

## Síntesis de Estructura de Datos y Algoritmos utilizados.

El proyecto tiene como objetivo diseñar e implementar, en Python, un sistema orientado a objetos que modele un cliente de correo electrónico, permitiendo la gestión de usuarios, carpetas, mensajes, filtros y operaciones típicas de un entorno de email.

A su vez, simula un sistema completo de gestión de correo electrónico que integra diversas estructuras de datos y algoritmos para garantizar la organización, eficiencia y escalabilidad del sistema.

### 1. Filtros automáticos.

Cuando llega un mensaje, si el sistema verifica el remitente. Si llegase a coincidir con una regla definida, el sistema lo guarda automáticamente en una subcarpeta.

Para esto se utilizan listas y diccionarios, donde cada filtro asocia una palabra con una carpeta destino.

Este método realiza una búsqueda lineal, por lo que su complejidad es  $O(n)$ , donde  $n$  es el número de filtros definidos.

### 2. Gestión de mensajes urgentes. (PRIORIDAD)

Para aquellos mensajes que son marcados con prioridad alta, se utilizó una lógica de enrutamiento directo hacia una subcarpeta denominada "Urgentes", la misma ubicada dentro de la carpeta recibidos. En lugar de mantener una estructura oculta en memoria (como un Heap) se optó por una estructura persistente en el árbol de directorios ya que esto mejora la experiencia del usuario teniendo a la inmediatez lo urgente. La eficiencia de inserción se mantiene optimizada ya que el sistema detecta la prioridad al recibir el mensaje ( $O(1)$ ) y lo inserta en la lista de carpeta correspondiente, la cual se mantiene ordenada cronológicamente.

### 3. Red de Servidores y recorrido BFS.

La red de servidores se presenta como un grafo, donde cada nodo es un servidor y las conexiones son los enlaces entre ellos.

Al momento de simular el envío de mensajes entre servidores se usa el algoritmo BFS, que recorre los nodos por niveles y encuentra el camino más corto entre dos puntos de la red.

Su complejidad es  $O(V + E)$ , donde  $V$  es el número de servidores y  $E$  el número de conexiones.

### 4. Jerarquía de Carpetas y recorrido DFS.

Las carpetas se organizan como un árbol, donde cada una puede contener subcarpetas. Los métodos que buscan o mueven mensajes usan recursividad, aplicando el algoritmo DFS para explorar todos los niveles de la jerarquía.

La complejidad de este recorrido es  $O(n)$ , siendo  $n$  el número total de carpetas o mensajes recorridos.

El sistema combina de manera coherente listas, diccionarios, grafos y árboles para resolver problemas de clasificación, priorización, búsqueda y comunicación.

Cada estructura y algoritmo fue seleccionado por su simplicidad conceptual y eficiencia práctica, integrándose en un modelo que refleja con precisión el funcionamiento de un entorno real de correo electrónico.