

IES AUGUSTO GONZALES DE LINARES

CFGS DAM2º

Gestión de Procesos en

Paralelo y Secuencial

Programación De Servicios Y Procesos



Daniel Espinosa García

2023/2024

Índice

[**Demostración de la funcionalidad de la aplicación con capturas de pantalla.** 2](#_Toc147503478)

[**Comparación de tiempos entre la ejecución secuencial y paralela. Para medir tiempos de ejecución** 5](#_Toc147503479)

[**Conclusión** 7](#_Toc147503480)

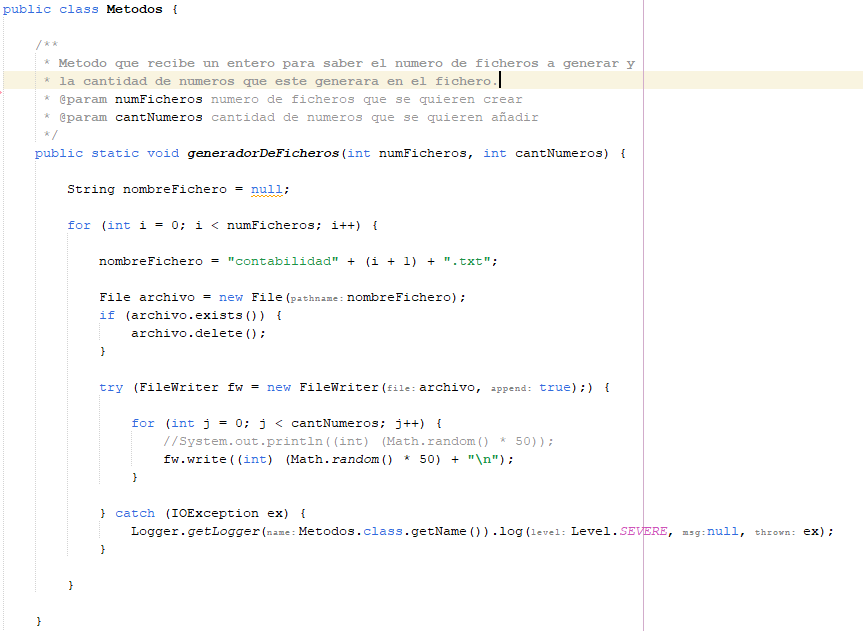
[**Bibliografía** 8](#_Toc147503481)

## **Demostración de la funcionalidad de la aplicación con capturas de pantalla.**

Para realizar la practica genere varios metodos que ejecutan cada uno de los requerimientos exigidos en la práctica y las llamadas de estos desde el Main.

