

Sistema de Gerenciamento de VLANs para Switches

Este projeto tem como objetivo simular um sistema de gerenciamento de VLANs, com controle de switches, portas, VLANs e usuários, utilizando interface gráfica em Java (Swing) e persistência de dados em banco SQLite.

Tecnologias utilizadas

- Java (JDK 17+)
- Swing (GUI)
- SQLite (banco de dados)
- JDBC (conexão com banco)
- iTextPDF (geração de relatórios)

Níveis de Usuário no Sistema

O sistema implementa **controle de acesso baseado em nível de usuário**, conforme os dois perfis definidos:

Nível: **TI**

- Acesso total ao sistema.
- Pode:
 - Cadastrar, editar e excluir **usuários**.
 - Cadastrar, editar e excluir **switches**.
 - Cadastrar, editar e excluir **VLANs**.
 - Atribuir/remover VLANs das portas.
 - Gerar relatórios PDF por switch.

Nível: MANUTENÇÃO

- Acesso restrito.
- Pode:
 - Tudo que o TI faz.
- Não pode:
 - Cadastrar usuários.

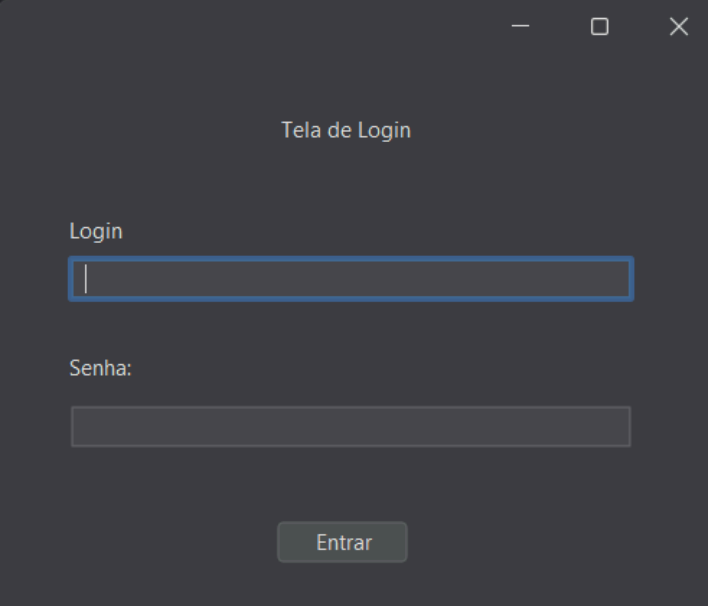
Como funciona o controle?

- O login define o nível do usuário (TI ou MANUTENCAO).
- Ao carregar cada tela, o sistema verifica o nível e:
 - **Desabilita a seleção de** Cadastro de Usuario se o nível for MANUTENCAO.
 - Permite acesso total apenas ao nível TI .

Estrutura do Sistema

🔑 Tela de Login

Permite que o usuário acesse o sistema com seu login e senha. O nível de acesso é usado para controlar o que o usuário pode visualizar ou editar.



The image shows a dark-themed login window titled "Tela de Login". It contains two input fields: "Login" and "Senha:". The "Login" field is highlighted with a blue border. Below the fields is an "Entrar" button.

👤 Tela de Cadastro de Usuários

Permite cadastrar novos usuários no sistema com nome, login, senha e nível (TI ou MANUTENÇÃO).

— □ ×

Cadastro Usuário

Nome:

Login:

Senha:

Nível

TI ▼

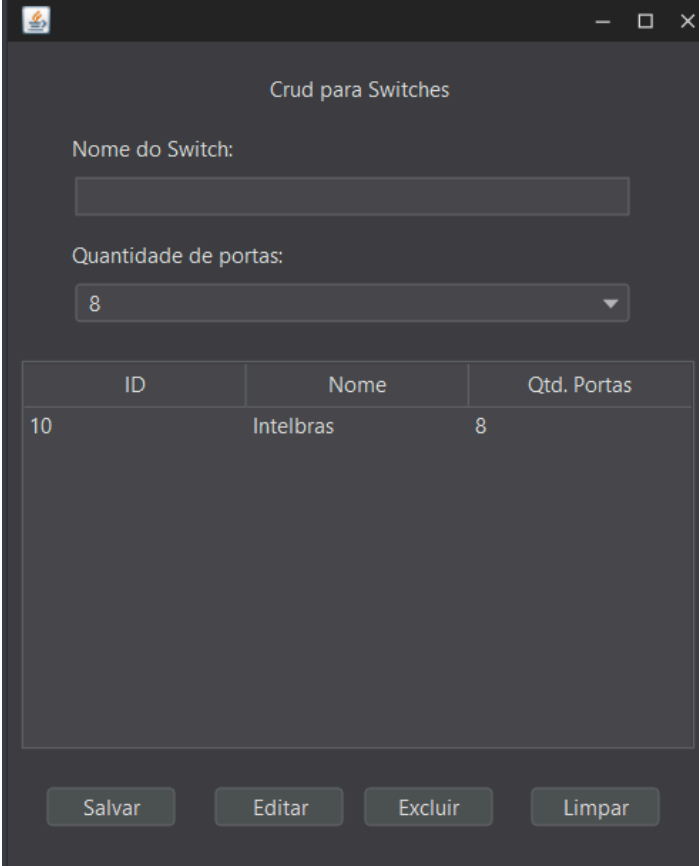
Nome	Login	Nível
Vini	vinireidelas	TI
Admin	admin	TI

