



Sumário

Descrição do Projeto	3
Funcionalidades do CRUD:	3
Tecnologias Utilizadas:	3
Criando o banco de dados	4
Criando conexão com o banco de dados	4
Criando a tela index	5
Criando função usuário	7
Criando tela de login	8
Criando formulário de cadastro	8
Criando a tela principal do sistema	11
Criando tela de editar cadastro	13
Criando função de excluir cadastro	15
Criando função de sair do sistema	16

Descrição do Projeto

O projeto consiste em desenvolver um sistema de gerenciamento de dados utilizando a linguagem de programação PHP no backend e MySQL como banco de dados, enquanto HTML, CSS e Javascript serão utilizados para a criação do frontend.

Funcionalidades do CRUD:

1. **Create (Criar):** Os usuários poderão adicionar novos registros ao banco de dados através de um formulário HTML. O PHP receberá os dados enviados pelo formulário, validará e inserirá no banco de dados usando consultas SQL.
2. **Read (Ler):** Os dados existentes no banco de dados serão exibidos em uma interface HTML, permitindo que os usuários visualizem as informações de maneira organizada e amigável. O PHP será responsável por recuperar os dados do banco de dados e enviá-los para o frontend.
3. **Update (Atualizar):** Os usuários poderão editar os registros existentes no banco de dados. Isso será feito através de um formulário HTML pré-preenchido com os dados do registro selecionado. O PHP receberá as alterações feitas pelo usuário, validará e atualizará os dados no banco de dados.
4. **Delete (Excluir):** Os registros existentes poderão ser removidos do banco de dados pelos usuários. Essa ação será realizada através de botões ou links na interface HTML. O PHP será responsável por receber a solicitação de exclusão e executar a operação no banco de dados.

Tecnologias Utilizadas:

1. **PHP:** Será usado para criar as operações de CRUD (Create, Read, Update, Delete) e interagir com o banco de dados MySQL.
2. **MySQL:** Servirá como o banco de dados para armazenar os dados do sistema.
3. **HTML:** Utilizado para criar a estrutura da página web e os formulários de entrada de dados.
4. **CSS:** Responsável pela estilização e aparência visual do frontend, tornando a interface do usuário mais atraente e fácil de usar.
5. **JavaScript:** Utilizado para adicionar interatividade à página web, como validação de formulários, confirmações de exclusão e outras funcionalidades dinâmicas.

Criando o banco de dados

Use o comando a baixo para criar um novo banco de dados:

```
1 create database projetophp;
```

Agora vamos criar a tabela cadastro:

```
1 CREATE TABLE IF NOT EXISTS projetophp.cadastro (  
2     id int(11) NOT NULL AUTO_INCREMENT,  
3     nome varchar(90) NOT NULL,  
4     email varchar(90) NOT NULL,  
5     senha varchar(90) NOT NULL,  
6     sexo varchar(90) NOT NULL,  
7     cidade varchar(90) NOT NULL,  
8     PRIMARY KEY (id)  
9 );
```

Usando o comando **insert into** insira os registros abaixo:

```
1 INSERT INTO projetophp.cadastro (id, nome, email, senha, sexo, cidade) VALUES  
2 (3, 'Judson Martins', 'judson@versatilti.com.br', '123', 'masculino', 'Brasília/DF'),  
3 (4, 'Cibele Santana', 'cibele@gmail.com', '123', 'feminino', 'Taguatinga/DF'),  
4 (5, 'Kauan Barros', 'kauan@versatilti.com.br', '123', 'masculino', 'Sobradinho/DF'),  
5 (6, 'Kamila Sales', 'kamila@versatilti.com.br', '123', 'feminino', 'Brasília/DF'),  
6 (7, 'Giovanna Santana', 'gi@versatilti.com.br', '123', 'feminino', 'Gama/DF');
```

Verifique se os registros foram inseridos corretamente utilizando o comando **select**:

```
1 SELECT * FROM projetophp.cadastro;
```

Criando conexão com o banco de dados

1. Crie um arquivo com o nome **conexao.php**:

```
1 <?php  
2  
3 $dbHost = 'localhost';  
4 $dbUsername = 'root';  
5 $dbPassword = '';  
6 $dbName = 'projetophp';  
7  
8 $conexao = new mysqli($dbHost,$dbUsername,$dbPassword,$dbName);  
9  
10 ?>
```


Criando a tela index

A tela **index.php** é a página inicial da aplicação web, onde os usuários serão direcionados ao acessar o sistema. Esta página normalmente apresentará um resumo ou uma visão geral dos dados disponíveis no sistema, juntamente com opções de navegação e ações que os usuários podem realizar.

