótipo e hiperligações de navegação.

`<div class="warpper">`: define um contêiner para o formulário de registo. Este contêiner tem um fundo com efeito de desfoque, uma borda arredondada e uma sombra. A classe `active-popup` permite que o contêiner seja ampliado quando necessário.

`<div class="form-box register">`: dentro do contêiner, há um formulário de registo com campos para nome, correio eletrónico, senha e confirmação de senha. Cada campo tem um ícone, rótulo e entrada de texto.

`<button type="submit">`: um botão "Registar" é fornecido para enviar o formulário de registo. O atributo `onsubmit` chama a função `validarSenha()` para garantir que a senha e a confirmação de senha coincidam.

A função `validarSenha()`: Compara a senha e a confirmação de senha. Se não coincidirem, um pop-up é exibido para alertar o utilizador.

Os 'scripts' do Ionicons são carregados para fornecer ícones (como ícones de bloqueio, envelope, etc.) na página.

Em resumo, este código cria uma página de registo com estilos de 'design' atraentes e uma função para verificar se as senhas inseridas coincidem antes de permitir o registo do utilizador. Também inclui um pop-up para mensagens de sucesso de 'login'.

PHP Criar Conta

```
<?php
// Início da sessão
session_start();
$headers = "XWE-Version: 1.0" . "\r\n";
$headers .= "Content-type: text/html; charset=UTF-8" . "\r\n";

// Conexão com o banco de dados
$servername = "localhost";
$username = "id20757658_miguelgaitas";
$password = "miguelgaitas24.";
$dbname = "id20757658_dados_dos_registros";

$conn = mysqli_connect($servername, $username, $password, $dbname);

if (!$conn) {
    die("Conexão falhou: " . mysqli_connect_error());
}

$nome = $_POST['nome'];
$email = $_POST['email'];
$senha = md5($_POST['senha']);

// Verificar se o usuário já está registrado
$sqlVerificar = "SELECT id FROM usuarios WHERE email = '$email'";
$resultadoVerificar = mysqli_query($conn, $sqlVerificar);

if (mysqli_num_rows($resultadoVerificar) > 0) {
    mysqli_close($conn);

    // Redirecionar para a página de registro com a mensagem de e-mail já registrado
    $mensagem = "Este e-mail já está registrado.";
    $redirecionarPara = "registro.php?mensagem=" . urlencode($mensagem);

    header("Location: $redirecionarPara");
    exit;
} else {
    // Inserir os dados na tabela de usuários com status "nao_verificado"
    $sqlInserir = "INSERT INTO usuarios (nome, email, senha, status) VALUES ('$nome', '$email', '$senha', 'nao_verificado')";

    if (mysqli_query($conn, $sqlInserir)) {
        // Recupere o nome do usuário do banco de dados
        $sqlBuscarNome = "SELECT nome FROM usuarios WHERE email = '$email'";
        $resultadoBuscarNome = mysqli_query($conn, $sqlBuscarNome);

        if ($resultadoBuscarNome) {
            $row = mysqli_fetch_assoc($resultadoBuscarNome);
            $nomeUsuario = $row['nome'];
        } else {
            $nomeUsuario = "Usuário"; // Defina um valor padrão se não conseguir recuperar o nome
        }
    }
}
```

Figura 19 PHP 1

```

51     mysqli_close($conn);
52
53
54     // Configuração de envio de e-mail usando o mailhog
55     ini_set("SMTP", "localhost");
56     ini_set("smtp_port", 1025);
57     ini_set("sendmail_from", "arduinocodemasters@gmail.com");
58
59     // Envie o e-mail de verificação aqui
60     $assunto = "Verificação de Conta";
61     $mensagem = "<html>
62 <head>
63 <style>
64 /* Estilos gerais do email */
65 body {
66 background-color: black;
67 background-repeat: repeat;
68 background-size: cover;
69 margin: 0;
70 padding: 0;
71 font-family: 'Poppins', sans-serif;
72 }
73
74 .container {
75 display: flex;
76 flex-wrap: wrap;
77 justify-content: center;
78 min-width: 960px;
79 min-height: 1000px;
80 background-color: rgba(0, 0, 0, 0.5);
81 }
82
83 h1 {
84 font-size: 24px;
85 color: #fff;
86 margin-bottom: 20px;
87 }
88
89 p {
90 font-size: 16px;
91 color: #fff;
92 line-height: 1.6;
93 }
94
95 /* Estilos para o botão de redefinição de senha */
96 .btn {
97 display: inline-block;
98 background-color: #0078FF;
99 color: #fff;

```

Figura 20 PHP 2

```

110
111
112 /* Estilos específicos para o link de redefinição de senha */
113 .reset-link {
114 color: #0078FF;
115 text-decoration: none;
116 font-weight: bold;
117 }
118
119 .reset-link:hover {
120 text-decoration: underline;
121 }
122
123
124 </style>
125
126 </head>
127 <body>
128 <center>
129 <p><img src='http://localhost/PAP/imagens/Capturar-removetg-preview.png' alt='Logo' class='logo-image' style='width: 160px; height: auto;'></p>
130 <p>Olá, $nomeUsuario</p>
131 <p>Clique no link abaixo para verificar a conta:</p>
132 <p>a class='reset-link' href='http://localhost/PAP/verificar_conta.php?email=$email'>Verificar Conta</a></p>
133 </center>
134 </body>
135 </html>";
136
137 mail($email, $assunto, $mensagem, $headers);
138
139 // Redirecionar para a página de login com a mensagem de registro bem-sucedido
140 $mensagem = "Registro bem-sucedido! Verifique seu e-mail para ativar sua conta.";
141 $redirecionarPara = "login.php?mensagem=" . urlencode($mensagem);
142
143 header("Location: $redirecionarPara");
144 exit;
145 } else {
146 echo "Erro ao registrar o usuário: " . mysqli_error($conn);
147 }
148 }
149 ?>
150
151

```

Figura 21 PHP 3

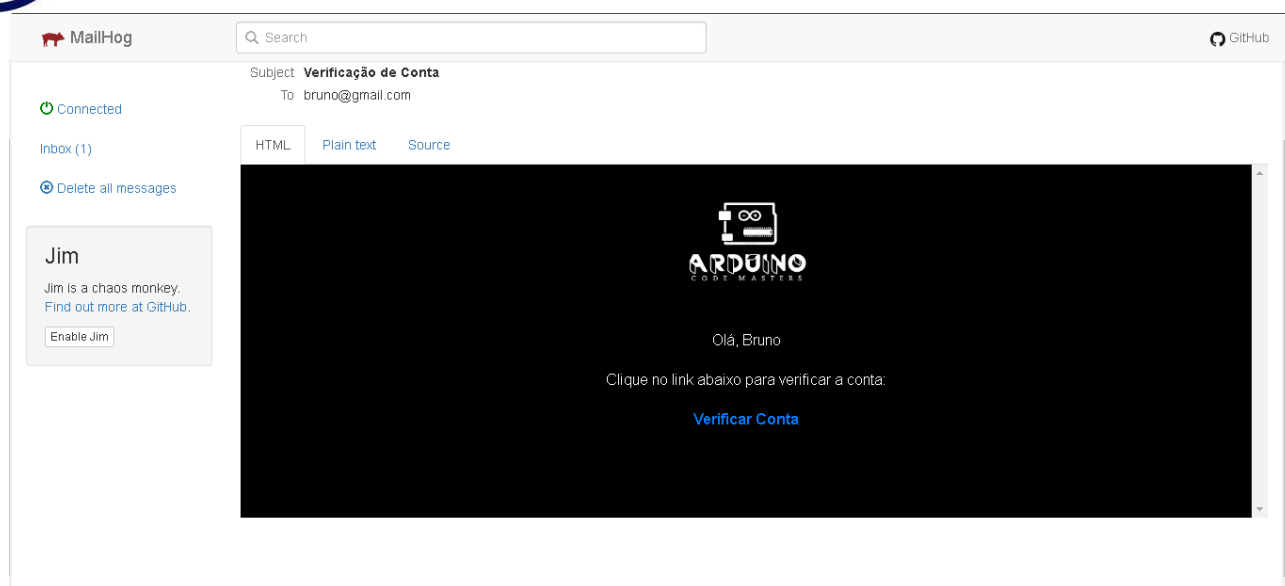


Figura 22 mailhog 1

`session_start()`: Inicia a sessão para o utilizador.

`$headers`: define cabeçalhos de correio eletrónico para garantir que a mensagem eletrónica gerada posteriormente seja exibido corretamente com UTF-8.

Conexão com o banco de dados: estabelece uma conexão com um banco de dados MySQL. São especificadas informações como nome do servidor, nome de utilizador, palavra-passe e nome do banco de dados. Se a conexão falhar, exibirá uma mensagem de erro.

`$nome`, `$email` e `$senha`: recupera os valores enviados pelo formulário de registo por meio do método POST.

Verificação de utilizador existente: Verifica se o correio eletrónico já está registado na base de dados. Se já existir, o código redireciona o utilizador de volta à página de registo com uma mensagem de erro.

Inserir dados na base de dados: se o correio eletrónico não estiver registado, os dados do utilizador são inseridos na tabela "utilizadores". A senha é encriptada com o algoritmo MD5 para segurança.

Recuperação do nome do utilizador: O código recupera o nome do utilizador da base de dados com base no correio eletrónico.

Configuração de envio de mensagem eletrónica: define as configurações de envio de correio eletrónico usando o servidor local e a porta 1025 (MailHog). Define o endereço eletrónico remetente.

Envio de mensagem eletrónica de verificação: cria uma mensagem de e-mail HTML que inclui uma hiperligação para verificar a conta. A conexão contém o e-mail do utilizador como um parâmetro na URL. O e-mail é enviado para o utilizador com uma hiperligação de verificação.

Redirecionamento para o 'login': após o registo bem-sucedido, o utilizador é redirecionado para a página de 'login' com uma mensagem de sucesso.

Este código trata do processo de registo de utilizadores, verificação de duplicatas de e-mail, inserção de dados na base de dados e envio de um e-mail de verificação. É uma parte essencial de um sistema de registo de utilizadores num sítio 'web'.

Página de login

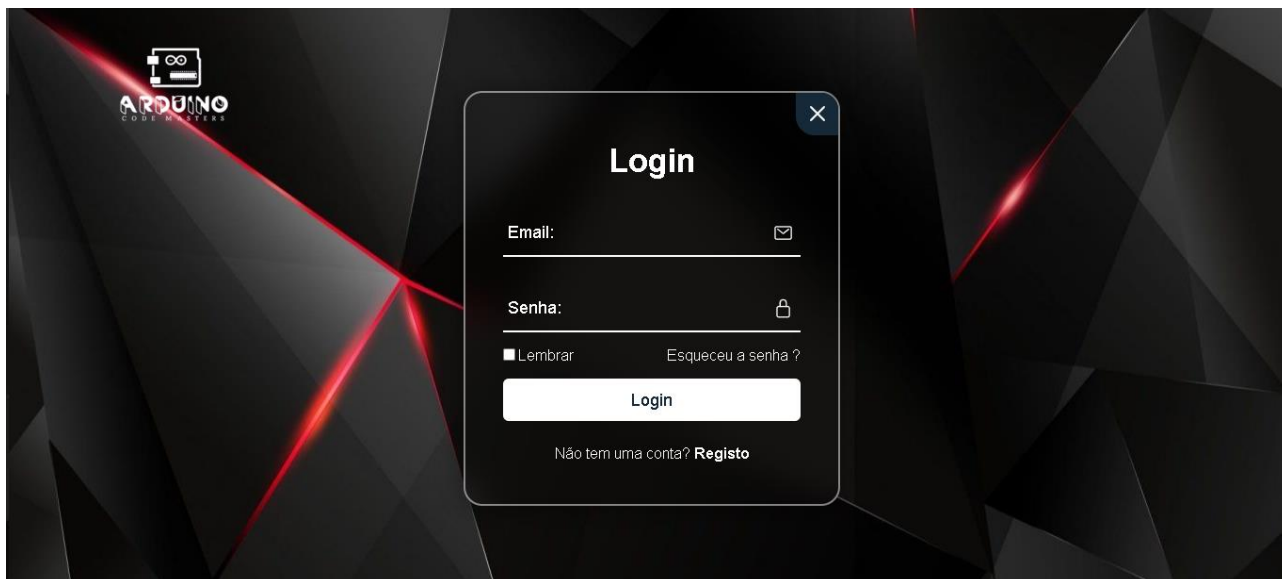


Figura 23

`<!DOCTYPE html>`: declara o tipo de documento HTML.

`<html>`: abre o elemento HTML.

`<head>`: contém informações sobre o documento, como o título da página, hiperligações para estilos CSS e 'scripts' JavaScript, bem como a definição do ícone da página.

`<link rel="icon" href="/imagens/favicon-32x32.png">`: define o ícone da página, exibido na aba do navegador.

`<script>`: contém um 'script' JavaScript que exibe uma mensagem de 'login' bem-sucedido num pop-up.

A função `exibirMensagem` verifica se há uma mensagem na URL (usando `$_GET`) e exibe essa mensagem num pop-up quando a página é carregada.

`<style>`: contém estilos CSS para personalizar a aparência da página de 'login'. Ele define o estilo do corpo da página, do cabeçalho, dos elementos de navegação, do formulário de 'login' e do pop-up. `<header>`: define o cabeçalho da página, que contém o logótipo da página.

`<nav class="navigation">`: define a barra de navegação da página, que pode ser usada para adicionar hiperligações para outras partes do sítio 'web'.

`<body>`: contém o conteúdo principal da página.

`<div class="warpper">`: define uma caixa (ou "wrapper") para o formulário de 'login'. Esta caixa é estilizada com fundo transparente, sombra e efeito de desfoque.

``: cria um botão de fechar no canto superior direito da caixa de 'login'.

`<div class="form-box login">`: contém o formulário de 'login'. Inclui campos para correio eletrónico, senha e opções de "Lembrar" e "Esqueceu a senha?".

`<form method="post" action="verifica_login.php">`: define o formulário que enviará os dados do utilizador para a página "verifica_login.php" para autenticação.

`<input>`: São campos de entrada para o correio eletrónico e a senha do utilizador.

`<label>`: rótulos para os campos de entrada. Quando os campos são preenchidos, o rótulo se move para cima, criando uma experiência de utilizador agradável.

`<input type="checkbox">`: cria uma caixa de seleção para lembrar o utilizador.

`Esqueceu a senha?`: uma hiperligação para redefinir a senha.

`<button type="submit" value="Login" class="btn">'Login'</button>`: Botão para submeter o formulário de 'login'.

`<div class="login-register">`: contém uma hiperligação para a página de registo.

`<script>`: inclui 'scripts' JavaScript que podem ser necessários para a funcionalidade da página. Ele também carrega os ícones Ionicons.

No geral, este código cria uma página de 'login' com estilos atraentes e funcionalidade para autenticação de utilizadores.

