



Escuela de Ciencia de la Computación  
Universidad Nacional de San Agustín Arequipa

Milagros Celia Cruz Mamani  
COMPUTACIÓN PARALELA Y DISTRIBUIDA  
**Laboratorio 3**

<https://github.com/milagroscm/COMPUATACIO-PARALELA>

## Trabajo de PThreads 1

### Resumen

El alumno debe realizar el informe en formato artículo donde la implementación, resultados y análisis de la ejecución para los siguientes problemas.

## 1. Ejercicios

### 1.1. Hardware donde se realizó las pruebas

Las pruebas de los algoritmos se realizaron en ubuntu

```
workspace $ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:             Little Endian
CPU(s):                 8
On-line CPU(s) list:   0-7
Thread(s) per core:    2
Core(s) per socket:    4
Socket(s):              1
NUMA node(s):          1
Vendor ID:              GenuineIntel
CPU family:             6
Model:                  79
Model name:             Intel(R) Xeon(R) CPU @ 2.20GHz
Stepping:               0
CPU MHz:                2200.000
BogoMIPS:               4400.00
Hypervisor vendor:     KVM
Virtualization type:    full
L1d cache:              32K
L1i cache:              32K
L2 cache:               256K
```

Figura 1: Características de mi ubuntu

### 1.2. Implementar y comparar las técnicas de sincronización Busy-waiting y Mutex, obtener una tabla similar a la Tabla 4.1 del libro.

- Para compilar `mpicc -o pth-pi-bw pth-pi.c -lm`

- Para ejecutar `./pth-pi-bw <numero de hilos><n>`

```

Lab3 $ ./pth-pi-bw 1 100000000
3.141593
Tiempo->1.209830
Lab3 $ ./pth-pi-bw 2 100000000
3.141593
Tiempo->0.697879
Lab3 $ ./pth-pi-bw 4 100000000
3.141593
Tiempo->0.786940
Lab3 $ ./pth-pi-bw 8 100000000
3.141593
Tiempo->0.897071
Lab3 $ ./pth-pi-bw 8 100000000
3.141593
Tiempo->0.635424
Lab3 $ ./pth-pi-bw 16 100000000
3.141593
Tiempo->0.941759
Lab3 $ ./pth-pi-bw 32 100000000
3.141593
Tiempo->1.411979
Lab3 $ ./pth-pi-bw 64 100000000

```

```

Lab3 $ ./pth-pi-mutex 1 100000000
3.141593
Tiempo->1.552016
Lab3 $ ./pth-pi-mutex 2 100000000
3.141593
Tiempo->0.790923
Lab3 $ ./pth-pi-mutex 4 100000000
3.141593
Tiempo->0.816084
Lab3 $ ./pth-pi-mutex 8 100000000
3.141593
Tiempo->0.777936
Lab3 $ ./pth-pi-mutex 16 100000000
3.141593
Tiempo->0.685544
Lab3 $ ./pth-pi-mutex 32 100000000
3.141593
Tiempo->0.707258
Lab3 $ ./pth-pi-mutex 64 100000000
3.141593
Tiempo->0.845720

```

(a) Capturas de pantalla con el uso de Busy-waiting

(b) Capturas de pantalla con el uso de Mutex

### 1.2.1. Resultados comparativos

Thread	Busy-Wait	Mutex
1	1.20983	1.552016
2	0.697879	0.790923
4	0.78694	0.816084
8	0.897071	0.777936
16	0.941759	0.685544
32	1.411979	0.707258
64	4.70559	0.84572

