



# Tecnológico de Monterrey

## **Actividad integradora 5.3 Resaltador de sintaxis en paralelo**

Francisco José Urquizo Schnaas A01028786

13 de junio de 2024

Implementación de métodos computacionales

Gilberto Echeverría Furió

## Análisis de tiempo y speed-up

Usando la función `:timer.tc()`, se logró determinar el tiempo de ejecución de cada uno de los modos empleados para resolver el problema en cuestión (secuencial y paralelo). Se ejecutó el programa 5 veces y se obtuvo un promedio de los tiempos de ejecución de cada modo. Este proceso se muestra a continuación:

1. Primera ejecución
  - Secuencial: 37432  $\mu$ s
  - Paralelo: 27449  $\mu$ s
2. Segunda ejecución
  - Secuencial: 43255  $\mu$ s
  - Paralelo: 26353  $\mu$ s
3. Tercera ejecución
  - Secuencial: 35641  $\mu$ s
  - Paralelo: 27091  $\mu$ s
4. Cuarta ejecución
  - Secuencial: 34916  $\mu$ s
  - Paralelo: 26210  $\mu$ s
5. Quinta ejecución
  - Secuencial: 39178  $\mu$ s
  - Paralelo: 24682  $\mu$ s

Habiendo tomado las medidas anteriores, se determinaron los siguientes tiempos promedio:

- **Secuencial: 38084.4  $\mu$ s**
- **Paralelo: 26357.0  $\mu$ s**

Con esta información, se concluyó que el modo paralelo es aproximadamente **1.44 veces más rápido** que el modo secuencial, usando todos los cores del equipo de cómputo en el que se realizó esta comparación (12 cores).

## Análisis de complejidad

La complejidad del programa secuencial es  $O(n \cdot m)$ , donde  $n$  es el número de archivos y  $m$  es el número de líneas por archivo. Esto se debe a que el programa lee y procesa cada archivo línea por línea, aplicando varias operaciones de regex y escribiendo la salida correspondiente. En contraste, el programa paralelo también tiene una complejidad de  $O(n \cdot m)$  en términos de trabajo total, pero distribuye las tareas entre múltiples núcleos de CPU. Idealmente, esto reduce el tiempo de ejecución a  $O(\frac{n \cdot m}{p})$ , donde  $p$  es el número de núcleos disponibles. Sin embargo, la eficiencia real del programa paralelo puede verse afectada por la sobrecarga de sincronización y otros factores relacionados con la paralelización.

Es por esta distinción en la complejidad de los diferentes programas que el tiempo se reduce con el modo paralelo.

## **Implicaciones éticas**

El desarrollo de tecnologías como el resaltador de sintaxis, que mejora la legibilidad y comprensión del código fuente, tiene implicaciones éticas importantes en la sociedad. Por un lado, facilita el aprendizaje y la colaboración en programación, democratizando el acceso al conocimiento técnico y fomentando la inclusión digital. Esto puede reducir barreras de entrada para nuevos programadores y contribuir al crecimiento de una comunidad de desarrolladores diversa y capacitada.

Sin embargo, también existen riesgos asociados. La automatización y optimización de tareas pueden reducir la demanda de ciertas habilidades humanas, desplazando trabajos y aumentando la dependencia en la tecnología. Además, la accesibilidad a herramientas avanzadas debe ser equitativa para evitar brechas tecnológicas que perpetúen desigualdades sociales. Es crucial que el desarrollo de estas tecnologías considere estos impactos y promueva un uso responsable, inclusivo y ético, asegurando que los beneficios sean compartidos ampliamente y no exacerben las disparidades existentes.