**Configuración de los Escenarios**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setupScenary1 | GraphAlgorithmsTest | Se crea el escenario con un nuevo Objeto tipo GraphAlgorithm para pruebas correctas |
| setupScenary2 | GraphAlgorithmsTest | Se crea un escenario con un grafo predefinido para pruebas de algoritmos sobre grafos |
| setupScenary3 | GraphAlgorithmsTest | Se crea un objeto grafo con algunos nodos para probar ciertos métodos |
| setupScenary4 | GraphAlgorithmsTest | vacio |

**Diseño de Casos de Prueba**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar que se añade y elimina exitosamente un nodo al grafo | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| AdjacenteMatrixTest | addVertex(Node T) | setupScenary1 | Ciudad(El nombre de la ciudad como Nodo) | Un primer nodo ha sido añadido exitosamente al grafo |
| AdjacenteMatrixTest | removeVertex(Node T) | setupScenary1 | Ciudad(El nombre de la ciudad como Nodo) | Un primer nodo ha sido eliminado exitosamente del grafo |
| AdjacenteListTest | addVertex(Node T) | setupScenary1 | Ciudad(El nombre de la ciudad como Nodo) | Un primer nodo ha sido añadido exitosamente al grafo |
| AdjacenteListTest | removeVertex(Node T) | setupScenary1 | Ciudad(El nombre de la ciudad como Nodo) | Un primer nodo ha sido eliminado exitosamente del grafo |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Añadir un camino al grafo (Teniendo en cuenta la condición de que se necesitan mínimo dos vértices añadidos previamente) | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| AdjacencyMatrixGraph | addEdge(Node A,Node B) | setupScenary3 | Node A,Node B | Se crea una camino bidireccional sin peso. |
| AdjacencyMatrixGraph | addEdge(Node A,int a ,Node B) | setupScenary3 | Node A,Node B,  int a, | Se crea una camino bidireccional con peso. |
| AdjacencyListGraph | addEdge(Node A,Node B) | setupScenary3 | Node A,Node B | Se crea una camino bidireccional sin peso. |
| AdjacencyListGraph | addEdge(Node A,int a ,Node B) | setupScenary3 | Node A,Node B,  int a, | Se crea una camino bidireccional con peso. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Eliminar un camino entre 2 nodos | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| AdjacencyMatrixGraph | RemoveEdge(Node A, Node B) | setupScenary2 | Node A, Node B | Se elimina el camino desde el Node A hasta el Node B |
| AdjacencyMatrixGraph | RemoveEdge(Node A, Node B) | setupScenary2 | Node A, Node B | Se elimina el camino desde el Node A hasta el Node B |
| AdjacencyListGraph | RemoveEdge(Node A, Node B) | setupScenary2 | Node A, Node B | Se elimina el camino desde el Node A hasta el Node B |
| AdjacencyListGraph | RemoveEdge(Node A, Node B) | setupScenary2 | Node A, Node B | Se elimina el camino desde el Node A hasta el Node B |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar si un nodo está balanceado | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
| Node | isWeighted() | setupScenary2 | Ninguno | Permite ver si el nodo del grafo está balanceado |
| Node | isWeight() | setupScenary1 | Ninguno | Debería salir una excepción porque el grafo no tiene ningún nodo para evaluar. |