

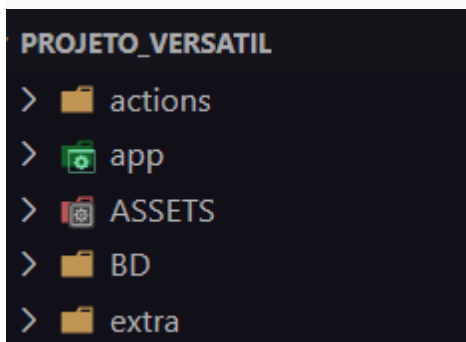


## Sumário

1. Construindo a estrutura do projeto. ....	3
2. Criando um banco de dados. ....	3
3. Criando Layout do projeto. ....	4
4. Criando sessões do projeto. ....	7
5. Criando objetos de acesso a dados do projeto. ....	11
6. Criando interface para manipulação de contatos. ....	15
7. Configurando ações para o controle da aplicação. ....	19
8. Criando Layout da página de login e cadastro ....	22
9. Criando arquivos de listagem de cadastros ....	24

## 1. Construindo a estrutura do projeto.

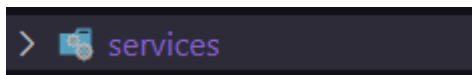
Crie uma pasta com o nome **projeto\_versatil** e dentro dela crie as seguintes subpastas: *actions*, *app*, *ASSETS*, *DB* e *extra* como mostra a imagem abaixo:



## 2. Criando um banco de dados.

Use o script fornecido pelo seu instrutor para montar a estrutura do banco de dados;

Dentro da pasta **App** crie uma subpasta com o nome **services**:



Crie uma classe, como o método construtor, conexão o banco de dados através da API PDO, os métodos `getConexao()` e `getInstance()` com o nome **MySQLDatabase.php**;

## MySQLDatabase.php

```
<?php

class MySQLDatabase
{
    private $conexao;
    private $Username = "root";
    private $Host = "localhost";
    private $Password = "";
    private $BancoDeDados = "versatil_projeto";

    public function __construct()
    {
        try {
            $this->conexao = new PDO("mysql:host={$this->Host};dbname={$this->BancoDeDados}", $this->Username, $this->Password);
        } catch (PDOException $e) {
            die("ERRO: " . $e->getMessage());
        }
    }

    public function getConexao()
    {
        return $this->conexao;
    }

    public static function getInstance()
    {
        $mysqlConnect = new MySQLDatabase;
        return $mysqlConnect->getConexao();
    }
}
```

Para testar a conexão com o banco de dados escreva: **echo “*Banco conectado com sucesso!*”**; na última linha.

Caso essa frase apareça na tela a conexão foi bem sucedida, caso contrário, aparecerá uma mensagem de erro, após sanado o erro, apague a mensagem de teste.

## 3. Criando Layout do projeto.

Crie os arquivos **header.php**, **footer.php** e **menu.php** dentro na pasta **extra**, com o cabeçalho, rodapé e menu que serão usados nos arquivos do projeto.

### header.php

```
<!DOCTYPE html>
<html lang='pt-br'>

<head>
    <meta charset='UTF-8'>
    <meta name='viewport' content='width=device-width, initial-scale=1.0'>

    <title> Versátil Cursos </title>

    <link rel='stylesheet' href='ASSETS/bootstrap-4.5.3/dist/css/bootstrap.min.css'>
    <link rel='stylesheet' href='ASSETS/bootstrap-icons-1.4.0/bootstrap-icons.css'>
    <link rel='stylesheet' href='ASSETS/custom.css' />
</head>

<body>
```

## footer.php

```
<script src='ASSETS/jquery-3.3.1.slim.min.js'></script>
<script src="ASSETS/jquery.mask.js"></script>
<script src='ASSETS/bootstrap-4.5.3/dist/js/bootstrap.bundle.min.js'></script>
<script src='ASSETS/main.js'></script>
</body>

</html>
```

## menu.php

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <div class="container">
    <p class="navbar-brand">
      
    </p>

    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarContent">
      <span class="navbar-toggler-icon"></span>
    </button>

    <div id="navbarContent" class="collapse navbar-collapse">

      <form class="d-flex m-auto mb-2" method="GET" action="home.php#listar">
        <input class="form-control mr-2" type="search" placeholder="Pesquisar por..." name="search">
        <button class="btn btn-success" type="submit">
          <i class="bi-search"></i>
        </button>
      </form>

