

Alejandro Torres Soto

Juan David Calderón

Cristian Botina

Santiago Castillo

## Algoritmo de ordenamiento distribuido \_ Informe

### **Introducción.**

El desarrollo de software distribuido se refiere a la creación de aplicaciones que se ejecutan en múltiples dispositivos o servidores interconectados. A diferencia del software monolítico, donde todo el código se ejecuta en una sola máquina, el software distribuido se divide en componentes que pueden residir en diferentes ubicaciones físicas.

### **Ventajas del software distribuido.**

- Escalabilidad
- Disponibilidad
- Eficiencia

### **El papel del middleware ICE.**

Es una capa de software que actúa como intermediario entre las aplicaciones y el sistema operativo o la red. Proporciona servicios como comunicación, seguridad, transacciones y persistencia.

### **Estrategia de distribución.**

- Nuestro proyecto cuenta con una pseudo arquitectura que puede llamarse como arquitectura de división de tareas o paralización de tareas. Puesto que, tenemos un máster que se encarga de dividir el trabajo entre los workers para que cada uno haga una parte mas pequeña. Al final, el máster terminara juntando el trabajo de todos los workers para que sea la respuesta a la solicitud que se le encargó.
- El proyecto esta desarrollado de forma que se pueda desplegar en varios equipos de computo donde uno va a ser el máster y puede haber tantos workers como sea necesario o posible.

### **Diseño de implementación con ICE.**

- Definición de interfaces Slice: Se creó una interfaz llamada workerI con un solo método. Dicha interfaz, se encarga principalmente de que todos los workers sean capaces de recibir las cadenas que deben ordenar.

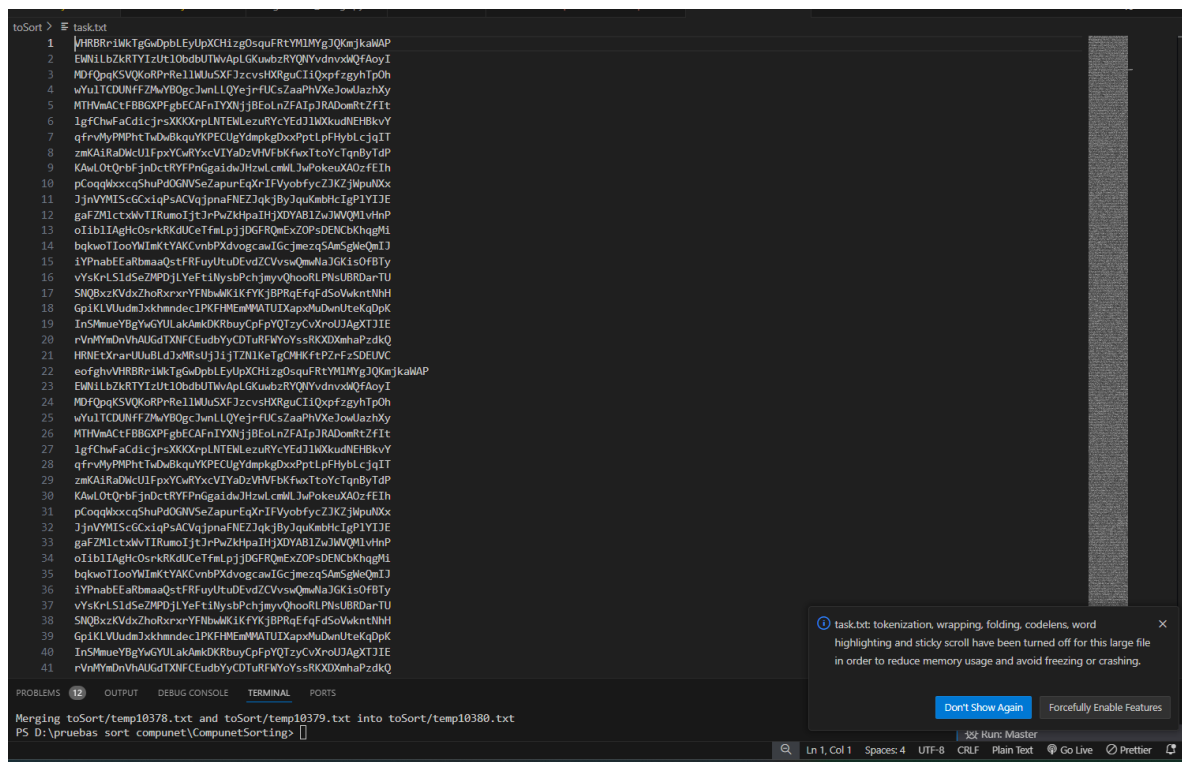
- Comunicación master-worker: El master usará los proxies de los workers para poder delegarles trabajo.

