

# PROTOCOLO DE ROTEAMENTO DE REDES



ENTENDENDO SEU FUNCIONAMENTO  
<https://github.com/joaovictorascia/AED>



# INTEGRANTES:



**Pedro Souto de Oliveira  
Feitosa Almeida**

232013265



**João Victor Machado  
Santos**

232013318



**Pedro Henrique Oliveira**

232029962

# PROGRAMA ★

Descrição



Menu



Funcionalidades



Conteúdos abordados





# INTRODUÇÃO



Descrição do projeto

Este projeto computacional foca no estudo e implementação de protocolos de roteamento de redes, visando otimizar o tráfego de dados e garantir robustez na comunicação. O objetivo é oferecer insights valiosos e soluções práticas para desafios comuns em redes modernas, contribuindo para avanços na área de redes de computadores.



```
...es;
nPartida;
op;
int bitsToSend;
HashTable tabelaDeRoteamento;

cout << "Insira a quantidade de nos: " << endl;
cin >> nNodes;

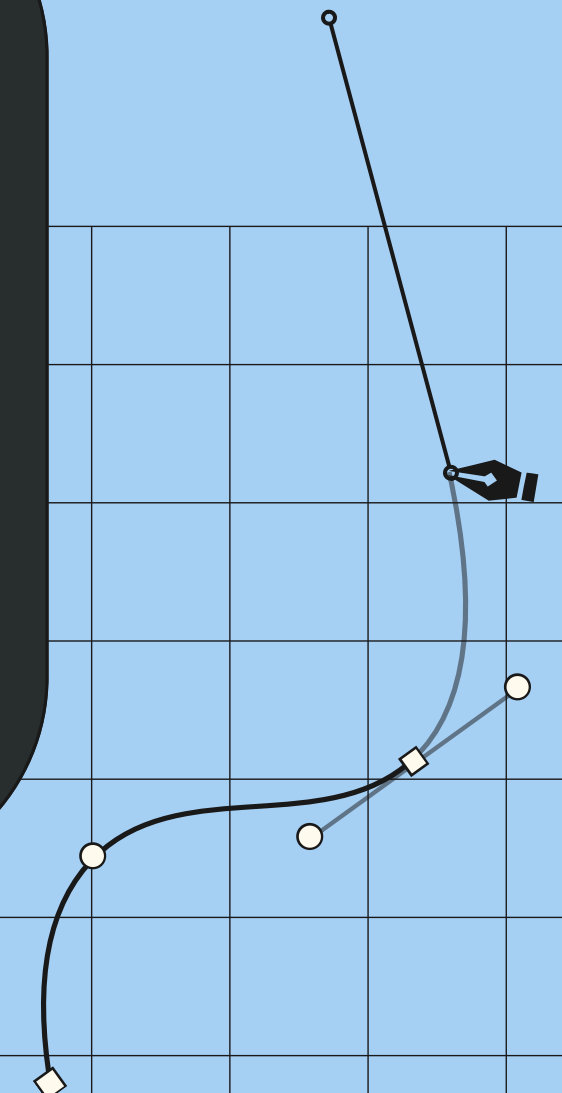
// Alocação dinâmica para poder passar array de duas dimensões para a função
int **graph = new int *[nNodes];
for (int i = 0; i < nNodes; i++)
{
    graph[i] = new int[nNodes];
}

for (int i = 0; i < nNodes; i++)
{
    char line = 65 + i;
    tabelaDeRoteamento.insert(Package(line, 0));
    for (int j = 0; j < nNodes; j++)
    {
        char column = 65 + j;
        cout << "A distancia entre " << line << " -> " << column << ": ";
        cin >> graph[i][j];
    }
}

while (true)
{
    cout << "1 - Enviar pacote" << endl;
    cout << "2 - Adicionar no" << endl;
    cout << "3 - Contabilizar distancia entre nos" << endl;
    cout << "4 - Finalizar" << endl;
    cout << "Opcao: ";
    cin >> op;

    switch (op)
    {
        case 1:
            cout << "Deseja enviar um pacote para qual no?: " << endl;
            cin >> nPartida;
            cout << "Quantos bits deseja enviar? " << endl;
            cin >> bitsToSend;
            if (nPartida < 'A' || nPartida >= 'A' + nNodes)
            {
                cout << "Opcao invalida" << endl;
            }
        }
    }
}
```

