



**Universidad Nacional Autónoma de México**  
FACULTAD DE CIENCIAS

Computación Concurrente

**Tarea 05**

Estructuras de Datos Concurrentes

**Profesor:**

Salvador González Arellano

**Integrantes:**

Contreras Ibarra Francisco

Marín Parra José Guadalupe de Jesús

Ortega González José Ethan

Ramírez López Alvaro

## 1. Teoría:

1. ¿Para que sirve el método Yield? (yield())

**Solución:** Este método sirve para indicarle al hardware que está ejecutando la tarea actual, que puede interrumpirla y darle la tarea a otro thread.

2. ¿Qué es un atributo atómico? (En la biblioteca atomic de Java)

**Solución:** Son variables que ayudan con la creación de algoritmos de bloqueo sin necesidad de haber espera.

3. Ventajas de usar atributos atomicos

**Solución:**

- Son útiles en la utilización de hilos.
- Las variables no afectan su valor original.
- No son interrumpidos por el procesador.

4. Desventajas de usar atributos atómicos

**Solución:** Una de las principales desventajas es que no todos los tipos de datos tienen un equivalente atómico. De manera similar, no existen las mismas operaciones que con el equivalente no atómico. Por ejemplo, tenemos métodos para sacar el módulo de un Integer, pero no para sacar directamente el módulo de un AtomicInteger.

5. Da 2 ejemplos en donde se puedan aplicar este tipo de atributos (no pongan en resolver la tarea plox)

**Solución:**

- Al momento de instanciar variables las cuales no queremos que tengan algún tipo de espera.
- Uso de aplicaciones como contadores que se incrementan de forma automática.

6. Algún uso que creas que se da en estructuras de datos concurrentes.

**Solución:**

Como en las estructuras de datos concurrentes generalmente se considera que reside en un entorno de almacenamiento abstracto llamado memoria compartida, en la cual pueden acceder múltiples subprocesos que pueden acceder a los datos simultáneamente porque se ejecutan en diferentes procesadores que se comunican entre sí por lo que los atributos atómicos permite la gestión de quien puede acceder al recurso compartido de manera segura y es libre de interbloqueos.

7. Además a eso, quiere ver como se comportan los hilos, por lo que te pide que le ayudes escribiendo las historias de ejecución del siguiente bloque de código:

```

public class Counter implements Runnable {
    public static final int ROUNDS = 5;
    private int counter = 0;

    @Override
    public void run() {
        for(int i=0; i< Counter.ROUNDS; i++) {
            counter++;
        }
    }
}

```

Muestra historias donde se puedan obtener los siguientes valores con la cantidad de hilos 2, 4 y 5 hilos, en caso de que no se pueda, explica el porque no:

- 0
- 2
- 3
- 5
- 10
- 13
- 15
- 20
- 26

### Solución:

- Para el caso de los valores que se obtienen haciendo la ejecución con 2 hilos recreamos 2 historias, una donde el valor máximo del counter es 5 y otra donde el valor máximo puede ser 10, estas abarcan el mínimo y máximo que counter puede tener, podemos ver que en este caso los valores que se pueden obtener son: **0, 2, 3, 5 y 10.**

Hilo 1	Hilo 2	Ejecución (Counter = )
<u>I</u>	<u>I</u>	0
Hilo 1: Lee contador ( <b>contador = 0</b> )	Hilo 2: Lee contador	0
Hilo 1: Escribe en contador		1
	Hilo 2: Escribe en Contador	1
Hilo 1: Lee contador	Hilo 2: Lee contador	1
Hilo 1: Escribe en contador		2
	Hilo 2: Escribe en Contador	2
Hilo 1: Lee contador	Hilo 2: Lee contador	2
Hilo 1: Escribe en contador		3
	Hilo 2: Escribe en Contador	3
Hilo 1: Lee contador	Hilo 2: Lee contador	3
Hilo 1: Escribe en contador		4
	Hilo 2: Escribe en Contador	4
Hilo 1: Lee contador	Hilo 2: Lee contador	4
Hilo 1: Escribe en contador		5
<b>T</b>	Hilo 2: Escribe en Contador	5
	<b>T</b>	5

Historia con el valor mínimo del counter con 2 hilos.