PHP Login

```

1  <?php
2  // Início da sessão
3  session_start();
4
5  // Conexão com o banco de dados
6  $servername = "localhost";
7  $username = "id20757658_miguelgaitas";
8  $password = "miguelgaitas24.";
9  $dbname = "id20757658_dados_dos_registros";
10
11 $conn = mysqli_connect($servername, $username, $password, $dbname);
12
13 if (!$conn) {
14     die("Conexão falhou: " . mysqli_connect_error());
15 }
16
17 $email = $_POST['email'];
18 $senha = md5($_POST['senha']);
19
20 // Verificar se o usuário já está registrado e se o status é "verificado"
21 $sql = "SELECT id, nome, admin, status FROM usuarios WHERE email = '$email' AND senha = '$senha'";
22 $resultado = mysqli_query($conn, $sql);
23
24 if (mysqli_num_rows($resultado) == 1) {
25     $linha = mysqli_fetch_assoc($resultado);
26
27     if ($linha['status'] === 'verificado') {
28         // Armazenamento do ID e do nome do usuário na sessão
29         $_SESSION['id_usuario'] = $linha['id'];
30         $_SESSION['nome_usuario'] = $linha['nome'];
31
32         if ($linha['admin'] == 1) {
33             $_SESSION['isAdmin'] = true;
34             $mensagem = "Login bem-sucedido! Esta conta é de admin";
35             $redirecionarPara = "pagina_de_admin.php";
36         } else {
37             $_SESSION['isAdmin'] = false;
38             $redirecionarPara = "primeira_pagina.php";
39             $mensagem = "Login bem-sucedido!";
40         }
41
42         mysqli_close($conn);
43
44         // Redirecionar para a página de destino com a mensagem na URL
45         header("Location: $redirecionarPara?mensagem=" . urlencode($mensagem));
46         exit;
47     } else {
48         // Conta não verificada, redirecionar para a página de login com mensagem de erro
49         mysqli_close($conn);

```

Figura 24 PHP 4

```

50     $mensagem = "Sua conta ainda não foi verificada. Verifique seu e-mail para ativar sua conta.";
51     $redirecionarPara = "login.php?mensagem=" . urlencode($mensagem);
52     header("Location: $redirecionarPara");
53     exit;
54 }
55 } else {
56     // E-mail ou senha incorretos
57     echo "E-mail ou senha estão incorretos." . mysqli_error($conn);
58 }
59
60 mysqli_close($conn);
61 ?>

```

Figura 25 PHP 5

1. Início da Sessão:

A sessão é iniciada com a função `session_start()`. Isso permite que dados da sessão, como o 'ID' do utilizador, sejam armazenados e acessados ao longo da visita do utilizador ao sítio 'web'.

2. Conexão com o Banco de Dados:

As credenciais do servidor de banco de dados, como nome de servidor, nome de utilizador e palavra-passe, são definidas. O código utiliza a extensão MySQL para estabelecer uma conexão com o banco de dados.

3. Verificação de Conexão com o Banco de Dados:

O código verifica se a conexão com o banco de dados foi estabelecida com sucesso. Se a conexão falhar, uma mensagem de erro é exibida e o programa é encerrado.

4. Obtenção de Dados do Formulário:

O código obtém o correio eletrónico e a senha submetidos por meio de um formulário HTML utilizando o método POST.

5. Consulta ao Banco de Dados:

Uma consulta SQL é executada para verificar se existe um utilizador com o correio eletrónico e a senha fornecidos. A consulta verifica também o estado da conta do utilizador.

6. Autenticação e Redirecionamento:

Se um único utilizador é encontrado (exatamente uma linha na resposta da consulta), o código verifica o estado da conta desse utilizador.

Se a conta estiver "verificada", o utilizador é autenticado. O seu 'ID' e nome são armazenados na sessão.

Se o utilizador for um administrador (campo "admin" igual a 1), a variável `$_SESSION['isAdmin']` é definida como verdadeira, e o utilizador é redirecionado para a página de administração.

Se o utilizador não for um administrador, ele é redirecionado para a página de início do utilizador com uma mensagem de 'login' bem-sucedido.

Se a conta não estiver verificada, o utilizador é redirecionado para a página de 'login' com uma mensagem informando-o de que a sua conta ainda não foi verificada. O utilizador deve verificar o seu endereço eletrónico para ativar a conta.

7. Mensagens de Erro:

Se a consulta SQL não retornar resultados (ou seja, correio eletrónico ou senha incorretos), uma mensagem de erro é exibida.

8. Fechamento da Conexão com o Banco de Dados:

Após a autenticação e redirecionamento, ou em caso de erro, a conexão com o banco de dados é encerrada.

Página sobre o Arduíno em si



Figura 26

O texto descreve o Arduino como uma plataforma aberta e versátil para a criação de projetos eletrônicos interativos. Ele destaca que o Arduino é utilizado em áreas como eletrônica, engenharia, 'design' e arte. A acessibilidade e o baixo custo tornam-no atrativo. O texto menciona a ampla gama de projetos que podem ser realizados com o Arduino, desde coisas simples, como acender uma luz, até projetos mais complexos, como a construção de robôs ou sistemas de automação residencial.

Além disso, a comunidade ativa de utilizadores do Arduino é referida. Essa comunidade partilha projetos, códigos e recursos 'online' para ajudar os outros a aprender e a desenvolver habilidades. O texto encoraja os interessados em eletrônica e programação a considerarem o Arduino como uma excelente opção para iniciar nesse mundo. Ele ressalta a capacidade de criar projetos personalizados e explorar novas ideias com esta plataforma.

Em suma, o texto oferece uma visão abrangente do Arduino, desde a sua utilização em diferentes áreas até à sua comunidade de utilizadores, e destaca como é uma escolha recomendada para quem deseja começar a explorar eletrônica e programação de forma prática e criativa.

Página dos projetos

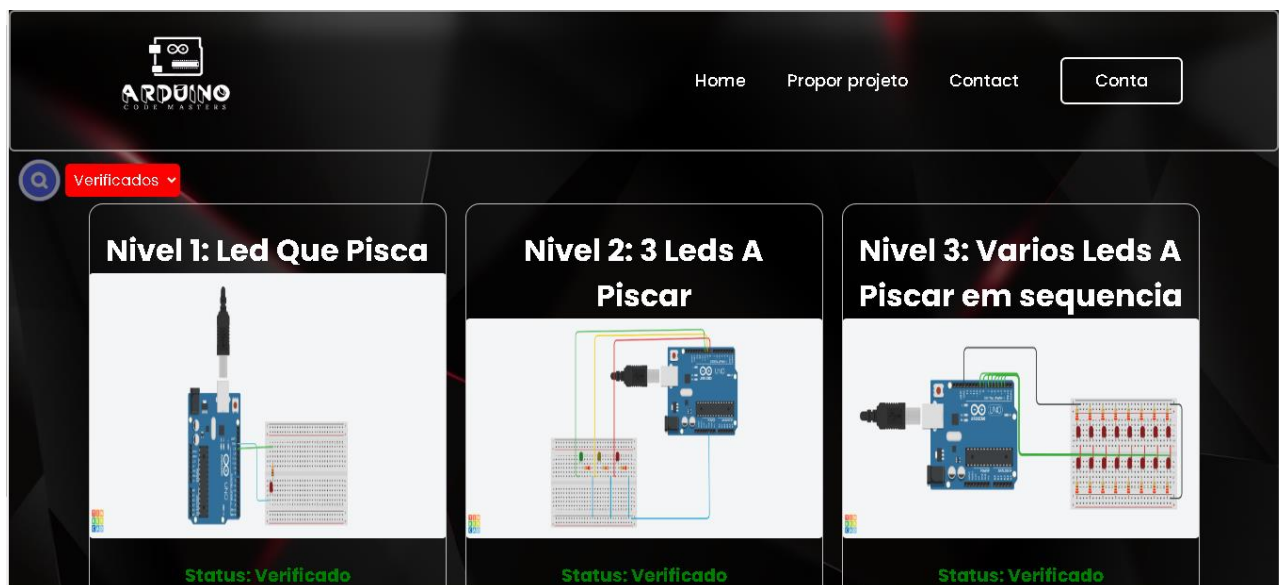


Figura 27

Cabeçalho:

O cabeçalho da página inclui um logótipo e hiperligações de navegação, como "Início," "Propor Projeto," "Contacto," e "Conta." Também existe um botão "Conta" que redireciona o utilizador para a página de detalhes da conta.

Lista de Projetos:

No centro da página, encontra uma lista de projetos Arduino. Cada projeto na lista inclui um nome, descrição, estado (como "verificado," "rejeitado," ou "pendente"), uma imagem, e o nome do criador do projeto. Os projetos são apresentados em caixas individuais com uma borda e um fundo ligeiramente transparente.

Botão "Voltar ao Topo":

À medida que rola a página para baixo, um botão "Voltar ao Topo" aparece no canto inferior direito. Se clicar nesse botão, a página rolará de volta para o topo.

‘Design’ Visual:

O ‘design’ visual da página inclui um fundo com uma imagem de tecnologia, e os projetos são exibidos em caixas transparentes para melhorar a legibilidade. A página utiliza cores e fontes específicas para criar uma experiência visual agradável.

Função do Botão "Conta":

Ao clicar no botão "Conta," o utilizador é redirecionado para a página de detalhes da conta, onde pode gerir informações de conta, se estiver autenticado.

Em resumo, este código PHP faz parte de uma página ‘web’ que exhibe projetos Arduino num layout organizado, permite que os utilizadores naveguem pelo sítio ‘web’ e tenham a opção de voltar ao topo da página quando necessário. O conteúdo dos projetos é recuperado de uma base de dados MySQL e exibido dinamicamente na página.

PHP Projetos

```

1  <?php
2  session_start();
3
4  $servername = "localhost";
5  $username = "id20757658_miguelgaitas";
6  $password = "MiguelGaitas24.";
7  $dbname = "id20757658_dados_dos_registros";
8
9  $conn = new mysqli($servername, $username, $password, $dbname);
10 if ($conn->connect_error) {
11     die("Falha na conexão com o banco de dados: " . $conn->connect_error);
12 }
13
14 function getProjetos($conn)
15 {
16     $meuID = $_SESSION['id_usuario'];
17     $search = ''; // Inicialize a variável de pesquisa
18     $filtro = ''; // Inicialize o filtro
19
20     if (isset($_POST['search'])) {
21         $search = mysqli_real_escape_string($conn, $_POST['search']);
22     }
23
24     if (isset($_POST['filtro'])) {
25         $filtro = $_POST['filtro'];
26     }
27
28     $sql = "SELECT p.id, p.nome, p.descricao, p.status, p.imagem, p.codigo, u.nome AS nome_utilizador
29 FROM projetos_arduino p
30 LEFT JOIN usuarios u ON p.autor = u.id
31 WHERE ";
32
33     if ($filtro === 'verificado') {
34         $sql .= "p.status = 'verificado'";
35     } elseif ($filtro === 'pendente') {
36         $sql .= "p.status != 'verificado' AND (p.status != 'rejeitado' OR p.prazo_edicao >= NOW()) AND p.autor = $meuID";
37     } elseif ($filtro === 'expirado') {
38         $sql .= "p.prazo_edicao < NOW()"; // Ajuste para a data de expiração
39     } else {
40         $sql .= "p.status = 'verificado'";
41     }
42
43     $sql .= " AND (p.nome LIKE '%$search%' OR p.descricao LIKE '%$search%')
44 ORDER BY p.id";
45
46     $result = mysqli_query($conn, $sql);
47
48     if (!$result || mysqli_num_rows($result) == 0) {

```

Figura 28 PHP 6

```

48  if (!$result || mysqli_num_rows($result) == 0) {
49      echo "Nenhum projeto encontrado.";
50      return;
51  }
52
53  while ($row = mysqli_fetch_assoc($result)) {
54      $id = $row["id"];
55      $nome = $row["nome"];
56      $desc = $row["descricao"];
57      $status = $row["status"];
58      $img = $row["imagem"];
59      $cod = $row["codigo"];
60      $nomeutilizador = $row["nome_utilizador"];
61
62      echo "<div class='projeto'>
63          <h1>$nome</h1>
64          <img src='./imagens/$img'>
65          <br>
66          <h3><p class=$status>status: " . ucfirst($status) . "</p></h3>
67          <p>Criado por: $nomeutilizador</p>
68          <br>
69          <center>
70              <a class='texto' href='projeto.php?id=$id'>Ver Mais</a>
71          </center>
72          </div>";
73  }
74  }
75  ?>

```

Figura 29 PHP 7

session_start(); esta linha inicia ou retoma uma sessão PHP. Isso é frequentemente usado para rastrear informações do utilizador entre diferentes páginas da 'web'b'b'.

Em seguida, você configura informações de conexão com o banco de dados MySQL:

\$servername: O nome do servidor MySQL (neste caso, "localhost" se refere ao servidor local).

\$username: O nome de utilizador do MySQL.

\$password: A senha do MySQL.

\$dbname: O nome do banco de dados que será usado.

\$conn = new mysqli(\$servername, \$username, \$password, \$dbname); esta linha cria uma conexão MySQL utilizando a classe mysqli do PHP. Se a conexão falhar, um erro é gerado, indicando a falha na conexão com o banco de dados.

A função getProjetos(\$conn) é definida, e ela recebe a conexão com o banco de dados como um parâmetro. Essa função é responsável por buscar projetos no banco de dados, com base em certos critérios.

`$meulD = $_SESSION['id_usuario'];` aqui, o código obtém o valor da variável de sessão chamada "id_usuario" e atribui à variável `$meulD`. Isso provavelmente é usado para identificar o utilizador que está logado.

O código verifica se há dados enviados por um formulário HTML (provavelmente via um método POST) para realizar a pesquisa e filtragem de projetos. Os valores de pesquisa e filtro são capturados nas variáveis `$search` e `$filtro`, respetivamente.

A consulta SQL é construída com base nos critérios de filtro e pesquisa. A consulta faz uma junção (LEFT JOIN) entre duas tabelas ("projetos_arduino" e "usuarios") e seleciona projetos que atendam aos critérios especificados no filtro e contenham os termos de pesquisa no nome ou descrição do projeto.

A consulta é executada usando `mysqli_query`, e os resultados são armazenados na variável `$result`.

O código verifica se houve resultados ou se a consulta retornou vazia. Se não houver projetos correspondentes, ele exibe "Nenhum projeto encontrado."

Caso haja projetos correspondentes, o código itera sobre os resultados usando um loop while. Para cada projeto encontrado, ele exibe informações como nome, imagem, status, autor e uma hiperligação para ver mais detalhes do projeto.

O HTML é gerado dinamicamente dentro do loop while, e cada projeto é exibido num formato específico.

Resumindo, este código PHP lida com a busca e exibição de projetos a partir de um banco de dados MySQL, permitindo ao utilizador filtrar e pesquisar projetos com base em critérios específicos. Os resultados são exibidos numa página da 'web', e os detalhes do projeto são acessíveis por meio de hiperligações gerados dinamicamente

Propor Projetos

Novo Projeto ✕

Deve conter um nome específico do projeto

Nome do Projeto:

Deve conter uma descrição passo a passo de como se fez o projeto ou seja o código

Descrição:

Deve conter uma imagem do projeto

Imagem:

 Nenhum arquivo escolhido

Deve conter o código do projeto do Arduino

Figura 30

Código:

Tem de ser o código da simulação embutido do Tinkercad que se pode arranjar quando o projeto é público, em seguida vá em partilhar e copie o código <->

Simulação:

Selecione os produtos utilizados no projeto

Placas Arduino

Sensores

Atuadores

Módulos de Comunicação

Outros Componentes

Figura 31

[Criar Projeto](#)

Projetos Pessoais

[Pesquisar](#)


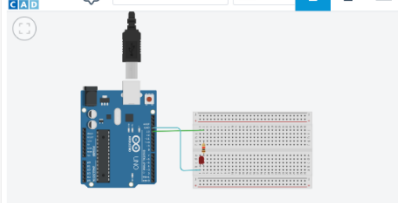
Nome do Projeto	Imagem	Código	Simulação	Descrição	Tempo Restante	Ações
Nível 1: Led Que Pisca		<pre>void setup() { pinMode(13, OUTPUT); } void loop() { digitalWrite(13, HIGH); delay(1000); // Wait for 1000 millisecond(s) digitalWrite(13, LOW); delay(1000); // Wait for 1000 millisecond(s) }</pre>		Mostrar/Esconder Descrição	Projeto em estado indefinido	Editar

Figura 32

Configurações Iniciais:

O código PHP começa por definir as configurações iniciais do banco de dados, como o nome do servidor ("localhost"), nome de utilizador ("id20757658_miguelgaitas"), palavra-passe ("MiguelGaitas24."), e nome da base de dados ("id20757658_dados_dos_registros").

