**UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA**

**FACULTAD DE INGENIERIA DE PRODUCCION Y SERVICIOS**

**ESCUELA PROFESIONAL DE CIENCIAS DE LA COMPUTACIÓN**

Un dibujo de una cara feliz

Descripción generada automáticamente con confianza baja

**LABORATORIO 10 – Templates – Sobrecarga de Operadores.**

**DOCENTE:**

Enzo Edir Velásquez Lobatón

**ALUMNO:**

Owen Haziel Roque Sosa.

**FECHA:**

22/05/2022

**Arequipa – Perú**

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

1. Imprimir Lista
2. Clase Lista - Destructor
3. Clase Nodo
4. Clase Lista
5. Defina una lista enlazada que permita insertar elementos al final de todos los elementos que ya se hayan ingresado. Por el momento no es necesario preservar un orden, simplemente los elementos nuevos deben de ingresar como el último elemento.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente



1. Con la implementación de la lista enlazada anterior, desarrolle una función que permita ingresar los elementos al inicio de todos los demás elementos. Tendrá que modificar el comportamiento del puntero que tiene referencia al primer elemento para que sea redireccionado al nuevo elemento por ingresar.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente



1. Desarrolle una función que permita ingresar elementos en el medio de dos elementos de la lista enlazada, como se muestra en la siguiente imagen. Solicite que se ingrese una posición válida dentro de la lista y permita que el valor ingresado se pueda anexar en esa posición.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

1. Elabore una función que permita eliminar el último elemento de una lista enlazada. (Evite copiar los elementos en una nueva lista para completar la eliminación del elemento).

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Imagen de la pantalla de un celular de un mensaje en letras negras

Descripción generada automáticamente con confianza baja

1. Desarrolle una función que permita eliminar el primer elemento de una lista sin perder referencia de los demás elementos que ya se encuentran almacenados en la estructura. (Evite copiar los elementos en una nueva lista para completar la eliminación de los elementos)

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Imagen de la pantalla de un celular de un mensaje en letras negras

Descripción generada automáticamente con confianza baja

1. Dado una posición válida dentro de la lista, permita al usuario eliminar un elemento de cualquier posición sin perder referencia de los demás elementos.

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Imagen de la pantalla de un celular de un mensaje en letras negras

Descripción generada automáticamente con confianza baja

1. Desarrolle un algoritmo de ordenamiento que permita ordenar los elementos de forma ascendente y descendente (8. ) de acuerdo a la elección del usuario. Se debe poder simular el ingreso de 10 mil elementos de forma aleatoria y ordenarlos en el menor tiempo posible ( < 2 seg).

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Imagen de la pantalla de un celular de un mensaje en letras negras

Descripción generada automáticamente con confianza baja

1. Algoritmo de ordenamiento de forma descendente:

Texto

Descripción generada automáticamente

Imagen de la pantalla de un celular de un mensaje en letras negras

Descripción generada automáticamente con confianza baja

* Método para imprimir lista a través de parámetro (Puntero a objeto Lista<T>)

Texto

Descripción generada automáticamente