Hilo 1	Hilo 2	Ejecución (Counter = )
<b>I</b>	<b>I</b>	0
Hilo 1: Lee contador ( <b>contador = 0</b> )		0
Hilo 1: Escribe en contador		1
Hilo 1: Lee contador		1
Hilo 1: Escribe en contador		2
Hilo 1: Lee contador		2
Hilo 1: Escribe en contador		3
Hilo 1: Lee contador		3
Hilo 1: Escribe en contador		4
Hilo 1: Lee contador		4
Hilo 1: Escribe en contador		5
<b>T</b>	Hilo 2: Lee contador	5
	Hilo 2: Escribe en Contador	6
	Hilo 2: Lee contador	6
	Hilo 2: Escribe en Contador	7
	Hilo 2: Lee contador	7
	Hilo 2: Escribe en Contador	8
	Hilo 2: Lee contador	8
	Hilo 2: Escribe en Contador	9
	Hilo 2: Lee contador	9
	Hilo 2: Escribe en Contador	10
	<b>T</b>	<b>10</b>

Historia con el valor máximo del counter con 2 hilos.

- Para el caso de los valores que se obtienen haciendo la ejecución con 4 hilos se recreo una historia donde un hilo se ejecuta y este mantiene el recurso ocupado, es decir hace su proceso y hasta que se desocupa otro proceso lo modifica, en este caso los valores que se pueden obtener son: **0, 2, 3, 5, 10, 13, 15 y 20**.

Hilo 1	Hilo 2	Hilo 3	Hilo 4	Ejecución (Counter = )
<b>I</b>	<b>I</b>	<b>I</b>	<b>I</b>	0
		Hilo 3: Lee contador ( <b>contador = 0</b> )		0
		Hilo 3: Escribe en contador		1
		Hilo 3: Lee contador		1
		Hilo 3: Escribe en contador		2
		Hilo 3: Lee contador		2
		Hilo 3: Escribe en contador		3
		Hilo 3: Lee contador		3
		Hilo 3: Escribe en contador		4
		Hilo 3: Lee contador		4
		Hilo 3: Escribe en contador		5
	Hilo 2: Lee contador			5
	Hilo 2: Escribe en Contador			6
	Hilo 2: Lee contador			6
	Hilo 2: Escribe en Contador			7
	Hilo 2: Lee contador			7
	Hilo 2: Escribe en Contador			8
	Hilo 2: Lee contador			8
	Hilo 2: Escribe en Contador			9
	Hilo 2: Lee contador			9
	Hilo 2: Escribe en Contador			10
Hilo 1: Lee contador	<b>T</b>			10
Hilo 1: Escribe en contador				11
Hilo 1: Lee contador				11
Hilo 1: Escribe en contador				12
Hilo 1: Lee contador				12
Hilo 1: Escribe en contador				13
Hilo 1: Lee contador				13
Hilo 1: Escribe en contador				14
Hilo 1: Lee contador				14
Hilo 1: Escribe en contador				15
<b>T</b>			Hilo 4: Lee contador	15
			Hilo 4: Escribe en contador	16
			Hilo 4: Lee contador	16
			Hilo 4: Escribe en contador	17
			Hilo 4: Lee contador	17
			Hilo 4: Escribe en contador	18
			Hilo 4: Lee contador	18
			Hilo 4: Escribe en contador	19
			Hilo 4: Lee contador	19
			Hilo 4: Escribe en contador	20
			<b>T</b>	<b>20</b>

Historia con el valor máximo del counter con 4 hilos.

