



## Práctica 4.

*Cuestión 1: Instale la aplicación. ¿Qué comando permite listar los benchmarks disponibles?*

Para listar los benchmark que tenemos hemos de utilizar el comando:  
> `phoronix-test-suite list-tests`

\*\*\*\*\*

*Cuestión 2: Seleccione e instale tres que desee y ejecútelos. Cambie los parámetros de experimentación ¿aprecia cambios en los resultados?*

- ☉ Por ejemplo vamos a instalar con la orden:  
> `phoronix-test-suite install tachyon`

```
Installed: pts/tachyon-1.1.0

System Information
Hardware:
Processor: AMD Phenom II X4 955 @ 3.21GHz (1 Core), Motherboard: Intel 440BX,
Chipset: Intel 440BX/2X/DX, Memory: 495MB, Disk: 21GB VMware Virtual I, Graphics
VMware SVGA II, Audio: Ensoniq ES1371, Network: AMD 79c970

Software:
OS: Ubuntu 12.04, Kernel: 3.2.0-29-generic-pae (i686), Display Driver: vmware,
Compiler: GCC 4.6, File-System: ext4, System Layer: VMware

Would you like to save these test results (Y/n): n

Estimated Run-Time: 8 Minutes

Tachyon 0.98.9:
pts/tachyon-1.1.0
Test 1 of 1
Estimated Trial Run Count: 3
Estimated Time To Completion: 8 Minutes
Started Run 1 @ 12:52:34_
```

el benchmark tachyon para procesadores. Y nos saldría un resultado como el que sigue:

```
Tachyon 0.98.9:
pts/tachyon-1.1.0
Test 1 of 1
Estimated Trial Run Count: 3
Estimated Time To Completion: 8 Minutes
Started Run 1 @ 12:52:34
Started Run 2 @ 12:54:39
Started Run 3 @ 12:56:43 [Std. Dev: 0.09%]

Test Results:
122.5071
122.3458
122.3061

Average: 122.39 Seconds
```



Tenemos una media de 122.39 segundos.

- ⦿ Otro podría ser el de "blogbench":

&> *phoronix-test-suite install blogbench*

para disco. Y tendremos la siguiente salida (teniendo 3 posibilidades):

```
eleion@ubuntu:~$ phoronix-test-suite benchmark blogbench
```

```
Phoronix Test Suite v3.6.1
  Installed: pts/blogbench-1.0.0

BlogBench Test Configuration

Test:
1: Read
2: Write
3: Test All Options
Enter Your Choice: _
```

este tarda un "poquito" más, más incluso de los 48 minutos que pone orientativos.

```
Enter Your Choice: 1

System Information

Hardware:
Processor: AMD Phenom II X4 955 @ 3.21GHz (1 Core), Motherboard: Intel 440BX, Chipset: Intel 440BX/ZX/DX, Memory: 495MB, Disk: 21GB VMware Virtual I, Graphics: VMware SVGA II, Audio: Ensoniq ES1371, Network: AMD 79c970

Software:
OS: Ubuntu 12.04, Kernel: 3.2.0-29-generic-pae (i686), Display Driver: vmware, Compiler: GCC 4.6, File-System: ext4, System Layer: VMware

Would you like to save these test results (Y/n): n

Estimated Run-Time: 48 Minutes

BlogBench 1.0:
pts/blogbench-1.0.0 [Test: Read]
Test 1 of 1
Estimated Trial Run Count: 3
Estimated Time To Completion: 48 Minutes
Started Run 1 @ 13:24:17_
```

En nuestro caso en poco más de 1h y 30min obtendríamos el siguiente resultado:

```
[Std. Dev: 21.50%]
Started Run 4 @ 14:11:45 [Std. Dev: 18.17%]
Started Run 5 @ 14:24:32 [Std. Dev: 16.32%]
Started Run 6 @ 14:39:48 [Std. Dev: 15.01%]

Test Results:
30259
45938
35375
34718
33877
33654

Average: 35637 Final Score
```



- ⦿ Y el último que vamos a probar es otro para procesadores llamado "bork"

&> *phoronix-test-suite benchmark bork*

```
Estimated Run-Time: 7 Minutes

There is not enough space (at /root/.phoronix-test-suite/installed-tests
/pts/bork-1.0.0/) for this test to run.

The following tests failed to properly run:

- pts/bork-1.0.0
```

por desgracia. No hay manera de que nos lo pueda instalar... Aun habiendo borrado todos los otros.

Nos dice que no hay suficiente espacio... Ocupando este 100 veces menos que el blogbench. Y tendiendo memoria libre. (Me ha sido imposible probarlo)

```
eleion@ubuntu:~$ free
              total        used        free      shared    buffers     cached
Mem:           507552       425940         81612          0        37196       312616
-/+ buffers/cache:        76128       431424
Swap:              0           0           0
```

Y como este unos cuantos más que se ha intentado instalar y no había memoria...

\*\*\*\*\*

*Cuestión 3: De los parámetros que le podemos pasar al comando ¿Qué significa -c 30 ? ¿y -n 1000?*

- ⌘ -c 30: La "c" hace referencia a la concurrencia. En definitiva lo que le mandamos es que se van a realizar 30 peticiones al mismo tiempo.
- ⌘ -n 1000: La "n" son las peticiones. Con lo que decimos que mandamos 1000 peticiones que es llevarán a cabo durante el benchmark.

