



Informe sobre la implementación y análisis de algoritmos de ordenamiento

**Algoritmos y Estructura de Datos**

Integrantes:

- Zapata Mariana Gabriela
- Weimer Valentin
- Kerbs Javier

2° Cuatrimestre, 2025

## 1. Introducción.

El primer ejercicio consistió en implementar y verificar el funcionamiento de una lista doblemente enlazada como tipo abstracto de datos (TAD). A partir de ello se midieron los tiempos de ejecución de tres operaciones en particular: copiar, len(para obtener la longitud) e invertir; usando el método `time.perf_counter()` para medirlo con precisión. Además se generó un gráfico con dichas medidas para comprobar y comparar los resultados obtenidos con el orden de complejidad teórica.

## 2. Desarrollo de la solución.

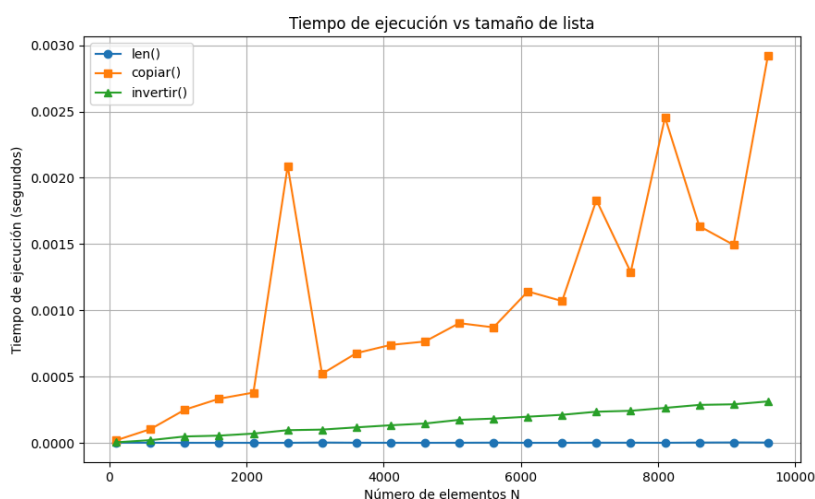
- Se generan listas de distintos tamaños  $n$
- Se realizaron mediciones de tiempo de ejecución para las operaciones **len()**, **copiar()** e **invertir()** en listas de tamaño  $N(100$  a  $10000)$ .
- Se genera una gráfica que presenta los promedios de ejecución entre las tres operaciones.
- Se comparan los resultados con el orden de complejidad.

## 3. Análisis de complejidad.

De manera teórica:

- **Copiar la lista**  $\rightarrow O(n)$
- **Longitud de la lista (len)**  $\rightarrow O(1)$
- **Invertir la lista**  $\rightarrow O(n)$ .

## 4. Resultados experimentales.



## 5. Conclusiones.

Al implementar una lista doble enlazada se comprobó que el comportamiento de las operaciones fue coherente con la teoría de la complejidad temporal. Las funciones invertir y copiar tienen complejidad  $O(n)$ , ya que recorren los nodos por medio de un ciclo while que se ejecuta hasta que el puntero llega a None, mientras que en invertir se intercambian los punteros siguiente y anterior de cada nodo, por lo que en ambos casos el tiempo de ejecución depende de la cantidad de elementos. En el caso de len, la complejidad es  $O(1)$  ya que devuelve el valor correspondiente al tamaño de la lista y no recorre la misma.