      <ul class="navbar-nav nav mb-2">

        <li class="nav-item">
          <a class="nav-link" href="home.php">
            <i class="bi bi-house-door-fill"></i>
            Início
          </a>
        </li>

        <li class="nav-item">
          <a class="nav-link" href="cadastro.php#novo">
            <i class="bi bi-person"></i>
            Cadastrar
          </a>
        </li>

        <li class="nav-item">
          <a class="nav-link" href="actions/logout.php">
            <i class="bi bi-box-arrow-left"></i>
            Sair
          </a>
        </li>

      </ul>

    </div>
  </div>
</nav>
```

Teste o arquivo `menu.php` usando a url [http://localhost/projeto\\_versatil/extra/menu.php](http://localhost/projeto_versatil/extra/menu.php) verifique se fica como abaixo.



## 4. Criando sessões do projeto.

Para que possamos guardar todos os dados do usuário, vamos utilizar a superglobal `$_SESSION`, pois, com ela podemos separar todos os dados do cliente de forma segura e não compartilhada com outras sessões (outros usuários acessando a página). De forma geral, os dados salvos em uma sessão só podem ser acessados por ela mesma.

Obs: Todos os dados de sessão são armazenados pelo back-end, onde o cliente conhece apenas o seu id, que é armazenado em forma de cookie pelo navegador.

Vamos utilizar o método **`getInstance`** para recuperar os dados do usuário e vamos armazená-lo sobre o índice `$_SESSION['user_data']`.

Crie o arquivo **`UserService.php`** dentro da pasta **services**:

### **UserService.php**

```
1  <?php
2
3  class UserService
4  {
5      private $Conexao;
6
7      public function __construct()
8      {
9          # conectando ao banco de dados
10         $this->Conexao = MySQLDatabase::getInstance();
11     }
12
13     public function getUsuario()
14     {
15         if (isset($_SESSION['user_data'])) {
16             return $_SESSION['user_data'];
17         } else {
18             return null;
19         }
20     }
21
22     public function estaLogado()
23     {
24         if ($this->getUsuario() != null) {
25             return true;
26         }
27         return false;
28     }
29 }
```

```
29  {
30      if ($this->getUsuario() != null) {
31          return true;
32      }
33      return false;
34  }
35
36  public static function proibirNaoLogado()
37  {
38      if (!isset($_SESSION['user_data'])) {
39          echo '<script>window.location.href = "denied.html"</script>';
40          exit;
41      }
42  }
43
44  public function login($username, $password)
45  {
46      $sql = "SELECT * FROM usuarios WHERE usuario = ? AND senha = ?";
47
48      $query = $this->Conexao->prepare($sql);
49
50      $query->bindParam(1, $username, PDO::PARAM_STR);
51      $query->bindParam(2, $password, PDO::PARAM_STR);
52      $query->execute();
53
54      $resultado = $query->fetch(PDO::FETCH_ASSOC);
55      if ($resultado !== false) {
56          $_SESSION['user_data'] = $resultado['usuario'];
57          return true;
58      }
59      return false;
60  }
61
62  public function logout()
63  {
64      # com a chamada do método 'session_destroy' todos os dados
65      # da $_SESSION serão apagados, invalidando nosso $_SESSION['user_data']
66      session_destroy();
67  }
68 }
```



Crie o arquivo **UtilService.php** dentro da pasta **services**:

### **UtilService.php**

```
<?php
class UtilService
{
    public static function decodeArrayToUtf8($array)
    {
        $arr = [];
        foreach($array as $k => $v) {
            $arr[$k] = utf8_decode($v);
        }
        return $arr;
    }
}
```

E vamos criar também um arquivo como o nome **autoloader.php** na pasta services para fazer o mapeamento de REQUIRE de qualquer classe utilizada em nosso projeto.

### **autoloader.php**

```
<?php

session_start();

spl_autoload_register( function($className)
{
    $diretorios = ["db", "services"];

    foreach($diretorios as $pasta) {
        $classProcurada = __DIR__ . "/" . $pasta . "/" . $className . ".php";

        if (file_exists($classProcurada)) {

            require_once $classProcurada;
            return;
        }
    }
});
```

## 5. Criando objetos de acesso a dados do projeto.

Para dar início a tarefa de conexão, sem o comando manual, crie o arquivo **DAO.php** dentro da pasta **bd**, que através de um método construtor, inicializa a conexão com o banco de dados.

### DAO.php