1. Crie uma pasta exclusivamente para o projeto.

Exemplo:  projetophp

2. Crie uma pasta chamada css. Nessa pasta iremos guarda os arquivos de estilização do projeto.

Exemplo:  css

3. Crie o arquivo **index.php** dentro da pasta que você criou para o projeto.

Index.php

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Projeto PHP Versátil Ti</title>
  <link rel="stylesheet" type="text/css" href="css/index.css">
</head>
<body>

  <div class="box">
    <h1>LOGIN DE ACESSO</h1>
    <br>
    <a href="login.php">Login</a>
    <a href="formulario.php">Cadastre-se</a>
  </div>
</body>
</html>
```

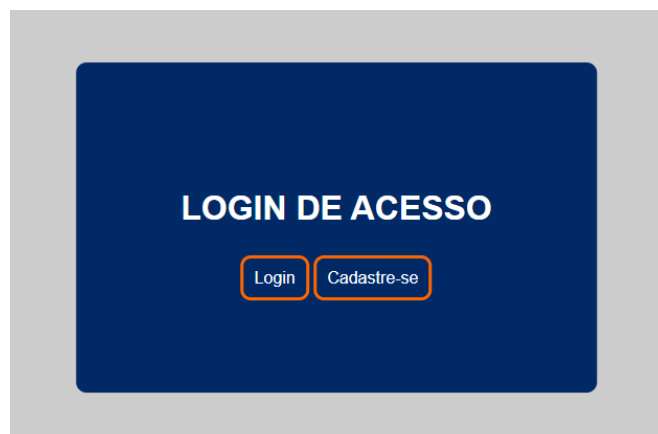
4. Agora vamos fazer a estilização criando um novo arquivo chamado **login.css** na pasta css:

login.css

```

1  body{
2      font-family: Arial, Helvetica, sans-serif;
3      background-color: #e0e0e0;
4  }
5  div{
6      background-color: #002966;
7      position: absolute;
8      top: 50%;
9      left: 50%;
10     transform: translate(-50%,-50%);
11     padding: 60px;
12     border-radius: 15px;
13     color: #fff;
14 }
15 input{
16     padding: 10px;
17     width: 300px;
18     border: none;
19     outline: none;
20     font-size: 15px;
21 }
22 .inputSubmit{
23     background-color: #0070c0;
24     border: none;
25     padding: 15px;
26     width: 100%;
27     border-radius: 10px;
28     color: #fff;
29     font-size: 15px;
30 }
31 .inputSubmit:hover{
32     background-color: #ff6600;
33     cursor: pointer;
34 }
35 .button {
36     font-family:Arial, Helvetica, sans-serif;
37     font-size:14px;
38     border-radius:4px;
39     padding-left:20px;
40     padding-right:20px;
41     padding-top:5px;
42     padding-bottom:5px;
43     color: #ffffff;
44     background-color: #0070c0;
45     outline:none;
46     border:none;
47     cursor:pointer;
48     display:inline-block;
49     text-decoration: none;
50     transition: .4s;
51 }
52 .button:hover {
53     color: #ffffff;
54     background-color: #ff6600;
55 }
56 
```

5. Agora rode o arquivo e verifique se ele está semelhante a imagem abaixo:



Criando função usuário

As funções de usuário são essenciais para a autenticação de usuários no sistema.

1. Crie o arquivo **usuario.php** dentro da pasta que você criou para o projeto.

usuario.php

