

# PRÁCTICA FINAL ACSIC

**Avaluació Comportament Sistemes  
Informàtics · 21716**

**Alumno:** Adrián Bennasar Polzin

**Correo:** [adribennasar@gmail.com](mailto:adribennasar@gmail.com)

**Grupo:** 01 (Mañana)

**Profesores:** Carlos Juiz Garcia,  
Belén Bermejo González



## Índice:

1. Introducción.....	Página 2.
2. Cuestiones teóricas.....	Página 3.
3. Estudio del sistema actual.....	Página 4.
3.1. Definición de la carga.....	Página 5.
3.2. Cálculos y resultados.....	Página 6.
3.3. Gráficos.....	Página 8.
4. Comparación de sistemas.....	Página 14.

## INTRODUCCIÓN

En esta práctica el alumno simula ser un analista de sistemas encargado de llevar a cabo un estudio de rendimiento de un sistema que proporciona servicio a clientes. Este sistema pertenece a una empresa ficticia llamada ACSIC que ofrece servicios de alta computación.

La idea principal de la práctica es que la empresa ficticia desea mejorar la calidad del servicio que proporciona a sus clientes, y el analista de sistemas debe comparar los sistemas actuales con otro sistema que es candidato para sustituir a los actuales.

Las técnicas que se han usado para la evaluación del rendimiento son las técnicas de monitorización y benchmarking.

Se presentarán unos resultados de manera gráfica y tras comparar las características y el rendimiento de ambos sistemas se llegará a una conclusión.

## CUESTIONES TEÓRICAS

1. **¿Qué ventajas presentan las técnicas de monitorización y benchmarking con respecto al modelado y la simulación a la hora de estudiar el rendimiento de un sistema? ¿Qué relación existe entre ellas?**

Las técnicas de monitorización y benchmarking tienen la ventaja de que se aplican sobre sistemas físicos reales mientras que las técnicas de modelado y simulación se aplican sobre modelos ficticios y haciendo uso de fórmulas matemáticas. Esto es una ventaja porque al trabajar sobre un sistema real es más fácil obtener una alta precisión en los resultados, que si trabajamos sobre modelos ficticios.

La relación principal que tienen estas dos técnicas es que se complementan de manera que, si se quiere montar un modelo o una simulación fiables, se deben alimentar con datos reales obtenidos a través de benchmark y/o monitorización.

Esto es porque aunque se puede montar un modelo o simulación inventando valores, habrá el riesgo de que las diferentes variables estén desequilibradas y las cosas no cuadren, mientras que si hacemos uso de la ayuda de datos reales obtenidos de benchmark y monitorización obtendremos un resultado más sólido.

2. **¿Cuál de los dispositivos del sistema tendría más relevancia a la hora de estudiar su rendimiento? ¿Por qué?**

El dispositivo del sistema que tiene mas relevancia a la hora de estudiar su rendimiento es claramente la CPU. Esto se debe a que el sistema se usa para llevar a cabo servicios de alta computación, donde el dispositivo más demandado es la CPU. Por lo tanto, si queremos mejorar el rendimiento del servicio, la manera más efectiva de hacerlo sería mejorando la potencia de la CPU, ya que, por ejemplo, mejorar la memoria, no aportaría una mejora de rendimiento notable.

3. **¿Qué tipo de carga sintética inyectarías en el sistema para imitar la carga real? ¿Por qué?**

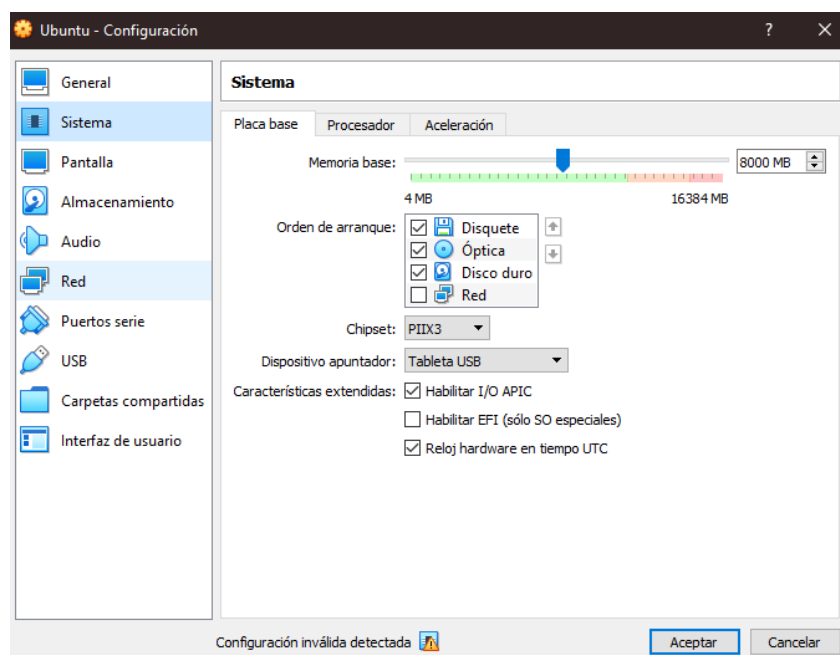
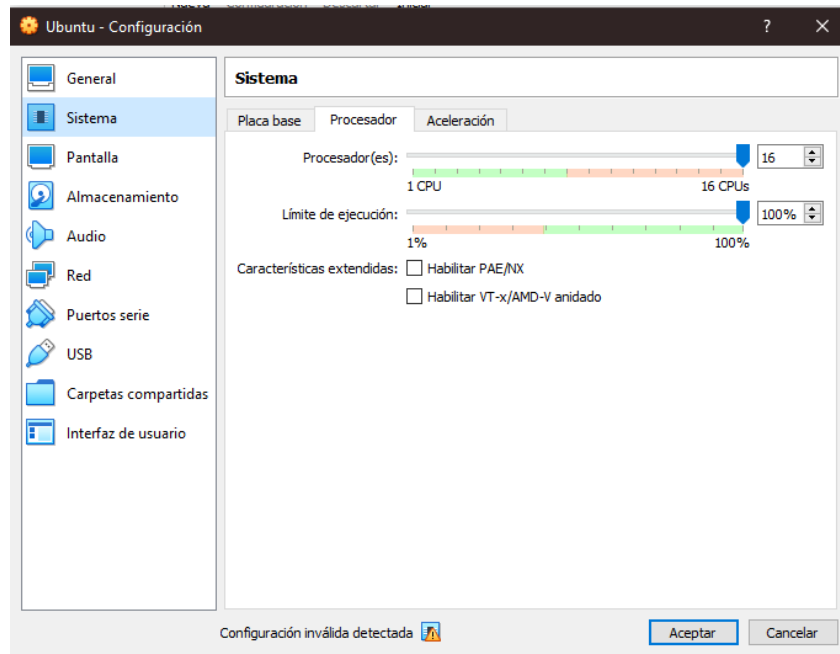
La carga que inyectaría para imitar la carga real sería evidentemente una carga que se centre en hacer trabajar a la CPU, porque el servicio que ha de ofrecer este sistema requiere trabajo de CPU.