- Para el caso de los valores que se obtienen haciendo la ejecución con 5 hilos se recreo una historia donde un hilo se ejecuta y este mantiene el recurso ocupado, es decir hace su proceso y hasta que se desocupa otro proceso lo modifica, en este caso los valores que se pueden obtener son: **0, 2, 3, 5, 10, 13, 15 y 20**, el valor 26 no es posible obtenerlo porque se sale fuera del rango del for.

Hilo 1	Hilo 2	Hilo 3	Hilo 4	Hilo 5	Ejecución (Counter=)
1	1	1	1	1	0
		Hilo 3: Lee contador (contador=0)			0
		Hilo 3: Escribe en contador			1
		Hilo 3: Lee contador			1
		Hilo 3: Escribe en contador			2
		Hilo 3: Lee contador			2
		Hilo 3: Escribe en contador			3
		Hilo 3: Lee contador			3
		Hilo 3: Escribe en contador			4
		Hilo 3: Lee contador			4
		Hilo 3: Escribe en contador			5
	Hilo 2: Lee contador				5
	Hilo 2: Escribe en Contador				6
	Hilo 2: Lee contador				6
	Hilo 2: Escribe en Contador				7
	Hilo 2: Lee contador				7
	Hilo 2: Escribe en Contador				8
	Hilo 2: Lee contador				8
	Hilo 2: Escribe en Contador				9
	Hilo 2: Lee contador				9
	Hilo 2: Escribe en Contador				10
Hilo 1: Lee contador	T				10
Hilo 1: Escribe en contador					11
Hilo 1: Lee contador					11
Hilo 1: Escribe en contador					12
Hilo 1: Lee contador					12
Hilo 1: Escribe en contador					13
Hilo 1: Lee contador					13
Hilo 1: Escribe en contador					14
Hilo 1: Lee contador					14
Hilo 1: Escribe en contador					15
T			Hilo 4: Lee contador		15
			Hilo 4: Escribe en contador		16
			Hilo 4: Lee contador		16
			Hilo 4: Escribe en contador		17
			Hilo 4: Lee contador		17
			Hilo 4: Escribe en contador		18
			Hilo 4: Lee contador		18
			Hilo 4: Escribe en contador		19
			Hilo 4: Lee contador		19
			Hilo 4: Escribe en contador		20
			T	Hilo 5: Lee contador	20
				Hilo 5: Escribe en contador	21
				Hilo 5: Lee contador	21
				Hilo 5: Escribe en contador	22
				Hilo 5: Lee contador	22
				Hilo 5: Escribe en contador	23
				Hilo 5: Lee contador	23
				Hilo 5: Escribe en contador	24
				Hilo 5: Lee contador	24
				Hilo 5: Escribe en contador	25
				T	25

Historia con el valor máximo del counter con 5 hilos.

## 2. Práctica:

Las PCs son las siguientes:

PC	Integrante	Procesador	RAM	Sistema Operativo
PC1	Francisco	Intel i7-9750H	16GB	Ubuntu 20.04.5
PC2	José	AMD Ryzen 5 2600	16 GB	Linux Mint
PC3	Ethan	Intel i5-4300M	8 GB	Arch Linux
PC4	Álvaro	Intel Xeon E5-2666 v3	32 GB	Windows 10 (WSL 2)

La tabla con los tiempos es la siguiente:

### ■ 2 hilos:

Pruebas	Tiempo de la prueba				Temperatura antes de				Temperatura después de			
	PC1	PC2	PC3	PC4	PC1	PC2	PC3	PC4	PC1	PC2	PC3	PC4
TASLock	4.33s	11.86s	7.44s	14.91s	45	40.6	48	40	55	41.8	63	42
TTASLock	2.20s	13.32s	5.04s	17.66s	44	41.0	48	41	50	42.2	57	44
BackoffLock	2.25s	12.16s	5.40s	12.98s	46	42.0	47	40	51	42.1	60	43
CLHLock	-	+30min	-	-	-	41.0	47	-	-	46.0	79	-
MCSLock	17.38s	16.71s	5.58s	39.91s	46	42.0	52	40	57	43.6	64	52
ALock	-	-	-	-	-	-	-	-	-	-	-	-