```
1 <?php
2 session_start();
3 if(isset($_POST['submit']) && !empty($_POST['email']) && !empty($_POST['senha']))
4 {
5
6     include_once('conexao.php');
7     $email = $_POST['email'];
8     $senha = $_POST['senha'];
9
10    $sql = "SELECT * FROM cadastro WHERE email = '$email' and senha = '$senha'";
11
12    $result = $conexao->query($sql);
13
14    if(mysqli_num_rows($result) < 1)
15    {
16        unset($_SESSION['email']);
17        unset($_SESSION['senha']);
18        header("Location: login.php");
19    }
20    else
21    {
22        $_SESSION['email'] = $email;
23        $_SESSION['senha'] = $senha;
24        header("Location: sistema.php");
25    }
26 }
27 else
28 {
29     header("Location: login.php");
30 }
31 ?>
```

Criando tela de login

A tela de login do sistema é a interface pela qual os usuários podem acessar o sistema, fornecendo credenciais de autenticação, como nome de usuário e senha.

1. Crie o arquivo **login.php** dentro da pasta que você criou para o projeto.

login.php

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Projeto PHP Versátil Ti</title>
8   <link rel="stylesheet" type="text/css" href="css/login.css">
9 </head>
10
11 <body>
12   <div>
13     <h1>login</h1>
14     <form action="usuario.php" method="POST">
15
16       <input type="text" name="email" placeholder="Email">
17       <br><br>
18       <input type="password" name="senha" placeholder="Senha">
19       <br><br>
20
21       <input class="inputSubmit" type="submit" name="submit" value="Acessar">
22       <br><br>
23       <a href="index.php" class="button">Voltar</a>
24     </form>
25   </div>
26 </body>
27 </html>
```

Criando formulário de cadastro

A tela **formulario.php** neste projeto seria responsável por fornecer um formulário de entrada de dados para permitir que os usuários adicionem novos registros ao banco de dados.

2. Crie um arquivo com o nome **formulario.php**:

formulario.php

```
1 <?php
2
3   if(isset($_POST['submit']))
4   {
5       include_once('conexao');
6
7       $nome = $_POST['nome'];
8       $email = $_POST['email'];
9       $senha = $_POST['senha'];
10      $sexo = $_POST['sexo'];
11      $cidade = $_POST['cidade'];
12
13      $result = mysqli_query($conexao, "INSERT INTO cadastro(nome,email,senha,sexo,cidade)VALUES ('$nome','$email','$senha','$sexo','$cidade')");
14      header('Location: login.php');
15  }
16
17 >
18
19 <!DOCTYPE html>
20 <html lang="pt-br">
21 <head>
22   <meta charset="UTF-8">
23   <meta http-equiv="X-UA-Compatible" content="IE=edge">
24   <meta name="viewport" content="width=device-width, initial-scale=1.0">
25   <title>Projeto PHP Versátil Ti</title>
26   <link rel="stylesheet" type="text/css" href="css/formulario.css">
27 </head>
```



```

29
30 <body>
31   <div class="box">
32     <h1>Cadastro de Clientes</h1>
33     <form action="formulario.php" method="POST">
34       <fieldset>
35
36         <br>
37         <div class="inputBox">
38           <input type="text" name="nome" id="nome" class="inputUser" required>
39           <label for="nome" class="labelInput">Nome</label>
40         </div>
41         <br><br>
42         <div class="inputBox">
43           <input type="text" name="email" id="email" class="inputUser" required>
44           <label for="email" class="labelInput">E-mail</label>
45         </div>
46         <br><br>
47         <div class="inputBox">
48           <input type="password" name="senha" id="senha" class="inputUser" required>
49           <label for="senha" class="labelInput">Senha</label>
50         </div>
51         <br>
52
53         <p>Sexo:</p>
54         <input type="radio" id="feminino" name="sexo" value="feminino" required>
55         <label for="feminino">Feminino</label>
56         <br>
57         <input type="radio" id="masculino" name="sexo" value="masculino" required>
58         <label for="masculino">Masculino</label>
59         <br>
60         <input type="radio" id="outro" name="sexo" value="outro" required>
61         <label for="outro">Outro</label>
62         <br><br><br>
63
64         <div class="inputBox">
65           <input type="text" name="cidade" id="cidade" class="inputUser" required>
66           <label for="cidade" class="labelInput">Cidade/UF</label>
67         </div>
68         <br><br>
69
70         <input type="submit" name="submit" id="submit" value="Cadastrar">
71         <br><br>
72         <a href="index.php" class="button">Voltar</a>
73       </fieldset>
74     </form>
75   </div>
76 </body>
77 </html>

