

Benchmark

Cuestiones de la Práctica 4

LOTHAR SOTO PALMA
Universidad de Granada
28 de mayo de 2015

Índice

Información	1
Cuestión 1	1
Cuestión 2	3
Cuestión 3	6

Índice de figuras

1.	Lista de Benchmarks de phoronix.	2
2.	Resultados benchmark apache.	3
3.	Estado final de los discos de sistema.	4
4.	Benchmark de Escritura de memoria.	5
5.	Benchmark de lectura, escritura de la memoria y caches.	5
6.	Gráfica de Tiempo de escritura.	8
7.	Gráfica de Velocidad de escritura.	8
8.	Gráfica de Tiempo de lectura.	9
9.	Gráfica de Velocidad de lectura.	10
10.	Gráfica de Tiempo de escritura.	11
11.	Gráfica de Velocidad de escritura.	11
12.	Gráfica de Tiempo de lectura.	12
13.	Gráfica de Velocidad de lectura.	13
14.	Diagrama de Flujo del programa que ejecutará la prueba de rendimiento.	14

Información

Cuestión 1

Instale Phoronix Suite, seleccione un benchmark de entre los disponibles, descárguelo y ejecútelo. Describa el propósito del benchmark y su interés en el mismo. Explique razonadamente el significado de los resultados obtenidos.

Solución:

En primer lugar descargamos Phoronix Suite de la web oficial [1] y seleccionaremos el paquete genérico que posteriormente descomprimiremos y ejecutaremos desde terminal el archivo "install-sh". Una vez se ha realizado esto podremos ejecutar Phoronix si tenemos las dependencias de php necesarias segun la web solo es necesario el paquete "php5-cli", sin embargo puede ser que se tengan que instalar las dependencias una a una dependiendo de la que falte como es mi caso, fue necesario la ejecución de:

```
yum install php-json
yum install php-gd
yum install php-xml
yum install php-fpdf
```

Una vez instaladas las dependencias ejecutamos phoronix con el comando "phoronix-test-suite" y completamos algunas opciones de configuración, para comprobar la lista de benchmarks que pueden descargarse y realizarse ejecutamos el comando "phoronix-test-suite list-tests" (figura 1).

Una vez decidamos el benchmark que vamos a utilizar, en nuestro caso usaremos "pts/apache", por lo que es necesario aplicar el siguiente comando: "phoronix-test-suite install pts/apache" y comenzara a instalar el benchmark, por último para usarlo solo es necesario el comando "phoronix-test-suite benchmark pts/apache"

```
[root@localhost ~]# phoronix-test-suite list-tests


Phoronix Test Suite v5.6.0
Available Tests

pts/aio-stress          - AIO-Stress          Disk
pts/apache             - Apache Benchmark    System
pts/apitest            - APITest             Graphics
pts/apitrace           - APITrace            Graphics
pts/battery-power-usage - Battery Power Usage System
pts/bioshock-infinite  - BioShock Infinite  Graphics
pts/blake2             - BLAKE2             Processor
pts/blogbench          - BlogBench          Disk
pts/bork               - Bork File Encrypter Processor
pts/botan              - Botan              Processor
pts/build-apache       - Timed Apache Compilation Processor
pts/build-firefox      - Timed Firefox Compilation Processor
pts/build-imagemagick  - Timed ImageMagick Compilation Processor
pts/build-linux-kernel - Timed Linux Kernel Compilation Processor
pts/build-mplayer      - Timed MPlayer Compilation Processor
pts/build-php          - Timed PHP Compilation Processor
pts/build-webkitgtk    - Timed WebKitGTK Compilation Processor
pts/bullet             - Bullet Physics Engine Processor
pts/byte               - BYTE Unix Benchmark Processor
pts/c-ray              - C-Ray              Processor
pts/cachebench         - CacheBench         Processor
pts/cairo-demos        - Cairo Performance Demos Graphics
pts/cairo-perf-trace   - cairo-perf-trace   Graphics
pts/civbe              - Civilization: Beyond Earth Graphics
pts/clomp              - CLOMP              Processor
pts/compilebench       - Compile Bench       Disk
pts/compress-7zip      - 7-Zip Compression  Processor
pts/compress-gzip      - Gzip Compression   Processor
pts/compress-lzma      - LZMA Compression   Processor
pts/compress-pbzip2    - Parallel BZIP2 Compression Processor
```

Figura 1: Lista de Benchmarks de phoronix.