AMOSTRA DE CÓDIGO



# JUSTIFICATIVA

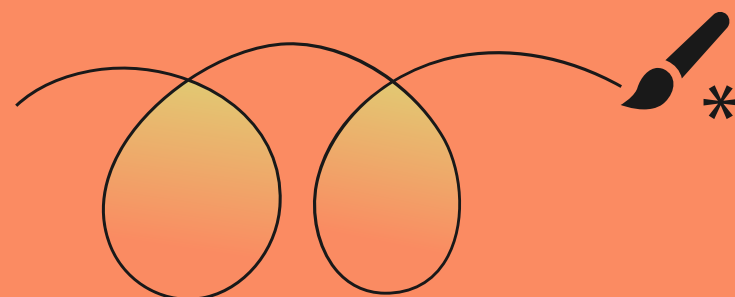


Descrição do Projeto

Os protocolos de roteamento são cruciais para a comunicação eficiente em redes de computadores, garantindo a transmissão rápida e segura de dados.



Desenvolver um protocolo de roteamento personalizado permite aos alunos aplicar conceitos teóricos na prática, entendendo melhor os algoritmos e a importância das tabelas de roteamento dinâmicas.



UPLOAD

# METAS



Descrição do projeto



Com a conclusão desse projeto, esperamos um protocolo de roteamento que:

Gere caminhos ótimos para a transmissão de dados entre aparelhos

Facilite o envio e a checagem de dados através de um menu interativo.

Permita a adição dinâmica de novos dispositivos e conexões de rede.

Garanta um ID único de todos os dados transmitidos

MENU

1	Enviar pacote
2	Adicionar no
3	Contabilizar distancia entre nos
4	Finalizar

# SOBRE O MENU:



Metas



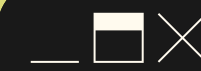
É capaz de representar a rede como um grafo.



Implementa o algoritmo de Dijkstra



Cria uma tabela de roteamento inteligente.

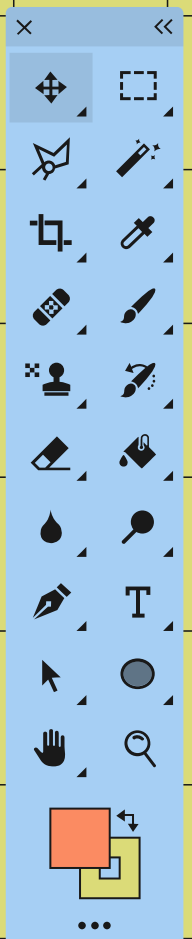


Menu interativo.



Possui um sistema de identificação de dados em uma tabela hash.

