

ADT Graph		
Graph = Set<Vertices> vertices, Set<Edge> edges		
{ inv: vertices != null \wedge edges != null }		
Primitive Operations		
• CreateGraph()		→ Graph
• AddVertice()	Graph X Value X Key	→ Graph
• AddEdge()	Graph X Vertice X Vertice	→ Graph
• RemoveVertice()	Graph X Key	→ Graph
• RemoveEdge()	Graph X Vertice X Vertice	→ Graph

CreateGraph()

“Create a Graph”

{ pre: Set<Vertices != isEmpty}

{ post: new Graph }

AddVertice(graph, value, key)

“add a new vertice to the graph”

{ pre: graph !=null \wedge value \in V }

{ post: newVertice = {key:key, value: value} \wedge vertice \in graph }

AddEdge(graph, origin, destination)

“add a new Edge to the graph”

{ pre: origen != null \wedge destino != null }

{ post: newEdge = { verticeA = origen, verticeB = destino } \wedge edge \in graph }

DeleteVertice(graph, key)

“identifies a vertice with the given key and removes it from the graph”

{ pre: graph != null \wedge key \in K }

{ post: vertice = {key:key, value:value} \wedge vertice \notin graph }

DeleteEdge(graph, origin, destination)

“identifies a edge with the origin vertice and destination vertice, then removes it from the graph”

{ pre: graph != null \wedge origin \in graph \wedge destination \in graph }

{ post: { verticeA = origen, verticeB = destino } \wedge edge \notin graph }