|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Clase* | *Método* | *Escenario* | *Valores de entrada* | *Resultado* |
| IStack | Pop():T | [10, 20, 5, 40] | Este Método no Tiene Entradas | 40.  Se retorna el 40. Y la lista queda con los siguientes valores.  [10, 20, 5] |
| IStack | Last():T | [10, 20, 5, 40, 100, 541] | Este Método no tiene entradas. | 541.  El valor esperado es 541. La lista no cambia. |
| IStack | removeLastElement():void | [10, 20, 5, 40, 100, 541] | Este Método no tiene entradas. | Se elimina 541 y la lista debe quedar de la siguiente manera.  [10, 20, 5, 40, 100] |
| IStack | Push(T Value):void | [10, 20, 5, 40, 100, 541] | 99 | Se agrega el 99 en la última posición y la lista queda de la siguiente forma.  [10, 20, 5, 40, 100, 541, 99] |
| IStack | IsEmpty():boolean | 1. [10, 20, 5, 40, 100, 541] 2. [] | Este método no tiene entradas. | * Retorna false, ya que el tamaño de la lista es mayor a 0 * Retorna True, ya que el tamaño de la lista es 0 y eso implica que está vacía |
| IStack | Size():int | 1. [10, 20, 5, 40, 100, 541] 2. [] | Este método no tiene entradas. | 1. 6 2. 0 |
| IQueue | Offer(T e):void | 1. [Hola, Joe, 10, 10.22] 2. [asd, qwe, ew] | 1. Hi 2. Kwe | 1. Se agrega “Hi” a la cola la lista queda de la siguiente manera: [Hi, Hola, Joe, 10, 10.22,] 2. En la segunda entrada queda de la siguiente manera: [kwe, asd, qwe, ew] |
| IQueue | removeFirstElement():void | 1. [Hola, Joe, 10, 10.22] 2. [asd, qwe, ew] | Este método no tiene entradas. | 1. Se elimina el “Hola” de la lista, queda de la siguiente manera: [Joe, 10, 10.22] 2. Se elimina el “asd” de la lista, queda de la siguiente manera [qwe, ew] |
| IQueue | Poll():T | [asd, qwe, ew] | Este método no tiene entradas | Asd  Retorna “asd” y la lista queda de la siguiente manera [qwe, ew] |
| IQueue | Firts():T | [lqew, duew, qweq] | Este método no tiene entradas | Lque  Este método te retonar el primer elemento de una lista |
| IQueue | Size():T | 1. [10, 20, 5, 40, 100, 541] 2. [] | Este método no tiene entradas | 1. u |
| IQueue | isEmpty():boolean | 1. [10, 20, 5, 40, 100, 541] 2. [] | Este método no tiene entradas | 1. Retorna false ya que el tamaño es mayor que 0 2. Retorna true ya que el tamaño es igual a 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Server | fillSaerver():void | Se instancia un objeto tipo Server | Un número entero que representa la cantidad de jugadores a ingresar en el servidor | Retorna True si el número de jugadores en el servidor coinciden con el esperado |
| Fornite | fillLeves():void | Se instancia un objeto tipo Fornite | Un objeto tipo player, el cúal representa el jugador a insertar actual | Retorna True si el numero de jugadores en los respectivos niveles coinciden con el esperado |
| Fornite | typePing():void | Se instancia un objeto tipo Fornite | Un objeto tipo player, el cúal representa el jugador a insertar actual. Y 3 números enteros que son el tipo de ping: malo, medio. bueno | True si el número de jugadores en los respetivos pings coinciden |
| IStack | IsEmpty():boolean | 1. [10, 20, 5, 40, 100, 541] 2. [] | Este método no tiene entradas. | * Retorna false, ya que el tamaño de la lista es mayor a 0 * Retorna True, ya que el tamaño de la lista es 0 y eso implica que está vacía |
| IStack | Size():int | 1. [10, 20, 5, 40, 100, 541] 2. [] | Este método no tiene entradas. | 1. 6 2. 0 |
| IQueue | Offer(T e):void | 1. [Hola, Joe, 10, 10.22] 2. [asd, qwe, ew] | 1. Hi 2. Kwe | 1. Se agrega “Hi” a la cola la lista queda de la siguiente manera: [Hi, Hola, Joe, 10, 10.22,] 2. En la segunda entrada queda de la siguiente manera: [kwe, asd, qwe, ew] |