En los resultados obtenemos que se han realizado 4637.51 peticiones por segundo (figura 2), además se muestra el error normal de 72.93 y la desviación típica 3.15 %; Se usan peticiones por segundo debido a que el benchmark de apache consiste en realizar peticiones al servidor HTTP GET (concurrentemente también) para ver cuanto es el tiempo de respuesta, esto permite realizar una comparación de servidores apache, en un futuro incluso se podría realizar un benchmark con distintos protocolos web como HTTP2 y ver que protocolo funciona mejor con apache.

Results Overview

results	
 Phoronix Test Suite	result
Apache Benchmark	4637.51
Standard Error	72.93
Standard Deviation	3.15%
PHORONIX-TEST-SUITE.COM	

Test Results

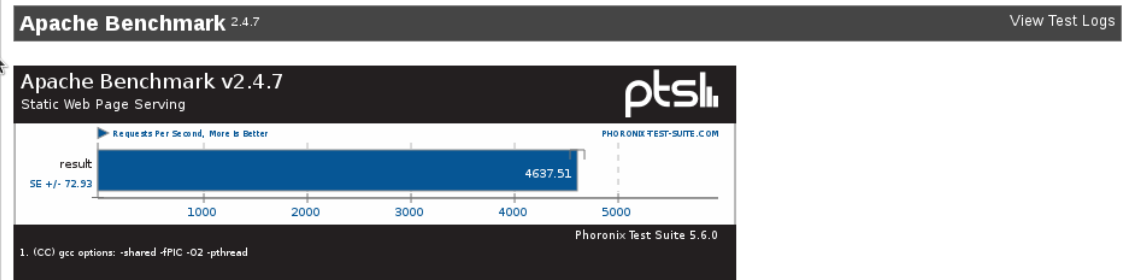


Figura 2: Resultados benchmark apache.

Cuestión 2

Instale una de las aplicaciones de benchmark: SisoftSandra o Aida. Seleccione un benchmark y ejecútelo. Describa el propósito del benchmark y su interés en el mismo. Explique razonadamente el significado de los resultados obtenidos.

Solución:

En el caso de benchmark para windows vamos a instalar AIDA64 que se puede descargar de [2], su instalación es muy sencilla, tan solo ejecutar el archivo ".exe" que se ha descargado y se iniciara la instalación. Una vez instalado obtenemos la vista que podemos apreciar en la figura .

