

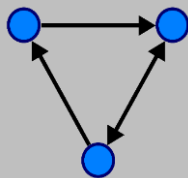
Tipo abstracto de datos: Grafo

Definición: Un grafo dirigido es una estructura de datos que contiene vértices y aristas, en este caso las aristas solo pueden tener una dirección.

Procedimientos

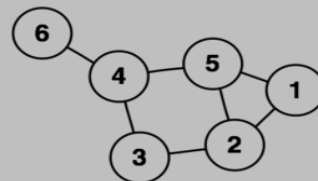
- createGraph (boolean **isDirected**).
- depthFirstSearch (Node **a**).
- breadthFirstSearch (Node **a**).
- addVertex (Node **a**).
- deleteVertex (Node **a**).
- createEdge (Node **a**, Node **b**).

Grafo Dirigido



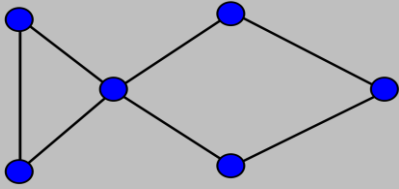
Invariante: Toda la arista tiene una o dos direcciones, las aristas siempre tienen un fin y un inicio.

Grafo No dirigido



Invariante: Las aristas no tienen dirección, se puede llegar al nodo o devolverse de este, mientras exista la arista.

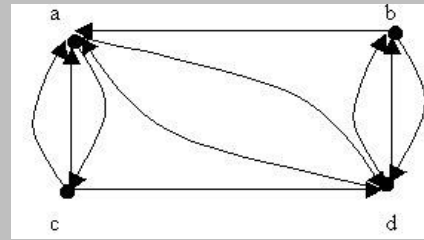
Grafo simple



Definición: Es un grafo en el cual se cumple que, para todo vértice, este tiene a lo máximo una arista. El grafo simple puede ser dirigido y no dirigido.

Invariante: La cantidad de aristas de un nodo puede ser a lo máximo 1.

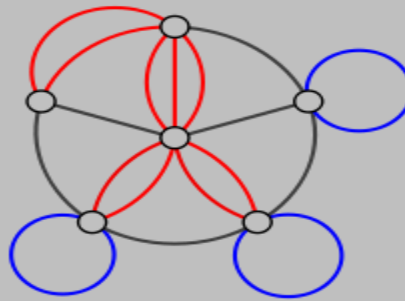
Multígrafo



Definición: Es un grafo en el cual se cumple que para todo vértice, este puede tener 0 o varias aristas, puede ser dirigido y no dirigido.

Invariante: La cantidad de aristas por nodos, puede ser tamaño variable.

Pseudografo



Definición: Es un multígrafo el cual cada vértice puede o no tener bucles, un bucle es una arista hacia el mismo.

Invariante: La cantidad de aristas por nodos puede ser variable, y el cada puede o no tener un bucle, la cantidad de bucles por nodo puede ser a lo máximo 1.

PROCEDIMIENTOS DE ACCESO

//pre: isDirected == true, el grafo será dirigido. IsDirected == false, el grafo será no dirigido.

- **createGraph (boolean isDirected)**

//post: Crea un grafo vacío.

“Crea un grafo sin vértices”.

//pre: a != null

- **dephtFirstSearch (Node a)**

//post: Si $a \in$ al grafo, y tiene vecinos, se realizara la búsqueda en profundidad, de lo contrario solo se mostrara el vértice a.

“Realiza la búsqueda en profundidad desde el nodo a”.

//pre: a != null

- **breadthFirstSearch (Node a)**

//post: Si $a \in$ al grafo, y tiene vértices adyacentes, se realizara la búsqueda en amplitud, en caso contrario solamente se mostrar el vértice a.

“Realiza la búsqueda en amplitud desde el nodo a”.

//pre: a != null, b != null, Si el grafo es dirigido se debe indicar las direcciones.

- **createEdge (Node a, Node b, Direction x, Direction y)**

Si el grafo es no dirigido, no se debe indicar las posibles direcciones.

- **createEdge (Node a, Node b)**

//post: Si $a \in$ al grafo y $b \in$ al grafo, se crea una arista entre a y b, Si no está alguno de los dos, no se realiza ninguna modificación.

“Crea una arista entre un nodo a y un nodo b.

//pre: a != null

- **addVertex(Node a)**

//post: Se agrega un nodo a la contenedora de vértices del grafo.

“Se agrega un vértice del grafo”.

//pre: a != null

- **deleteVertex (Node a)**

//post: a.key() \in al grafo, se elimina del grafo.

“Se elimina un vértice del grafo”.