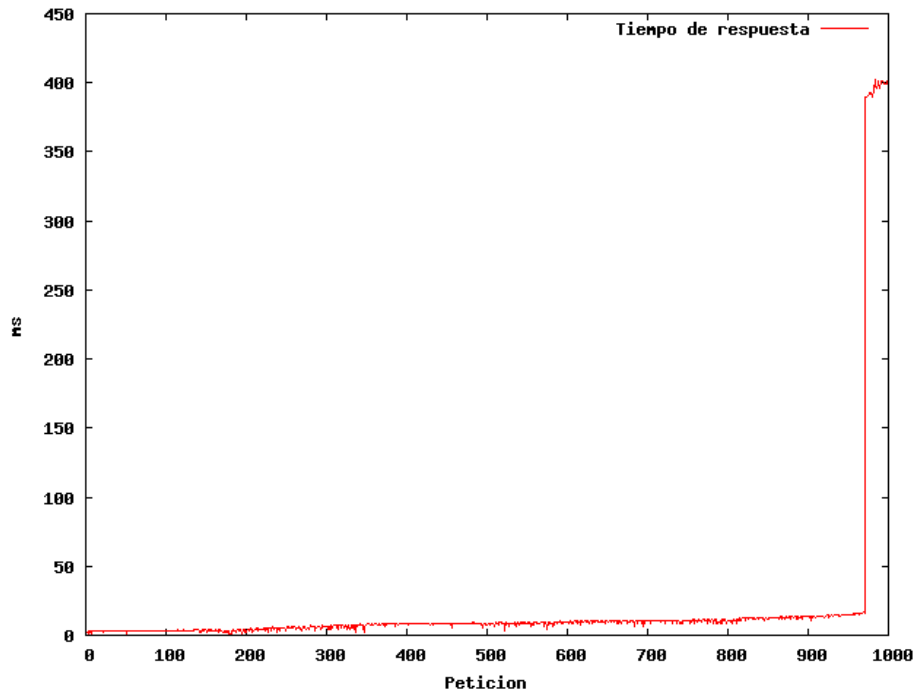
\*\*\*\*\*



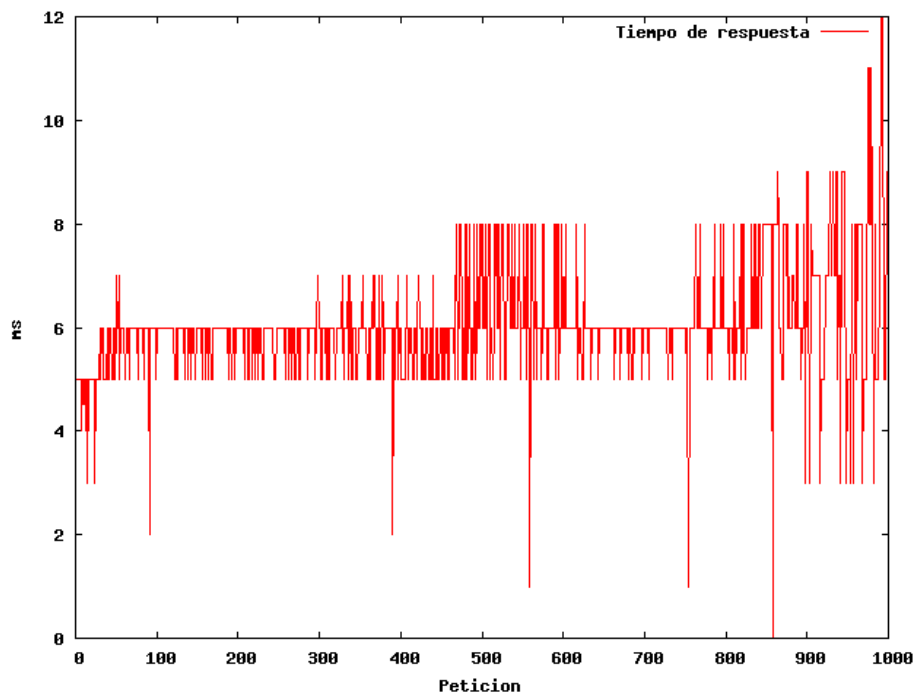
*Cuestión 4: Ejecute ab contra a las tres máquinas virtuales una a una (arrancadas por separado) y muestre las estadísticas. ¿Cuál es la que proporciona mejores resultados?*

Se ha realizado 3 gráficas una por cada servidor, con su tiempo de respuesta. Con el comando "ab -c 30 -n 1000 http://<servidor>"

✚ Windows:

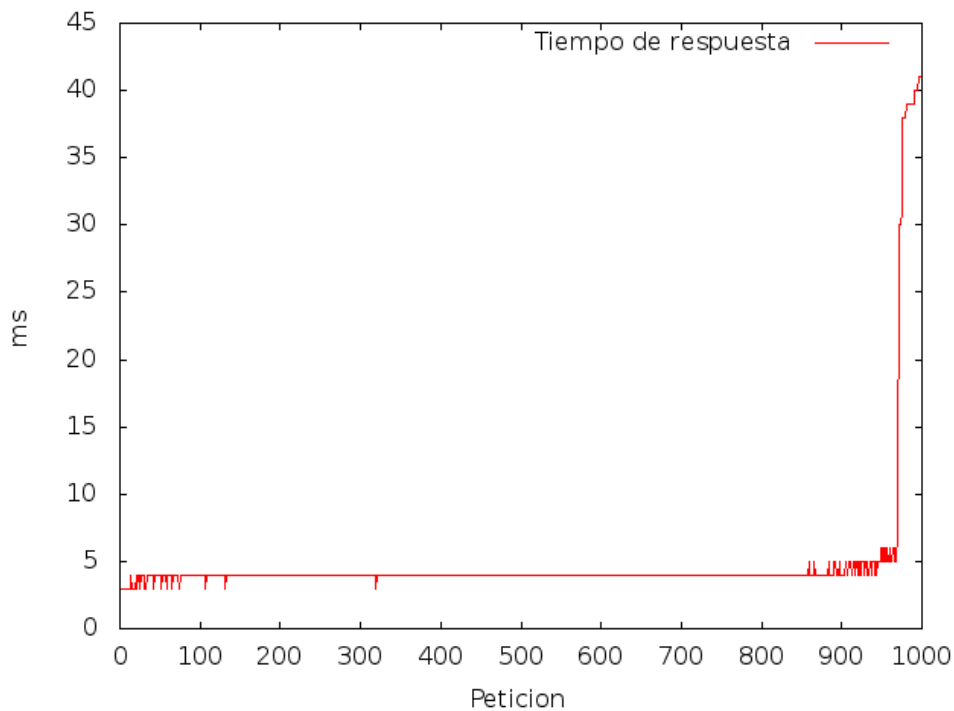


✚ CentOs





#### ✚ Ubuntu



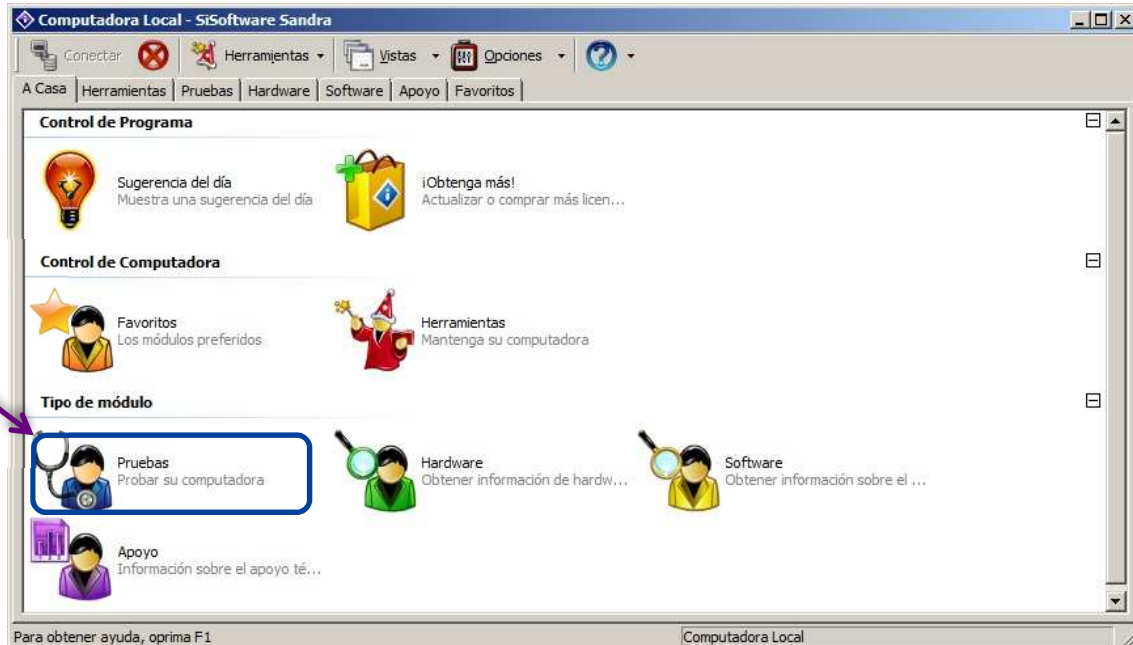
Puede observarse que se han puesto las gráficas en orden, de mayor tiempo a menor tiempo. Se puede ver como el que peor tiempo de respuesta es el de Windows server. (Aunque no se distinga exactamente los valores que toma) Al final ser sobrecarga demasiado y tarda hasta 40-50 veces más de lo normal.

Y después le sigue, con un tiempo un poco más constante la máquina CentOS que aunque también se le envíen 1000 peticiones no llega a sobrecargarse tanto como la de Windows. Pero de media tiene unos tiempo un poco peores que los de Ubuntu.