```
<?php
abstract class DAO
{
    protected $Conexao;

    public function __construct() {
        $this->Conexao = MySQLDatabase::getInstance();
    }
}
```

Crie um arquivo com o nome **CargoDAO.php** dentro da pasta **bd**, para fazer a conexão com o banco de dados selecionando os cargos pretendidos dentro de uma comboBox.

### CargoDAO.php

```
<?php
class CargoDAO extends DAO
{
    public function buscarTodos()
    {
        $query = $this->Conexao->prepare("SELECT id, descricao FROM cargo ORDER BY descricao");
        $query->execute();
        return $query->fetchAll(PDO::FETCH_ASSOC);
    }

    public function buscarPorId($id)
    {
        $query = $this->Conexao->prepare("SELECT id, descricao FROM cargo WHERE id = ? ORDER BY descricao");
        $query->bindParam(1, $id, PDO::PARAM_INT);
        $query->execute();
        return $query->fetchAll(PDO::FETCH_ASSOC);
    }
}
```

Cargo Pretendido:

Estagiário

Estagiário

Gerente

Operacional

Supervisor

Trainee

Crie um arquivo com o nome **AreaDeAtuacao.php** dentro da pasta **bd**, para fazer a conexão com banco de dados selecionando as áreas de atuação pretendidas dentro de uma comboBox.

## AreaDeAtuacao.php

```
<?php
class AreaDeAtuacaoDAO extends DAO
{
    public function buscarTodos()
    {
        $query = $this->Conexao->prepare("SELECT id, descricao FROM area_atuacao ORDER BY descricao");
        $query->execute();
        return $query->fetchAll(PDO::FETCH_ASSOC);
    }

    public function buscarPorId($id)
    {
        $query = $this->Conexao->prepare("SELECT id, descricao FROM area_atuacao WHERE id = ? ORDER BY descricao");
        $query->bindParam(1, $id, PDO::PARAM_INT);
        $query->execute();
        return $query->fetchAll(PDO::FETCH_ASSOC);
    }
}
```

Área de Atuação:

Analista de Banco de Dados

Analista de Banco de Dados

Design

Help desk

Programador

Crie um arquivo com o nome **CapitalDAO.php** dentro da pasta **bd**, para fazer a conexão com o banco de dados selecionando as capitais pretendidas dentro de uma comboBox.

## CapitalDAO.php

```
<?php
class CapitalDAO extends DAO
{
    public function buscarTodos()
    {
        $query = $this->Conexao->prepare("SELECT id, descricao FROM capital ORDER BY descricao");
        $query->execute();
        return $query->fetchAll(PDO::FETCH_ASSOC);
    }

    public function buscarPorId($id)
    {
        $query = $this->Conexao->prepare("SELECT id, descricao FROM capital WHERE id = ? ORDER BY descricao");
        $query->bindParam(1, $id, PDO::PARAM_INT);
        $query->execute();
        return $query->fetchAll(PDO::FETCH_OBJ);
    }
}
```

Capital onde pretende trabalhar:

☐ Brasília

☐ Goiás

☐ Rio de Janeiro

☐ São Paulo

Crie um arquivo com o nome **TempoDeExperienciaDAO.php** dentro da pasta **bd**, para fazer a conexão com o banco de dados selecionando os tempos de experiências pretendidos dentro de uma comboBox.

### TempoDeExperienciaDAO.php

```
<?php
class TempoDeExperienciaDAO extends DAO
{
    public function buscarTodos()
    {
        $query = $this->Conexao->prepare("SELECT id, descricao FROM tempo_experiencia ORDER BY descricao DESC");
        $query->execute();
        return $query->fetchAll(PDO::FETCH_ASSOC);
    }

    public function buscarPorId($id)
    {
        $query = $this->Conexao->prepare("SELECT id, descricao FROM tempo_experiencia WHERE id = ? ORDER BY descricao");
        $query->bindParam(1, $id, PDO::PARAM_INT);
        $query->execute();
        return $query->fetchAll(PDO::FETCH_ASSOC);
    }
}
```

Tempo de Experiência:

Sem experiência

Sem experiência

7 anos ou mais.

5 a 7 anos

3 a 5 anos

1 ano ou menos

1 a 3 anos.

Crie um arquivo com o nome **ContatosDAO.php** dentro da pasta **bd**, para fazer a conexão com o banco de dados selecionando os contatos pretendidos dentro de uma tabela.

## ContatosDAO.php

