GraphAdjList Unit test designs

Name	Class	Scenary
Setupscenary1	**GraphAdjListTest**	3 nodes added each one of them with a name, an average of people and an average of minutes for them.
Setupscenary1	**GraphAdjListTest**	The first node called “El ciclón” with an average people of 25 people and an average of minutes of 10.
Setupscenary1	**GraphAdjListTest**	The second node called “La cumbre” with an average of people of 40 and an average of minutes of 45.
Setupscenary1	**GraphAdjListTest**	The third node called “Los rápidos” with an average of people of 30 and an average of minutes of 25
SetUpscenary1	**GraphAdjListTest**	1 edge added each one of them with different weights and a pair of nodes
SetUpscenary1	**GraphAdjListTest**	The first edge with a weight of 12 and between “El ciclón” and “La cumbre”

GraphAdjList Test Case Design

Test objective: Validate that a node or vertex is correctly added to the graph				
Class	**Method**	**Scenary**	**Input Values**	**Result**
GraphAdjListTest	addVertex	Setupscenary1	A new node with name "Horses", average number of people "10" and average time in minutes "15".	The node “El ciclón” got added to the hash table and the attractions amount changed to amount +1.

Test objective: Verify that an edge is correctly added between 2 nodes of the graph				
Class	**Method**	**Scenary**	**Input Values**	**Result**
GraphAdjList	AddEdge	Setupscenary1	The 2 nodes to be joined by the edge and the weight of the edge	A new edge was added to the list of edges with a weight "weight" and nodes "start" and "end".

Test objective: Validate that the weights of the shortest paths between a “source” node and the rest of the nodes of the graph are correctly obtained.				
Class	**Method**	**Scenary**	**Input Values**	**Result**
GraphAdjList	Dijkstra	Setupscenary1	A "source" node and an array of keys that identify the rest of the nodes of the graph.	A "DijkstraResults" object containing an array with the path weights between the source node and each of the nodes of the graph.

Test objective: Validate that the weight of the shortest path between all nodes of the graph can be obtained.				
Class	**Method**	**Scenary**	**Input Values**	**Result**
GraphAdjList	Floyd	Setupscenary1	N/A	An mxm matrix containing the weight of the shortest path from any node to any of the other nodes in the graph.