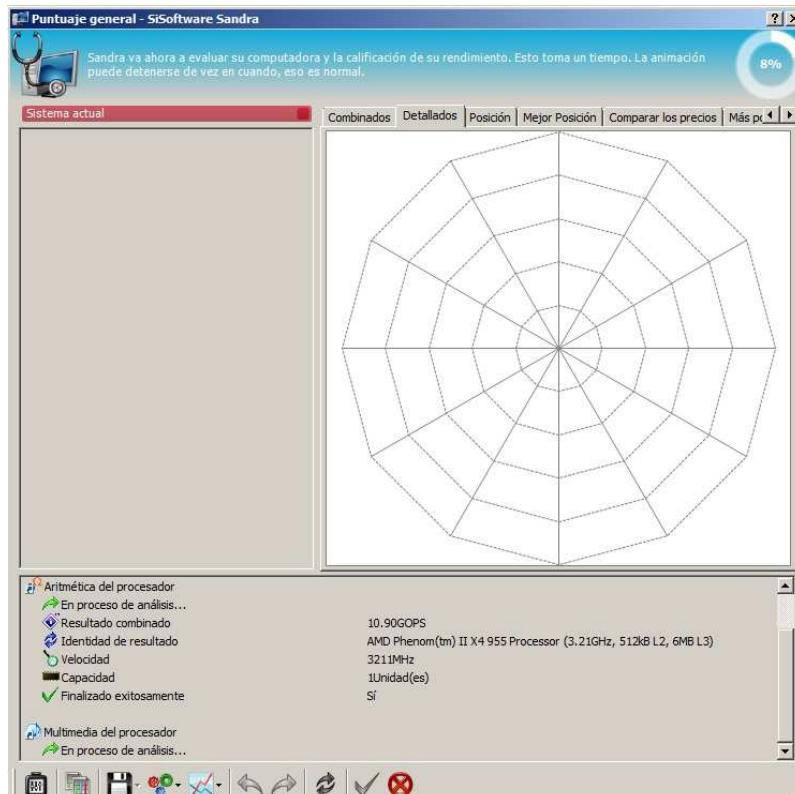
Y finalmente Ubuntu Server tiene unos tiempo constantes inferiores a 5 ms, pero llega un momento en el que se sobrecarga y comienza a tardar hasta 20 veces más lento.

Cuestión 5: Instale la aplicación y realice varios tests a la máquina virtual cambiando los parámetros (número de núcleos y RAM). Muestre capturas de pantalla.

- ✗ Para realizar los test a la máquina virtual lo que vamos a hacer es ir a la sección de pruebas de "SiSoftware Sandra". **Esta primeras pruebas se harán con 1GB de RAM.**

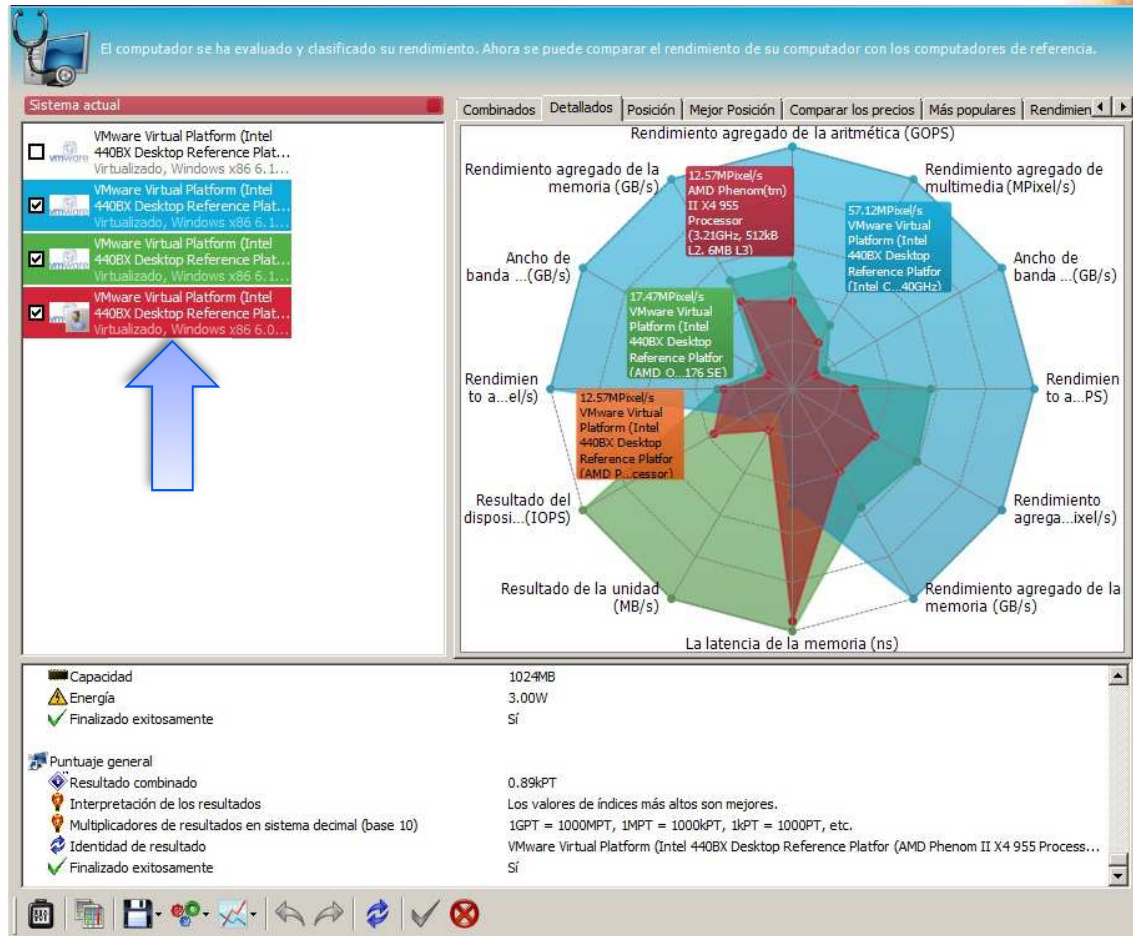


Una vez habiendo esperado lo suficiente para que se realicen las pruebas pertinentes