## Resultados de pruebas.

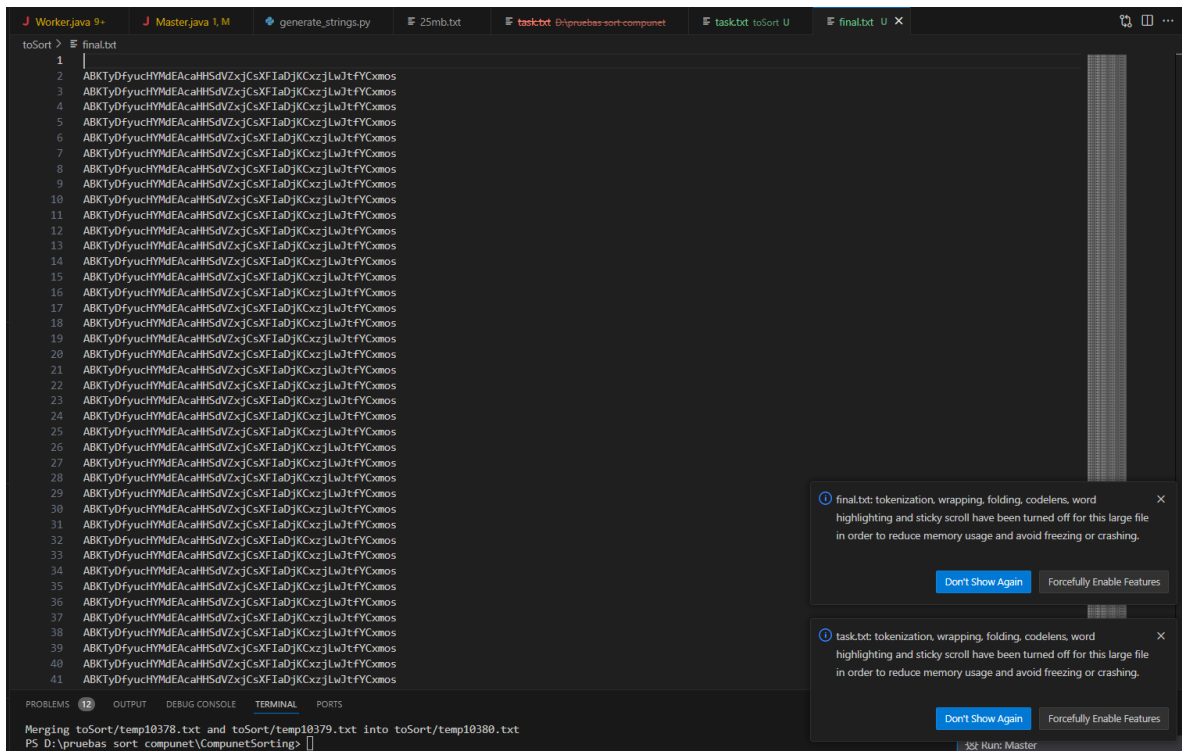
Realizamos nuestras pruebas ordenando archivos de texto plano de diferentes tamaños para poder verificar que al sistema es capaz de ordenar diferentes tamaños de archivos y para poder medir el tiempo que se toma para organizar cada uno de ellos.

