**Texto

Descripción generada automáticamenteRepo:** [**https://github.com/CarEsteban/estructura-datos-2024/tree/main/HTD3**](https://github.com/CarEsteban/estructura-datos-2024/tree/main/HTD3)

**Profiler Austin:**

Se tuvieron diferentes inconvenientes al momento de usar el profiler Austin, se buscó documentación o vídeos de uso del profiler y no se encontró, por lo cual se usaron dos alternativas diferentes para sacar los resultados en segundos de cada sort.

Texto

Descripción generada automáticamente

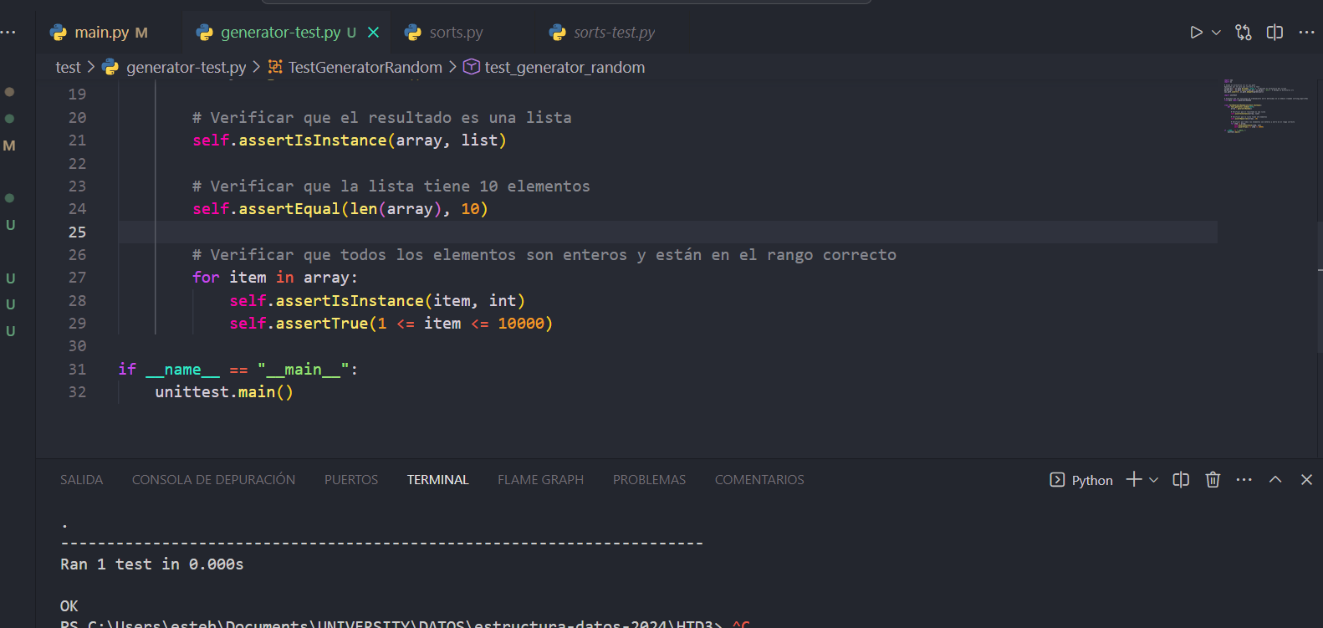
Usando cProfile, se logró ver de una forma general cuando tardaban todos los métodos usados, se usó el comando: > python -m cProfile main.py para poder corres el Profiler y dar un estimado del tiempo.

