***Universidad de Caldas***

***Programación Concurrente y Distribuida***

***Reinel Tabares Soto***

***Taller OpenMP - Multiprocessing - Programación Concurrente y Distribuida***

**Objetivo**: Este taller tiene como objetivo comparar el rendimiento de OpenMP en C y Multiprocessing en Python en la ejecución de cálculos intensivos en paralelo.

Contestar y/o programar las siguientes preguntas, utilizando programación OpenMP en C, y Multiprocessing en Python.

1. ¿Qué es la programación paralela y cómo se relaciona con: la biblioteca multiprocessing, y con OpenMP?
2. Diferencias entre procesos y subprocesos.
3. ¿Cómo utilizar multiprocessing para dividir una tarea en múltiples procesos?
4. Actividad práctica: suma de números enteros.
   1. Utilizar **multiprocessing** para acelerar el cálculo de la suma de los primeros N números enteros.
   2. Comparar el tiempo de ejecución de la versión secuencial con la versión paralela.
   3. Experimentar con diferentes valores para el número de procesos para ver cómo afecta el rendimiento.
   4. Crear una tabla de comparación.
5. Realizar lo mismo que en (4) para OpenMP.
6. Crear y resolver un problema que le parezca interesante usando tanto multiprocessing como OpenMP, y entregar una tabla de comparación donde se mire el tiempo de ejecución, y los parámetros modificados en los distintos experimentos.
7. Redactar una discusión de cuándo es apropiado utilizar programación paralela y cuándo no lo es.