```

3. Agora vamos fazer a estilização criando um novo arquivo chamado **formulario.css** na pasta css:

```

1 body {
2   font-family: Arial, Helvetica, sans-serif;
3   background-color: #cccccc;
4 }
5 .box {
6   color: white;
7   position: absolute;
8   top: 50%;
9   left: 50%;
10  transform: translate(-50%, -50%);
11  background-color: #002966;
12  padding: 15px;
13  border-radius: 15px;
14  width: 30%;
15 }
16 .inputBox {
17   position: relative;
18 }


```

```

18 }
19 .inputUser {
20   background: none;
21   border: none;
22   border-bottom: 1px solid #white;
23   outline: none;
24   color: #white;
25   font-size: 15px;
26   width: 100%;
27   letter-spacing: 2px;
28 }
29 .labelInput {
30   position: absolute;
31   top: 0px;
32   left: 0px;
33   pointer-events: none;
34   transition: .5s;
35 }
36 .inputUser:focus~.labelInput,
37 .inputUser:valid~.labelInput {
38   top: -20px;
39   font-size: 12px;
40   color: #dodgerblue;
41 }
42 #submit {
43   background-color: #dodgerblue;
44   width: 100%;
45   border: none;
46   padding: 15px;
47   color: #white;
48   font-size: 15px;
49   cursor: pointer;
50   border-radius: 5px;
51 }
52 #submit:hover {
53   background-color: #ff6600;
54 }
55 .button {
56   font-family: Arial, Helvetica, sans-serif;
57   font-size: 14px;
58   border-radius: 4px;
59   padding-left: 20px;
60   padding-right: 20px;
61   padding-top: 5px;
62   padding-bottom: 5px;
63   color: #ffffff;
64   background-color: #dodgerblue;
65   outline: none;
66   border: none;
67   cursor: pointer;
68   display: inline-block;
69   text-decoration: none;
70   transition: .4s;
71 }
72 .button:hover {
73   color: #ffffff;
74   background-color: #ff6600;
75 }

```

4. Agora rode o arquivo e verifique se ele está semelhante a imagem abaixo:




```

111 <script>
112 var search = document.getElementById('pesquisar');
113
114 search.addEventListener("keydown", function(event) {
115   if (event.key === "Enter")
116   {
117     searchData();
118   }
119 });
120
121 function searchData()
122 {
123   window.location = 'sistema.php?search='+search.value;
124 }
125 </script>
126 </html>

```

2. Agora vamos fazer a estilização criando um novo arquivo chamado **sistema.css** na pasta css.

```

1 body {
2   background-color: #002966;
3   color: white;
4   text-align: left;
5 }
6
7 .table-bg {
8   background: white;
9   border-radius: 10px 10px 0 0;
10 }
11
12 .box-search {
13   display: flex;
14   justify-content: center;
15   gap: .1%;
16 }

```

3. Agora rode o arquivo e verifique se ele está semelhante a imagem abaixo:

Obs: para ter acesso a tela principal do sistema faça login com alguns dos logins que foram inseridos no banco de dados.

Tela do sistema:



Cadastro de Clientes

Bem vindo kamila@versatilti.com.br

Pesquisar

#	Nome	E-mail	Sexo	Cidade	...
7	Giovanna Santana	gi@versatilti.com.br	feminino	Gama/DF	✎ ✖
6	Kamila Sales	kamila@versatilti.com.br	feminino	Brasília/DF	✎ ✖
5	Kauan Barros	kauan@versatilti.com.br	masculino	Sobradinho/DF	✎ ✖
4	Cibele Santana	cibele@gmail.com	feminino	Taguatinga/DF	✎ ✖
3	Judson Martins	judson@versatilti.com.br	masculino	Brasília/DF	✎ ✖

Criando tela de editar cadastro

A tela de edição do sistema seria a interface onde os usuários podem modificar os dados de um registro existente no banco de dados.