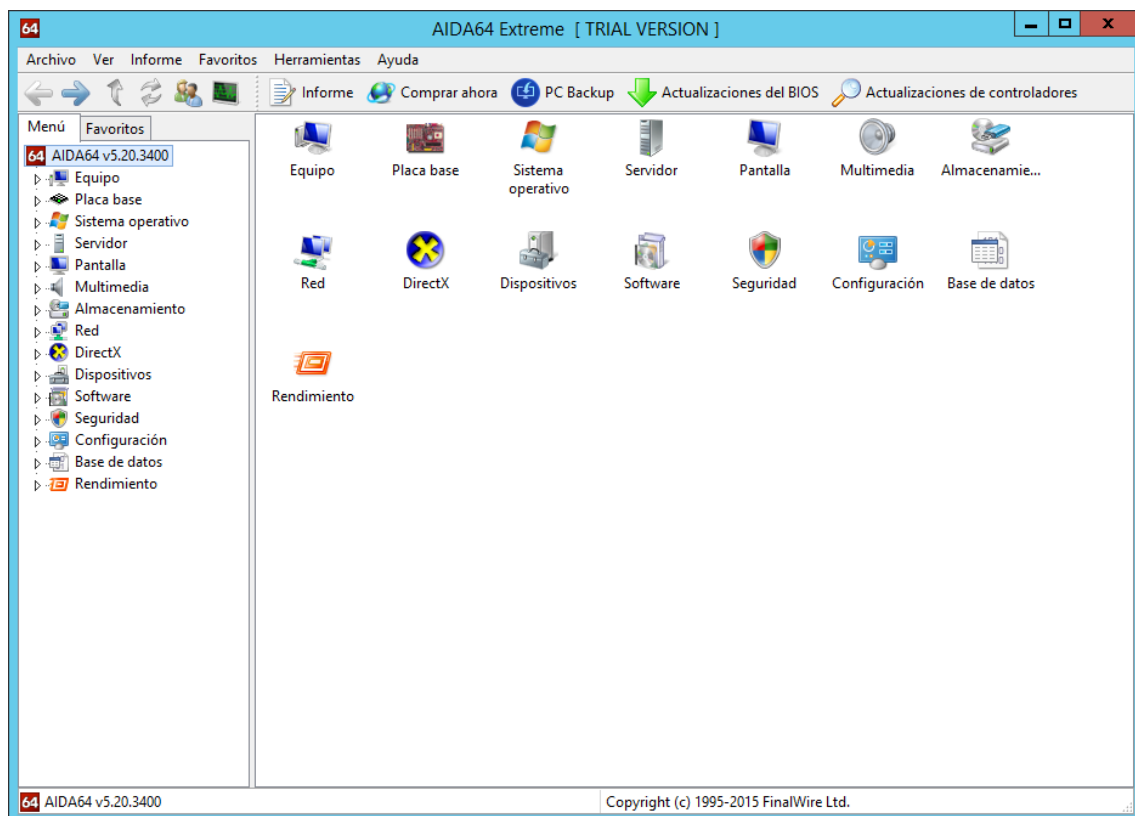


Figura 3: Estado final de los discos de sistema.

AIDA64 proporciona varios benchmark de lectura, escritura de memoria y benchmark de CPU, además tiene opciones para crear informes completos con todos estos benchmark realizados por lo que podemos pulsar en la pestaña de rendimiento con click derecho presionar en informe rápido y elegir un tipo de salida de forma que se realizarán todos los benchmark, estos benchmark nos permiten comparar el rendimiento de nuestros componentes con otros componentes del mercado y hacernos una idea de lo que tenemos en nuestra máquina por ejemplo en el benchmark de velocidad de lectura se miden las velocidades de escritura en memoria de nuestra máquina para poder compararla con el resto de componentes (figura). Además AIDA64 también nos proporciona herramientas como pruebas de rendimiento para cache y memoria (figura) que calcula las velocidades de escritura, lectura de nuestra memoria y caches. (Como es una versión Trial hay datos que no se muestran).