Ordenar txt de 25 megas:

Antes del ordenamiento y Durante el ordenamiento:

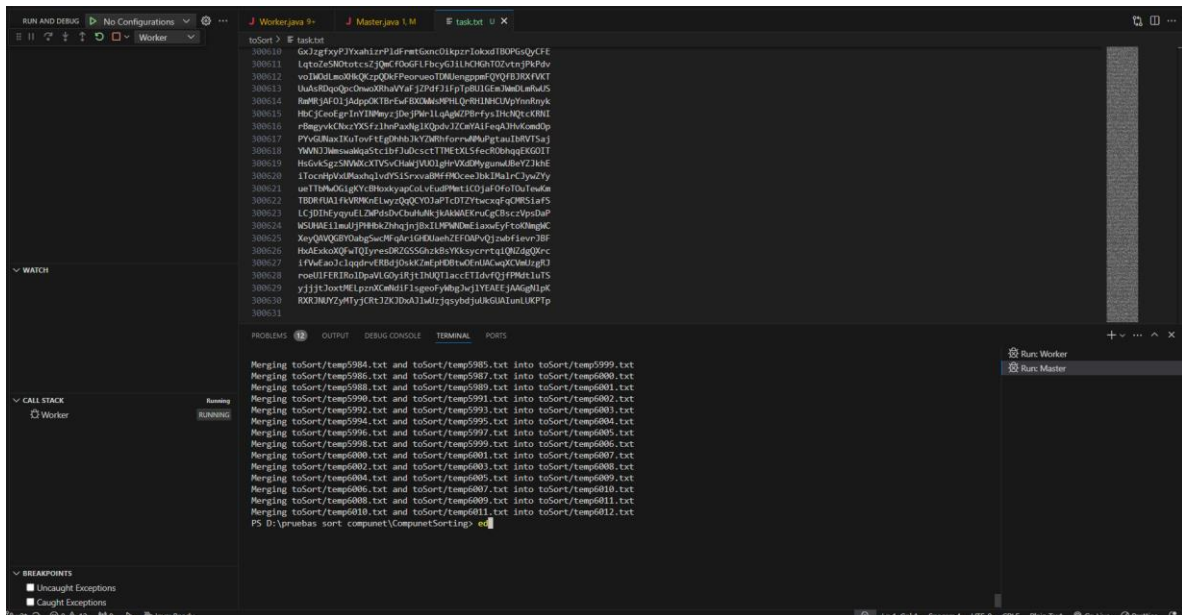


Después del ordenamiento:

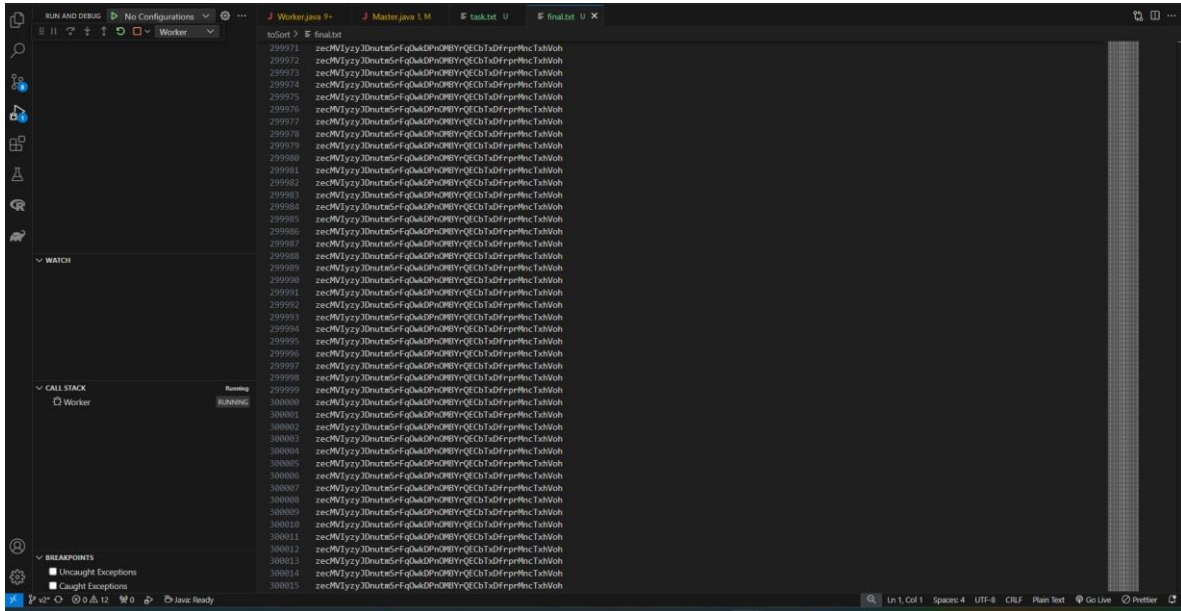


15 megas

Antes y durante el ordenamiento:

