

Laboratorio 9 – Colas de Prioridad

Grupo 03 – SEC04

- a.** Una cola de prioridad en Python normalmente se implementa con un heap binario, almacenado en una lista. El heap mantiene el elemento con mejor prioridad en la raíz y permite reorganizar la estructura sin ordenar todo en cada operación.
- b.** Una cola FIFO atiende en orden de llegada, mientras que una cola de prioridad atiende según el valor asignado. El elemento más importante sale primero, incluso si llegó después de los demás.
- c.** Python ofrece el módulo `heapq`, que implementa funciones basadas en un min-heap para insertar, sacar el elemento principal y mantener la estructura de manera eficiente.
- d.** El uso de una cola de prioridad basada en heap evita ordenar manualmente listas grandes. Con un heap, insertar y extraer elementos importantes es más rápido y escalable cuando el número de datos crece.
- e.** Si dos elementos tienen la misma prioridad, el heap no garantiza estabilidad global. Normalmente queda primero el que ya estaba más arriba en la estructura, aunque no necesariamente el que entró antes.
- f.** En la vida real, las colas de prioridad se usan en muchos escenarios, como la planificación de tareas urgentes en un sistema operativo o la selección del nodo más prometedor en algoritmos de rutas como Dijkstra.
- g.** En un sistema médico con triage, la cola de prioridad sirve para atender primero a los pacientes más críticos, sin depender del orden de llegada. La urgencia define quién pasa antes.
- h.** Una lista simple obliga a buscar y eliminar al elemento prioritario recorriendo todos los datos, lo cual es costoso. Un heap permite hacer estas operaciones en menos tiempo, manteniendo la eficiencia incluso con muchos elementos.
- i.** En una cola de prioridad basada en heap, insertar y eliminar el elemento con mejor prioridad toma $O(\log n)$. Consultar el primer elemento cuesta $O(1)$, porque siempre está en la raíz del heap.