Em seguida, o código tenta estabelecer uma conexão com o banco de dados utilizando o objeto mysqli. Se a conexão falhar, ele apresenta uma mensagem de erro.

Consulta de Projetos Existentes:

O código SQL é definido como "SELECT * FROM projetos_arduino", que tem o objetivo de selecionar todos os projetos na tabela "projetos_arduino".

Esta consulta é executada na base de dados através da conexão estabelecida, e os resultados são armazenados na variável "\$result".

Definição de Estilos CSS:

O código HTML inclui uma secção de estilos CSS que define a aparência da página. Esta secção de estilos controla as formatações, tamanhos de letra, cores de fundo e elementos HTML.

Formulário de Adição de Projeto Arduino:

O código HTML contém um formulário para adicionar um projeto Arduino. O formulário inclui campos para o nome do projeto, descrição, imagem, código e simulação.

O atributo "enctype" no formulário é definido como "multipart/form-data" para permitir o envio de ficheiros, como imagens.

Os campos do formulário são marcados como "required," o que significa que são obrigatórios para submissão.

Apresentação de Projetos Arduino Existentes:

O código HTML inclui uma tabela que será usada para apresentar os projetos Arduino existentes.

A tabela tem colunas para "Nome," "Descrição," "Imagem," "Código," e "Simulação."

O código PHP verifica se existem projetos Arduino na base de dados e, se houver, itera através deles e cria uma linha na tabela para cada projeto, preenchendo as colunas com os respetivos dados.

O nome do ficheiro da imagem é apresentado na coluna "Imagem."

Cada projeto na tabela inclui hiperligações para editar ou excluir o projeto.

Fecho da Conexão com o Banco de Dados:

Após a listagem dos projetos, o código fecha a conexão com o banco de dados usando "\$conn->close()". Isso é uma boa prática para liberar recursos após o uso.

Em resumo, este código PHP/HTML cria uma página onde os utilizadores podem adicionar novos projetos Arduino e visualizar os projetos Arduino existentes. Os projetos são apresentados numa tabela, e os utilizadores têm a opção de editar ou excluir cada projeto.

PHP Propor Projetos

```
1 <?php
2 session_start();
3 // Configurações do banco de dados
4 $servername = "localhost";
5 $username = "id28757688_miguelgaitas";
6 $password = "MiguelGaitas24.";
7 $dbname = "id28757688_dados_dos_registros";
8
9 // Conecta ao banco de dados
10 $conn = new mysqli($servername, $username, $password, $dbname);
11 if ($conn->connect_error) {
12     die("Falha na conexão com o banco de dados: " . $conn->connect_error);
13 }
14
15 // Obtém os dados do formulário
16 $nome = $_POST['nome'];
17 $descricao = $_POST['descricao'];
18 $imagem = $_FILES['imagem']['name'];
19 $codigo = $_POST['codigo'];
20 $sim = $_POST['simula'];
21 // Move a imagem para um diretório do servidor
22 $targetDir = "./imagens/";
23 $targetFile = $targetDir . basename($_FILES['imagem']['name']);
24 if (move_uploaded_file($_FILES['imagem']['tmp_name'], $targetFile)) {
25     echo "Imagem enviada com sucesso!<br>";
26 } else {
27     echo "Erro ao enviar a imagem.<br>";
28 }
29
30 $meuid = $_SESSION['id_usuario'];
31 $sql = "INSERT INTO projetos_arduino (autor, nome, descricao, imagem, codigo, simula) VALUES ('$meuid', '$nome', '$descricao', '$imagem', '$codigo', '$sim')";
32
33 if ($conn->query($sql) === TRUE) {
34     echo "Projeto adicionado com sucesso!";
35 } else {
36     echo "Erro ao adicionar o projeto: " . $conn->error;
37 }
38
39 // Fecha a conexão com o banco de dados
40 $conn->close();
41 ?>
42 <a href="pagina1.php" style="text-decoration: none;">
43     <button type="button" style="padding: 10px 20px; font-size: 16px;">Voltar</button>
```

Figura 33 PHP 8

Este código PHP trata do envio de dados de um formulário para um banco de dados e, em seguida, redireciona o utilizador para outra página.

`session_start()`; esta linha inicia ou retoma uma sessão PHP. Isso é frequentemente usado para rastrear informações do utilizador entre diferentes páginas da 'web'.

Em seguida, são definidas as configurações de conexão com o banco de dados MySQL:

\$servername: O nome do servidor MySQL (neste caso, "localhost" se refere ao servidor local).

\$username: O nome de utilizador do MySQL.

\$password: A senha do MySQL.

\$dbname: O nome do banco de dados que será usado. O código estabelece uma conexão com o banco de dados usando a classe mysqli do PHP. Se a conexão falhar, ele exibirá uma mensagem de erro.

Os dados do formulário são obtidos através das variáveis \$_POST e \$_FILES:

\$nome, \$descricao, \$codigo, \$simula: estas variáveis recebem os valores enviados através do formulário. \$imagem: esta variável recebe o nome do arquivo de imagem enviado pelo formulário.

O código move o arquivo de imagem para um diretório no servidor usando a função move_uploaded_file. Se a imagem for movida com sucesso, ele exibe "Imagem enviada com sucesso!", caso contrário, exibe "Erro ao enviar a imagem."

O código obtém o 'ID' do utilizador da sessão (presumivelmente o 'ID' do utilizador que está logado). A consulta SQL é construída para inserir os dados do projeto no banco de dados. Os dados do projeto, incluindo o nome, descrição, imagem, código e simulação, são inseridos na tabela "projetos_arduino".

O código executa a consulta SQL usando \$conn->query(). Se a inserção for bem-sucedida, ele exibirá "Projeto adicionado com sucesso!". Caso contrário, exibirá "Erro ao adicionar o projeto" e uma mensagem de erro específica.

Após o processo de inserção no banco de dados, a conexão com o banco de dados é fechada usando \$conn->close().

Finalmente, é exibido um botão "Voltar" que redireciona o utilizador para a página "pagina1.php" quando clicado.

Resumindo, este código PHP lida com o envio de informações de um formulário para um banco de dados MySQL, armazena os dados do projeto, incluindo uma imagem, no banco de dados e, em seguida, redireciona o utilizador para outra página após a conclusão do processo.

PHP tabela onde se exhibe os projetos do utilizador

```
112 <?php
113 session_start();
114
115 // Verifique se o usuário está autenticado (por exemplo, usando algum outro mecanismo de autenticação)
116 if (!isset($_SESSION['id_usuario'])) {
117     die("Você não está autenticado.");
118 }
119
120 $usuarioLogadoId = $_SESSION['id_usuario'];
121
122 $servername = "localhost";
123 $username = "id20757658_miguelgaitas";
124 $password = "miguelgaitas24.";
125 $dbname = "id20757658_dados_dos_registros";
126
127 $conn = new mysqli($servername, $username, $password, $dbname);
128 if ($conn->connect_error) {
129     die("Falha na conexão com o banco de dados: " . $conn->connect_error);
130 }
131
132 $sql = "SELECT * FROM projetos_arduino WHERE autor = $usuarioLogadoId";
133 $result = $conn->query($sql);
134
135 echo "<h1>Seus Projetos Arduino</h1>";
136 echo "<table>";
137 echo "<tr>";
138 echo "<th>Nome do Projeto</th>";
139 echo "<th>Descrição</th>";
140 echo "<th>Imagem</th>";
141 echo "<th>Código</th>";
142 echo "<th>Simulação</th>";
143 echo "<th>Tempo Restante</th>"; // Coluna para exibir o tempo restante
144 echo "<th>Ações</th>"; // Coluna para exibir o botão de edição
145 echo "</tr>";
146
147 while ($row = $result->fetch_assoc()) {
148     echo "<tr>";
149     echo "<td>" . $row['nome'] . "</td>";
150     echo "<td>" . $row['descricao'] . "</td>";
151     echo "<td class='img-cell'><img src='./imagens/" . $row['imagem'] . "' alt='Imagem do Projeto'></td>";
152     echo "<td>" . $row['codigo'] . "</td>";
153     echo "<td>" . $row['simula'] . "</td>";
```

Figura 34 PHP 9

```

154 echo "<td>" . $row['codigo'] . "</td>";
155 echo "<td>" . $row['simula'] . "</td>";
156
157 if ($row['status'] == 'verificado') {
158     $tempoRestante = "Projeto verificado, sem edição permitida";
159     $acoes = "";
160 } elseif ($row['status'] == 'rejeitado') {
161     $prazoEdicao = strtotime($row['prazo_edicao']);
162     $tempoRestanteSegundos = $prazoEdicao - time(); // Diferença em segundos
163
164     if ($tempoRestanteSegundos > 0) {
165         $dias = floor($tempoRestanteSegundos / (60 * 60 * 24));
166         $horas = floor(($tempoRestanteSegundos - ($dias * 60 * 60 * 24)) / (60 * 60));
167         $minutos = floor(($tempoRestanteSegundos - ($dias * 60 * 60 * 24) - ($horas * 60 * 60)) / 60);
168         $segundos = $tempoRestanteSegundos % 60;
169
170         $tempoRestante = "Projeto Rejeitado tem agora : $dias dias, $horas horas, $minutos minutos e $segundos segundos para editar o projeto se nao ele ficara expirado e nao podera ma
171         $acoes = "<a href='editar_projeto_comunitario.php?id=' . $row['id'] . '>Editar</a>";
172     } else {
173         $tempoRestante = "Tempo para edição esgotado";
174         $acoes = "";
175     }
176 } else {
177     $tempoRestante = "Projeto em estado indefinido";
178     $acoes = "";
179 }
180
181 echo "<td>$tempoRestante</td>";
182 echo "<td>$acoes</td>";
183 echo "</tr>";
184 }
185
186 echo "</table>";
187 >>

```

Figura 35 PHP 10

Este código PHP trata da exibição de projetos Arduino pertencentes a um utilizador específico e oferece informações sobre cada projeto, incluindo ações possíveis com base no status do projeto.

`session_start()`; esta linha inicia ou retoma uma sessão PHP.

Isso é frequentemente usado para rastrear informações do utilizador entre diferentes páginas da 'web'.

O código verifica se o utilizador está autenticado. Se o 'ID' do utilizador não estiver presente na sessão, ele exibirá a mensagem "Você não está autenticado" e encerrará a execução do 'script'.

Em seguida, o código define as configurações de conexão com o banco de dados MySQL:

`$servername`: O nome do servidor MySQL (neste caso, "localhost" se refere ao servidor local).

`$username`: O nome de utilizador do MySQL.

`$password`: A senha do MySQL.

`$dbname`: O nome do banco de dados que será usado.

O código estabelece uma conexão com o banco de dados usando a classe mysqli do PHP. Se a conexão falhar, ele exibirá uma mensagem de erro.

Ele constrói uma consulta SQL para selecionar projetos Arduino pertencentes ao utilizador atual. A consulta seleciona todos os campos da tabela "projetos_arduino" onde o campo "autor" corresponde ao 'ID' do utilizador atual.

O código executa a consulta SQL usando `$conn->query()` e armazena os resultados na variável `$result`.

É exibido um cabeçalho com o título "os seus Projetos Arduino" e é criada uma tabela HTML para exibir os detalhes dos projetos.

As colunas da tabela são definidas, incluindo Nome do Projeto, Descrição, Imagem, Código, Simulação, Tempo Restante e Ações.

O código entra num 'loop' while para percorrer cada projeto recuperado da consulta. Para cada projeto, ele exibe os detalhes nas células da tabela, incluindo o nome, descrição, imagem, código e simulação.

A coluna "Tempo Restante" é preenchida com base no status do projeto. Se o projeto estiver "verificado", ele informa que não é permitida a edição. Se o projeto estiver "rejeitado", ele calcula o tempo restante para edição, se houver algum. Se o projeto estiver em qualquer outro estado, ele informa que o projeto está em estado indefinido.

A coluna "Ações" exibe uma hiperligação de edição para projetos "rejeitados" com tempo restante ou uma célula vazia para projetos "verificados" ou em estados indefinidos.

O código fecha a tabela HTML e, portanto, conclui a exibição dos projetos.

Em resumo, este código PHP exibe os projetos Arduino pertencentes a um utilizador autenticado, fornecendo informações detalhadas sobre cada projeto, incluindo status e ações possíveis com base no status do projeto. Isso permite ao utilizador gerir os seus projetos na plataforma.

Página dos top 5 componentes mais utilizados no sítio 'web'



Figura 36

Este código PHP e HTML constitui uma página web dinâmica que se conecta a uma base de dados MySQL para recuperar informações sobre os cinco componentes mais utilizados. A página apresenta um layout atraente, utilizando HTML, CSS e JavaScript para criar uma experiência visualmente agradável para o utilizador.

1. Conexão à Base de Dados:

- O código PHP no início estabelece uma conexão com a base de dados MySQL usando variáveis de conexão, como host, nome de utilizador, senha e nome da base de dados.

2. Estilo Visual:

- O CSS é utilizado para estilizar a página, criando um fundo com uma imagem de paisagem, tornando a página mais atrativa. O uso de efeitos de desfoque (blur) e transparência adiciona uma camada estilizada aos elementos.

3. Layout Responsivo:

- A página é projetada para ser responsiva, ajustando-se à largura da tela do dispositivo do utilizador. O uso de flexbox e percentagens permite que os elementos se ajustem de maneira adequada.

4. Barra de Navegação:

- A barra de navegação no topo contém links para diferentes secções da página, como a página inicial, projetos, lojas, fórum e contacto. Há também um botão "Conta" que redireciona para a página de detalhes da conta.

5. Funcionalidade de Pesquisa:

- Um campo de pesquisa permite que os utilizadores pesquisem componentes na base de dados. O resultado da pesquisa é exibido dinamicamente sem recarregar a página, usando AJAX e jQuery.

6. Tabela de Componentes:

- Abaixo da barra de navegação, a página exibe uma tabela contendo os cinco componentes mais utilizados, recuperados da base de dados. As colunas incluem o nome do componente e a quantidade utilizada.

7. Botão "Voltar ao Topo":

- Um botão "Voltar ao Topo" é exibido no canto inferior direito da página quando o utilizador rola para baixo. Esse botão é implementado com JavaScript para rolar suavemente de volta ao topo da página quando clicado.

Em resumo, a página combina aspetos de design visualmente atraentes com funcionalidades dinâmicas, como a exibição de dados da base de dados e a capacidade de pesquisa, para proporcionar uma experiência interativa ao utilizador.

PHP da página dos 5 componentes

```
469 <?php
470 // Substitua as configurações do banco de dados conforme necessário
471 $servername = "localhost";
472 $username = "id20757658_miguelgaitas";
473 $password = "MiguelGaitas24.";
474 $dbname = "id20757658_dados_dos_registros";
475
476 // Conectar ao banco de dados
477 $conn = new mysqli($servername, $username, $password, $dbname);
478
479 // Verificar a conexão
480 if ($conn->connect_error) {
481     die("Falha na conexão com o banco de dados: " . $conn->connect_error);
482 }
483
484 // Consulta para obter os top 5 componentes mais utilizados
485 $sql = "SELECT * FROM componentes ORDER BY quantidade_utilizada DESC LIMIT 5";
486 $result = $conn->query($sql);
487
488 // Exibir os resultados em uma tabela HTML
489 if ($result->num_rows > 0) {
490     echo "<table>";
491     echo "<tr><th>Nome</th><th>Quantidade Utilizada</th></tr>";
492     while ($row = $result->fetch_assoc()) {
493         echo "<tr><td>" . $row["nome"] . "</td><td>" . $row["quantidade_utilizada"] . "</td></tr>";
494     }
495     echo "</table>";
496 } else {
497     echo "Não foram encontrados componentes.";
498 }
499
500 // Fechar a conexão com o banco de dados
501 $conn->close();
502 ?>
```

Figura 37 PHP 11

1. Conexão à Base de Dados:

- O código conecta-se a uma base de dados MySQL usando informações como o servidor, nome de utilizador, senha e nome da base de dados.
- Se a ligação não for bem-sucedida, uma mensagem de erro é apresentada e o script é interrompido.