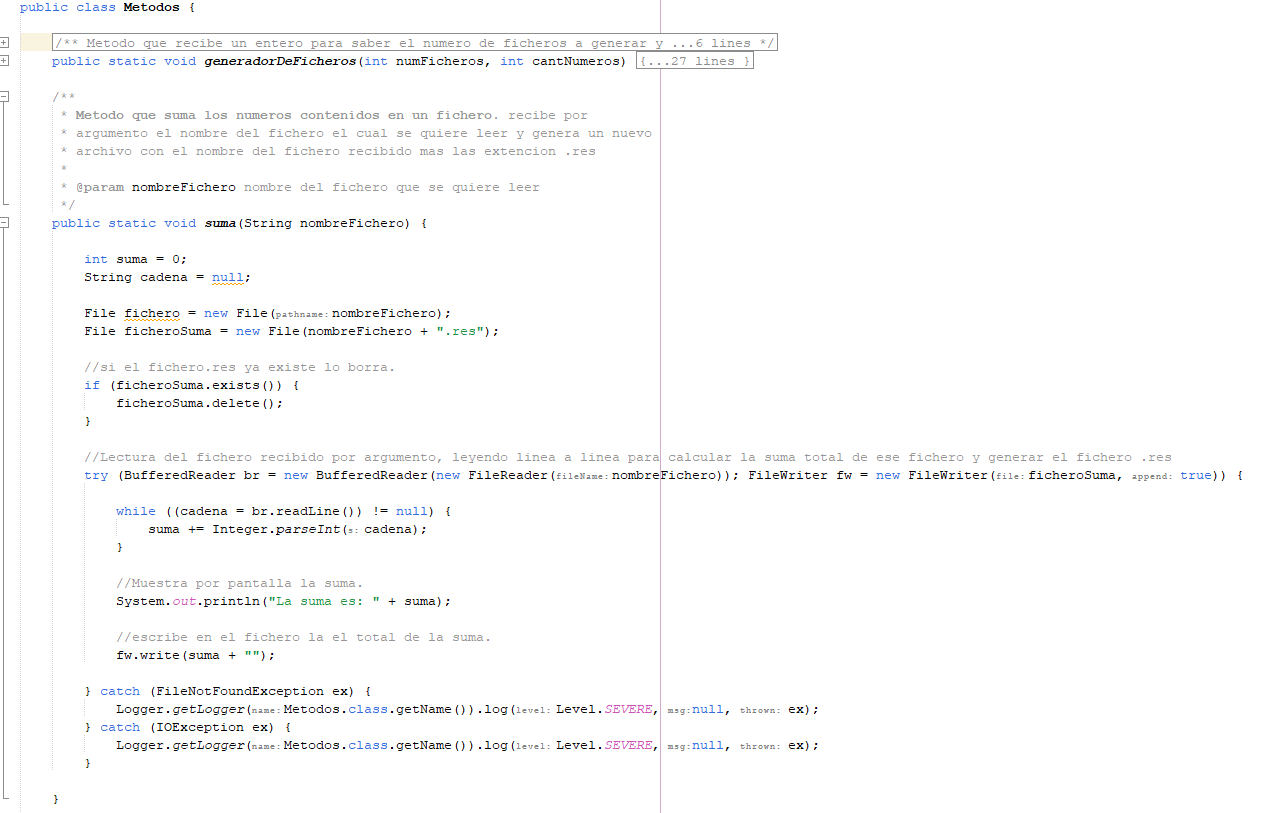
**Método Generar ficheros:**

Este método que recibe un entero para saber el número de ficheros a generar la cantidad de números que este generara en el fichero.