### ■ 3 hilos:

Pruebas	Tiempo de la prueba				Temperatura antes de				Temperatura después de			
	PC1	PC2	PC3	PC4	PC1	PC2	PC3	PC4	PC1	PC2	PC3	PC4
TASLock	6.05s	14.82s	6.76s	20.96s	46	42.0	48	48	54	42.4	63	50
TTASLock	3.13	20.05s	6.40s	20.67s	46	41.2	47	48	53	43.5	69	52
BackoffLock	3.52s	13.15s	5.58s	13.80s	45	40.6	45	44	51	41.5	61	49
CLHLock	-	+30min	-	-	-	40.8	-	-	-	49.2	-	-
MCSLock	18.12	31.83s	26.48s	56.46s	47	43.4	45	43	57	49.6	55	47
ALock	-	-	-	-	-	-	-	-	-	-	-	-

### ■ 7 hilos:

Pruebas	Tiempo de la prueba				Temperatura antes de				Temperatura después de			
	PC1	PC2	PC3	PC4	PC1	PC2	PC3	PC4	PC1	PC2	PC3	PC4
TASLock	9.32s	42.93s	8.86s	58.09s	46	43.1	48	44	55	44.2	61	48
TTASLock	2.202s	66.45s	10.88s	63.59s	47	43.2	47	44	50	46.6	53	50
BackoffLock	3.25	18.12s	5.70s	14.24s	46	44.2	46	45	51	44.2	52	49
CLHLock	-	+30min	-	-	-	43.2	-	-	-	56.5	-	-
MCSLock	26.44	48.47s	82.89s	116.73s	46	43.1	47	49	59	57.0	60	57
ALock	-	-	-	-	-	-	-	-	-	-	-	-

### ■ 15 hilos:

Pruebas	Tiempo de la prueba				Temperatura antes de				Temperatura después de			
	PC1	PC2	PC3	PC4	PC1	PC2	PC3	PC4	PC1	PC2	PC3	PC4
TASLock	16.02s	54.2s	9.48s	192.58s	43	44.0	40	42	57	46.0	46	49
TTASLock	11.99s	31.5s	14.10s	196.87s	45	43.0	45	44	56	48.8	56	54
BackoffLock	4.75	13.45s	14.11s	14.23s	45	43.2	49	42	54	44.1	53	48
CLHLock	-	+30min	-	-	-	47.1	-	-	-	57.2	-	-
MCSLock	+30min	136.9s	159.18	312.52s	52	42.4	47	47	64	66.2	58	63
ALock	-	-	-	-	-	-	-	-	-	-	-	-

## ■ 21 hilos:

Pruebas	Tiempo de la prueba				Temperatura antes de				Temperatura después de			
	PC1	PC2	PC3	PC4	PC1	PC2	PC3	PC4	PC1	PC2	PC3	PC4
TASLock	21.15s	206.9s	8.03s	443.04s	47	42.6	49	46	53	46.4	62	55
TTASLock	12.56s	292.7s	7.44s	306.31s	50	43.8	46	46	55	50.5	59	55
BackoffLock	5.26s	12.96s	6.53s	14.59s	51	45.6	47	46	54	45.6	57	49
CLHLock	-	+30min	-	-	-	46.8	-	-	-	60.5	-	-
MCSLock	+30min	193.5s	219.64s	1377.33s	53	44.5	45	44	63	67.5	65	74
ALock	-	-	-	-	-	-	-	-	-	-	-	-

