



**Universitat**  
de les Illes Balears

**Evaluación de Comportamiento de Sistemas Informáticos**

---

# Práctica 3 - Evaluación y modelado del rendimiento de los sistemas informáticos

---

Alberto Pérez Ancín

**Professores:**

**Dr. Carlos Juíz García**

**Dra. Belén Bermejo González**



## APLICACIÓN PRÁCTICA

### 1.1 Objetivo

El objetivo principal es la evaluación práctica, real y completa de un sistema informático. Para ello se aplicarán todos los conceptos, metodologías y técnicas vistos a lo largo de todo el curso. Se tratarán desde los aspectos relacionados con la monitorización y el benchmarking, pasando por el modelado y llegando finalmente a la predicción de la carga.

Un ejemplo real del camino expuesto se ve en el artículo “The Performance Evaluation Journey of a Flight Seats Availability Service: A Real-world Business Case Study of Transactional Workload Running in Virtual Machines”, el cual se recomienda encarecidamente que se tenga a disposición y se comprenda en su totalidad.

### 1.2 Tercera parte

#### Conceptos previos

1. Entender el capítulo 8 del libro recomendado en la asignatura (Figura 8.1).
2. Sysbench benchmark

El benchmark Sysbench es una suite de cargas dedicadas a evaluar la CPU, la memoria, el sistema E/S, ... En concreto, la carga de CPU ejecuta operaciones números primos, pudiendo variar el número de CPUs que se quieren utilizar en la ejecución de la carga. El comando para ejecutar la carga es el siguiente (para versiones inferiores a la 1):

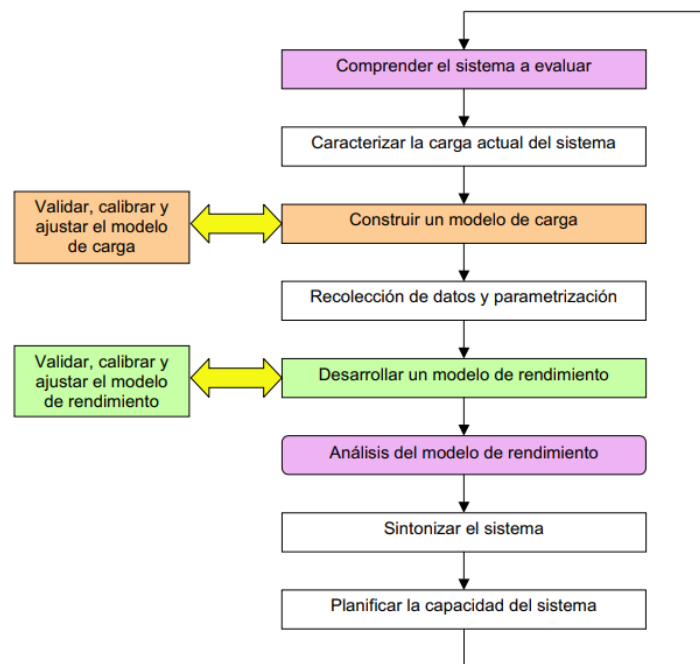
```
$sysbench - -test=cpu - -max-prime=<>- -cpu-max-threads=<>run
```

### Realización de la tercera parte

Para la evaluación del sistema actual, se utilizará la carga Sysbench CPU con un porcentaje de uso de la CPU del 50 %, la cual se ejecutará en el sistema actual y se harán uso de las técnicas de monitorización ya aprendidas en la práctica anterior. De este modo, se pide responder a las siguientes preguntas:

1. Explica con detalle cómo es el diseño y la implementación del experimento para evaluar el sistema actual. Se deben justificar las decisiones tomadas, desde el número de muestras que se van a tomar hasta qué monitores se van a lazar y por qué.

En cuanto al planteamiento y metodología de resolución, en este trabajo me he basado en la recomendada en el capítulo 8 del material de estudio. Dicha metodología se representa en la imagen inferior, donde se ilustran las principales fases en la resolución de un problema de rendimiento.