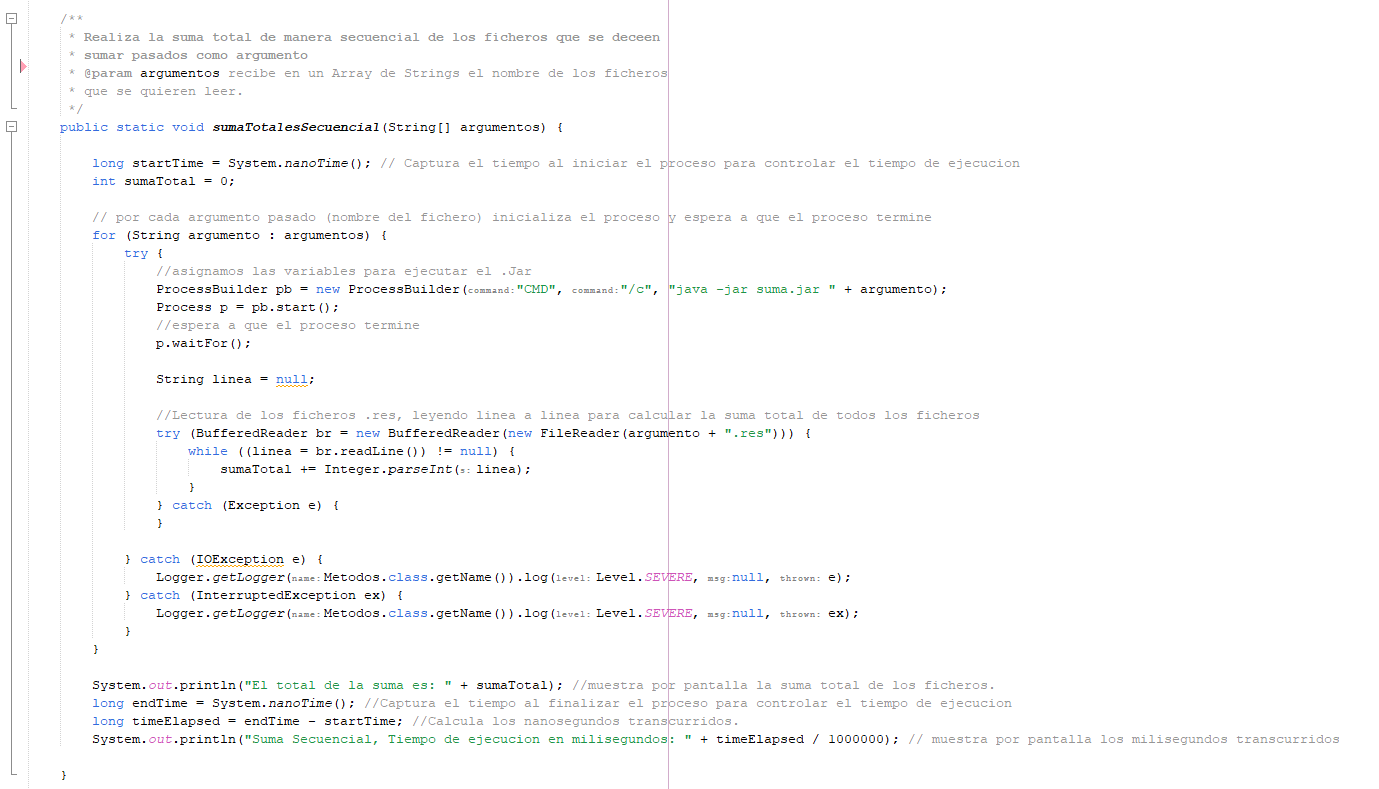
**Método suma:**

Método que suma los números contenidos en un fichero. recibe porargumento el nombre del fichero el cual se quiere leer y genera un nuevo archivo con el nombre del fichero recibido más la extensión .res



