

---

## PROYECTO 2- SISTEMA DE ATENCION AL CLIENTE

---

202111718 – Walter Javier Santizo Mazariegos

### Resumen

Para la realización del proyecto No. 2 se realizó una aplicación de servicio al cliente mediante la cual se atiende y se brinda información a la empresa sobre las diversas características de su organización. las funciones del programa permiten analizar dos archivos con formato XML, los cuales cargan al sistema los datos de varias empresas y las configuraciones iniciales que permiten realizar pruebas sobre las empresas previamente cargadas y configuradas. La función principal del programa es la de modelar y simular el servicio de cualquier empresa con sus características que cuenta con varios puntos de servicio, así como con escritorios de atención, la aplicación se encarga de la estimación del tiempo de espera y de servicio, así como la de mostrar gráficamente el estado del punto de servicio.

Para la solución del problema se hizo uso de TDA, usando listas simples, listas doblemente enlazadas, que pretendían simular el proceso mediante el cual una persona espera servicio y recibe este.

Para mostrar gráficamente el estado del punto de atención se se hizo uso del software Graphviz, la cual muestra los escritorios activos, los clientes que están siendo atendidos, así como los clientes pendientes de atención.

### Palabras clave

Estructuras de datos, cola, pila, memoria dinámica, listas, estados, Python

### Abstract

For the realization of project No. 2, a customer service application was made through which the company is attended and provided with information about the various characteristics of its organization. The functions of the program allow the analysis of two files with XML format, which load the data of several companies into the system and the initial configurations that allow tests to be carried out on the companies previously loaded and configured. The main function of the program is to model and simulate the service of any company with its characteristics that has several service points, as well as customer service desks, the application is responsible for estimating waiting and service times, as well such as graphically displaying the status of the service point.

To solve the problem, TDA was used, using simple lists, doubly linked lists, which were intended to simulate the process by which a person expects service and receives it.

To graphically display the status of the service point, the Graphviz software was used, which shows the active desks, the clients that are being attended, as well as the clients pending attention

### Keywords

*Data structures, queue, stack, dynamic memory, lists, states, Python.*

## Introducción

Para poder llegar a la resolución del proyecto del tipo del cual se desarrolló. Para poder realizar el proyecto 2 de la clase de Introducción a la Programación y Computación 2 fue necesario la utilización de un lenguaje de programación el cual sea capaz de realizar las tareas que son solicitadas dentro del proyecto. Para esta ocasión se utilizó el lenguaje de programación python, uno de los lenguajes más completos en cuanto a la memoria dinámica que se pueden encontrar, por medio de este lenguaje se pudo acceder y utilizar las herramientas que este nos brinda, como la utilización de programación Orientada a objetos y lectores de archivos xml o arboles binarios.

El proyecto tiene como enfoque la implementación de estructuras de datos TDA's utilizando listas enlazadas y doblemente enlazadas, para tomar una serie de datos correspondiente de las empresas.

El algoritmo que se ha implementado se ha hecho con lista de listas usando estructura de datos e implementando condicionales y cíclicas.

## Desarrollo del tema

### Estructuras cíclicas

**Estructuras cíclicas** Estas estructuras también llamadas bucles, se utilizan para realizar o ejecutar operaciones o acciones repetitivas, las cuales tienen un fin o limite

Las estructuras cíclicas se utilizan para ejecutar fragmentos de código un número limitado de veces. Existen tres estructuras cíclicas generales, las cuales son:

- Desde Hasta (For)
- Hacer Mientras (While)

Las estructuras tienen el mismo objetivo, ejecutar un fragmento de código un número limitado de veces, su principal diferencia se encuentra en la forma en la cual limitan el número de ocasiones que se ejecutará el código.

### Estructuras condicionales

**Estructuras condicionales** Las estructuras condicionales existen en todos los lenguajes de programación y nos dejan realizar algo según una condición.

La estructura if permite que un programa ejecute unas instrucciones cuando se cumplan una condición.

La estructura de control if else permite que un programa ejecute unas instrucciones cuando se cumple una condición y otras instrucciones cuando no se cumple esa condición.

