



Instituto Politécnico Nacional  
Escuela superior de Cómputo  
Aplicaciones para Comunicaciones de Red  
Grupo: 3CV8

*Práctica 1.*  
**“Generación de Hilos”**

Elaborado por:  
Quiroz Ortiz Jordi  
Ruvalcaba Montoya Jesús Eduardo

Profesor:  
Rangel González Josué

# Introducción

## Thread (Hilo)

Un Thread es un "hilo" de ejecución de un programa. Casi podríamos decir que un Thread o Hilo es un programa en ejecución ya que en general un programa tiene un solo hilo de ejecución. Cuando se ejecuta un programa, el sistema operativo crea un hilo para ejecutar el programa. Todos los sistemas operativos que se utilizan actualmente son multihilo, es decir, pueden ejecutar simultáneamente varios hilos (o varios programas si se quiere).

## Imágenes RGB

Un píxel es la menor unidad de color que compone una imagen digital, a su vez el pixel está conformado al menos tres elementos; RGB o rojo, verde y azul, por sus siglas en inglés. En ocasiones está compuesto por un cuarto elemento llamado alpha el cual determina la transparencia del pixel. Cualquiera de los cuatro valores mencionados anteriormente toman un valor que va desde 0 hasta 255 en decimal. Entendiendo lo anterior podemos decir que si modificamos cada uno de sus píxeles y los "saturamos" cambiando su valor a 255 del elemento rojo, por ejemplo; obtendremos una imagen saturada en su parte roja.

## Implementación para multiplicación de matrices

Una biblioteca de hilos proporciona las herramientas necesarias para crear y administrar hilos en la mayoría de los lenguajes de programación. De manera general se distinguen dos tipos de hilos: hilos a nivel de usuario e hilos a nivel de kernel. De igual manera, el soporte para hilos debe ser proporcionado a nivel de usuario o a nivel de kernel

- Los subprocesos de nivel de kernel son compatibles y gestionados directamente por el sistema operativo.
- Los subprocesos de nivel de usuario se admiten por encima del kernel en el espacio de usuario y se administran sin soporte de kernel.