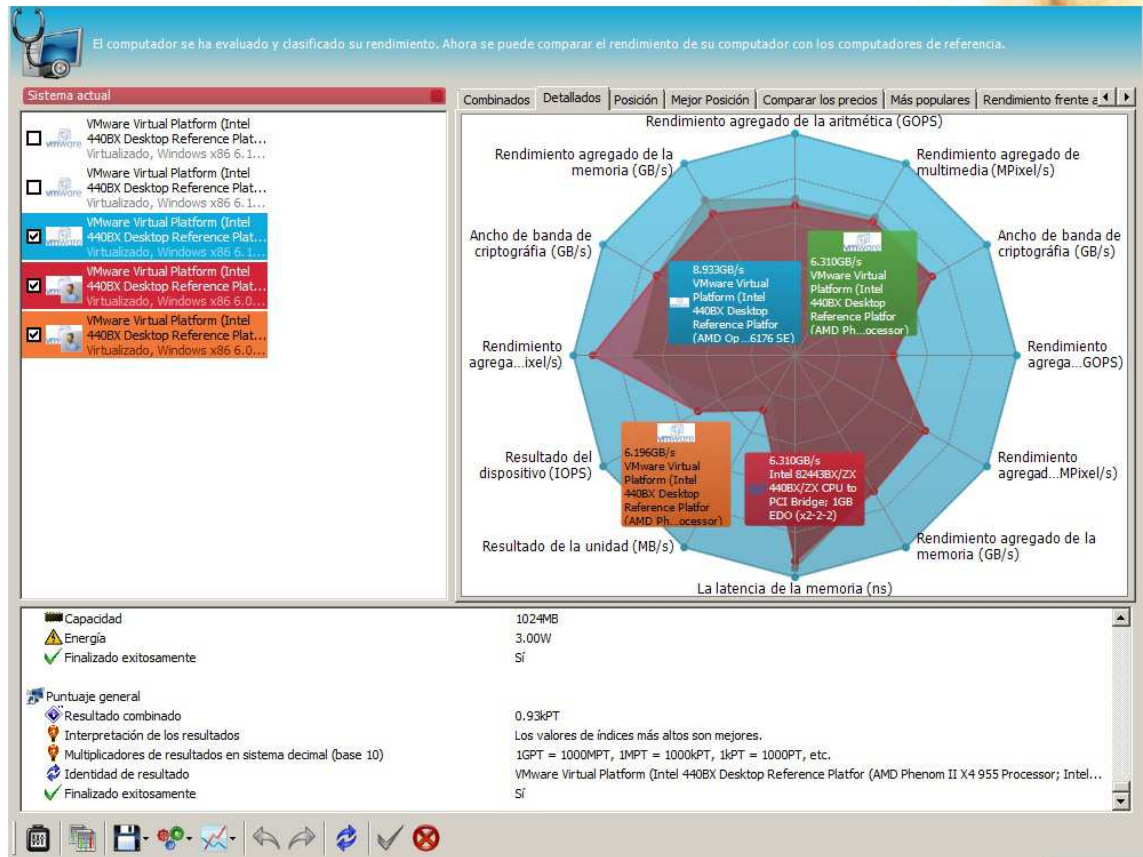
Obtendríamos los resultados, comparándonos con otros tantos computadores. Siendo el nuestro el rojo.



Lo que podemos ver de nuestra máquina es que la latencia de la memoria está bastante bien, comparándolos con los otros dos. Pero refiriéndose al resto de atributos (rendimiento aritmético, rendimiento multimedia, ancho de banda...) deja bastante que desear.



- ❌ La siguiente prueba que vamos a realizar va a ser con 2GB de RAM. Así obtendríamos el siguiente resultado.