2. Consulta SQL e Apresentação em Tabela HTML:

- Realiza uma consulta SQL para obter os cinco componentes mais utilizados da tabela "componentes".
- Os resultados são apresentados numa tabela HTML, mostrando os nomes e quantidades utilizadas.
- Se não houver resultados, é exibida uma mensagem indicando a ausência de componentes.

3. Encerramento da Ligação à Base de Dados:

- Após a apresentação dos resultados, a ligação à base de dados é encerrada.

Em resumo, o código realiza uma operação de leitura na base de dados para obter informações sobre os componentes mais utilizados, exibe esses dados numa tabela HTML e, por fim, fecha a ligação à base de dados.

Página das lojas



Figura 38

Cabeçalho da Página:

O cabeçalho da página inclui um logótipo e ligações de navegação, como "Início", "Projetos", "Produtos", "Fórum" e "Contacto".

Existe também um botão "Conta" que redireciona o utilizador para a página de detalhes da conta.

Tabela de Lojas e Produtos:

Uma tabela HTML é usada para apresentar informações sobre as lojas e os produtos disponíveis em cada loja.

A tabela tem as seguintes colunas: "Nome da Loja", "Morada", "Cidade", "Código Postal", "Telefone" e "Produtos Disponíveis".

Os dados são preenchidos dinamicamente a partir da base de dados.

Conexão ao Banco de Dados:

O código PHP estabelece uma conexão com a base de dados MySQL utilizando informações de hospedagem, nome de utilizador, senha e nome da base de dados.

Recuperação de Dados das Lojas e Produtos:

A página realiza uma consulta SQL para obter informações sobre as lojas e os produtos disponíveis.

Primeiro, as informações sobre as lojas são recuperadas da tabela "lojas".

Depois, a página efetua uma consulta para obter a lista de produtos disponíveis em cada loja.

Exibição dos Dados:

É utilizado um ciclo para percorrer as lojas recuperadas da base de dados.

As informações sobre cada loja são exibidas numa linha da tabela.

Para cada loja, a página recupera a lista de produtos disponíveis naquela loja e apresenta-os na coluna "Produtos Disponíveis".

Botão "Voltar ao Topo":

Existe um botão "Voltar ao topo" que surge no canto inferior direito da página quando o utilizador faz scroll para baixo.

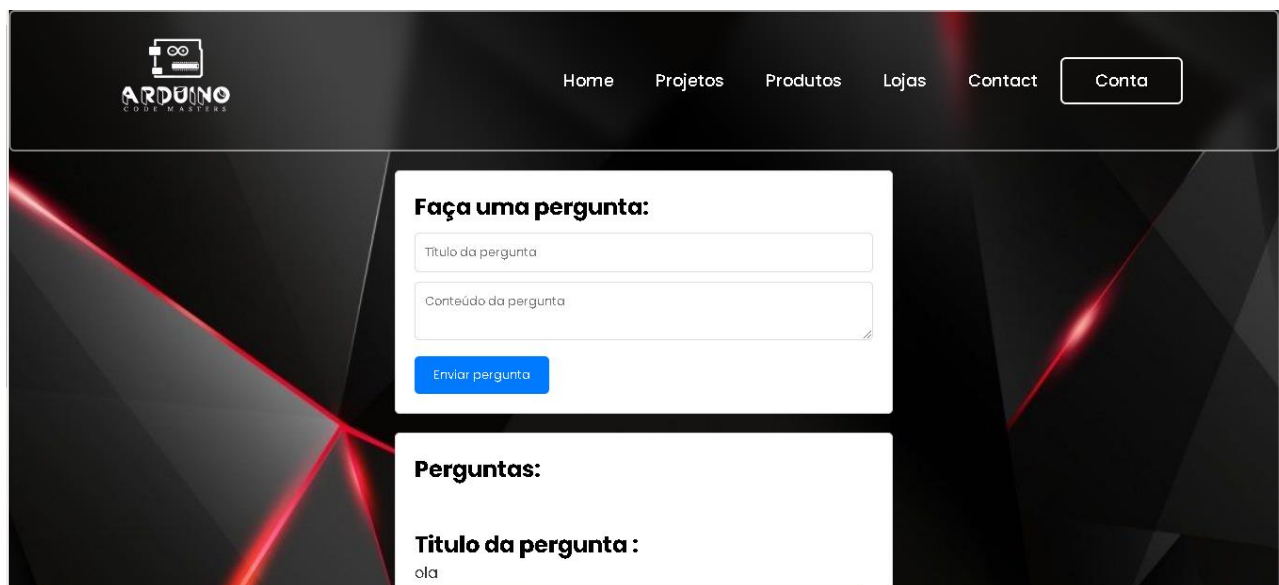
Se o utilizador clicar neste botão, a página fará scroll de volta para o topo.

Estilo Visual:

A página tem um estilo visual apelativo, com um fundo de imagem, estilos de fonte e cores específicas para melhorar a experiência do utilizador.

Em resumo, esta página é utilizada para mostrar informações sobre as lojas e os produtos disponíveis em cada uma. Os dados são recuperados da base de dados e exibidos numa tabela HTML. Além disso, a página oferece funcionalidades de navegação e um botão "Voltar ao Topo" para melhorar a experiência do utilizador

Forum de perguntas e respostas



The screenshot shows a web interface for a forum. At the top, there is a navigation bar with the Arduino Code Masters logo on the left and links for Home, Projetos, Produtos, Lojas, Contact, and a 'Conta' button on the right. The main content area has a dark background with red geometric patterns. It features a 'Faça uma pergunta:' section with two input fields: 'Título da pergunta' and 'Conteúdo da pergunta', followed by a blue 'Enviar pergunta' button. Below this is a 'Perguntas:' section with the heading 'Título da pergunta :' and the text 'ola'.

Figura 39

Cabeçalho da Página:

O cabeçalho contém um logótipo e links de navegação semelhantes aos exemplos anteriores.

Há também um botão "Conta" no canto superior direito, que direciona o utilizador para a página de detalhes da conta.

Secção de Formulário de Perguntas:

Os utilizadores podem fazer perguntas nesta secção.

Um formulário permite que eles insiram um título para a pergunta e o conteúdo da pergunta.

Quando o formulário é enviado, as perguntas são processadas no ficheiro "submit_question.php".

Lista de Perguntas:

As perguntas feitas pelos utilizadores são exibidas numa lista.

Cada pergunta tem um título e conteúdo.

As perguntas também podem ser expandidas ou recolhidas clicando no botão "alternar" para mostrar ou ocultar as respostas.

Botão de Alternância de Respostas:

Existe um botão "alternar" em cada pergunta que permite aos utilizadores mostrar ou ocultar as respostas a essa pergunta. Isso é feito usando JavaScript.

Rodapé da Página:

O rodapé contém informações de contacto, é estilizado com cores escuras e brancas e é fixado na parte inferior da página.

Estilos Visuais:

A página possui um estilo visual atraente com uma imagem de fundo, cores específicas e estilos de fonte.

Os campos do formulário de pergunta são estilizados com bordas e cores de fundo.

Botões, como o botão "Enviar pergunta" e o botão "alternar" para mostrar respostas, têm cores de destaque.

Lógica PHP:

A página faz uso de PHP para verificar se o utilizador está autenticado. Se o utilizador estiver autenticado, ele pode fazer perguntas; caso contrário, não.

Geralmente, esta página é um exemplo de um fórum de perguntas e respostas simples, com funcionalidade básica de fazer perguntas e mostrar respostas. Os utilizadores podem interagir com o conteúdo, fazendo perguntas e visualizando as respostas a outras perguntas.

PHP Forum para submeter uma pergunta

```
1 <?php
2 session_start();
3 include 'db.php';
4
5 if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_SESSION['id_usuario'])) {
6     $questionTitle = $_POST['question_title'];
7     $questionContent = $_POST['question_content'];
8     $authorId = $_SESSION['id_usuario']; // Obter o ID do usuário da sessão
9
10    $query = "INSERT INTO perguntas (titulo, pergunta, autor_id) VALUES ('$questionTitle', '$questionContent', $authorId)";
11    $result = mysqli_query($conn, $query);
12
13    if ($result) {
14        header("Location: perguntas_e_respostas.php"); // Redirecionar para a página principal
15    } else {
16        echo "Erro ao enviar pergunta: " . mysqli_error($conn);
17    }
18 } else {
19     echo "Você não está logado ou não enviou uma pergunta válida.";
20 }
21 ?>
22
```

Figura 40 PHP 12

Este código em PHP lida com a submissão de perguntas num sistema de perguntas e respostas.

`session_start();` inicia ou retoma uma sessão PHP.

Isso é frequentemente usado para rastrear informações do utilizador entre diferentes páginas da 'web'.

`include 'db.php';` inclui um arquivo chamado 'db.php'. Isso é comum para importar configurações de conexão com o banco de dados e outras funções relacionadas ao banco de dados.

O código verifica se a solicitação HTTP é do tipo POST e se o utilizador está autenticado (possui uma sessão ativa).

Se a condição do ponto 3 for verdadeira, o código recolhe os dados da pergunta enviados por meio de um formulário HTML. Os dados incluem o título da pergunta (`$questionTitle`), o conteúdo da pergunta (`$questionContent`), e o 'ID' do autor da pergunta (`$authorId`), obtido a partir da sessão.

Em seguida, é construída uma consulta SQL para inserir a pergunta na base de dados. A consulta é montada com os valores das variáveis \$questionTitle, \$questionContent, e \$authorId.

A consulta SQL é executada usando a função `mysqli_query`. Se a inserção for bem-sucedida, o código redireciona o utilizador para a página 'perguntas_e_respostas.php' usando a função `header("Location: perguntas_e_respostas.php")`.

Se a inserção falhar, é exibida uma mensagem de erro que inclui informações do erro MySQL usando `mysqli_error($conn)`.

Se a condição do ponto 3 for falsa (ou seja, se o utilizador não estiver autenticado ou a solicitação não for do tipo POST), o código exibe a mensagem "Você não está logado ou não enviou uma pergunta válida."

Em resumo, este código PHP permite que utilizadores autenticados submetam perguntas a um sistema de perguntas e respostas. Ele recolhe os dados da pergunta, insere esses dados na base de dados e redireciona o utilizador para a página principal do sistema após o envio bem-sucedido. Se ocorrer um erro durante a inserção, ele mostra uma mensagem de erro. Se o utilizador não estiver autenticado ou a solicitação for inválida, exibe uma mensagem informando o utilizador sobre o problema.

PHP das respostas

```
1  <?php
2  session_start();
3  include 'db.php';
4
5  if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_SESSION['id_usuario'])) {
6      $questionId = $_POST['question_id'];
7      $answerContent = $_POST['answer_content'];
8      $authorId = $_SESSION['id_usuario']; // Obter o ID do usuário da sessão
9
10     $query = "INSERT INTO respostas (pergunta_id, resposta, autor_id) VALUES ($questionId, '$answerContent', $authorId)";
11     $result = mysqli_query($conn, $query);
12
13     if ($result) {
14         header("Location: perguntas_e_respostas.php"); // Redirecionar para a página principal
15     } else {
16         echo "Erro ao enviar resposta: " . mysqli_error($conn);
17     }
18 } else {
19     echo "Você não está logado ou não enviou uma resposta válida.";
20 }
21 ?>
```

Figura 41 PHP 13

Este código em PHP lida com a submissão de respostas a perguntas num sistema de perguntas e respostas.

`session_start();` inicia ou retoma uma sessão PHP. Isso é frequentemente usado para rastrear informações do utilizador entre diferentes páginas da 'web'.

`include 'db.php';` inclui um arquivo chamado 'db.php'. Isso é comum para importar configurações de conexão com o banco de dados e outras funções relacionadas ao banco de dados.

O código verifica se a solicitação HTTP é do tipo POST e se o utilizador está autenticado (possui uma sessão ativa).

Se a condição do ponto 3 for verdadeira, o código recolhe os dados da resposta enviados por um formulário HTML. Os dados incluem o 'ID' da pergunta à qual a resposta está relacionada

(\$questionId), o conteúdo da resposta (\$answerContent), e o 'ID' do autor da resposta (\$authorId), obtido a partir da sessão.

Em seguida, é construída uma consulta SQL para inserir a resposta na base de dados. A consulta é montada com os valores das variáveis \$questionId, \$answerContent, e \$authorId.

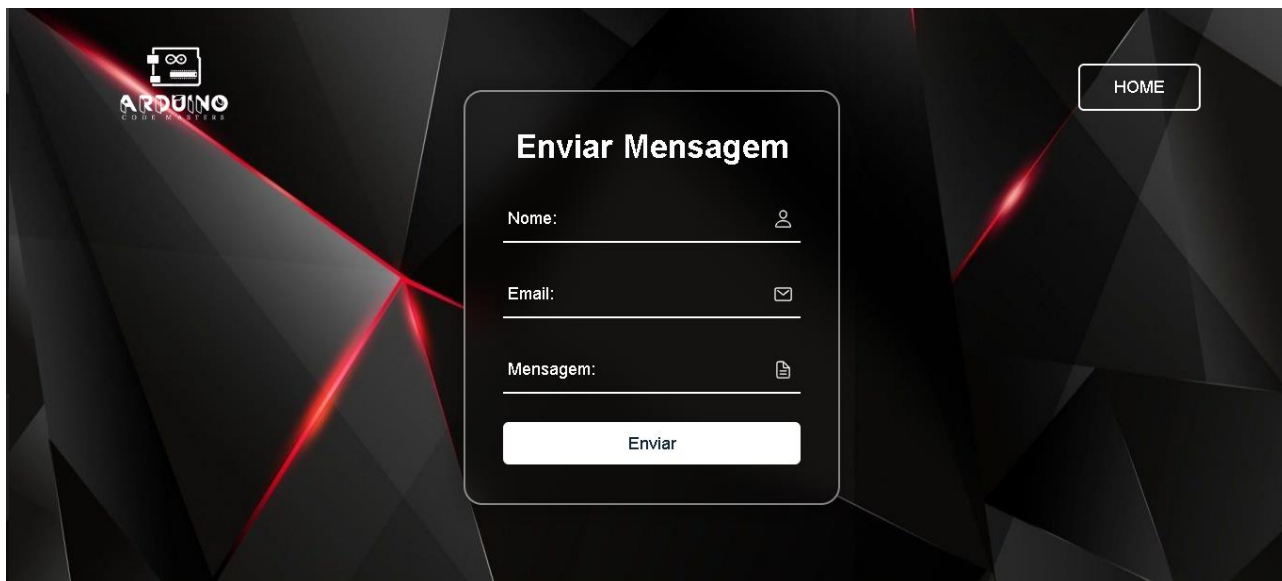
A consulta SQL é executada usando a função `mysqli_query`. Se a inserção for bem-sucedida, o código redireciona o utilizador para a página 'perguntas_e_respostas.php' usando a função `header("Location: perguntas_e_respostas.php")`.

Se a inserção falhar, é exibida uma mensagem de erro que inclui informações do erro MySQL usando `mysqli_error($conn)`.

Se a condição do ponto 3 for falsa (ou seja, se o utilizador não estiver autenticado ou a solicitação não for do tipo POST), o código exibe a mensagem "Você não está logado ou não enviou uma resposta válida."

Em resumo, este código PHP permite que utilizadores autenticados submetam respostas a perguntas num sistema de perguntas e respostas. Ele recolhe os dados da resposta, insere esses dados na base de dados e redireciona o utilizador para a página principal do sistema após o envio bem-sucedido. Se ocorrer um erro durante a inserção, ele mostra uma mensagem de erro. Se o utilizador não estiver autenticado ou a solicitação for inválida, exibe uma mensagem informando o utilizador sobre o problema.