Figura 2: Tablas de tiempo Busy-waiting vs Mutex

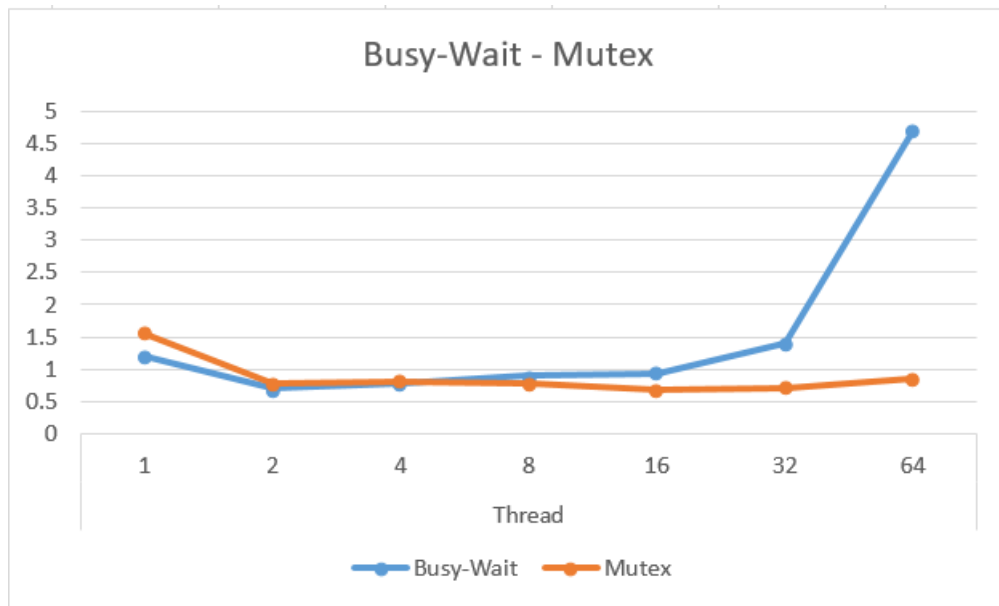


Figura 3: Cuadro de comparación de tiempo Busy-waiting vs Mutex

#### 1.2.2. Análisis:

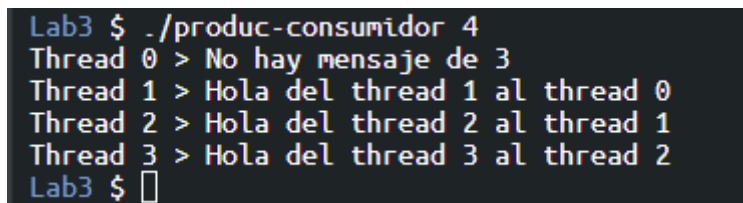
Podemos ver que en Busy-waiting vs Mutex podemos ver como el algoritmo Busy-waiting al inicio mantiene su tiempo menor al tiempo de mutex pero mientras aumenta mas el numero de threads va ir aumentando su tiempo y crece de gran manera.

En cambio con el uso de Mutex, estos tiempos se mantienen en un rango y no suben tan drásticamente como podemos ver en el grafico de esta comparación por ende concluimos que Mutex es mas eficiente.

### 1.3. Basado en la sección 4.7, implementar un ejemplo de productor-consumidor. Explicar porque no se debe utilizar MUTEX.

#### Productor - Consumidor

- Hay un hilo productor y hay un hilo consumidor
- El hilo productor genera datos que serán utilizados por el hilo consumidor.
- Los datos se almacenan en bloques de memoria compartida entre el productor y el consumidor.
- Este tipo de procesos requiere sincronización ya que si el consumidor utiliza los datos mas rápido que el productor, entonces el consumidor se tendría que estar deteniendo
- Si el productor genera datos mas rápido que la utilización del consumidor, se podrían generar sobre-escritura de buffers de memoria.
- Este modelo también es conocido como Cliente-Servidor



```
Lab3 $ ./produc-consumidor 4
Thread 0 > No hay mensaje de 3
Thread 1 > Hola del thread 1 al thread 0
Thread 2 > Hola del thread 2 al thread 1
Thread 3 > Hola del thread 3 al thread 2
Lab3 $
```

Figura 4: Productor-Consumidor con 5 hilos

#### 1.3.1. Análisis:

Como se puede ver, el productor solo produce en este caso procesos, y mediante el ejemplo en donde el productor crea un mensaje y el consumidor procesa ese mensaje se pudo observar que hay un problema a la hora de paralelizarlo es que aveces había procesos de consumidores que se disparaba mas rápido que el mismo productor , y usando mutex se generada un punto muerto

### 1.4. Implementar y explicar las diferentes formas de barreras en PThreads mostradas en el libro.

#### Barreras Pthreads

- `pthread_barrier_init(barrier, attr, n)` Aqui se inicializa la barrera `barrier` con los atributos `attr` para que funcione con `n` hebras.
- `pthread_barrier_t` Tipo de brrera
- `pthread_barrier_wait(barrier)` Bloquea a la hebra llamadora hasta que `n` hebras ejecuten `pthread_barrier_wait(barrier)`
- `pthread_barrier_destroy(barrier)` Destruye la barrera `barrier`

