

**UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS**

**2022 - 01**

**CURSO:** [**Algoritmos y Estructuras de Datos**](https://aulavirtual.upc.edu.pe/webapps/blackboard/execute/courseMain?course_id=_317376_1)

**DOCENTE:**  **Abraham Sopla Maslucán**

**SECCIÓN:** [**CC32**](https://aulavirtual.upc.edu.pe/webapps/blackboard/execute/courseMain?course_id=_317376_1)

**INTEGRANTES**:

| Alumno | Código | Carrera |
| --- | --- | --- |
| Pilco Chiuyare, André Dario | **u202110764** | Ciencias de la Computación |
| Roque Ponce, Christian Alonso | **u20201a917** | Ciencias de la Computación |
| Velasquez Chambi , Ruben Genaro | **u202117342** | Ingeniería de Software |

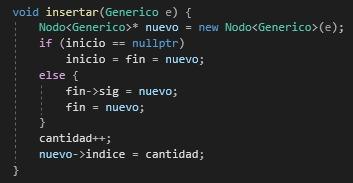
**Introducción**

El presente trabajo se refiere a la implementación de las Estructura de Datos en C++ en un programa tipo “Gestor de Datos” (inspirado en la app “Microsoft To Do”). El uso de estas estructuras nos permite ordenar secuencialmente datos que, gracias al uso de Templates, recibe y almacena datos de tipo Class en Nodos. Para desarrollar el programa trabajamos con el paradigma de POO en C++ y codificamos una interfaz en consola para verificar la funcionalidad de nuestro “Gestor de Datos”.

**Explicación del Caso de Estudio:**

Empleamos dos estructuras de datos: listas simples y listas dobles. Una lista simple únicamente tiene un apuntador al siguiente nodo, mientras que una lista doble tiene un apuntador al siguiente nodo y también uno al anterior. Estas estructuras se utilizan según el problema que se desee resolver. Por ejemplo, una lista simple la emplearemos para guardar datos que se requieran en secuencia y una lista doble se implementa cuando se debe tener la posibilidad de hacer secuencia hacia adelante o atrás a través de los nodos que conforman la lista.

**Big O Del Algoritmo Principal:**

****

Nodo<Generico>\* nuevo = new Nodo<Generico>(e); —> // 1

if( inicio== nullptr)

inicio=fin= nuevo; —>2

else {

fin→sig=nuevo; —>//1

fin=nuevo; →//1

}

cantidad++; —>// 2;

nuevo→índice=cantidad; —-->//1

Tiempo detallado: 2+2+1=4 (constastante)  
Tiempo Asintótico: O(1)

**Descripción de las tareas realizadas:**

| **Nombres y apellidos:** | **Tarea realizada:** | **Tiempo** |
| --- | --- | --- |
| Christian Alonso Roque Ponce | Clase Lista y Nodo | 2 horas |
| Christian Alonso Roque Ponce | Métodos Insertar e Imprimir | 2 horas |
| Christian Alonso Roque Ponce | Constructores de Entidades | 30 minutos |
| André Dario Pilco Chiuyare | Métodos de recursividad y eliminar | 30 minutos |
| André Dario Pilco Chiuyare | Interfaz y Listas de cada una de las Entidades | 3 horas |
| André Dario Pilco Chiuyare | Lectura y escritura de archivos | 2 horas |
| Ruben Genaro Velasquez Chambi | Implementar la clase lista doble | 1 hora |
| Ruben Genaro Velasquez Chambi | Análisis Big O | 20 min |