Página de contacto



The screenshot shows a contact form titled "Enviar Mensagem" on a dark, geometric background with red laser lines. The form has three input fields: "Nome:" with a person icon, "Email:" with an envelope icon, and "Mensagem:" with a document icon. Below these fields is a white "Enviar" button. In the top right corner of the page, there is a "HOME" button. The Arduino Code Masters logo is visible in the top left corner of the form area.

Figura 42

Cabeçalho da Página:

O cabeçalho inclui um logótipo e um botão "INÍCIO" que permite aos utilizadores regressar à página inicial.

Formulário de Contacto:

Os utilizadores podem enviar mensagens através deste formulário de contacto.

Os campos do formulário incluem "Nome", "Email" e "Mensagem".

Quando os utilizadores preenchem os campos e clicam no botão "Enviar", o formulário é enviado para o URL especificado em "action" (neste caso, <https://formspre.io/f/xpzenyqk>).

Botão "Cancelar":

Existe também um botão "Cancelar" que permite aos utilizadores cancelar o envio do formulário e regressar à página inicial. Isso é feito através da função JavaScript "cancelar()".

Estilos Visuais:

A página possui um estilo visual atraente, com um fundo e estilos de fonte.

Os campos do formulário são estilizados com ícones e etiquetas.

Os botões "Enviar" e "Cancelar" são estilizados com cores de destaque.

Ícones:

A página utiliza ícones com a biblioteca Ionicons para melhorar a experiência visual.

Em resumo, esta página exemplifica um formulário de contacto simples. Os utilizadores podem preencher os seus detalhes, escrever uma mensagem e clicar em "Enviar" para enviar a mensagem de contacto. Se desejarem cancelar o envio, podem clicar no botão "Cancelar".

Página da conta de utilizador



Figura 43

Início da Sessão:

O script começa por iniciar uma sessão utilizando `session_start()`. Isso é geralmente usado para acompanhar o estado do utilizador e armazenar informações de sessão, como o ID do utilizador.

Verificação da Conexão do Utilizador:

O script verifica se o utilizador está autenticado ao verificar se a variável de sessão `$_SESSION['id_usuario']` está definida. Isto é feito para garantir que apenas utilizadores autenticados possam aceder a esta página.

Conexão com a Base de Dados:

Se o utilizador estiver autenticado, o script estabelece uma ligação a uma base de dados MySQL utilizando as credenciais de ligação (`$servername`, `$username`, `$password`, `$dbname`).

Ele utiliza a função `mysqli_connect` para estabelecer a ligação.

Recuperação dos Detalhes do Utilizador:

Após uma ligação bem-sucedida, o script recupera o ID do utilizador da sessão e executa uma consulta SQL para buscar os detalhes do utilizador na base de dados. A consulta SQL é armazenada na variável \$sql.

Exibição dos Detalhes do Utilizador:

Se a consulta retornar um único registo (o utilizador existe), os detalhes do utilizador, como nome e correio eletrónico, são recuperados da consulta e exibidos na página HTML.

Os detalhes do utilizador são exibidos num layout estilizado.

Botões de Ação:

Três botões são exibidos na parte inferior:

"Editar informações" redireciona o utilizador para uma página onde pode editar as suas informações.

"Voltar para a página inicial" leva o utilizador de volta à página inicial.

"Terminar Sessão" permite ao utilizador encerrar a sessão e sair da conta.

Estilos Visuais:

A página possui estilos visuais personalizados, incluindo fundos, sombras e tipos de letra.

Geralmente, este script é usado para exibir os detalhes do perfil de um utilizador após o login. Ele também fornece opções para editar informações ou sair da conta. Certifique-se de que a base de dados e as credenciais estejam configuradas corretamente para que o script funcione conforme o esperado.

Página para editar detalhes de utilizador

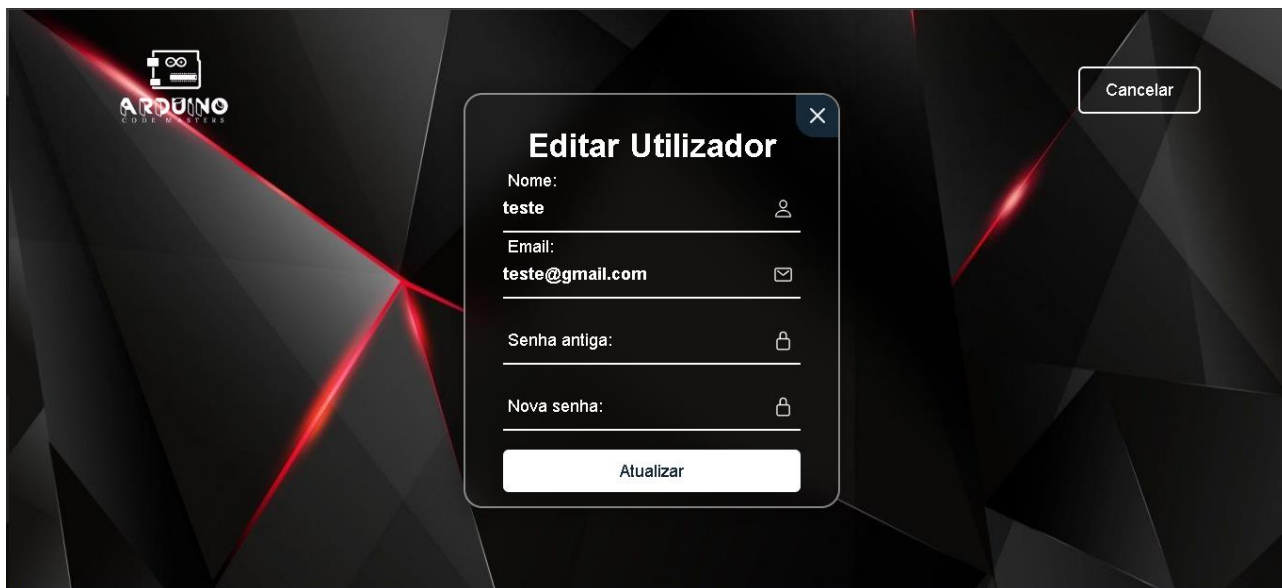


Figura 44

Início da Sessão:

O script começa por iniciar uma sessão com `session_start()`, utilizada para acompanhar o estado do utilizador e armazenar informações de sessão.

Verificação de Conexão do Utilizador:

O script verifica se o utilizador está autenticado verificando se a variável de sessão `$_SESSION['id_usuario']` está definida. Isso garante que apenas utilizadores autenticados possam editar as informações do perfil.

Conexão com a Base de Dados:

Após a verificação da autenticação, o script estabelece uma conexão com um banco de dados MySQL. Ele usa as credenciais de conexão (`$servername`, `$username`, `$password`, `$dbname`) e a função `mysqli_connect`.

Edição das Informações do Utilizador:

Se o método da solicitação for POST (ou seja, o formulário foi submetido), o script processará as informações fornecidas pelo utilizador para atualizar o perfil.

Ele obtém os campos do formulário, como nome, email, senha antiga e nova senha, e verifica se a senha antiga corresponde à senha armazenada na base de dados antes de permitir a atualização dos detalhes do perfil.

Se as informações estiverem corretas, a base de dados é atualizada com os novos dados do utilizador.

Exibição do Formulário de Edição:

Se o utilizador estiver autenticado e a solicitação não for POST, o script exibe um formulário com os detalhes atuais do utilizador (nome e email) e campos para inserir a senha antiga e a nova senha.

Os campos são pré-preenchidos com os dados atuais do utilizador.

Botões de Ação:

A parte inferior do formulário exibe um botão "Atualizar" que permite ao utilizador enviar o formulário para atualização. Além disso, há um botão "Cancelar" que redireciona o utilizador de volta à página de detalhes da conta.

Estilos Visuais:

O script inclui estilos visuais personalizados para formatar o formulário e a página. Ele também faz uso de ícones para melhorar a experiência do utilizador.

Este script é utilizado para permitir que os utilizadores autenticados atualizem as suas informações de perfil, como nome, email e senha, desde que forneçam a senha antiga correta. Certifique-se de configurar corretamente a base de dados e as credenciais de conexão para que o script funcione conforme o esperado.

Dashboard

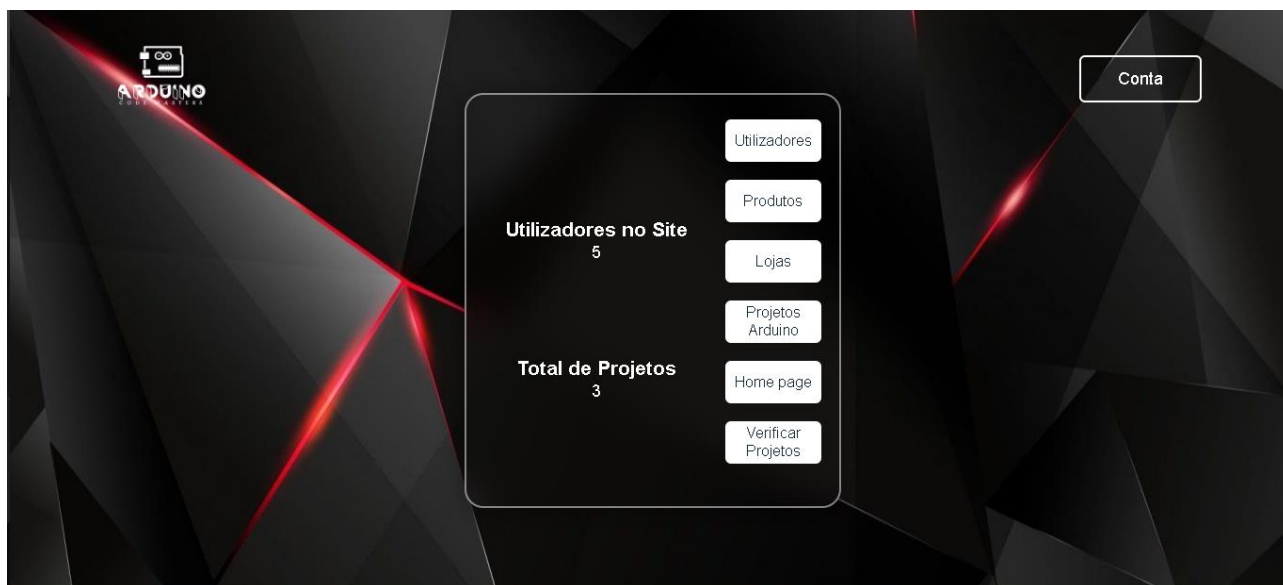


Figura 45

Inclusão do Verificador de Administrador:

O código começa por incluir o arquivo "verificaradm.php". Esta inclusão indica que esta página só é acessível a utilizadores com privilégios administrativos.

Configuração Inicial:

São definidas as configurações iniciais da página, como o tipo de conteúdo (HTML), a codificação (UTF-8) e o título da página ("Dashboard").

Estilos CSS:

São definidos estilos CSS para formatar a página. Isso inclui definições para o corpo, cabeçalho, caixas de informação, botões e outros elementos visuais.

Cabeçalho da Página:

O cabeçalho da página inclui um título "Dashboard" e um botão "Detalhes da Conta". Este botão permite ao utilizador aceder aos detalhes da sua conta.

Configuração da Ligação à Base de Dados:

As configurações de conexão à base de dados MySQL são definidas, incluindo o nome do servidor, nome de utilizador, palavra-passe e nome da base de dados.

Estabelecimento de Conexão com a Base de Dados:

É criada uma conexão com o servidor MySQL utilizando as configurações definidas anteriormente. Se a conexão falhar, o código informa o erro.

Consulta de Número Total de Utilizadores:

Uma consulta SQL é executada para obter o número total de utilizadores registados no sistema a partir da tabela "usuarios". O resultado é armazenado na variável \$totalUsuarios.

Consulta de Número Total de Projetos:

Outra consulta SQL é executada para obter o número total de projetos Arduino na base de dados a partir da tabela "projetos_arduino". O resultado é armazenado na variável \$totalProjetos.

Encerramento da Conexão com a Base de Dados:

A conexão com o servidor MySQL é encerrada para libertar recursos.

Exibição de Informações:

São apresentadas duas caixas na secção da página. Uma mostra o número total de utilizadores e a outra mostra o número total de projetos Arduino, utilizando as variáveis PHP \$totalUsuarios e \$totalProjetos.

Botões de Navegação:

É apresentada uma série de botões de navegação no final da página. Cada botão redireciona o administrador para diferentes partes do sistema, como a gestão de utilizadores, produtos, lojas, projetos Arduino, página inicial, etc.

Estilos dos Botões:

Os botões são estilizados com cores específicas e efeitos de destaque quando o cursor do rato passa por cima.

Este código PHP/HTML cria uma página de painel de controlo administrativo que apresenta informações importantes e fornece acesso fácil a várias funcionalidades do sistema.

Página de gestão de utilizadores

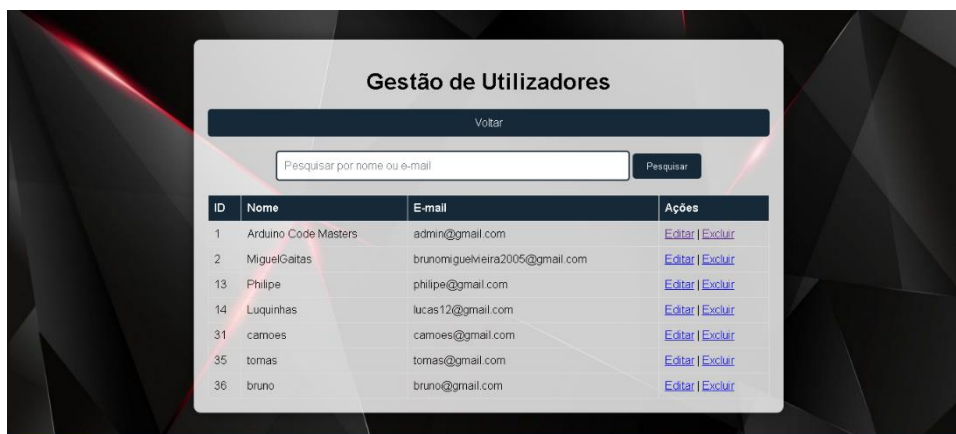


Figura 46

Inclusão do Verificador de Administrador:

O código inicia com a inclusão do arquivo "verificaradm.php", que verifica se o utilizador tem privilégios de administrador. Este controle garante que apenas administradores podem aceder a esta página.

Configuração Inicial:

O código define as configurações iniciais da página, incluindo o título "Gestão de Utilizadores" e a ligação para o ficheiro CSS "adminusuarios.css" para estilos adicionais.

Corpo da Página:

A estrutura HTML da página começa com o corpo (<body>) e um elemento <div> com a classe "contêiner".

Título da Página e Botão de Voltar:

No interior da <div>, é exibido o título "Gestão de Utilizadores" e um botão "Voltar". Este botão permite ao utilizador voltar à página de administração principal ("pagina_de_admin.php").

Tabela de Utilizadores:

Abaixo do botão "Voltar", é exibida uma tabela que mostra os detalhes dos utilizadores. A tabela tem uma estrutura de cabeçalho com as colunas: "ID", "Nome", "Correio eletrónico" e "Ações".

Conexão à Base de Dados:

O código define as configurações de conexão à base de dados, como o nome do servidor, nome de utilizador, palavra-passe e nome da base de dados. Em seguida, estabelece uma ligação à base de dados MySQL.

Se a ligação falhar, é apresentada uma mensagem de erro.

Consulta SQL e Exibição dos Utilizadores:

Uma consulta SQL é executada para selecionar todos os utilizadores da tabela "usuarios". Os resultados da consulta são armazenados na variável \$resultado.

