

Semana 6: Sincronización

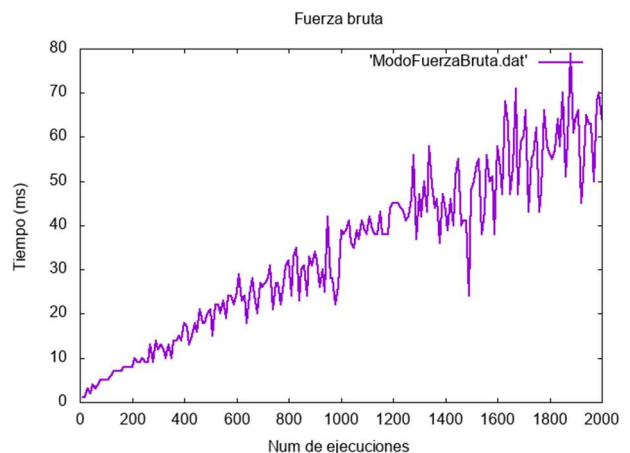
P3 – Método de fuerza bruta

En el primer apartado de la entrega, se reutilizo el código realizado en la semana anterior. Este realizar una serie de N movimientos repartidos en M hilos de tal forma que ejecuten de forma síncrona el método *EnterAndWait()*. Se modifica la restricción para que el único hilo que se pueda ejecutar es el siguiente en orden numérico. Para esta primera parte hacemos esperar a todos los hilos en el objetoA. Tras cada ejecución se despiertan todos los hilos. Cuando intentan ejecutar el método se pone a dormir a todos los hilos con id distinto al que se debe ejecutar, asegurando la restricción propuesta.

El resultado obtenido es el siguiente (Para 100 ejecuciones y 10 hilos):

```
Ejecutando hilo 1
Hilo 1 acabando de ejecutarse. Quedan 99 ejecuciones.
Ejecutando hilo 2
Hilo 2 acabando de ejecutarse. Quedan 98 ejecuciones.
Ejecutando hilo 3
Hilo 7 acabando de ejecutarse. Quedan 3 ejecuciones.
Ejecutando hilo 8
Hilo 8 acabando de ejecutarse. Quedan 2 ejecuciones.
Ejecutando hilo 9
Hilo 9 acabando de ejecutarse. Quedan 1 ejecuciones.
Ejecutando hilo 10
Hilo 10 acabando de ejecutarse. Quedan 0 ejecuciones.
Program of exercise P4 has terminated
```

Para la realización de los *plots* se le eliminan todas las salidas por pantalla, se añade un *timer* y se imprime el número de ejecuciones y el tiempo de ejecución. El fichero resultante se entrega con el nombre *ModoFuerzaBruta* para su posible revisión. Los resultados obtenidos son los siguientes:



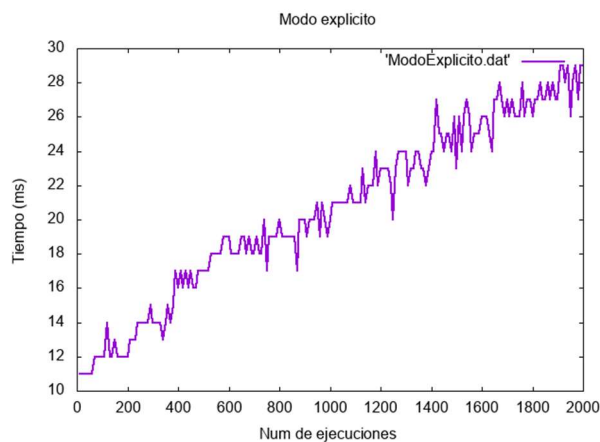
P3 – Método explícito

En el apartado c de la entrega se modifica el código para aumentar la eficiencia. Esta modificación consiste en que cada hilo tenga asignado un enlace al siguiente hilo (nombrado como compañero). Una vez realizado la propia ejecución, cada hilo pasara a estado de espera y llamará a su compañero para que despierte.

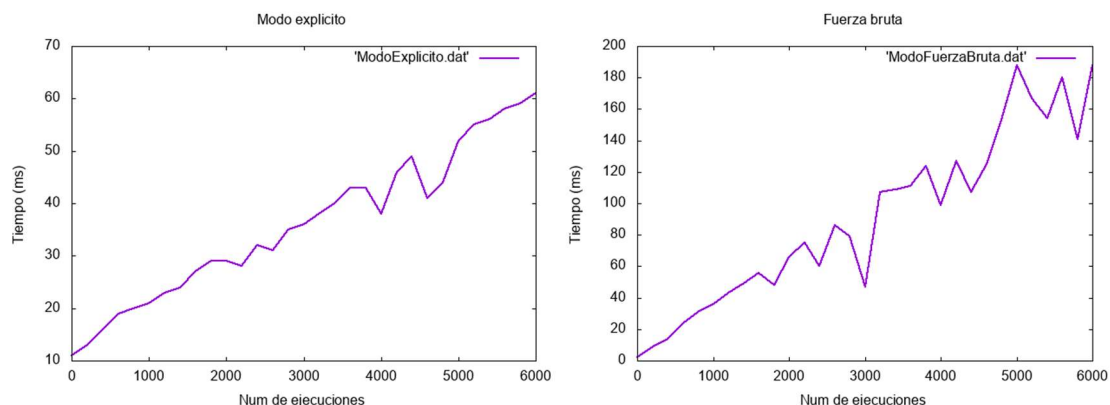
Esta modificación consigue aumentar el rendimiento, ya que no se despiertan todos los hilos en cada ejecución sino únicamente el que tiene que ejecutarse.

