

## TALLER 1B

ALEJANDRO AHOGADO PRIETO

201920701

### Parte 1

1. El resultado esperado es correcto, ya que sería 10.000.000 y si hacemos lo que dice el enunciado de ejecutar 1000 veces un método que incrementa en 10000 el contador, se obtendría el mismo resultado.
2. De acuerdo al enunciado, el valor esperado sería 10.000.000, pero en este caso el resultado no corresponde a ese valor. Esto ocurre por una variable compartida de los threads, por lo que estos no están sincronizados y generan ese error en el resultado.
- 3.

Ejecución	Valor Obtenido
1	9231354
2	9091463
3	9239384
4	9215379
5	9001356

4. La variable contador es una variable compartida, y dado que todos los threads tratan de hacer uso de ella cada vez que se ejecutan el valor de esta cambia (en el run). Cada vez que se ejecuta el programa da un resultado distinto ya que cada vez se ejecutan los threads en un orden aleatorio, causando que cuando se hayan ejecutado los 1000 threads el valor del contador sea diferente al esperado.

### Parte 2

- 1.

Ejecución	Valor Obtenido (Maximo Global)	Valor Esperado (Máximo de los máximos locales)
1	58880	60700
2	88734	103048
3	89778	97504
4	95720	95720
5	97596	83982

**Ejemplo de cómo se llenaron los datos de la tabla:**

ID Thread: 1 - Maximo Local: 83982d - Maximo Global: 83982  
ID Thread: 2 - Maximo Local: 65347d - Maximo Global: 83982  
ID Thread: 0 - Maximo Local: 97596d - Maximo Global: 83982

El máximo global fue 83982, pero si miramos los máximos locales el mayor fue 97596, por lo tanto debería ser el máximo global.

2. Dado que la variable mayor es una variable compartida, existe un acceso concurrente en esta, como por ejemplo cuando se compara con el valor de mayorFila, por lo que cada thread comparara con un valor de mayor distinto y esto causará que se dé un resultado diferente.
3. Si tenemos diferentes threads que van a usar la misma información es muy importante tener en cuenta la sincronización y los protocolos que debemos llevar a cabo para que un programa logre llegar al resultado esperado. Como es el computador el que decide en que orden ejecutar los threads, esto causará que las variables compartidas causen resultados distintos al ser ejecutadas en un orden incierto.