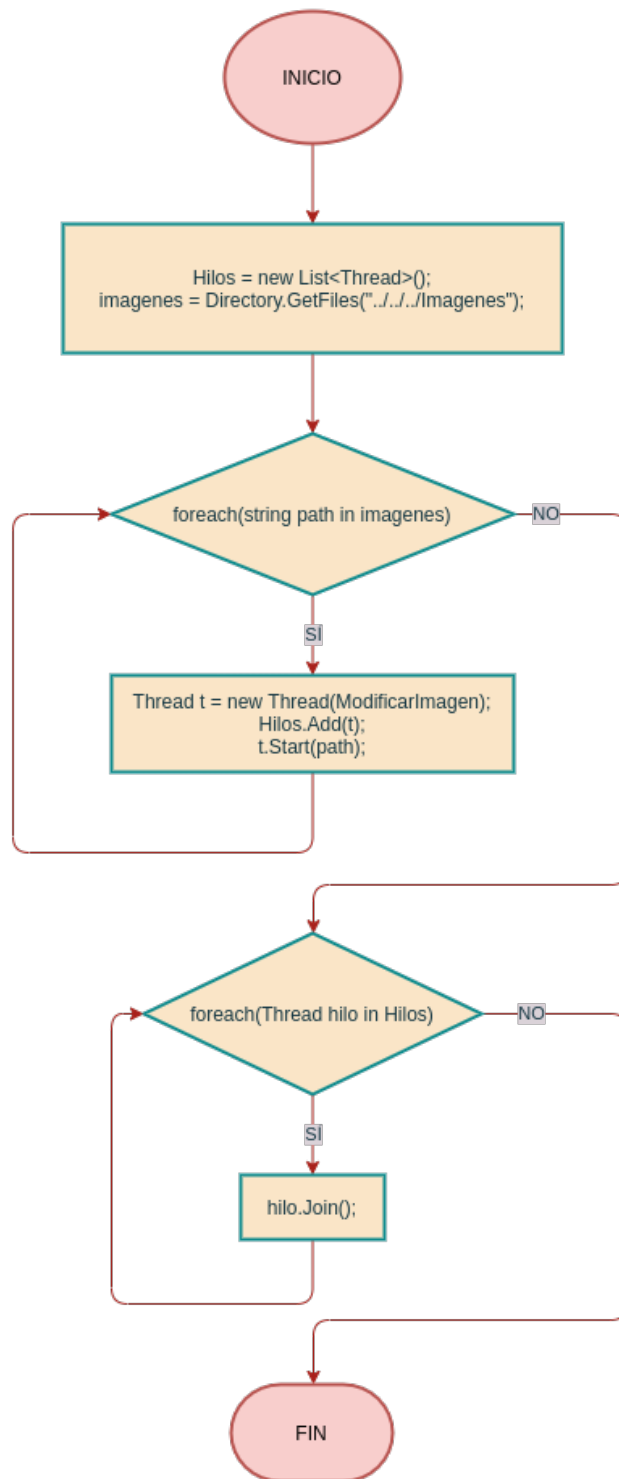
## Descripción del problema

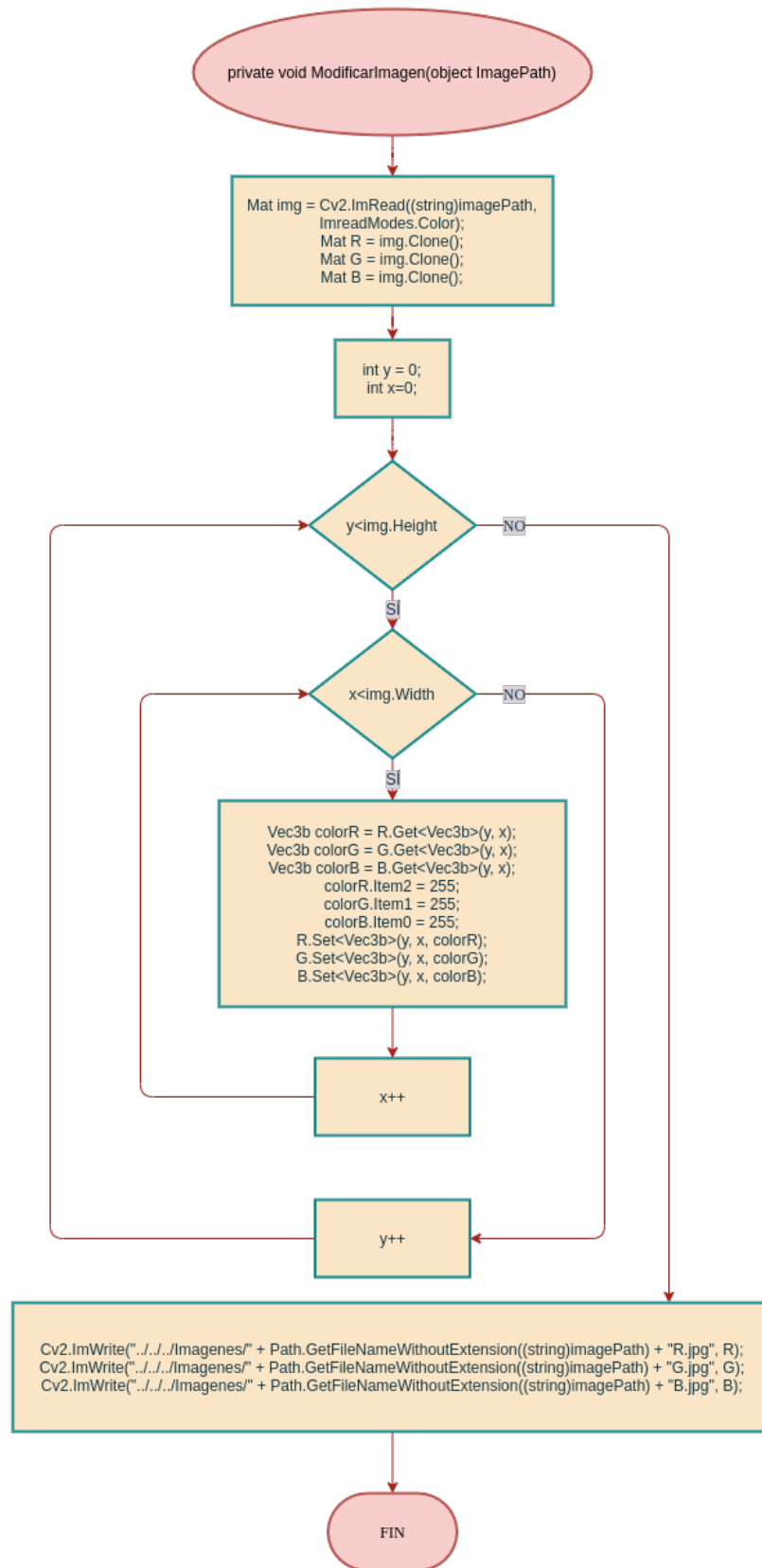
Con el fin de comprender la utilización e implementación de hilos, se plantean los siguientes problemas a resolver:

- Dentro de un directorio copiar +10 imágenes, cada una de estas imágenes van a ser asignadas a un hilo. El hilo va a abrir la imagen como una matriz RGB y va a generar 3 copias de dicha imagen, la primera copia va a asignar el valor R a 255, en la segunda copia va a asignar el valor G a 255 y en la tercera copia va a asignar al valor B a 255. Cada una de las copias modificadas serán guardadas nuevamente como una imagen, lo cual provocará que al final se tengan por cada imagen original 3 más; una saturada en Red, otra saturada en Green y otra saturada en Blue.
- Generar la multiplicación de matrices donde  $A = 2000 \times 1500$ ,  $B = 1500 \times 2000$ . Se van a ejecutar diferentes cantidades de hilos que resuelvan esta matriz, en cada ejecución se