## ESTUDIO DEL SISTEMA ACTUAL

El análisis del sistema se ha llevado acabo en una máquina virtual con Ubuntu.

Las características del sistema actual son:

- CPU: Intel Core I9-9900K 3.6 GHz (8 cores físicos | 16 threads)
- Memoria principal: 8GB RAM
- Sistema operativo: Ubuntu 20.04 LTS



## Definición de la carga

Se han definido cuatro niveles de carga diferentes (ascendentes) para inyectar en el sistema, con el benchmark Sysbench.

La versión de Sysbench que se ha utilizado es la **0.4.12**, debido que a varios compañeros me la han recomendado porque las versiones más recientes de Sysbench les causaron problemas que esta versión no presenta.

La carga seleccionada es la de cálculo de números primos, y los 4 niveles de carga establecidos han sido:

- Nivel de carga 1: 250.000 números primos.
- Nivel de carga 2: 500.000 números primos.
- Nivel de carga 3: 750.000 números primos.
- Nivel de carga 4: 1.000.000 números primos.

He elegido estos valores para poder hacer una comparación equilibrada con el sistema con el que hay que comparar este, ya que para el otro sistema se han ejecutado estos mismos valores de carga. Es decir, lo que pretendemos ver es, para la misma carga, cómo rinden 2 ordenadores diferentes. Si se hubieran ejecutado diferentes valores de carga en un sistema que en el otro, sería difícil decir cuál rinde mejor porque no le estamos haciendo las mismas pruebas.

El comando utilizado para ejecutar el Sysbench es el siguiente:

```
sysbench --test=cpu --cpu-max-prime=número_primos --num-threads=16 run
```

Cada uno de los niveles de carga se ha ejecutado un total de 5 veces de manera consecutiva, y se han guardado los resultados del sysbench, para usar los resultados de cada ejecución para realizar una media y de esta manera obtener realizar unos cálculos más precisos.

Mientras el sistema ejecuta la carga, este se ha monitorizado a través del monitor **top**.

Al monitor top se le han añadido una serie de parámetros de manera que aporta gran parte de la información necesaria para realizar los cálculos que se piden en el enunciado. Esto es:

- Uso de CPU que consumen los monitores (en este caso el único es el top).
- Utilización de CPU a lo largo del período de observación.
- Utilización de memoria principal a lo largo del periodo de observación.

El comando es el siguiente:

```
top -b -d5 -nX | grep -w "top$\\|%Cpus(s) \\|MiB Mem"
```

Donde X es el número de muestras, que cambia de un nivel de carga a otro.

Para que el monitor top se ejecuta con una duración de manera que coincida con la duración del sysbench, se ha hecho una prueba piloto con el sysbench para cada nivel de carga para obtener la duración, y entonces se ha ajustado el parámetro -nX del top. El parámetro -d5 es fijo porque como se pide en el enunciado, se obtienen muestras cada 5 segundos.

## Cálculos y resultados

### **Datos monitores:**

#### **a) Sobrecarga que ejerce el monitor top sobre el sistema:**

La sobrecarga de un monitor se calcula dividiendo el tiempo de ejecución del monitor para una muestra entre el intervalo de muestreo:

$$(0.164 \text{ s} / 5) * 100 = 3.28 \% \text{ sobrecarga.}$$

Esto es una sobrecarga baja, y se debe principalmente a dos factores:

- El sistema tiene una potencia que hace que la ejecución de un simple monitor no tenga un gran impacto.
- El intervalo de muestreo es de 5 segundos, por lo que no se ejerce una presión alta con el muestro, como se ejercería si tomáramos muestras cada 1 segundo, por ejemplo.

Este valor es el mismo para todos los niveles de carga, ya que por como esta definida la sobrecarga, podemos ver que no depende de qué nivel de carga evaluemos, porque para todos los niveles de carga se usa el mismo intervalo de muestreo, 5 segundos.

#### **b) Uso medio de la CPU que consume el monitor top:**

La información necesaria para realizar este cálculo la aporta el propio monitor top. Se ha guardado esta información para cada nivel de carga y para cada ejecución y se ha realizado una media. Se han obtenido los siguientes resultados:

Carga de trabajo	Uso medio CPU monitor TOP[%]
250.000	0,1 %
500.000	0,11 %
750.000	0,12 %
1.000.000	0,13 %

Sobrecarga Monitor TOP [%]
3,28 %

Como se puede apreciar, ocurren 2 cosas:

- El uso de CPU por parte del monitor top es casi nulo:

Esto es porque como se ha comentado antes, el sistema es tan potente que ejecutar un monitor simple como el top no le lleva ningún esfuerzo.

