

# Tarea 02 de Sistemas Operativos

Salvador Gallardo Ortega, [salvador.gallardo@alumnos.uv.cl](mailto:salvador.gallardo@alumnos.uv.cl)

Nataniel Palacios Toro, [nataniel.palacios@alumnos.uv.cl](mailto:nataniel.palacios@alumnos.uv.cl)

Martin Carrillo Salinas, [martin.carrillo@alumnos.uv.cl](mailto:martin.carrillo@alumnos.uv.cl)

## 1. Introducción

En el contexto de la computación de alto rendimiento, el procesamiento paralelo y concurrente se ha convertido en una herramienta esencial para manejar grandes volúmenes de datos de manera eficiente. Este informe se enfoca en un programa específico escrito en C++ que utiliza multi-threading para procesar y analizar texto, creando un histograma de palabras. El objetivo principal es entender y visualizar el flujo de ejecución de los threads y cómo interactúan para lograr un procesamiento más rápido y eficiente.

## 2. Materiales y Métodos

### 2.1. Materiales:

- Datos: Un archivo de texto que contiene un extracto de un libro.

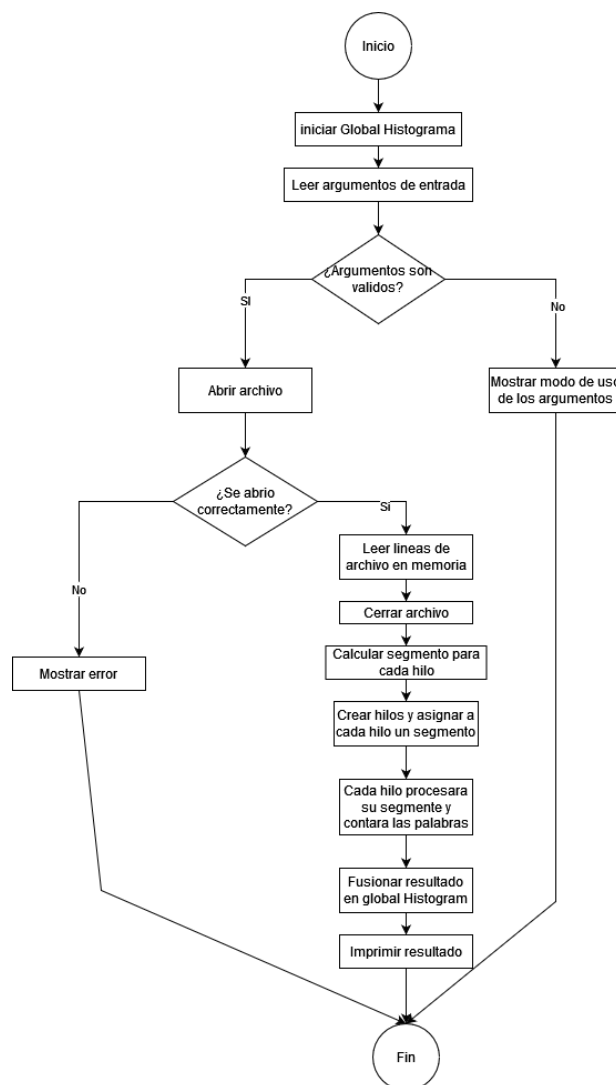
### 2.2. Métodos:

- Procesamiento de Texto: Se utilizaron funciones para limpiar y procesar el texto, eliminando puntuaciones y convirtiendo todas las letras a minúsculas.
- Multi-Threading: Se crearon múltiples threads para procesar diferentes segmentos del texto en paralelo.
- Sincronización: Se utilizó un mutex para garantizar que la actualización del histograma global sea segura en términos de threads.

### 3. Resultados

#### 3.1. Diagrama de Flujo de la Implementación

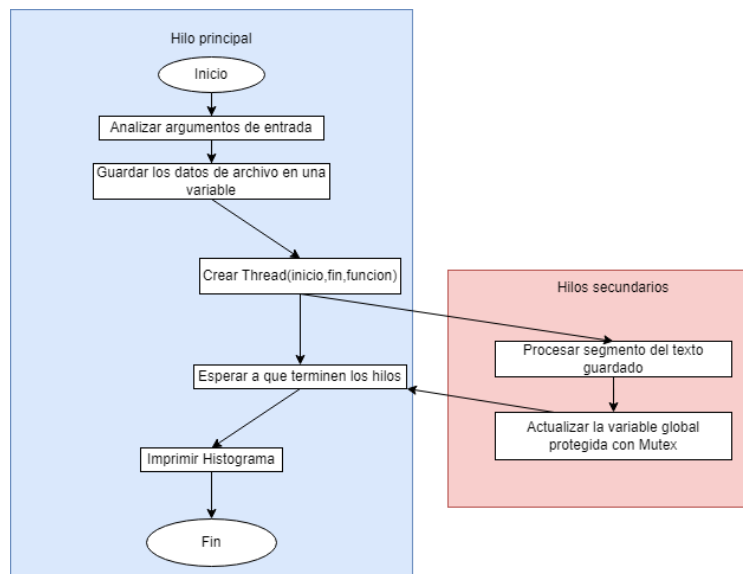
El siguiente diagrama de flujo ofrece una visión general de la implementación del programa, desde la lectura de los argumentos de la línea de comandos hasta la finalización del mismo. Es importante mencionar que este diagrama no detalla de manera específica el funcionamiento de los hilos en paralelo, pero proporciona una comprensión general de cómo opera el programa. Para una comprensión más detallada de cómo funcionan los hilos responsables de la generación del histograma, se puede consultar la sección 3.2, donde se proporciona una explicación más completa.



**Figura 1:** Implementación general del programa, elaboración propia

### 3.2. Diseño multi-thread

El siguiente diagrama de diseño multihilo describe la estrategia utilizada para desarrollar un programa que genera un histograma haciendo uso de múltiples hilos y mutex. Es importante destacar que, dado mi falta de experiencia en la creación de diagramas de diseño multihilo, consulté diversas fuentes en Internet para obtener orientación. Aunque la notación utilizada podría no ser la más precisa, logra transmitir la idea central del diseño que se planteó inicialmente.

**Figura 2:** Diseño de resolución con multi-thread

## 4. Discusión y conclusiones

En esta tarea, hemos adquirido conocimientos sobre la programación en C++ y la implementación de programación multihilo. Este aprendizaje nos ha permitido descubrir cómo procesar texto utilizando múltiples hilos, destacando la importancia de la programación multihilo en el desarrollo de aplicaciones de alto rendimiento. Además, hemos comprendido la relevancia de la sincronización y la protección de las secciones críticas para asegurar la coherencia en el resultado final.

## Referencias

- [1] Author 1, A.B.; Author 2, C.D. Title of the article. Abbreviated Journal Name Year, Volume, page range.
- [2] Author 1, A.; Author 2, B. Title of the chapter. In *Book Title*, 2nd ed.; Editor 1, A., Editor 2, B., Eds.; Publisher: Publisher Location, Country, 2007; Volume 3, pp. 154–196.
- [3] Author 1, A.; Author 2, B. *Book Title*, 3rd ed.; Publisher: Publisher Location, Country, 2008; pp. 154–196.

