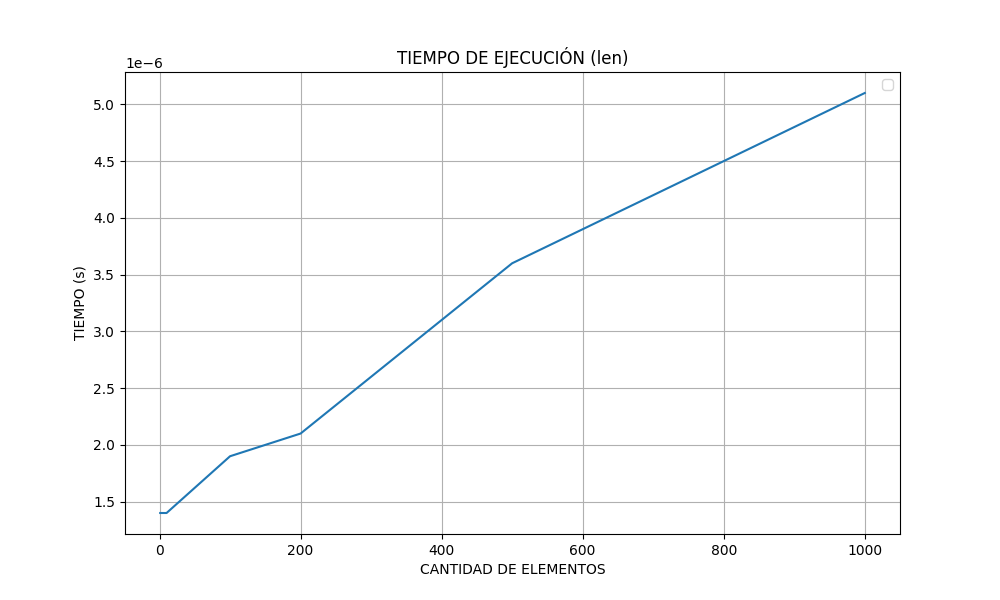
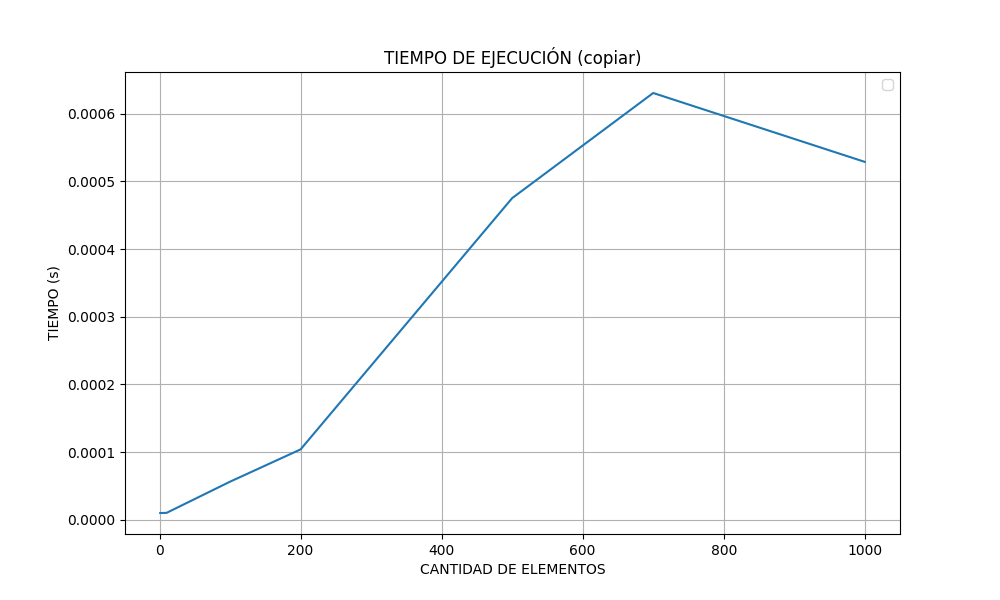
**GRÁFICA DEL MÉTODO: len**

Si realizamos un análisis a priori del código, podemos determinar que el orden de complejidad es , ya que no depende de los n elementos de la lista.