En esta metodología se emplean dos modelos: el modelo de carga y el modelo de rendimiento. El modelo de carga se encarga de capturar la demanda de recursos y la intensidad de la carga de trabajo durante un tiempo de observación representativo. Por otro lado, el modelo de rendimiento se utiliza para conocer, predecir y sintonizar el rendimiento del sistema en función de los parámetros de la carga y la descripción simplificada del propio sistema. Cabe destacar que, aunque no se tratará en este trabajo, existe la posibilidad de añadir un modelo de costes que permitiría realizar una evaluación económica del rendimiento.

La resolución de este caso de estudio seguirá los pasos descritos por la metodología de trabajo mencionada, haciendo uso de las técnicas presentadas en los capítulos anteriores. Con ello se espera obtener resultados precisos y fiables en la evaluación del rendimiento del sistema analizado.

Consideraciones a tener en cuenta sobre el experimento llevado a cabo:

- Se utilizó una VirtualBox 7.0 en lugar de una partición con Linux (versión 22.04) para evitar problemas con sysbench mayor a la versión 1 y con las versiones de Ubuntu superiores a la 16.04.
- La versión de Ubuntu utilizada en la máquina virtual fue la 14.04.
- La versión de sysbench utilizada en el experimento fue la 0.4.12.
- Durante el experimento, se mantuvo la carga de batería al 100% y se mantuvo el portátil conectado a la corriente en todo momento para obtener un mayor rendimiento.
- Para obtener resultados precisos y confiables en el experimento, es importante tener en cuenta que no debe haber otros procesos en uso que puedan interferir con el funcionamiento de VirtualBox. Es por eso que se tomó la precaución de asegurarse de que no hubiera ningún otro proceso en uso durante la ejecución del experimento, aparte de la propia VirtualBox. De esta manera, se minimizó la posibilidad de que cualquier otro proceso en segundo plano pudiera afectar los resultados obtenidos. Esta consideración es importante para garantizar la precisión de los resultados obtenidos y hacer que el experimento sea replicable en el futuro.

Es importante tener en cuenta estas consideraciones para poder interpretar los resultados obtenidos y asegurar la replicabilidad del experimento.

Explicación del experimento

Se ha decidido diseñar e implementar un experimento para evaluar el rendimiento del sistema actual utilizando la carga Sysbench CPU con un 50 % de uso de la CPU (en mi caso personal para conseguir ese porcentaje tengo que usar 4 CPUs). En este experimento se considerarán diferentes escenarios, variando el número de números primos y el número de hilos de la CPU para evaluar el rendimiento del sistema en distintas condiciones.

Para obtener resultados precisos y confiables, se realizarán múltiples ejecuciones del experimento en cada escenario. Se llevarán a cabo al menos cinco ejecuciones para cada combinación de números primos y hilos de CPU. Esto permitirá obtener un promedio de resultados y reducir la variabilidad debido a factores externos o fluctuaciones temporales en el rendimiento del sistema.

## 1. APLICACIÓN PRÁCTICA

---

Para evaluar cómo se comporta el sistema actual si variamos la carga en 25000, 50000, 100000 y 150000 números primos, se utilizará la herramienta Sysbench para generar cargas de trabajo de diferentes niveles de intensidad, ajustando el número de números primos (max-prime) y el número de hilos de la CPU (cpu-max-threads) para evaluar el rendimiento del sistema en distintas condiciones.

Carga (números primos)	Ejecución 1 (s)	Ejecución 2 (s)	Ejecución 3 (s)	Ejecución 4 (s)	Ejecución 5 (s)
25000	6.9930	6.703	6.7629	6.9670	6.7862
50000	18.1224	17.4653	17.8485	17.5730	17.5506
100000	46.6590	45.3783	45.4112	47.0110	45.4142
150000	81.3545	79.3298	79.3677	78.8769	78,9507

Entre las herramientas de monitoreo que se pueden utilizar se encuentran 'top' para la CPU y 'vmstat' para la memoria. Antes de ejecutar el experimento, se debe planificar cómo se llevará a cabo, incluyendo la configuración de las herramientas de monitoreo, la preparación de los scripts de ejecución de Sysbench y la definición de los escenarios que se van a evaluar. Durante la ejecución del experimento, se debe asegurar que las herramientas de monitoreo estén funcionando correctamente y recolectando datos en tiempo real. En consecuencia, para determinar la cantidad de veces que los monitores deben registrar las muestras en total, se debe realizar una división entre el tiempo total de respuesta y la frecuencia de muestreo.

