



# RESUMO COMPLETO DO PROJETO - SISTEMA DE GERENCIAMENTO DE DISPOSITIVOS



## DESCRIÇÃO DO PROJETO

Sistema web completo para gerenciamento de dispositivos de rede (switches, roteadores, servidores) com operações CRUD (Create, Read, Update, Delete) integrado com MySQL.



## ARQUITETURA E TECNOLOGIAS

### Backend

- **Python + Flask** - Framework web principal
- **MySQL Connector** - Conexão com banco de dados
- **Jinja2** - Sistema de templates

### Frontend

- **HTML5** - Estrutura das páginas
- **CSS3** - Estilização e responsividade
- **JavaScript** - Confirmações e interatividade

### Banco de Dados

- **MySQL** - Armazenamento dos dispositivos
- **Tabela devices** - Campos: id, device\_name, device\_type, ip\_address, vlan, configuration, notes



## ESTRUTURA DO PROJETO

rack\_management/

├─ app.py (Backend Flask + Rotas)

├─ static/

```
|   └─ style.css (Estilos CSS)
└─ templates/
    ├── base.html (Layout principal)
    ├── index.html (Lista dispositivos)
    ├── add_device.html (Formulário adicionar)
    └─ edit_device.html (Formulário editar)
```

## FLUXO DA APLICAÇÃO

### Rotas e Funcionalidades

- 1./ **(GET)** - Lista todos os dispositivos
- 2./add **(GET)** - Mostra formulário de adição
- 3./add **(POST)** - Processa dados do formulário
- 4./edit/<id> **(GET)** - Mostra formulário de edição
- 5./edit/<id> **(POST)** - Processa atualização
- 6./delete/<id> **(POST)** - Remove dispositivo

### Características Técnicas

- Conexão por requisição** - Evita problemas de concorrência
- Tratamento de erros** - Try-catch em operações de banco
- Flash messages** - Feedback visual para usuário
- Validação básica** - Campos obrigatórios
- Design responsivo** - Funciona em mobile e desktop
- Segurança** - DELETE via POST, confirmações JavaScript

## BOAS PRÁTICAS IMPLEMENTADAS

- Herança de templates com `base.html`
- Métodos HTTP corretos para cada ação
- Estilos CSS organizados e consistentes
- Mensagens de sucesso/erro para usuário
- Conexões de banco fechadas adequadamente