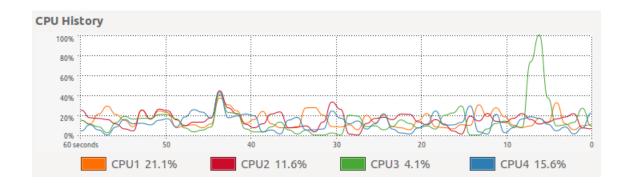
TAREA 3:

ALEJANDRO GÜEREÑA MORÁN

JESÚS DAVID SANCHEZ CARBALLO

La computadora usada tiene un procesador Intel i5-2520M con HyperThreading activado

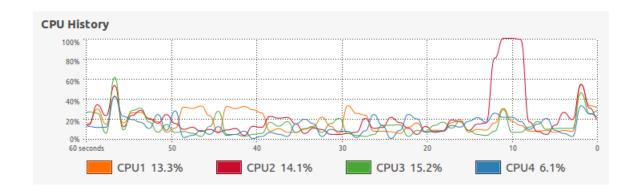
Versión: Serial sin modificaciones **Tiempo de Ejecución:** 1.8900 s



Versión: Serial con modificaciones

Esta versión intercambia la forma de procesar los pixeles, donde se va recorriendo 1 columna por el resto de las filas. Esto provocó que el tiempo de ejecución fuera mayor, ya que no hay un bueno uso de la caché, porque los valores calculados y almacenados en caché para su uso frecuente no se aprovechan, ya que no se están procesando los pixeles vecinos en seguida, sino se va saltando por distintas regiones de la imagen y procesando los pixeles sin que los vecinos sean procesados necesariamente en seguida.

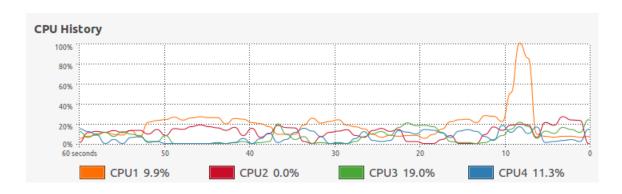
Tiempo de Ejecución: 2.90



Versión: Paralela en un procesador. Unidades Funcionales: 1 Cores Físicos, 1

Thread (Cores Lógicos)

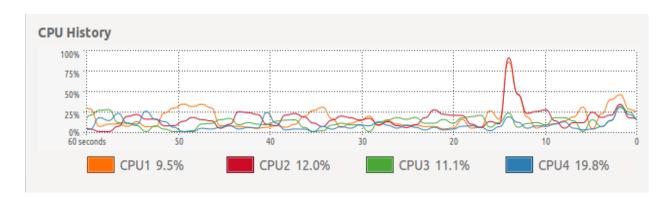
Tiempo de Ejecución: 0.12 s



Versión: Paralela en un procesador. Unidades Funcionales: 1 Cores Físicos, 2

Threads (Cores Lógicos)

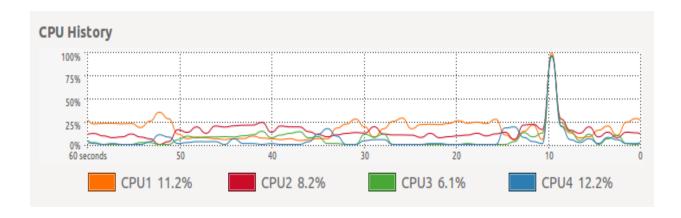
Tiempo de Ejecución: 0.16 s



Versión: Paralela en un procesador. Unidades Funcionales: 2 Cores Físicos, 4

Threads (Cores Lógicos)

Tiempo de Ejecución: 0.18 s



No existe una variación significativa entre los procesos cuando se ejecutan con mas unidades funcionales de cores y threads disponibles del procesador, aunque se observa que mientras mas unidades funcionales ejecutaron los 4 threads, aumento el tiempo de ejecución por 0.02-0.04 segundos. Suponemos que esto es debido a la forma en que el programa esta midiendo el tiempo, porque se esperaría mejor desempeño si más unidades funcionales ejecutaran los threads por separado.

Se observa un aumento de 0.05 o hasta 0.1 segundos cuando el sistema operativo esta corriendo otros procesos independientes al programa al mismo tiempo que el programa de la práctica.

Otra característica que se observó durante la práctica, es que al aumentar el número de pixeles procesados por cada thread, el tiempo de ejecución era menor respecto a tener menos pixeles a procesar por cada thread. Ejecutar un programa con varios threads mejora al desempeño de éste, hasta llegar a un número óptimo de threads que pueden ser procesados paralelamente por las unidades funcionales. Sin embargo, agregar más threads del número óptimo, provocará una degradación en el desempeño del programa, ya que las unidades funcionales disponibles de la computadora no podrá ejecutarlos paralelamente y los threads quedarán esperando para ser ejecutados.