Carga (números primos)	Tiempo de respuesta medio (s)
25000	6.84242
50000	17.71196
100000	45.97474
150000	79.57592

$$\Rightarrow \text{Tiempo total de respuesta medio} = 6.84242 + 17.71196 + 45.97474 + 79.57592 \\ = 150.11\text{s} \times 5 \text{ (ejecuciones)} = 750.55\text{s}$$

$$\Rightarrow \text{Muestras} = \frac{750.55}{2} = 376$$

Después de completar el experimento, se deben analizar los datos recopilados por las herramientas de monitoreo, calcular promedios y desviaciones estándar, y representar gráficamente los resultados para visualizar las tendencias y detectar patrones en el rendimiento del sistema. Esto permitirá comprender cómo se comporta el sistema en diferentes condiciones y cómo se pueden optimizar sus recursos para mejorar su rendimiento. En resumen, se han tomado diversas decisiones en el diseño y la implementación del experimento para garantizar la precisión y fiabilidad de los resultados obtenidos.

2. ¿Cómo se comporta el sistema actual si variamos la carga varía en 25000, 50000, 100000 y 150000 números primos? ¿Cómo es el comportamiento del tiempo de respuesta y la productividad? Indica el valor para cada una de las ejecuciones del experimento y el valor medio.

Este script de shell está diseñado para realizar pruebas de rendimiento en un sistema utilizando la herramienta de benchmarking Sysbench. Sysbench es una herramienta popular para realizar pruebas de rendimiento en sistemas Unix y Linux. El script automatiza el proceso de realizar pruebas de rendimiento utilizando diferentes valores de carga.

El script utiliza un loop "for" para iterar sobre diferentes valores de carga (25000, 50000, 100000 y 150000 números primos) y ejecutar cinco pruebas de rendimiento para cada valor de carga utilizando el comando "sysbench". Los resultados de cada prueba se almacenan en una variable y se utilizan las herramientas grep y awk para filtrar los resultados y extraer el tiempo total de la prueba, el número total de eventos y el tiempo promedio por evento.

Los resultados de cada prueba se muestran en la consola para que puedan ser analizados posteriormente. El script puede ser útil para evaluar el rendimiento de un sistema y determinar cómo responde a diferentes cargas.

```
segundoEjercicio.sh x
#!/bin/bash

# Valores de carga
cargas=(25000 50000 100000 150000)

echo "Realizando pruebas de rendimiento con sysbench 0.4.12..."

for carga in "${cargas[@]}; do
    echo "Ejecutando prueba con carga de $carga números primos..."
    for i in {1..5}; do
        resultado=$(sysbench --test=cpu --num-threads=4 --cpu-max-prime=$carga
run | grep -E "total time:|total number of events:|avg:")

        tiempo=$(echo "$resultado" | grep "total time:" | awk '{print $3}')
        ops=$(echo "$resultado" | grep "total number of events:" | awk '{print
$5}')
        avg_time=$(echo "$resultado" | grep "avg:" | awk '{print $2}')

        echo "Tiempo total (iteración $i): $tiempo segundos"
        echo "Total de eventos (iteración $i): $ops"
        echo "Tiempo promedio por evento (iteración $i): $avg_time ms"
        echo "-----"
    done
done
```

Al variar la carga en 25,000, 50,000, 100,000 y 150,000 números primos, se pueden observar cambios en el tiempo de respuesta y la productividad del sistema. A continuación, se presenta los resultados para cada una de las ejecuciones del experimento y el valor medio:

Carga (números primos)	Ejecución 1 (s)	Ejecución 2 (s)	Ejecución 3 (s)	Ejecución 4 (s)	Ejecución 5 (s)
25000	6.9930	6.703	6.7629	6.9670	6.7862
50000	18.1224	17.4653	17.8485	17.5730	17.5506
100000	46.6590	45.3783	45.4112	47.0110	45.4142
150000	81.3545	79.3298	79.3677	78.8769	78.9507