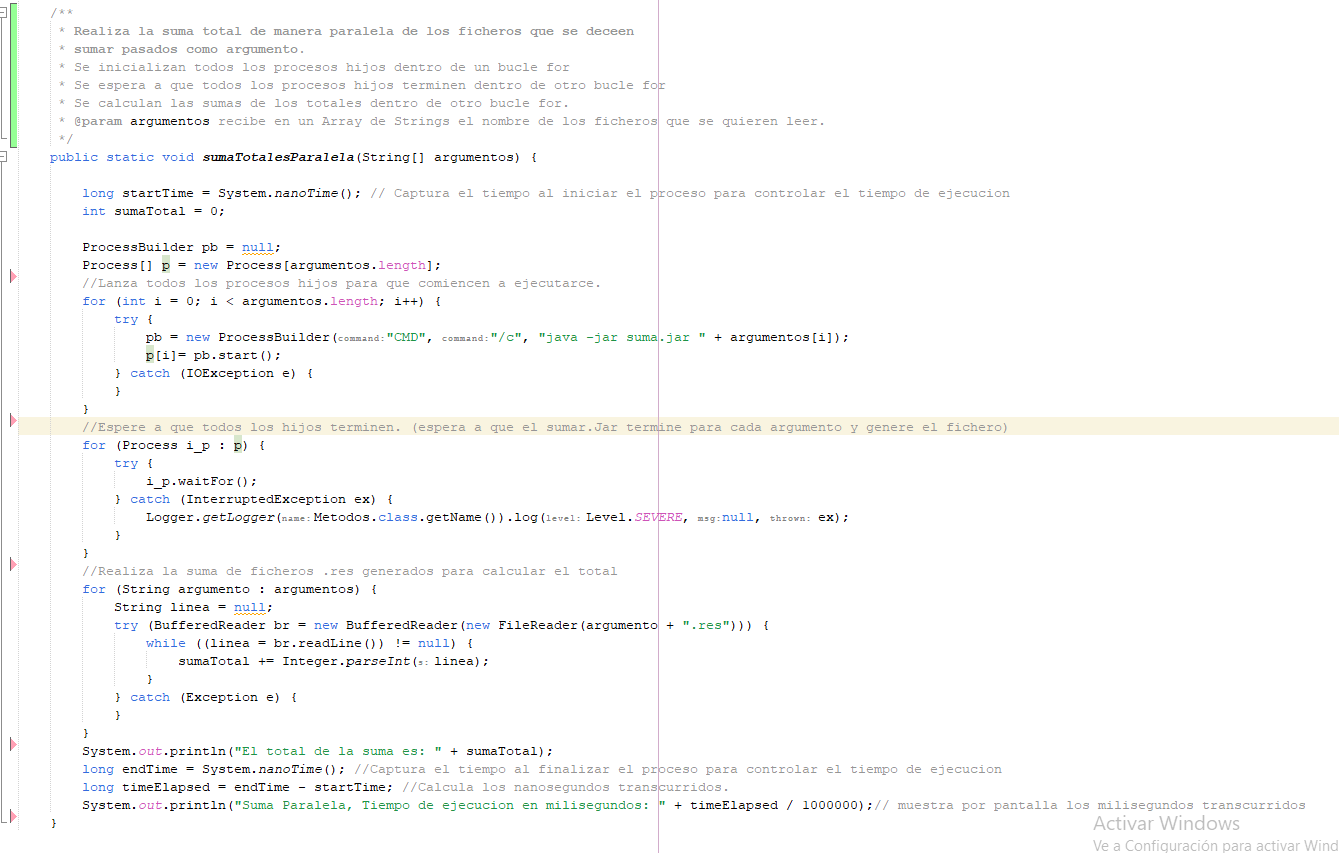
**Método sumaTotalSecuencial:**

Realiza la suma total de manera secuencial de los ficheros que se deseen sumar pasados como argumento, utilizando un bucle for.



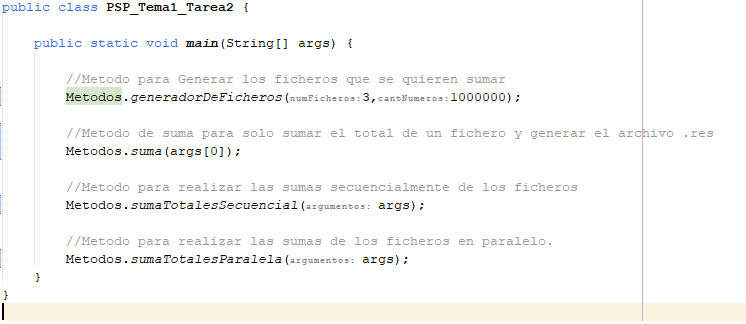
**Método sumaTotalParalela:**

Realiza la suma total de manera paralela de los ficheros que se deseen sumar pasados como argumento. Se inicializan todos los procesos hijos dentro de un bucle for se espera a que todos los procesos hijos terminen dentro de otro bucle for se calculan las sumas de los totales dentro de otro bucle for.

