**Agregar**

1. Agregar un vértice cuando no hay ningún vértice en el grafo
2. Agregar un vértice cuando hay vértices en el grafo
3. Agregar un vértice repetido
4. Agregar un vértice que solo conecta con otro vértice
5. Agregar un vértice que tiene 2 o más aristas con otro.

**Buscar**

1. Cuando existen vértices a buscar
2. Cuando no existe el vértice a buscar

**Eliminar**

1. Eliminar un vértice correctamente cuando hay vértices a eliminar
2. Eliminar un vértice cuando este no existe
3. Eliminar un vertice cuando la arista sea dirigada
4. Eliminar una arista cuando esta no sea dirigidad

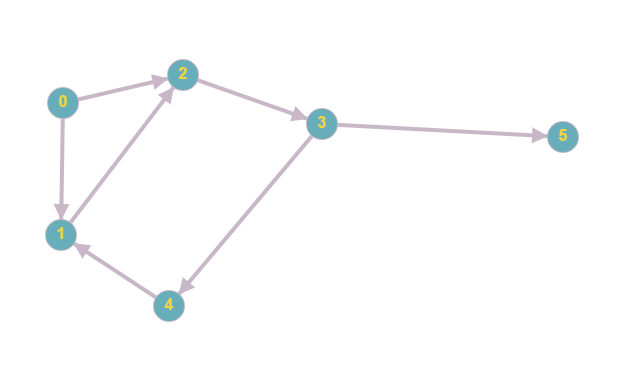
Escenario

|  |  |  |
| --- | --- | --- |
| Nombre | Clase | Escenario |
| Setup1 | GraphTest | Grafo vacio |
| Setup2 | GraphTest |  |
| Setup3 | Graph Test |  |
| Setup4 | Graph Test |  |
| Setup5 | GraphTest |  |

**Agregar**

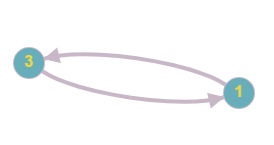
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba | Agregar un vertice al grafo cuando no existen vértices ni aristas en el grafo | | | |
| Clase | método | escenario | entrada | resultado |
| GrahpTest | Insert | Setup1 | 1 | Vértice agregado correctamente |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba | Agregar un vértice al grafo cuando existen vértices conectados con aristas en el grafo | | | |
| Clase | método | Escenario | entrada | Resultado |
| GrahpTest | Insert | Setup2 | 5 | Vertice agregado correctamente |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba | Agregar un vértice repetido al grafo | | | |
| Clase | método | Clase | entrada | Resultado |
| GrahpTest | Insert | Setup2 | 2 | El vertice no se agrega |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba | Agregar un vértice que conecta bidireccionalmente con otro vértice | | | |
| Clase | método | Clase | entrada | Resultado |
| GrahpTest | Insert | Setup2 | 3 | El vertice se agrega |



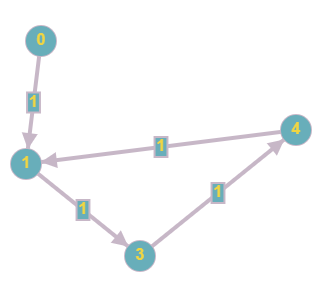
**Buscar**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba | Buscar el vértice indicado en el grafo | | | |
| Clase | método | escenario | entrada | resultado |
| GraphTest | searchVertex | Setup2 | 1 | TRUE. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba | Buscar un vértice que no existe en el grafo | | | |
| Clase | método | escenario | entrada | resultado |
| GraphTest | searchVertex | Setup2 | 10 | FALSE |

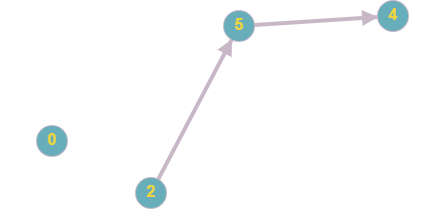
**Eliminar**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba | Eliminar el vertice indicado en el grafo | | | |
| Clase | método | escenario | entrada | resultado |
| GraphTest | DeleteVertex | Setup4 | 2 | True |



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba | Eliminar un vértice que no existe en el grafo | | | |
| Clase | método | escenario | entrada | resultado |
| GraphTest | DeleteVertex | Setup2 | 48 | FALSE |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba | Eliminar más de un vértice que existe en el grafo | | | |
| Clase | método | escenario | entrada | resultado |
| GraphTest | DeleteVertex | Setup2 | 1,3 | TRUE |

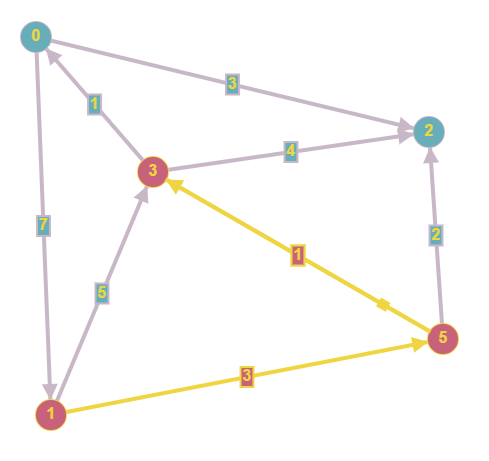


Escenarios algoritmos

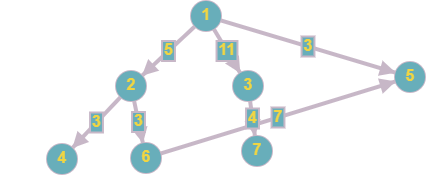
|  |  |  |
| --- | --- | --- |
| Nombre | Clase | Escenario |
| Setup1 | AlgoritmTest |  |
| Setup2 | AlgoritmTest |  |