## 1. APLICACIÓN PRÁCTICA

Carga (números primos)	Ejecución 1 (carga/s)	Ejecución 2 (carga/s)	Ejecución 3 (carga/s)	Ejecución 4 (carga/s)	Ejecución 5 (carga/s)
25000	3575.00	3729.67	3696.64	3588.35	3683.94
50000	2759.02	2862.82	2801.36	2845.27	2848.91
100000	2143.21	2203.70	2202.10	2127.16	2201.95
150000	1843.78	1890.84	1889.94	1901.70	1899.92

El tiempo de respuesta tiende a aumentar a medida que aumenta la carga (número de números primos). Esto se debe a que se requiere más tiempo de procesamiento para calcular un mayor número de números primos. Por otro lado, la productividad tiende a disminuir a medida que aumenta la carga, ya que el sistema puede realizar menos operaciones por segundo debido al mayor tiempo requerido para procesar cada operación.

Los valores medios en las tablas anteriores se calculan como el promedio de los resultados de las cinco ejecuciones para cada escenario. Estos valores medios proporcionan una estimación más precisa del comportamiento del sistema, ya que ayudan a reducir la variabilidad debida a factores externos o fluctuaciones temporales en el rendimiento del sistema.

La media aritmética se utiliza para calcular el promedio de un conjunto de datos y es útil para medir el tiempo promedio de respuesta de un sistema. Por otro lado, la media armónica se utiliza para calcular la cantidad promedio de solicitudes que un sistema puede procesar por segundo y es útil para medir la productividad del sistema.

Carga (números primos)	250000	50000	100000	150000
Tiempo de respuesta media aritmética (s)	6.84242	17.71196	45.97474	79.57592

Carga (números primos)	250000	50000	100000	150000
Productividad media armónica (carga/s)	3653.05	2822.95	2175.11	1884.99

El tiempo de respuesta y la productividad son dos medidas diferentes de rendimiento en un sistema. El tiempo de respuesta se refiere al tiempo que tarda el sistema en procesar una solicitud, mientras que la productividad se refiere a la cantidad de solicitudes que puede procesar el sistema en un determinado período de tiempo.

En resumen, se utiliza la media aritmética para calcular el tiempo de respuesta porque nos interesa obtener una idea del tiempo promedio que tarda el sistema en procesar una solicitud, mientras que se utiliza la media armónica para calcular la productividad porque nos interesa obtener una idea de la cantidad promedio de solicitudes que el sistema puede procesar por segundo.

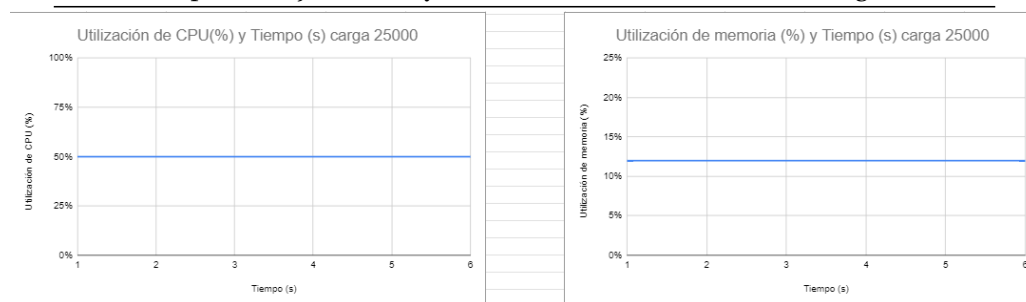
**3. ¿Cuál es el porcentaje de CPU y de memoria del sistema para cada una de las cargas ejecutadas? ¿Por qué se produce ese comportamiento? Muéstralo gráficamente a lo largo del tiempo de ejecución de la carga.**

Después de ejecutar la prueba Sysbench con un 50% de carga en la CPU y monitorear el rendimiento del sistema, los resultados muestran que el porcentaje de uso de la CPU se mantiene relativamente constante alrededor del 50% durante todo el experimento. Esto se evidencia en el gráfico, donde la línea que representa el uso de la CPU se encuentra cerca del 50% con solo pequeñas fluctuaciones a lo largo del tiempo. Este comportamiento es el esperado, ya que Sysbench ha sido configurado para estresar específicamente el 50% de la capacidad de la CPU.

