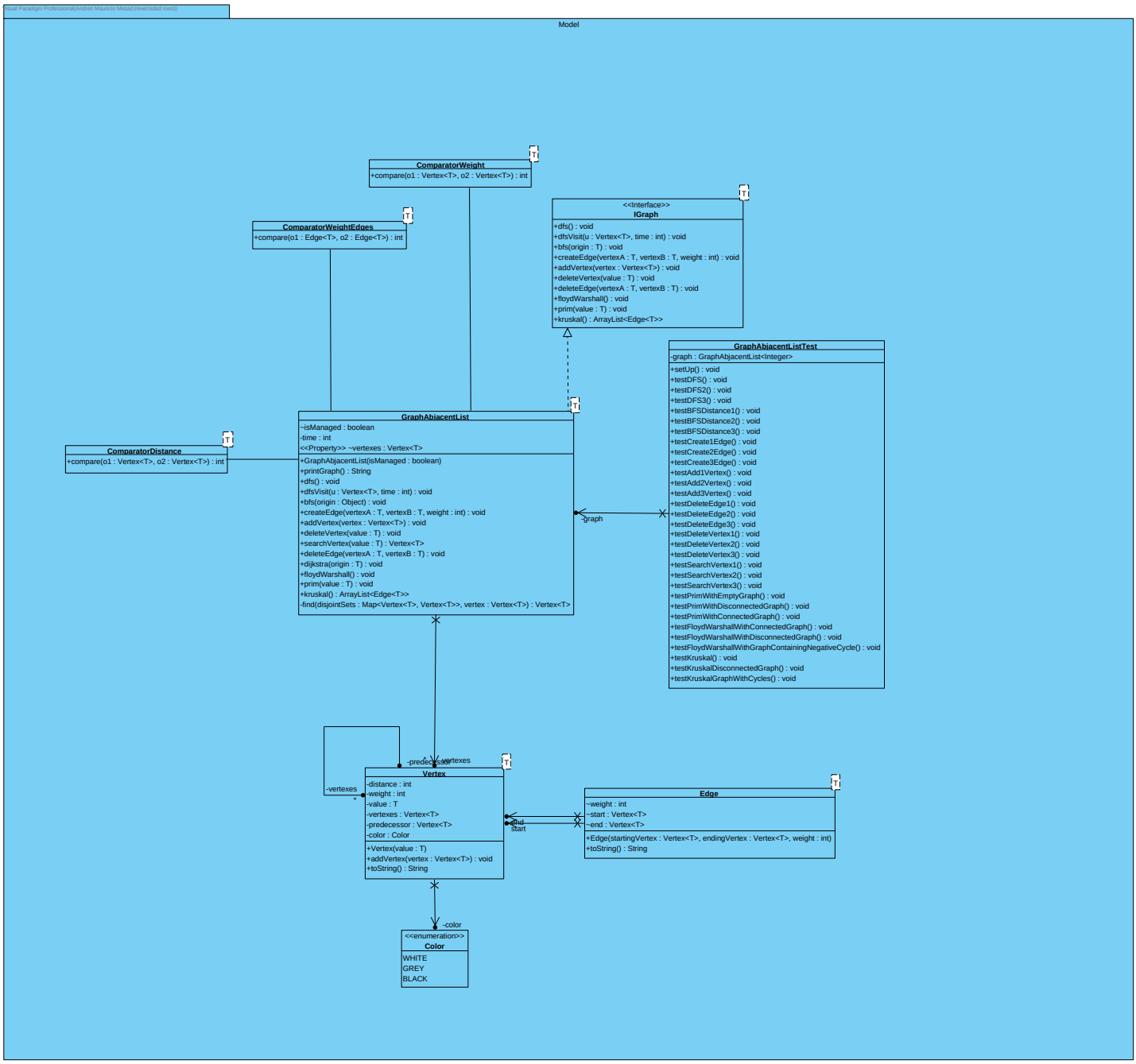
Visual Paradigm Professional(Andres Mauricio Mesa(Universidad Ices))
Model
T
ComparatorWeight
+compare(o1 : Vertex<T>, o2 : Vertex<T>) : int
T
ComparatorWeightEdges
+compare(o1 : Edge<T>, o2 : Edge<T>) : int
T
<<Interface>>
IGraph
+dfs() : void
+dfsVisit(u : Vertex<T>, time : int) : void
+bfs(origin : T) : void
+createEdge(vertexA : T, vertexB : T, weight : int) : void
+addVertex(vertex : Vertex<T>) : void
+deleteVertex(value : T) : void
+deleteEdge(vertexA : T, vertexB : T) : void
+floydWarshall() : void
+prim(value : T) : void
+kruskal() : ArrayList<Edge<T>>
GraphAbiacentListTest
-graph : GraphAbiacentList<Integer>
+setUp() : void
+testDFS() : void
+testDFS2() : void
+testDFS3() : void
+testBFSDistance1() : void
+testBFSDistance2() : void
+testBFSDistance3() : void
+testCreate1Edge() : void
+testCreate2Edge() : void
+testCreate3Edge() : void
+testAdd1Vertex() : void
+testAdd2Vertex() : void
+testAdd3Vertex() : void
+testDeleteEdge1() : void
+testDeleteEdge2() : void
+testDeleteEdge3() : void
+testDeleteVertex1() : void
+testDeleteVertex2() : void
+testDeleteVertex3() : void
+testSearchVertex1() : void
+testSearchVertex2() : void
+testSearchVertex3() : void
+testPrimWithEmptyGraph() : void
+testPrimWithDisconnectedGraph() : void
+testPrimWithConnectedGraph() : void
+testFloydWarshallWithConnectedGraph() : void
+testFloydWarshallWithDisconnectedGraph() : void
+testFloydWarshallWithGraphContainingNegativeCycle() : void
+testKruskal() : void
+testKruskalDisconnectedGraph() : void
+testKruskalGraphWithCycles() : void
T
GraphAbiacentList
~isManaged : boolean
-time : int
<<Property>> ~vertexes : Vertex<T>
+GraphAbiacentList(isManaged : boolean)
+printGraph() : String
+dfs() : void
+dfsVisit(u : Vertex<T>, time : int) : void
+bfs(origin : Object) : void
+createEdge(vertexA : T, vertexB : T, weight : int) : void
+addVertex(vertex : Vertex<T>) : void
+deleteVertex(value : T) : void
+searchVertex(value : T) : Vertex<T>
+deleteEdge(vertexA : T, vertexB : T) : void
+dijkstra(origin : T) : void
+floydWarshall() : void
+prim(value : T) : void
+kruskal() : ArrayList<Edge<T>>
-find(disjointSets : Map<Vertex<T>, Vertex<T>>, vertex : Vertex<T>) : Vertex<T>
T
ComparatorDistance
+compare(o1 : Vertex<T>, o2 : Vertex<T>) : int
-graph
-predecessor
-vertexes
-vertexes
T
Vertex
-distance : int
-weight : int
-value : T
-vertexes : Vertex<T>
-predecessor : Vertex<T>
-color : Color
+Vertex(value : T)
+addVertex(vertex : Vertex<T>) : void
+toString() : String
end
start
T
Edge
~weight : int
~start : Vertex<T>
~end : Vertex<T>
+Edge(startingVertex : Vertex<T>, endingVertex : Vertex<T>, weight : int)
+toString() : String
-color
<<enumeration>>
Color
WHITE
GREY
BLACK