Salvar

Excluir

Tela de Cadastro de Switches

Permite cadastrar um novo switch informando nome e quantidade de portas. Após o cadastro, as portas são automaticamente criadas numeradas de 1 até `qtd_portas`.



Crud para Switches

Nome do Switch:

Quantidade de portas:


8

ID	Nome	Qtd. Portas
10	Intelbras	8

Salvar Editar Excluir Limpar

Tela de Cadastro de VLANs

Permite registrar novas VLANs no sistema com um ID numérico e nome descritivo (opcional).



— □ ×

Crud de VLANs

ID VLAN (numérico):

Nome da VLAN:

ID	ID VLAN	Nome
8	1	default
5	100	PCs
9	101	Impressoras

Salvar

Editar

Excluir

Limpar

Tela de Atribuição de VLANs às Portas

Permite:

- Selecionar um switch e visualizar suas portas
- Atribuir VLANs a cada porta especificando o modo (TAGGED ou UNTAGGED)
- Definir o tipo da porta (ACCESS, TRUNK, HYBRID)
- Remover VLANs já atribuídas

Atribuição de VLANs

Selecione o Switch:

Intelbras

Selecione sua respectiva porta:

Número	Tipo	VLANs
1	HYBRID	1, 100, 101
2	TRUNK	100
3	TRUNK	100
4	TRUNK	100
5	TRUNK	100
6	TRUNK	101
7	TRUNK	101
8	TRUNK	101

Digite a VLAN desejada:

Tipo:

ACCESS

Atribuir

Atualizar Tipo

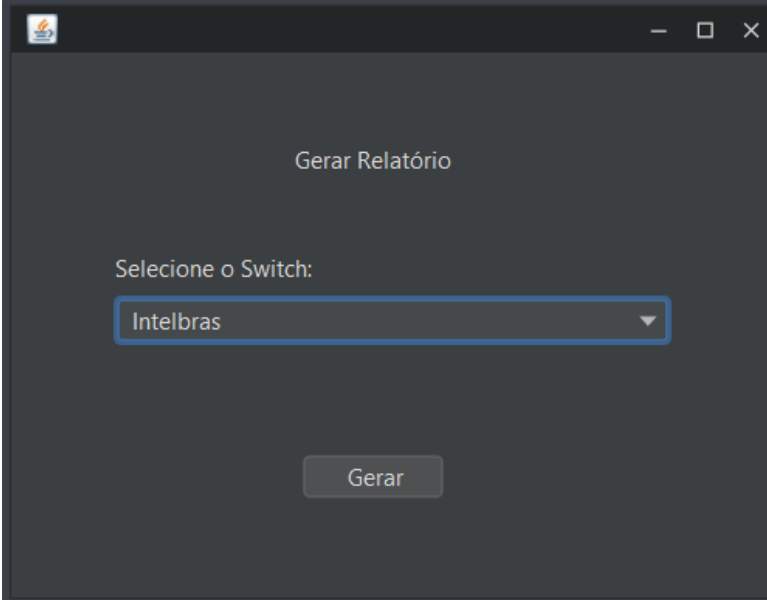
Remover

Limpar

Tela de Relatório em PDF

Permite selecionar um switch e gerar um relatório em PDF contendo:

- Todas as portas do switch
- Tipo da porta
- Lista das VLANs atribuídas (com ID, nome e modo)



Gerar Relatório

Selecione o Switch:

Intelbras

Gerar

Estrutura do Banco de Dados

```
CREATE TABLE switch (  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  nome TEXT NOT NULL,  
  qtd_portas INTEGER NOT NULL  
);
```

```
CREATE TABLE porta (  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  numero INTEGER NOT NULL,  
  tipo TEXT NOT NULL,  
  switch_id INTEGER NOT NULL,  
  FOREIGN KEY (switch_id) REFERENCES switch(id)  
);
```

```
CREATE TABLE vlan (  
  id INTEGER PRIMARY KEY AUTOINCREMENT,  
  id_vlan INTEGER NOT NULL,
```



```

    nome TEXT
);

CREATE TABLE porta_vlan (
    porta_id INTEGER NOT NULL,
    vlan_id INTEGER NOT NULL,
    modo TEXT CHECK (modo IN ('TAGGED', 'UNTAGGED')) NOT NULL DEFAULT
'TAGGED',
    PRIMARY KEY (porta_id, vlan_id),
    FOREIGN KEY (porta_id) REFERENCES porta(id),
    FOREIGN KEY (vlan_id) REFERENCES vlan(id)
);

CREATE TABLE usuario (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    nome TEXT NOT NULL,
    login TEXT NOT NULL UNIQUE,
    senha TEXT NOT NULL,
    nivel TEXT CHECK(nivel IN ('TI', 'MANUTENCAO')) NOT NULL
);

```

Organização das Classes

- **entity.***: classes de modelo (Switch, Porta, Vlan, Usuario etc)
- **dao.***: classes responsáveis pela persistência
- **view.***: todas as interfaces gráficas (CRUDs, atribuições, relatórios)
- **report.***: geração de PDF com iText
- **ConnectionFactory.java**: cria conexões com o banco de dados

Como executar o projeto

1. Compile todas as classes Java
2. Certifique-se de que o banco SQLite (**database.db**) está no diretório correto
3. Execute a classe principal (**Login.java**)
4. Use o login de teste:

- Login: `admin`
- Senha: `1234`
- Nível: `TI`



Observações

- A tela de login define o nível de acesso.
- Ao cadastrar um novo switch, as portas são automaticamente criadas.
- O relatório em PDF é gerado com base no switch selecionado, evitando poluição de dados.



Diagrama de classes



1. Camada Entity (Modelo de dados)

Switch

- `int id`
- `String nome`
- `int qtdPortas`



Relacionamento: 1 Switch → * N Portas

Porta

- `int id`
- `int numero`
- `String tipo` → "ACCESS", "TRUNK", "HYBRID"

- `int switchId`

Relacionamento:

- Cada `Porta` pertence a um `Switch`
- 1 `Porta` → * N VLANs (via `PortaVlan`)

Vlan

- `int id`
- `int idVlan`
- `String nome`

Relacionamento:

- 1 VLAN → * N Portas (via `PortaVlan`)

`PortaVlan` (associação N:N entre `Porta` e `Vlan`)

- `int portaId`
- `int vlanId`
- `String modo` → "TAGGED" ou "UNTAGGED"

Relacionamento:


- Liga `Porta` ↔ `Vlan` com um modo

Usuario

- `int id`
- `String nome`
- `String login`
- `String senha`
- `String nivel` → "TI" ou "MANUTENCAO"

2. Camada DAO (acesso a dados)

SwitchDAO

- `inserir(Switch s)`
- `atualizar(Switch s)`
- `excluir(int id)`
- `listarTodos()`
- `buscarPorId(int id)`
-  Gera `Portas` automaticamente ao inserir

PortaDAO

- `listarPorSwitch(int switchId)`
- `atualizarTipo(int portaId, String novoTipo)`
- `excluirPorSwitch(int switchId)`

VlanDAO

- `inserir(Vlan v)`
- `atualizar(Vlan v)`
- `excluir(int id)`
- `listarTodos()`
- `buscarPorId(int id)`
- `existeVlan(int idVlan)`

PortaVlanDAO

- `atribuirVlan(int portaId, int vlanId, String modo)`
- `removerVlan(int portaId, int vlanId)`
- `listarVlansPorPorta(int portaId)`
- `listarVlansComModoPorPorta(int portaId)`
- `buscarRelatorioVlanPorSwitch(int switchId)`

UsuarioDAO

- `inserir(Usuario u)`
- `atualizar(Usuario u)`
- `excluir(int id)`
- `listarTodos()`

- `buscarPorLoginSenha(String login, String senha)`

3. Camada Controller

Cada controller chama os respectivos métodos DAO e trata lógica intermediária.

- `SwitchController` → `SwitchDAO`
- `PortaController` → `PortaDAO`
- `VlanController` → `VlanDAO`
- `PortaVlanController` → `PortaVlanDAO`
- `UsuarioController` → `UsuarioDAO`

4. Camada View (Swing GUI)

`Crud_Switch.java`

- Cadastro de Switches
- Gera as portas automaticamente

`Crud_VLANs.java`

- Cadastro de VLANs com ID numérico + nome

`Atribuicao_VLANs.java`

- Seleciona Switch
- Mostra portas
- Atribui/remover VLAN

- Muda tipo (ACCESS, TRUNK, HYBRID)
- Híbrido pede 2 VLANs (untagged e tagged)

Cadastro_User.java

- Cadastro de usuários com nome, login, senha, nível

Gerar_Relatorio.java

- Gera PDF das portas e VLANs de um switch
- Mostra tipo de porta e nome das VLANs atribuídas