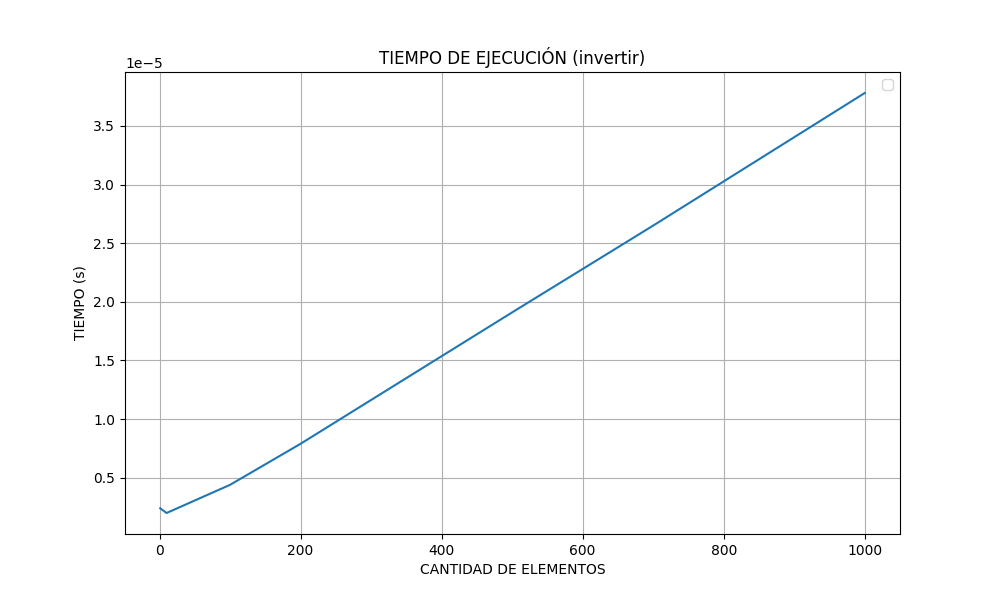
**GRÁFICA DEL MÉTODO: copiar**

Analizando el código, determinamos que es de complejidad . Esto es debido a que el método depende de los n valores dentro de la lista, ya que itera elemento a elemento para hacer la copia.



**GRÁFICA DEL MÉTODO: invertir**

Aplicando el mismo análisis, podemos decir que el orden de complejidad es de , ya que recorre todos los elementos para invertir el orden.

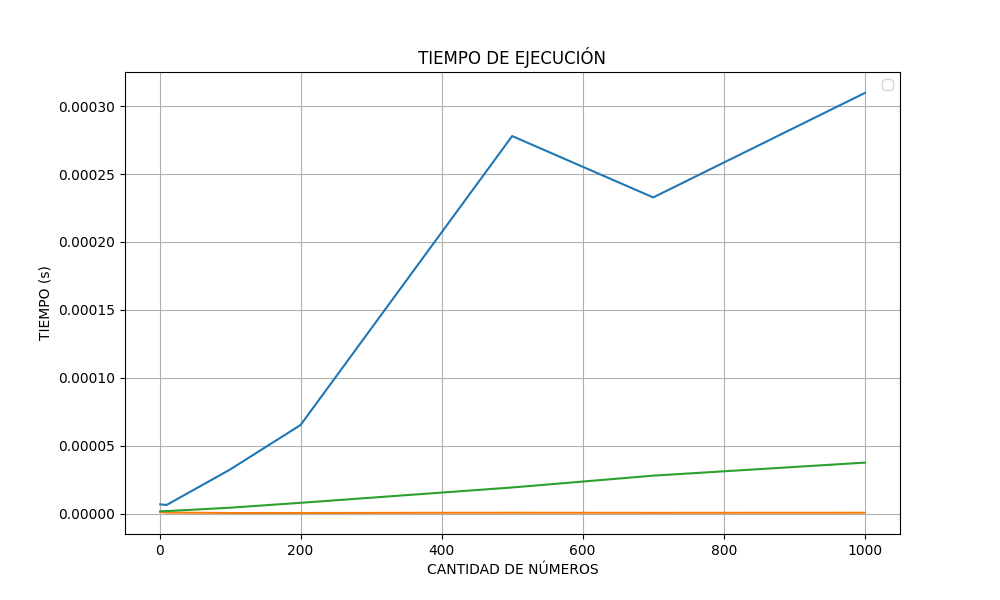


**GRÁFICA COMPARATIVA ENTRE LOS TRES TIPOS DE ORDENAMIENTOS**

Método copiar.

Método len.

Método invertir.



**INFORME**

En este problema se nos pedía implementar el TAD lista doblemente enlazada, la cual es una lista en donde cada nodo tiene un puntero al nodo anterior y otro al nodo siguiente. Esto permite recorrer la lista en ambos sentidos.

Implementamos la lista doblemente enlazada, definimos cada operación que se nos pedía y mediante el test provisto por la cátedra corroboramos su funcionamiento. Luego, modificamos un código base visto en clase para medir el tiempo de ejecución de los métodos len, copiar e invertir y graficarlos. En base a estas gráficas, logramos deducir el orden de complejidad de estos tres métodos mencionados.

Debido al tamaño de la muestra, se ven anomalías en la gráfica del método copiar, sin embargo, con un tamaño de muestra mayor, se podría apreciar de una mejor manera que el orden de complejidad del método copiar es .