
DOCUMENTAÇÃO TÉCNICA: SISTEMA CRUD EM CODEIGNITER3

1. VISÃO GERAL

Este sistema foi desenvolvido utilizando o framework **PHP CodeIgniter 3** com banco de dados **MySQL**. O objetivo principal é a gestão de registros vinculados a usuários, utilizando um controle de permissões baseado em dois níveis de acesso: **Administrador** e **Comum**.

2. MODELAGEM DE DADOS

O banco de dados é composto por duas tabelas principais relacionadas para garantir a integridade e a auditoria das informações.

2.1 Tabela: usuários

Responsável pelo armazenamento de credenciais e definições de privilégios.

- id**: Identificador único (Chave Primária).
- nome**: Nome completo do colaborador.
- e-mail**: Identificador de login único.
- senha**: Hash de segurança criptografado via algoritmo Bcrypt.
- nivel_acesso**: Define as permissões (admin ou comum).
- admin_criador_id**: Chave Estrangeira que identifica o administrador responsável pela criação da conta.

2.2 Tabela: cadastros

Responsável pelo armazenamento dos registros operacionais.

- id**: Identificador único (Chave Primária).
- nome, endereco, bairro**: Atributos descritivos do registro.
- usuario_id**: Chave Estrangeira que vincula o registro ao autor da inserção.

3. MATRIZ DE PERMISSÕES

O controle de acesso é validado em cada requisição através de sessões ativas no servidor.

Funcionalidade	Usuário Comum	Administrador
Autenticação (Login/Logout)	Sim	Sim
Listar próprios cadastros	Sim	Sim
Criar e Editar cadastros	Sim	Sim
Visualizar cadastros de terceiros	Não	Sim
Filtrar registros por usuário	Não	Sim
Gestão de Usuários (CRUD)	Não	Sim
Exclusão de Registros	Não	Sim (Requer Senha Master)

4. SEGURANÇA E REGRAS DE NEGÓCIO

4.1 Autenticação e Sessão

O sistema utiliza a função `password_hash()` para garantir que senhas originais não sejam armazenadas. O controle de sessão é implementado no construtor dos controladores, redirecionando usuários não autenticados para a interface de login.

4.2 Integridade na Exclusão

Para evitar registros órfãos, o sistema aplica a **Transferência de Custódia**. Ao remover um usuário, todos os seus cadastros vinculados são atualizados para o ID do administrador que executa a ação, preservando o histórico antes da deleção do perfil.

4.3 Senha Master

Ações críticas, especificamente a exclusão de contas de usuários, exigem a confirmação da senha mestre `qwaszx123`, mitigando riscos de deleções acidentais.

5. ARQUITETURA DE SOFTWARE (MVC)

O sistema segue o padrão **Model-View-Controller**, garantindo separação de responsabilidades.

- **Módulo Auth:** Responsável pela porta de entrada. O controlador valida as entradas e o modelo executa a verificação de hash via `password_verify`.
- **Módulo Cadastro:** Gerencia o fluxo operacional. O controlador filtra a visualização de dados de acordo com o nível de acesso contido na sessão.
- **Módulo Usuario:** Área restrita a perfis administrativos. Implementa travas de segurança e gerencia a lógica de transferência de registros.

6. INSTALAÇÃO DO BANCO DE DADOS

6.1 Criação do Schema

SQL

```
CREATE DATABASE sistema_gestao CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
```

6.2 Estrutura das Tabelas

SQL

```
CREATE TABLE usuarios (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL,  
    email VARCHAR(100) NOT NULL UNIQUE,  
    senha VARCHAR(255) NOT NULL,  
    nivel_acesso ENUM('admin', 'comum') DEFAULT 'comum',  
    admin_criador_id INT NULL,  
    data_criacao TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (admin_criador_id) REFERENCES usuarios(id) ON DELETE SET NULL  
);
```

```
CREATE TABLE cadastros (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(150) NOT NULL,  
    endereco VARCHAR(255),  
    bairro VARCHAR(100),  
    usuario_id INT NOT NULL,  
    data_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (usuario_id) REFERENCES usuarios(id)  
);
```

7. FLUXO DE NAVEGAÇÃO

1. **Acesso Inicial:** Validação de sessão em auth/index.
2. **Dashboard:** Após login, redirecionamento para cadastro/index.
3. **Operação:** Inserções e edições via métodos salvar e editar.
4. **Administração:** Acesso a usuario/index filtrado por nível de privilégio.
5. **Finalização:** Método logout encerra a sessão e limpa dados temporários.