Para realizar las pruebas se vuelven a eliminar todas las salidas y reducir los tiempos, sin embargo, en este método necesitamos un tiempo para que arranquen todos los hilos y se pongan en estado de espera por lo que las pruebas tienen un retraso de 10ms.

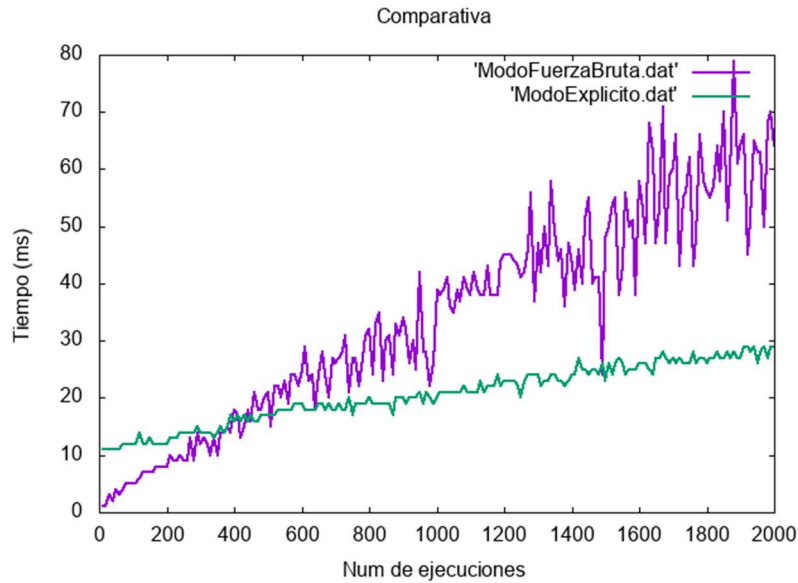
Los resultados obtenidos son los siguientes:



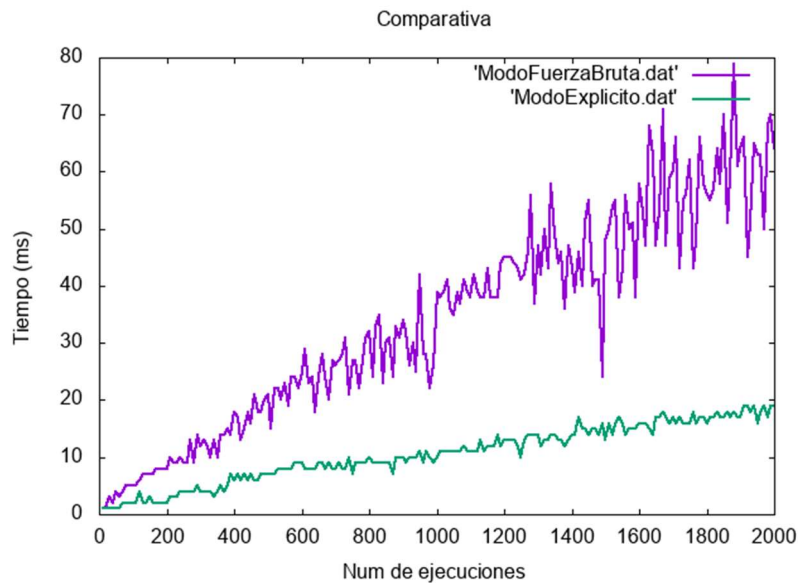
Las pruebas han sido realizadas para un numero de ejecuciones incremental de 10 en 10 hasta 2000. Los resultados se ven muy oscilantes por el bajo rango de incremento y la baja duración de la prueba. Se puede suavizar la oscilación haciendo ejecuciones con un incremento mayor. Por ejemplo, incrementos de 200 ejecuciones:



P3 – Comparar Métodos.



Como se aprecia en el gráfico, el método Explicito o Eficiente, como su propio nombre expresa obtiene un rendimiento muy superior al método de fuerza bruta. Cabe destacar que en un número de ejecuciones pequeño el método Explicito obtiene un rendimiento menor debido a los 10ms de espera que se le conceden para la preparación de los hilos.



Si se le restan 10ms a los tiempos obtenidos con el método explicito se observa mejor el rendimiento de dicho método. Si se compara numéricamente los resultados se llega a la conclusión de que el método nunca es mas lento y llega a un 450% de aumento en la velocidad de ejecución:

Rend máximo	453,33%
Rend mínimo	100,00%