

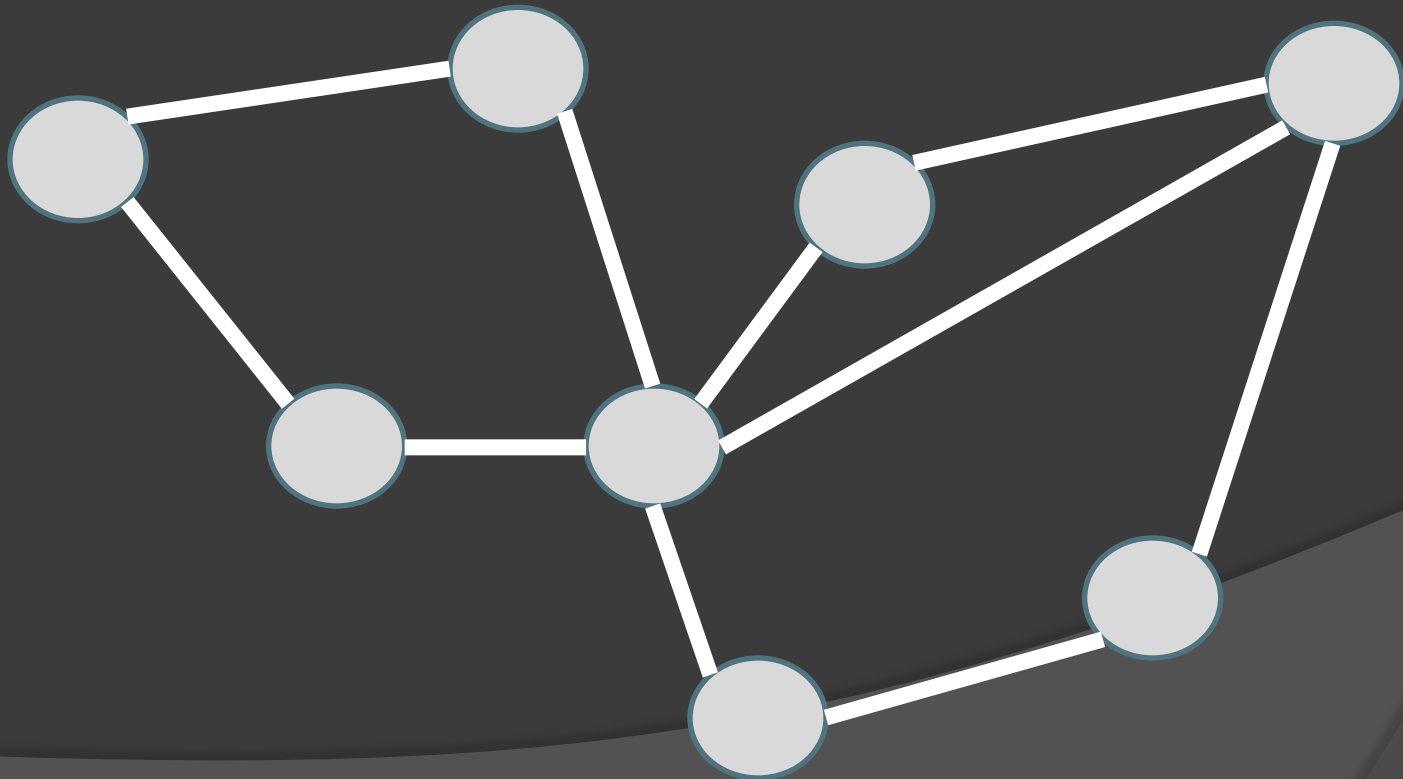
GRAFOS

Introdução

- Teoria dos grafos estuda a relação entre os objetos de um conjunto
- Todo o grafo é formado por um conjunto de Vértices e Arestas
- Vértice: elemento de um conjunto, objetos da estrutura
- Arestas: conjunto de ligações entre vértices os vértices

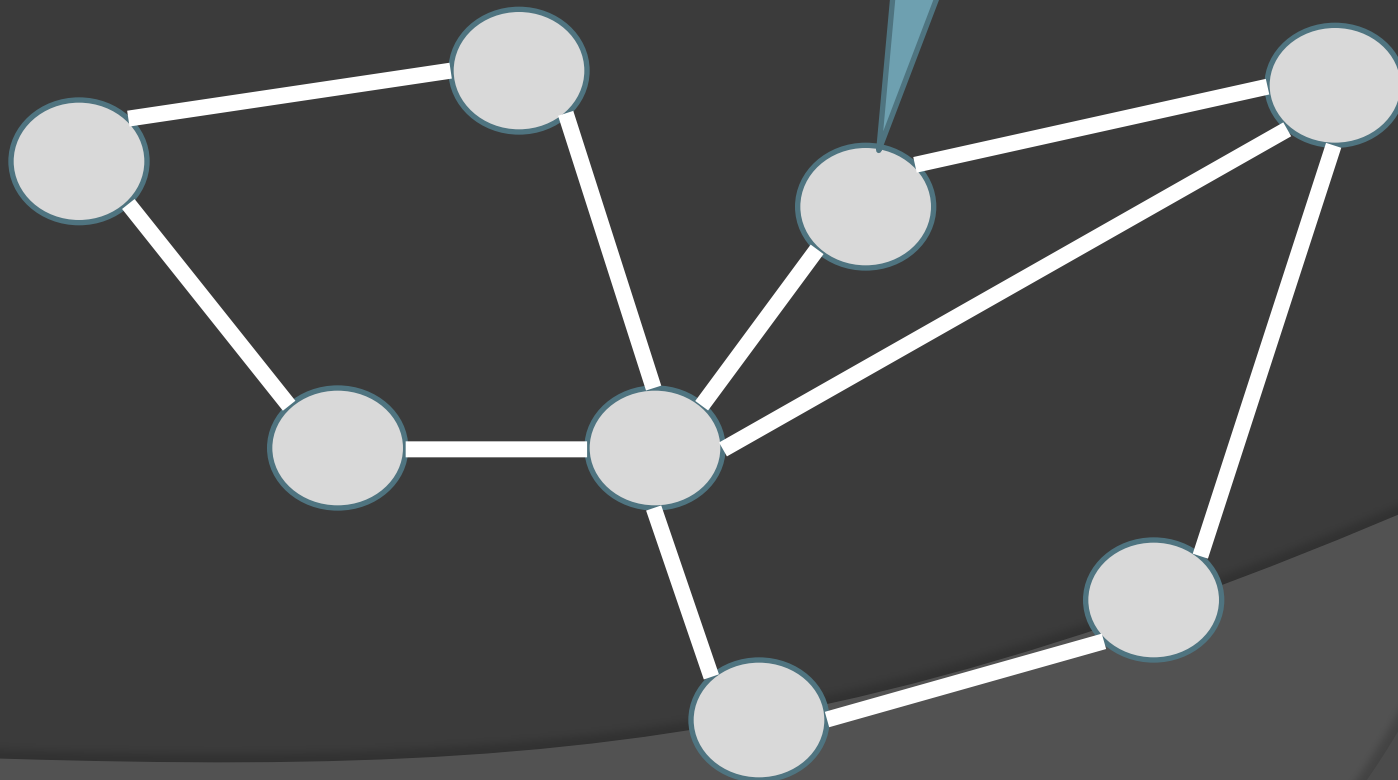
Introdução

- Vejamos este exemplo



Introdução

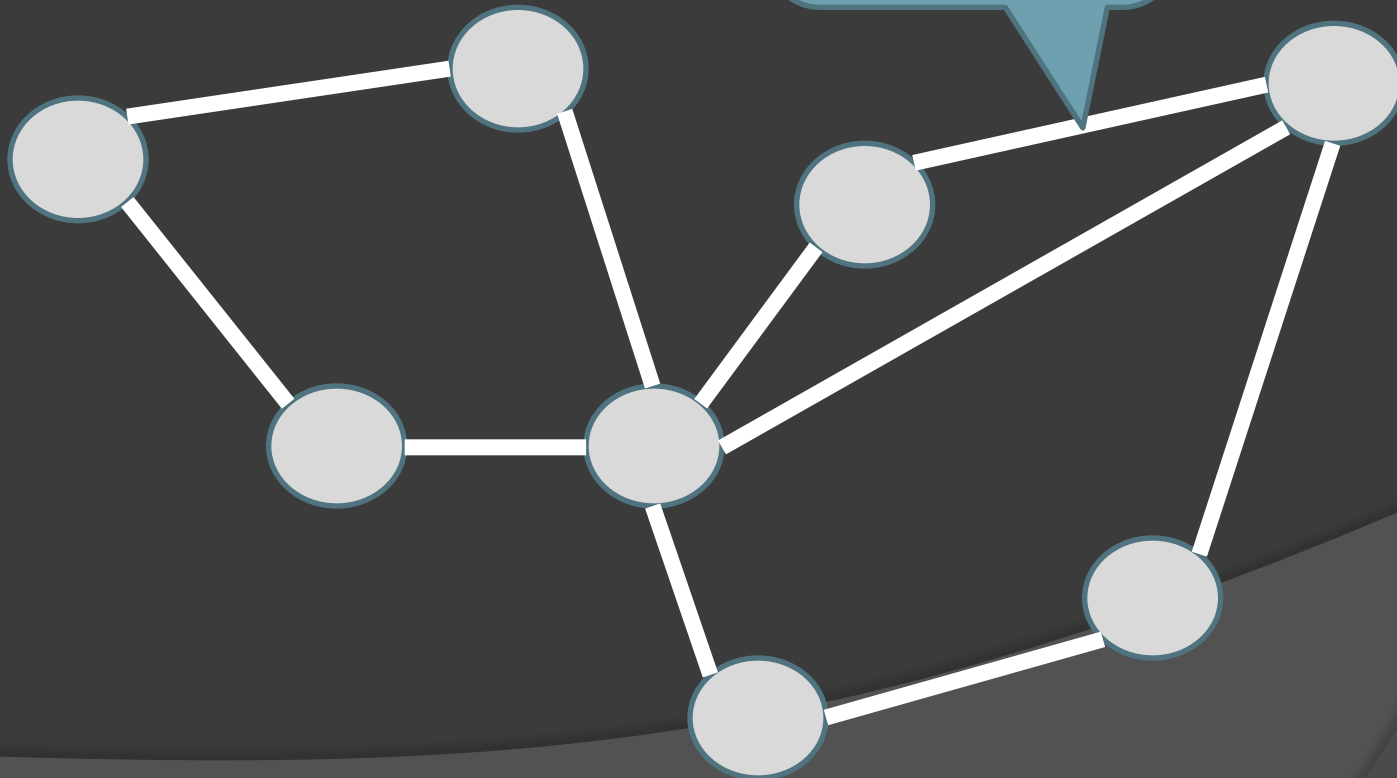
- Vejamos este exemplo



Introdução

- Vejamos este exemplo

Aresta, ligando
um par de
vértices



Grafos

- ⦿ As arestas podem ou não ter pesos, indicando preços, quilometragens, ...
- ⦿ Ou sem pesos: apenas evidenciando uma relação entre os dois vértices.

Grafos

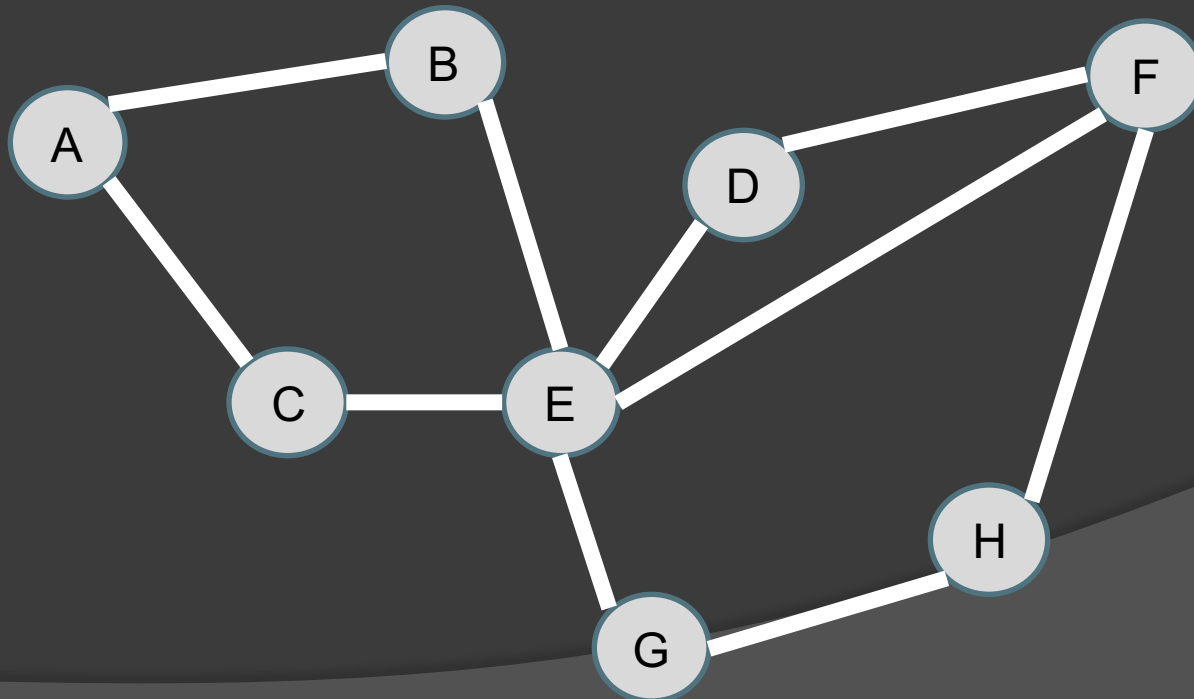
- Há duas possibilidades de implementação:
 - matriz de adjascências
 - conjuntos de vértices e arestas

Matriz de Adjascências

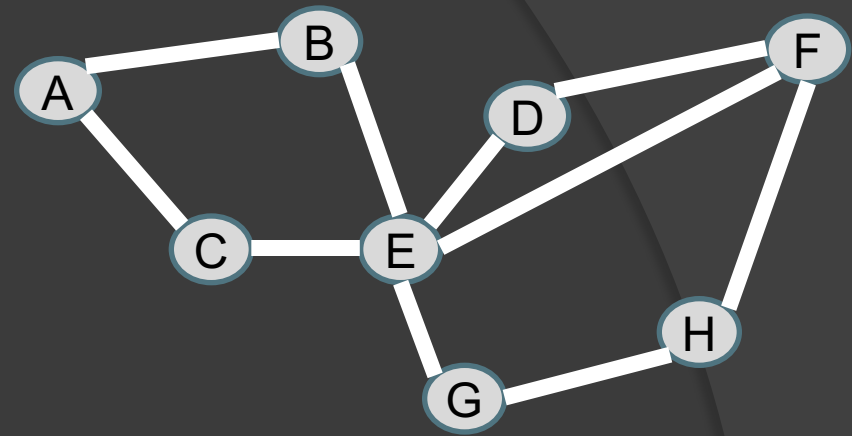
- Na matriz de adjascências temos uma matriz ou tabela, em que cada linha ou coluna representa um vértice e cada célula é a relação existente entre dois vértices.

Matriz de Adjascências

- Supondo que tenhamos que representar o seguinte grafo, como ficaria uma matriz de adjascências?



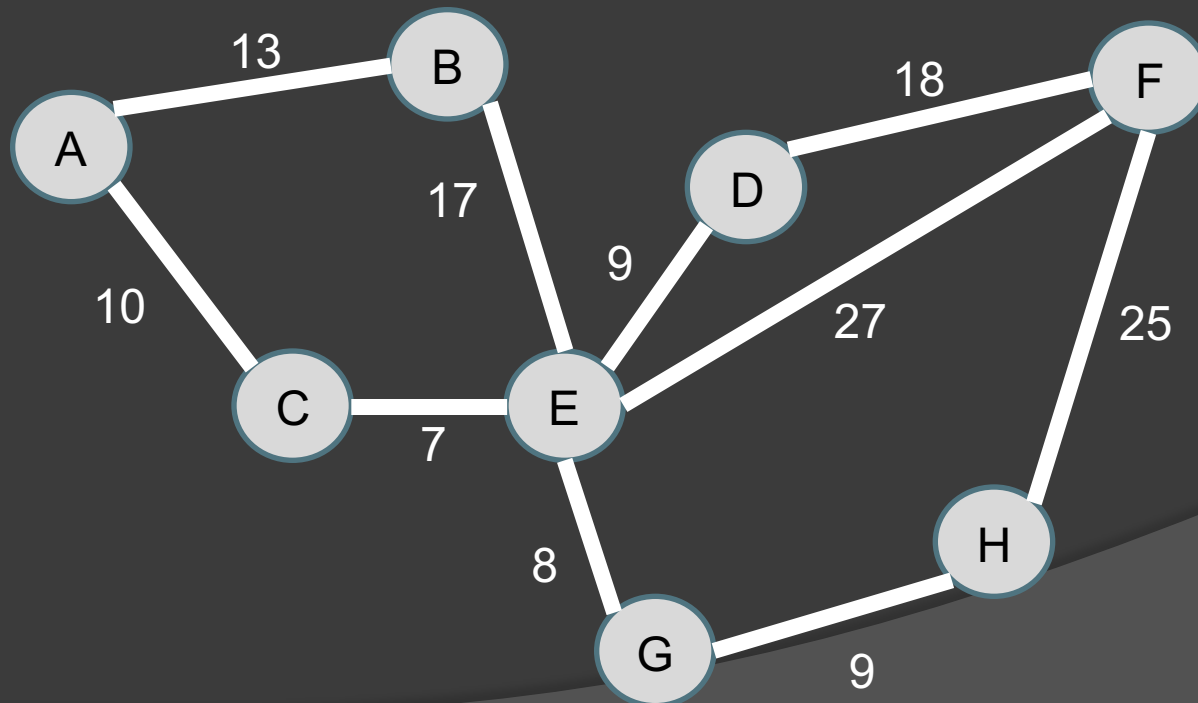
Matriz de Adjascências



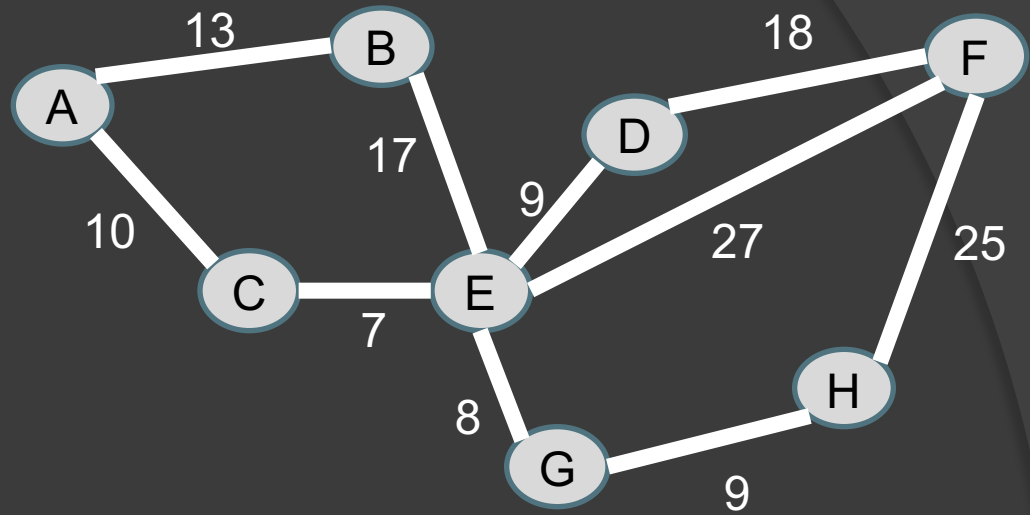
	A	B	C	D	E	F	G	H
A	0	1	1	0	0	0	0	0
B	1	0	0	0	1	0	0	0
C	1	0	0	0	1	0	0	0
D	0	0	0	0	1	1	0	0
E	0	1	1	1	0	1	1	0
F	0	0	0	1	1	0	0	1
G	0	0	0	0	1	0	0	1
H	0	0	0	0	0	1	1	0

Matriz de Adjascências

- Se trabalharmos com peso, como poderíamos fazer?



Matriz de Adjascências



	A	B	C	D	E	F	G	H
A	0	13	10	0	0	0	0	0
B	13	0	0	0	17	0	0	0
C	10	0	0	0	7	0	0	0
D	0	0	0	0	9	18	0	0
E	0	17	7	9	0	27	8	0
F	0	0	0	18	27	0	0	25
G	0	0	0	0	8	0	0	9
H	0	0	0	0	0	25	9	0

Conjunto de Vértices e Arestas

- ⦿ Quando criamos através de um conjunto de vértices e arestas devemos utilizar estruturas como arrays ou sets
- ⦿ Devemos ter:
 - uma estrutura para manter os vértices;
 - uma estrutura para manter as arestas (cada posição de uma aresta tem um array de duas posições, estabelecendo a relação entre dois vértices); e
 - de acordo com o problema, devemos ter um array para armazenar os custos de cada aresta. O índice do array custos corresponde ao índice da aresta.

Operações importantes

- adicionarVertice(vertice): dado um vértice, adicioná-lo à estrutura de vértices;
- adicionarAresta(verticeX, verticeY, peso=1): dado dois vértices e um peso opcional, insere uma adjascência;

Operações importantes

- getVertices(): retorna todos os vértices do grafo;
- getArestas(): retorna todas as arestas do grafo;
- getAdjascencias (vertice): dado um vértice, retorna um array de vértices adjacentes;
- getPeso (verticeX, verticeY): dado dois vértices, retornar o peso da aresta correspondente à adjascência.

(matriz de adjascência)

```
class Grafo{
    $matriz_adjascencias[][];
    function adicionarVertice($vertice) {
        //?
    }
    function adicionarAresta($vx, $vy, $custo=1){
        //?
    }
    function getVertices(){
        //?
    }
    function getArestas(){
        //?
    }
    function getAdjacencias($vertice){
        //?
    }
    function getPeso($vx, $vy){
        //?
    }
}
```


(conjuntos: vértices e arestas)

```
Classe Grafo{
    $vertices = array();
    $arestas = array();
    $pesos = array();
    function adicionarVertice($vertice) {
        //?
    }
    function adicionarAresta($vx, $vy, $custo=1){
        //?
    }
    function getVertices(){
        //?
    }
    function getArestas(){
        //?
    }
    function getAdjacencias($vertice){
        //?
    }
    function getPeso($vx, $vy){
        //?
    }
}
```

```

<?php
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
class Grafo {
    var $matriz = null;
    function criarMatriz ($vertices){
        $this->matriz = array();
        foreach($vertices as $v){
            $this->matriz[$v] = array();
            foreach($vertices as $w){
                $this->matriz[$v][$w] = 0;
            }
        }
    }
    function adicionaAresta($v1, $v2, $peso=1){
        $this->matriz[$v1][$v2] = $peso;
        $this->matriz[$v2][$v1] = $peso;
    }
    function getAdjs($v){
        $resultado = array();
        foreach($this->matriz[$v] as $key => $peso){
            if($peso != 0){
                $resultado[] = $key;
            }
        }
        return $resultado;
    }
    function getPeso($v1, $v2){
        return $this->matriz[$v1][$v2];
    }
    function getVertices(){
        return array_keys($this->matriz);
    }
}

$g = new Grafo();
$g->criarMatriz(array("João", "Maria", "José"));
$g->adicionaAresta("João", "Maria");
$g->adicionaAresta("João", "José");
echo "<pre>";
print_r($g->getAdjs("João"));
echo "</pre>";
?>

```