Atividade 10

Nome: Debi Junia de Paula.

O código implementa um sistema web completo de gerenciamento de usuários, composto por uma interface HTML, um script PHP para processamento dos dados e um banco de dados MySQL para armazenamento. Ele oferece funcionalidades de cadastro, listagem, edição e exclusão de registros, garantindo uma interação fluida entre a interface e o backend por meio de requisições assíncronas realizadas com a Fetch API.

Na interface HTML, há um formulário contendo campos de entrada para "Nome" e "Email", ambos obrigatórios, e um botão para envio dos dados. Uma tabela é utilizada para exibir os registros cadastrados, organizados em colunas que incluem ID, Nome, Email e Ações. Através de botões específicos, o usuário pode editar ou excluir registros diretamente da interface. O JavaScript é responsável por gerenciar as interações com o servidor, enviando e recebendo dados para manter a tabela sincronizada com o banco de dados.

O script PHP atua como o backend do sistema, processando diferentes tipos de requisições HTTP. Ele permite a inserção de novos registros com o método POST, o retorno de todos os registros com GET, a atualização de dados com PUT e a exclusão de registros com DELETE. A classe Conexao é utilizada para estabelecer uma conexão segura com o banco de dados MySQL, utilizando PDO para proteger contra ataques de injeção SQL e facilitar o tratamento de erros.

O banco de dados MySQL, chamado atividade11, contém uma tabela registros estruturada com três colunas: ID (chave primária e auto-incremento), Nome e Email. Scripts SQL são fornecidos para criar e inicializar a base de dados e a tabela, além de executar as operações CRUD (Create, Read, Update, Delete). Durante o carregamento inicial da página, os registros já existentes na tabela são automaticamente buscados e exibidos na interface.

index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Cadastro e Listagem</title>
    <link rel="stylesheet" href="style.css">
```

```
</head>

<body>
    <h1>Cadastro de Dados</h1>
    <form id="form-cadastro">
        <label for="nome">Nome:</label>
        <input type="text" id="nome" name="nome" required>
        <br><br>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>
        <br><br>
        <button type="submit">Cadastrar</button>
    </form>

    <h2>Lista de Registros</h2>
    <table>
        <thead>
            <tr>
                <th>ID</th>
                <th>Nome</th>
                <th>Email</th>
                <th>Ações</th>
            </tr>
        </thead>
        <tbody id="registros"></tbody>
    </table>

    <script>
        async function carregarRegistros() {
            try {
                const resposta = await fetch('processa.php');
                const registros = await resposta.json();
                const tabela = document.getElementById('registros');
                tabela.innerHTML = registros.map(registro => `
                    <tr>
                        <td>${registro.id}</td>
                        <td>${registro.nome}</td>
                        <td>${registro.email}</td>
                        <td>
                            <button class="btn-editar"
onclick="editarRegistro(${registro.id}, '${registro.nome}',
'${registro.email}')">Editar</button>
                            <button class="btn-excluir"
onclick="excluirRegistro(${registro.id})">Excluir</button>
                        </td>
                    </tr>
                `).join('');
            } catch (erro) {
                console.error('Erro ao carregar registros:', erro);
```

```
            }
        }

        async function editarRegistro(id, nome, email) {
            const novoNome = prompt('Novo nome:', nome);
            const novoEmail = prompt('Novo email:', email);
            if (novoNome && novoEmail) {
                try {
                    const resposta = await fetch('processa.php', {
                        method: 'PUT',
                        headers: { 'Content-Type': 'application/json' },
                        body: JSON.stringify({ id, nome: novoNome, email:
novoEmail })
                    });
                    const resultado = await resposta.json();
                    if (resultado.sucesso) {
                        alert('Registro atualizado com sucesso!');
                        carregarRegistros();
                    } else {
                        alert('Erro ao atualizar registro.');
                    }
                } catch (erro) {
                    console.error('Erro ao editar registro:', erro);
                }
            }
        }

        async function excluirRegistro(id) {
            if (confirm('Deseja realmente excluir este registro?')) {
                try {
                    const resposta = await fetch('processa.php', {
                        method: 'DELETE',
                        headers: { 'Content-Type': 'application/json' },
                        body: JSON.stringify({ id })
                    });
                    const resultado = await resposta.json();
                    if (resultado.sucesso) {
                        alert('Registro excluído com sucesso!');
                        carregarRegistros();
                    } else {
                        alert('Erro ao excluir registro.');
                    }
                } catch (erro) {
                    console.error('Erro ao excluir registro:', erro);
                }
            }
        }
```