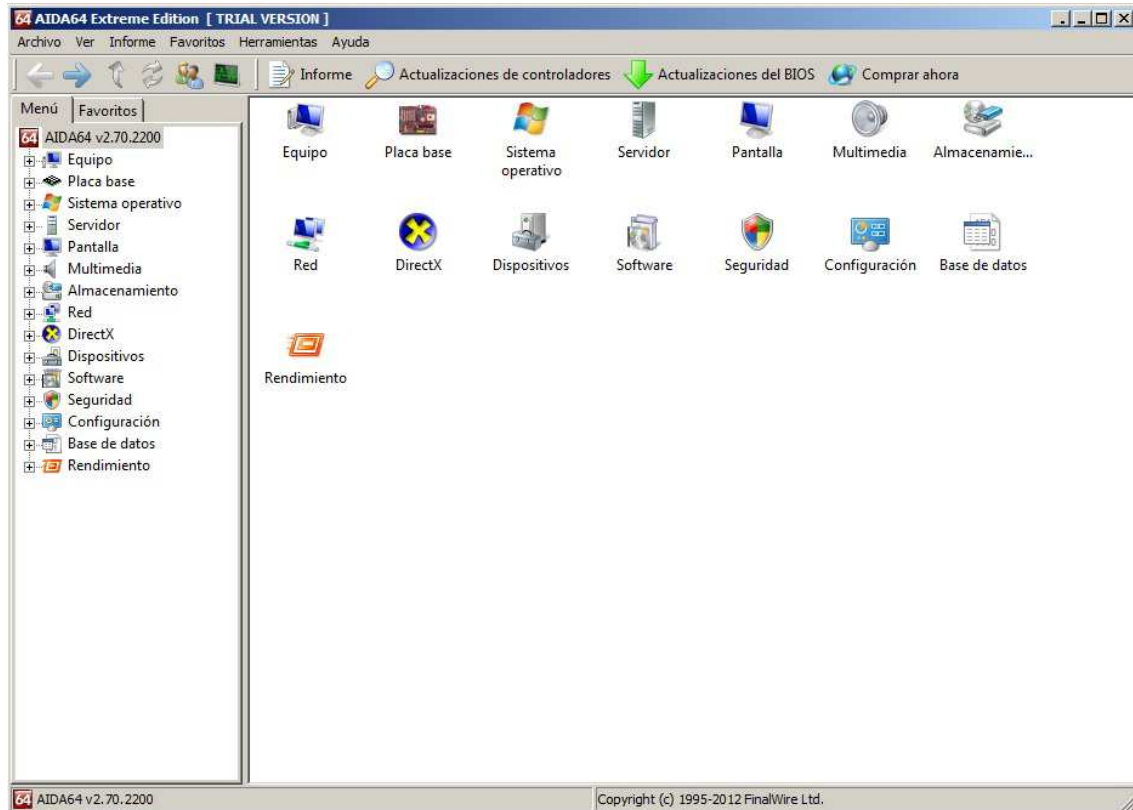
Aquí podemos ver que mejora un poquito (rojo) al anterior (naranja) pero más o menos están igual. No hay una mejora considerable.

.....



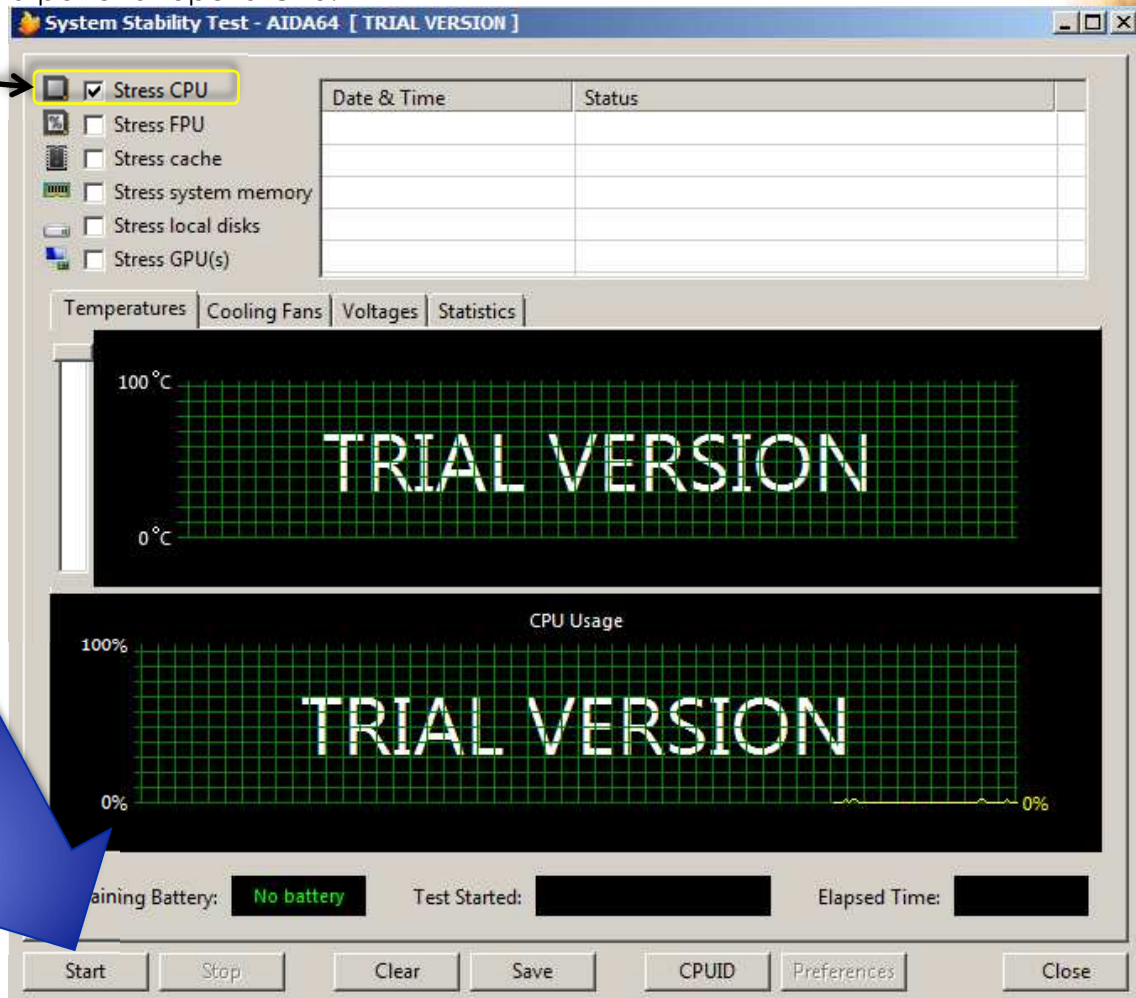
Cuestión 6: *Instale la aplicación y realice varios tests a la máquina virtual cambiando los parámetros (número de núcleos y RAM). Muestre capturas de pantalla de las distintas ejecuciones y comente los resultados.*