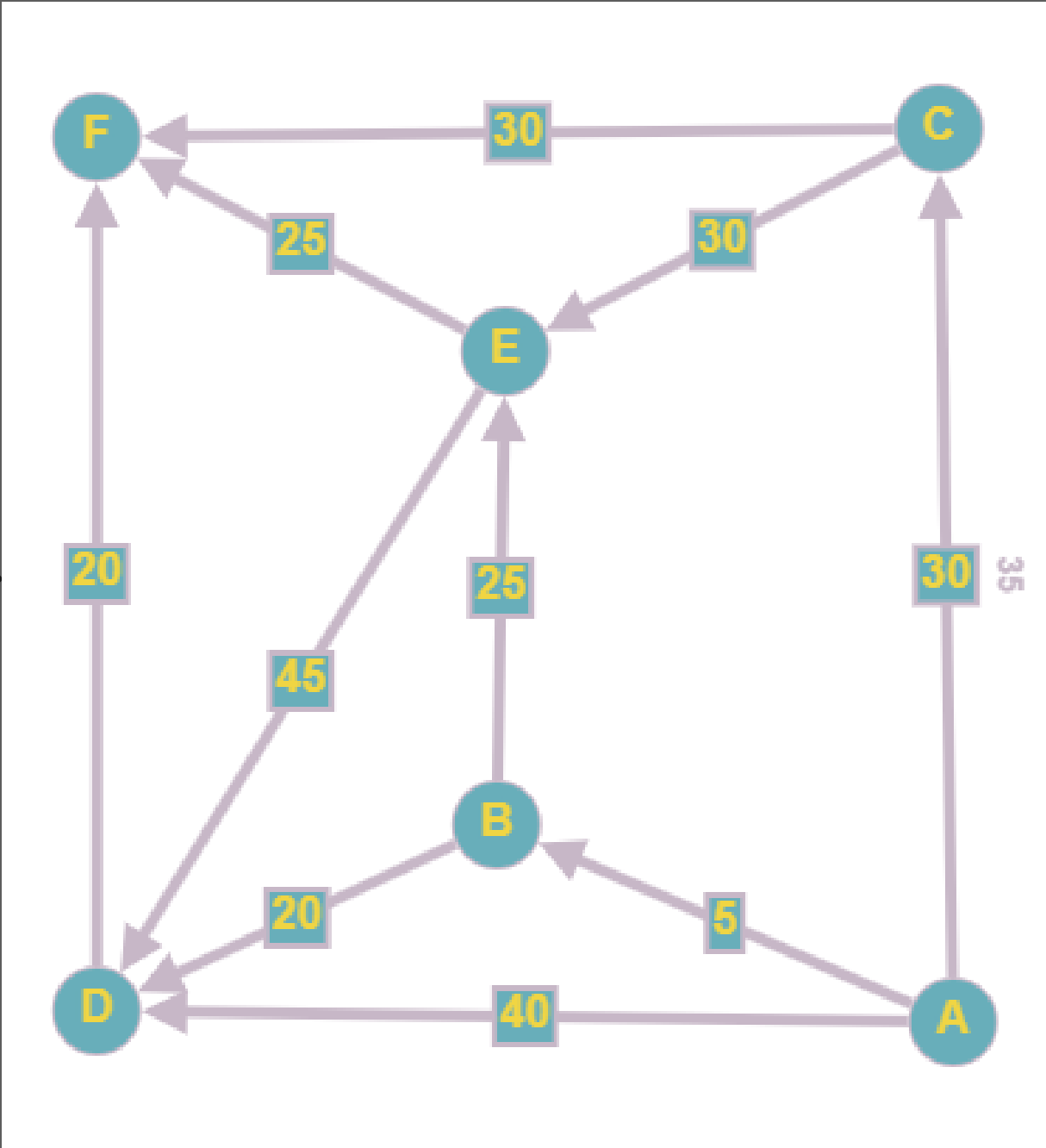
MENU	
1	Enviar pacote
2	Adicionar no
3	Contabilizar distancia entre nos
4	Finalizar



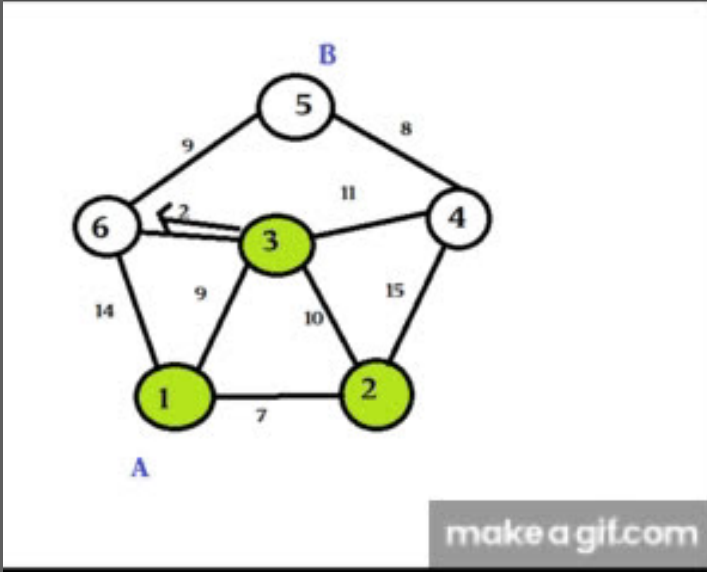
# REPRESENTAÇÃO DO GRAFO:

Predecessores	
A	B
A	C
B	D
B	E
D	E

```
int graph[6][6] = {
    {0, 5, 40, 0, 0, 0},
    {0, 0, 0, 20, 25, 0},
    {0, 0, 0, 0, 0, 30},
    {0, 0, 0, 0, 0, 20},
    {0, 0, 0, 25, 0, 25},
    {0, 0, 0, 0, 0, 0}
};
```



A letra A corresponde ao no 0.  
Nodo                    Caminho  
A -> B                3 A -> B  
A -> C                A -> C  
A -> D                A -> B -> D  
A -> E                A -> B -> E  
A -> F                A -> B -> D -> F  
Foi escolhido o melhor caminho.





# EXPLICAÇÃO DO HASH & PACKAGE:

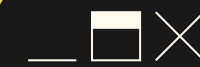


Sobre o menu:

Uma tabela hash é uma estrutura de dados que permite armazenar pares de chave de forma eficiente. Ela utiliza uma função hash para calcular um índice a partir da chave, o que permite acesso rápido aos valores armazenados.

## Aplicação no Código:

No nosso projeto, a tabela hash "HashTable" é usada para armazenar e gerenciar pacotes de dados "Package". Cada package é identificado por um char "identificacao" e contém informações sobre os dados transmitidos, além de listas que registram de quais nós o pacote já recebeu dados "receivedFrom" e para quais nós ele já enviou dados "deliveredTo".



Operações básicas:  
Inserção - "insert"  
Remoção - "erase"  
Busca - "find",  
"table.end()"

# EXPLICAÇÃO DO HASH & PACKAGE:

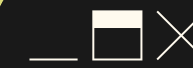


Sobre o menu:

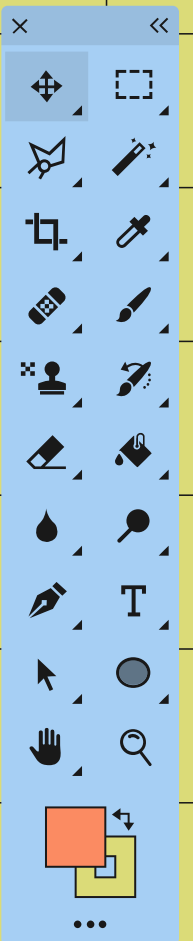
```
1 class Package
2 {
3 public:
4     char identificacao;
5     int data;
6     std::vector<char> receivedFrom; // Para armazenar de quais nodos ele já recebeu
7     std::vector<char> deliveredTo; // Para armazenar quais nodos ela já enviou dados
8
9     ...
10 };
```



A classe “Package” é fundamental para o funcionamento da tabela hash e contém um identificador, um campo de dados, e duas listas para rastrear a origem e o destino dos dados recebidos e enviados



Em suma a tabela hash é utilizada neste código para armazenar e gerenciar pacotes de dados associados a nós em um grafo. Ela permite acesso eficiente para inserção, atualização e visualização dos pacotes, facilitando o rastreamento das transmissões de dados entre os nós.



Conteudos abordados:



Programa

C L A S S E S	MODELAGEM DA REDE	ALGORITMO DE ROTEAMENTO	
	A rede será modelada como um grafo, onde os nós representam os dispositivos (incluindo a internet e o roteador) e as arestas representam as conexões entre eles.	O algoritmo de Dijkstra será utilizado para calcular os caminhos mais curtos entre os nós da rede, assegurando que os dados percorram o caminho ótimo até o destino final.	Será desenvolvida uma tabela de roteamento que armazene os caminhos mais curtos para cada nó. A tabela será dinâmica, permitindo a adição de novos nós e arestas à rede.
	GESTÃO DE DADOS	INTERFACE DE INTERAÇÃO	
	Utilizando uma tabela Hash, será criado um sistema para gerar IDs únicos para os dados transmitidos, com base no endereço de destino. Esses IDs garantirão que os dados sejam corretamente encaminhados e armazenados.	Um menu interativo será criado para permitir o envio de dados para destinos específicos e a consulta dos dados recebidos. O menu facilitará a interação com o protocolo de roteamento desenvolvido.	
PROTOCOLO DE ROTEAMENTO DE REDES			