Texto

Descripción generada automáticamente

Al final, se terminó pidiendo a la IA que generase un código para poder contar los tiempos de cada función de una forma manual, usando la librería **timeit.**

**Tests**

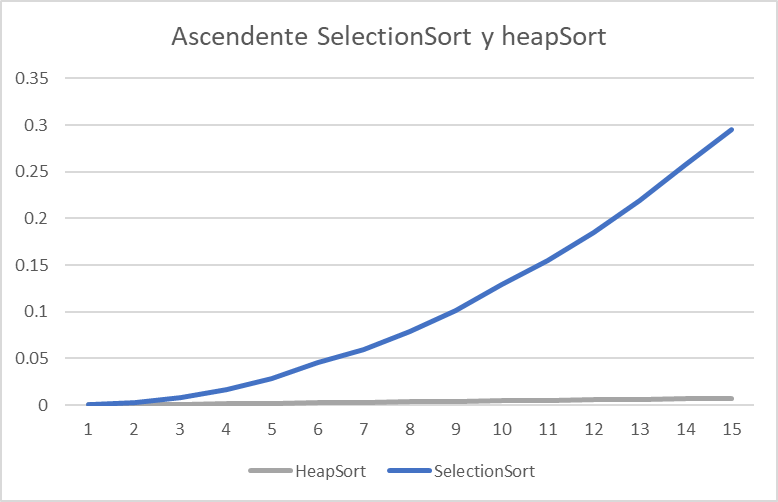


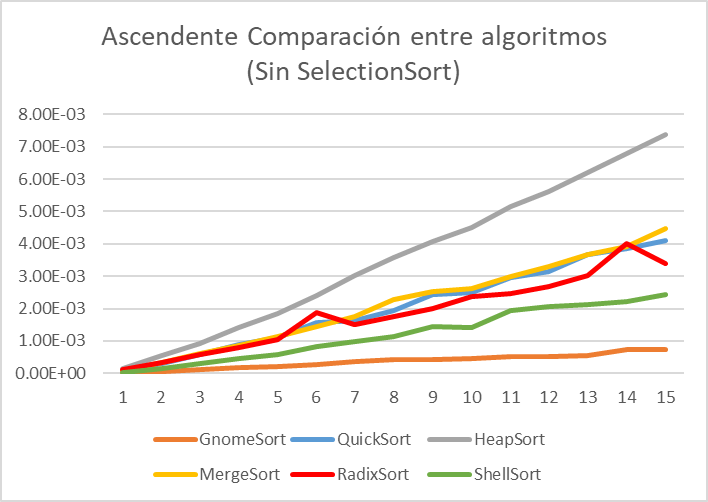
Funciona el generador de números random

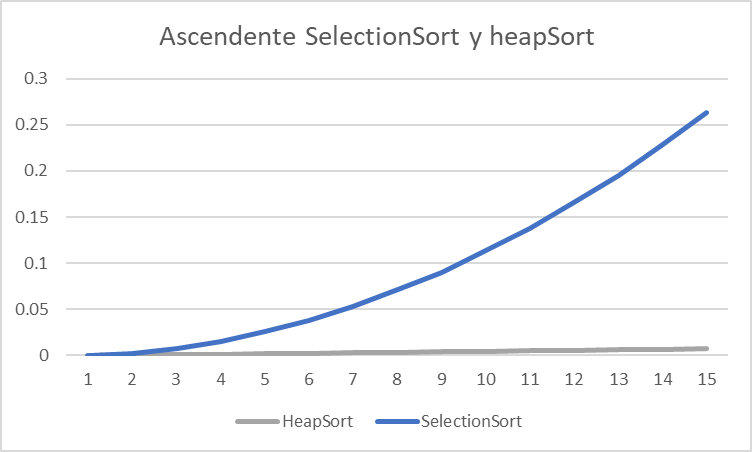
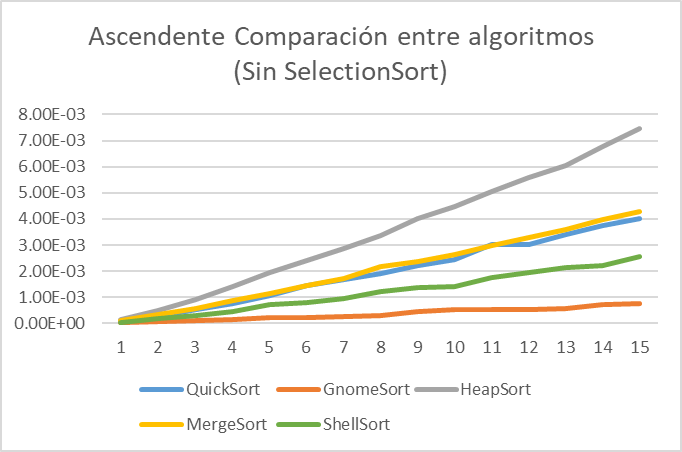
Texto

