



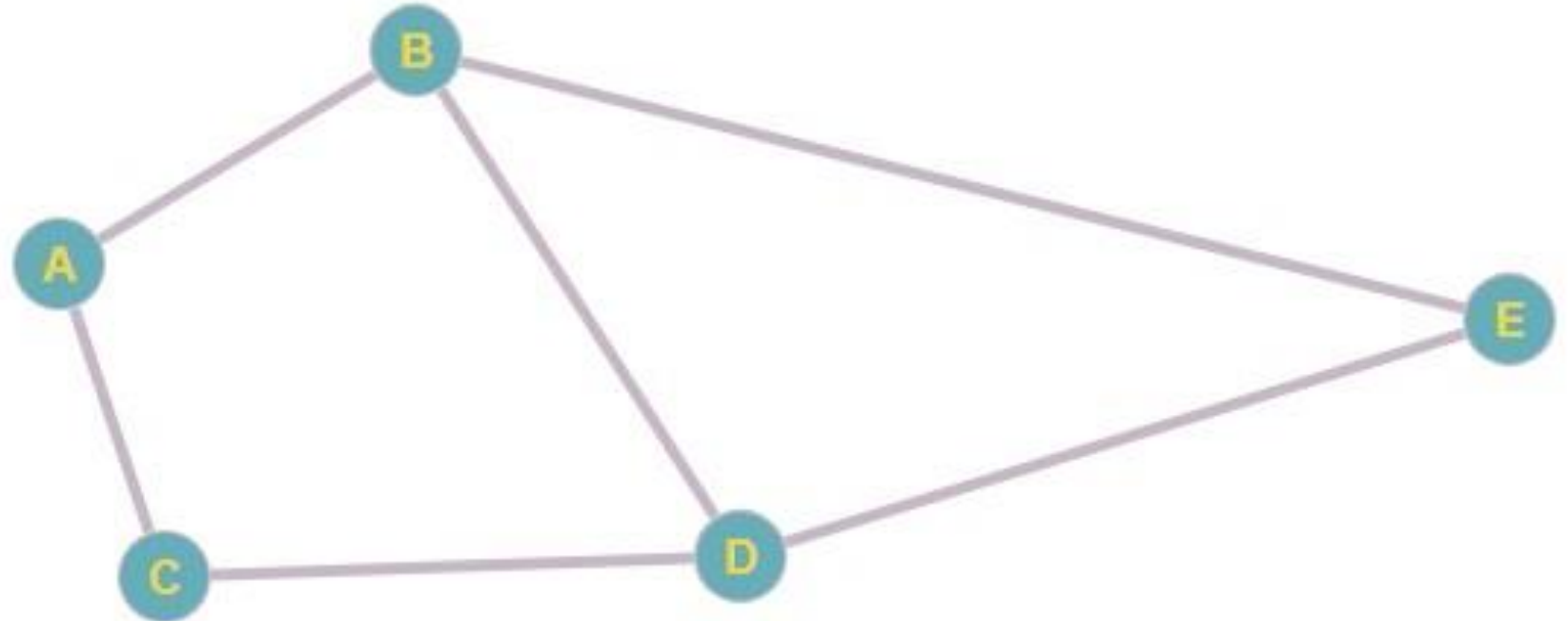
Inteligência Artificial Aplicada



Busca em largura (BFS – *Breadth-First Search*)