```
1 <?php
2 class ContatosDAO extends DAO
3 {
4     public function buscarTodos()
5     {
6         $query = $this->Conexao->prepare("SELECT * FROM contatos");
7         $query->execute();
8
9         return $query->fetchAll(PDO::FETCH_ASSOC);
10    }
11
12    public function buscarTodosPorPesquisa($busca)
13    {
14        $query = $this->Conexao->prepare("SELECT * FROM contatos WHERE nome LIKE '%$busca%' OR telefone LIKE '%$busca%' OR email LIKE '%$busca%'");
15        $query->execute();
16        return $query->fetchAll(PDO::FETCH_ASSOC);
17    }
18
19    public function buscarPorId($id)
20    {
21        $query = $this->Conexao->prepare("SELECT * FROM contatos WHERE id = ?");
22        $query->bindParam(1, $id, PDO::PARAM_INT);
23        $query->execute();
24
25        return $query->fetch(PDO::FETCH_ASSOC);
26    }
27
28    public function apagarPorId($id)
29    {
30        return $this->Conexao->exec("DELETE FROM contatos WHERE id = $id");
31    }
32
33    public function atualizarPorId($id, $data = [])
34    {
35        $campos = [];
36        foreach($data as $k => $v) {
37            $campos[] = "$k = '$v'";
38        }
39        $toSql = join(", ", $campos);
40        return $this->Conexao->exec("UPDATE contatos SET $toSql WHERE id = $id");
41    }
42
43    public function cadastrar($data = [])
44    {
45        $query = $this->Conexao->prepare("
46        INSERT INTO contatos
47        (nome, telefone, email, comentario, area_atuacao_id, tempo_experiencia_id, capital_id, cargo_id)
48        VALUES (?, ?, ?, ?, ?, ?, ?, ?)");
49
50        $query->bindParam(1, utf8_decode($data['nome']), PDO::PARAM_STR);
51        $query->bindParam(2, $data['telefone'], PDO::PARAM_STR);
52        $query->bindParam(3, $data['email'], PDO::PARAM_STR);
53        $query->bindParam(4, $data['comentario'], PDO::PARAM_STR);
54        $query->bindParam(5, $data['area_atuacao_id'], PDO::PARAM_STR);
55        $query->bindParam(6, $data['tempo_experiencia_id'], PDO::PARAM_STR);
56        $query->bindParam(7, $data['capital_id'], PDO::PARAM_STR);
57        $query->bindParam(8, $data['cargo_id'], PDO::PARAM_STR);
58
59        return $query->execute();
60    }
61 }
62
```

## 6. Criando interface para manipulação de contatos.

Crie um arquivo com o nome **formularioContato.php** dentro da pasta **extra**. Nesse arquivo iremos criar um formulário com **actions/salvar.php** e os campos: nome, telefone, email, capital, cargo pretendido, tempo de experiência e área de atuação como segue na imagem abaixo.

### formularioContato.php

```
1 <form method="POST" action="actions/salvar.php">
2 <div class="row mb-3">
3 <!-- Campo Nome -->
4 <div class="col-md-4">
5 <label for="nome">Nome:</label>
6 <input type="text" required class="form-control" id="nome" name="nome">
7 </div>
8 <!-- Campo Telefone -->
9 <div class="col-md-2">
10 <label for="telefone">Telefone:</label>
11 <input type="tel" required class="form-control" id="telefone" name="telefone">
12 </div>
13 <!-- Campo Email -->
14 <div class="col-md-6">
15 <label for="email">Email:</label>
16 <input type="email" required class="form-control" id="email" name="email">
17 </div>
18 </div>
19
20 <div class="row mb-3">
21
22 <div class="col-md-6">
23 <!-- -->
24 <p class="form-label">Capital onde pretende trabalhar:</p>
25
26 <?php
27 $capitalDao = new CapitalDAO;
28 foreach ($capitalDao->buscarTodos() as $capital) {
29 $descricao = utf8_encode($capital['descricao']);
30 echo "
31 <div class='form-check mb-2'>
32 <input class='form-check-input' type='radio' name='capital_id'
33 id='radioCapital-{$capital['id']}' value='{$capital['id']}'>
34 <label class='form-check-label' for='radioCapital-{$capital['id']}'>
35 {$descricao}
36 </label>
37 </div>
38 ";
39 }
40 ?>
41
42 </div>
43
44 <div class="col-md-6">
45
46 <div class="mb-2">
47 <label for="cargo">
48 Cargo Pretendido:
49 </label>
50
51 <select name="cargo_id" class="form-control" id="cargo">
52 <?php
53 $cargoDao = new CargoDAO;
54 foreach ($cargoDao->buscarTodos() as $cargo) {
55 $descricao = utf8_encode($cargo['descricao']);
56 echo "<option value='{$cargo['id']}'> {$descricao} </option>";
57 }
58 ?>
59 </select>
60 </div>
```