Algoritmo Dijkstra

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba | Encontrar el camino mas corto desde 1 a 3 | | | |
| Clase | método | escenario | entrada | resultado |
| AlgoritmTest | Dijkstra | Setup1 | Grafo con 5 vertices conectados por aristas las cuales tienen determinado peso ( ver grafico) | 1=>5=>3 |



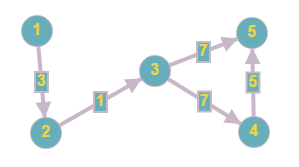
Algoritmo DFS



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba | Buscar y examinar los vértices de un grafo mediante el DFS | | | |
| Clase | método | escenario | entrada | resultado |
| AlgoritmTest | DFS | Setup3 | Grafo con 7 vértices conectados, el cual el recorrido inicia en 1. | 1,2,4,6,5,3,7 |

Algoritmo BFS (Búsqueda en anchura)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la prueba | Buscar y examinar los vértices de un grafo mediante el bfs | | | |
| Clase | método | escenario | entrada | resultado |
| AlgoritmTest | BFS | Setup2 | Grafo con 5 vertices conectados por aristas las cuales tienen determinado peso ( ver grafico) | 0,1,2,3,4,5 |



Algoritmo Floyd Warshall El algoritmo encuentra el camino entre todos los pares de vértices en una única ejecución.

Algoritmo Prim encontrar un [árbol recubridor mínimo](https://es.wikipedia.org/wiki/%C3%81rbol_recubridor_m%C3%ADnimo) en un [grafo](https://es.wikipedia.org/wiki/Grafo) [conexo](https://es.wikipedia.org/wiki/Teor%C3%ADa_de_los_grafos#Grafos_conexos), **no** dirigido y cuyas [aristas](https://es.wikipedia.org/wiki/Arista_(Teor%C3%ADa_de_grafos)) están etiquetadas.

el [algoritmo](https://es.wikipedia.org/wiki/Algoritmo) encuentra un subconjunto de [aristas](https://es.wikipedia.org/wiki/Arista_(Teor%C3%ADa_de_grafos)) que forman un [árbol](https://es.wikipedia.org/wiki/%C3%81rbol_(teor%C3%ADa_de_grafos)) con todos los [vértices](https://es.wikipedia.org/wiki/V%C3%A9rtice_(Teor%C3%ADa_de_grafos)), donde el peso total de todas las [aristas](https://es.wikipedia.org/wiki/Arista_(Teor%C3%ADa_de_grafos)) en el árbol es el mínimo posible.

Algoritmo pruskal

busca un subconjunto de aristas que, formando un árbol, incluyen todos los vértices y donde el valor de la suma de todas las aristas del árbol es el mínimo. Si el grafo no es conexo, entonces busca un bosque expandido mínimo

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Alternativa | FACTIBLE | EFICIENTE | USO Y MANEJO | CONOCIMIENTO DEL TEMA | COMPLEJIDAD | INTEGRAL | Total/Validada |
| **Solucion** |  |  |  |  |  |  |  |
| Alternativa 1 | 3 | 5 | 3 | 4 | 5 | 2 | 22 |
| Alternativa 2 | 4 | 3 | 4 | 4 | 4 | 5 | 24 Validada |
| **Ingreso de la solucion** |  |  |  |  |  |  |  |
| Alternativa 1 | 4 | 3 | 4 | 5 | 4 | 3 | 23  Validada |
| Alternativa 2 | 4 | 4 | 5 | 2 | 3 | 2 | 20 |
| Alternativa 3 | 4 | 2 | 4 | 4 | 3 | 3 | 20 |
| **Interfaz** |  |  |  |  |  |  |  |
| Alternativa 1 | 5 | 5 | 4 | 3 | 5 | 5 | 27  Validada |
| Alternativa 2 | 3 | 2 | 2 | 4 | 5 | 4 | 20 |
| **Animaciones** |  |  |  |  |  |  |  |
| Alternativa 1 | 4 | 5 | 5 | 4 | 3 | 5 | 26  Validada |
| Alternativa 2 | 3 | 3 | 5 | 4 | 4 | 3 | 22 |
| Alternativa 3 | 2 | 3 | 4 | 4 | 1 | 3 | 17 |
| **Diseño de la interfaz** |  |  |  |  |  |  |  |
| Alternativa 1 | 2 | 2 | 2 | 1 | 5 | 3 | 15 |
| Alternativa 2 | 3 | 3 | 2 | 3 | 4 | 3 | 18 |
| Alternativa 3 | 5 | 5 | 4 | 5 | 3 | 5 | 27 Validada |

|  |  |  |
| --- | --- | --- |
| TAD GRAFO | | |
| Ejemplos de un grafo dirigido y un grafo no dirigido. | Download Scientific  Diagram | | |
| Invariante | | |
| Operaciones primitivas | | |
| operación | Entradas | Salida |
| Crear Grafo | Un vértice T(int) | Grafo |
| Añadir vértice | Vértice, Vértice | Grafo |
| Añadir vértice con peso | Vértice, Vértice, inti | Grafo |
| Obtener vértice | clave Vértice | vértice |
| Obtener Vértice | Grafo | Lista de todos los vértices del grafo |
| Borrar vértice | Vértice | True |