Una vez instalado, ejecutando AIDA64 obtendríamos la siguiente ventana:

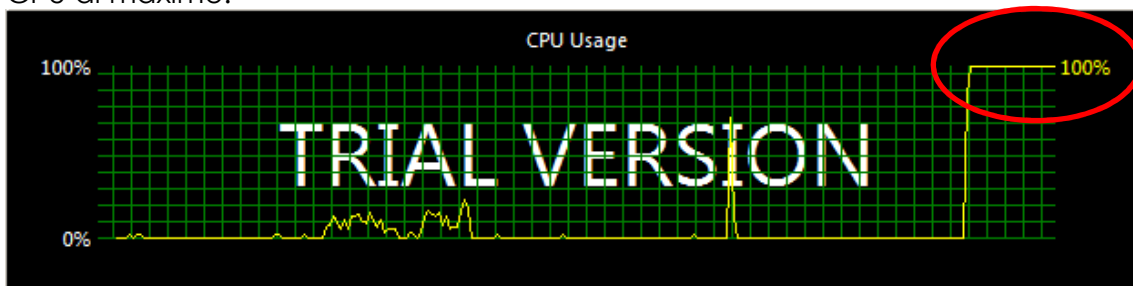




Vamos a realizar un prueba de estabilidad del sistema, para ello vamos a poner a tope la CPU:



A los poco segundos podemos ver como empieza a sobrecargarse la CPU al máximo:



Y se puede notar bastante estabilidad. Ya que no se empieza a ralentizar demasiado la computadora.



Otro test que podríamos hacer es un test para disco. Aquí una vez hecho podríamos ver el resultado que hemos obtenido. La relación MB/s. Y al final obtenemos una media de lectura de unos 14.69 ms.

**AIDA64 Disk Benchmark [ TRIAL VERSION ]**

File Options

Read Test Suite

Test	Block Size	Result	CPU%	Drive
Linear Read (Begin)	32 KB	[ TRIAL ]	18 %	Disk Drive #0 [VMware, VMware Virtual S] (10240 MB)
Linear Read (Middle)	32 KB	77.2 MB/s	20 %	Disk Drive #0 [VMware, VMware Virtual S] (10240 MB)
Linear Read (End)	32 KB	[ TRIAL ]	13 %	Disk Drive #0 [VMware, VMware Virtual S] (10240 MB)
Random Read	32 KB	64.8 MB/s	30 %	Disk Drive #0 [VMware, VMware Virtual S] (10240 MB)
Buffered Read	1 MB	2017.3 MB/s	100 %	Disk Drive #0 [VMware, VMware Virtual S] (10240 MB)
Average Read Access		14.69 ms	2 %	Disk Drive #0 [VMware, VMware Virtual S] (10240 MB)

Read Test Suite [v] Disk Drive #0 [VMware, VMware Virtual S] (10240 MB) [v]

Start Stop Save Clear

Finalmente una última prueba que vamos a hacer es con la Caché y con la memoria. Una vez hecho el test obtendríamos el siguiente ventanal:

**AIDA64 Cache & Memory Benchmark**

	Read	Write	Copy	Latency
Memory	7644 MB/s	TRIAL VERSION	TRIAL VERSION	-6.4 ns
L1 Cache	TRIAL VERSION	50853 MB/s	18695 MB/s	1.0 ns
L2 Cache	14406 MB/s	14380 MB/s	15911 MB/s	TRIAL VERSION
L3 Cache	9458 MB/s	8701 MB/s	9928 MB/s	TRIAL VERSION
CPU Type	AMD Athlon II 15560 (Socket AM2+)			
CPU Clock	528.1 MHz			
CPU FSB	66.0 MHz (original: 200 MHz)			
CPU Multiplier	8x	CPU Stepping	RB-C3	
Memory Bus		DRAM:FSB Ratio		
Memory Type				
Chipset	Intel 82440BX/ZX			
Motherboard	[ TRIAL VERSION ]			

AIDA64 v2.70.2200 / BenchDLL 3.0.478-x32 (c) 1995-2012 FinalWire Ltd.

