

---

## Proyecto 2 IPC2

---

202002912 – Vinicio Manfredo López Pérez

### Resumen

Este proyecto se basa en el uso de la programación para resolver problemas que acontecen en nuestro entorno haciendo uso de uno de sus pilares la programación orientada a objetos. En este proyecto se implementó la programación en el área de la de la logística de una empresa, un programa que consta de la lectura de un archivo XML que contiene datos de empresas que serán gestionadas por nuestro programa, dicho datos contienen el nombre de la empresa, sus puntos de atención y sus transacciones. Para cada punto de atención la empresa cuenta con distintos escritorios activos o inactivos que son controlados mediante una aplicación web que los gestiona de manera en que cuando entra un nuevo cliente al centro de atención a este se le asigna un numero de orden por el cual será atendido, dicho número sigue la lógica de una cola y la desactivación de los distintos escritorios de atención siguen una la lógica de una pila ya que Conforme hayan sido activados se desactivarán.

### Palabras clave

Algoritmo  
Nodo  
Apuntadores  
TDA  
POO

### Abstract

*This project is based on the use of programming to solve problems that occur in our environment using one of its pillars object-oriented programming. In this project the programming was implemented in the area of logistics of a company, a program that consists of reading an XML file containing data from companies that will be managed by our program, this data contains the name of the company, its points of care and transactions. For each point of attention the company has different active or inactive desks that are controlled by a web application that manages them in such a way that when a new client enters the attention center he is assigned an order number by which he will be attended, this number follows the logic of a queue and the deactivation of the different attention desks follows the logic of a stack, because when they have been activated they are deactivated.*

*As soon as they have been activated they will be deactivated.*

### Keywords

Algorithm  
Node  
Pointers  
TDA  
POO.

## Introducción

El proyecto trata acerca de los datos de tipo abstractos a través de una mezcla de listas simplemente enlazadas, las mismas se utilizaron para poder almacenar los datos de cada empresa dentro de estas listas, dichos datos fueron previamente leídos por medio de una librería que contiene Python para la gestión de este tipo de archivos.

El programa permite leer dos tipos de archivo, uno para darle una configuración de empresas al programa y otro para poder ingresar datos de manera rápida con una configuración inicial al momento de ser cargado. Además, se pueden ingresar datos de manera manual y todo esto se gestiona de manera rápida y eficaz haciendo cálculos de tiempos de espera de los clientes y del promedio del tiempo de atención para los escritorios de servicio activos.

## Desarrollo del Proyecto

### a. Funcionamiento de la lectura de Archivo XML

El funcionamiento de la lectura del archivo XML comienza a gestionarse desde un menú donde se debe de introducir manualmente el path del archivo que se desea leer, dicho archivo debe de estar en la raíz del proyecto para su correcto funcionamiento o en dado caso ingresar la path completa del mismo. Luego de ingresar el archivo correctamente se debe de crear previamente las listas enlazadas con sus clases nodo y sus constructores que almacenaran los datos de cada empresa.

### b. Creación de las listas

Para la creación de las listas enlazadas solamente se crearon listas simplemente enlazadas para optimizar el rendimiento de las y

que no se ocupara más memoria en apuntadores. Tomando en cuenta lo anterior se crearon listas enlazadas, en donde cada una tenia su respectiva función por dar un ejemplo hay una lista que gestiona el comportamiento de la cola dentro de un punto de atención con clientes que tienen sus respectivos tiempos de espera y atención al igual que se tiene control de los escritorios de servicio activos e inactivos que dan soporte y almacenan información de sus respectivos clientes, además de tener el comportamiento de un pila al activar y desactivar escritorios de servicio, el primero en ser activado es el último en ser desactivado.

### c. Método insertar

Este método funciona por medio de dos apuntadores, una lista enlazada se compone de una serie de nodos enlazados mediante apuntadores. Cada nodo es una combinación de dos partes: un tipo de dato (entero, real, doble, carácter o tipo predefinido) y un enlace (apuntador) al siguiente nodo. Se hizo uso del apuntador siguiente que representa hacia donde apunta el nodo actual, cada vez que se agrega un nuevo dato el apuntador siguiente apuntara vacío cuando este esté en la primera posición al insertar un nuevo nodo el apuntador siguiente apuntara a actual y el nodo siguiente tomara el puesto de actual por lo que de nuevo siguiente de actual apuntara a vacío.

