**Computación y Estructuras Discretas I**

**Formato de escenarios y casos de uso - Tarea Integradora I**

**Configuración de los Escenarios**

| **Nombre** | **Clase** | **Escenario** |
| --- | --- | --- |
| TestAddAndGet | HashTableTest | Prueba que verifica si un elemento es insertado correctamente en la tabla hash y que se puede recuperar utilizando la misma clave |
| TestRemove | HashTableTest | Prueba que verifica que un elemento es eliminado correctamente de la tabla hash |
| TestAddMultipleElements | HashTableTest | Prueba para agregar varios elementos y verificar que el tamaño aumente correctamente |
| TestHash | HashTableTest | Probar que los hashes para varias claves sean distintos |
| TestGet | HashTableTest | Probar que funcion el get pero insertando un valor-clave |

| **Nombre** | **Clase** | **Escenario** |
| --- | --- | --- |
| TestIsEmpty | PQueueTest | Verificar si la cola de prioridad está vacía o no. Se crea una cola de prioridad vacía y se verifica que está vacía. Luego se inserta un elemento en la cola de prioridad y se verifica que no está vacía |
| TestInsert | PQueueTest | Probar la funcionalidad de insertar elementos en la cola de prioridad. Se crea una cola de prioridad vacía, se insertan algunos elementos y se verifica que la cola de prioridad no está vacía y que tiene el tamaño correcto. |
| TestRemove | PQueueTest | Probar la funcionalidad de eliminar elementos de la cola de prioridad. Se insertan algunos elementos en la cola de prioridad y luego se eliminan uno por uno para verificar que se eliminen en el orden correcto. Además, se verifica que una excepción sea lanzada cuando se intenta eliminar un elemento de una cola de prioridad vacía |
| TestRandomInsertionAndRemoval | PQueueTest | Probar la funcionalidad de insertar y eliminar elementos de la cola de prioridad de forma aleatoria. Se insertan 1000 elementos aleatorios en la cola de prioridad y se verifica que se eliminen en el orden correcto. |

| **Nombre** | **Clase** | **Escenario** |
| --- | --- | --- |
| TestEnqueue | QueueTest | Se crea una cola vacía, se agrega un elemento y se verifica que la cola no esté vacía y que el elemento agregado sea el primero en la cola. |
| TestDequeue | QueueTest | Se crea una cola con tres elementos y se verifica que el primer elemento de la cola sea igual al primer elemento encolado (1), luego se elimina el primer elemento de la cola y se verifica que el siguiente elemento sea igual al segundo elemento encolado (2). |
| TestPeek | QueueTest | Se crea una cola con cinco elementos y se verifica que el primer elemento de la cola sea igual al primer elemento encolado (1). |
| TestIsEmpty | QueueTest | Se crea una cola vacía, se verifica que esté vacía, se agrega un elemento y se verifica que no esté vacía, finalmente se elimina el elemento y se verifica que esté vacía de nuevo. |

**Diseño de Casos de Prueba**

| **Objetivo de la Prueba:** Probar la funcionalidad de la estructura de datos hash table. | | | | |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| HashTableTest | TestAddAndGet | testAddCorrectly | Elementos(clave, valor) tipo < String e Integer> | El elemento es agregado exitosamente. |
| HashTableTest | TestRemove | testRemoveElement | Elementos(clave, valor) tipo < String e Integer> | Se elimina el elemento correctamente. |
| HashTableTest | TestAddMultipleElements | testMultipleElements | Elementos(clave, valor) tipo < String e Integer> | Se agregan los elementos correctamente y el tamaño aumenta |
| HashTableTest | TestHash | testHashing | Elementos(clave, valor) tipo < String e Integer> | Se muestra que el hash de la misma clave sea consistente |
| HashTableTest | TestGet | testGetHash | Elementos(clave, valor) tipo < String e Integer> | Se comprueba que se pueden recuperar los valores utilizando las claves correspondientes |

| **Objetivo de la prueba:** Probar la funcionalidad de una PriorityQueu.. | | | | |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| PQueueTest | TestIsEmpty | testEmpty | Un valor de entrada tipo Integer | Verificar si la cola de prioridad está vacía o no. |
| PQueueTest | TestInsert | testInsert | Un valor de entrada tipo Integer | Se insertan algunos elementos y se verifica que la cola de prioridad no está vacía y que tiene el tamaño correcto. |
| PQueueTest | TestRemove | testRemove | Un valor de entrada tipo Integer | Se elimina elementos de la cola de prioridad |
| PQueueTest | TestRandomInsertionAndRemoval | testInsertRandomAndRemove | Un valor de entrada tipo Integer | Se inserta y elimina elementos de la cola de prioridad de forma aleatoria. |

| **Objetivo de la prueba:** Probar la funcionalidad del buscador de órdenes dentro de la aplicación | | | | |
| --- | --- | --- | --- | --- |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| QueueTest | TestEnqueue | testAddElementToQueue | Un valor de entrada tipo Integer | Se crea una cola vacía, se agrega un elemento y se verifica que la cola no esté vacía y que el elemento agregado sea el primero en la cola. |
| QueueTest | TestDequeue | testDeleteElementDequeue | Un valor de entrada tipo Integer | Se crea una cola con tres elementos y se verifica que el primer elemento de la cola sea igual al primer elemento encolado (1), luego se elimina el primer elemento de la cola y se verifica que el siguiente elemento sea igual al segundo elemento encolado. |
| QueueTest | TestPeek | testToPeek | Un valor de entrada tipo Integer | Se crea una cola con cinco elementos y se verifica que el primer elemento de la cola sea igual al primer elemento encolado.. |
| QueueTest | TestIsEmpty | testToSeeIfitisEmpty | Un valor de entrada tipo Integer | Se creará una cola vacía, se verificará que esté vacía, se agrega un elemento y se verificará que no esté vacía, finalmente se elimina el elemento y se verifica que esté vacía de nuevo. |