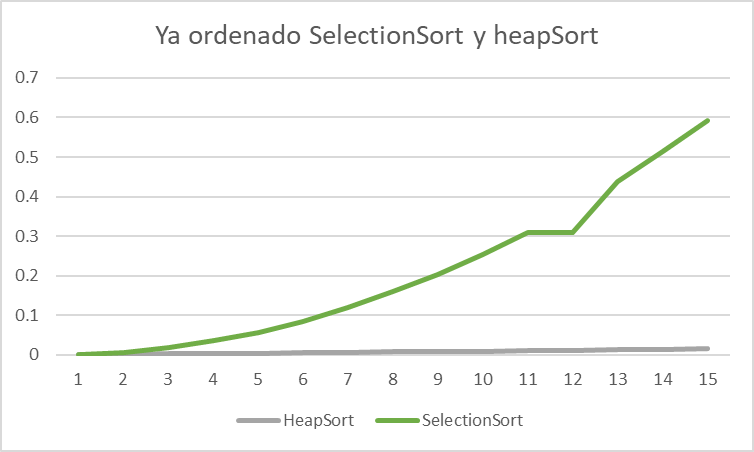
Descripción generada automáticamente

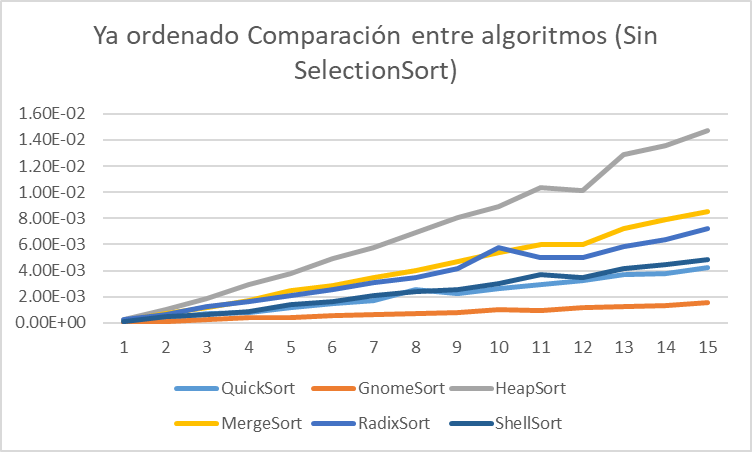
Funcionan los tests que validan el funcionamiento de cada sort.

