**Formato de escenarios y casos de uso**

**Configuración de los Escenarios**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| caseSetup1 | HashTable | **C**rea la hash definiendo el elemento String y el key como un entero |
| caseSetup2 | HashTable | Crea la hash definiendo sus elemento y su key con el signo de ? |
| caseSetup3 | HashTable | **C**rea la hash definiendo el elemento String y el key como un String |

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| caseSetup1 | Queue | **Crea la cola cola definiendo el primer elemento como String y el segundo como ?** |
| caseSetup2 | Queue | **Crea la cola cola definiendo el primer elemento como String y el segundo como String y encola un elemento** |

…

\* El nombre de los escenarios puede ser setupStage1, setupStage2, etc.

\* La clase es la clase de testing correspondiente al modelo donde acontece el escenario. Por ejemplo si usted está probando User, clase será UserTest.

\* El escenario es la descripción de las condiciones iniciales del escenario.

**Diseño de Casos de Prueba**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** El propósito de este test es verificar si la implementación de la clase HashTable es capaz de agregar nuevos elementos a la tabla correctamente y de buscarlos posteriormente. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| HashTable | add  search | caseSetup1 | 2,2 | True |

,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:**  Este test tiene como propósito verificar si la implementación de la clase HashTable es capaz de manejar de manera genérica diferentes tipos de datos al agregarlos a la tabla y no lanza una excepción inesperada. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| HashTable | search | caseSetup3 | Ninguno | El elemento no se encontro |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:**  El propósito de este test es verificar si el método de búsqueda (search) de la clase HashTable está funcionando correctamente. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| HashTable | search | caseSetup3 | 1,2,3 | True |

…

\* Una prueba se compone de un conjunto de casos de prueba.

\* Cada fila representa un ***caso de prueba*** difente

\* En el objetivo de la prueba debe escribir una descripción sobre qué es lo que específicamente está probando del modelo del programa.

\* La clase es la clase del modelo que está siendo puesto a prueba.

\* El método es específicamente el método de la clase que está siendo puesto a prueba.

\* El escenario se refiere al nombre del escenario que usted definió. Todos los casos de prueba corresponden a escenarios.

\* Los valores de entrada son valores que entran al método puesto a prueba.

\* El resultado esperado es lo que se espera que suceda luego de ejecutar el método

.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** El propósito de este test es verificar si la implementación de la clase Queue es capaz de mostrar la valides de la cola si esta vacia o de lo contrario tiene objetos | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Queue | isEmpty | caseSetup1 | Ninguno | True |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:** El propósito de este test es verificar si la implementación de la clase Queue es capaz de mostrar la valides de la cola si esta vacia o de lo contrario tiene objetos con una bandera que nos dice si la cola esta o no vacia | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Queue | enqueue  isEmpty | caseSetup1 | Element, key | True |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo de la Prueba: El propósito de este test es verificar si el método dequeue() de la clase Queue está eliminando correctamente el primer elemento de la cola y si la cola está vacía después de eliminar ese elemento. | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Queue | dequeue  isEmpty | caseSetup2 | Ninguno | True |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la Prueba:**  El propósito de este test es verificar si el método enqueue() de la clase Queue está agregando correctamente un elemento a la cola y si los elementos se eliminan en el orden correcto utilizando el método dequeue(). | | | | |
| **Clase** | **Método** | **Escenario** | **Valores de Entrada** | **Resultado esperado** |
| Queue | enqueue  dequeue | caseSetup2 | ObjectB , KeyB | True |