**Diseño de Pruebas unitarias.**

Especificación de Escenarios de prueba:

|  |  |  |
| --- | --- | --- |
| Nombre: | Clase: | Especificación: |
| Stage1 | Date | Vacío |
| Stage2 | Flight | Vacío |
| Stage3 | Airport | Vacío |
| Stage4 | Airport | Una instancia de Airport ya ha sido creada. |
| Stage5 | Airport |  |

Diseño de pruebas:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prueba No 1: | Objetivo: Probar las operaciones de agregar de la interfaz IGraph. | | | |
| Clase | **Método** | **Escenario** | **Entradas** | **Salidas** |
| IGraphTest | addVertex(V u) | Stage1 | u : V; u no se encuentra en el grafo. | true |
| IGraphTest | addVertex(V u) | Stage1 | u : V; u se encuentra en el grafo. | false |
| IGraphTest | addEdge(V u, V v) | Stage1 | u : V; v : V; g no ponderado; u no pertenece al grafo. | ElementNotFoundException |
| IGraphTest | addEdge(V u, V v) | Stage2 | u : V; v : V; g no ponderado; v no pertenece al grafo. | ElementNotFoundException |
| IGraphTest | addEdge(V u, V v) | Stage3 | u : V; v : V; g no ponderado; g no dirigido. | true |
| IGraphTest | addEdge(V u, V v) |  | u : V; v : V; g no ponderado; g dirigido. | true |
| IGraphTest | addEdge(V u, V v) |  | u : V; v : V; u.equals(v) (admite bucles). | true |
| IGraphTest | addEdge(V u, V v, double w) |  | u : V; v : V; w : double; u no pertenece al grafo. | ElementNotFoundException |
| IGraphTest | addEdge(V u, V v, double w) |  | u : V; v : V; w : double; v no pertenece al grafo. | ElementNotFoundException |
| IGraph | addEdge(V u, V v, double w) |  | u : V; v : V; w : double; el grafo no es ponderado. | WrongEdgeTypeException |
| IGraphTest | addEdge(V u, V v, double w) |  | u : V; v : V; w : double. | true |
| IGraphTest | addEdge(V u, V v, double w) |  | u : V; v : V; w : double; u.equals(v) (admite bucles). | true |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prueba No 2: | Objetivo: Probar las operaciones de eliminar de la interfaz IGraph. | | | |
| Clase | **Método** | **Escenario** | **Entradas** | **Salidas** |
| IGraphTest | removeVertex(V v) | Stage1 | 1000 | Ninguna de las 1000 entradas es creada con errores, todos los vuelos tienen un identificador diferente. |
| IGraphTest |  |  |  |  |
| IGraphTest |  |  |  |  |
| IGraphTest |  |  |  |  |
| IGraphTest |  |  |  |  |
| IGraphTest |  |  |  |  |
| IGraphTest |  |  |  |  |
| IGraphTest |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prueba No 3: | Objetivo: Probar que los métodos de búsqueda y ordenamiento funcionan correctamente. | | | |
| Clase | **Método** | **Escenario** | **Entradas** | **Salidas** |
| IGraphTest | search(Sortings, String) | Stage5 | Sortings.AIRLINE, “Avianca” | Una lista de un único elemento con el tercer elemento de la lista original en ella. |
| IGraphTest | search(Sortings, String) | Stage5 | Sortings.AIRLINE, “Ecolines” | Se entrega una lista vacía. |
| IGraphTest | Search(Sortings, String) | Stage5 | Sortings.DESTINATION, “Bogota” | Una lista de un único elemento con el cuarto elemento de la lista original en ella. |
| IGraphTest | search(Sortings, String) | Stage5 | Sortings.DESTINATION, “Ciudad de Mexico” | Se entrega una lista vacía. |
| IGraphTest | Search(Sortings, String) | Stage5 | Sortings.FLIGHT\_NUMBER, “C80192” | Una lista de un único elemento con el segundo elemento de la lista original en ella. |
| IGraphTest | Search(Sortings, String) | Stage5 | Sortings.FLIGHT\_NUMBER, “D80192” | Se entrega una lista vacía. |
| IGraphTest | Search(Sortings, String) | Stage5 | Sortings.BOARDING\_GATE, 4 | Una lista de un único elemento con el cuarto elemento de la lista original en ella. |
| AirportTest | Search(Sortings, String) | Stage5 | Sortings.BOARDING\_GATE, 5 | Se entrega una lista vacía. |
| AirportTest | Sort() | Stage5 | Sortings.AIRLINE | La lista se ordena en el siguiente orden: cuarto, tercero, segundo, primero. |
| AirportTest | Sort() | Stage5 | Sortings.DESTINATION | La lista se ordena en el siguiente orden: cuarto, primero, segundo, tercero. |
| AirportTest | Sort() | Stage5 | Sortings.FLIGHT\_NUMBER | La lista se ordena en el siguiente orden: tercero, cuarto, segundo, primero. |
| AirportTest | Sort() | Stage5 | Sortings.BOARDING\_GATE | La lista se ordena en el siguiente orden: primero, segundo, tercero, cuarto. |