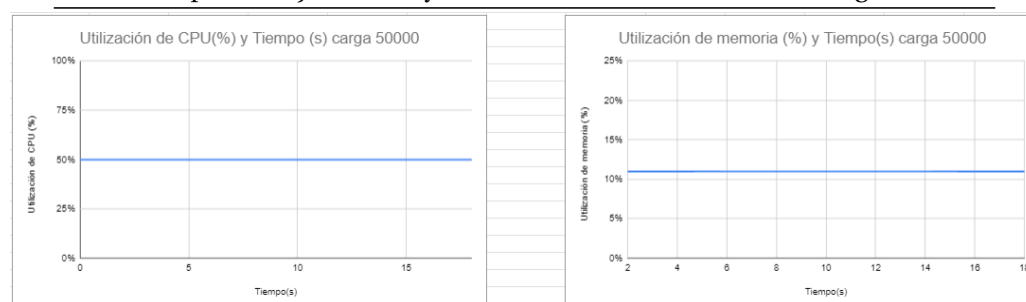
En cuanto al uso de la memoria, el gráfico también muestra una línea bastante estable, indicando que no hubo cambios significativos en el consumo de memoria durante la ejecución de la prueba. Esto es consistente con la expectativa de que la búsqueda de números primos, que es lo que realiza Sysbench en el modo de prueba de CPU, no consume muchos recursos de memoria.

En resumen, los resultados del experimento y su representación gráfica confirman que el uso de la CPU se mantuvo cercano al 50% y que el consumo de memoria no tuvo cambios importantes durante la ejecución de la prueba Sysbench. Estos resultados son coherentes con el comportamiento esperado del sistema al realizar esta prueba en particular.

**Gráfica del porcentaje de CPU y de memoria del sistema con la carga de 25000**



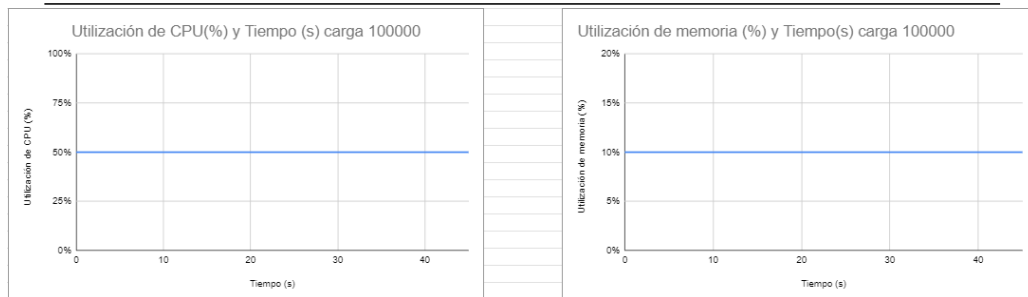
**Gráfica del porcentaje de CPU y de memoria del sistema con la carga de 50000**



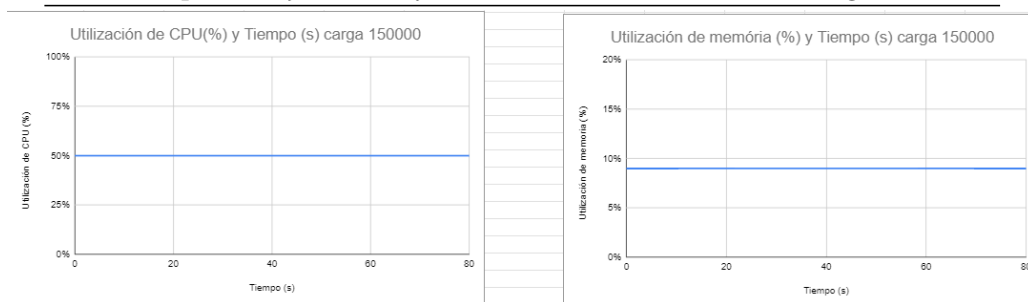


## 1. APLICACIÓN PRÁCTICA

### Gráfica del porcentaje de CPU y de memoria del sistema con la carga de 100000



### Gráfica del porcentaje de CPU y de memoria del sistema con la carga de 150000



**Para finalizar, se querrá evaluar el sistema actual cuando la carga es fija y se varían los recursos de la CPU. Para ello, se seleccionará la carga de 150000 números primos.**