- El uso de CPU por parte del monitor es prácticamente el mismo para los 4 niveles de carga:

Esto se debe a que aumentar el número de primos solo provoca que el monitor se ejecute durante más tiempo, pero no que tenga un mayor impacto por sí solo.

### Datos generales de rendimiento:

Carga de trabajo	Tiempo medio de respuesta	Productividad media	Utilización media CPU[%]	Utilización media memoria[%]
250.000	38,31 s	6525,71 cálculos/s	97,53 %	19,77 %
500.000	100,52 s	4974,13 cálculos/s	99,13 %	20,11 %
750.000	177,43 s	4227,02 cálculos/s	99,5 %	20,5 %
1.000.000	265,38 s	3768,18 cálculos/s	99,67 %	21,9 %

- a) **Tiempo medio de respuesta:** es el tiempo que ha tardado en completarse el cálculo de número de primos.

Para obtener el tiempo de respuesta, se ha cogido el tiempo que devuelve el propio sysbench y se ha hecho la media para cada nivel de carga.

- b) **Productividad media:** se define como el número de operaciones llevadas a cabo por unidad de tiempo. En este caso la productividad media representa cálculos por segundo (cálculos/s). Esto es porque son las unidades que mejor representan el objetivo del rendimiento del sistema.

Para obtener los valores de productividad media, se ha dividido el número total de cálculos que ha hecho el sistema para un nivel determinado (250K, 500K, 750K o 1000K), entre el tiempo de respuesta medio de ese mismo nivel.

- c) **Utilización media de CPU:** consiste en la utilización global media de la CPU a lo largo del periodo de observación. Se han obtenido valores cercanos al 100% para los 4 niveles de carga porque las pruebas se han realizado bajo condiciones de saturación donde absolutamente todos los cores trabajan (--num-threads=16).

Para obtener los valores de utilización media de CPU, se han realizado medias sobre los valores devueltos por el monitor top.

- d) **Utilización media de memoria principal:** es la utilización de memoria principal (RAM) a lo largo del periodo de observación.

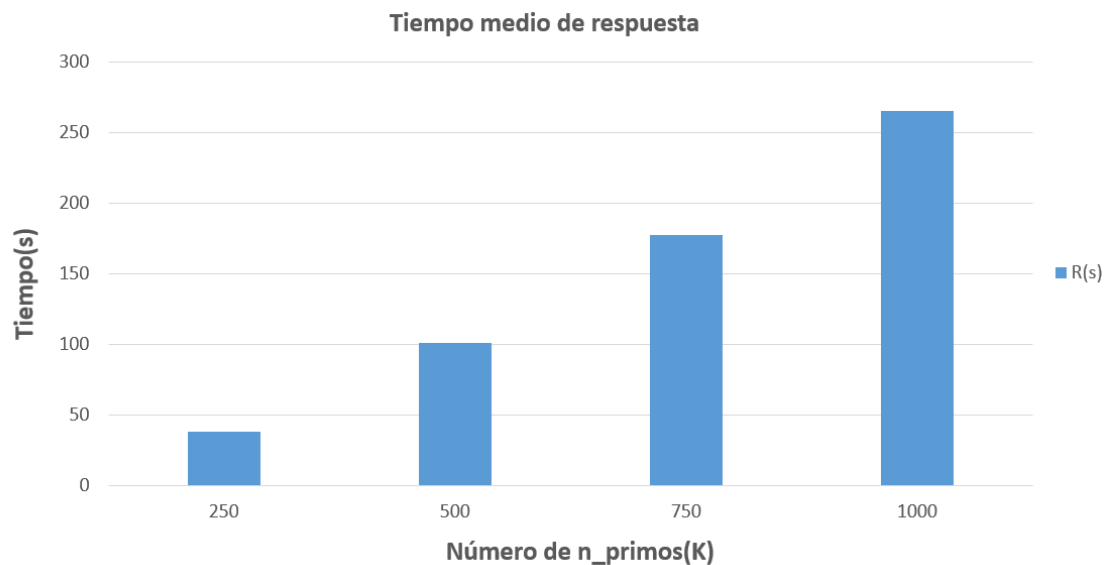


Para obtener los valores, se ha aplicado una fórmula sobre el valor MemTotal y los valores MemFree de cada nivel de carga, obtenidos del monitor top. La fórmula es la siguiente:

$$[(\text{MemTotal} - (\text{valor medio}) \text{ free}) / \text{MemTotal}] * 100$$

## Gráficos

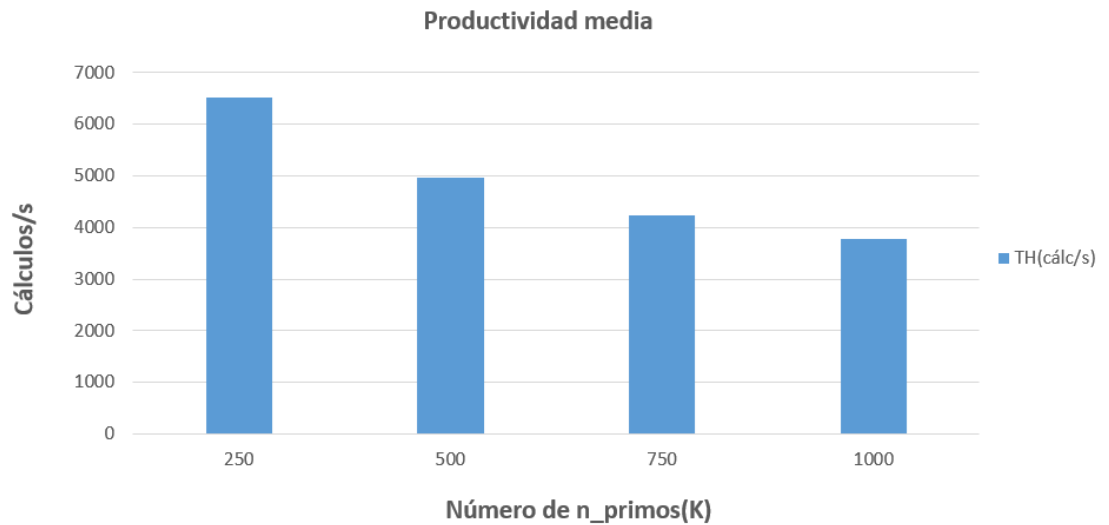
- Tiempo de respuesta:



