**Configuración de los escenarios.**

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setupScenary1 | Cola |  |
| setupScenary2 | Cola | Hay una cola de tipo Integer con 4 elementos agregados en el siguiente orden. |

**Diseño de las pruebas.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la prueba:** Verificar el correcto funcionamiento de método insertar() (enqueue) de la clase Cola, cuando:   1. La cola se encuentre vacía. 2. La cola tenga 5 elementos. | | | | |
| **Clase** | **Método** | **Escenario** | **Entradas** | **Resultado** |
| Cola | insertar() | setupScenary1 | Un entero = 9 | Se ha agregado un entero a la cola en la primera posición exitosamente. |
| Cola | Insertar() | setupScenary2 | Un entero = 3 | Se agrega un entero a la cola y ha quedado en la ultima posición. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la prueba:** Verificar que el método extraer() (dequeue) funcione correctamenete, cuando:   1. Hay elementos en la cola. 2. La cola se encuentre vacia. | | | | |
| **Clase** | **Método** | **Escenario** | **Entradas** | **Resultado** |
| Cola | extraer() | setupScenary2 |  | Se elimina el primer elemento en entrar y se retorna. |
| Cola | extraer() | setupScenary1 |  | Se retorna un null, ya que no hay elementos para extraer. |

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Clase** | **Escenario** |
| setupScenary1 | HashMap | Una estructura de datos lineal HashMap con un Integer como key y un String como valor ( HashMap<Integer,String> ) inicializada, sin ningún valor añadido. |
| setupScenary2 | HashMap | Una estructura de datos lineal HashMap con un Integer como key y un String como valor ( HashMap<Integer,String> ) inicializada, con 3 datos agregados:  Dato 1:  Key=14 , Value=”FA”  Dato 2:  Key=6 , Value=”MI”  Dato 3:  Key=2 , Value=”DO” |

**Diseño de las pruebas.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la prueba:** Verificar el correcto funcionamiento de método put() de la clase HashMap, cuando:   1. El HashMap se encuentre vacío. | | | | |
| **Clase** | Método | Escenario | **Entradas** | **Resultado** |
| HashMap | put() | setupScenary1 | Un nuevo valor con Key=28, Value=”SOL” | Se ha agregado el valor “SOL” con key=28 al HashMap, por lo que ahora el HashMap contiene esta Key. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la prueba:** Verificar el correcto funcionamiento de método contains() de la clase HashMap, cuando:   1. El HashMap contenga la llave indicada. 2. El HashMap no contenga la llave indicada. | | | | |
| **Clase** | Método | Escenario | **Entradas** | **Resultado** |
| HashMap | contains() | setupScenary1 | Un nuevo valor a insertar con KeyInsertar=28, ValueInsertar=”SOL” | (true) Se ha encontrado el valor que contiene la key=28. |
| HashMap | contains() | setupScenary1 | KeyQueNoEsta=56 | (false) No se ha encontrado el valor que contiene la key=56. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la prueba:** Verificar el correcto funcionamiento de método get() de la clase HashMap, cuando:   1. El HashMap contenga la llave indicada. | | | | |
| **Clase** | Método | Escenario | **Entradas** | **Resultado** |
| HashMap | get() | setupScenary2 | Un nuevo valor a insertar con KeyInsertar=196, ValueInsertar=”CALI” | Se ha retornado el valor insertado previamente. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objetivo de la prueba:** Verificar el correcto funcionamiento de método delete() de la clase HashMap, cuando:   1. El HashMap contenga el elemento indicado. | | | | |
| **Clase** | Método | Escenario | **Entradas** | **Resultado** |
| HashMap | delete() | setupScenary2 | Un nuevo valor a insertar con KeyInsertar=196, ValueInsertar=”CALI” (Para lograr ver que se agregó y después ya no estará) | El elemento que se encontraba en el hashMap después de usar el método delete ya no se encontró. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo: Poder sacar el dinero que el cliente posee en su cuenta de banco | | | | |
| Clase | Método | Escenario | Entradas | resultado |
| Client | retirarDinero() | Setup1() | -cantidad de dinero que desea sacar. | -valor de verdad sobre si se pudo sacar o no. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo: Poder agregar dinero al cliente en su cuenta de banco | | | | |
| Clase | Método | Escenario | Entradas | resultado |
| Client | ConsignarDinero() | Setup2() | -cantidad de dinero que desea sacar. | -se le agrega la suma de dinero que el cliente indique. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo: Poder abonar un dinero que el usuario debe por el uso de la tarjeta | | | | |
| Clase | Método | Escenario | Entradas | resultado |
| Client | PagarTarjeta() | Setup3() | -el dinero que quiere abonar a la deuda  -si es efectivo o si es por cuenta | -se le descuenta la deuda a la tarjeta  -se le descuenta la deuda a la tarjeta y se le reduce el monto de la cuenta de banco |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo: dar una copia del cliente antes de las operaciones realizadas | | | | |
| Clase | Método | Escenario | Entradas | resultado |
| Client | darCopia() | Setup4() |  | -una copia del cliente sin operaciones para deshacer sus cambios |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo: Poder agregar objetos al Stack | | | | |
| Clase | Método | Escenario | Entradas | resultado |
| Stack | push() | Setup1() | -El elemento o Nodo | -se agrega el nodo a la estructura |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo:Saber el tamaño del Stack o número de nodos | | | | |
| Clase | Método | Escenario | Entradas | resultado |
| Stack | size() | Setup2() |  | -la cantidad de nodos que hay en la estructura |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo: Borrar el elemento de arriba del stack | | | | |
| Clase | Método | Escenario | Entradas | resultado |
| Stack | Pop() | Setup3() |  | -se elimina el top de el stack |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo: Conocer si el arreglo tiene Nodos o esta vacio | | | | |
| Clase | Método | Escenario | Entradas | resultado |
| Stack | IsEmpty() | Setup4() |  | -un valor verdadero cuando esta vacío  -un valor negativo cuando esta lleno |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Objetivo: Conocer el elemento tope del stack | | | | |
| Clase | Método | Escenario | Entradas | resultado |
| Stack | Top() | Setup5() |  | -Nos retorna el Nodo que se encuentra en el tope del stack |