PROJETO T-ADMINISTRATOR – ADMINISTRAÇÃO DE EMPRESA

**1ª TELA:**

ARQUIVO: FORNECEDOR.PHP

DESCRIÇÃO: Formulário de início, utilizado para entrada de dados e visualização de fornecedores.

LOCAL: Raiz da pasta Administrativo(root)

Código:

<form action="/src/cadastro\_fornecedor.php" method="post">

    <label for="nome">id:</label>

    <input type="text" name="id" value="<?php echo $\_POST['id'] ?? ''; ?>" id="id" >

    <label for="nome">Nome:</label>

    <input type="text" name="nome" value="<?php echo $\_POST['nome'] ?? ''; ?>" id="nome" >

    <label for="cnpj">CNPJ:</label>

    <input type="text" name="cnpj" value="<?php echo $\_POST['cnpj'] ?? ''; ?>" id="cnpj" >

    <label for="email">E-mail:</label>

    <input type="email" name="email" value="<?php echo $\_POST['email'] ?? ''; ?>" id="email" >

    <label for="telefone">Telefone:</label>

    <input type="tel" name="telefone" value="<?php echo $\_POST['telefone'] ?? ''; ?>" id="telefone" >

    <label for="endereco">Endereço:</label>

    <input type="text" name="endereco" value="<?php echo $\_POST['endereco'] ?? ''; ?>" id="endereco" >

    <br><br>

    <input class="buttom" name="CadSentenca" type="submit" value="Cadastrar">

    <input class="buttom" name="AltSentenca" type="submit" value="Alterar">

    <input class="buttom" name="DelSentenca" type="submit" value="Excluir">

    <input class="buttom" name="ListaSentenca" type="submit" value="Lista Todas">

</form>

**2ª TELA:**

ARQUIVO: CADASTRO\_FORNECEDOR.PHP

DESCRIÇÃO: Arquivo de troca entre o front end (formulário de início, arquivo FORNECEDOR.PHP), e o backend (Camada Controller, arquivo FORNECEDOR\_CONTROLLER.PHP).

LOCAL: Pasta src/cadastro\_fornecedor.php

Código:

<?php

session\_start();

include\_once('../routes/FornecedorController.php');

require\_once('../utils/Conexao.php');

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    $id = $\_POST["id"] ?? null;

    $nome = $\_POST["nome"];

    $cnpj = $\_POST["cnpj"];

    $email = $\_POST["email"];

    $telefone = $\_POST["telefone"];

    $endereco = $\_POST["endereco"];

    $fornecedorController = new FornecedorController();

    // Chamada dos métodos do FornecedorController

    $CadSentenca = filter\_input(INPUT\_POST, 'CadSentenca');

    if (isset ($CadSentenca)) {

        $fornecedorController->criarFornecedor($id, $nome, $cnpj, $email, $telefone, $endereco);

    }

    $ListaSentenca = filter\_input(INPUT\_POST, 'ListaSentenca');

    if (isset ($ListaSentenca)) {

        $fornecedorController->buscarFornecedor($id, $nome, $cnpj, $email, $telefone, $endereco);

    }

    $AltSentenca = filter\_input(INPUT\_POST, 'AltSentenca');

    if (isset($AltSentenca)) {

         $fornecedorController->atualizarFornecedor($id, $nome, $cnpj, $email, $telefone, $endereco);

    }

    $DelSentenca = filter\_input(INPUT\_POST, 'DelSentenca');

    if (isset ($DelSentenca)) {

        $fornecedorController->deletarFornecedor($id);

    }

    // Redirecionar para uma página de sucesso ou exibir uma mensagem ao usuário

    //header("Location: sucesso.php");

    //exit();

}

?>

**3ª TELA:**

ARQUIVO: FORNECEDOR\_CONTROLLER.PHP

DESCRIÇÃO: Arquivo que recebe os dados do arquivo cadastro\_fornecedor.php e envia ao arquivo FornecedorDAO.php.

LOCAL: Pasta routes/FornecedorController.php

Código:

<?php

require\_once('../utils/Conexao.php');

require\_once('../dao/FornecedorDAO.php');

class FornecedorController {

    private $fornecedorDAO;

    public function \_\_construct() {

        $this->fornecedorDAO = new FornecedorDAO(Conexao::getConnection());

    }

    public function criarFornecedor($id, $nome, $cnpj, $email, $telefone, $endereco) {

        return $this->fornecedorDAO->create($id, $nome, $cnpj, $email, $telefone, $endereco);

    }

    public function buscarFornecedor($id, $nome, $cnpj, $email, $telefone, $endereco) {

        return $this->fornecedorDAO->read($id, $nome, $cnpj, $email, $telefone, $endereco);

    }

    public function atualizarFornecedor($id, $nome, $cnpj, $email, $telefone, $endereco) {

        return $this->fornecedorDAO->update($id, $nome, $cnpj, $email, $telefone, $endereco);

    }

    public function deletarFornecedor($id) {

       return $this->fornecedorDAO->delete($id);

    }

}

?>

**4ª TELA:**

ARQUIVO: CONEXAO.PHP

DESCRIÇÃO: Arquivo que processa a conexão com o banco de dados.