Este gráfico ilustra el tiempo que tarda en ejecutarse la carga para los 4 diferentes niveles de carga, o, dicho de otra manera, el tiempo que necesita el sistema para llevar a cabo el cálculo de números primos.

Como se puede observar, el tiempo de respuesta crece a medida que aumentamos el número de número primos. Esto es lógico ya que, para calcular más, se necesita más tiempo. En concreto, el tiempo de respuesta aumenta de manera lineal. Esto se debe a que hemos aumentado los números primos siguiendo la misma proporción, es decir, 250K.

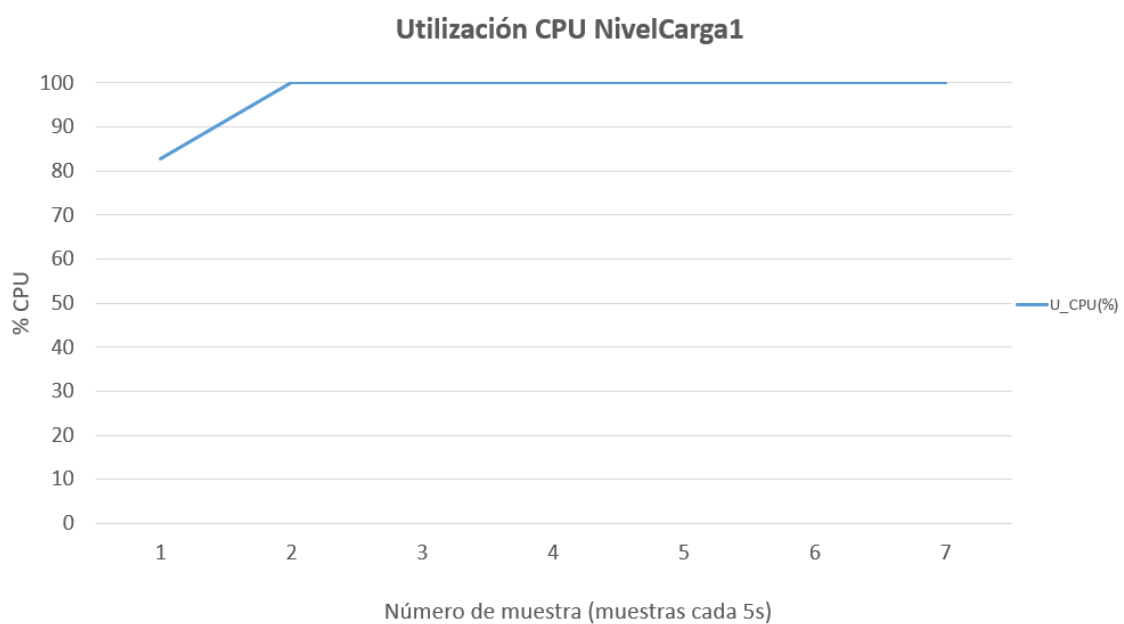
- Productividad media:



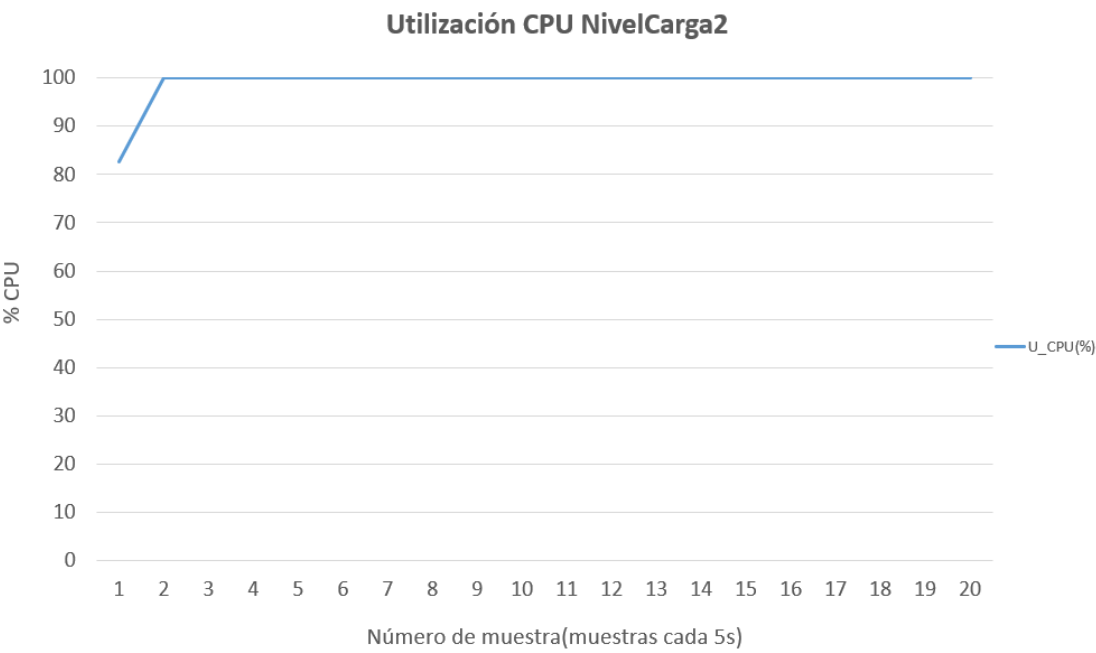
Este gráfico ilustra las operaciones por unidad de tiempo que es capaz de hacer el sistema. La productividad baja a medida que se incrementa el número de primos porque por mucho que le digamos que calcule más primos, llega un momento en que no puede ir más rápido y al tener que calcular más primos cada vez tarda más y esto provoca que el denominador (tiempo) reduzca el valor de la productividad (cálculos / s).