1. Crie um arquivo com o nome **edit.php**:

edit.php

```

1  <?php
2      include_once('conexao.php');
3
4      if(!empty($_GET['id']))
5      {
6          $id = $_GET['id'];
7          $sqlSelect = "SELECT * FROM cadastro WHERE id=$id";
8          $result = $conexao->query($sqlSelect);
9          if($result->num_rows > 0)
10         {
11             while($user_data = mysqli_fetch_assoc($result))
12             {
13                 $nome = $user_data['nome'];
14                 $email = $user_data['email'];
15                 $senha = $user_data['senha'];
16                 $sexo = $user_data['sexo'];
17                 $cidade = $user_data['cidade'];
18             }
19         }
20     }
21     else
22     {
23         header("Location: sistema.php");
24     }
25     else
26     {
27         header("Location: sistema.php");
28     }
29 }
30
31 <!DOCTYPE html>
32 <html lang="pt-br">
33 <head>
34     <meta charset="UTF-8">
35     <meta http-equiv="X-UA-Compatible" content="IE=edge">
36     <meta name="viewport" content="width=device-width, initial-scale=1.0">
37     <title>Projeto PHP Versátil Ti</title>
38     <link rel="stylesheet" type="text/css" href="css/edit.css">
39 </head>
40
41 <body>
42
43     <div class="box">
44         <form action="saveEdit.php" method="POST">
45             <fieldset>
46                 <legend><b><h2>Editar Cliente</b></h2></legend>
47                 <br>
48                 <div class="inputBox">
49                     <input type="text" name="nome" id="nome" class="inputUser" value=<?php echo $nome;?> required>
50                     <label for="nome" class="labelInput">Nome</label>
51                 </div>
52                 <br>
53                 <div class="inputBox">
54                     <input type="text" name="email" id="email" class="inputUser" value=<?php echo $email;?> required>
55                     <label for="email" class="labelInput">Email</label>
56                 </div>
57                 <br>
58                 <div class="inputBox">
59                     <input type="text" name="senha" id="senha" class="inputUser" value=<?php echo $senha;?> required>
60                     <label for="senha" class="labelInput">Senha</label>
61                 </div>
62                 <br>
63                 <p>Sexo:</p>
64                 <input type="radio" id="feminino" name="sexo" value="feminino" <?php echo ($sexo == 'feminino') ? 'checked' : ''?> required>
65                 <label for="feminino">Feminino</label>
66                 <br>
67                 <input type="radio" id="masculino" name="sexo" value="masculino" <?php echo ($sexo == 'masculino') ? 'checked' : ''?> required>
68                 <label for="masculino">Masculino</label>
69                 <br>
70                 <input type="radio" id="outro" name="sexo" value="outro" <?php echo ($sexo == 'outro') ? 'checked' : ''?> required>
71                 <label for="outro">Outro</label>
72                 <br>
73                 <div class="inputBox">
74                     <input type="text" name="cidade" id="cidade" class="inputUser" value=<?php echo $cidade;?> required>
75                     <label for="cidade" class="labelInput">Cidade</label>
76                 </div>
77                 <br>
78                 <input type="hidden" name="id" value=<?php echo $id;?>
79                 <input type="submit" name="update" id="submit" value="Alterar">
80             </fieldset>
81         </form>
82     </div>
83 </body>
84 </html>

