

TAD Graph		
Graph $G = (V, E)$		
{inv: $ V = n \wedge \{\{u, v\} / u, v \in V \rightarrow \{u, v\} \in E\}$ }		
Operaciones primitivas:		
Graph:		→ Graph
addVertex:	Vertex v	→ Graph
addEdge:	Vertex v1 X Vertex v2	→ Graph
deleteVertex:	Vertex v	→ Graph
deleteEdge:	Edge e	→ Graph
searchVertex:	Element e	→ Vertex
searchEdge:	Vertex v1 X Vertex v2	→ Edge
BFS:	Vertex v1	→ Graph
DFS:	-	→ Graph
dijkstra:	Vertex v1 X Vertex v2	→ Graph
floydWharshall:	-	→ Graph
prim:	Vertex v1	→ Graph
kruskal:	-	→ Graph
Especificación operaciones:		
Graph() Builds an empty Graph {pre: TRUE} {post: <i>Graph G</i> }		
addVertex(Vertex v) adds a Vertex v in Graph {pre: TRUE} {post: Graph G}		
addEdge(Vertex v1, Vertex v2) adds a Edge e in Graph {pre: Graph G $\wedge v1 \in V \wedge v2 \in V$ } {post: Graph G}		
deleteVertex(Vertex v) delete a Vertex v from Graph {pre: Graph G $\wedge v \in V$ } {post: Graph G}		

deleteEdge(Edge e)

delete a Edge e from Graph

{pre: Graph $G \wedge e \in E$ }

{post: Graph G }

searchVertex(Element e)

search a Vertex v from Graph

{pre: Graph $G \wedge v \in V$ }

{post: *Vertex v*}

searchEdge(Vertex v1, Vertex v2)

search an Edge {v1, v2} from Graph

{pre: Graph $G \wedge \{v1, v2\} \in E$ }

{post: *Edge e*}