- Utilización media de CPU:

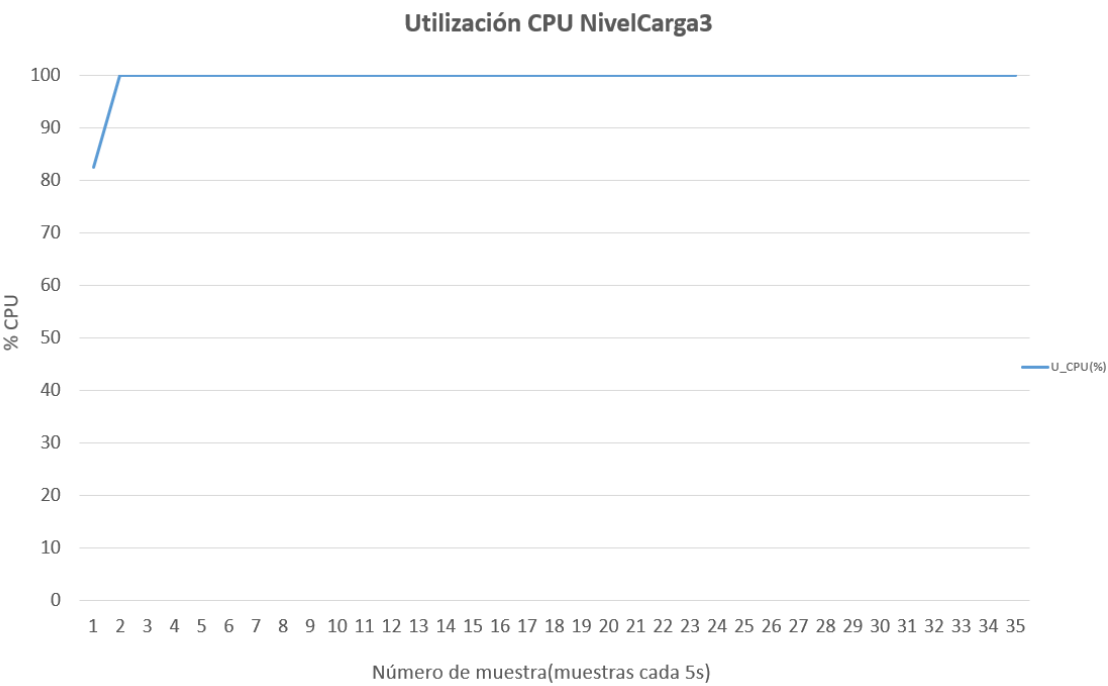
#### **Nivel de carga 1:**

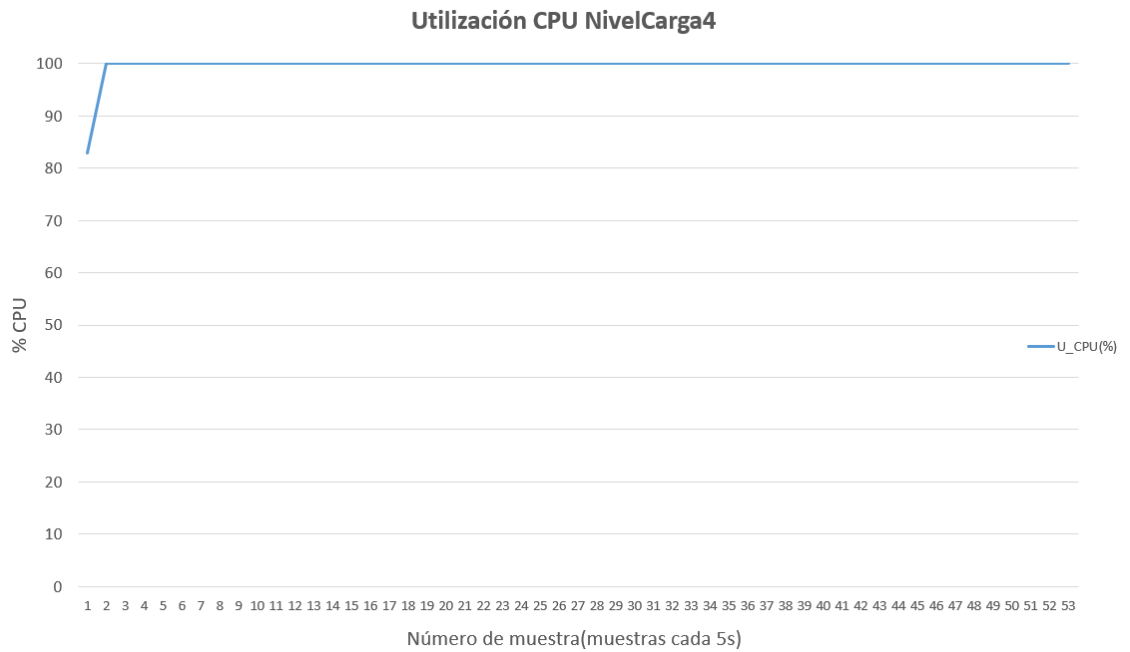


**Nivel de carga 2:**



**Nivel de carga 3:**





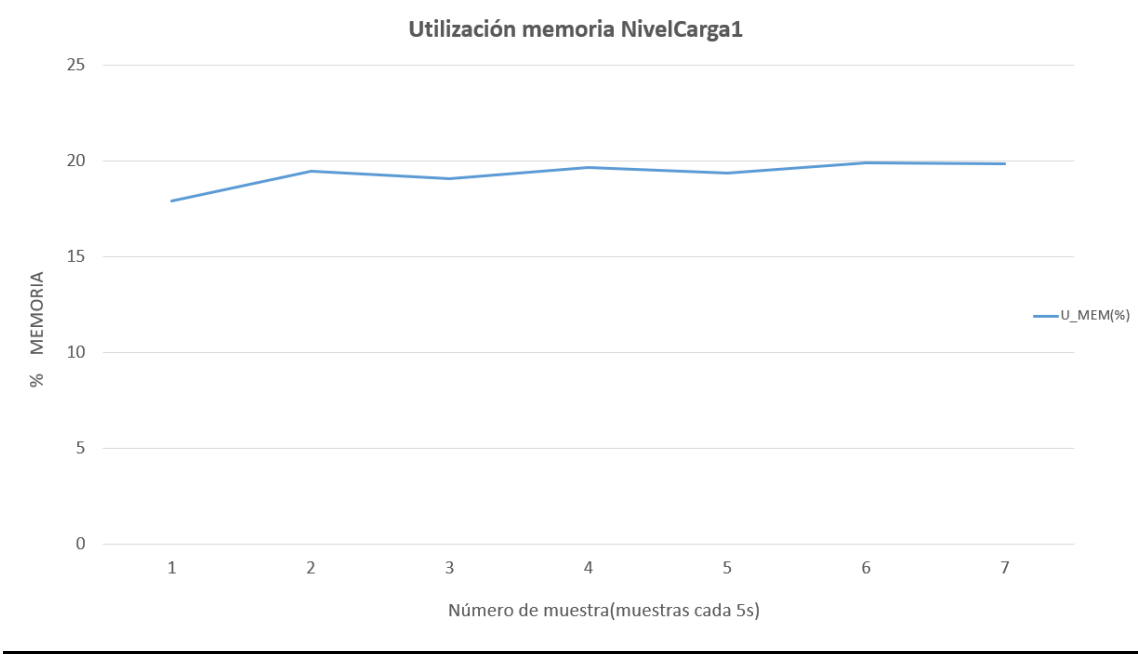
Como se puede ver en los 4 gráficos, al principio del periodo de observación la utilización de CPU sube drásticamente, y una vez llega a 100% se mantiene en ese valor hasta el final de la ejecución.

