**Casos de prueba:** teniendo en cuenta que tenemos que hacer casos de prueba para TAD grafo y sus algoritmos, y para la solución del problema, separamos los casos de pruebas en dos partes. Para esto debemos tener en cuenta que todos los casos de prueba cuentan con grafos ponderados dirigidos (Nota: nuestro algoritmo también permite trabajar con grafos no dirigidos)

**SetUp TAD grafo y sus algoritmos:**

|  |  |  |
| --- | --- | --- |
| Nombre | Clase | Escenario |
| setupStage1() | MatrixGraph | Se crea un escenario en el que existen 3 vértices con las siguientes conexiones:  1-2(peso: 2)  2-3(peso: 1)  2-1(peso: 2)  3-1(peso: 8)  3-2(peso: 3) |
| setupStage2() | ListGraph | Se crea un escenario en el que existen 3 vertices con las siguientes conexiones:  1-2(peso: 2)  2-3(peso: 1)  2-1(peso: 2)  3-1(peso: 8)  3-2(peso: 3) |

**Casos de prueba para TAD grafo y sus algoritmos:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Clase | Metodo | Setup | Entrada | Salida |
| MatrixGraph | addVertex | setupStage1() | Un vertice nuevo | Una matriz con 4 vertices |
| MatrixGraph | floydWarshall | setupStage1() | ir del vertice 3, al 1 | el camino mas corto siendo: 3-2-1 con un peso de: 5 |
| ListGraph | addVertex | setupStage2() | Un vertice nuevo | Una lista con 4 vertices |
| ListGraph | dijkstra | setupStage2() | ir del vertice 3, al 1 | el camino más corto siendo: 3-2-1 con un peso de: 5 |
| MatrixGraph | addEdge | setupStage1() | Se agrega la siguiente conexión:  1-3(peso: 1) | La posición en la matriz [1][3] = 1 |
| ListGraph | addEdge | setupStage2() | Se agrega la siguiente conexión:  1-3(peso: 1) | Vértice 1 con una arista mas |

**SetUp Solución del problema:**

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre | Clase | | Escenario |
| setupStage1() | DistributionAlgorithms | Se crean 4 oficinas:  Office1:  City = San Francisco  Products = 100  Office2:  City = New York  Products = 123  Office3:  City = Chicago  Products = 231  Office4:  City = Miami  Products = 98  Con las siguientes conexiones:  San Francisco – Miami (peso: 121)  San Francisco – New York (peso: 190)  Chicago – Miami (peso: 12)  Chicago – New York (peso: 123)  New York – Miami (peso: 270)  New York – Chicago (peso: 98)  Miami - San Francisco (peso: 129) | |

**Casos de prueba para Solución del problema:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Clase | Metodo | Setup | Entrada | Salida |
| DistributionAlgorithms | determineShortestPath | setupStage1 | Ir de New York a miami | New york ---- Chicago---- Miami  Un peso de: 110 |
| DistributionAlgorithms | determineMeanProductsPerGroup | setupStage1 |  | Un solo grupo con 4 oficinas dentro. Promedio = 138 |