**Diseño de los casos de prueba.**

**Clase:** HashTable.

**Prueba:** El método permite buscar a un cliente dentro de la Hash table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método probado** | **Escenario** | **Valores de entrada** | **Resultado** |
| HashTable | Search (K key) | Setup (): Se crea una nueva tabla Hash a la cual se agregan 7 clientes con una clave asignada. | Una clave K la cual tiene asignado un usuario. | Verdadero. |

**Clase:** HashTable.

**Prueba:** Comprobar que no pueden agregar elementos repetidos a la tabla Hash.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método probado** | **Escenario** | **Valores de entrada** | **Resultado** |
| HashTable | Add (K key, V value) | Setup (): Se crea una nueva tabla Hash a la cual se agregan 7 clientes con una clave asignada. | Un cliente con valor asignado “c” y un valor de “Violetta” | Verdadero. |

**Clase:** HashTable.

**Prueba:** El método permite eliminar clientes almacenados en la Hash table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método probado** | **Escenario** | **Valores de entrada** | **Resultado** |
| HashTable | Delete (K key) | Setup (): Se crea una nueva tabla Hash a la cual se agregan 7 clientes con una clave asignada. | Se ingresan diferentes claves con valor {“a”, “c”, “e”, “g”, “i”, “k”, “m”} | Verdadero. |

**Clase:** Queue.

**Prueba:** El método permite obtener el primer elemento de la cola.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método probado** | **Escenario** | **Valores de entrada** | **Resultado** |
| Queue | Front () | Setup1 (): Se crea una nueva cola vacía. | Ninguno. | Falso. |
| Queue | Front () | Setup2 (): Se crea una nueva cola vacía y se agrega una cadena. | Ninguno. | Verdadero. |

**Clase:** Queue.

**Prueba:** El método permite agregar un elemento a la cola.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método probado** | **Escenario** | **Valores de entrada** | **Resultado** |
| Queue | Enqueue () | Setup2 (): Se crea una nueva cola vacía y se agrega una cadena. | Una cadena con valor “Davif Fiat”. | Verdadero. |

**Clase:** Queue.

**Prueba:** El método permite eliminar el primer elemento de la cola.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método probado** | **Escenario** | **Valores de entrada** | **Resultado** |
| Queue | Dequeue () | Setup3 (): Se crea una nueva cola vacía y se agregan 3 elementos. | Ninguno. | Verdadero. |

**Clase:** Stack.

**Prueba:** El método permite retornar el último elemento de la pila.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método probado** | **Escenario** | **Valores de entrada** | **Resultado** |
| Stack | Top () | Setup1 (): Se crea una nueva pila vacía y se agregan 3 elementos. | Ninguno. | Verdadero. |

**Clase:** Stack.

**Prueba:** El método permite agregar un elemento a la pila.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método probado** | **Escenario** | **Valores de entrada** | **Resultado** |
| Queue | Push () | Setup1 (): Se crea una nueva pila vacía y se agregan 3 elementos. | Se agrega un elemento con valor “Camila Giraldo” | Verdadero. |

**Clase:** Stack.

**Prueba:** El método permite eliminar el último elemento de la pila.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clase** | **Método probado** | **Escenario** | **Valores de entrada** | **Resultado** |
| Queue | Pop () | Setup1 (): Se crea una nueva pila vacía y se agregan 3 elementos. | Ninguno. | Verdadero. |