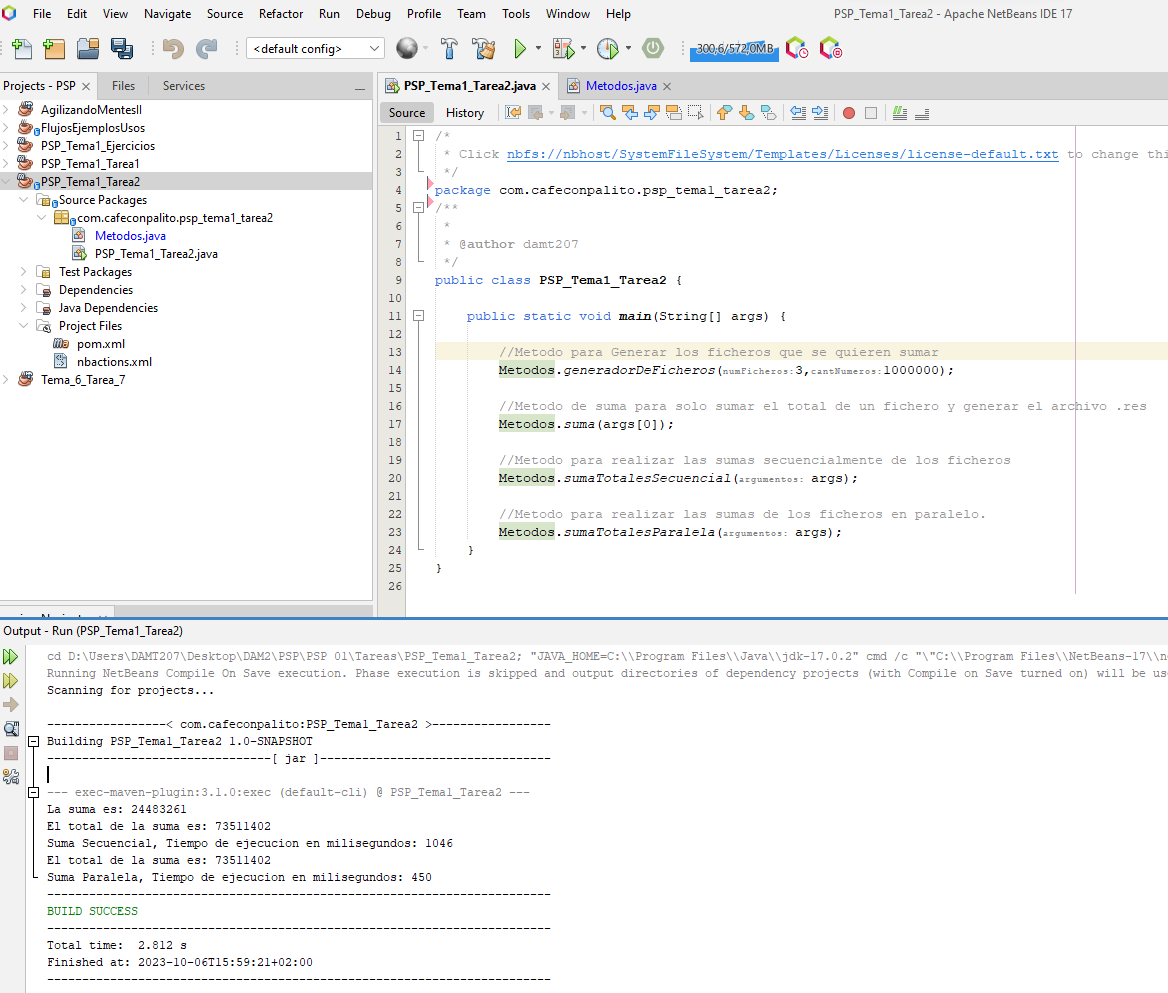


**Main:**

Para comprobar su funcionamiento dentro del Main se llaman a los metodos que se necesitan usar.

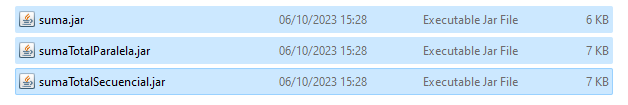


Ejecución de todos los métodos, para realizar una prueba inicial de que todo funciona correctamente.



Con todos los test pasados desde NetBeans activo solo el método que necesito para crear el .jar del módulo necesario utilizando el clean and build.

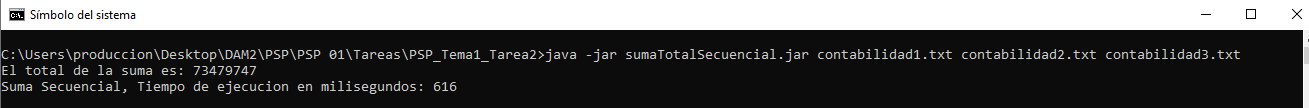
Obteniendo 3 .jar uno para cada método.



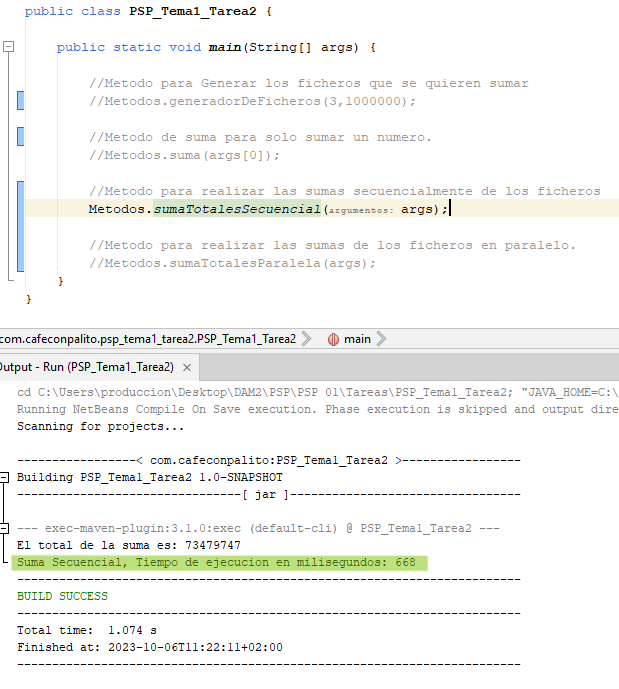
## **Comparación de tiempos entre la ejecución secuencial y paralela. Para medir tiempos de ejecución**

Con los .jar ya generados realizamos la ejecución de estos mismos para controlar los tiempos de ejecución.