Informe - AIDA64							
Archivo Guardar en archivo Enviar por correo electrónico Enviar a FinalWire Vista previa de impresión Imprimir Cerrar							
21404 MB/s	4x A10-5800K	3800 MHz	Asus F2A55-M	A55 Int.	Dual DDR3-1866	9-11-10-27 CR2	21404 MB/s
21284 MB/s	4x A10-6800K	4100 MHz	Gigabyte GA-F2A85X-UP4	A85X Int.	Dual DDR3-2133	9-11-10-27 CR2	21284 MB/s
21210 MB/s	4x Core i7-965 Extreme HT	3200 MHz	Asus P6T Deluxe	X58	Triple DDR3-1333	9-9-9-24 CR1	21210 MB/s
21110 MB/s	6x Core i7-990X Extreme HT	3466 MHz	Intel DX58SO2	X58	Triple DDR3-1333	9-9-9-24 CR1	21110 MB/s
21006 MB/s	4x A10-7850K	3700 MHz	Gigabyte GA-F2A88XM-D3H	A88X Int.	Dual DDR3-2133	9-11-10-31 CR2	21006 MB/s
19763 MB/s	12x Opteron 2431	2400 MHz	Supermicro H8DI3+-F	SR5690	Unganged Dual DDR2-800R	6-6-6-18 CR1	19763 MB/s
19552 MB/s	4x Core i7-2600 HT	3400 MHz	Asus P8P67	P67	Dual DDR3-1333	9-9-9-24 CR1	19552 MB/s
18834 MB/s	8x Opteron 2378	2400 MHz	Tyan Thunder n3600R	nForcePro-3600	Unganged Dual DDR2-800R	6-6-6-18 CR1	18834 MB/s
17639 MB/s	4x Xeon X3430	2400 MHz	Supermicro X8SIL-F	i3420	Dual DDR3-1333	9-9-9-24 CR1	17639 MB/s
16638 MB/s	4x A8-3850	2900 MHz	Gigabyte GA-A75M-UD2H	A75 Int.	Dual DDR3-1333	9-9-9-24 CR1	16638 MB/s
14730 MB/s	6x Phenom II X6 Black 1100T	3300 MHz	Gigabyte GA-890GPA-UD3H v2	AMD890GX Int.	Unganged Dual DDR3-1333	9-9-9-24 CR2	14730 MB/s
13453 MB/s	Celeron J1900	2000 MHz	Gigabyte GA-J1900N-D3V	BayTrailD Int.	Dual DDR3-1333	9-9-9-24 CR1	13453 MB/s
13118 MB/s	8x Opteron 2344 HE	1700 MHz	Supermicro H8DME-2	nForcePro-3600	Unganged Dual DDR2-667R	5-5-5-15 CR1	13118 MB/s
12592 MB/s	Core i7	2575 MHz	[TRIAL VERSION]	i440FX			12592 MB/s
11534 MB/s	4x Opteron 2210 HE	1800 MHz	Tyan Thunder h2000M	BCM5785	Dual DDR2-600R	5-5-5-15 CR1	11534 MB/s
11178 MB/s	2x Core i5-650 HT	3200 MHz	Supermicro C7SIM-Q	Q57 Int.	Dual DDR3-1333	9-9-9-24 CR1	11178 MB/s
10112 MB/s	4x Phenom II X4 Black 940	3000 MHz	Asus M3N78-EM	GeForce8300 Int.	Ganged Dual DDR2-800	5-5-5-18 CR2	10112 MB/s
9714 MB/s	2x Athlon64 X2 Black 6400+	3200 MHz	MSI K9N SLI Platinum	nForce570SLI	Dual DDR2-800	4-4-4-11 CR1	9714 MB/s

Figura 4: Benchmark de Escritura de memoria.

AIDA64 Cache & Memory Benchmark

	Read	Write	Copy	Latency
Memory	16324 MB/s	TRIAL VERSION	TRIAL VERSION	75.2 ns
L1 Cache	TRIAL VERSION	390209.7 GB/s	390209.7 GB/s	1.3 ns
L2 Cache	249734.2 GB/s	31838 MB/s	249734.2 GB/s	TRIAL VERSION
L3 Cache				
L4 Cache				
CPU Type	Intel Core i7 (Ivy Bridge-MB)			
CPU Stepping	E1/L1/N0/P0			
CPU Clock	2601.4 MHz (original: [TRIAL VERSION] MHz)			
CPU FSB				
CPU Multiplier	North Bridge Clock			
Memory Bus	DRAM:FSB Ratio			
Memory Type				
Chipset	Intel Natoma 82440FX			
Motherboard	[TRIAL VERSION]			

AIDA64 v5.20.3400 / BenchDLL 4.1.633-x64 (c) 1995-2015 FinalWire Ltd.

Save

Start Benchmark

Close

Figura 5: Benchmark de lectura, escritura de la memoria y caches.

