**Diseño de Casos de Prueba**

**HashTable**

**Pruebas para el método add**

**Configuración del Escenario**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUp1Hash() | HashTable | hashtable = newHashTable<>(10); |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar que se agregue correctamente un elemento a una tabla hash vacia. Es el caso mas sencillo. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| HashTable | add(K key, V value) | setUp1Hash | Llave: “uno” valor: 1 | Al buscar en la tabla hash la clave “uno” debemos obtener el valor 1 pues la tabla estaba vacia, garantizando que no hay colisiones. |

**Configuración del Escenario**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUp2Hash() | HashTable | hashtable = newHashTable<>(10);  hashtable.add("uno", 1); |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar que al *agregar un elemento con la misma clave se esté llevando a cabo el encadenamiento de manera apropiada*. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| HashTable | add(K key, V value) | setUp2Hash | Llave: “uno” valor: 343 | Al buscar en la tabla hash la clave “uno” debemos obtener el valor 1. Sin embargo, al eliminar de la tabla ingresando dicha clave y luego volviendo a buscar debemos obtener el valor 343. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar que al intentar *agregar un elemento con una clave nula se retorne false y no se haga ninguna operación.* | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| HashTable | add(K key, V value) | setUp1Hash | Llave: null valor: 343 | false |

**Pruebas para el método search**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar el comportamiento de buscar un elemento que exista en la tabla hash. Es el caso más sencillo. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| HashTable | search(K key) | setUp2Hash | Llave: “uno” | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar el comportamiento de buscar un elemento que no exista en la tabla hash. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| HashTable | search(K key) | setUp2Hash | Llave: “dos” | null |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar el comportamiento de buscar un elemento en una tabla hash vacía. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| HashTable | search(K key) | setUp1Hash | Llave: “uno” | null |

**Pruebas para el método delete**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar el comportamiento de eliminar un elemento que exista en la tabla hash. Es el caso más sencillo. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| HashTable | delete(K key) | setUp2Hash | Llave: “uno” | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar el comportamiento de eliminar un elemento que no exista en la tabla hash. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| HashTable | delete (K key) | setUp2Hash | Llave: “dos” | null |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar el comportamiento de eliminar un elemento en una tabla hash vacía. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| HashTable | search(K key) | setUp1Hash | Llave: “uno” | null |

**Queue**

**Pruebas para el método enqueue**

**Configuración del Escenario**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUp1Queue() | Queue | Queue<Integer> queue = new Queue<>(); |

**Configuración del Escenario**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUp2Queue() | Queue | Queue<Integer> queue = new Queue<>();  queue.enqueue(1); |

**Configuración del Escenario**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setUp3Queue() | Queue | Queue<Integer> queue = new Queue<>();  queue.enqueue(1);  queue.enqueue(2);  queue.enqueue(3);  queue.enqueue(4);  queue.enqueue(5); |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar el comportamiento de Insertar un elemento en una cola vacía. Debe ser por lo tanto, el frente y back de la cola. Por lo que cuando se desencole debemos obtener dicho elemento. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Queue | enqueue | setUp1Queue() | 1 | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar el comportamiento de Insertar múltiples elementos en una cola vacía. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Queue | enqueue | setUp1Queue() | 1, 2, 3, 4, 5, 6, 7 | 1, 2, 3, 4, 5, 6, 7 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar el comportamiento de Insertar un elemento nulo en una cola vacía. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Queue | enqueue | setUp1Queue() | null | null |

**Pruebas para el método dequeue**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar el comportamiento de eliminar un elemento de una cola con un solo elemento. Es el mas sencillo. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Queue | enqueue | setUp2Queue() |  | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar el comportamiento de eliminar múltiples elementos hasta vaciar la cola | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Queue | enqueue | setUp2Queue() |  | 1, 2, 3, 4, 5 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar el comportamiento de eliminar un elemento de una cola vacia. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Queue | enqueue | setUp1Queue() |  | null |

**Pruebas para el método peek**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar el comportamiento de examinar el peek o primer elemento de una cola con un elemento. Es el mas sencillo. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Queue | enqueue | setUp2Queue() |  | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar el comportamiento de examinar el peek o primer elemento de una cola vacia. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Queue | enqueue | setUp1Queue() |  | null |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** Verificar el comportamiento de examinar el peek o primer elemento de una cola con varios elementos. La idea es comprobar la idempotencia de devolver el mismo resultado sin modificar la cola. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Queue | enqueue | setUp3Queue() |  | 1, 1, 1, 1, 1 |

**Stack**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
|  |  |  |  |  |

**Max-heap**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
|  |  |  |  |  |

**PriorityQueue**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado** |
|  |  |  |  |  |