Es algo compleja la interpretación de esta pero una vez se entiende se logra abstraer los datos de manera correcta para el gestionamiento de los mismos sea mejor ya que estos no gastan memoria mientras no están en uso.

#### **d. Método imprimir**

Todos los datos se deben de ver en consola para esto se creó un método que por medio de un ciclo imprime todos los datos de manera ordenada de cada empresa, el ciclo tiene una condición de que el dato de referencia debe de tener un valor, cuando este sea vacío se rompe el ciclo y se imprime la cadena que se desea imprimir, los datos que se imprimen corresponden a los datos de transacciones, punto de atención, empresas registradas, cliente en el punto de atención, escritorios de servicio entre otras utilidades. Se hace uso de una segunda cadena para poder almacenar la matriz que se logra por medio de un ciclo similar al general, pero este solo almacena los atributos de cada lista específica. Mientras la validación de que el siguiente no este vacío se asignara el valor del nodo actual al siguiente de actual y cuando esta validación no se cumpla se termina el ciclo.

#### **e. Método recalcular**

Este método recalcula los tiempos de espera de cada cliente que este en la cola, además se le da el mismo uso para recalcular el tiempo de atención que se le dará a cada cliente en un escritorio de servicio al salir un cliente dejando un espacio vacío para el siguiente en la cola.

#### **e. Método tiempos**

Estos son varios métodos que se utilizan para poder encontrar el tiempo mínimo, máximo y promedio de espera y de atención para un punto de atención o para un escritorio de servicio.

#### **g. Definiciones**

### **TDA y POO**

En el mundo de la programación existen diversos lenguajes de programación que se han ido creando con el paso del tiempo y que se han perfeccionado debido a las necesidades de los programadores de la época a la que pertenecen. Los primeros lenguajes de programación eran de tipo lineal, ya que un programa se recorría desde un punto marcado como Inicio hasta llegar a un punto Fin. Con el tiempo se fueron creando nuevos lenguajes y en nuestros días los más utilizados son los llamados “orientados a objetos”.

Los lenguajes de programación orientados a objetos (POO) tienen la característica de que no son lenguajes lineales, sino que se forman de diversas funciones, las cuales son llamadas en el orden en que el programa mismo las pide o el usuario determina. Para entender mejor cómo funcionan los lenguajes orientados a objetos, vamos a introducir un concepto fundamental en las Estructuras de Datos denominado Abstracción de Datos y que es parte importante de estos Lenguajes y de la manera en que funciona la mayoría del software comercial de nuestros días.

#### **Listas enlazadas**

En ciencias de la computación, una lista enlazada es una de las estructuras de datos fundamentales, y puede ser usada para implementar otras estructuras de datos. Consiste en una secuencia de nodos, en los que se guardan campos de datos arbitrarios y una o dos referencias, enlaces o punteros al nodo anterior o posterior. El principal beneficio de las listas enlazadas respecto a los vectores convencionales es que el orden de los elementos enlazados puede ser diferente al orden de almacenamiento en la memoria o el disco,

permitiendo que el orden de recorrido de la lista sea diferente al de almacenamiento.

Una lista enlazada es un tipo de dato autorreferenciado porque contienen un puntero o enlace (en inglés *link*, del mismo significado) a otro dato del mismo tipo. Las listas enlazadas permiten inserciones y eliminación de nodos en cualquier punto de la lista en tiempo constante (suponiendo que dicho punto está previamente identificado o localizado), pero no permiten un acceso aleatorio. Existen diferentes tipos de listas enlazadas: listas enlazadas simples, listas doblemente enlazadas, listas enlazadas circulares y listas enlazadas doblemente circulares.

## Conclusiones

El manejo de documentos por medio de listas enlazadas es efectivo y de gran ayuda, ya que estos tienen un mayor rendimiento respecto a las listas nativas que usa cualquier lenguaje de programación u programa informático.

El uso de POO nos hace más sencillo trabajar con listas ya que esta posee un constructor con los atributos que tienen los datos que se van a almacenar.

## Referencias bibliográficas

Roldán Blay, C. (2021). Definición y operaciones con listas en Python.