Cuestión 3

- a) Diseño del Benchmark: Debe elegir un recurso IT sobre el que plantear su hipótesis. Se valorará especialmente que se opte por elementos de la infraestructura IT (servidores, recursos del SO, hardware, networking, etc.) frente a componentes de desarrollo de software (lenguajes de programación, algoritmos, etc.). En todo caso, el experimento debe poder reproducirse en el aula de prácticas. Por ello, no plantee experimentos que no puedan simularse en el aula. Explique el interés de la hipótesis elegida, así como el de los factores y niveles analizados. Como mínimo debe plantear un experimento de 1 factor con 3 niveles. Describa la carga de trabajo elegida, la forma de registrarla o simularla y cómo alimenta el experimento. Describa las medidas tomadas y tratamiento estadístico aplicado para obtener las métricas. Justifique si la técnica empleada para la obtención de las medidas puede, o no, distorsionar los resultados observados. Finalmente, explique si se cumple o no la hipótesis de acuerdo con el factor de confianza elegido. El alumno deberá justificar, de manera especialmente detallada y razonada, la no obtención de resultados significativos para niveles de confianza estándar del 95 %.
- b) Programación del Benchmark: Programe el benchmark en un lenguaje de su elección. Se valorará que la tecnología elegida y el diseño sean adecuados a los factores a analizar. Mediante un organigrama, describa el funcionamiento del benchmark. El resultado de la ejecución del programa, será, como mínimo, un fichero de texto plano tabulado, en un formato estándar (csv, xml, ...) con las métricas obtenidas para los factores y niveles del experimento. El alumno podrá analizar estos resultados de dos formas:
- Integrando la lógica de análisis en el propio programa, mediante el uso de una librería estadística. Esta será la opción más valorada.
 - Importando los resultados a una herramienta estadística de su elección y realizando el permitente análisis off-line.

En ambos casos, el alumno deberá explicar los resultados obtenidos. El benchmark debe funcionar de forma autónoma, reduciendo al máximo la necesidad de intervención del usuario. Por ejemplo, limitándola al cambio de dispositivo hardware. En todo caso, el programa debe interactuar con el usuario de forma sencilla y clara.

- c) Entregables: Las respuestas al apartado de diseño se realizarán en el habitual documento de memoria de prácticas. El alumno adjuntará, en un archivo comprimido y organizado en subdirectorios, todos los artefactos de desarrollo empleados en la creación del benchmark: código fuente debidamente comentado, documentación de diseño, archivos de datos con la carga de trabajo, etc. Además proporcionará un manual con la organización de los contenidos del archivo comprimido e instrucciones para la instalación y ejecución del programa.

Solución:

- a) El recurso sobre el que se va a plantear la hipótesis después de pensarlo, se hará usando memorias extraíbles USB por motivos más que obvios como pueden ser la falta de tiempo, y la facilidad de implementación y trabajo físico con el ordenador ya que al no disponer de HDD sata u otros no puedo realizar otro tipo de analisis, aunque el analisis implementado sirve también para HDD sata u extraíbles. La idea es hacer una comparativa entre tres elementos, un disco duro extraíble que usa puerto USB TrekStor, una memoria USB Alcor 2.0 y una memoria USB Kingston 3.0, para estudiarlos nos interesa obtener datos acerca de la escritura y la lectura de archivos sean grandes o pequeños por eso se han planteado dos experimentos acordes a esta idea:

- El primer experimento consistirá en medir el tiempo y la velocidad de operaciones de escritura, lectura y una mezcla de dichas operaciones de un archivo con un volumen de 2.0GB.
- El segundo experimento consistirá en medir el tiempo y la velocidad de operaciones de escritura, lectura y una mezcla de dichas operaciones de muchos archivos (en mi caso 100) cada uno con un volumen de 50 MB lo que viene a corresponder a 5.00 GB aproximadamente.

