**ADTS OF PROJECT**

**Structure Package**

|  |  |
| --- | --- |
| **The IGraph ADT** | |
| **IGraph** = 〈Nodes, Edges〉 | |
| {inv 0 ≤ n ∧ Size(IGraph) = n ∧ Edges is a set of edges that connects the nodes of the graph} | |
| Operations   * builder * addNode * addEdge * getNeighbors * hasEdge * getNodes * getEdge | IGraph --🡪 IGraph  IGraph x Node --🡪 IGraph  IGraph x Edge --🡪 IGraph  IGraph --🡪 Node  IGraph --🡪 Node  IGraph --🡪 Node  IGraph --🡪 Edge |

|  |
| --- |
| **builder()** |
| Construct a vertex with a unique identifier. |
| **Pre:** - |
| **Post**: IGraph g = 〈∅, ∅〉. |

|  |
| --- |
| **addNode()** |
| Add a node to the graph. |
| **Pre:** IGraph g = 〈Nodes, Edges〉 and node n does not exist in Nodes. |
| **Post**: IGraph g = 〈Nodes ∪ {n}, Edges〉. |

|  |
| --- |
| **addEdge** |
| Adds an edge between two nodes in the graph. |
| **Pre:** IGraph g = 〈Nodes, Edges〉, There are nodes v1, v2 and there is no edge between them. |
| **Post**: IGraph g = 〈Nodes, Edges ∪ {e}〉 |

|  |
| --- |
| **getNeighbors()** |
| Returns the neighbors of a node. |
| **Pre:** IGraph g = 〈Nodes, Edges〉 and there is node n. |
| **Post:** Returns a set of neighboring nodes of n. |

|  |
| --- |
| **hasEdge()** |
| Checks to see if there is an edge between two nodes. |
| **Pre:** IGraph g = 〈Nodes, Edges〉 y v1, v2 are nodes in the graph. |
| **Post:** Returns true if an edge exists between v1 and v2, otherwise false. |

|  |
| --- |
| **getNodes()** |
| Returns the set of nodes in the graph. |
| **Pre:** IGraph g = 〈Nodes, Edges〉. |
| **Post:** Returns Nodes |

|  |
| --- |
| **getEdge()** |
| Gets an edge between two nodes |
| **Pre:** IGraph g = 〈Nodes, Edges〉 and there is an edge between v1 and v2. |
| **Post:** Returns the edge e between v1 and v2. |

|  |  |
| --- | --- |
| **The AdjacencyMatrix ADT** | |
| **AdjacencyMatrix** = 〈Matrix, Size〉 | |
| {inv Size = n ∧ Matrix es una matriz de adyacencia de tamaño n x n} | |
| Operations   * builder * addEdge * getEdgeWeight * hasEdge * getNeighbors | Queue --🡪 Queue  Queue x Element -🡪Queue  Queue -🡪 Element  Queue 🡪 Element  Queue 🡪 boolean |