Diseño de casos de prueba

**Estructura: Hash Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Escenario | Entrada | Salida |
| Search() | Se crea una Hash Table con los valores {<122,5>, <12,3>, <542,1>, <788,2>, <321,4>, <254,6>, <311,7>, <451,8>, <39,9>} | Se ingresa la clave 542 | Retorna el elemento 1 correctamente. |
| Delete() | Se crea una Hash Table con los valores {<122,5>, <12,3>, <542,1>, <788,2>, <321,4>, <254,6>, <311,7>, <451,8>, <39,9>} | Se ingresa la clave 122 | El elemento <122,5> ha sido eliminado correctamente. |
| Push() | Se crea una Hash Table vacía. | Se agregan los valores {<122,5>, <12,3>, <542,1>, <788,2>, <321,4>, <254,6>, <311,7>, <451,8>, <39,9>} | La Hash Table queda con los valores {<122,5>, <12,3>, <542,1>, <788,2>, <321,4>, <254,6>, <311,7>, <451,8>, <39,9>} |

**Estructura: Stack**

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Escenario | Entrada | Salida |
| Pop() | Se crea una Pila con los valores {4, 200, -1, 0, 5, 99, 4, 14, 7, 510, 1} | Ninguna | Retorna el elemento 1 correctamente, el tamaño disminuye 1. |
| Peek() | Se crea una Pila con los valores {4, 200, -1, 0, 5, 99, 4, 14, 7, 510, 1} | Ninguna | Retorna el elemento 1 correctamente. |
| Push() | Se crea una Pila vacía. | Se agregan los valores {4, 200, -1, 0, 5, 99, 4, 14, 7, 510, 1} | La Pila queda con los valores {4, 200, -1, 0, 5, 99, 4, 14, 7, 510, 1} |

**Estructura: Queue**

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Escenario | Entrada | Salida |
| Poll() | Se crea una Cola con los valores {4, 200, -1, 0, 5, 99, 4, 14, 7, 510, 1} | Ninguna | Retorna el elemento 4 correctamente, el tamaño disminuye 1. |
| Peek() | Se crea una Cola con los valores {4, 200, -1, 0, 5, 99, 4, 14, 7, 510, 1} | Ninguna | Retorna el elemento 4 correctamente. |
| Offer() | Se crea una Cola vacía. | Se agregan los valores {4, 200, -1, 0, 5, 99, 4, 14, 7, 510, 1} | La Cola queda con los valores {1, 510, 7, 14, 4, 99, 5, 0, -1, 200, 4} |

**Estructura: Priority Queue**

|  |  |  |  |
| --- | --- | --- | --- |
| Método | Escenario | Entrada | Salida |
| Poll() | Se crea una PriorityQueue con los valores {5, 3, 10, 1, 8, 9, 3, 15, 0, 6} | Ninguna | Retorna el menor elemento, que es 0 correctamente y el tamaño disminuye 1. |
| Peek() | Se crea una PriorityQueue con los valores {5, 3, 10, 1, 8, 9, 3, 15, 0, 6} | Ninguna | Retorna el menor elemento, que es 0, sin eliminarlo. |
| Offer() | Se crea una PriorityQueue vacía. | Se agregan los valores {5, 3, 10, 1, 8, 9, 3, 15, 0, 6} | La Pila queda con los valores {0, 1, 3, 3, 5, 6, 8, 9, 10, 15} |