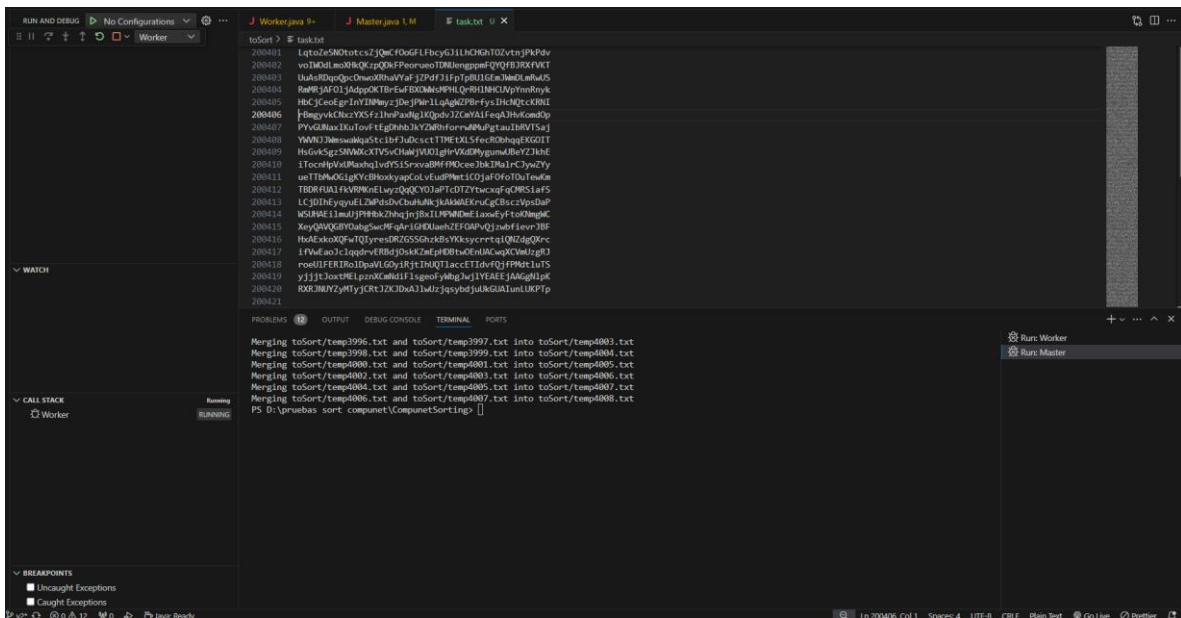


Después

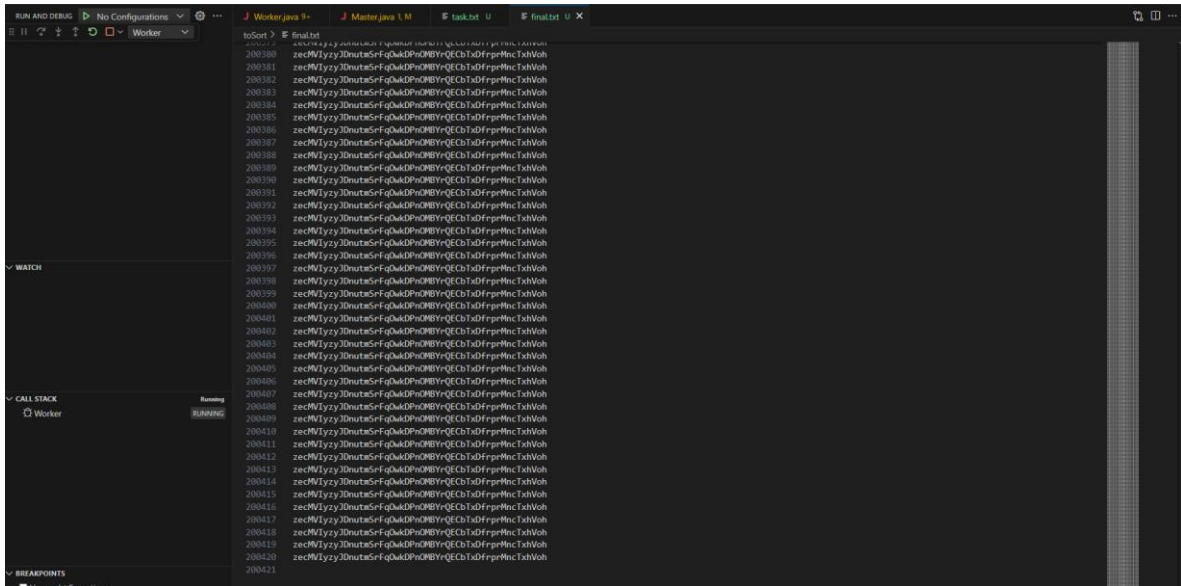


10 megas

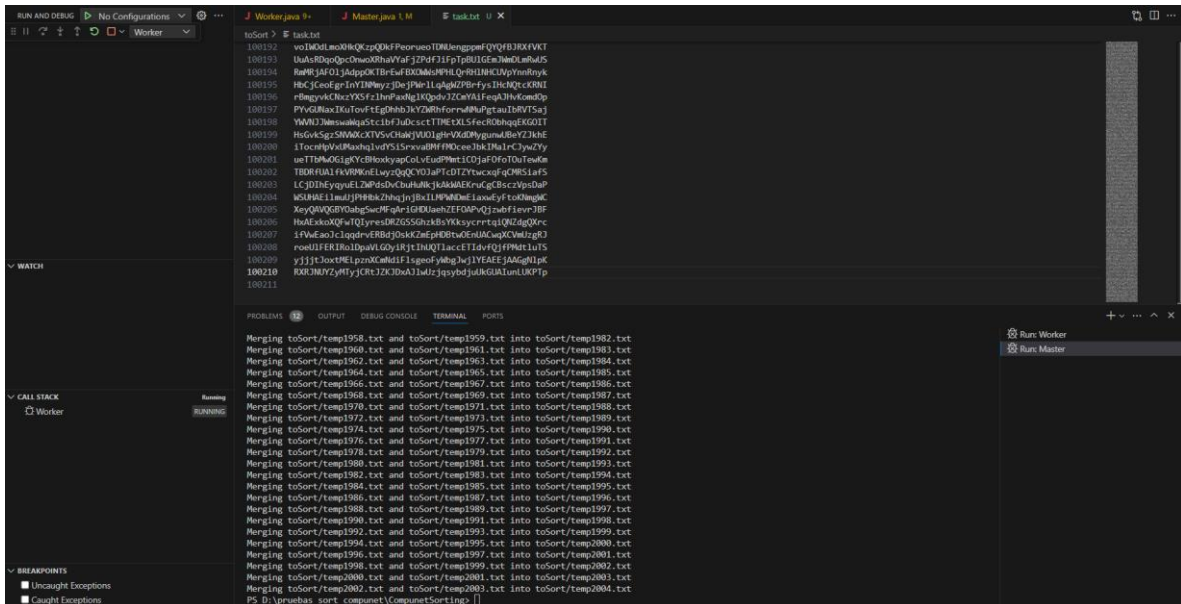
Antes y durante el ordenamiento



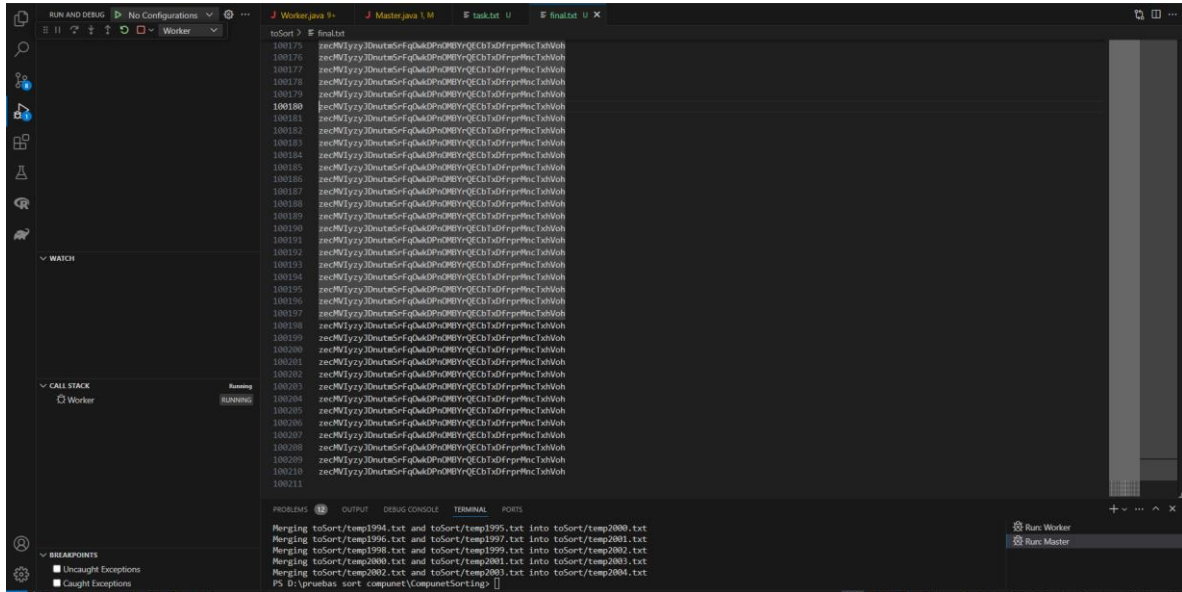
Despues del ordenamiento:



Con 5 megas:  
antes y durante el ordenamiento:



Despues del ordenamiento:



```
toSort > finalist
100175 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100176 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100177 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100178 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100179 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100180 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100181 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100182 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100183 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100184 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100185 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100186 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100187 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100188 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100189 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100190 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100191 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100192 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100193 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100194 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100195 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100196 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100197 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100198 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100199 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100200 