```

59     </div>
60
61     <div class="mb-2">
62         <label for="tempoExperiencia">Tempo de Experiência:</label>
63         <select name="tempo_experiencia_id" class="form-control" id="tempoExperiencia">
64             <?php
65                 $tempoExperienciaDao = new TempoDeExperienciaDAO;
66                 foreach($tempoExperienciaDao->buscarTodos() as $tempo) {
67                     $descricao = utf8_encode($tempo['descricao']);
68                     echo "<option value=\"{$tempo['id']}\"> {$descricao} </option>";
69                 }
70             ?>
71         </select>
72     </div>
73
74     <div class="mb-2">
75         <label for="areaAtuacao">Área de Atuação:</label>
76         <select name="area_atuacao_id" class="form-control" id="areaAtuacao">
77             <?php
78                 $areaDao = new AreaDeAtuacaoDAO;
79                 foreach($areaDao->buscarTodos() as $area) {
80                     $descricao = utf8_encode($area['descricao']);
81                     echo "<option value=\"{$area['id']}\"> {$descricao} </option>";
82                 }
83             ?>
84         </select>
85     </div>
86
87 </div>
88
89 </div>
90
91 <div class="row mb-3">
92
93     <div class="col-12">
94         <!-- Campo Comentário -->
95         <div class="form-group">
96             <label for="comentario">Comentários</label>
97             <textarea class="form-control" name="comentario" id="comentario" rows="3"></textarea>
98         </div>
99     </div>
100
101 </div>
102
103 <div class="row justify-content-end mb-4">
104     <div class="col-3">
105         <!-- Botão de Envio dos Dados -->
106         <button type="submit" class="btn btn-primary float-right">
107             Salvar
108         </button>
109     </div>
110
111 </div>
112 </form>

```



Crie um arquivo com o nome **formularioEditaContato.php** dentro da pasta **extra**. Nesse arquivo iremos criar um formulário com **actions/salvar.php** e os campos: nome, telefone, email, capital, cargo pretendido, tempo de experiência e área de atuação como segue na imagem abaixo.

## formularioEditaContato.php

```

1  <?php
2  $contatoDao = new ContatosDAO;
3  if (isset($_GET['id'])) {
4      $c = $contatoDao->buscarPorId($_GET['id']);
5  } else {
6      var_dump($_GET);
7      die("Esqueceu de passar o ID.");
8      exit;
9  }
10 ?>
11
12 <form method="POST" action="actions/salvar.php">
13     <!--
14         Chave primária do contato, não será visível pelo usuário,
15         a menos é claro que seja inspecionado pelo navegador.
16     -->
17     <input type="hidden" name="id" value="<?= $c['id'] ?>" />
18
19     <div class="row mb-3">
20         <!-- Campo Nome -->
21         <div class="col-md-4">
22             <label for="nome">Nome:</label>
23             <input type="text" required class="form-control" id="nome" name="nome" value="<?= $c['nome'] ?>">
24         </div>
25         <!-- Campo Telefone -->
26         <div class="col-md-2">
27             <label for="telefone">Telefone:</label>
28             <input type="tel" required class="form-control" id="telefone" name="telefone"
29                 value="<?= $c['telefone'] ?>">
30         </div>
31         <!-- Campo Email -->
32         <div class="col-md-6">
33             <label for="email">Email:</label>
34             <input type="email" required class="form-control" id="email" name="email"
35                 value="<?= $c['email'] ?>">
36         </div>
37     </div>
38
39     <div class="row mb-3">
40
41         <div class="col-md-6">
42             <!-- -->
43             <p class="form-label">Capital onde pretende trabalhar:</p>
44
45             <?php
46             $capitalDao = new CapitalDAO;
47             foreach ($capitalDao->buscarTodos() as $capital) {
48                 $descricao = utf8_encode($capital['descricao']);
49
50                 echo "<div class='form-check mb-2'>";
51
52                 if ($capital['id'] == $c['capital_id']) {
53                     echo "<input checked class='form-check-input' type='radio' name='capital_id'
54                         id='radioCapital-{$capital['id']}' value='{$capital['id']}'>";
55                 } else {
56                     echo "<input class='form-check-input' type='radio' name='capital_id'
57                         id='radioCapital-{$capital['id']}' value='{$capital['id']}'>";
58                 }
59                 echo "
60                 <label class='form-check-label' for='radioCapital-{$capital['id']}'>
61                     {$descricao}
62                 </label>
63                 </div>";

```