La subida inicial se debe a que cuando se monitoriza la carga, los primeros segundos el benchmark está arrancando por lo que aún no llega a consumir todos los recursos que necesita.

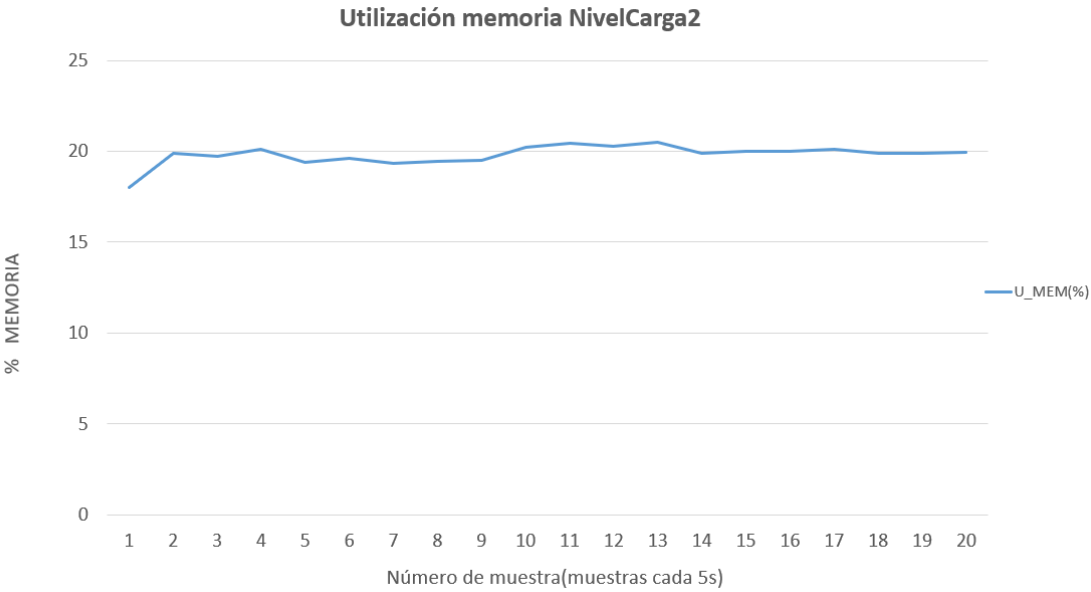
Por otra parte, la razón por la que el valor de utilización se mantiene al 100% durante toda la ejecución es porque se utilizan todos los cores del sistema.