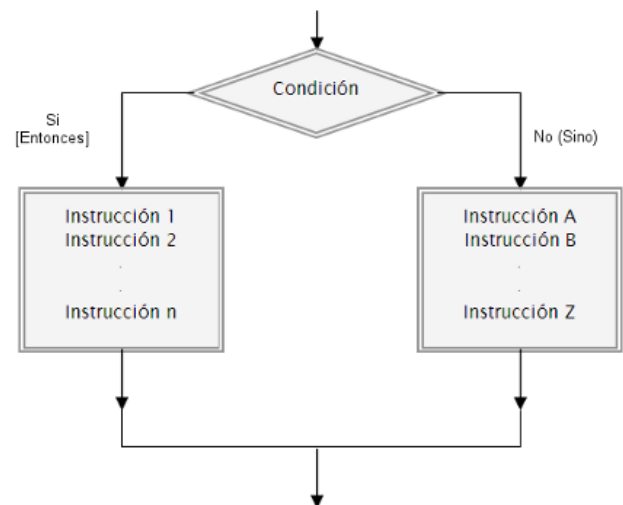


Figura 1. Algoritmo  
Fuente: Elaboración propia

### Archivos XML

XML consiste en un lenguaje de marcado que tiene la finalidad de definir una sintaxis para la codificación de documentos, que tanto los usuarios como las propias máquinas en sí puedan ser capaces de leer. Lo hace mediante la utilización de una serie de etiquetas que definen la estructura que posee el documento en cuestión, además de cómo debe ser transportado y almacenado. Se

parece mucho al lenguaje de marcado de hipertexto (HTML), usado esencialmente para la codificación de las páginas web, y que utiliza un conjunto de símbolos de marcado predefinidos que describen el formato que posee el contenido de una página web.

### Memoria Dinámica

Estas estructuras se caracterizan porque no sólo pueden variar su contenido durante la ejecución del programa sino que además pueden variar su tamaño. Debido a esto, las variables que son de este tipo se denominan estructuras de datos dinámicas

Las estructuras dinámicas lineales son: las listas simplemente enlazadas, las listas circulares, las listas doblemente enlazadas, las pilas y las colas.

En una asignación dinámica de memoria se hace referencia al hecho de crear variables o reservar espacio en memoria para estas variables en tiempo de ejecución del programa, así como liberar el espacio reservado para dichas variables anónimas, cuando ya no son necesarias, también durante el tiempo de ejecución.

Lista	Elementos
Simplemente enlazado	Mismo Tipo
Circular	El ultimo Nodo apunta al primero
Doblemente Enlazada	Nodo con doble acceso
Pilas	Mismo tipo de elemento en la cima
Colas	Elemento final es el inicial

Fuente: elaboración propia

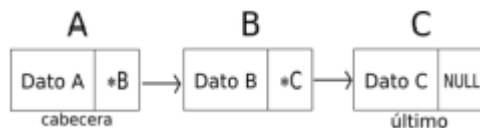


Figura 2: Lista Simple

Fuente: elaboración propia

### Listas

Las listas son una colección de elementos homogéneos (del mismo tipo: Tipo Elemento) con una relación LINEAL

establecida entre ellos. Pueden estar ordenadas o no con respecto a algún valor de los elementos y se puede acceder a cualquier elemento de la lista. Operaciones

- Crear()
- Imprimir()
- Insertar()
- Eliminar()
- Buscar()

### Implementación

Representación secuencial: el orden físico coincide con el orden lógico de la lista. Por ejemplo para insertar o eliminar elementos podría exigirse que se desplazaran los elementos de modo que no existieran huecos.

### Conclusiones

- Se ha reconocido que existen varios métodos para escoger el camino más optimo utilizando programación orientada a objetos y el uso de memoria dinámica para ahorrar recursos en nuestro sistema.
- En problemas reales pueden presentarse problemas donde necesitemos almacenar cierto conjunto de datos y es buena práctica utilizar memoria dinámica para ir guardando cada uno de estos datos.
- Las listas simplemente enlazadas, así como las listas doblemente enlazadas son útiles para trabajar con PILAS y COLAS ya que manejan la misma lógica para agregar y borrar elementos.
- Las listas enlazadas son estructuras de datos fáciles de comprender y muy flexibles para realizar funciones que en las listas nativas no son posibles

## Referencias bibliográficas

- <http://tutorialms-dos.bligoo.com>, "Listas Enlazadas Simples, Circulares... Y Árboles Binarios", 2002.
- C. S. Severance, (2009). Python para informaticos. Explorando la informacion, version 2.7.2, Inc.
- C. Larman, (2003). Introducción al análisis y diseño orientado a objetos, Hall Prentice
- Estructuras de datos: listas enlazadas, pilas y colas. (s. f.). Estructuras de datos: listas enlazadas, pilas y colas. Recuperado 6 de marzo de 2022, de <https://calcifer.org/documentos/librognome/gliblistsqueues.html>
- Lewis, John; Loftus, William (2009). Java Software Solutions Foundations of Programming Design 6th ed. Pearson Education Inc.

## Diagrama de Clases