Con respecto a la carga de trabajo, se han elegido estos tamaños puesto que con lo que vamos a trabajar tiene un espacio limitado, y los dispositivos que usan USB 2.0 no son tan rápidos a la hora de escribir archivos grandes por lo que con esos tamaños será más que suficiente para analizar las memorias, para crear dichos archivos hacemos uso de instrucciones de c++ para escribir archivos binarios con el tamaño que se desee. Con respecto a las medidas tomadas, se analizará tanto el tiempo de escritura total de los archivos como la velocidad en MB/s. La toma de medidas implementada puede ser un poco molesta por el tiempo que se demora ya que son necesarias una cantidad de datos para realizar las medias y por tanto para la realización de ambos experimentos puede demorarse alrededor de 90 minutos para que los datos obtenidos sean más fiables. La hipótesis a la que se quiere llegar realizando estas pruebas es: Los discos duros extraíbles tienen mayor velocidad de escritura y lectura que otros dispositivos USB sean 2.0 o 3.0. Para realizar las pruebas se ha realizado el benchmark que se pide implementar, y posteriormente se ha realizado un análisis teórico que se encuentra en la carpeta "Análisis" del benchmark. Vamos a comentar los resultados que me parecen más relevantes:

Vamos a comenzar con el experimento 1:

- **Escritura en dispositivo:** Como podemos observar en la figura 6 y por los datos aportados por el benchmark, el tiempo de Escritura en las memoria USB, sea 2.0 o 3.0, es mucho más grande que en el disco Extraíble, esto puede ser debido a la estructura hardware de cada elemento, ya que las memorias USB son memorias flash y el disco duro usa platos, por lo que escribir en un plato en una determinada sección es posiblemente más rápido que hacerlo en un dispositivo flash.

Tiempos:

USB3.0 DISCOEXTRAIBLE USB2.0

224,827	73,0471	253,679
238,093	63,5761	270,637
240,019	63,2625	299,595
242,304	61,6963	293,076
242,38	67,9739	314,684

Intervalos de confianza(respectivamente):

[228,4385433164, 246,6106566836]

[60,1736212308, 71,6487387692]

[256,3166292447, 316,3517707553]

Velocidades:

USB3.0 DISCOEXTRAIBLE USB2.0

9,10922	28,0367	8,07319
8,6017	32,2133	7,56734
8,53266	32,373	6,83589
8,45218	33,1949	6,98795
8,44954	30,1292	6,50812

Intervalos de confianza(respectivamente):

[8,2866954294, 8,9714245706]

[28,5897085763, 33,7891314237]

[6,4206993818, 7,9682966182]