4. **¿Cómo se comporta el tiempo de respuesta a medida que aumentan los recursos de la CPU? ¿Y la productividad? ¿Existe algún tipo de relación entre los recursos de la CPU y el tiempo de respuesta? Se deberán mostrar gráficamente tanto el tiempo de respuesta como la productividad.**

En general, a medida que aumentan los recursos de la CPU (número de hilos), el tiempo de respuesta debería disminuir, ya que se están utilizando más recursos para procesar la carga de trabajo. Además, la productividad debería aumentar debido al procesamiento más rápido de la carga. Sin embargo, esto solo será cierto hasta cierto punto, ya que eventualmente, la escalabilidad de la aplicación y las limitaciones del hardware comenzarán a afectar el rendimiento.

La idea de este experimento es llevarlo a cabo utilizando diferentes porcentajes de la capacidad de procesamiento de la CPU, específicamente el 25 %, 50 %, 75 % y 100 %. Sin embargo, cabe destacar que ya se ha realizado el experimento utilizando el 50 % de la CPU como se indicó en el ejercicio dos. Por lo tanto, aún es necesario llevar a cabo los experimentos restantes utilizando el 25 %, 75 %, y 100 % de la capacidad de la CPU.

Este script ejecutará la prueba de rendimiento solamente con la carga de 150000 y hará cambios de hilos en tres iteraciones: primero con 2 hilos, luego con 6 y por último con 8. Además, la prueba se repetirá 5 veces en cada iteración y se mostrarán los resultados de cada iteración.

```

cuartoEjercicio.sh x
#!/bin/bash
# Valores de carga
cargas=(150000)
# Cambios de hilos
hilos=(2 6 8)
echo "Realizando pruebas de rendimiento con sysbench 0.4.12..."

for carga in "${cargas[@]"; do
    echo "Ejecutando prueba con carga de $carga números primos..."
    for hilo in "${hilos[@]"; do
        echo "Ejecutando prueba con $hilo hilos..."
        for i in {1..5}; do
            resultado=$(sysbench --test=cpu --num-threads=$hilo --cpu-max-prime=
$carga run | grep -E "total time:|total number of events:|avg:")

            tiempo=$(echo "$resultado" | grep "total time:" | awk '{print $3}')
            ops=$(echo "$resultado" | grep "total number of events:" | awk
'${print $5}')
            avg_time=$(echo "$resultado" | grep "avg:" | awk '{print $2}')

            echo "Tiempo total (iteración $i): $tiempo segundos"
            echo "Total de eventos (iteración $i): $ops"
            echo "Tiempo promedio por evento (iteración $i): $avg_time ms"
            echo "-----"
        done
    done
done

