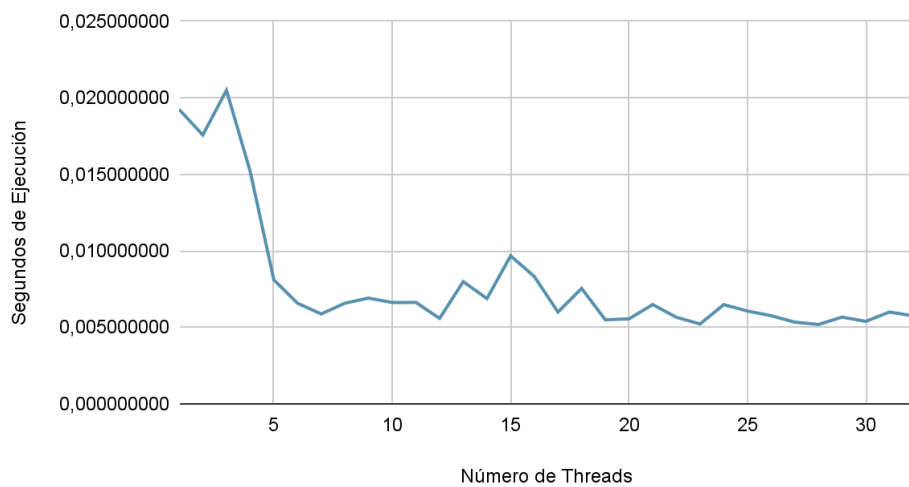


# TP2:Imagine

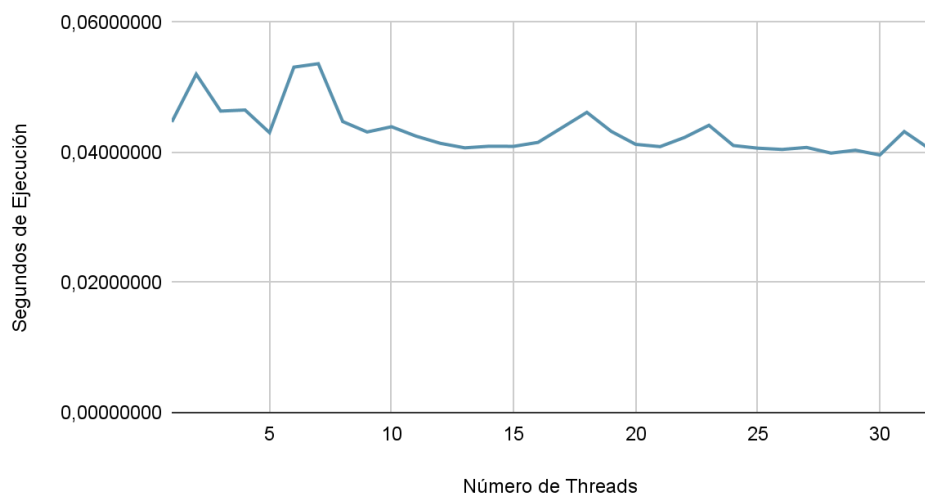
Alumnos: Santiago Abelle y Gabriel Atienza

- 1) A la hora de comparar el tiempo de espera, decidimos usar un filtro pixel sencillo como blackWhite, ya que para testear cómo rinden un multithread en un filtro extremadamente sencillo.

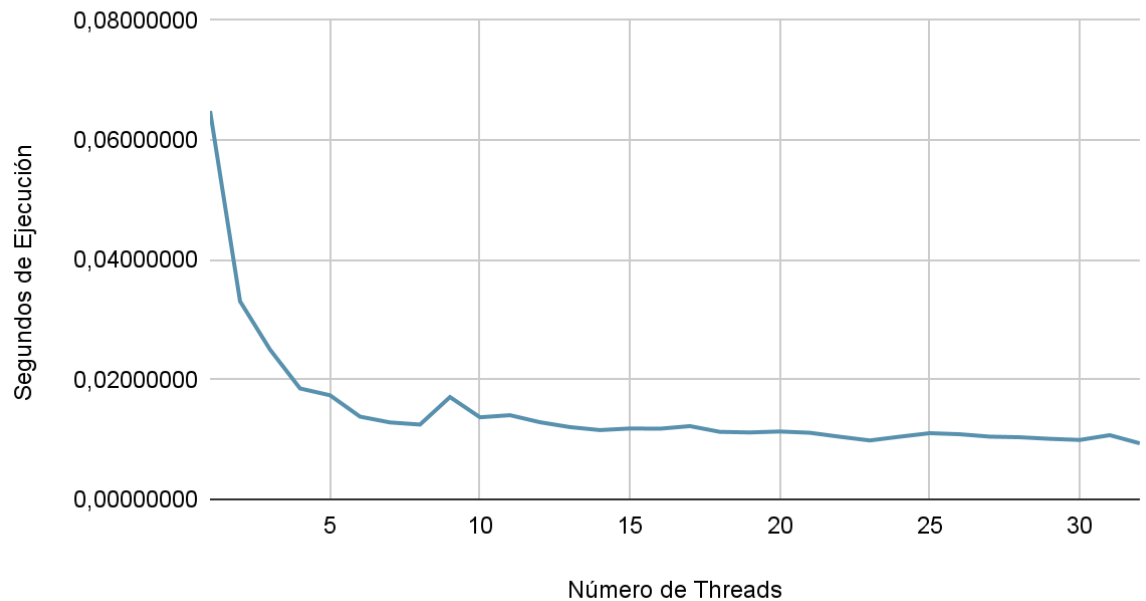
Black and White Imágen cebra.ppm



Zoom x2 Imágen cebra.ppm



## Sharpen Imágen cebra.ppm



Como se puede ver, en un filtro sencillo el uso de multi threads afecta al rendimiento por un muy pequeño margen. El cual aunque tenga un momento en el que deja de ser útil (usando 3 threads) generalmente significa una pequeña ventaja.

En una imagen con menor tamaño, se puede esperar que dependiendo de los threads a usar, tenga un efecto negativo en el rendimiento, ya que si se usan muchos threads en una imagen de menor tamaño puede resultar en una desventaja.

Fuera del tema rendimiento, hay un tema que se debe tener en cuenta, que es el hardware de la computadora. Ya que si se usan una gran cantidad de threads en una computadora con un mal hardware, puede resultar en problemas con el rendimiento de la misma computadora.

- 2) Aunque parezca que siempre puede ser de gran utilidad el paralelizar, este puede llegar a ser una desventaja en casos donde la tarea a paralelizar no es lo suficientemente compleja como para que sea paralelizada. Este ejemplo se puede ver en el caso de utilizar una imagen de menor tamaño usando algunos de los filtros que realizamos.
- 3) Aunque no llegamos a terminar el loader, con lo aprendido en clase, se puede teorizar que un loader single-thread sería mucho más lento, ya que al ya

dividirse la tarea a la hora de realizar los filtros en muchas imágenes, el hacerlo con multithread lograría un mejor rendimiento.