Um loop é usado para percorrer todos os utilizadores e exibir as suas informações nas linhas da tabela.

As informações exibidas para cada utilizador incluem o "ID", "Nome" e "Correio eletrónico". Além disso, são apresentadas ligações "Editar" e "Excluir" que permitem ao administrador editar ou eliminar utilizadores.

Encerramento da Conexão com a Base de Dados:

Após a conclusão da exibição dos utilizadores, a ligação com a base de dados é encerrada para libertar recursos.

Este código PHP/HTML cria uma página de gestão de utilizadores, onde os administradores podem ver a lista de utilizadores, editar ou eliminar utilizadores e voltar à página principal de administração.

Página adicionar componentes

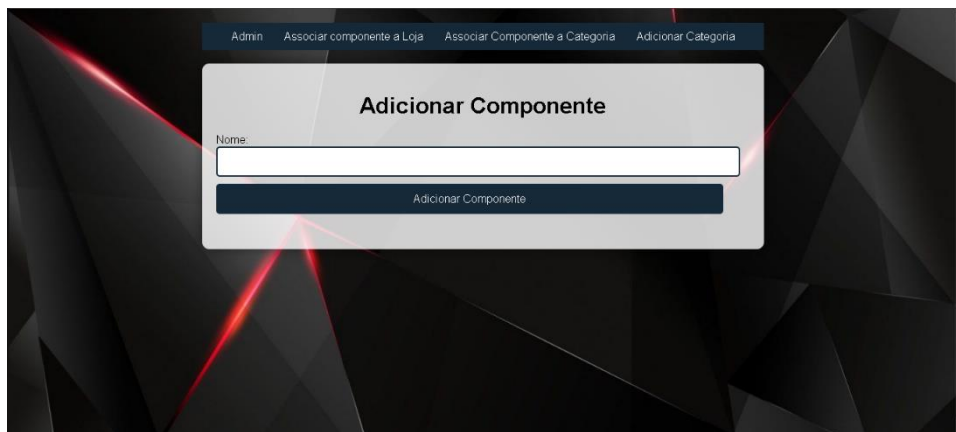


Figura 47

Este código PHP está associado a uma página web que permite a um administrador adicionar novos componentes a um sistema.

1. Autenticação de Administrador:

- Antes de permitir a execução do código, há uma inclusão (`include`) de um arquivo chamado "verificaradm.php". Isso sugere que existe algum tipo de verificação para garantir que apenas administradores têm permissão para adicionar novos componentes.

2. Formulário de Adição de Componente:

- A página contém um formulário HTML que coleta informações sobre um novo componente.
- Os campos do formulário incluem o "Nome" do componente e uma quantidade utilizada (definida como 0, conforme indicado por um campo oculto).
- Quando o formulário é submetido, os dados são enviados para a mesma página (`action=""`), onde o código PHP processa a adição do componente ao banco de dados.

3. Conexão ao Banco de Dados e Inserção de Dados:

- O código estabelece uma conexão com um banco de dados MySQL usando informações de conexão como servidor, nome de utilizador, senha e nome do banco de dados.

- Após verificar a conexão bem-sucedida, os dados do formulário são recebidos e uma instrução SQL é executada para inserir o novo componente na tabela "componentes" do banco de dados
- . - Mensagens de alerta são exibidas ao administrador, informando se a operação foi bem-sucedida ou ocorreu-se algum erro durante o processo.

4. Estilo da Página:

- A página possui um estilo simples usando CSS para garantir uma apresentação visual agradável.
- O fundo da página é definido como uma imagem de repetição, e o conteúdo centralizado na tela para uma melhor experiência de utilizador.
- A navegação é fornecida através de links na parte superior da página para outras áreas administrativas.

Em resumo, esta página permite que um administrador insira novos componentes no sistema, garantindo que apenas administradores autenticados possam realizar essa operação. A interface é projetada para ser intuitiva, e mensagens de feedback são exibidas para informar sobre o sucesso ou falha da adição do componente.

Página Associar Loja a Componente

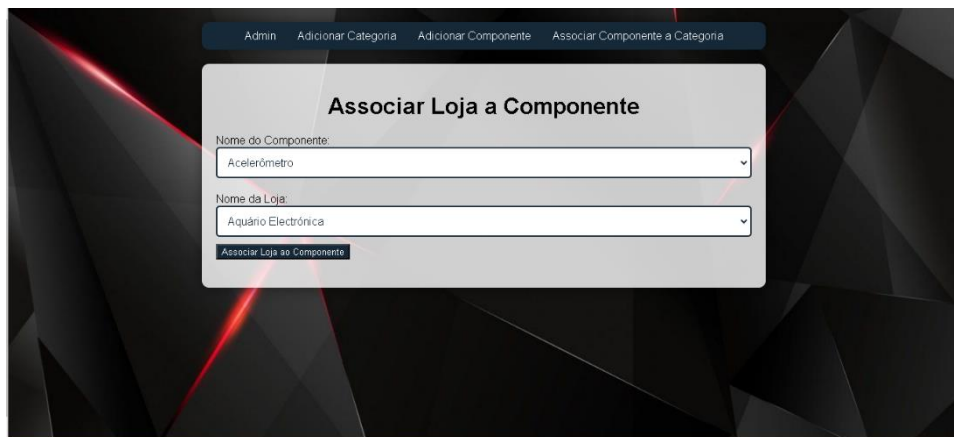


Figura 48

Este código PHP e HTML está associado a uma página web que permite a um administrador associar componentes a lojas num sistema. Abaixo estão alguns pontos notáveis sobre esta página:

1. Conexão com o Banco de Dados:

- Estabelece uma conexão com um banco de dados MySQL utilizando as credenciais fornecidas.

2. Funções para Obter IDs:

- Fornece duas funções (`obterIdLoja` e `obterIdComponente`) para obter os IDs de lojas e componentes com base nos seus nomes, respetivamente.

3. Obtenção de Listas do Banco de Dados:

- Executa consultas SQL para obter listas de componentes e lojas presentes no banco de dados, ordenadas por nome.

4. Processamento do Formulário:

- Ao enviar o formulário, os dados são processados no lado do servidor.
- Os IDs do componente e da loja são obtidos usando as funções mencionadas.
- Se ambos os IDs são obtidos com sucesso, uma nova entrada é adicionada à tabela "loja_componente" associando a loja ao componente.

5. Mensagens de Status:

- Exibe mensagens de status (como "Associação bem-sucedida" ou mensagens de erro) no topo da página para informar o administrador sobre o resultado da operação.

6. Estilo da Página:

- A página possui um layout simples e limpo com um estilo visual agradável.
- A navegação é fornecida por links na parte superior da página para outras áreas administrativas.
- O fundo é definido com uma imagem de repetição, e o conteúdo é centralizado na tela.

Em resumo, esta página permite que um administrador associe lojas a componentes no sistema. A interface é projetada para ser fácil de usar, e mensagens claras são fornecidas para informar sobre o resultado das operações de associação.

Página Adicionar Categoria

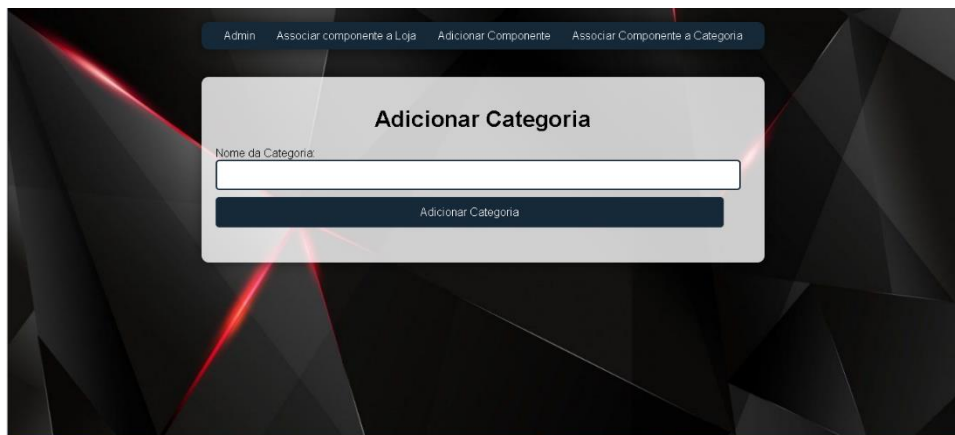


Figura 49

Inclusão de Verificação de Administrador:

Inclui um script PHP ("verificaradm.php") para verificar se o utilizador é um administrador antes de permitir o acesso à página. Isso é uma medida de segurança para restringir o acesso apenas a administradores.

Processamento do Formulário:

Ao enviar o formulário, os dados são processados no lado do servidor.

Conecta-se ao banco de dados MySQL utilizando as credenciais fornecidas.

Recebe o nome da categoria a ser adicionada e insere esses dados na tabela "categoria" no banco de dados.

Mensagens de Status:

Exibe mensagens de status (como "Categoria adicionada com sucesso" ou mensagens de erro) utilizando JavaScript. Essas mensagens são exibidas na forma de pop-ups ("alert").

Estilo da Página:

A página possui um layout simples e limpo com um estilo visual agradável.

A navegação é fornecida por links na parte superior da página para outras áreas administrativas.

O fundo é definido com uma imagem de repetição, e o conteúdo é centralizado na tela.

Página Associar Componente a Categoria

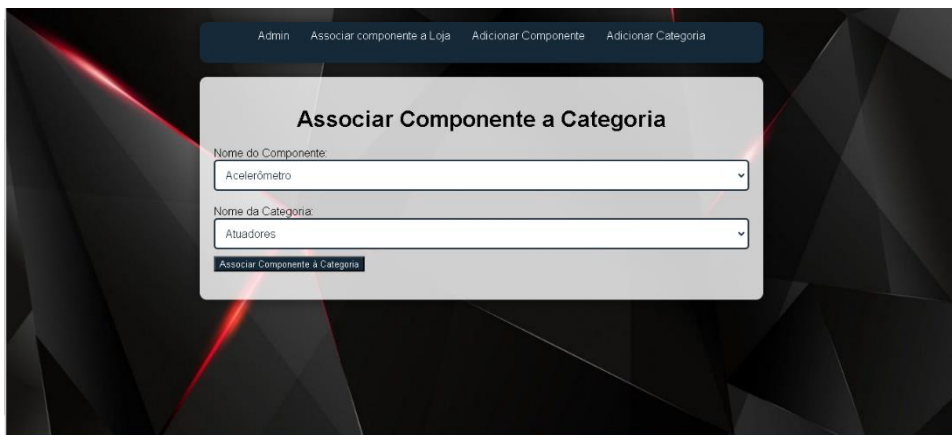


Figura 50

Este código PHP e HTML está associado a uma página web que permite a um administrador associar um componente a uma categoria.

Página "Associar Componente a Categoria":

Conexão com o Banco de Dados:

Realiza a conexão com o banco de dados MySQL usando as credenciais fornecidas.

Funções para Obter IDs:

Fornece duas funções (obterIdCategoria e obterIdComponente) para obter o ID de uma categoria ou componente com base no nome.

Consulta e Exibição de Opções:

Obtém a lista de componentes e categorias do banco de dados para exibição em listas suspensas (<select>).

Processamento do Formulário:

Ao enviar o formulário, os dados são processados no lado do servidor.

Obtém o nome do componente e da categoria a serem associados e, em seguida, obtém os IDs correspondentes usando as funções mencionadas anteriormente.

Realiza a associação inserindo um novo registo na tabela "componentes_categorias" no banco de dados.

Mensagens de Status:

Exibe mensagens de status utilizando JavaScript para indicar se a associação foi bem-sucedida ou ocorreu-se algum erro.

Estilo da Página:

A página possui um layout simples e limpo com um estilo visual agradável.

A navegação é fornecida por links na parte superior da página para outras áreas administrativas.

O fundo é definido com uma imagem de repetição, e o conteúdo é centralizado na tela.

Página adicionar lojas



The screenshot shows a web form titled "Adicionar loja". At the top, there is a dark blue button labeled "Voltar para a página de administração". Below this, the form contains five input fields, each with a label to its left: "Nome da loja:", "Morada:", "Cidade:", "Código postal:", and "Telefone:". The form is set against a dark background with red light effects.

Figura 51

Inclusão do Verificador de Administrador:

O código começa por incluir o arquivo "verificaradm.php" para verificar se o utilizador possui privilégios de

administrador. Isso assegura que apenas administradores tenham acesso a esta página.

Configuração Inicial:

O código define as configurações iniciais da página, incluindo o título "Adicionar Loja" e o ícone da página ("favicon").

Cabeçalho da Página:

O cabeçalho da página contém um título "Adicionar Loja" e um botão "Voltar para a admin page" que permite ao utilizador regressar à página de administração principal ("pagina_de_admin.php").

Formulário para Adicionar Loja:

O código apresenta um formulário para adicionar informações sobre uma nova loja.

O formulário inclui campos para "Nome da Loja," "Endereço," "Cidade," "Código Postal," "Telefone" e uma secção para seleccionar produtos disponíveis. O utilizador pode preencher estes campos e seleccionar produtos por meio de caixas de verificação.

Há um botão "Adicionar Loja" para submeter o formulário.

Estilos CSS:

O código CSS incorporado define os estilos gerais da página, do cabeçalho, do menu, do conteúdo, do formulário, da tabela e das mensagens de erro/sucesso.

Também define o estilo de um botão "Voltar" fixo no canto superior direito da página.

Lista de Lojas Existente:

O código apresenta uma tabela que lista lojas existentes.

A tabela possui as seguintes colunas: "Nome," "Endereço," "Cidade," "Código Postal," "Telefone," "Produtos Disponíveis" e "Ações".

O código PHP recolhe informações sobre lojas a partir da base de dados e exibe-as na tabela. Além disso, exibe os produtos disponíveis em cada loja.

A coluna "Ações" inclui uma hiperligação "Editar" para editar detalhes da loja.

Encerramento da Página:

A página é fechada com as etiquetas `</body>` e `</html>`.

Este código PHP/HTML cria uma página para adicionar e gerir lojas, onde os administradores podem adicionar novas lojas, ver a lista de lojas existentes, editar detalhes de lojas e regressar à página principal de administração.

Página adicionar projetos do admin

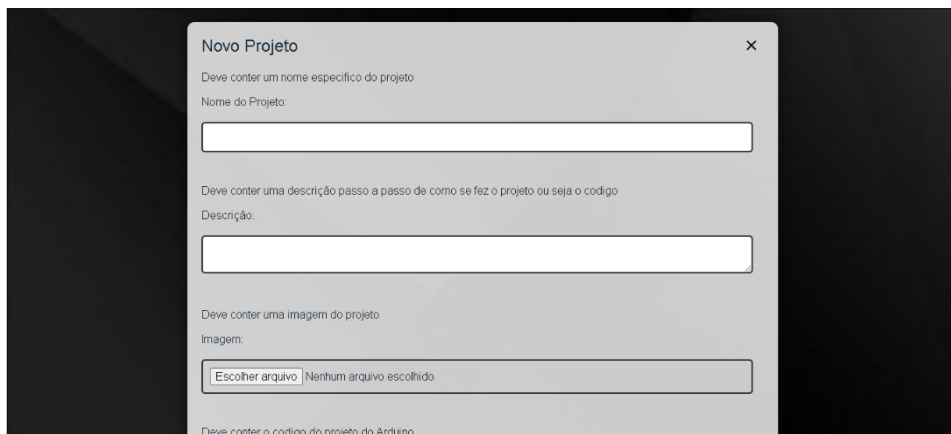


Figura 52

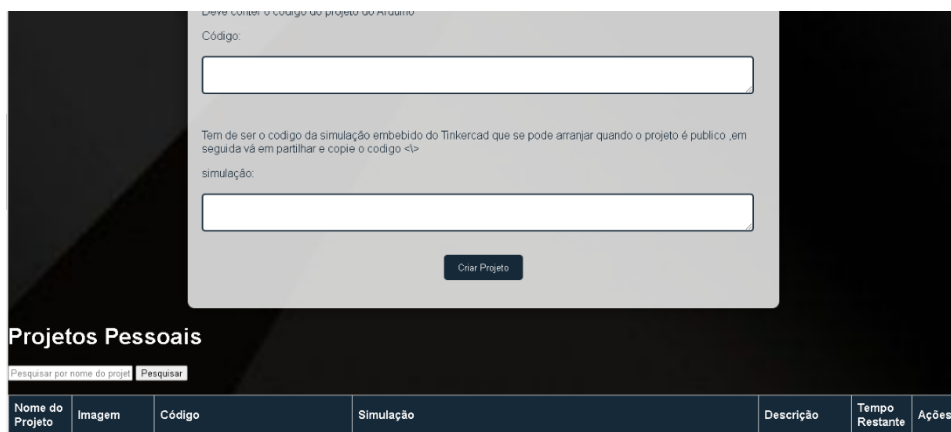


Figura 53

Inclusão do Verificador de Administrador:

O código começa por incluir o arquivo "verificaradm.php" para verificar se o utilizador possui privilégios de administrador. Isso assegura que apenas administradores tenham acesso a esta página.