```

64     }
65     ?>
66
67 </div>
68
69 <div class="col-md-6">
70
71     <div class="mb-2">
72         <label for="cargo">
73             Cargo Pretendido:
74         </label>
75
76         <select name="cargo_id" class="form-control" id="cargo">
77             <?php
78                 $cargoDao = new CargoDAO;
79                 foreach($cargoDao->buscarTodos() as $cargo) {
80                     $descricao = utf8_encode($cargo['descricao']);
81
82                     if ($cargo['id'] == $c['cargo_id']) {
83                         echo "<option value=\"{$cargo['id']}\" selected> {$descricao} </option>";
84                     } else {
85                         echo "<option value=\"{$cargo['id']}\"> {$descricao} </option>";
86                     }
87                 }
88             <?>
89         </select>
90     </div>
91
92     <div class="mb-2">
93         <label for="tempoExperiencia">Tempo de Experiência:</label>
94         <select name="tempo_experiencia_id" class="form-control" id="tempoExperiencia">
95             <?php
96                 $tempoExperienciaDao = new TempoDeExperienciaDAO;
97                 foreach($tempoExperienciaDao->buscarTodos() as $te) {
98                     $descricao = utf8_encode($te['descricao']);
99
100                     if ($te['id'] == $c['tempo_experiencia_id']) {
101                         echo "<option value=\"{$te['id']}\" selected> {$descricao} </option>";
102                     } else {
103                         echo "<option value=\"{$te['id']}\"> {$descricao} </option>";
104                     }
105                 }
106             <?>
107         </select>
108     </div>
109
110     <div class="mb-2">
111         <label for="areaAtuacao">Área de Atuação:</label>
112         <select name="area_atuacao_id" class="form-control" id="areaAtuacao">
113             <?php
114                 $areaDao = new AreaDeAtuacaoDAO;
115                 foreach($areaDao->buscarTodos() as $area) {
116                     $descricao = utf8_encode($area['descricao']);
117
118                     if ($area['id'] == $c['area_atuacao_id']) {
119                         echo "<option value=\"{$area['id']}\" selected> {$descricao} </option>";
120                     } else {
121                         echo "<option value=\"{$area['id']}\"> {$descricao} </option>";
122                     }
123                 }
124             <?>
125         </select>
126     </div>
127
128 </div>
129
130 </div>
131
132 <div class="row mb-3">
133
134     <div class="col-12">
135         <!-- Campo Comentário -->
136         <div class="form-group">
137             <label for="comentario">Comentários</label>
138             <textarea class="form-control" name="comentario" id="comentario"
139                 rows="3"><?=$c["comentario"] ?></textarea>
140         </div>
141     </div>
142
143 </div>
144
145 <div class="row justify-content-end mb-4">
146     <div class="col-3">
147         <!-- Botão de Envio dos Dados -->
148         <button type="submit" class="btn btn-primary float-right">
149             Salvar
150         </button>
151     </div>
152
153 </div>
154
155 </form>

```

## 7. Configurando ações para o controle da aplicação.

Crie um arquivo com o nome **salvar.php** dentro da pasta **actions**. Neste arquivo iremos criar uma ação para salvar o registro na tabela de contatos.

### salvar.php