```

2. Agora vamos fazer a estilização criando um novo arquivo chamado **edit.css** na pasta css.

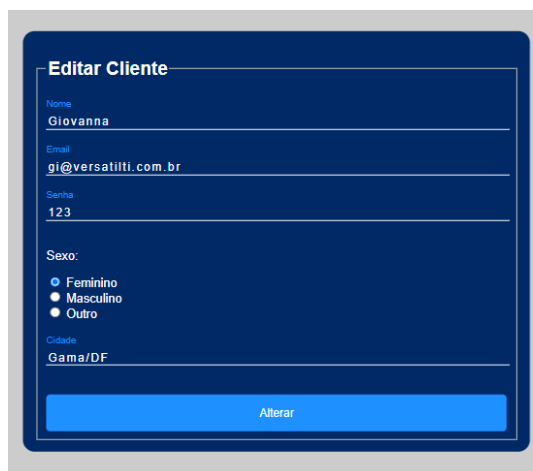
```

1 body {
2     font-family: Arial, Helvetica, sans-serif;
3     background-color: #cccccc;
4 }
5
6 .box {
7     color: #white;
8     position: absolute;
9     top: 50%;
10    left: 50%;
11    transform: translate(-50%, -50%);
12    background-color: #002966;
13    padding: 15px;
14    border-radius: 15px;
15    width: 40%;
16 }
17
18 .inputBox {
19     position: relative;
20 }
21
22 .inputUser {
23     background: none;
24     border: none;
25     border-bottom: 1px solid #white;
26     outline: none;
27     color: #white;
28     font-size: 15px;
29     width: 100%;
30     letter-spacing: 2px;
31 }
32
33 .labelInput {
34     position: absolute;
35     top: 0px;
36     left: 0px;
37     pointer-events: none;
38     transition: .5s;
39 }
40
41 .inputUser:focus+.labelInput,
42 .inputUser:valid+.labelInput {
43     top: -20px;
44     font-size: 12px;
45     color: #dodgerblue;
46 }
47
48 #submit {
49     background-color: #dodgerblue;
50     width: 100%;
51     border: none;
52     padding: 15px;
53     color: #white;
54     font-size: 15px;
55     cursor: pointer;
56     border-radius: 5px;
57 }
58
59 #submit:hover {
60     background-color: #ff6600;
61 }

```

3. Agora rode o arquivo e verifique se ele está semelhante a imagem abaixo:

Obs: clique no ícone de um lápis de algum cadastrado que se encontra na tabela da tela principal.



4. Agora crie o arquivo **saveEdit** que fará comunicação com o banco de dados e efetuará o update das informações.

saveEdit.php

```
1 <?php
2
3 include_once('conexao.php');
4 if(isset($_POST['update']))
5 {
6     $id = $_POST['id'];
7     $nome = $_POST['nome'];
8     $email = $_POST['email'];
9     $senha = $_POST['senha'];
10    $sexo = $_POST['sexo'];
11    $cidade = $_POST['cidade'];
12
13    $sqlInsert = "UPDATE cadastro
14    SET nome='$nome',email='$email',senha='$senha',sexo='$sexo',cidade='$cidade'
15    WHERE id=$id";
16    $result = $conexao->query($sqlInsert);
17    print_r($result);
18 }
19 header('Location: sistema.php');
20
21 ?>
```

Criando função de excluir cadastro

A tela de exclusão de cadastro no sistema do projeto seria uma interface onde os usuários teriam a capacidade de selecionar um registro existente para remoção do banco de dados.

1. Crie um arquivo com o nome **delete.php**:

delete.php

```
1 <?php
2
3 if(!empty($_GET['id']))
4 {
5     include_once('conexao.php');
6
7     $id = $_GET['id'];
8
9     $sqlSelect = "SELECT * FROM cadastro WHERE id=$id";
10
11    $result = $conexao->query($sqlSelect);
12
13    if($result->num_rows > 0)
14    {
15        $sqlDelete = "DELETE FROM cadastro WHERE id=$id";
16        $resultDelete = $conexao->query($sqlDelete);
17    }
18 }
19 header('Location: sistema.php');
20
21 ?>
```

2. Agora rode o arquivo:

Obs: clique no ícone de uma lixeira de algum cadastrado que se encontra na tabela da tela principal.

Criando função de sair do sistema

A função de "Sair do Sistema" é uma parte importante de qualquer aplicação web que requer autenticação de usuário. Ela fornece aos usuários uma maneira segura de encerrar sua sessão e sair do sistema

1. Crie um arquivo com o nome **sair.php**:

sair.php

```
1 <?php
2     session_start();
3     unset($_SESSION['email']);
4     unset($_SESSION['senha']);
5     header("Location: index.php");
6 ?>
```

Agora teste todas as funções do sistema, se alguma funcionalidade não funcionar solicite a orientação do(a) professor(a).