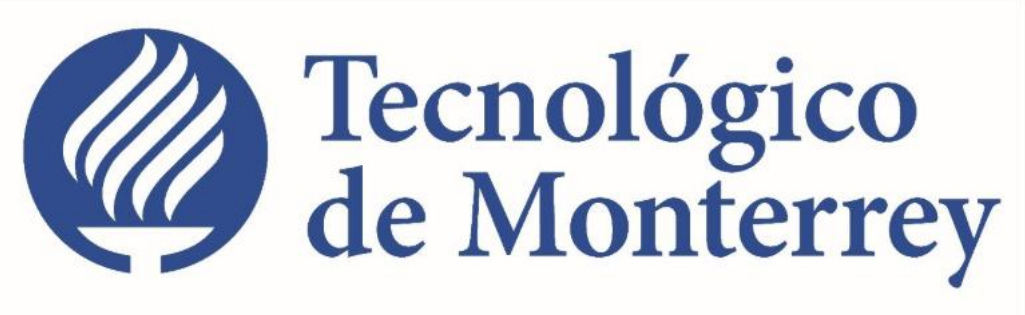


Instituto Tecnológico y de Estudios Superiores de Monterrey

Programación Paralela con OpenMP:

Suma de Arreglos en Paralelo

Profesores titulares: Gilberto Echeverría Furió,

Yetnalezi Quintas Ruiz y

Maestra María Mylen Treviño Elizondo

A01688888 Jeanette Rios Martinez

1ro de febrero 2026

Introducción

Acorde a las revisiones de lecturas y material podemos decir que la programación paralela es una técnica fundamental en el desarrollo de software moderno, ya que nos permite aprovechar de manera eficiente la capacidad de procesamiento de los sistemas actuales, los cuales cuentan con múltiples núcleos de ejecución lo que nos permite hacer un mejor uso de los mismos. A través del uso de múltiples hilos, es posible dividir un problema en tareas más pequeñas de manera granulada que pueden ejecutarse de forma simultánea, reduciendo significativamente el tiempo de ejecución y nos apoya a optimizar nuestros resultados.

En esta actividad se implementa una solución paralela para la suma de dos arreglos numéricos utilizando la librería **OpenMP** en lenguaje C++como se revisó en la guía de ayuda. El objetivo es demostrar cómo un problema numérico sencillo puede beneficiarse del paralelismo, cumpliendo con los principios de diseño de algoritmos paralelos y la correcta selección de un modelo de paralelización adecuado a manera básica.

Repositorio de GitHub

El proyecto desarrollado se encuentra disponible en el siguiente repositorio personal de GitHub para su consulta y evaluación:

https://github.com/Jeanette-Rios/A01688888_A4.2/blob/main/progParaleloA01688888.cpp