Configurações Iniciais:

O código define as configurações iniciais da página, incluindo o título "Adicionar Projeto Arduino" e o ícone da página ("favicon").

Formulário para Adicionar Projeto Arduino:

O código apresenta um formulário que permite ao utilizador adicionar informações sobre um novo projeto Arduino.

O formulário inclui campos para "Nome do Projeto," "Descrição," "Imagem," "Código," "Simulação."
O utilizador pode preencher estes campos e fazer o upload de uma imagem do projeto.

Há um botão "Adicionar Projeto" para submeter o formulário.

Estilos CSS:

O código CSS incorporado define os estilos gerais da página, incluindo os estilos para os cabeçalhos, formulário, tabelas, imagens e links.

Lista de Projetos Arduino Existente:

O código apresenta uma tabela que lista os projetos Arduino existentes.

A tabela inclui colunas para "Nome," "Descrição," "Imagem," "Código," "Simulação," "Editar" e "Excluir."

O código PHP recolhe informações sobre os projetos Arduino a partir da base de dados e exibe-as na tabela. Além disso, o nome da imagem é apresentado na coluna "Imagem."

A coluna "Editar" contém um link que permite ao utilizador editar os detalhes do projeto Arduino.

A coluna "Excluir" contém um link que permite ao utilizador excluir o projeto Arduino, mostrando uma mensagem de confirmação.

Se não houver projetos Arduino na base de dados, é apresentada uma mensagem a informar "Nenhum projeto encontrado."

Este código PHP/HTML cria uma página que permite aos administradores adicionar projetos Arduino, ver a lista de projetos existentes, editar detalhes dos projetos e excluir projetos.

Projetos pendentes para verificação



Figura 54

Inclusão do Verificador de Administrador:

O código começa por incluir o arquivo "verificaradm.php" para verificar se o utilizador possui privilégios de administrador. Isso assegura que apenas administradores tenham acesso a esta página.

Configurações Iniciais:

O código define as configurações iniciais da página, incluindo o título "Projetos Pendentes" e os estilos CSS que não estão especificados no código, mas são indicados para serem adicionados.

Consulta de Projetos Pendentes:

O código estabelece uma conexão com o banco de dados, utilizando as credenciais especificadas.

Uma consulta SQL é executada para recuperar os projetos que estão marcados como "pendentes" na base de dados.

Os resultados desta consulta são armazenados na variável "\$result."

Listagem de Projetos Pendentes:

O código PHP verifica se existem projetos pendentes na base de dados, percorrendo os resultados da consulta.

Para cada projeto pendente, o código exibe um item de lista (representado por "") que inclui o nome do projeto e um link para "verificar_projeto.php" com o ID do projeto como parâmetro. Isso permite que o administrador verifique ou tome ações sobre o projeto.

Se não houver projetos pendentes na base de dados, o código exibe a mensagem "Nenhum projeto pendente para verificação."

Fecho da Conexão com o Banco de Dados:

Após a listagem dos projetos pendentes, o código fecha a conexão com o banco de dados usando "\$conn->close()". Isso é uma boa prática para liberar recursos após o uso.

Em resumo, este código PHP/HTML cria uma página onde os administradores podem visualizar os projetos que estão pendentes para verificação. Cada projeto pendente é exibido na forma de uma lista, e o administrador pode clicar no nome do projeto para verificar o projeto ou tomar as ações necessárias.

Base de dados

Ligação

Para proceder à ligação do sítio 'web' à base de dados, foi necessário criar um ficheiro PHP com 4 variáveis necessárias para essa ligação seja efetuada com sucesso. São elas o nome do servidor onde está alojada a base de dados, o nome de utilizador do servidor, a password e o nome da base de dados que vais ser manipulada pelo sítio 'web'.

```
<?php
// Configurações do banco de dados
$servername = "localhost";
$username = "id20757658_miguelgaitas";
$password = "MiguelGaitas24.";
$dbname = "id20757658_dados_dos_registros";

// Cria a conexão com o banco de dados
$conexao = mysqli_connect($servername, $username, $password, $dbname);

// Verifica a conexão
if (!$conexao) {
    die("Conexão falhou: " . mysqli_connect_error());
}
?>
```

Figura 55 PHP 14

Definição das tabelas

Lojas: Esta tabela armazena informações sobre lojas. Cada registo na tabela representa uma loja e inclui detalhes como nome, endereço, cidade, código postal e telefone. Pode ser utilizada para listar todas as lojas disponíveis.

Perguntas: A tabela "perguntas" é utilizada para armazenar perguntas feitas pelos utilizadores. Cada registo representa uma pergunta e contém informações como o 'ID' da pergunta, o conteúdo da pergunta e o 'ID' do utilizador que fez a pergunta.

Projetos_arduino: esta tabela é utilizada para armazenar informações sobre projetos Arduino. Cada registo representa um projeto e inclui detalhes como nome, descrição, imagem, código e simulação. Pode ser usada para listar e gerir projetos Arduino disponíveis.

Respostas: A tabela "respostas" armazena respostas a perguntas feitas pelos utilizadores. Cada registo contém informações como o 'ID' da resposta, o conteúdo da resposta e o 'ID' da pergunta associada. Ela permite que os utilizadores obtenham respostas às suas perguntas.

token_reset_senha: esta tabela é utilizada para gerir os 'tokens' de reposição de senha. Quando um utilizador solicita uma reposição de senha, um 'token' é gerado e armazenado nesta tabela. Ela regista o 'token', o 'ID' do utilizador associado e a data de expiração.

Utilizadores: A tabela "utilizadores" armazena informações sobre os utilizadores registados na sua aplicação. Cada registo inclui dados como o nome, email, password e informações de perfil. É fundamental para autenticar e gerir utilizadores no sistema.

componentes: Esta tabela armazena informações sobre os componentes utilizados em projetos. Cada registo representa um componente específico e contém detalhes como nome, descrição, quantidade em 'stock', entre outros. O campo quantidade_utilizada pode ser usado para rastrear a quantidade desse componente utilizada em projetos.

categorias: A tabela categorias organiza os componentes em diferentes categorias com base em características semelhantes. Cada categoria é identificada por um 'ID' único e possui um nome descritivo. Esta tabela ajuda na classificação e organização dos componentes para facilitar a busca e a compreensão.

componentes_categorias: Esta tabela estabelece uma relação muitos-para-muitos entre componentes e categorias. Cada registo indica que um componente específico pertence a uma determinada categoria. Os campos componente_id e categoria_id são chaves estrangeiras que referenciam as tabelas componentes e categorias, respetivamente. Essa tabela permite associar um componente a várias categorias e vice-versa.

projetos_componentes: A tabela "projetos_componentes" é uma entidade que estabelece uma relação de associação entre projetos e componentes num sistema de gestão. Ela é utilizada para mapear quais componentes estão associados a quais projetos. A tabela geralmente contém registos que representam as conexões entre identificadores únicos de projetos e componentes. Essa estrutura é comumente empregada em contextos nos quais é necessário rastrear e gerir a relação dinâmica entre projetos e os diversos componentes que os compõem.

Apresentarei então a estrutura das 11 tabelas

Tabela lojas :

	#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Ação
<input type="checkbox"/>	1	id 	int(11)			Não	Nenhum		AUTO_INCREMENT	 Muda  Eliminar  Mais
<input type="checkbox"/>	2	nome	varchar(255)	utf8mb4_general_ci		Não	Nenhum			 Muda  Eliminar  Mais
<input type="checkbox"/>	3	endereco	varchar(255)	utf8mb4_general_ci		Não	Nenhum			 Muda  Eliminar  Mais
<input type="checkbox"/>	4	cidade	varchar(255)	utf8mb4_general_ci		Não	Nenhum			 Muda  Eliminar  Mais
<input type="checkbox"/>	5	codigo_postal	varchar(8)	utf8mb4_general_ci		Não	Nenhum			 Muda  Eliminar  Mais
<input type="checkbox"/>	6	telefone	varchar(20)	utf8mb4_general_ci		Não	Nenhum			 Muda  Eliminar  Mais

Figura 56 base de dados 1

Tabela perguntas:

	#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Ação
<input type="checkbox"/>	1	id 	int(11)			Não	Nenhum		AUTO_INCREMENT	 Muda  Eliminar  Mais
<input type="checkbox"/>	2	titulo	varchar(255)	utf8mb4_general_ci		Sim	NULL			 Muda  Eliminar  Mais
<input type="checkbox"/>	3	pergunta	text	utf8mb4_general_ci		Sim	NULL			 Muda  Eliminar  Mais
<input type="checkbox"/>	4	data_publicacao	timestamp			Não	current_timestamp()			 Muda  Eliminar  Mais
<input type="checkbox"/>	5	autor_id	int(11)			Sim	NULL			 Muda  Eliminar  Mais

Figura 57 base de dados 2

Tabela projetos_arduino :




























	#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Ação
<input type="checkbox"/>	1	id 🔑	int(11)			Não	Nenhum		AUTO_INCREMENT	 Muda  Eliminar  Mais
<input type="checkbox"/>	2	autor 🔑	int(11)			Não	Nenhum			 Muda  Eliminar  Mais
<input type="checkbox"/>	3	nome	varchar(255)	utf8mb4_general_ci		Não	Nenhum			 Muda  Eliminar  Mais
<input type="checkbox"/>	4	descricao	text	utf8mb4_general_ci		Não	Nenhum			 Muda  Eliminar  Mais
<input type="checkbox"/>	5	imagem	varchar(255)	utf8mb4_general_ci		Não	Nenhum			 Muda  Eliminar  Mais
<input type="checkbox"/>	6	codigo	text	utf8mb4_general_ci		Não	Nenhum			 Muda  Eliminar  Mais
<input type="checkbox"/>	7	simula	text	utf8mb4_general_ci		Sim	NULL			 Muda  Eliminar  Mais
<input type="checkbox"/>	8	status	varchar(50)	utf8mb4_general_ci		Não	pendente			 Muda  Eliminar  Mais
<input type="checkbox"/>	9	prazo_edicao	datetime			Sim	NULL			 Muda  Eliminar  Mais

Figura 58 base de dados 3

Tabela respostas :

	#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Ação
<input type="checkbox"/>	1	id 🔑	int(11)			Não	Nenhum		AUTO_INCREMENT	 Muda  Eliminar  Mais
<input type="checkbox"/>	2	pergunta_id 🔑	int(11)			Sim	NULL			 Muda  Eliminar  Mais
<input type="checkbox"/>	3	resposta	text	utf8mb4_general_ci		Sim	NULL			 Muda  Eliminar  Mais
<input type="checkbox"/>	4	data_publicacao	timestamp			Não	current_timestamp()			 Muda  Eliminar  Mais
<input type="checkbox"/>	5	autor_id	int(11)			Sim	NULL			 Muda  Eliminar  Mais

Figura 59 base de dados 4

Tabela tokens_reset_senha :








	#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Ação
<input type="checkbox"/>	1	id 	int(11)			Não	Nenhum		AUTO_INCREMENT	 Muda  Eliminar Mais
<input type="checkbox"/>	2	usuario_id	int(11)			Não	Nenhum			 Muda  Eliminar Mais
<input type="checkbox"/>	3	token	varchar(255)	utf8mb4_general_ci		Não	Nenhum			 Muda  Eliminar Mais
<input type="checkbox"/>	4	expiracao	datetime			Não	Nenhum			 Muda  Eliminar Mais

Figura 60 base de dados 5

Tabela utilizadores :














	#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Ação
<input type="checkbox"/>	1	id 	int(11)			Não	Nenhum		AUTO_INCREMENT	 Muda  Eliminar Mais
<input type="checkbox"/>	2	nome	varchar(50)	utf8mb4_general_ci		Não	Nenhum			 Muda  Eliminar Mais
<input type="checkbox"/>	3	email	varchar(100)	utf8mb4_general_ci		Não	Nenhum			 Muda  Eliminar Mais
<input type="checkbox"/>	4	senha	varchar(100)	utf8mb4_general_ci		Não	Nenhum			 Muda  Eliminar Mais
<input type="checkbox"/>	5	admin	tinyint(1)			Não	0			 Muda  Eliminar Mais
<input type="checkbox"/>	6	status	enum('nao_verificado', 'verificado')	utf8mb4_general_ci		Não	nao_verificado			 Muda  Eliminar Mais

Figura 61 base de dados 6

Tabela categorias:





	#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Ação
<input type="checkbox"/>	1	id 	int(11)			Não	Nenhum		AUTO_INCREMENT	 Muda  Eliminar Mais
<input type="checkbox"/>	2	nome	varchar(255)	utf8mb4_general_ci		Não	Nenhum			 Muda  Eliminar Mais

Figura 62 base de dados 7

Tabela componentes:

	#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Ação
<input type="checkbox"/>	1	id 🔑	int(11)			Não	Nenhum		AUTO_INCREMENT	Muda Eliminar Mais
<input type="checkbox"/>	2	nome	varchar(255)	utf8mb4_general_ci		Não	Nenhum			Muda Eliminar Mais
<input type="checkbox"/>	3	quantidade_utilizada	int(11)			Não	0			Muda Eliminar Mais

Figura 63 base de dados 8

Tabela componentes_categorias:

	#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários	Extra	Ação
<input type="checkbox"/>	1	id 🔑	int(11)			Não	Nenhum		AUTO_INCREMENT	Muda Eliminar Mais
<input type="checkbox"/>	2	componente_id 🔑	int(11)			Sim	NULL			Muda Eliminar Mais
<input type="checkbox"/>	3	categoria_id 🔑	int(11)			Sim	NULL			Muda Eliminar Mais

Figura 64 base de dados 9

Tabela projetos_componentes:

	#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários
<input type="checkbox"/>	1	projeto_id 🔑	int(11)			Não	Nenhum	
<input type="checkbox"/>	2	componente_id 🔑 🔑	int(11)			Não	Nenhum	

Figura 65 base de dados 10

Tabela loja_componente:

	#	Nome	Tipo	Agrupamento (Collation)	Atributos	Nulo	Predefinido	Comentários
<input type="checkbox"/>	1	id 🔑	int(11)			Não	Nenhum	
<input type="checkbox"/>	2	loja_id 🔑	int(11)			Sim	NULL	
<input type="checkbox"/>	3	componente_id 🔑	int(11)			Sim	NULL	

Figura 66 base de dados 11

Estas tabelas são componentes essenciais de uma aplicação que envolve lojas, produtos, projetos, perguntas, respostas e gestão de utilizadores. Cada tabela desempenha um papel específico na organização e armazenamento de dados relacionados a esses elementos, permitindo que a aplicação funcione de forma eficaz.