#### 1.4.1. Capturas de pantalla de la ejecución de los códigos

```
Lab3 $ mpicc -o barrera150-pth pth_barreras.c -lpthread
Lab3 $ mpirun -np 1 barrera150-pth 1
Elapsed time = 2.200603e-04 seconds
Lab3 $ mpirun -np 1 barrera150-pth 2
Elapsed time = 6.729126e-03 seconds
Lab3 $ mpirun -np 1 barrera150-pth 3
Elapsed time = 9.931803e-03 seconds
Lab3 $ mpirun -np 1 barrera150-pth 4
Elapsed time = 9.174109e-03 seconds
Lab3 $ mpirun -np 1 barrera150-pth 8
Elapsed time = 2.956510e-02 seconds
Lab3 $ mpirun -np 1 barrera150-pth 16
Elapsed time = 2.626395e-02 seconds
Lab3 $ mpirun -np 1 barrera150-pth 32
Elapsed time = 9.348989e-02 seconds
Lab3 $ mpirun -np 1 barrera150-pth 64
Elapsed time = 7.587600e-02 seconds
```

(a) BARRIER\_COUNT 150

```
Lab3 $ mpicc -o barrera300-pth pth_barreras.c -lpthread
Lab3 $ mpirun -np 1 barrera300-pth 1
Elapsed time = 8.008480e-04 seconds
Lab3 $ mpirun -np 1 barrera300-pth 2
Elapsed time = 9.064913e-03 seconds
Lab3 $ mpirun -np 1 barrera300-pth 4
Elapsed time = 1.297307e-02 seconds
Lab3 $ mpirun -np 1 barrera300-pth 8
Elapsed time = 3.944612e-02 seconds
Lab3 $ mpirun -np 1 barrera300-pth 16
Elapsed time = 7.512093e-02 seconds
Lab3 $ mpirun -np 1 barrera300-pth 32
Elapsed time = 1.217830e-01 seconds
Lab3 $ mpirun -np 1 barrera300-pth 64
Elapsed time = 3.437879e-01 seconds
```

(b) BARRIER\_COUNT 300

```
Lab3 $ mpicc -o barrera450-pth pth_barreras.c -lpthread
Lab3 $ mpirun -np 1 barrera450-pth 1
Elapsed time = 1.001835e-03 seconds
Lab3 $ mpirun -np 1 barrera450-pth 2
Elapsed time = 5.809069e-03 seconds
Lab3 $ mpirun -np 1 barrera450-pth 4
Elapsed time = 1.981592e-02 seconds
Lab3 $ mpirun -np 1 barrera450-pth 8
Elapsed time = 3.202701e-02 seconds
Lab3 $ mpirun -np 1 barrera450-pth 16
Elapsed time = 9.166598e-02 seconds
Lab3 $ mpirun -np 1 barrera450-pth 32
Elapsed time = 2.325270e-01 seconds
Lab3 $ mpirun -np 1 barrera450-pth 64
Elapsed time = 2.151239e-01 seconds
```

(c) BARRIER\_COUNT 450

```
Lab3 $ mpicc -o barrera600-pth pth_barreras.c -lpthread
Lab3 $ mpirun -np 1 barrera600-pth 1
Elapsed time = 7.150173e-04 seconds
Lab3 $ mpirun -np 1 barrera600-pth 2
Elapsed time = 1.916790e-02 seconds
Lab3 $ mpirun -np 1 barrera600-pth 4
Elapsed time = 4.782581e-02 seconds
Lab3 $ mpirun -np 1 barrera600-pth 8
Elapsed time = 4.711413e-02 seconds
Lab3 $ mpirun -np 1 barrera600-pth 16
Elapsed time = 8.307600e-02 seconds
Lab3 $ mpirun -np 1 barrera600-pth 32
Elapsed time = 1.786270e-01 seconds
Lab3 $ mpirun -np 1 barrera600-pth 64
Elapsed time = 2.723980e-01 seconds
```