**Gráficas comparando de forma ascendente cada sort**

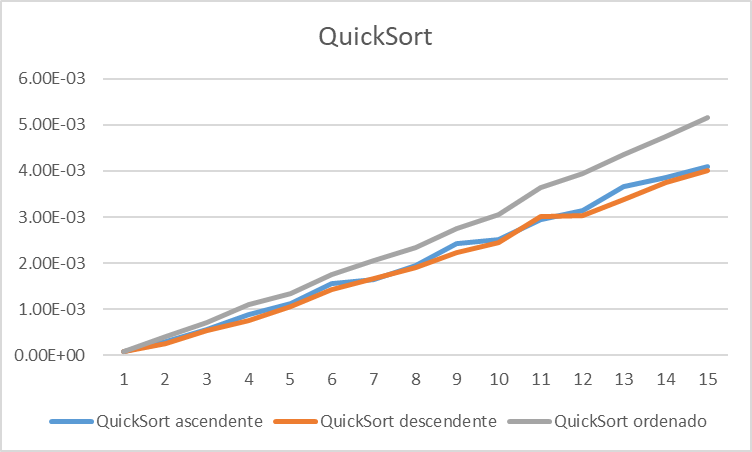
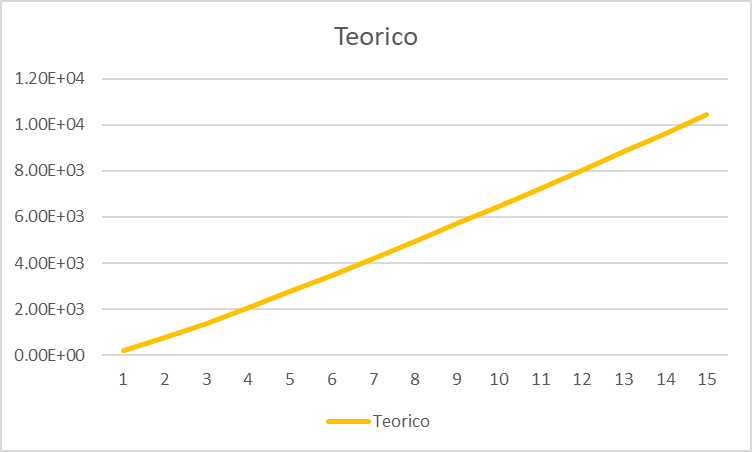
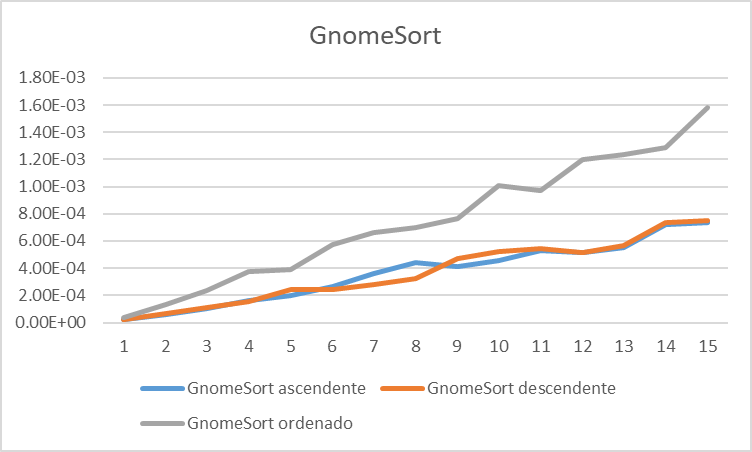


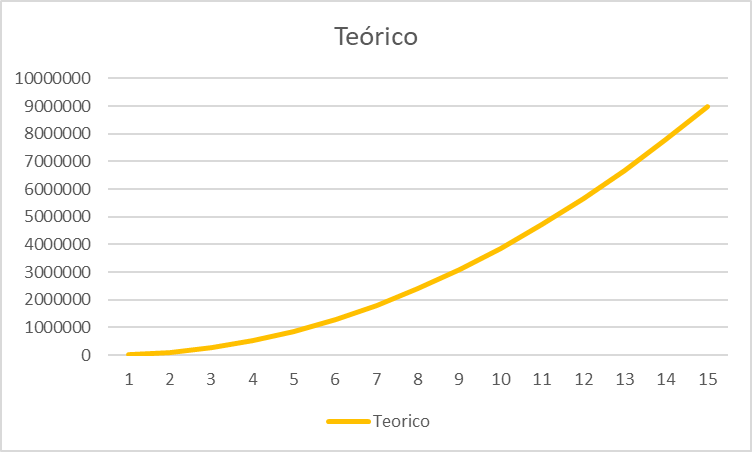
**Gráficas comparando de forma descendente cada sort**