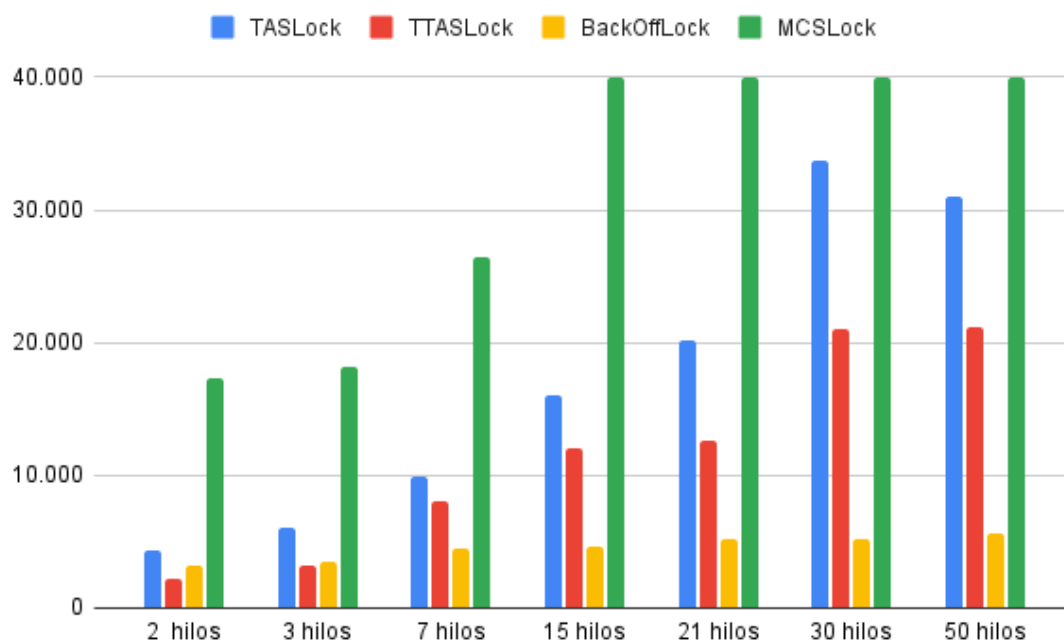
## ■ 30 hilos:

Pruebas	Tiempo de la prueba				Temperatura antes de				Temperatura después de			
	PC1	PC2	PC3	PC4	PC1	PC2	PC3	PC4	PC1	PC2	PC3	PC4
TASLock	33.65s	43.3s	221.30s	599.98s	50	46.0	36	46	57	49.6	56	63
TTASLock	21.07s	45.6s	9.10s	344.60s	48	45.5	47	53	54	51.1	50	65
BackoffLock	5.26	13.87s	7.03s	13.80s	49	43.5	43	51	52	46.8	47	55
CLHLock	-	+30min	-	-	-	47.2	-	-	-	62.1	-	-
MCSLock	+30min	246.1s	308.61s	2070.67s	47	43.5	45	48	62	66.8	60	80
ALock	-	-	-	-	-	-	-	-	-	-	-	-