```
        document.getElementById('form-
cadastro').addEventListener('submit', async function (evento) {
            evento.preventDefault();
            const formData = new FormData(evento.target);

            try {
                const resposta = await fetch('processa.php', {
                    method: 'POST',
                    body: formData
                });
                const resultado = await resposta.json();
                if (resultado.sucesso) {
                    alert('Cadastro realizado com sucesso!');
                    carregarRegistros();
                    evento.target.reset();
                } else {
                    alert('Erro ao cadastrar.');
                }
            } catch (erro) {
                console.error('Erro ao enviar dados:', erro);
            }
        });

        carregarRegistros();
    </script>
</body>

</html>
```

processa.php

```
<?php
require_once 'Conexao.php';

try {
    $pdo = Conexao::getConexao();

    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $nome = $_POST['nome'] ?? '';
        $email = $_POST['email'] ?? '';
        if ($nome && $email) {
            $stmt = $pdo->prepare('INSERT INTO registros (nome, email)
VALUES (:nome, :email)');
            $stmt->execute(['nome' => $nome, 'email' => $email]);
            echo json_encode(['sucesso' => true]);
        } else {
            echo json_encode(['sucesso' => false, 'mensagem' => 'Campos
obrigatórios ausentes.']);
```

```
        }
        exit;
    }

    if ($_SERVER['REQUEST_METHOD'] === 'PUT') {
        $dados = json_decode(file_get_contents('php://input'), true);
        $stmt = $pdo->prepare('UPDATE registros SET nome = :nome, email =
:email WHERE id = :id');
        $stmt->execute(['nome' => $dados['nome'], 'email' =>
$dados['email'], 'id' => $dados['id']]);
        echo json_encode(['sucesso' => true]);
        exit;
    }

    if ($_SERVER['REQUEST_METHOD'] === 'DELETE') {
        $dados = json_decode(file_get_contents('php://input'), true);
        $stmt = $pdo->prepare('DELETE FROM registros WHERE id = :id');
        $stmt->execute(['id' => $dados['id']]);
        echo json_encode(['sucesso' => true]);
        exit;
    }

    $stmt = $pdo->query('SELECT * FROM registros');
    $registros = $stmt->fetchAll(PDO::FETCH_ASSOC);
    echo json_encode($registros ?: []);
} catch (PDOException $e) {
    echo json_encode(['erro' => $e->getMessage()]);
}
```

conexão.php

```
<?php
class Conexao
{
    private static $dsn =
'mysql:host=localhost;port=3306;dbname=atividade11;charset=utf8';
    private static $usuario = 'root';
    private static $senha = 'root';
    private static $conexao = null;

    // Função para conectar ao banco de dados
    private static function conecta()
    {
        if (self::$conexao === null) {
            self::$conexao = new PDO(
                self::$dsn,
                self::$usuario,
                self::$senha
```

```
            );
            self::$conexao->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
        }
    }

    // Retorna a conexão ativa
    public static function getConexao()
    {
        self::conecta();
        return self::$conexao;
    }

    // Prepara um comando SQL
    public static function preparaComando($sql)
    {
        self::conecta();
        return self::$conexao->prepare($sql);
    }
}
?>
```

style.css

```
body {
    font-family: Arial, sans-serif;
    margin: 20px;
    background-color: #f4f4f4;
    color: #333;
}

h1, h2 {
    color: #333;
    text-align: center;
}

form {
    margin: 0 auto 20px auto;
    background: #fff;
    padding: 20px;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    max-width: 400px;
}

label {
    font-weight: bold;
    display: block;
    margin-top: 10px;
```

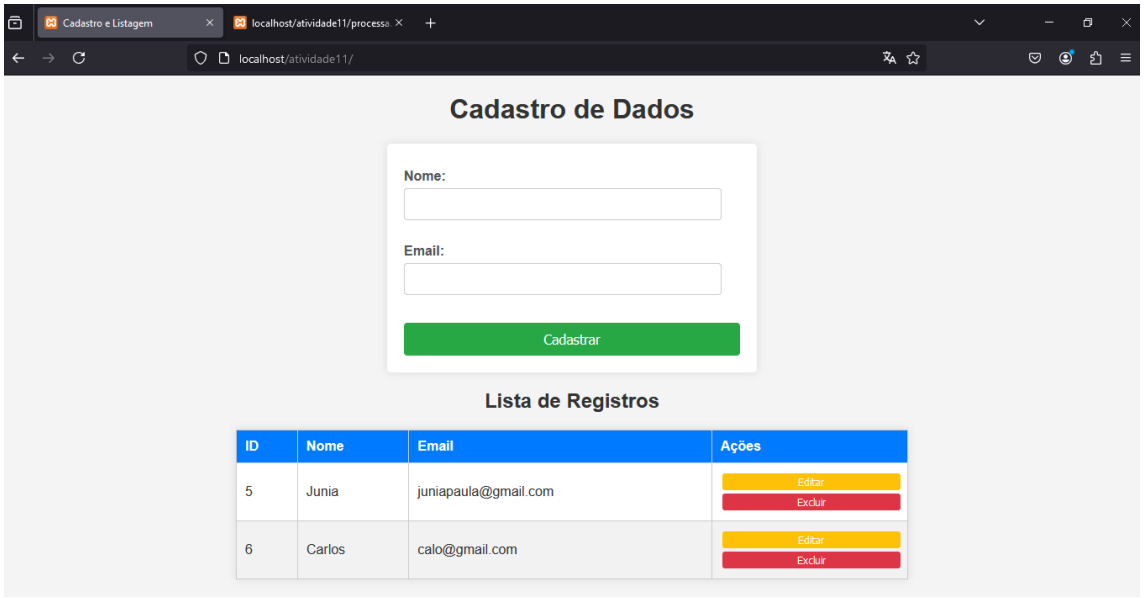
```
    color: #555;
}