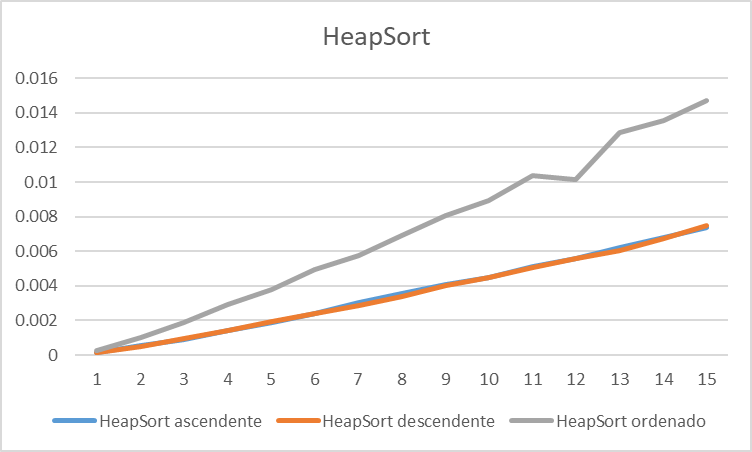
**Gráficas comparando de forma ordenada, ordenando cada sort**

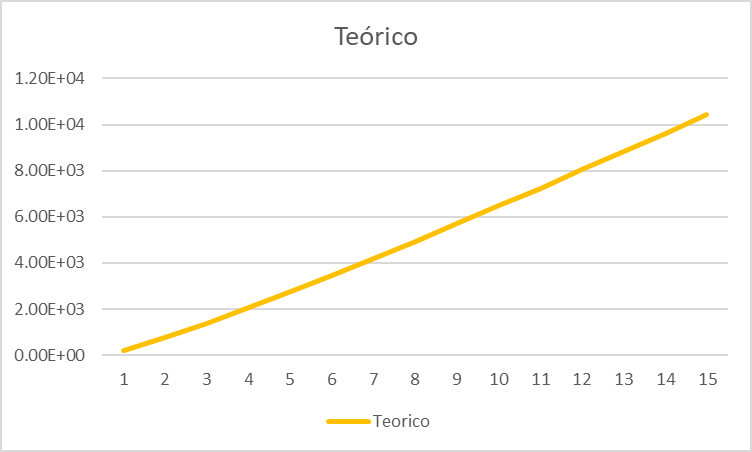
****

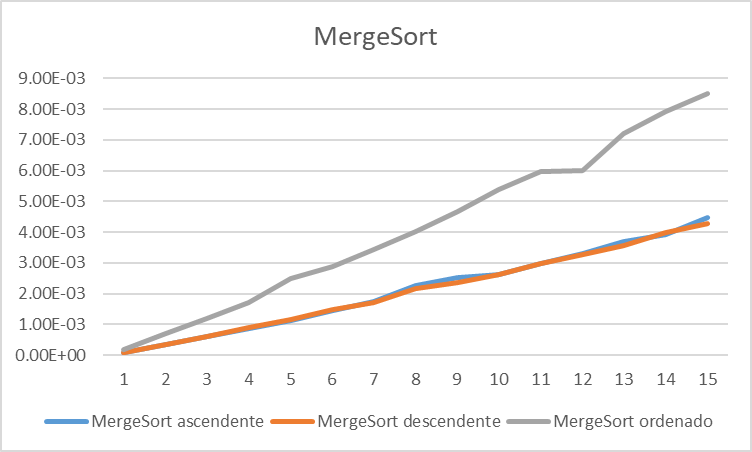
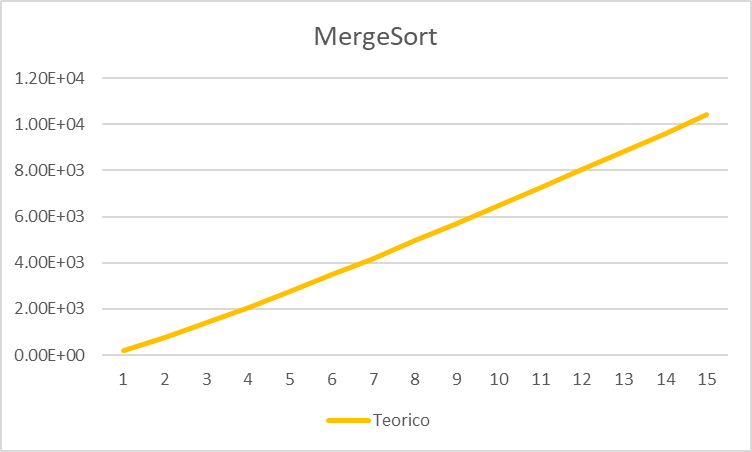
