

Arquivo cliIncluir.php

The screenshot shows a web browser window with the address bar displaying 'https://localhost/cliIncluir.php'. The page title is 'Novo Cliente'. The form contains the following fields:

- CPF (somente números): - Nome: - Endereço: - Complemento: - Bairro: - Estado: - Cidade: - País: - e-Mail: - Senha: - Repetir a Senha:

At the bottom right, there are two buttons: 'Salvar' and 'Voltar'.

Figura 1: Formulário de Cliente

2 `error_reporting(E_ALL);`

3 `ini_set("display_errors", 1);` comandos para que o servidor PHP exiba no

navegador durante a execução, todo os tipos de erros que possam ter no código.

5 `$conexao = conecta();` chama a função conecta que retorna a conexão com o banco

de dados. A função conecta fica no arquivo `funcoes.php`.

6 `// Variáveis dos campos do formulário` `//` comentário de uma linha no

código do programa. `/*` comentário de mais de uma linha `*/`

7 `$cli_cpf = "";` forma de iniciar ou declarar uma variável sem conteúdo.

18 `if (isset($_POST["cli_cpf"])) {` testa se foi recebida através do método

POST uma variável com o nome `cli_cpf`. Na primeira vez que a página é carregada a condição do `if` não será satisfeita o que faz com que o formulário do Cliente seja carregado com todos os campos vazios para preenchimento pelo usuário. Depois que os dados forem preenchidos, e confirmados pelo usuário ao clicar no botão Salvar, essa variável estará definida e a condição do `if` será satisfeita e os comandos internos ao `if` serão executados.

Comandos internos ao if da linha 18

19 `$cli_cpf = $_POST["cli_cpf"];` atribui a variável `$cli_cpf` o que foi enviado

pelo POST do campo do formulário com o mesmo nome `cli_cpf`. O mesmo acontece com os outros campos de nome a email e senha. Com a senha

28 `$cli_senha = md5($_POST["cli_senha"]);` a senha digitada pelo usuário é criptografada pela função `md5()` para armazenamento no banco de dados.

29 `if (achaCliente($cli_cpf)) {` testa o retorno da função `achaCliente()`

passando o CPF, variável `$cli_cpf` como parâmetro.

Se o cliente já existir, se a função `achaCliente()` retornar `true`, verdade, exibe

30 `echo "<script>alert('Cliente já existe');</script>";` faz o php chamar a função `javascript alert()` que mostra uma caixa de mensagem *modal* (que exige uma ação para fechar a mensagem antes de fazer qualquer outra coisa no sistema).

31 `} else {` se o cliente não existir

```
32 if ($result = $conexao->query("INSERT INTO cliente (cli_cpf,
                                cli_nome,
                                cli_end_lograd,
                                cli_end_compl,
                                cli_end_bairro,
                                cli_end_cidade,
                                cli_end_uf,
                                cli_end_pais,
                                cli_login,
                                cli_senha)
                                VALUES ('.$cli_cpf.',
                                '.$cli_nome.',
                                '.$cli_end_lograd.',
                                '.$cli_end_compl.',
                                '.$cli_end_bairro.',
                                '.$cli_end_cidade.',
                                '.$cli_end_uf.',
                                '.$cli_end_pais.',
                                '.$cli_email.',
                                '.$cli_senha.'))");
```

Teste se o comando `$conexao->query("INSERT INTO ...VALUES ...")` que faz a inclusão no banco de dados foi realizada com sucesso.

Se a inclusão foi feita, libera a memória da variável `$result`

52 `mysqli_free_result($result);`

senão 53 `} else {` se inclusão não foi feita

Exibe a mensagem avisando do erro

54 `echo "<script>alert('Erro na inclusão do Cliente');</script>";`

57 `header("location:cliListar.php");` navega para a página inicial da listagem dos clientes.

Além do que já foi descrito em `cliListar.php` temos a função javascript `Valida()` { ... } que como exemplo é usada para validar o CPF e conferir se a senha e a repetição da senha digitadas pelo usuário conferem.

68 `function Valida() {` abertura da função

69 `var conf = true;` inicia uma variável com o valor `true`, ou verdadeiro.

70 `var msg = "";` variável que será usada para exibir a mensagem do aviso caso algum dado não passe na validação.

71 `with (document.fcliIncluir){` indica que todas as referências aos elementos HTML serão daqueles que se encontram dentro do `document` e no form `fcliIncluir`.

72 `// Valida tudo e submit` comentário de uma linha javascript, igual ao PHP, o javascript também tem comentários `/*` com várias linhas `*/`

73 `conf = confereCPF(cli_cpf.value);` atribui a variável `conf` o retorno da função javascript `confereCPF()` com o parâmetro `cli_cpf.value`. Aqui a referência completa ao elemento HTML `cli_cpf`, seria `document.fcliIncluir.cli_cpf.value`, mas, como vimos acima na linha 71, podemos omitir `document.fcliIncluir`. A função `confereCPF()` está em `funcoes.js`.