input {
    padding: 10px;
    width: calc(100% - 22px);
    margin-top: 5px;
    border: 1px solid #ccc;
    border-radius: 4px;
    box-sizing: border-box;
}

button {
    background-color: #28a745;
    color: #fff;
    border: none;
    padding: 10px 15px;
    cursor: pointer;
    border-radius: 4px;
    font-size: 16px;
    width: 100%;
    margin-top: 15px;
    transition: background-color 0.3s ease;
}

button:hover {
    background-color: #218838;
}

table {
    width: 100%;
    border-collapse: collapse;
    background: #fff;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    margin: 0 auto;
    max-width: 800px;
}

th, td {
    border: 1px solid #ccc;
    padding: 10px;
    text-align: left;
}

th {
    background-color: #007bff;
    color: white;
    font-weight: bold;
}
```

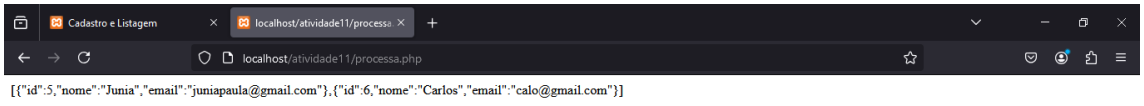
```css
tr:nth-child(even) {
    background-color: #f2f2f2;
}

tr:hover {
    background-color: #ddd;
}

@media (max-width: 600px) {
    form {
        max-width: 100%;
    }

    input, button {
        width: 100%;
    }

    table {
        font-size: 14px;
    }
}

.btn-editar, .btn-excluir {
    padding: 3px 5px;
    font-size: 12px;
    margin: 2px;
    border: none;
    border-radius: 3px;
    cursor: pointer;
}

.btn-editar {
    background-color: #ffc107;
    color: #fff;
}

.btn-editar:hover {
    background-color: #e0a800;
}

.btn-excluir {
    background-color: #dc3545;
    color: #fff;
}

.btn-excluir:hover {
    background-color: #c82333;
}
```

atividade11.sql

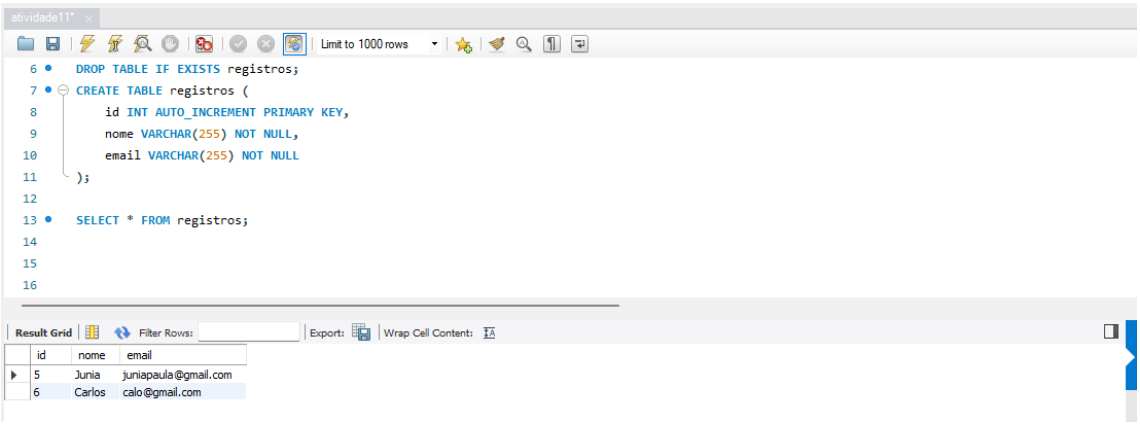
```sql
DROP DATABASE IF EXISTS atividade11;
CREATE DATABASE atividade11;