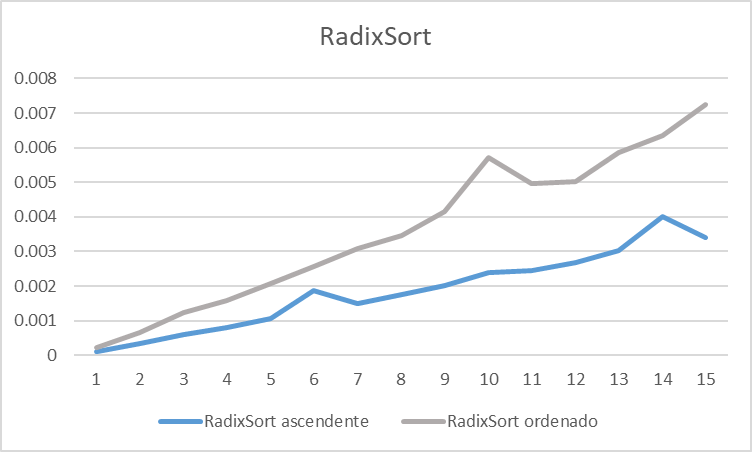
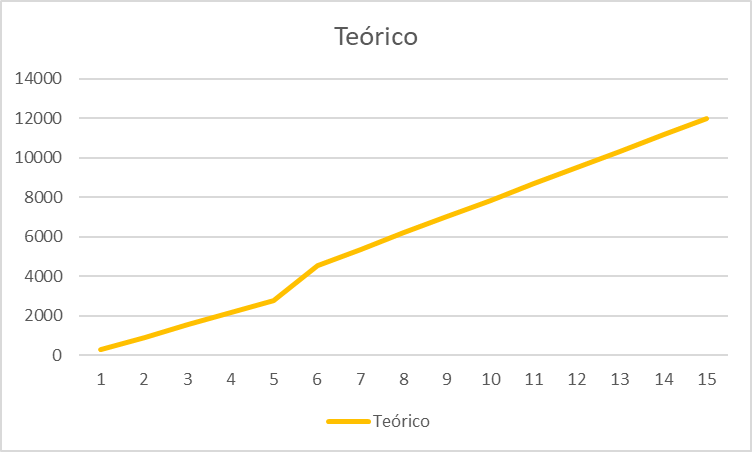
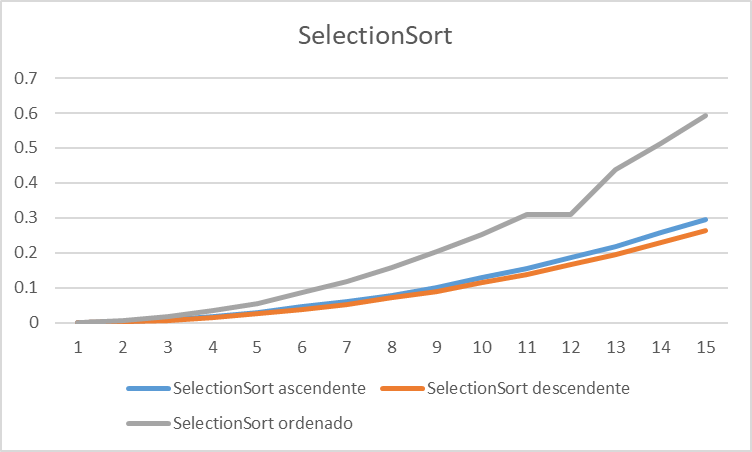
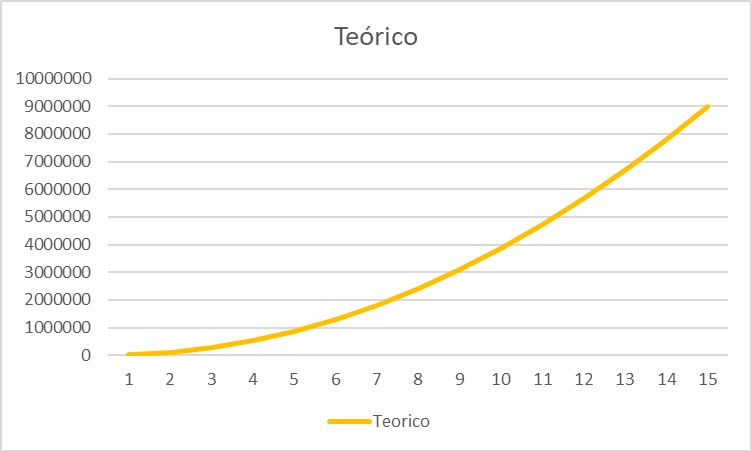
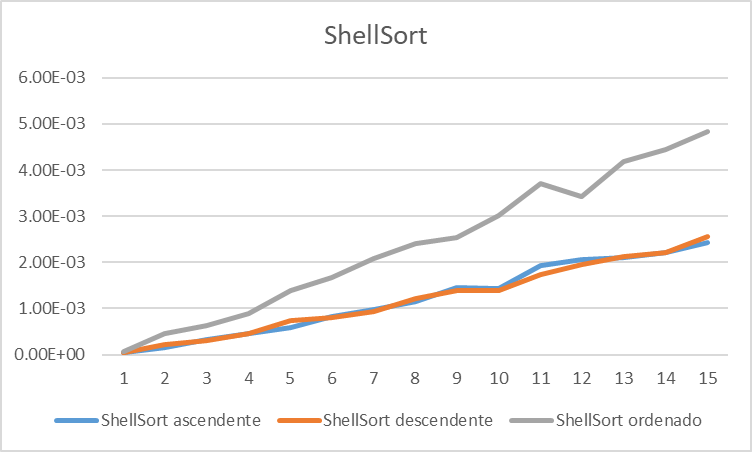
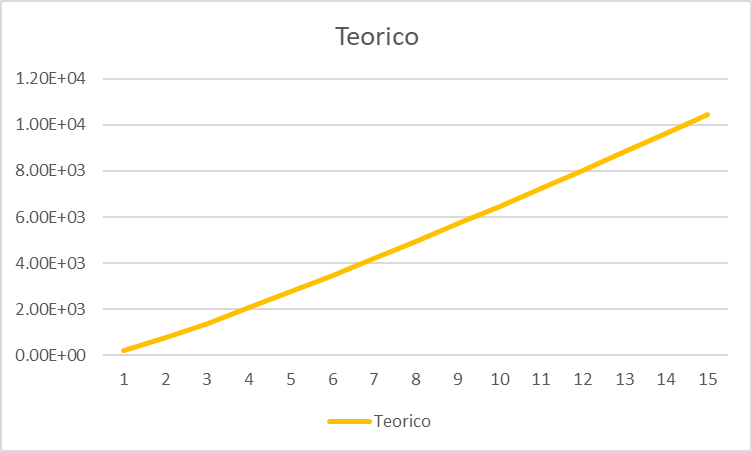
**Comparación entre ascendente, descendente, teórico y de forma ordenada cada sort:**

1. **QuickSort**
2. **GnomeSort**

****

1. **HeapSort**

****

1. **MergeSort**
2. **RadixSort**
3. **SelectionSort**
4. **ShellSort**