(d) BARRIER\_COUNT 600

```
Lab3 $ mpicc -o barrera750-pth pth_barreras.c -lpthread
Lab3 $ mpirun -np 1 barrera750-pth 1
Elapsed time = 8.640289e-04 seconds
Lab3 $ mpirun -np 1 barrera750-pth 2
Elapsed time = 5.208969e-03 seconds
Lab3 $ mpirun -np 1 barrera750-pth 4
Elapsed time = 4.221082e-02 seconds
Lab3 $ mpirun -np 1 barrera750-pth 8
Elapsed time = 3.279114e-02 seconds
Lab3 $ mpirun -np 1 barrera750-pth 16
Elapsed time = 1.208611e-01 seconds
Lab3 $ mpirun -np 1 barrera750-pth 32
Elapsed time = 2.776721e-01 seconds
Lab3 $ mpirun -np 1 barrera750-pth 64
Elapsed time = 4.297159e-01 seconds
```

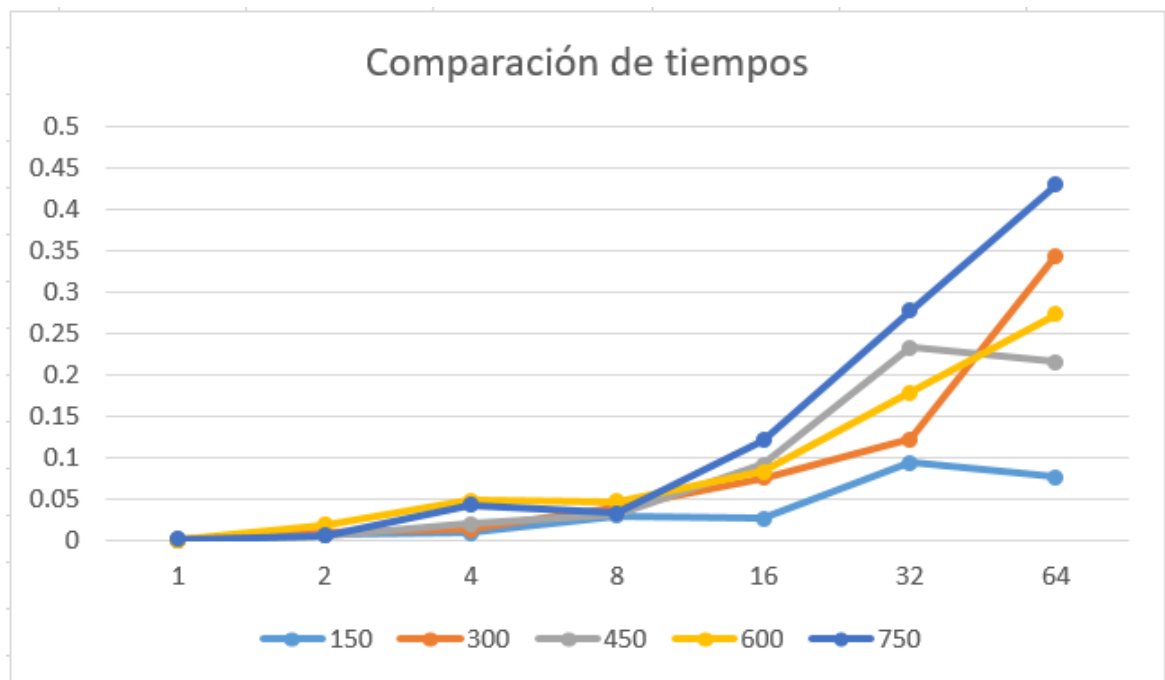
(e) BARRIER\_COUNT 750

Figura 5: Compilación de barreras con Pthreads.

#### 1.4.2. Resultados Comparativos Pthread

Procesos 1					
Thread	BARRIER_COUNT				
	150	300	450	600	750
1	0.00022006	0.0008008	0.0010018	0.000715	0.000864
2	0.006729126	0.0090649	0.0058091	0.0191679	0.005209
4	0.009174109	0.0129731	0.0198159	0.0478258	0.0422108
8	0.0295651	0.0394461	0.032027	0.0471141	0.0327911
16	0.02626395	0.0751209	0.091666	0.083076	0.1208611
32	0.09348989	0.121783	0.232527	0.178627	0.2776721
64	0.075876	0.3437879	0.2151239	0.272398	0.4297159