Ejecuciones del programa

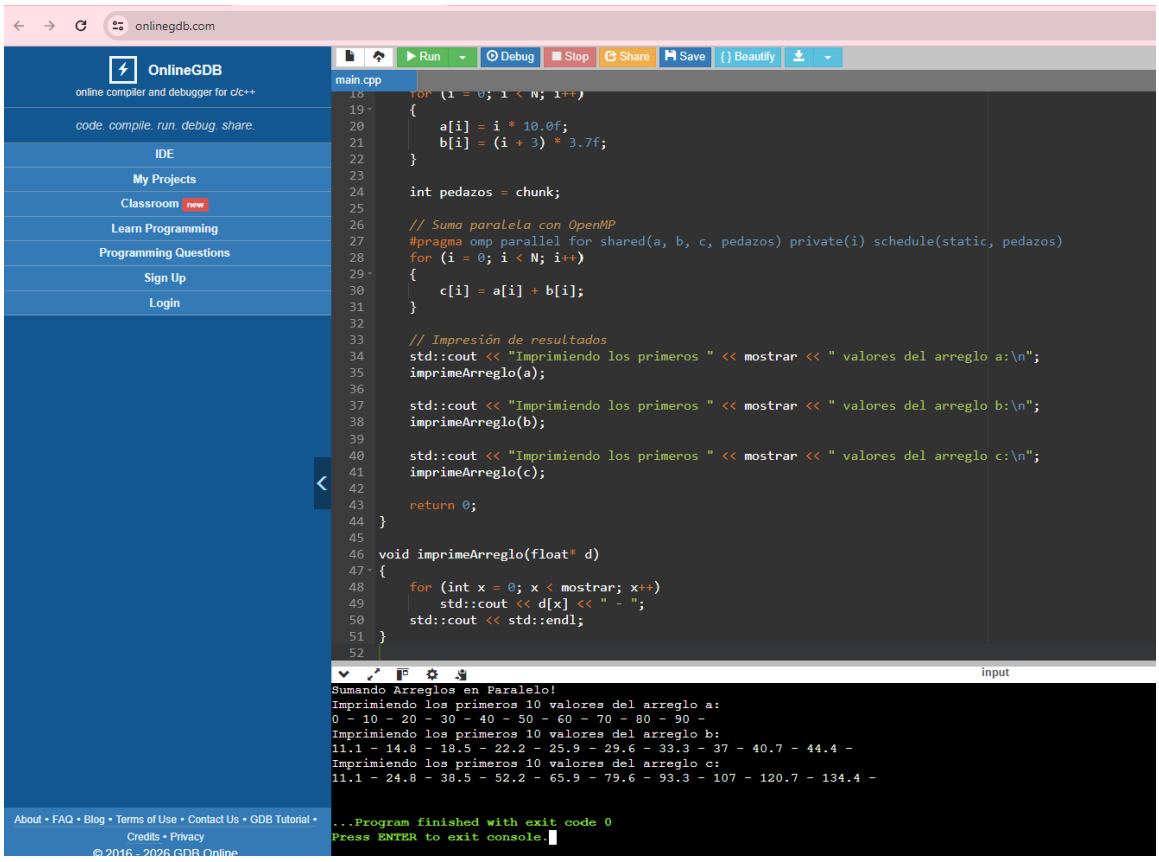
A continuación se muestran capturas de pantalla correspondientes a **dos ejecuciones diferentes del programa**, en las cuales se modificaron únicamente los parámetros como el tamaño del arreglo (N) y el tamaño del bloque (chunk) para observar y documentar su comportamiento.