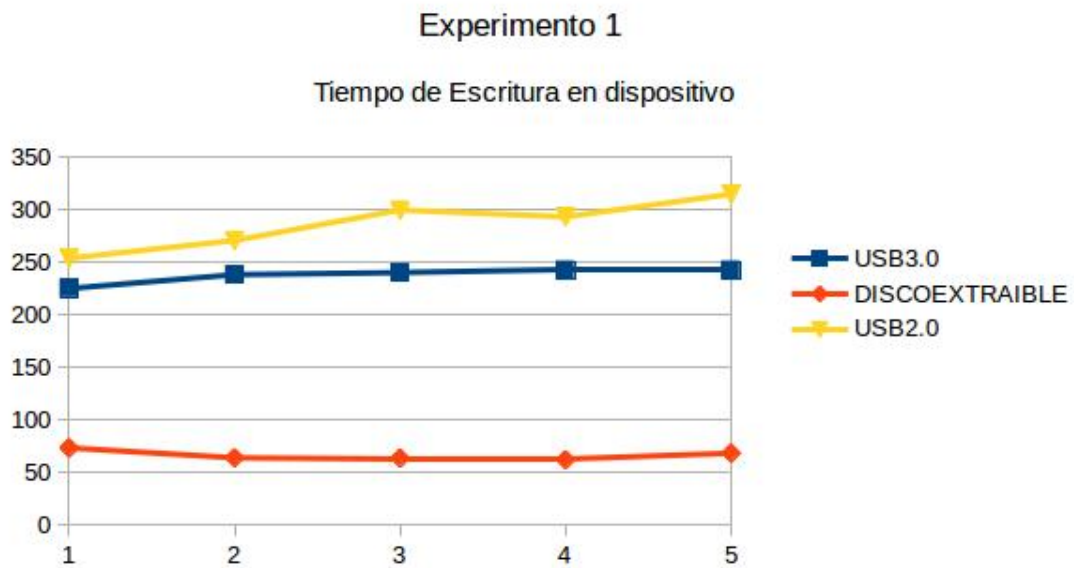


Figura 6: Gráfica de Tiempo de escritura.

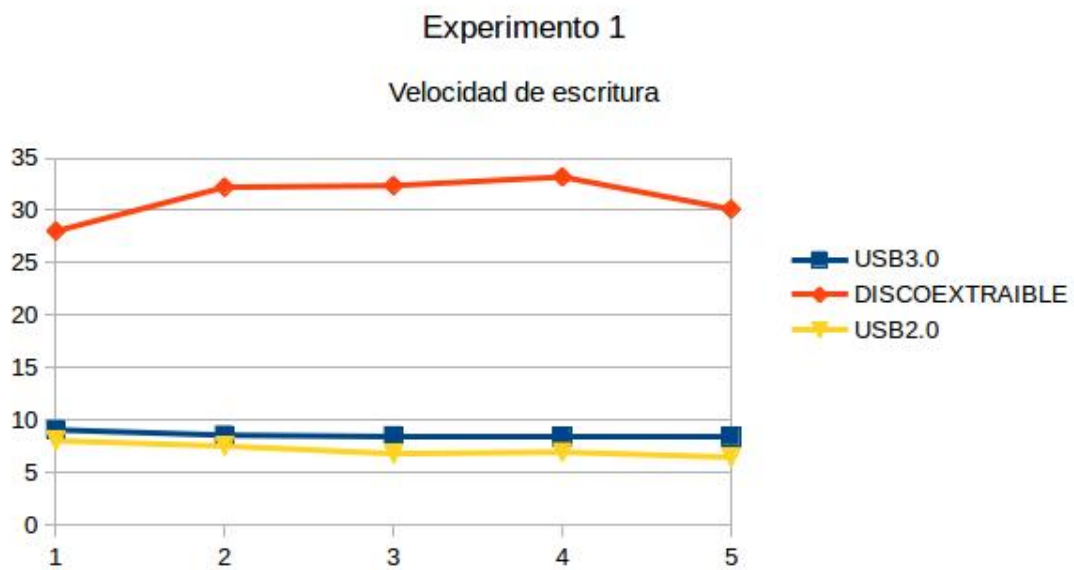


Figura 7: Gráfica de Velocidad de escritura.

- **Lectura en dispositivo:** Sin embargo los datos obtenidos nos dicen que las memorias flash 3.0 son más rápidas a la hora de leer un fichero contenido en la misma, esto puede ser por la velocidad de transferencia de los puertos que están preparados para estas unidades 3.0, y por la tardanza del disco en buscar las secciones. (figura 8)

Tiempos:

USB3.0 DISCOEXTRAIBLE USB2.0

39,2158 85,5879 134,683
 37,97 94,0525 134,744
 41,489 96,4047 141,512
 40,0996 90,4485 140,747
 38,7784 93,6922 142,632

Intervalos de confianza(respectivamente):

[37,8380588981, 41,1830611019]
 [86,8431682756, 97,2311517244]
 [134,0863965121, 143,6408034879]

Velocidades:

USB3.0 DISCOEXTRAIBLE USB2.0

52,2238 23,9286 15,206
 53,9373 21,7751 15,1991
 49,3625 21,2438 14,4723
 51,0728 22,6427 14,551
 52,8129 21,8588 14,3586

Intervalos de confianza(respectivamente):

[49,7124678733, 54,0512521267]
 [20,9942087077, 23,5853912923]
 [14,2457254748, 15,2690745252]