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100201 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100202 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100203 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100204 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100205 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100206 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100207 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100208 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100209 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100210 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh
100211 zecWtIzy3Dnuta5rFqakDhQWBYQECtAdFrpfHcTshVoh

Merging toSort/temp1994.txt and toSort/temp1995.txt into toSort/temp2000.txt
Merging toSort/temp1996.txt and toSort/temp1997.txt into toSort/temp2001.txt
Merging toSort/temp1998.txt and toSort/temp1999.txt into toSort/temp2002.txt
Merging toSort/temp2000.txt and toSort/temp2001.txt into toSort/temp2003.txt
Merging toSort/temp2002.txt and toSort/temp2003.txt into toSort/temp2004.txt
PS D:\pruebas> sort -computer [ComputerSortApp] {}
```

## Análisis de rendimiento.

En general, el sistema ordenaba los documentos con una complejidad logarítmica. Puesto que, esta es la complejidad de los algoritmos de ordenamiento usados.

El principal causante de que a medida que aumente el tamaño de documento también aumenta el tiempo de ejecución, es por el master que ahora debe distribuir una tarea mucho mas grande.

## Conclusión.

El uso de ICE como middleware para la implementación distribuida nos proporcionó una base sólida para el proyecto. Con una estrategia adecuada, pruebas y un análisis continuo del rendimiento, pudimos construir un sistema que calificamos como eficiente y confiable.