Save Start Benchmark Close



```

/*
 * Mi propio benchmark.
 *
 * Este Benchmark, básicamente de lo que se encarga
 * es de realizar una multiplicación de matrices,
 * con un tamaño relativamente grande.
 *
 * Autor: Luis Baca Ruiz
 */
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>
#include <sys/wait.h>
#include <unistd.h>
#include <errno.h>

#define SIZE (1<<10)
typedef int MATRIZ[SIZE][SIZE];
SIZE

MATRIZ A, B, C;

/**
 * Código para la multiplicación de matrices básico, se multiplican las matrices con los tres bucles.
 */
void multiplicaMatrices(MATRIZ A, MATRIZ B, MATRIZ C){
    int i;
    int j;
    int k;

    for(k = 0; k < SIZE; ++k)
        for(i = 0; i < SIZE; ++i)
            for(j = 0; j < SIZE; ++j)
                C[k][i] += A[k][j] * B[j][i];
}

void crono(void (*func)(), char* msg){
    struct timeval tv1, tv2;
    long tv_usecs;

    gettimeofday(&tv1, NULL);
    func(A, B, C);
    gettimeofday(&tv2, NULL);

    tv_usecs = (tv2.tv_sec - tv1.tv_sec) * 1E6 +
               (tv2.tv_usec - tv1.tv_usec);

    printf("%s: %9ld us\n", msg, tv_usecs);
}

int main() {
    int i, j, ini = 0;

    for (i = 0; i < SIZE; i++)
        for (j = 0; j < SIZE; j++) {
            A[i][j] = (ini);
            B[i][j] = (ini);
            ini = (ini + 1) % 100;
        }

    crono(multiplicaMatrices, "Multiplicación de matrices (en lenguaje C)");

    printf("\n");
    exit(0);
}

```



Se va a realizar la comparación entre el Ubuntu Server y el CentOS.<sup>1</sup> En dos ámbitos. El primero de ellos es con los ordenadores del laboratorio. Y el segundo con el de casa. (Es decir vamos a comparar distintos SO y además distintos hardware a la vez).

Sistema Operativo	Tiempo ( $\mu\text{seg}$ )
Ubuntu Server (Laboratorio)	56082373
	57161956
	56249765
	59709790
Media	<b>57300971</b>

Sistema Operativo	Tiempo ( $\mu\text{seg}$ )
CentOs(Laboratorio)	62697724
	61250495
	64458759
	60837548
Media	<b>62311131</b>

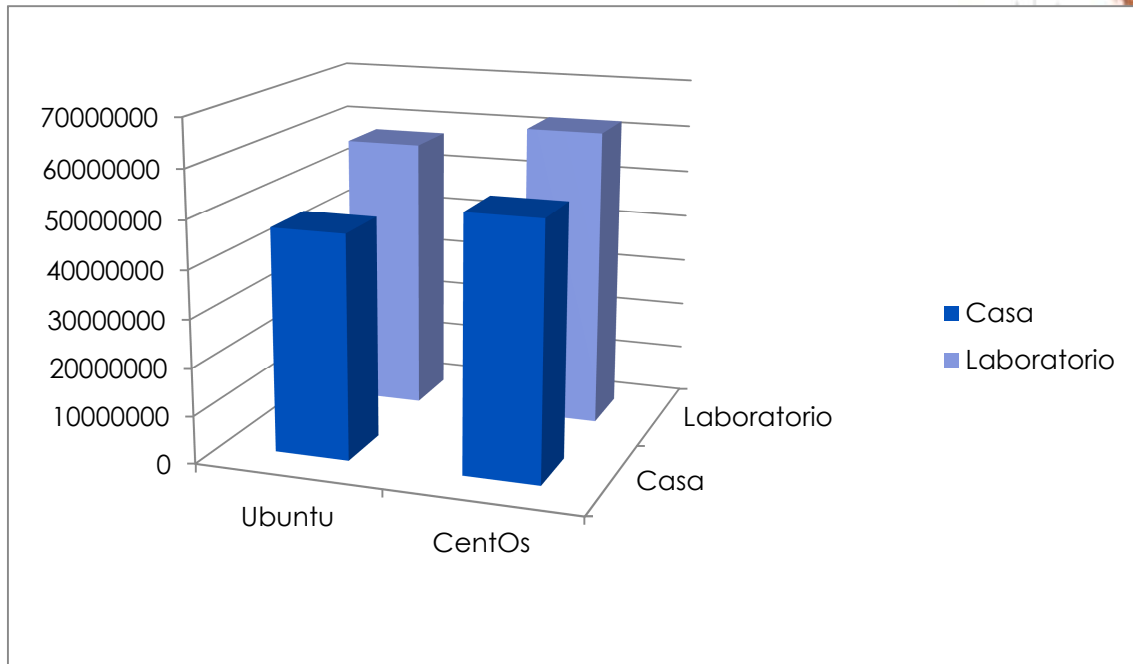
Sistema Operativo	Tiempo ( $\mu\text{seg}$ )
Ubuntu Server (Casa)	47737625
	45241420
	46756835
	47476759
Media	<b>46803159</b>

Sistema Operativo	Tiempo ( $\mu\text{seg}$ )
CentOs(Casa)	52494690
	53781543
	52932611
	52569261
Media	<b>52944526</b>

---

<sup>1</sup> Porque para intentar ejecutar/compilar el código en Windows hace falta configurar mil cosas, instalar el compilador... luego no están las librerías y hay que configurarlas...





Podemos observar que el computador de casa es bastante más rápido que el de laboratorio. Además también se ve que el Ubuntu Server es más rápido que el CentOs en cualquier caso.

\*\*\*\*\*