

Documentação Técnica – Sistema de Gestão de Condomínios

Arquitetura

- Camada de GUI: Tkinter (janelas `LoginWindow`, `Dashboard` e módulos de cadastro)
- Camada de Dados: classe `Database`
([database.py](file:///c:/Users/afons/Documents/trae_projects/condominio/database.py))
- Driver MySQL: `mysql-connector-python`
- Configuração: variáveis de ambiente `DB_HOST`, `DB_USER`, `DB_PASSWORD`, `DB_NAME`

Fluxo de Login

1. Usuário informa credenciais
2. `LoginWindow.login()` instancia `Database`
3. `Database.verify_user(username, password)` valida via hash SHA-256
4. Em caso de sucesso, abre `Dashboard` com papel e vínculo de condomínio

Controle de Acesso

- Papel do usuário (coluna `nivel`): `admin` ou `sindico`
- `Dashboard.create_menu()` exibe menus conforme papel
- Janelas (Moradores, Gastos, Obras, Funcionários) filtram por `condominio_id` quando papel é `sindico`

Banco de Dados (MySQL)

Schema criado automaticamente na inicialização:

- `condominios`
 - `id` INT PK AUTO_INCREMENT
 - `nome` VARCHAR(255) NOT NULL
 - `endereco` VARCHAR(255) NOT NULL
- `usuarios`
 - `id` INT PK AUTO_INCREMENT
 - `username` VARCHAR(255) UNIQUE NOT NULL
 - `password_hash` VARCHAR(64) NOT NULL
 - `nivel` VARCHAR(20) NOT NULL
 - `condominio_id` INT NULL FK -> `condominios(id)` ON DELETE SET NULL
- `moradores`
 - `id` INT PK AUTO_INCREMENT
 - `nome` VARCHAR(255) NOT NULL
 - `unidade` VARCHAR(100) NOT NULL

- `contato` VARCHAR(255)
- `condominio_id` INT NOT NULL FK -> `condominios(id)` ON DELETE CASCADE
- `valor_condominio` DECIMAL(10,2)
- `gastos`
- `id` INT PK AUTO_INCREMENT
- `descricao` VARCHAR(255) NOT NULL
- `valor` DECIMAL(10,2) NOT NULL
- `data` DATE NOT NULL
- `condominio_id` INT NOT NULL FK -> `condominios(id)` ON DELETE CASCADE
- `obras`
- `id` INT PK AUTO_INCREMENT
- `descricao` VARCHAR(255) NOT NULL
- `custo` DECIMAL(10,2) NOT NULL
- `status` VARCHAR(50) NOT NULL
- `condominio_id` INT NOT NULL FK -> `condominios(id)` ON DELETE CASCADE
- `funcionarios`
- `id` INT PK AUTO_INCREMENT
- `nome` VARCHAR(255) NOT NULL
- `cargo` VARCHAR(100) NOT NULL
- `salario` DECIMAL(10,2)
- `condominio_id` INT NOT NULL FK -> `condominios(id)` ON DELETE CASCADE

Operações Principais (Database)

- Usuários:

- `add_user(username, password, nivel, condominio_id)`
- `verify_user(username, password)` retorna `(id, username, nivel, condominio_id)`

- Condomínios:

- `add_condominio(nome, endereco)`

- `get_condominios()`

- Moradores:

- `add_morador(nome, unidade, contato, condominio_id, valor)`
- `get_moradores(condominio_id=None)`

- Gastos:

- `add_gasto(descricao, valor, data, condominio_id)`
- `get_gastos(condominio_id=None)`

- Obras:

- `add obra(descricao, custo, status, condominio_id)`
- `get_oberas(condominio_id=None)`

- Funcionários:

- `add_funcionario(nome, cargo, salario, condominio_id)`
- `get_funcionarios(condominio_id=None)`

Execução e Configuração

- Instalação do driver:
- Windows: `py -m pip install mysql-connector-python`
- Variáveis de ambiente:
- `DB_HOST`, `DB_USER`, `DB_PASSWORD`, `DB_NAME`
- Inicialização:
- `py main.py`

Segurança

- Hash de senha: SHA-256
- Não armazenar senhas em texto plano
- Recomendado usar usuário MySQL dedicado e restringir privilégios a `condominio.*`