

Arquivo funcoes.php

Declaração de função 2 `function conecta() {`

3 `$conexao = new mysqli("127.0.0.1", "root", "", "progweb");` cria o objeto

`$conexão` usando o endereço IP do servidor, o usuário, sem senha e o banco de dados.

4 `if ($conexao->connect_errno) {` se retorna o código de erro do último erro da função `connect`. Exibe a mensagem de erro de `connect_error`

5 `die("Falha na conexão com o Banco de Dados: " . $conexao->connect_error);`

6 `exit();` termina o script atual.

Se tiver sucesso na conexão 7 `} else {`, retorna a conexão `return 9 $conexao;`

Função formata CPF 12 `function formataCPF($inCPF) {` `$inCPF` é o CPF sem pontos nem hífen, retorna `$outCPF` com pontos separadores e hífen. `$ouCPF` inicialmente vazio, recebe com 13 `$outCPF = substr($inCPF, 0, 3).".";` os primeiros três dígitos de `$inCPF` e um ponto. Em seguida os próximos três dígitos e mais um ponto 14 `$outCPF .= substr($inCPF, 3, 3).".";` Os próximos três dígitos e o hífen 15 `$outCPF .= substr($inCPF, 6, 3)."-";` e por último os dois dígitos verificadores 16 `$outCPF .= substr($inCPF, 9, 2);`. Retorna `$outCPF` sem nenhum espaços em branco 17 `return trim($outCPF);`.

Função que faz a busca do CPF na tabela cliente 19 `function achaCliente($inCPF) {`.

Declara a variável global `$conexão`, a mesma da função `conecta()` acima. Faz a consulta

21 `if ($result = $conexao->query("SELECT * FROM cliente WHERE cli_cpf = '" . $inCPF . "'")) {`

Se encontrou 22 `if ($result->num_rows > 0) {` retorna verdadeiro `true` 24 `return true;`,

senão 26 `return false;`. Libera a memória com 29 `$result->close();`.

Arquivo funcoes.js

Função 1 `function validaCPF(inCPF) { ... }`. Declara variável `cpf` e limpa tudo o que não é número. Se o *string* texto CPF tiver um comprimento `cpf.length` ou número de caracteres

diferente `!=` de 11 `5` `if (cpf.length != 11) ||` ou se a sequência de 11 caracteres for toda composta com 0, com 1 até 9 retorna falso *false*.

Essa função calcula os dois dígitos verificadores do CPF e confere com os dígitos informados para conferir qualquer erro de consistência da informação fornecida.

O operador `%` em `21` `rev = 11 - (add % 11);` retorna o resto da divisão de `add` por 11. A função `parseInt()` em `24` `if (rev != parseInt(cpf.charAt(9)))` transforma o caractere em número para poder realizar o cálculo.

Arquivo `estilo.css`

Define um margem de 30 pixels `2` `margin: 30px;` nos quatro lados em volta do tag HTML `body` `1` `body {` e `4` `}`.

Define a cor do fundo das células `th` `4` `th {` `6` `}` da tabela `table` HTML `5` `background-color: rgb(200,200,200);`, usa a função `rgb()` que tem o nome das iniciais em inglês para vermelho *red*, verde *green* e azul *blue* que compõem a cor usada. Os valores vão de 0 a 255, 0 = ausência de cor, ou preto, e 255 o branco. Quando os três valores são iguais 0, 0, 0 = totalmente preto e 255, 255, 255 = branco, quaisquer outros três valores iguais definem uma escala de cinza.

Definição da linha da borda das células `th` e `td` `7` `th, td {` `8` `border: 1px solid gray;` define a borde com 1 pixel de espessura, com linha contínua e cinza.

Retira as margens `11` `margin: 0;` e espaçamento `12` `padding: 0;` interno da imagem `10` `img`.

Define as características da classe `button` `14` `.button {` que é usada para definir o elemento HTML `button`. Define separadamente as margens esquerda e direita

`15` `margin-left: 10px;` e `16` `margin-right: 10px;` e os espaçamentos esquerdo e direito

`17` `padding-left: 20px;` e `18` `padding-right: 20px;`. As bordas esquerda e direita

`19` `border-left: 4px solid rgb(200,200,200);` e

`20` `border-right: 4px solid rgb(200,200,200);` o tamanho da letra, fonte, *font*

21 `font-size: 14pt;` e a cor de fundo 22 `background-color: rgb(200,200,200);`

Arquivo `index.php`

2 `$opcao = "cliListar.php";` defina a opção padrão para a página inicial. Se
3 `if (isset($_GET["opcao"])) {` for passada, usa a opção recebida.

Estilo para definir as duas áreas da tela dividida verticalmente

```
13 #menu {width: 14%;  
    height: 99vh;  
    background: rgb(210,110,90);  
    margin: 0;  
    float: left;}  
#pagina {width: 86%;  
    height: 99vh;  
    background: rgb(255,220,210);  
    float: left;}
```

Carrega o arquivo `menu.htm` do lado esquerdo

```
26 <iframe src="menu.htm" width="100%" height="100%" style="min-height: 98vh;"  
frameborder="0">  
</iframe>
```

e a opção do lado direito

```
31 echo "<iframe src='".$opcao."' width='100%' height='100%' style='min-height: 98vh;'  
frameborder='0'></iframe>";
```

Arquivo `menu.htm`

Estilo que define o visual do botão e dos seus links (tag `a`)

```
8 .botao {width: 80%;  
    background-color: rgb(190,90,70);  
    margin: 10px;}  
a {font-size: 16pt;  
    color: white;  
    text-decoration: none;  
    padding-left: 20px;}  
body {color: white;}
```

Cada link recarrega a página inicial mudando o conteúdo do lado direito da tela.

22 `Venda`

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.