# Laboratorio 05

***Competencias para desarrollar***

Distribuir la carga de trabajo entre hilos utilizando programación en C y OpenMP.

***Instrucciones***

Esta actividad se realizará individualmente. Al finalizar los períodos de laboratorio o clase, deberá entregar este archivo en formato PDF y los archivos .c en la actividad correspondiente en Canvas.

1. **(18 pts.)** Explica con tus propias palabras los siguientes términos:
2. Private : Especifica que cada subproceso debe de tener su propia instancia de una variable.
3. Shared: Especifica que una o varias variables deben compartirse entre todos los subprocesos.
4. Firstprivate: Determina que cada subproceso debe de tener su propia instancia de una variable y que la variable se debe inicializar con el valor de la variable.
5. Barrier: Especifica una barrera explicita en el punto en donde el construct aparece.
6. Critical: Restricte la ejecución del bloque estructurado asociado a un single thread en el momento.
7. Atomic: Se asegura que se acceda a una locación especifica de almacenamiento automáticamente, en vez de
8. **(12 pts.)** Escribe un programa en C que calcule la suma de los primeros N números naturales utilizando un ciclo ***for* paralelo**. Utiliza la cláusula **reduction con +** para acumular la suma en una variable compartida.
9. Define N como una constante grande, por ejemplo, N = 1000000.
10. Usa omp\_get\_wtime() para medir los tiempos de ejecución.
11. **(15 pts.)** Escribe un programa en C que ejecute tres funciones diferentes en paralelo usando la **directiva #pragma omp sections**. Cada sección debe ejecutar una función distinta, por ejemplo, una que calcule el factorial de un número, otra que genere la serie de Fibonacci, y otra que encuentre el máximo en un arreglo, operaciones matemáticas no simples. Asegúrate de que cada función sea independiente y no tenga dependencias con las otras.
12. **(15 pts.)** Escribe un programa en C que tenga un ciclo for donde se modifiquen dos variables de manera paralela usando #pragma omp parallel for.
    1. Usa la cláusula shared para gestionar el acceso a la variable1 dentro del ciclo.
    2. Usa la cláusula private para gestionar el acceso a la variable2 dentro del ciclo.
    3. Prueba con ambas cláusulas y explica las diferencias observadas en los resultados.
13. **(30 pts.)** Analiza el código en el programa Ejercicio\_5A.c, que contiene un programa secuencial. Indica cuántas veces aparece un valor key en el vector a. Escribe una versión paralela en OpenMP utilizando una descomposición de tareas **recursiva**, en la cual se generen tantas tareas como hilos.
14. **REFLEXIÓN DE LABORATORIO: se habilitará en una actividad independiente.**