(a) Barrera con Pthread



(b) Cuadro comparativo

Figura 6: Resultados gráficos

```

Lab3 $ mpicc -o barrera-mutex-150 pth_barre_mutex.c -lpthread
Lab3 $ mpirun -np 1 barrera-mutex-150 1
Elapsed time = 4.618168e-04 seconds
Lab3 $ mpirun -np 1 barrera-mutex-150 2
Elapsed time = 7.170448e-01 seconds
Lab3 $ mpirun -np 1 barrera-mutex-150 4
Elapsed time = 2.951136e+00 seconds
Lab3 $ mpirun -np 1 barrera-mutex-150 8
Elapsed time = 4.587369e+00 seconds
Lab3 $ mpirun -np 1 barrera-mutex-150 16
Elapsed time = 7.152640e+00 seconds
Lab3 $ mpirun -np 1 barrera-mutex-150 32
Elapsed time = 1.561842e+01 seconds
Lab3 $ mpirun -np 1 barrera-mutex-150 64
Elapsed time = 3.692263e+01 seconds

```

(a) BARRIER\_COUNT 150

```

Lab3 $ mpicc -o barrera-mutex-300 pth_barre_mutex.c -lp
Lab3 $ mpirun -np 1 barrera-mutex-300 1
Elapsed time = 1.080036e-04 seconds
Lab3 $ mpirun -np 1 barrera-mutex-300 2
Elapsed time = 6.818771e-04 seconds
Lab3 $ mpirun -np 1 barrera-mutex-300 4
Elapsed time = 5.702431e+00 seconds
Lab3 $ mpirun -np 1 barrera-mutex-300 8
Elapsed time = 9.111765e+00 seconds
Lab3 $ mpirun -np 1 barrera-mutex-300 16
Elapsed time = 1.321956e+01 seconds
Lab3 $ mpirun -np 1 barrera-mutex-300 32
Elapsed time = 2.919284e+01 seconds
Lab3 $ mpirun -np 1 barrera-mutex-300 64
Elapsed time = 6.779855e+01 seconds

```

(b) BARRIER\_COUNT 300

```

Lab3 $ mpicc -o barrera-mutex-450 pth_barre_mutex.c -lpthread
Lab3 $ mpirun -np 1 barrera-mutex-450 1
Elapsed time = 4.739761e-04 seconds
Lab3 $ mpirun -np 1 barrera-mutex-450 2
Elapsed time = 1.303201e-01 seconds
Lab3 $ mpirun -np 1 barrera-mutex-450 4
Elapsed time = 1.141113e+01 seconds
Lab3 $ mpirun -np 1 barrera-mutex-450 8
Elapsed time = 2.357294e+01 seconds
Lab3 $ mpirun -np 1 barrera-mutex-450 16
Elapsed time = 3.461493e+01 seconds
Lab3 $ mpirun -np 1 barrera-mutex-450 32
Elapsed time = 9.900186e+01 seconds
Lab3 $ mpirun -np 1 barrera-mutex-450 64

```

(c) BARRIER\_COUNT 450

```

Lab3 $ mpirun -np 1 barrera-mutex-600 1
Elapsed time = 1.389980e-04 seconds
Lab3 $ mpirun -np 1 barrera-mutex-600 2
Elapsed time = 2.151513e-02 seconds
Lab3 $ mpirun -np 1 barrera-mutex-600 4
Elapsed time = 9.739628e+00 seconds
Lab3 $ mpirun -np 1 barrera-mutex-600 8
Elapsed time = 1.657961e+01 seconds
Lab3 $ mpirun -np 1 barrera-mutex-600 16
Elapsed time = 2.736131e+01 seconds
Lab3 $ mpirun -np 1 barrera-mutex-600 32
Elapsed time = 5.347492e+01 seconds
Lab3 $ mpirun -np 1 barrera-mutex-600 64
Elapsed time = 1.230497e+02 seconds

```

(d) BARRIER\_COUNT 600

```

Lab3 $ mpicc -o barrera-mutex-750 pth_barre_mutex.c -lpthread
Lab3 $ mpirun -np 1 barrera-mutex-750 1
Elapsed time = 1.320839e-04 seconds
Lab3 $ mpirun -np 1 barrera-mutex-750 2
Elapsed time = 3.604698e-02 seconds
Lab3 $ mpirun -np 1 barrera-mutex-750 4
Elapsed time = 1.326402e+01 seconds
Lab3 $ mpirun -np 1 barrera-mutex-750 8
Elapsed time = 2.406984e+01 seconds
Lab3 $ mpirun -np 1 barrera-mutex-750 16
Elapsed time = 3.144532e+01 seconds
Lab3 $ mpirun -np 1 barrera-mutex-750 32
Elapsed time = 7.268289e+01 seconds
Lab3 $ mpirun -np 1 barrera-mutex-750 64
Elapsed time = 1.692425e+02 seconds