USE atividade11;

DROP TABLE IF EXISTS registros;
CREATE TABLE registros (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL
);

SELECT * FROM registros;
```

Saída

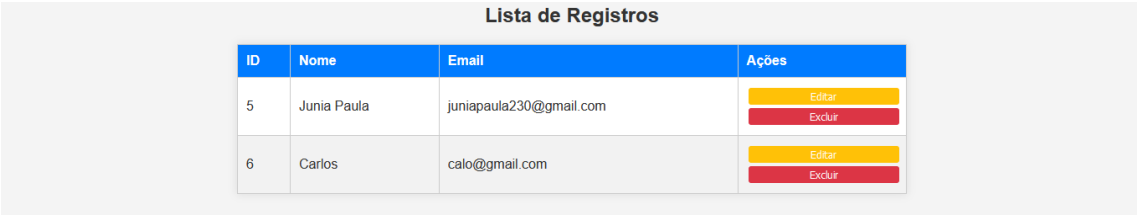


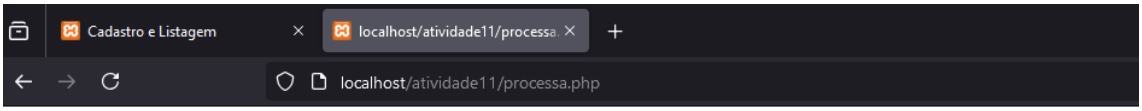


[{"id":5,"nome":"Junia","email":"juniapaula@gmail.com"},{"id":6,"nome":"Carlos","email":"calo@gmail.com"}]



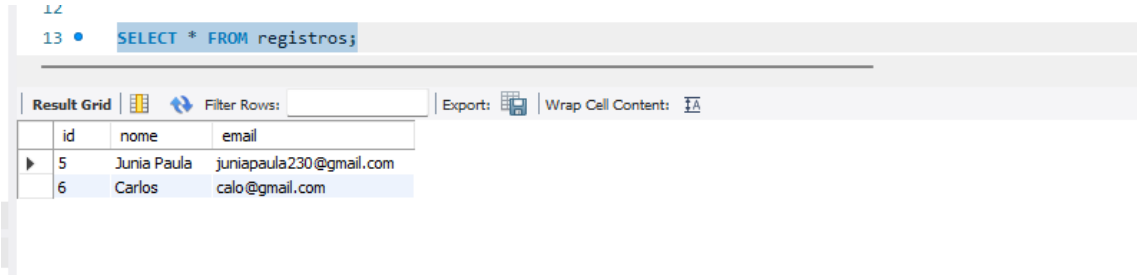
Edição

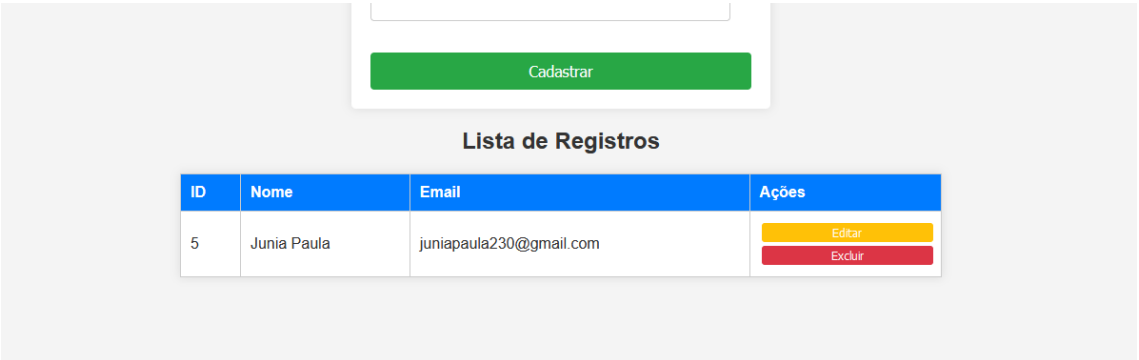


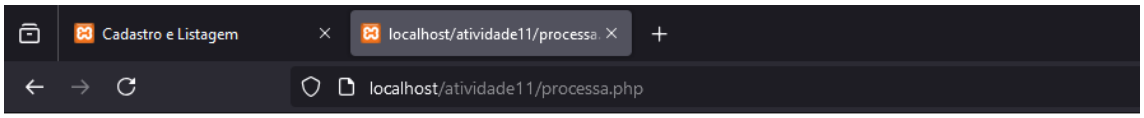


[{"id":5,"nome":"Junia Paula","email":"juniapaula230@gmail.com"},{"id":6,"nome":"Carlos","email":"calo@gmail.com"}]

Exclusão







[{"id":5,"nome":"Junia Paula","email":"juniapaula230@gmail.com"}]