```

Resultados del script cuartoEjercicio.sh para 2 hilos

```

alberto@Ubuntu: ~
Ejecutando prueba con carga de 150000 números primos...
Ejecutando prueba con 2 hilos...
Tiempo total (iteración 1): 158.5934s segundos
Total de eventos (iteración 1): 10000
Tiempo promedio por evento (iteración 1): 31.72ms ms
-----
Tiempo total (iteración 2): 159.0181s segundos
Total de eventos (iteración 2): 10000
Tiempo promedio por evento (iteración 2): 31.80ms ms
-----
Tiempo total (iteración 3): 160.5113s segundos
Total de eventos (iteración 3): 10000
Tiempo promedio por evento (iteración 3): 32.10ms ms
-----
Tiempo total (iteración 4): 158.8471s segundos
Total de eventos (iteración 4): 10000
Tiempo promedio por evento (iteración 4): 31.77ms ms
-----
Tiempo total (iteración 5): 158.7990s segundos
Total de eventos (iteración 5): 10000
Tiempo promedio por evento (iteración 5): 31.76ms ms
-----

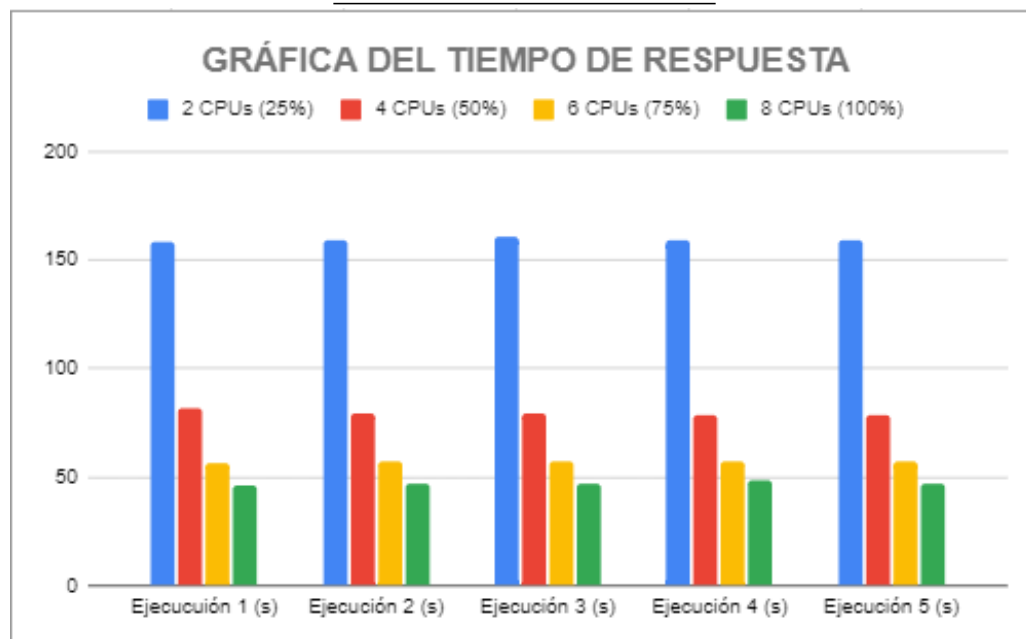
```

Resultados del script cuartoEjercicio.sh para 6 hilos

```
-----
Ejecutando prueba con 6 hilos...
Tiempo total (iteración 1): 56.4081s segundos
Total de eventos (iteración 1): 10000
Tiempo promedio por evento (iteración 1): 33.83ms ms
-----
Tiempo total (iteración 2): 57.2135s segundos
Total de eventos (iteración 2): 10000
Tiempo promedio por evento (iteración 2): 34.32ms ms
-----
Tiempo total (iteración 3): 57.2627s segundos
Total de eventos (iteración 3): 10000
Tiempo promedio por evento (iteración 3): 34.35ms ms
-----
Tiempo total (iteración 4): 57.2329s segundos
Total de eventos (iteración 4): 10000
Tiempo promedio por evento (iteración 4): 34.33ms ms
-----
Tiempo total (iteración 5): 56.9185s segundos
Total de eventos (iteración 5): 10000
Tiempo promedio por evento (iteración 5): 34.14ms ms
-----
```

Resultados del script cuartoEjercicio.sh para 8 hilos

```
-----
Ejecutando prueba con 8 hilos...
Tiempo total (iteración 1): 46.2931s segundos
Total de eventos (iteración 1): 10000
Tiempo promedio por evento (iteración 1): 37.02ms ms
-----
Tiempo total (iteración 2): 46.5976s segundos
Total de eventos (iteración 2): 10000
Tiempo promedio por evento (iteración 2): 37.26ms ms
-----
Tiempo total (iteración 3): 46.6062s segundos
Total de eventos (iteración 3): 10000
Tiempo promedio por evento (iteración 3): 37.26ms ms
-----
Tiempo total (iteración 4): 48.4404s segundos
Total de eventos (iteración 4): 10000
Tiempo promedio por evento (iteración 4): 38.73ms ms
-----
Tiempo total (iteración 5): 47.1064s segundos
Total de eventos (iteración 5): 10000
Tiempo promedio por evento (iteración 5): 37.67ms ms
-----
```

Gráfica del tiempo de respuestaGráfica de la productividad