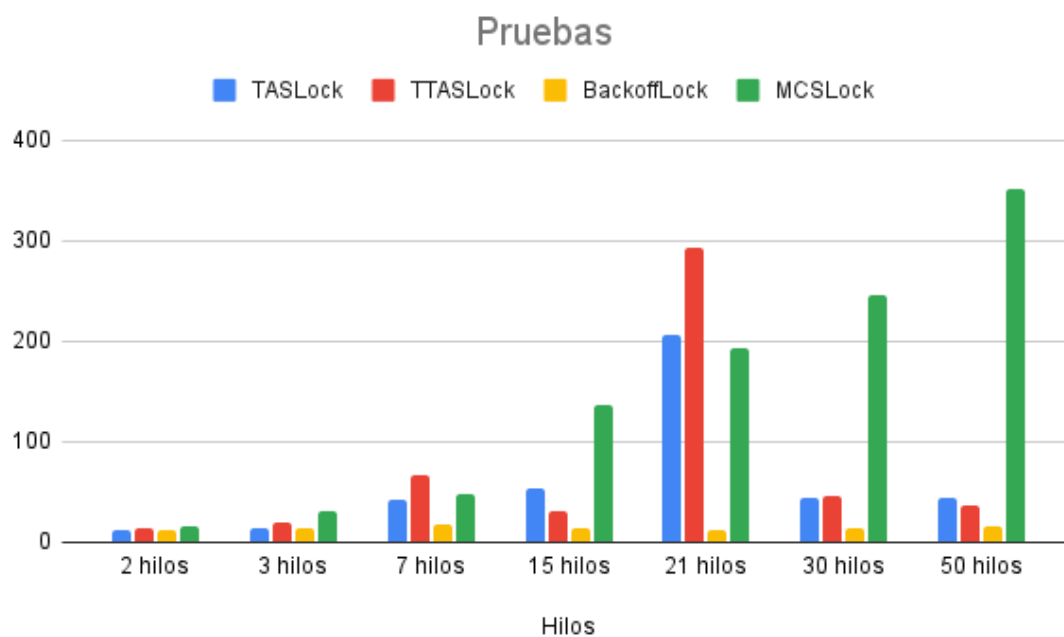
## ■ 50 hilos:

Pruebas	Tiempo de la prueba				Temperatura antes de				Temperatura después de			
	PC1	PC2	PC3	PC4	PC1	PC2	PC3	PC4	PC1	PC2	PC3	PC4
TASLock	31.07s	44.8s	8.58s	694.10s	50	48.1	47	47	58	48.8	50	62
TTASLock	21.17s	35.9s	8.68s	315.01s	53	46.2	48	50	58	50.5	48	65
BackoffLock	5.16	16.09s	8.79s	17.35s	51	42.4	47	50	54	44.5	51	53
CLHLock	-	+30min	-	-	-	47.6	-	-	-	62.0	-	-
MCSLock	+30min	350.6s	510.65s	4217.68s	52	44.0	48	50	65	62.0	60	92
ALock	-	-	-	-	-	-	-	-	-	-	-	-

## Gráficas.

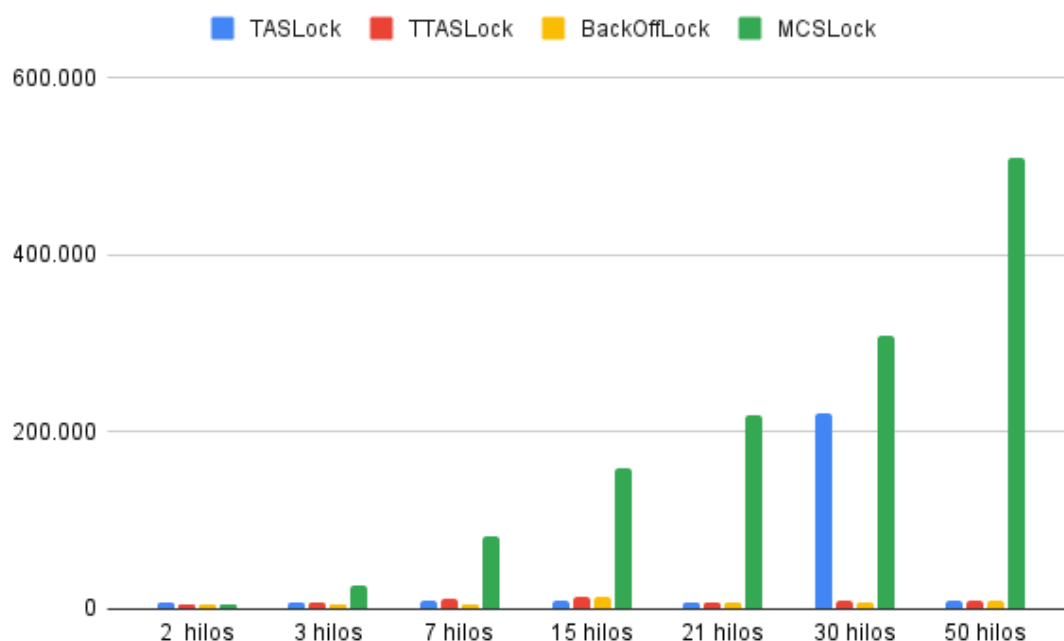


Gráfica Francisco.

