



**Braz Cubas**  
Centro Universitário

## Projeto: NETSIM

### Otimização de Redes de Distribuição com Árvore Geradoras Mínimas

### Simulador Dijkstra com Cisco Packet Tracer



#### **Analistas de Sistemas Responsáveis:**

Thiago Ventura Silva

Paulo Roberto Alves Genuíno Junior

Eduardo Ferreira Bernardino


#### **Coordenadora do Projeto:**

Profª. Andrea Ono Sakai

**2025.1**

**Análise e Desenvolvimento de  
Sistemas**

# Introdução

 **Redes de Distribuição com Árvores Geradoras Mínima.**

Projeto acadêmico que simula, o comportamento dinâmico de redes, evidenciando como o algoritmo de **Dijkstra calcula rotas mínimas** em tempo real.

## Objetivo do Projeto

Modelar redes de distribuição com foco na eficiência e resiliência, usando o **OSPF** como mecanismo de formação de caminhos ideais, **comparável à construção de uma árvore geradora mínima no grafo da rede**.

**Algoritmo:** O roteador cria a tabela de topologia usando resultados de cálculos baseados no algoritmo **Dijkstra Shortest Path First (SPF)**. O algoritmo **SPF** é baseado no custo acumulado para acessar um destino.

REDES.TEC. *Como funciona o algoritmo **SPF** no **OSPF**?* Disponível em: <https://redes.tec.br/lab/viewtopic.php?t=564>. Acesso em: 26 abr. 2025.

## Como funciona?

O algoritmo **Shortest Path First (SPF) de Dijkstra**, também conhecido como algoritmo de **Dijkstra**, é um **algoritmo ganancioso** usado para encontrar os caminhos mais curtos entre os nós em um gráfico.

Ele determina os caminhos mais curtos de um nó inicial designado (a origem) para todos os outros nós no grafo. Este algoritmo é amplamente utilizado em protocolos de roteamento como **OSPF (Open Shortest Path First)**.

DIJKSTRASPF. *Shortest Path First Algorithm*. GitHub Repository.

Disponível em: <https://github.com/ahojukka5/DijkstraSPF>. Tradução livre. Acesso em: 26 abr. 2025.

## A Árvore do Caminho

O algoritmo **SPF** cria uma árvore de caminho mais curto para todos os hosts em uma **área** ou no **backbone da rede**, com o roteador que está realizando o **cálculo na raiz dessa árvore**.

Para que o algoritmo **SPF** funcione corretamente, todos os roteadores na área devem ter as mesmas informações de banco de dados.

HOWTONETWORK. *Understanding the Shortest Path First Algorithm (SPF)*. Disponível em:

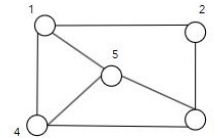
<https://www.howtonetwork.com/technical/protocols/understanding-the-shortest-path-first-algorithm-spf/>. Tradução livre. Acesso em: 26 abr.

2025.

## Área 0

### Grafo conexo

- Trata-se de um grafo na qual existe um caminho entre qualquer par de vértices.



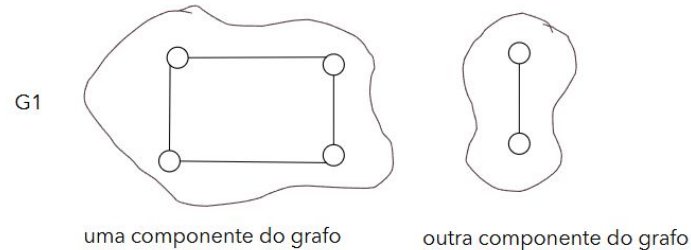
Para ir de 4 a 2 pelo caminho  $4 \rightarrow 5 \rightarrow 1 \rightarrow 2$

Para ir de 3 a 1 pelo caminho  $3 \rightarrow 5 \rightarrow 1$

## Área 0 e Outras Áreas

### Grafo desconexo

- Trata-se de um grafo que possui mais de dois componentes conexos que não estão conectados, ou seja, se houver pelo menos um par de vértices entre os quais não há um caminho entre eles.





