

### Diseño de Pruebas Unitarias

Nombre	Clase	Método
<b>setUpStage1</b>	AdjListGraphTest	new AdjListGraph<String>(false)
<b>setUpStage2</b>	AdjListGraphTest	new AdjListGraph<String>(false) addVertex("Pueblo Paleta") addVertex("Ciudad Verde") addVertex("Ciudad Plateada")
<b>setUpStage3</b>	AdjListGraphTest	new AdjListGraph<String>(false) addVertex("Pueblo Paleta") addVertex("Ciudad Verde") addVertex("Ciudad Plateada") alg.addEdge("Pueblo Paleta", "Ciudad Verde", 8) alg.addEdge("Ciudad Verde", "Ciudad Plateada", 15)

Nombre	Clase	Método
<b>setUpStage1</b>	AdjMatrixGraphTest	new AdjListGraph<String>(false)
<b>setUpStage2</b>	AdjMatrixGraphTest	new AdjListGraph<String>(false) addVertex("Pueblo Paleta") addVertex("Ciudad Verde") addVertex("Ciudad Plateada")
<b>setUpStage3</b>	AdjMatrixGraphTest	new AdjListGraph<String>(false) addVertex("Pueblo Paleta") addVertex("Ciudad Verde") addVertex("Ciudad Plateada") alg.addEdge("Pueblo Paleta", "Ciudad Verde", 8) alg.addEdge("Ciudad Verde", "Ciudad Plateada", 15)

Clase	Método	Escenario	Entrada	Salida
<b>AdjListGraphTest</b>	addVertexTest()	setUpStage1	"Pueblo Paleta"	Correcto, porque "Pueblo Paleta" es el primer vértice del grafo.
<b>AdjListGraphTest</b>	addEdgeTest1()	setUpStage2	("Pueblo Paleta", "Ciudad Verde") ("Ciudad Verde", "Ciudad Plateada")	Correcto, porque se las aristas que se agregan en la prueba conectan, "Pueblo Paleta" con "Ciudad Verde" y "Ciudad Verde" con "Ciudad Plateada"
<b>AdjListGraphTest</b>	addEdgeTest2()	setUpStage2	("Pueblo Paleta", "Ciudad Verde") ("Ciudad Verde", "Ciudad Plateada")	Correcto, porque se la arista que se agrega en la prueba conecta "Pueblo Paleta" con "Ciudad Verde" y "Ciudad Verde", "Ciudad Plateada" no son adyacentes.

<b>AdjListGraphTest</b>	searchTest1()	setUpStage1	0	Correcto, porque la lista de adyacencia está vacía.
<b>AdjListGraphTest</b>	searchTest2()	setUpStage2()	("Pueblo Paleta", (0)) ("Ciudad Plateada") ("Hola")	Correcto, porque el primer vértice de la lista de adyacencia es "Pueblo Paleta" ya que es el primero agregado, "Ciudad Plateada" si se encuentra en el grafo y la última búsqueda es nula ya que no existe ("Hola")
<b>AdjListGraphTest</b>	weightMatrixTest()	setUpStage3()	("Pueblo Paleta", (0)) ("Ciudad Plateada") ("Hola")	Correcto porque los vértices agregados representan una posición en la matriz, y esa matriz tienen el peso específico de las aristas agregadas como también las posiciones restantes no tendrán un peso específico entonces será infinito.
<b>AdjListGraphTest</b>	getIndexVTest()	setUpStage2()	(0, "Pueblo Paleta") (1, "Ciudad Plateada")	Correcto, porque el primer vértice en el grafo es "Pueblo Paleta" entonces su índice será 0 como también el segundo vértice "Ciudad Plateada" será índice 1.
<b>AdjListGraphTest</b>	getVertexTest()	setUpStage2()	(3, alg.getVertex()) ("Ciudad Celeste") (4, alg.getVertex())	Correcto, porque en primera instancia el tamaño del grafo es igual a 3 vértices y después de agregar uno mas en el test completa la cantidad (4).

<b>Nombre</b>	<b>Método</b>	<b>Escenario</b>	<b>Entrada</b>	<b>Salida</b>
AdjMatrixGraphTest	addVertexTest()	setUpStage1	"Pueblo Paleta"	Correcto, porque "Pueblo Paleta" es el primer vértice del grafo y el primer agregado.
AdjMatrixGraphTest	addEdgeTest1()	setUpStage2	("Pueblo Paleta", "Ciudad Verde")	Correcto, porque se las aristas que se agregan

			("Ciudad Verde", "Ciudad Plateada")	en la prueba conectan, "Pueblo Paleta" con "Ciudad Verde" y "Ciudad Verde" con "Ciudad Plateada"
AdjMatrixGraphTest	addEdgeTest2()	setUpStage2	("Pueblo Paleta", "Ciudad Verde") ("Ciudad Verde", "Ciudad Plateada")	Correcto, porque se la arista que se agrega en la prueba conecta "Pueblo Paleta" con "Ciudad Verde" y "Ciudad Verde", "Ciudad Plateada" no son adyacentes.
AdjMatrixGraphTest	searchTest1()	setUpStage1	0	Correcto, porque la lista de adyacencia está vacía.
AdjMatrixGraphTest	searchTest2()	setUpStage2()	("Pueblo Paleta", (0)) ("Ciudad Plateada") ("Hola")	Correcto, porque el primer vértice de la lista de adyacencia es "Pueblo Paleta" ya que es el primero agregado, "Ciudad Plateada" si se encuentra en el grafo y la última búsqueda es nula ya que no existe ("Hola")
AdjMatrixGraphTest	weightMatrixTest()	setUpStage3()	("Pueblo Paleta", (0)) ("Ciudad Plateada") ("Hola")	Correcto porque los vértices agregados representan una posición en la matriz, y esa matriz tienen el peso específico de las aristas agregadas como también las posiciones restantes no tendrán un peso específico entonces será infinito.
AdjMatrixGraphTest	getIndexVTest()	setUpStage2()	(0, "Pueblo Paleta") (1, "Ciudad Plateada")	Correcto, porque el primer vértice en el grafo es "Pueblo Paleta" entonces su índice será 0 como también el segundo vértice "Ciudad Plateada" será índice 1.

AdjMatrixGraphTest	getVertexTest()	setUpStage2()	(3, alg.getVertex()) ("Ciudad Celeste") (4, alg.getVertex())	Correcto, porque en primera instancia el tamaño del grafo es igual a 3 vértices y después de agregar uno mas en el test completa la cantidad (4).
--------------------	-----------------	---------------	--	---