- Utilización media de memoria principal:

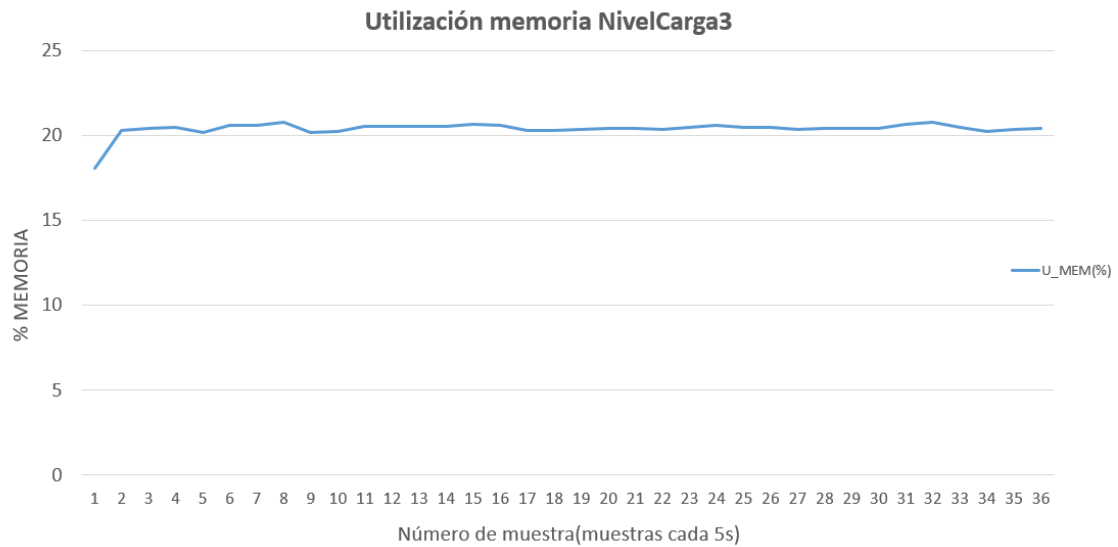
**Nivel de carga 1:**



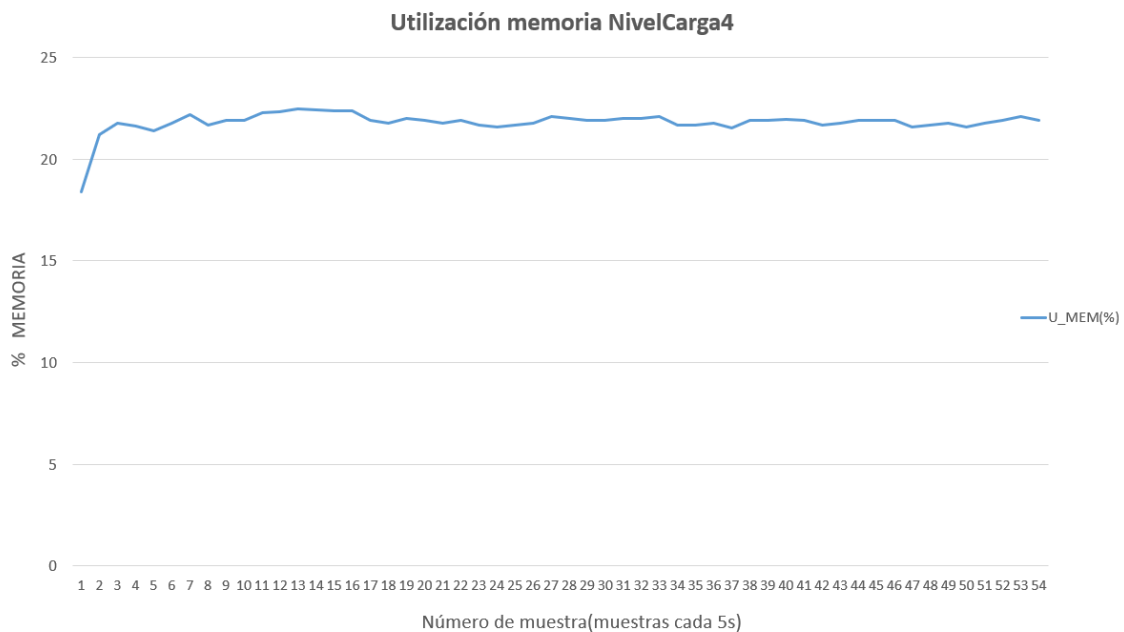
**Nivel de carga 2:**



### **Nivel de carga 3:**



### **Nivel de carga 4:**



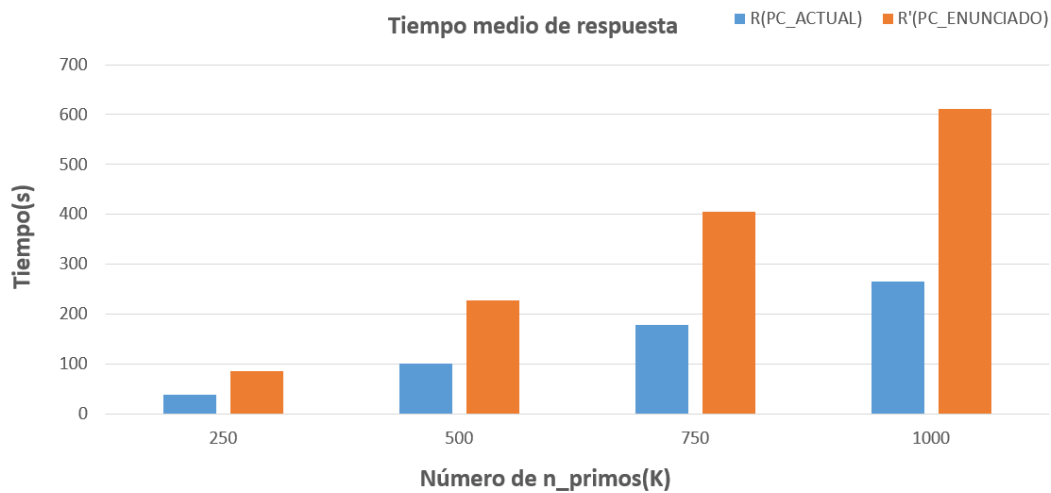
En los 4 gráficos de arriba se puede observar que el uso de memoria principal no varia mucho de un nivel de carga a otro. Esto es porque el benchmark se centra en hacer trabajar a la CPU por lo que gran parte del uso de memoria es del propio sistema operativo y los

