



# PROYECTO DE ALGORITMOS Y ESTRUCTURAS DE DATOS

## *MULTITHREADING*

*~PRIMERA ENTREGA~*

### Integrantes:

- David Felipe Martinez Castiblanco
- Miguel Angel Castillo Espitia

Fecha de entrega: 11 de noviembre del 2018

Objetivos alcanzados:

- Crear funciones usando multithreading.
- Corroborar la reducción del tiempo al usar multithreading en vez de un solo thread, ejecutando funciones simples como la suma de vectores, el producto punto entre vectores, la multiplicación de un vector por un escalar.

Tuvimos problemas porque el tiempo era irregular en algunas funciones; para problemas muy complejos computacionalmente es mejor usar multithreading de lo contrario solo se incurre en gasto de memoria y recursos computacionales donde el uso de un solo thread es perfecto.

- Imprimir cadenas de caracteres de forma correcta entre cada thread.

Dado que al usar multithreading los procesos son simultáneos, las impresiones entre cada thread no se estaban realizando, esto ocurría debido a que el método “cout” acumula los caracteres para posteriormente mandarlos a la consola. Nos dimos cuenta que usando multi-thread las impresiones se hacen al final, por esa razón debemos usar flush y así imprimir inmediatamente cada cadena de caracteres que queremos.

- Adquirir conocimientos sobre multithreading:

-Métodos de la clase thread y future.

-Funcionamiento de threads mediante recursión; la recursión implica la creación de más de un thread por nivel de recursión, ocasionando mucho gasto computacional y en algunos casos más ineficiente que la recursión con un solo thread(realizamos pruebas con la sucesión de fibonacci mediante recursión).

Nuevos objetivos:

- Afianzar conocimientos sobre multithreading.
- Encontrar el punto de equilibrio entre cuántos threads usar y de qué manera usarlos.
- Intentar llevar el uso de multi-thread hacia algoritmos de ordenamiento de datos para comparar su tiempo de ejecución con respecto al de los algoritmos que usan un único thread.
- Diferenciar correctamente el uso de multi-threads y multiprocesos para evitar la existencia de ambigüedades.

\*\*\*\*\*