```
<?php
require_once __DIR__ . '/../app/autoloader.php';
UserService::proibirNaoLogado();

$contatoDao = new ContatosDAO();

echo "<pre>";
var_dump($_POST);
echo "</pre>";

try {
    if (isset($_POST['id'])) {
        $contatoDao->atualizarPorId($_POST['id'], $_POST);
    } else {
        $contatoDao->cadastrar($_POST);
    }

    echo'
    <script>
        alert("Registro salvo com sucesso!");
        window.location.href = "../home.php";
    </script>
    ';
} catch (PDOException $e) {
    die("Falha ao salvar registro: " . $e->getMessage());
    exit;
}
```

Crie um arquivo com o nome **logout.php** dentro da pasta **actions**. Neste arquivo iremos criar uma ação para sair, de tal forma que o usuário não acessar nenhum recurso do sistema sem digitar sua senha novamente.

### logout.php

```
<?php
require_once __DIR__ . "../app/autoload.php";

$userService = new UserService;
$userService->logout();

echo <<< HTMLA
<script>
    alert('Logout efetuado com sucesso');
    window.location.href = "../index.php";
</script>
HTMLA;

exit;
```

Crie um arquivo com o nome **apagar.php** dentro da pasta **actions**. Neste arquivo iremos criar uma ação para deletar um registro do banco de dados.

### **apagar.php**

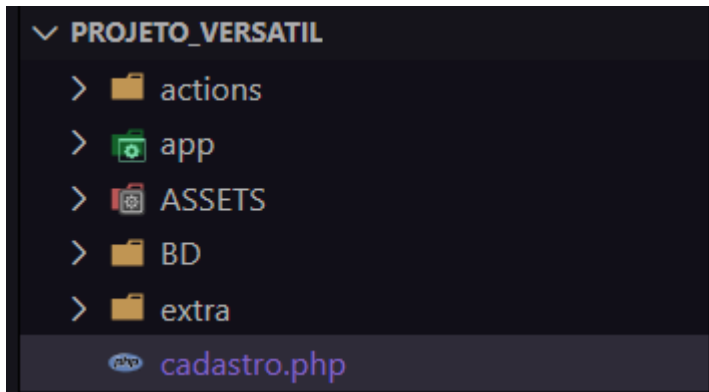
```
<?php
require_once __DIR__ . '/../app/autoloader.php';
UserService::proibirNaoLogado();

$contatoDao = new ContatosDAO;

try {
    if (isset($_GET['id'])) {
        $contatoDao->apagarPorId($_GET['id']);
        echo '
        <script>
            alert("Objeto apagado com sucesso!");
            window.history.back();
        </script>';
    } else {
        die("ID desconhecido ou não enviado.");
    }
} catch(PDOException $e) {
    die("Falha ao deletar registro: " . $e->getMessage());
    exit;
}
```

## 8. Criando Layout da página de login e cadastro

Crie um arquivo com o nome **cadastro.php** dentro da pasta do **projeto\_versatil** como mostra a imagem abaixo:



### cadastro.php

```
<?php require_once __DIR__ . "/app/autoloader.php"; ?>

<?php UserService::proibirNaoLogado(); ?>
<?php include __DIR__ . "/extra/header.php" ?>
<?php include __DIR__ . "/extra/menu.php" ?>

<main class="container" data-spy="scroll" data-target="#navbarContent" data-offset="0">
    <fieldset id="novo" class="bg-light p-3">
        <legend>Cadastrar um novo Contato</legend>
        <?php include __DIR__ . "/extra/formularioContato.php" ?>
    </fieldset>
</main>

<?php include __DIR__ . "/extra/footer.php" ?>
```

Em seguida crie um arquivo com o nome **index.php** dentro da pasta do **projeto\_versatil**.

## index.php