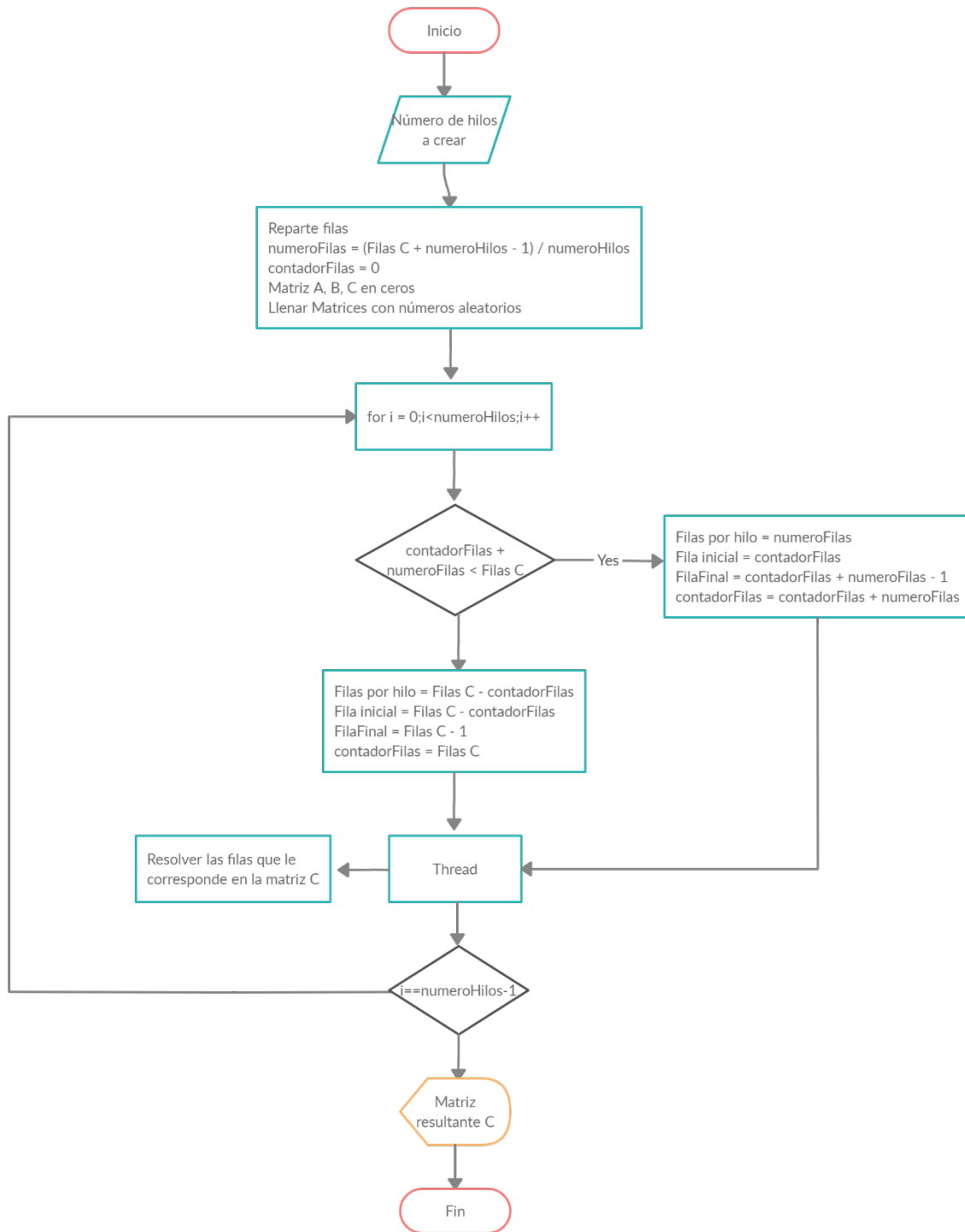
deben medir los tiempos y al final se debe generar una conclusión: ¿Qué sucede cuando más hilos ejecutan la matriz?, ¿Vale la pena crear grandes cantidades de hilos para resolver el problema? La cantidad de hilos a ejecutar es 2, 4, 8, 16, 32.

### **Diagrama de flujo (Filtros RGB)**





## Diagrama de flujo (Multiplicación de Matrices)



Las mediciones de tiempo con respecto al número de hilos son las siguientes:

Número de hilos	Tiempo (segundos)
2	301.189s
4	513.891s
8	494.734s
16	643.081s
32	647.513s

Como se puede observar en la tabla, para este caso en particular, mientras más hilos se ejecutan, más tiempo se toma en terminar las tareas, por lo tanto, no vale la pena crear grandes cantidades de hilos.

## Conclusión

Al utilizar aplicaciones concurrentes, utilizamos de manera óptima el poder de nuestra computadora, sin embargo, como se vio en las pruebas, utilizar un número excesivo de hilos (innecesarios), el resultado puede ser el opuesto a lo que esperamos en cuanto a tiempo de ejecución.