## Protocolo de Estado de Link

O algoritmo **SPF de Dijkstra** é uma parte fundamental de protocolos de roteamento link-state, como o OSPF.

O algoritmo determinará os melhores caminhos para o roteamento do tráfego de rede.

AHOJUKKA5. *DijkstraSPF*. GitHub Repository. Disponível em: <https://github.com/ahojukka5/DijkstraSPF>. Tradução livre. Acesso em: 26 abr. 2025..

## Conceitos Aplicados

- ✓ **Redes como grafos: roteadores = vértices, links = arestas.**
- ✓ **OSPF utiliza o algoritmo de Dijkstra para determinar o menor caminho (custo).**
- ✓ **A estrutura resultante é otimizada como uma árvore geradora com menor custo total.**

# Algoritmo Utilizado

## Dijkstra e sua relação com Árvores Geradoras

- Determina os caminhos de menor custo a partir de um nó origem.
- OSPF constrói, a partir disso, a árvore de rotas com menor custo por área.
- Comparável a Prim/Kruskal\* em estrutura de resultado, mas com foco em caminhos mínimos por origem.

**\*Prim e Kruskal** são algoritmos usados para encontrar a **Árvore Geradora Mínima (MST)** em um grafo — Este assunto está fora do escopo desta demonstração.

# Ferramentas e Tecnologias

## Ambiente Técnico

- **Dispositivo:** Cisco Systems.
- **Linguagem:** Cisco IOS.
- **Software de Simulação:** Cisco Packet Tracer (versão recomendada: 8.2 ou superior).
- **Biblioteca ou Framework:** OSPF (Open Shortest Path First): **Protocolo de roteamento dinâmico de Dijkstra (SPF)**, responsável por calcular os melhores caminhos entre os nós da rede. Configurado por meio da CLI e utilizado como núcleo da simulação de convergência e falhas do caminho em rota.

# Estrutura do Projeto

## Organização da Simulação

- Topologia com múltiplos roteadores interligados.
- Configuração OSPF nas interfaces.
- Simulação de quedas e análise de novas rotas.
- Capturas e documentação visual dos testes.

netsim\_ospf\_project/

— topologia/

  |— redes\_visuais/

    |— topologia\_inicial.png

    |— topologia\_com\_falha.png

  |— packet\_tracer/

    |— netsim\_final.pkt

— configuracoes/

  |— roteador\_A.txt

  |— roteador\_B.txt

  |— roteador\_C.txt

— resultados/

  |— tabelas\_roteamento\_iniciais.txt

  |— tabelas\_roteamento\_pos\_falha.txt

  |— comparativo\_convergencia.txt

— apresentacao/

  |— slides\_apresentacao.pdf

  |— graficos\_convergencia.png

— README.md

# Funcionalidades

## Principais Recursos da Simulação

- Construção da topologia como grafo conectado.
- Aplicação de OSPF (Dijkstra-SPF).
- Simulação de falhas em links ou nós.
- Modo passo a passo para análise de convergência.
- Exportação de resultados técnicos

# Resultados Esperados



## Impactos e Aprendizados

- Compreensão visual da reorganização automática das rotas.
- Relacionamento entre redes reais e teoria de grafos.
- Aplicação prática de conceitos como caminhos mínimos e otimização.
- Produção de relatórios com dados reais da simulação.



# Conclusão

## ✓ Fechamento do Projeto

- A simulação prova que protocolos como o OSPF, ao aplicar Dijkstra, geram rotas otimizadas e adaptativas, permitindo uma estrutura de rede comparável a árvores geradoras mínimas em eficiência

# Integrantes



## Equipe do Projeto

- Eduardo Ferreira Bernardino – Gerente de Projeto
- Paulo Roberto Alves Genuíno – Backend (Configuração)
- Thiago Ventura Silva – Frontend (Visualização)



**Braz Cubas**  
Centro Universitário