```

1  <?php require_once __DIR__ . "/app/autoloader.php"; ?>
2
3  <?php include __DIR__ . "/extra/header.php" ?>
4
5  <main class="container">
6      <div class="pt-5 row justify-content-center">
7
8          <div class="col-9 col-sm-6 col-md-4 col-lg-3">
9
10             
11             <h3>Entre com as suas credenciais:</h3>
12
13             <p class="alert alert-info alert-dismissible fade show">
14                 Usuário: <br>
15                 <strong>versatil@versatilti.com.br</strong>
16                 <br>
17                 Senha: <br>
18                 <strong>admin2022</strong>
19
20                 <button type="button" class="close" data-dismiss="alert" aria-label="Close">
21                     <span aria-hidden="true">&times;</span>
22                 </button>
23             </p>
24
25             <form method="POST" class="needs-validation" novalidate>
26                 <div class="form-group">
27                     <label for="usuario">Usuário:</label>
28                     <input type="email" name="usuario" required class="form-control">
29                     <div class="invalid-feedback">Seu usuário deve ser um e-mail válido</div>
30                 </div>
31
32                 <div class="form-group">
33                     <label for="senha">Senha:</label>
34                     <input type="password" name="senha" required class="form-control" minlength="8">
35                     <div class="invalid-feedback">Sua senha deve possuir ao menos 8 dígitos</div>
36                 </div>
37                 <button type="submit" class="btn btn-primary w-100 mb-3">
38                     Acessar a conta
39                 </button>
40             </form>
41
42             <?php
43             $user = new UserService;
44
45             if (isset($_POST['usuario']) and isset($_POST['senha'])) {
46
47                 $res = $user->login($_POST['usuario'], $_POST['senha']);
48                 switch((bool) $res) {
49                     case true:
50                         echo '<script>
51                             alert("Acesso autorizado pelo sistema.");
52                             window.location.href = "home.php";
53                         </script>';
54                         exit;
55                         break;
56                     default:
57                         echo '<p class="alert alert-danger">Usuário e/ou senha inválidos</p>';
58                     }
59                 }
60             ?>
61
62             </div>
63         </div>
64     </main>
65
66     <?php include __DIR__ . "/extra/footer.php" ?>
67

```

## 9. Criando arquivos de listagem de cadastros

Crie um arquivo como o nome **detalhes.php** dentro da pasta do **projeto\_versatil**.

### **detalhes.php**

```
<?php require_once __DIR__ . "/app/autoloader.php"; ?>

<?php UserService::proibirNaoLogado(); ?>
<?php include __DIR__ . "/extra/header.php" ?>
<?php include __DIR__ . "/extra/menu.php" ?>

<main class="container" data-spy="scroll" data-target="#navbarContent" data-offset="0">

    <fieldset id="listar">
        <legend>Detalhes do Contato</legend>
        <?php include __DIR__ . "/extra/formularioEditaContato.php" ?>
    </fieldset>

</main>

<?php include __DIR__ . "/extra/footer.php" ?>
```



Crie um arquivo com o nome **tabelaContato.php** dentro da pasta **extra**.

### **tabelaContato.php**

```
<div class="row">

  <div class="table-responsive">
    <table class="table table-hover table-striped">
      <tbody>
        <?php
          $contatosDao = new ContatosDAO;

          if (isset($_GET['search'])) {
            $contatos = $contatosDao->buscarTodosPorPesquisa($_GET['search']);
          } else {
            $contatos = $contatosDao->buscarTodos();
          }

          foreach($contatos as $contato) {
            $nome = utf8_encode($contato['nome']);
            echo "
            <tr>
              <td>{$nome}</td>
              <td>{$contato['email']}</td>
              <td>{$contato['telefone']}</td>
              <td>
                <a class="btn btn-link" href="actions/apagar.php?id={$contato['id']}">Apagar</a>
                <a class="btn btn-link" href="detalhes.php?id={$contato['id']}">Ver Detalhes</a>
              </td>
            </tr>";
          }
        <?>
      </tbody>
    </table>
  </div>
</div>
```

Agora acesse o seu projeto no **localhost**, se aparecer como a imagem abaixo solicite ao instrutor(a) os arquivos de estilização para mudar a interface para finalizar o projeto.

#### PROJETO FINALIZADO

(SEM ESTILIZAÇÃO)



Entre com as suas credenciais:

Usuario:  
**demo@versatildf.com.br**  
 Senha:  
**#SouVersatil2022** ×

Usuário:   
 Seu usuário deve ser um e-mail válido  
 Senha:   
 Sua senha deve possuir ao menos 8 dígitos

#### PROJETO FINALIZADO

(COM ESTILIZAÇÃO)



Entre com as suas credenciais:

Usuario: ×  
**demo@versatildf.com.br**  
 Senha:  
**#SouVersatil2022**

Usuário:   
 Senha: