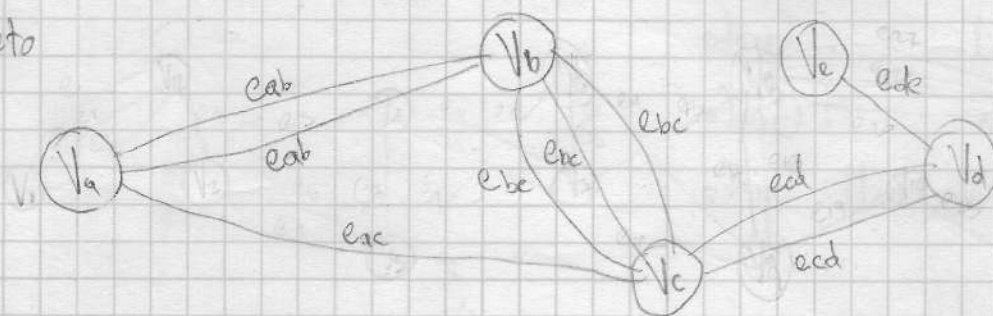


# TAD Grafo (Grafos)

Objeto



## Invariantes

- En el grafo, cualquier vertice  $V_n$  tiene al menos una arista  $e_{no}$  con al menos un vertice  $V_o$ .
- Un vertice  $V_n$  no tiene aristas de tipo  $e_{nn}$ .
- Un vertice  $V_n$  nunca va tener una arista  $e_{nn}$ , es decir, que apunte a si mismo.

## Operaciones primitivas

- Crear Grafo :  $\rightarrow$  Grafo
- Buscar Vertice : Grafo x Etiqueta  $\rightarrow$  Vertice
- Es Adyacente : Vertice1 x Vertice2  $\rightarrow$  Booleano
- Vacio : Grafo  $\rightarrow$  Booleano
- Etiqueta Arista : Grafo x Vertice1 x Vertice2  $\rightarrow$  Etiqueta
- Agregar Vertice : Grafo x Vertice  $\rightarrow$  Grafo
- Agregar Arista : Grafo x Vertice1 x Vertice2  $\rightarrow$  Grafo
- Eliminar Vertice : Grafo x Etiqueta  $\rightarrow$  Grafo
- Eliminar Arista : Grafo x Etiqueta V1 x Etiqueta V2  $\rightarrow$  Grafo
- Búsqueda en Profundidad : Grafo  $\rightarrow$  Grafo
- Búsqueda en Anchura : Grafo  $\rightarrow$  Grafo

## Operaciones principales

- Prim : Grafo
- Kruskal : Grafo
- Dijkstra : Grafo
- Floyd-Warshall : Grafo

CrearGrafo()

"Crea un grafo vacío"

pre: No había grafo

pos: Hay un grafo vacío

BuscarVertice(Grafo, Etiqueta)

"Busca en el grafo un vertice con esa etiqueta y lo devuelve si lo encuentra"

pre: El grafo existe y no es Nil

pos: El vertice buscado o Nil si no está

EsAdyacente(Vertice1, Vertice2)

"Verifica si el vertice1 comparte una arista con el vertice2"

pre: los vertices existen y no son Nil

pos: Un booleano que indica si son o no adyacentes

Vacio() (grafo)

"Verifica si el grafo está vacío"

pre: el grafo existe

pos: Un booleano que indica si el grafo está vacío

EtiquetaArista(Vertice1, Vertice2)

"Devuelve la etiqueta entre las aristas"

pre: el grafo existe, los vertices 1 y 2 tambien

pos: la etiqueta de la arista o Nil



AgregarVertice (Grafo, Vertice)

"Agrega un vertice al grafo"

pre: el grafo y el vertice existen y no son Nil

pos: El grafo con el nuevo vertice

AgregarArista (Grafo, Vertice1, Vertice2)

"Agrega una arista entre los vertices 1 y 2"

pre: El grafo y los vertices existen y no son Nil

pos: El grafo con una arista entre los vertices

EliminarVertice (Grafo, Etiqueta)

"Elimina el vertice del grafo que contenga la etiqueta"

pre: El grafo y la etiqueta existen y no son Nil

pos: El grafo sin el vertice con la etiqueta

EliminarArista (Grafo, Vertice1, Vertice2)

"Elimina la arista entre los vertices 1 y 2"

pre: El grafo y los vertices existen y son diferentes de Nil, ademàs los vertices son adyacentes.

pos: El grafo sin la arista entre los vertices

Eliminar vertice (Grafo, Etiqueta)

"Elimina un vertice del grafo con esa etiqueta"

pre: El grafo existe y es diferente de Nil

pos: El grafo sin el vertice de la etiqueta

## Operaciones primitivas

Eliminar Arista (Grafo, Vertice1, Vertice2)

"Elimina una arista entre los vertices 1 y 2"

pre: El grafo y los vertices existen y no son Nil, ademas los vertices son adyacentes.

pos: Un grafo donde los vertices 1 y 2 no son adyacentes

Busqueda En Profundidad (Grafo)

"Ejecuta el algoritmo DFS en el grafo"

pre: el grafo existe y no es Nil o vacio

pos: Un grafo resultado del DFS.

Busqueda En Amplitud (Grafo)

"Ejecuta el algoritmo BFS en el grafo"

pre: el grafo existe y no es Nil o vacio

pos: Un grafo resultado del BFS.

Prim (Grafo)

"Ejecuta el algoritmo de Prim en el grafo"

pre: El grafo existe y no es Nil o vacio.

pos: Un grafo resultado del algoritmo de Prim.

Kruskal (Grafo)

"Ejecuta el algoritmo de Kruskal en el grafo"

pre: El grafo existe y no es Nil o vacio

pos: Un grafo resultado del algoritmo de Kruskal

Operaciones primitivas

- Dijkstra (Grafo)

"Ejecuta el algoritmo de Dijkstra en el grafo"

pre: el grafo existe y no es Nil o vacío

pos: Un grafo resultado del algoritmo de Dijkstra

Floyd-Warshall

"Ejecuta el algoritmo de Floyd-Warshall en el grafo"

pre: el grafo existe y no es Nil o vacío

pos: Un grafo resultado del algoritmo de Floyd-Warshall