```

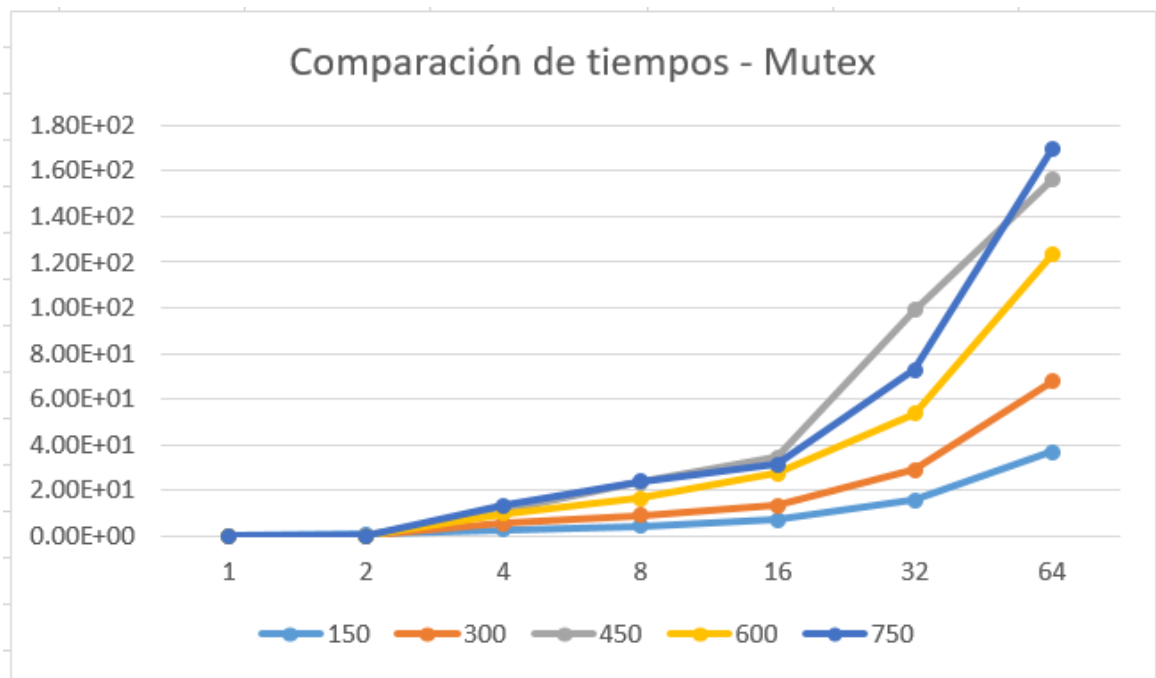
(e) BARRIER\_COUNT 750

Figura 7: Compilación de barreras con Mutex.

### 1.4.3. Resultados Comparativos Mutex

Thread	Procesos 1				
	BARRIER_COUNT				
	150	300	450	600	750
1	4.62E-04	1.08E-04	4.74E-04	1.39E-04	1.32E-04
2	7.17E-01	6.82E-04	1.30E-01	2.15E-02	3.60E-02
4	2.95E+00	5.70E+00	1.14E+01	9.74E+00	1.33E+01
8	4.59E+00	9.11E+00	2.36E+01	1.66E+01	2.41E+01
16	7.15E+00	1.32E+01	3.46E+01	2.74E+01	3.14E+01
32	1.56E+01	2.92E+01	9.90E+01	5.35E+01	7.27E+01
64	3.69E+01	6.78E+01	1.56E+02	1.23E+02	1.69E+02

(a) Barrera con Mutex



(b) Cuadro comparativo

Figura 8: Resultados gráficos

### 1.4.4. Análisis:

El uso de barreras es muy importante para el uso en la depuración y para determinar donde se esta produciendo un error.

En este ejercicio se pudo determinar la barrera Pthread es mucho mejor que la barrera con mutex , ya que en el código utiliza el `pthread_barrier` la cual aumenta el rendimiento