Utilizando los siguientes datos

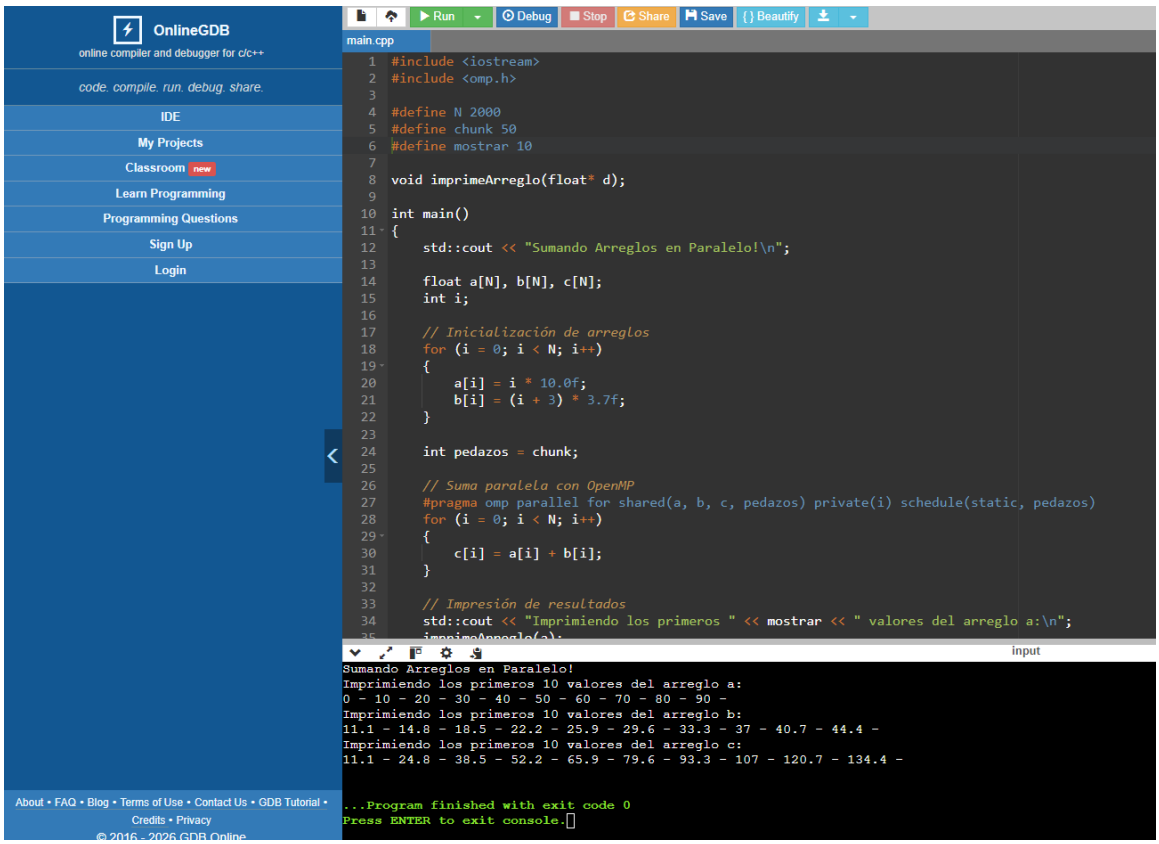
Ejecución 1: N = 1000, chunk = 100

Ejecución 2: N = 2000, chunk = 50

CAPTURA 1:



CAPTURA 2



Explicación del código y resultados

El programa inicia definiendo constantes de precompilación que determinan el tamaño de los arreglos, el tamaño de los bloques que serán procesados por cada hilo y la cantidad de elementos que se mostrarán en pantalla para verificar los resultados.

Posteriormente, se declaran tres arreglos:

a y b, los cuales son inicializados con valores calculados.

c, que almacena el resultado de la suma de los arreglos a y b.

La parte principal del programa es el ciclo for paralelo, el cual utiliza la directiva #pragma omp parallel for. Esta directiva permite distribuir las iteraciones del ciclo entre múltiples hilos, especificando:

Variables compartidas (shared) para los arreglos.

Variables privadas (private) para el índice del ciclo.

Una planificación estática (schedule(static, chunk)) que divide el trabajo en bloques de tamaño definido.

Finalmente, el programa imprime los primeros elementos de cada arreglo para comprobar que la suma se realizó correctamente. Los resultados obtenidos muestran que cada elemento del arreglo c corresponde a la suma correcta de los elementos en el mismo índice de los arreglos a y b, validando el correcto funcionamiento del algoritmo paralelo.

Reflexión sobre la programación paralela

La programación paralela representa una herramienta clave para resolver problemas que requieren alto rendimiento computacional. En esta práctica se pudo observar que, cuando las operaciones son independientes entre sí, como en la suma de arreglos, el paralelismo resulta especialmente eficiente y sencillo de implementar.

El uso de OpenMP facilita la creación de versiones paralelas de programas secuenciales sin necesidad de modificar drásticamente la estructura del código. Además, permite explotar de manera efectiva los recursos de hardware disponibles en los procesadores modernos. Esta actividad refuerza la importancia de identificar problemas paralelizables y seleccionar correctamente el modelo de paralelización para obtener soluciones correctas, eficientes y escalables.