74 `if ((conf == true) && (cli_nome.value === "")) {...}` se a variável `conf` tiver passado no teste do CPF, se a função `confereCPF()` tiver retornado `true` e `&&` se o campo `cli_nome` estiver vazio faz a atribuições 75 `msg = "Nome necessita ser informado";` e 76 `conf = false;` em seguida testa `conf` novamente e se as senhas digitadas não são iguais com 78 `if ((conf == true) && (!confereSenhas(cli_senha.value, cli_senha_repetida.value))) {` o caractere de `!` Inverte o valor retornado da função `confereSenhas()`.

Se o CPF estiver correto, se o nome tiver sido preenchido, e agora se as senhas não conferirem, limpa os campos de senhas em 79 `cli_senha.value = "";` e 80 `cli_senha_repetida.value = "";` e atribui *false* a `conf` 81 `conf = false;` Se os dados passaram em todos os testes `conf` estará com *true*.

Se 83 `if (conf == true) {` envia o formulário com os dados do cliente pelo método POST para o próprio arquivo `cliIncluir.php`.

Senão 85 `} else {`, exibe a mensagem do erro encontrado chamando a função `Alerta()` 86 `Alerta("Erro", msg);`

93 `<h2>Novo Cliente</h2>` cabeçalho da página.

94 `<form name="fcliIncluir" action="<?php echo $_SERVER['PHP_SELF']; ?>" method="POST">` tag inicial do form, formulário dos dados do cliente, nome `name="fcliIncluir"` ação `action="<?php echo $_SERVER['PHP_SELF']; ?>"` usa a constante do servidor que contém o próprio nome do arquivo atual da página como destino do envio dos dados do cliente e o método POST.

Primeiro campo do formulário, CPF

95 CPF (somente números): `<input type="text" name="cli_cpf" size="15" maxlength="11" placeholder="Qual seu CPF?" value="<?php echo $cli_cpf ?>" required>`

O texto que aparece antes do tag `<input>` é o que é exibido na tela. O tipo do campo `type="text"` embora CPF só tenha números esse campo tem que permitir a exibição do CPF com a máscara de pontos e hífen, como 123.456.789-99, o que não seria possível num campo numérico. O que se pode melhorar é que a cada tecla digitada, que se bloqueie a digitação de qualquer caractere que não seja número. Nome do campo `name="cli_cpf"`, igual ao nome do campo da tabela cliente. Tamanho na tela `size="15"` e quantidade de caracteres aceitos `maxlength="11"` texto que aparece no fundo campo `placeholder="Qual seu CPF?"` valor inicial do campo, na inclusão é vazio, `value="<?php echo $cli_cpf ?>"` definido pelo conteúdo da variável PHP `$cli_cpf` e por último o campo será exibido como obrigatório `required>` e o final do tag form.

97 Nome: `<input type="text" name="cli_nome" size="100" maxlength="100" placeholder="Qual seu nome?" value="<?php echo $cli_nome ?>" required>`

Campo Nome. ` ` representa o equivalente a um espaço em branco na tela.

99 e 100 `
` insere uma quebra de linha.

107 Estado: `<select name="cli_end_uf" id="cli_end_uf">`
`<option value="AC">AC</option>`
`<option value="AL">AL</option>`
`<option value="AP">AP</option>`
`<option value="AM">AM</option>`
`<option value="BA">BA</option>`
`<option value="CE">CE</option>`
`<option value="DF">DF</option>`
`<option value="ES">ES</option>`
`<option value="GO">GO</option>`
`<option value="MA">MA</option>`
`<option value="MT">MT</option>`
`<option value="MS">MS</option>`
`<option value="MG">MG</option>`
`<option value="PA">PA</option>`
`<option value="PB">PB</option>`
`<option value="PR">PR</option>`
`<option value="PE">PE</option>`
`<option value="PI">PI</option>`

```

<option value="RJ" selected>RJ</option>
<option value="RN">RN</option>
<option value="RS">RS</option>
<option value="RO">RO</option>
<option value="RR">RR</option>
<option value="SC">SC</option>
<option value="SP">SP</option>
<option value="SE">SE</option>
<option value="TO">TO</option>
</select>

```

Exibe uma lista de seleção da UF do endereço do cliente. A opção RJ, aparece selecionada inicialmente

```
<option value="RJ" selected>RJ</option>
```

Figura 2: Lista de Seleção de UFs

148 `<input type="button" class="button" value="Salvar" onClick="Valida();" >` tipo

button, recebe a formatação definida no arquivo estilo.css através da classe button `class="button"`

value define o texto do botão, e define o comportamento quando o botão é clicado

`onClick="Valida();" >` chama a função javascript `Valida()` e faz o fechamento do botão.

```

149 <input type="button" class="button" value="Voltar"
onClick="window.location.href='cliListar.php';">

```

151, 152 e 153 `</form>`, `</body>` e `</html>` faz os fechamentos.

Referência

Manual online de PHP Disponível em https://www.php.net/manual/pt_BR/index.php Acesso: Ago, 2020.

W3C PHP. Disponível em <https://www.w3schools.com/php/default.asp> Acesso: Ago, 2020.