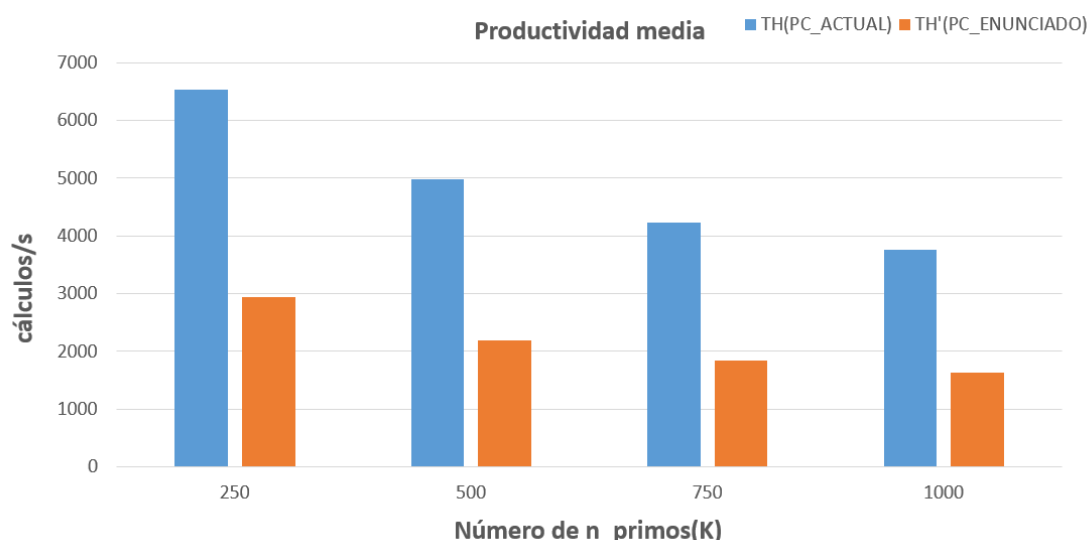
pequeños incrementos que vemos de un nivel a otro son porque el benchmark parece necesitar un poco mas de memoria cuando incrementamos la carga.

### Comparación de sistemas

En este apartado se decidirá si se cambia el sistema actual por el propuesto. Para comparar los 2 sistemas se analizarán una serie de gráficos y tablas. La información que aparece en estas tablas/gráficos servirá para llegar a una conclusión.

N_primos(K)	R(PC_ACTUAL)	R'(PC_ENUNCIADO)	Speedup(R'/R)	TH(PC_ACTUAL)	TH'(PC_ENUNCIADO)	Speedup(TH'/TH)
250	38,31	85,1	2,221352127	6525,71	2937,68	0,450170173
500	100,52	227,73	2,2655193	4974,13	2195,58	0,441399803
750	177,43	405,66	2,286310094	4227,02	1848,82	0,437381418
1000	265,38	611,51	2,304280654	3768,18	1635,28	0,433970776





Tras ejecutar el benchmark en el sistema del enunciado, se ha obtenido la siguiente tabla:

Carga de trabajo	Tiempo medio de respuesta	Productividad media
250.000	85,10 s	2937,68 cálculos/s
500.000	227,73 s	2195,58 cálculos/s
750.000	405,66 s	1848,82 cálculos/s
1.000.000	611,51 s	1635,28 cálculos/s

Al ver estas tablas y gráficos el analista se hace las siguientes preguntas:

**1. ¿Qué ocurre en este sistema(el del enunciado) cuando la carga de trabajo aumenta?**

Cuando aumenta la carga de trabajo, el tiempo de respuesta aumenta, y la productividad disminuye. Las razones por las que ocurre esto son las mismas que se han explicado cuando se ha hecho un análisis del tiempo de respuesta y la productividad del sistema actual.

**¿En términos de rendimiento, ¿vale la pena cambiar el sistema actual por éste?**

Si nos fijamos en las tablas de ambos sistemas, podemos ver que, para todos los niveles de carga, el sistema actual tiene un mayor rendimiento que el sistema del enunciado. En concreto, para los 4 niveles de carga el sistema actual es más del doble de rápido ( $R'/R = 2,2$  aprox) y también tiene el doble de productividad para todos los niveles ( $\text{Speedup}(\text{TH}'/\text{TH}) = 0,45$  aprox).

Podría darse el caso en que el sistema actual rinde mejor para una carga baja, y a medida que aumentamos la carga enseguida se iguale el rendimiento de ambos sistemas, pero como se puede observar el sistema actual es muy superior desde el primer nivel de carga hasta al último.



Dicho esto, para concluir si vale la pena cambiar el sistema actual por el del enunciado, si nos basamos únicamente en el factor rendimiento, sin tener en cuenta otros factores como la diferencia de precio, la respuesta es que no, no vale la pena cambiar el sistema actual.

**Teniendo en cuenta que el posible sistema futuro tiene un coste de un 25% más que el sistema actual, ¿vale la pena cambiarlo?**

Dadas las circunstancias, como el sistema del enunciado tiene peor rendimiento que el sistema actual, por supuesto no vale la pena cambiarlo si encima tiene un coste superior, en este caso de un 25%, que es una diferencia de coste notable.

En el caso de que el sistema del enunciado hubiera obtenido mejores resultados que el sistema actual, esta pregunta sería más importante. En ese caso habría que ver la relación que hay entre cuánto aumenta el rendimiento de un sistema a otro, y cuánto aumenta el precio. Por ejemplo, sabiendo que el precio aumenta un 25%, si el sistema del enunciado solo aporta una mejora de un 2 % de rendimiento, entonces no valdría la pena cambiarlo.

Si por el contrario aporta una mejora de un 25%, caso en que habría una relación 1:1 (aumentoCoste:aumentoRendimiento), entonces sí valdría la pena cambiarlo, si la empresa tiene los suficientes fondos para permitírselo.