|  |  |
| --- | --- |
| **TAD** | Simple Graph |
| **Grafico** | **G=** |
| **Invariantes** | * No puede haber múltiples aristas entres dos vértices * No puede haber bucles en cada vértice * Debe tener más de una arista |
| **Operaciones** | * InsertVertex Vértice x Vértice x peso x dato G |
| * InsertEdge Vértice x Vértice x peso x dato G |
| * Search Vértice x G Vértice |
| * DeleteVertex Vértice x G G |
| * DeleteEdge Arista x G G |
| * BFS Vértice x G Árbol de vértices |
| * DFS Vértice X G Árbol de vértices |
| * Dijkstra Vértice X G Arreglo de distancias |
| * Floyd-Warshall Matriz de adyacencia X G Matriz de adyacencia |
| * Prim Vértice X G Matriz de Adyacencia |
| * Kruskal Vértice X G Matriz de Adyacencia |

InsertEdge(Vértice1 x Vértice2 x peso x dato)

“Inserta una arista en el grafo simple”

{Pre: G(V,E) y V1, V2 ϵ G}

{Post: G U (V1,V2) }

InsertVertex(Vértice x peso x dato)

“Inserta un vértice en el grafo simple”

{Pre: G(V,E) }

{Post: G U (V) }

Search(Dato)

“Busca un vértice en el grafo simple”

{Pre: Matriz de adyacencia diferente de null}

{Post: Vértice}

DeleteVertex(Vértice)

“Borra un vértice del grafo simple”

{Pre: Matriz de adyacencia diferente de null}

{Post: Matriz de adyacencia con el vértice borrado}

DeleteEdge(Vértice)

“Borra una arista del grafo simple”

{Pre: Matriz de adyacencia diferente de null}

{Post: Matriz de adyacencia con la arista borrada}

BFS(Vértice)

“Busca por anchura un vértice en el grafo simple”

{Pre: Matriz de adyacencia diferente de null }

{Post: Árbol o bosque}

DFS(Vértice)

“Busca por profundidad un vértice en el grafo simple”

{Pre: Matriz de adyacencia diferente de null }

{Post: Árbol o bosque}

Dijkstra(Vértice)

“Halla el camino más corto que hay desde un vértice a otro vértice en el grafo simple”

{Pre: Matriz de adyacencia diferente de null}

{Post: Arreglo de distancias}

FloydWarshall(Matriz de adyacencia)

“Halla el camino más corto de todos los vértices hacia todos los vértices en el grafo simple”

{Pre: Matriz de adyacencia diferente de null}

{Post: Matriz de adyacencia con el camino más corto de un nodo hacia todos los nodos}

Prim(Vértice)

“Halla el árbol de mínima expansión en el grafo simple”

{Pre: Matriz de adyacencia diferente de null}

{Post: Árbol o bosque}

Kruskal (T)

“Halla el árbol de mínima expansión en el grafo simple”

{Pre: Matriz de adyacencia diferente de null }

{Post: Árbol o bosque }