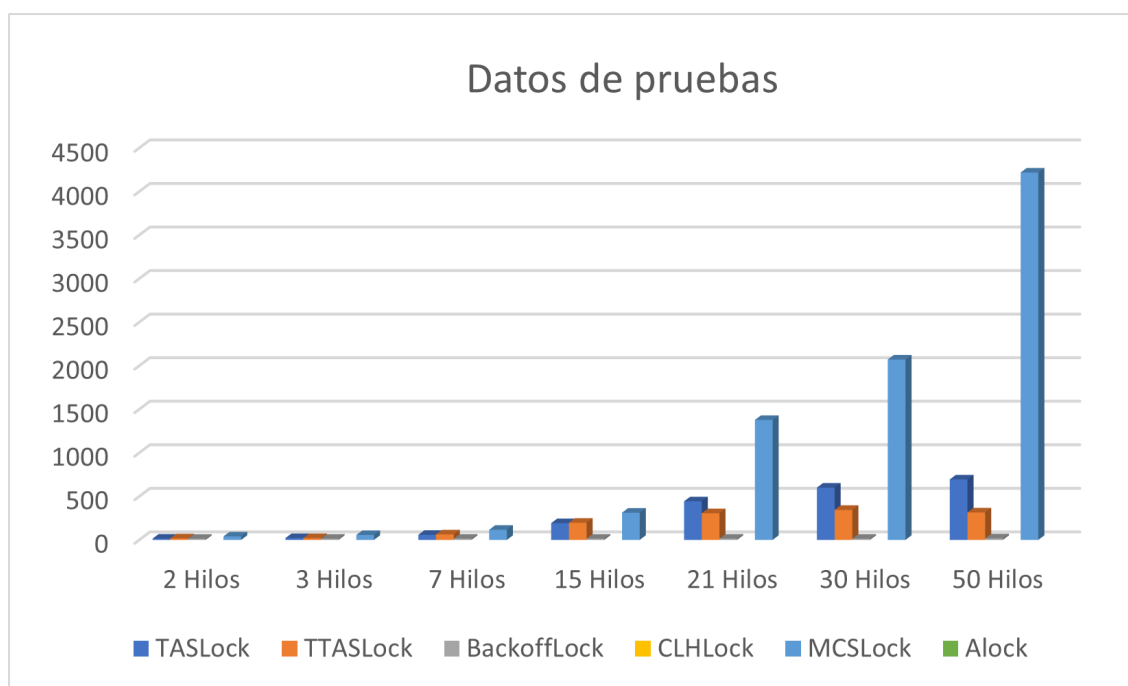


Gráfica José.





Gráfica Ethan.



Gráfica Álvaro.

### 3. Referencias:

- Ayuda. (s/f). Upm.es. Recuperado el 23 de octubre de 2022, de <https://www.dit.upm.es/pepe/libros/concurrency/index.html>
- Castelnovo, P. (2018, abril 3). Variables volátiles en Java. Medium. <https://medium.com/@pablocastelnovo/variables-vol%C3%A1tiles-en-java-f5ae078bf8b9>
- tok.wiki. (s. f.). Estructura de datos concurrentes Principios básicos y Diseño e implementación. Recuperado 23 de octubre de 2022, de [https://hmong.es/wiki/Concurrent\\_data\\_structure](https://hmong.es/wiki/Concurrent_data_structure)

- Atomic Variables (The Java™ Tutorials ¿Essential Java Classes ¿Concurrency). (s. f.). Recuperado 23 de octubre de 2022, de <https://docs.oracle.com/javase/tutorial/essential/concurrency/atomicvars.html>