**Tiempo de ejecución secuencial:**

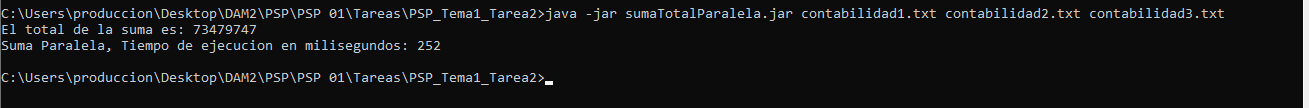
Mediante el CMD ejecuto él archivo sumaTotalSecuencial.jar, como se puede observar el tiempo de ejecución es de 616 milisegundos

Lanzando el mismo programa desde el NetBeans observamos que el tiempo de ejecución en secuencia es de 668 milisegundos.

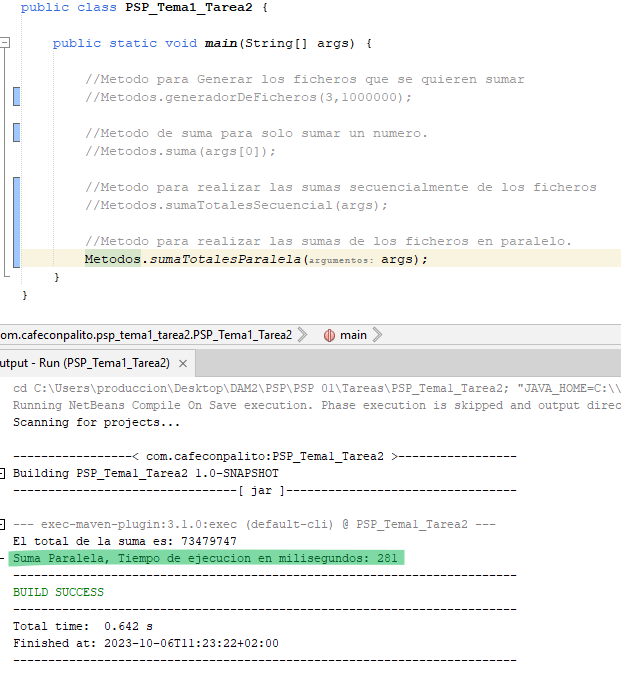
****

**Tiempo de ejecución paralela:**

Mediante el CMD ejecuto él archivo sumaTotalParalela.jar, como se puede observar el tiempo de ejecución es de 252 milisegundos.



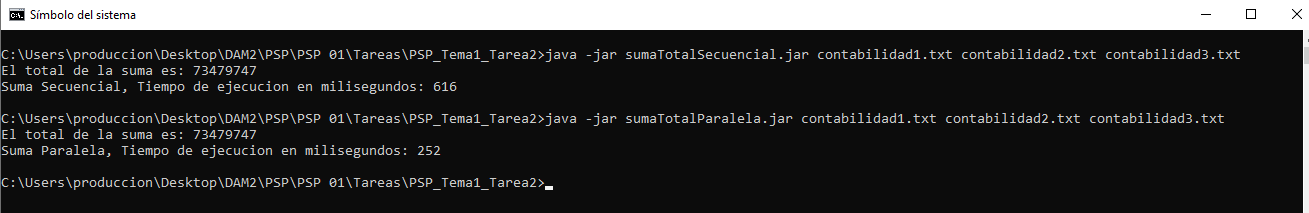
Lanzando el mismo programa desde el NetBeans observamos que el tiempo de ejecución en secuencia es de 281 milisegundos.



## **Conclusión**

Como se puede observar por los tiempos obtenidos para la ejecución Secuencial de 616 milisegundos y la ejecución Paralela de 252 milisegundos. Para analizar 3 archivos con un millón de números cada uno. Estos tiempos son bastante diferentes, por lo cual trabajar de manera paralela es mucho más rápida para realizar esta tarea, ya que la en Secuencial tiene que esperar a que termine la ejecución de cada Hijo, suma.jar, para pasar al siguiente.

En paralelo se ejecutan todos los Hijos, sumar.jar, a la vez y solo tiene que espera a que cada hijo termine para empezar a realizar las sumas totales.



Se realizaron varias pruebas con la cantidad de números que contenían los archivos para comprobar los tiempos de ejecución, pasando de 10, 100, 1000, 10000 y 1000000. Al final opte por trabajar con un millón números.

## **Bibliografía**

* Teoría del Módulo de Programación De Servicios Y Procesos que se nos facilita por Moodle.
* Método para poder calcular los milisegundos obtenido en el enunciado de la tarea Link: <https://www.techiedelight.com/es/measure-elapsed-time-execution-time-java/>