**Taller Recuperativo Computación Paralela y Distribuida**

Se realizaron pruebas para determinar si era conveniente segmentar la entrada de datos al mostrarla por pantalla, dando como resultado que el cambio no era significativo, en ocasiones el código sin segmentar era mas eficiente. Por ejemplo, con n = 20665001 se obtuvo por consola que el rendimiento del algoritmo sin segmentar era más eficiente.

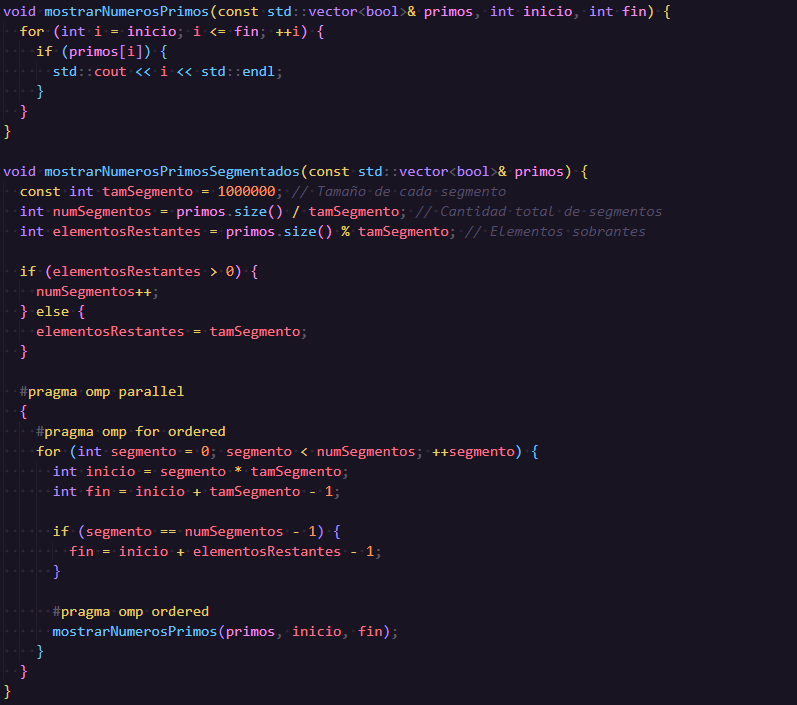


Ilustración código con segmentación

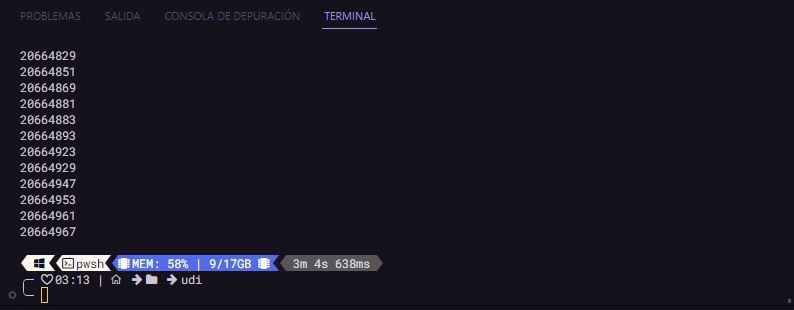
****

Ilustración algoritmo con segmentación de código

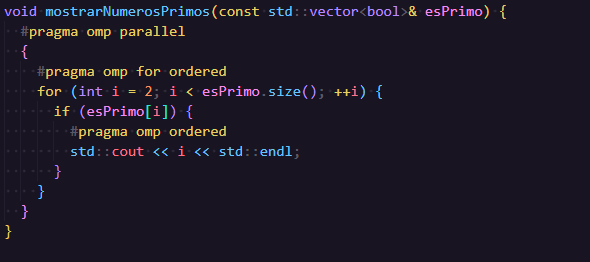


Ilustración código sin segmentación

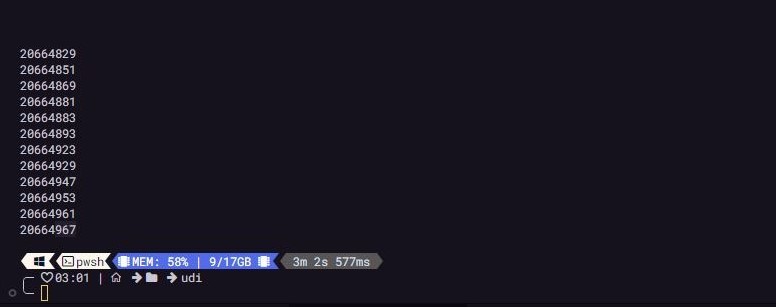
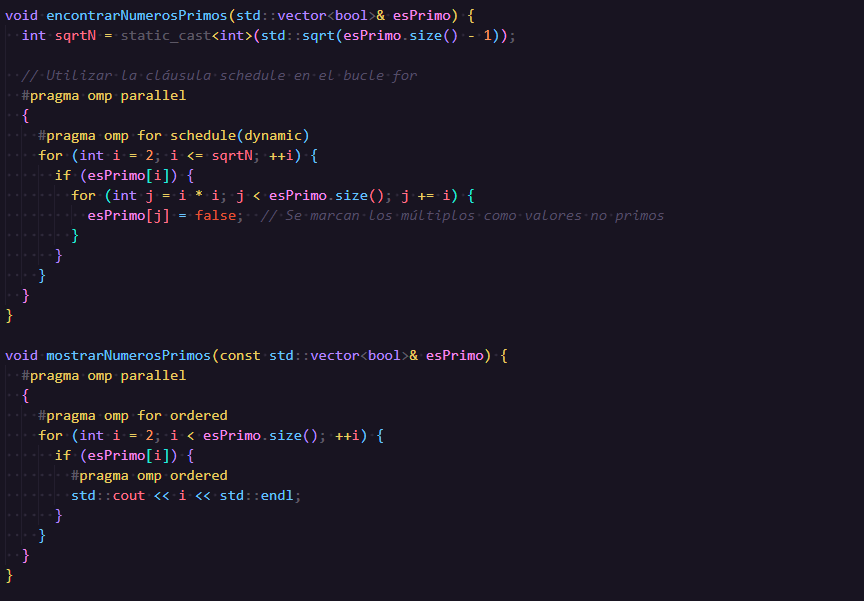


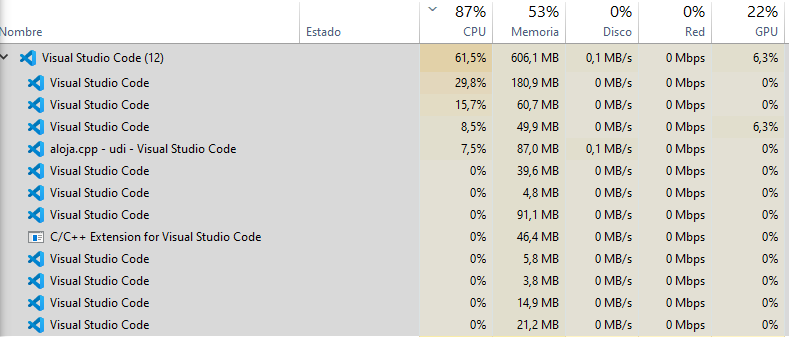
Ilustración algoritmo sin segmentación de código

Por otro lado, sin la implementación de la directiva ordered en el for, lograba mejores resultados en lo que a tiempo se refiere. Esto ya que no se necesita la coordinación de los hilos al entrar en el loop. Sin embargo, si nos enfocamos en los resultados legibles por pantalla esto era imposible, ya que los números se mezclaban al mostrarse y no era posible identificar si se estaba cumpliendo o no el objetivo de encontrar los números primos correctamente.

Inicialmente se implementó la criba de Aretosthenes, utilizando el 80% de los hilos totales, de la siguiente manera:  
Forma

Descripción generada automáticamente con confianza media  




Esto si bien, funcionaba, la carga de CPU era significativa. Donde en ocasiones con aplicaciones de bajo rendimiento abiertas, el consumo del algoritmo era demasiado.  


Buscando otras alternativas se creo una nueva implementación, la cual es en la que se trabajara y que dio resultados favorables, distribuyendo la carga entre la cpu y memoria ram  
  