LOCAL: Pasta utils/Conexao.php

Código:

<?php

// Configurações de acesso ao banco de dados

class Conexao {

    private static $db\_host = "localhost";

    private static $db\_user = "root";

    private static $db\_pass = "10205618";

    private static $db\_name = "administrativo";

    public static function getConnection() {

        $conn = "mysql:host=" . self::$db\_host . ";dbname=" . self::$db\_name;

        try {

            $conexao = new PDO($conn, self::$db\_user, self::$db\_pass);

            $conexao->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

            return $conexao;

        } catch (PDOException $e) {

            die("Falha na conexão com o banco de dados: " . $e->getMessage());

        }

    }

}

?>

**5ª TELA:**

ARQUIVO: FORNECEDORDAO.PHP

DESCRIÇÃO: Arquivo que recebe os dados do arquivo FornecedorController.php, implementa a interface FornecedorDAOinterface.php e envia e retorna o processamento ao Banco de dados.

LOCAL: Pasta dao/FornecedorDao.php

Código:

<?php

require\_once('FornecedorDAOinterface.php');

class FornecedorDAO implements FornecedorDAOinterface {

    private $connection;

    public function \_\_construct($connection) {

        $this->connection = $connection;

    }

public function create($id, $nome, $cnpj, $email, $telefone, $endereco) {

    // Lógica para criar um registro no banco de dados

    $sql = "INSERT INTO fornecedores (nome

                                    , cnpj

                                    , email

                                    , telefone

                                    , endereco

                                    )

                            VALUES (:nome

                                    , :cnpj

                                    , :email

                                    , :telefone

                                    , :endereco

                                    )";

    $stmt = $this->connection->prepare($sql);

    $stmt->bindParam(":nome", $nome);

    $stmt->bindParam(":cnpj", $cnpj);

    $stmt->bindParam(":email", $email);

    $stmt->bindParam(":telefone", $telefone);

    $stmt->bindParam(":endereco", $endereco);

    if ($stmt->execute()) {

        echo "Fornecedor cadastrado com sucesso!";

    } else {

        echo "Erro ao cadastrar fornecedor: " . $stmt->errorInfo()[2];

    }

}

    public function read($id, $nome, $cnpj, $email, $telefone, $endereco) {

    // Lógica para ler um registro do banco de dados

        $sql = "SELECT \* FROM fornecedores";

        $stmt = $this->connection->prepare($sql);

        //$stmt->bindParam(":id", $id, PDO::PARAM\_INT);

        $stmt->execute();

        $result = $stmt->fetchAll(PDO::FETCH\_ASSOC);

    if ($result) {

        echo "<table>";

        echo "<tr>";

        echo "<th><strong>NUMERO</strong></th>";

        echo "<th><strong>NOME</strong></th>";

        echo "<th><strong>CNPJ</strong></th>";

        echo "<th><strong>EMAIL</strong></th>";

        echo "<th><strong>TELEFONE</strong></th>";

        echo "<th><strong>ENDEREÇO</strong></th>";

        echo "</tr>";

        foreach ($result as $row) {

            $id = $row['id'];

            $nome = $row['nome'];

            $cnpj = $row['cnpj'];

            $email = $row['email'];

            $telefone = $row['telefone'];

            $endereco = $row['endereco'];

            echo "<tr>";

            echo "<td>$id</td>";

            echo "<td>$nome</td>";

            echo "<td>$cnpj</td>";

            echo "<td>$email</td>";

            echo "<td>$telefone</td>";

            echo "<td>$endereco</td>";

            echo "</tr>";

        }

        echo "</table>";

    } else {

        return null;

    }

}

    public function update($id, $nome, $cnpj, $email, $telefone, $endereco) {

    // Lógica para atualizar um registro no banco de dados

    $sql = "UPDATE fornecedores SET

                                nome = :nome

                              , cnpj = :cnpj

                              , email = :email

                              , telefone = :telefone

                              , endereco = :endereco

                              WHERE id = :id";

    $stmt = $this->connection->prepare($sql);

    $stmt->bindParam(":nome", $nome);

    $stmt->bindParam(":cnpj", $cnpj);

    $stmt->bindParam(":email", $email);

    $stmt->bindParam(":telefone", $telefone);

    $stmt->bindParam(":endereco", $endereco);

    $stmt->bindParam(":id", $id);

    if ($stmt->execute()) {

        echo "Fornecedor atualizado com sucesso!";

    } else {

        echo "Erro ao atualizar fornecedor: " . $stmt->errorInfo()[2];

    }

}

    public function delete($id) {

    // Lógica para deletar um registro do banco de dados

        $sql = "DELETE FROM fornecedores WHERE :id = $id";

        $stmt = $this->connection->prepare($sql);

        $stmt->bindParam(":id", $id);

        if ($stmt->execute()) {

        echo "Fornecedor deletado com sucesso!";

    } else {

        echo "Erro ao deletar fornecedor: " . $stmt->errorInfo()[2];

    }

}

}

**6ª TELA:**

ARQUIVO: FORNECEDORDAOINTERFACE.PHP

DESCRIÇÃO: Arquivo que recebe os dados e processa os arquivos FornecedorDAO.php e FornecedorController.php.

LOCAL: Pasta dao/FornecedorDaointerface.php.

Código:

<?php

//require\_once('utils/Conexao.php');

//require\_once('routes/FornecedorController.php');

interface FornecedorDAOinterface {

    public function create($id,$nome, $cnpj, $email, $telefone, $endereco);

    public function read($id,$nome, $cnpj, $email, $telefone, $endereco);

    public function update($id, $nome, $cnpj, $email, $telefone, $endereco);

    public function delete($id);

}