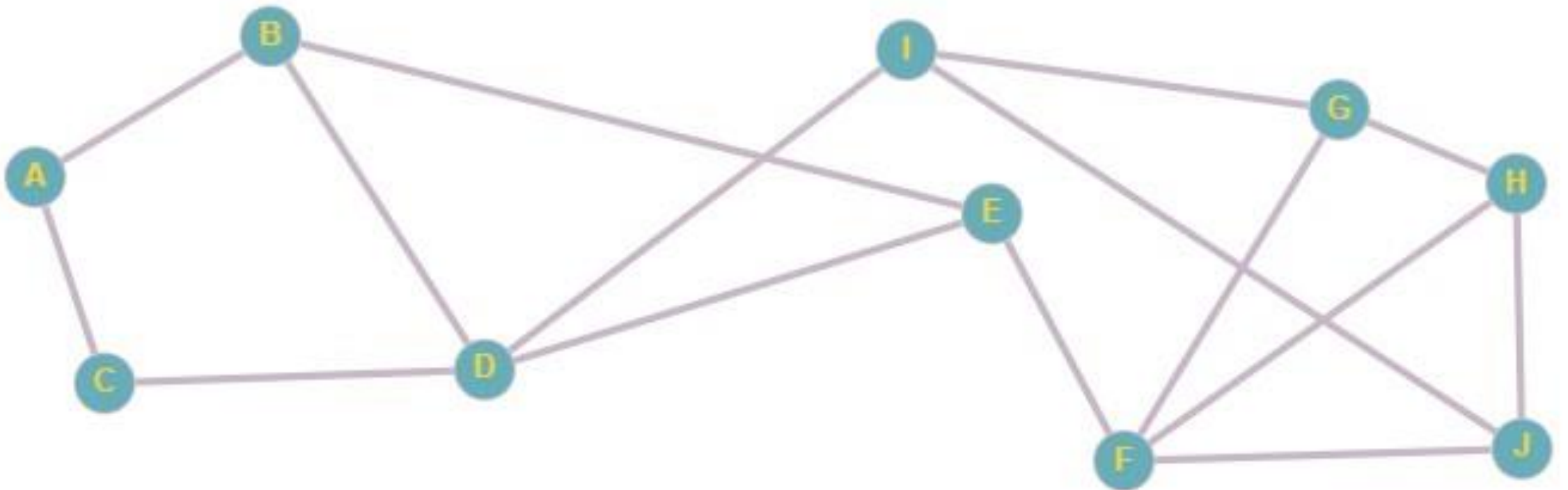
BFS – 1

Dado o grafo não ponderado a seguir, encontre o caminho a partir do nó A utilizando o algoritmo BFS.



BFS – 2

Dado o grafo não ponderado a seguir, encontre o caminho a partir do nó A utilizando o algoritmo BFS.



BFS – 3

Elabore o grafo e utilize o algoritmo BFS para demonstrar o caminho para percorrer todos os bairros, nos seguintes pontos de partida: a) Centro; b) Iná; e c) Campo Largo da Roseira .

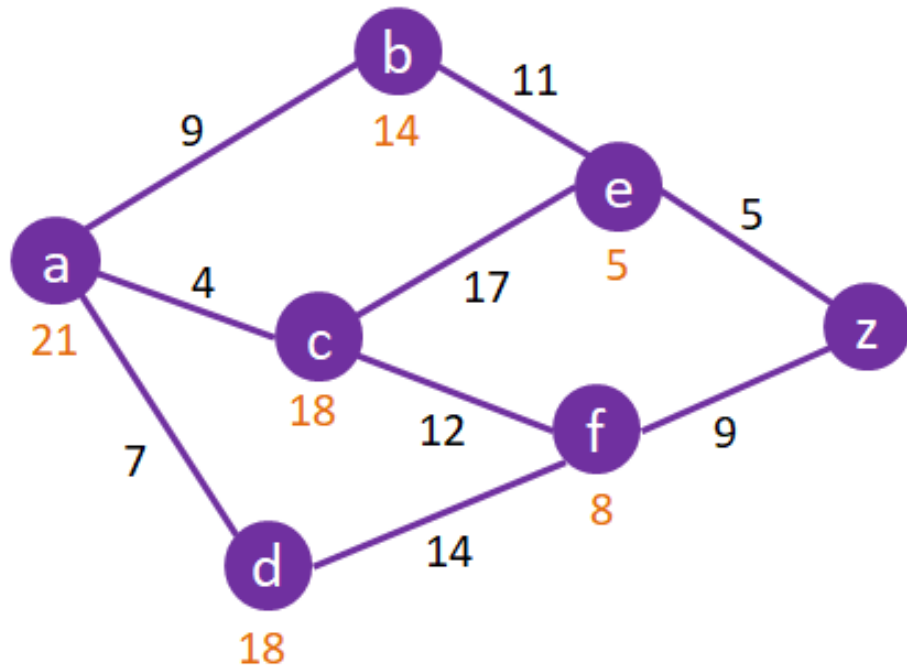
	Centro	Afonso Pena	São Domingos	Costeira	Guatupê	Braga	Iná	Boneca do Iguaçu	Campo Largo da Roseira
Centro	0	1	0	1	0	0	0	1	0
Afonso Pena	1	0	1	0	1	0	0	0	0
São Domingos	0	1	0	0	0	1	0	0	0
Costeira	1	0	0	0	1	0	1	0	0
Guatupê	0	1	0	1	0	0	0	0	1
Braga	0	0	1	0	0	0	0	0	0
Iná	0	0	0	1	0	0	0	0	0
Boneca do Iguaçu	1	0	0	0	0	0	0	0	0
Campo Largo da Roseira	0	0	0	0	1	0	0	0	0



A*



Busca A* (A – estrela)



- **Ideia principal:** Algoritmo heurístico que prioriza caminhos mais promissores.
- **Fórmula:** $f(n) = g(n) + h(n)$

A*– caminho no Google Maps

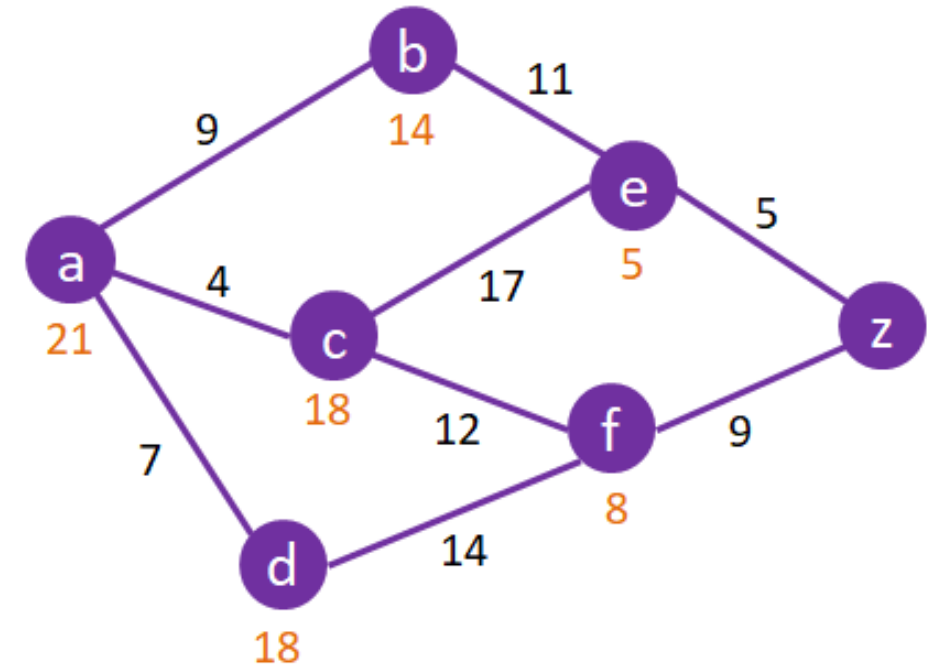
Situação: você está indo de casa para a universidade e quer o caminho mais rápido.

Passos:

1. Você pode pegar ruas menores (menos trânsito) ou avenidas (mais rápidas, mas podem estar congestionadas).
2. O Google Maps calcula custo real (distância já percorrida) e custo estimado (quanto falta).
3. O melhor caminho é aquele com menor custo total.

Fundamentos

Grafos



- **Vértices (ou nós)** → pontos no mapa (ex: cidades, células em um grid).
- **Arestas** → conexões entre os pontos (ex: estradas, caminhos possíveis).
- **Peso das arestas** → custo do movimento entre os nós (ex: distância, tempo).

Fila de Prioridade (*Heap*)

- Estrutura de dados baseada em árvore usada para gerenciar prioridades de forma eficiente.
- Utiliza-se para sempre acessar o nó com menor custo primeiro.



Funções

$g(n) \rightarrow$ Custo do caminho do início até o nó atual n .

$h(n) \rightarrow$ Estimativa heurística do custo do nó atual n até o objetivo.

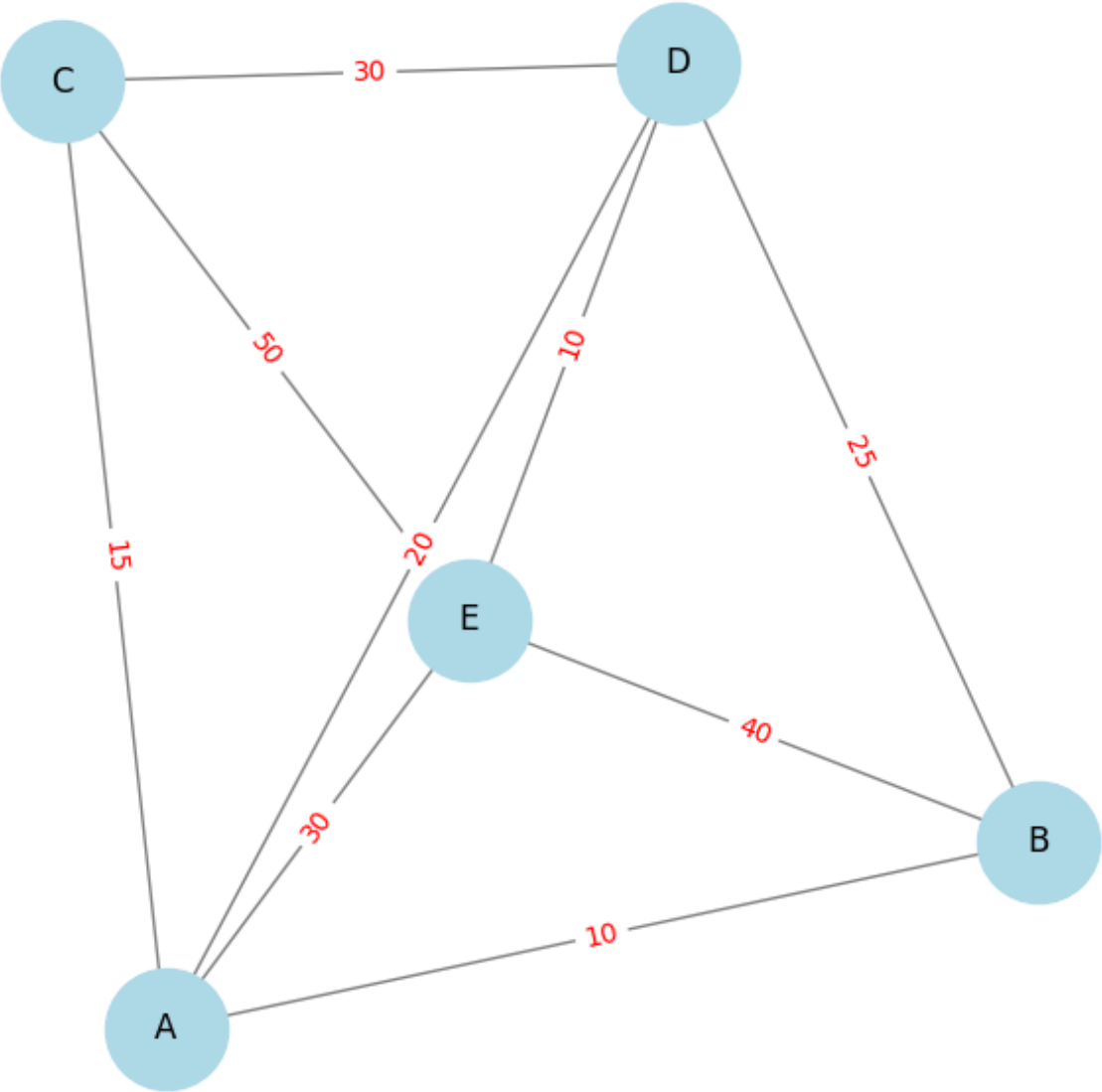
$f(n) = g(n) + h(n) \rightarrow$ Soma dos custos acima, usada para priorizar a busca.

A* - funcionamento

1. Adicionar o nó inicial à lista de nós a serem explorados;
2. Repetir os seguintes passos até encontrar o objetivo ou esgotar os caminhos possíveis:
 - * Escolher o nó com menor $f(n)$ (soma dos custos)
 - * Se for o objetivo, termina.
 - * Caso contrário, expande os vizinhos do nó, calculando $g(n)$, $h(n)$ e $f(n)$,
 - * Se um vizinho tem um custo menor do que uma versão anterior dele, atualizar os custos.
 - * Mover o nó explorado para a lista de nós já processados.
3. Retornar o caminho encontrado.

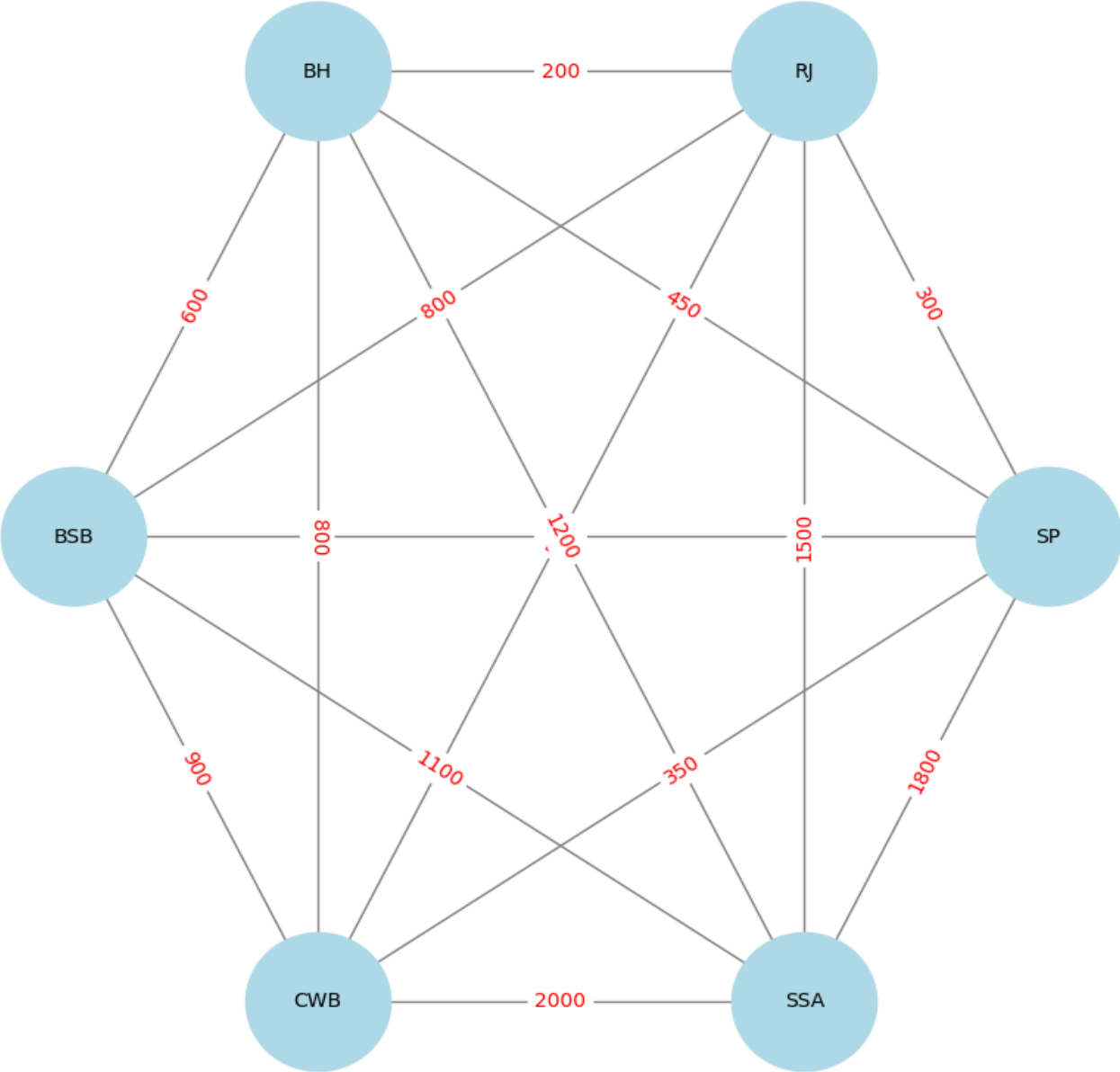
A* - implementação

A*-1 Determine o caminho ótimo a partir da cidade A até a cidade E, levando em consideração as distâncias diretas entre as cidades.



Heurísticas	
Cidade	$h(n)$ - Estimativa até E
A	20
B	25
C	30
D	5
E	0

A*-2 São Paulo (SP) até Salvador (SSA).



Heurísticas	
Cidade	h(n) - Estimativa até SSA (km)
SP	1500
RJ	1200
BH	900
BSB	1100
CWB	1700
SSA	0