Modelo Relacional

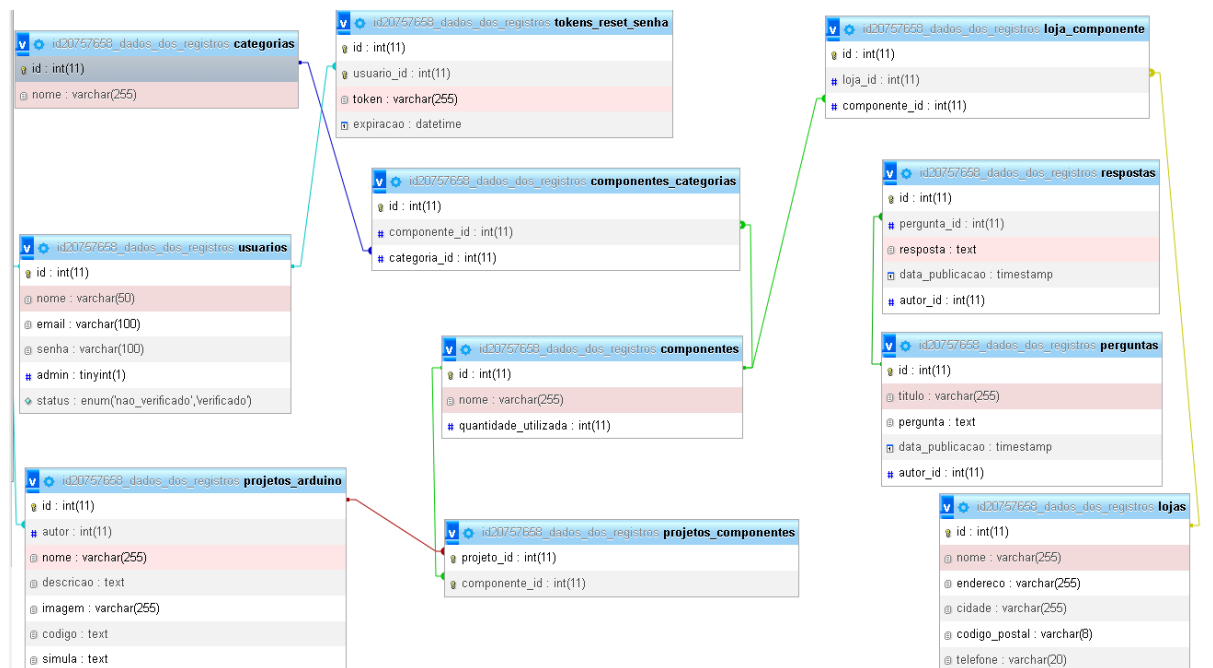


Figura 67 base de datos 12

Resultados do Desenvolvimento

Embora o site ainda não tenha sido publicado, a fase de desenvolvimento proporcionou uma visão aprofundada das potencialidades e desafios da plataforma.

Testes ao Produto Desenvolvido:

Apesar de não estar disponível publicamente, foram realizados testes internos para avaliar a funcionalidade do sistema em ambiente controlado. Esses testes identificaram áreas de melhoria, destacando a importância de refinamentos na interface do utilizador e na eficácia das ferramentas de criação de projetos.

Constrangimentos Identificados:

A análise revelou constrangimentos, especialmente relacionados à interface do utilizador e à documentação de projetos. A ausência de feedback imediato e a necessidade de aprimoramentos na navegação surgiram como aspetos críticos a serem endereçados.

Sucessos Potenciais:

Mesmo antes da publicação, o projeto revelou sucessos conceituais, como a estrutura robusta da plataforma e a implementação eficiente de funcionalidades como o fórum de discussão. O potencial para criar uma comunidade envolvente permanece evidente.

Desenvolvimentos Futuros Antecipados:

Com base nos resultados dos testes, futuros desenvolvimentos focarão em melhorias significativas na interface do utilizador para garantir uma experiência mais intuitiva. Também se planeia a implementação de recursos adicionais, como tutoriais interativos e elementos de gamificação para incentivar a participação dos utilizadores.

Contribuição para o Desenvolvimento Pessoal e Profissional:

Apesar da não publicação, a experiência de desenvolvimento contribuiu para a evolução de competências em gestão de projetos, design de interfaces e interação com a comunidade. A resolução de desafios e a adaptação constante representaram uma valiosa oportunidade de crescimento técnico e interpessoal.

Conclusão

Penso que o desenvolvimento do projeto correu como eu estava à espera, com vários erros que foram resolvidos, várias tentativas, e, como principal objetivo, que o projeto funcione!

As principais dificuldades que senti foram, ao início, as formatações da página para ficarem com o aspeto que tem hoje, com vários erros, desformatações e 'bugs'.

Para superar estas dificuldades eu pesquisei, estudei, testei persistentemente atar corrigir o erro. Também graças a todas as ajudas das pessoas que coloquei nos agradecimentos, sejam de me ajudar a desenvolver o código, seja apoio moral e dar confiança para não desistir.

Com este trabalho, fui capaz de aprender bastante mais como funciona a criação de páginas 'web' dinâmicas, ligação remota a uma base de dados e também a manipular os dados e atributos via formulários da página.

Assim como referi na introdução, eu escolhi este tema, pois gosto muito de ver e criar projetos com o Arduino.

No futuro disponho-me a adicionar ainda mais funcionalidades ao projeto.

Comecei a perceber um pouco melhor também que só conseguimos concretizar todos os nossos objetivos após um árduo trabalho, daí ter-me dedicado bastante ao desenvolvimento da minha PAP. Espero que, com este trabalho, tenha atingido o objetivo que o curso me exigia.

Autoavaliação

Principais Dificuldades e Obstáculos Encontrados:

Complexidade Tecnológica: A complexidade inerente ao desenvolvimento web, especialmente em termos de front-end e back-end, apresentou desafios significativos. O manuseio de tecnologias específicas e a integração eficiente foram áreas que demandaram atenção especial.

Feedback Limitado durante o Desenvolvimento: A ausência de feedback externo durante as fases iniciais do desenvolvimento, devido à não publicação, dificultou a validação de decisões técnicas e de design.

Formas de Superar as Dificuldades:

Estudo Profundo de Tecnologias: Encarei a complexidade tecnológica através de um estudo profundo das tecnologias utilizadas. Recorri a documentação oficial, tutoriais e fóruns para superar obstáculos específicos e otimizar a implementação.

Testes e Simulações Locais: A falta de feedback externo foi contornada através da realização de testes e simulações locais intensivos. Isso permitiu identificar potenciais falhas e otimizar o desempenho antes da exposição ao público.

Aprendizados e Crescimento Pessoal e Profissional:

Aprimoramento Técnico: Enfrentar desafios técnicos significativos resultou em um aprimoramento notável das minhas habilidades como programador de sistemas informáticos. A resolução de problemas complexos consolidou meu conhecimento prático.

Gestão Eficiente de Projeto: A experiência contribuiu para o desenvolvimento de competências em gestão de projetos, especialmente na identificação proativa de obstáculos e na implementação de soluções eficazes.

Avaliação Global da Execução:

Pontos Fortes: A abordagem de aprendizado contínuo permitiu superar desafios técnicos, resultando em uma arquitetura sólida para a plataforma. A autocrítica constante identificou áreas de melhoria, demonstrando uma postura de aprimoramento contínuo.

Áreas de Melhoria: A ausência de feedback externo realçou a necessidade de estratégias alternativas para validar escolhas de design e funcionalidades. A otimização da interface do utilizador continua a ser uma prioridade.

Planos para o Futuro:

Iteração Contínua: Planeio continuar a iterar sobre a plataforma, incorporando feedback da comunidade quando o site for publicado. A evolução constante das funcionalidades e da experiência do utilizador será uma prioridade.

Partilha e Colaboração: Explorar estratégias para envolver a comunidade desde as fases iniciais, partilhando insights e recebendo feedback em tempo real.

Webgrafia

Insiro, para finalizar, as respetivas hiperligações de copyright das imagens e a data que foram pesquisadas, e também, alguns sites de pesquisa.

<http://www.w3schools.com> (Acedido em 3 de Agosto de 2023)

<http://developer.mozilla.org> (Acedido em 3 de Agosto de 2023)

<http://www.php.net> (Acedido em 3 de Agosto de 2023)

<http://stackoverflow.com> (Acedido em 3 de Agosto de 2023)

<http://github.com> (Acedido em 3 de Agosto de 2023)

<http://www.codecademy.com> Acedido em 3 de Agosto de 2023)

<http://www.coursera.org> (Acedido em 5 de Agosto de 2023)

<http://www.udemy.com> Acedido em 5 de Agosto de 2023)

<http://www.sitepoint.com> Acedido em 5 de Agosto de 2023)

<http://tutsplus.com> (Acedido em 9 de Agosto de 2023)

<http://codepen.io> (Acedido em 9 de Agosto de 2023)

<http://www.smashingmagazine.com> (Acedido em 9 de Agosto de 2023)

<http://css-tricks.com> (Acedido em 9 de Agosto de 2023)

<http://alistapart.com> (Acedido em 15 de Agosto de 2023)

<http://dev.to> (Acedido em 15 de Agosto de 2023)

<http://www.webfx.com>(Acedido em 15 de Agosto de 2023)

<http://codersrank.io> (Acedido em 15 de Agosto de 2023)

<http://www.hackerrank.com/domains/tutorials/10-days-of-javascript> (Acedido em 15 de Agosto de 2023)

<http://www.hackerearth.com/practice/> (Acedido em 15 de Agosto de 2023)

<http://leetcode.com> (Acedido em 20 de Agosto de 2023)

<http://netninja.dev> (Acedido em 20 de Agosto de 2023)

<http://thenewboston.com> (Acedido em 20 de Agosto de 2023)

<http://www.geeksforgeeks.org> (Acedido em 20 de Agosto de 2023)

<http://www.phpclasses.org> (Acedido em 20 de Agosto de 2023)

<http://www.fullstackpython.com> (Acedido em 20 de Agosto de 2023)

<http://phptherightway.com> (Acedido em 24 de Agosto de 2023)

<http://phpdelusions.net> (Acedido em 24 de Agosto de 2023)

<http://laracasts.com> (Acedido em 24 de Agosto de 2023)

<http://laravel.com> (Acedido em 24 de Agosto de 2023)

<http://symfony.com> (Acedido em 24 de Agosto de 2023)

<http://framework.zend.com> (Acedido em 24 de Agosto de 2023)

<http://codeigniter.com> (Acedido em 29 de Agosto de 2023)

<http://cakephp.org> (Acedido em 29 de Agosto de 2023)

<http://www.yiiframework.com> (Acedido em 29 de Agosto de 2023)

<http://phalcon.io> (Acedido em 29 de Agosto de 2023)

<http://frontendmasters.com> (Acedido em 3 de setembro de 2023)

<http://www.freecodecamp.org> (Acedido em 3 de setembro de 2023)

<http://roadmap.sh> (Acedido em 3 de setembro de 2023)

<http://www.codewall.co.uk> (Acedido em 3 de setembro de 2023)

<http://www.echojs.com> (Acedido em 3 de setembro de 2023)

<http://www.honeypot.io> (Acedido em 10 de setembro de 2023)

<http://tutorials.jenkov.com> (Acedido em 10 de setembro de 2023)

<http://guides.rubyonrails.org> (Acedido em 10 de setembro de 2023)

<http://emberjs.com> (Acedido em 10 de setembro de 2023)

<http://vuejs.org> (Acedido em 14 de setembro de 2023)

<http://reactjs.org> (Acedido em 14 de setembro de 2023)

<http://angular.io> (Acedido em 14 de setembro de 2023)

<http://jquery.com> (Acedido em 14 de setembro de 2023)

<http://www.djangoproject.com> (Acedido em 14 de setembro de 2023)

<http://flask.pocoo.org> (Acedido em 19 de setembro de 2023)

<http://www.arduino.cc> (Acedido em 19 de setembro de 2023)

<http://playground.arduino.cc> (Acedido em 19 de setembro de 2023)

<http://forum.arduino.cc> (Acedido em 19 de setembro de 2023)

<http://www.arduino.org> (Acedido em 19 de setembro de 2023)

<http://www.adafruit.com> (Acedido em 23 de setembro de 2023)

<http://www.sparkfun.com> (Acedido em 23 de setembro de 2023)

<http://learn.adafruit.com> (Acedido em 23 de setembro de 2023)

<http://learn.sparkfun.com> (Acedido em 23 de setembro de 2023)

<http://arduino.stackexchange.com> (Acedido em 23 de setembro de 2023)

<http://www.electronicwings.com> (Acedido em 23 de setembro de 2023)

<http://create.arduino.cc> (Acedido em 28 de setembro de 2023)

<http://www.instructables.com> (Acedido em 28 de setembro de 2023)

<http://randomnerdtutorials.com> (Acedido em 28 de setembro de 2023)

<http://www.arduinoocia.com.br> (Acedido em 28 de setembro de 2023)

<http://howtomechatronics.com> (Acedido em 28 de setembro de 2023)

<http://www.makerguides.com> (Acedido em 28 de setembro de 2023)

<http://electronicsforu.com> (Acedido em 28 de setembro de 2023)

<http://arduinogetstarted.com> (Acedido em 30 de setembro de 2023)

<http://circuitdigest.com> (Acedido em 30 de setembro de 2023)

<http://create.arduino.cc/projecthub> (Acedido em 30 de setembro de 2023)

<http://www.electronicwings.com> (Acedido em 30 de setembro de 2023)

<http://lastminuteengineers.com> (Acedido em 30 de setembro de 2023)

<http://randomnerdtutorials.com> (Acedido em 5 de outubro de 2023)

<http://www.electronicclinic.com> (Acedido em 5 de outubro de 2023)

<http://mechatronics4you.com> (Acedido em 5 de outubro de 2023)

<http://circuitdigest.com> (Acedido em 5 de outubro de 2023)

<http://www.arduino-for-beginners.com> (Acedido em 5 de outubro de 2023)

<http://create.arduino.cc/get-started> (Acedido em 5 de outubro de 2023)

<http://www.electronicwings.com> (Acedido em 5 de outubro de 2023)

<http://randomnerdtutorials.com/getting-started> (Acedido em 5 de outubro de 2023)

<http://www.instructables.com/technology/arduino> (Acedido em 11 de outubro de 2023)

<http://www.instructables.com/technology/arduino> (Acedido em 11 de outubro de 2023)

<http://learn.sparkfun.com> (Acedido em 11 de outubro de 2023)

<http://learn.adafruit.com/category/learn-arduino> (Acedido em 11 de outubro de 2023)

<http://arduino.stackexchange.com> (Acedido em 11 de outubro de 2023)

<http://www.arduino-tutorial.net> (Acedido em 11 de outubro de 2023)

<http://electronicsforu.com/electronics-projects> (Acedido em 11 de outubro de 2023)

<http://create.arduino.cc/projecthub/arduino> (Acedido em 17 de outubro de 2023)

<http://create.arduino.cc/projecthub/arduino> (Acedido em 17 de outubro de 2023)

<http://howtomechatronics.com/category/arduino> (Acedido em 17 de outubro de 2023)

<http://howtomechatronics.com/category/arduino> (Acedido em 17 de outubro de 2023)

<http://www.makerguides.com> (Acedido em 17 de outubro de 2023)

<http://www.makerguides.com> (Acedido em 17 de outubro de 2023)

<http://electronicsforu.com/electronics-projects> (Acedido em 21 de outubro de 2023)

<http://www.electronicclinic.com> (Acedido em 21 de outubro de 2023)

<http://mechatronics4you.com> (Acedido em 21 de outubro de 2023)

<http://arduinogetstarted.com> (Acedido em 21 de outubro de 2023)

<http://create.arduino.cc/projecthub/electronicwings> (Acedido em 21 de outubro de 2023)

<http://circuitdigest.com> (Acedido em 21 de outubro de 2023)