Experimento 1

Tiempo de Lectura de dispositivo

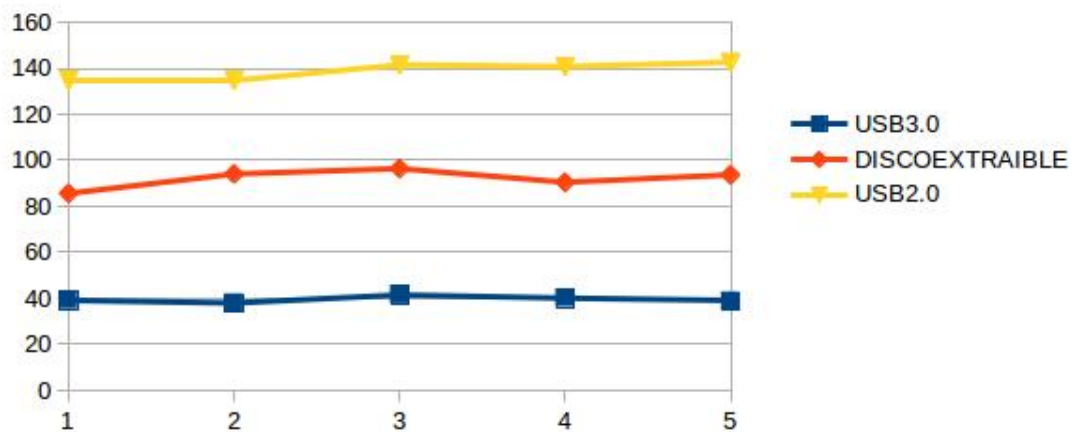


Figura 8: Gráfica de Tiempo de lectura.

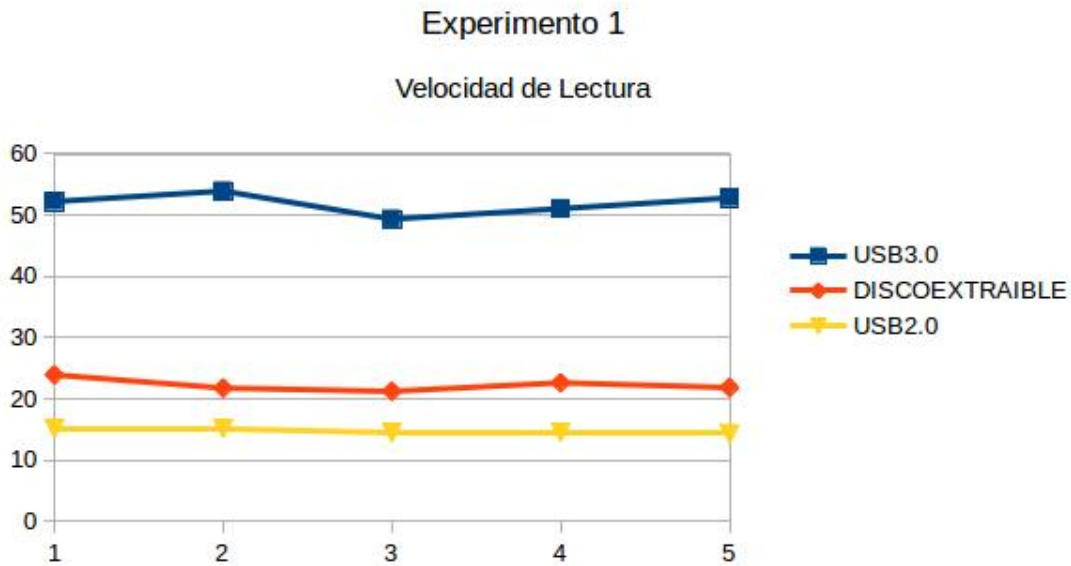


Figura 9: Gráfica de Velocidad de lectura.

- **Escritura/Lectura en dispositivo:** La escritura y lectura es una operación alternada entre las dos anteriores sin embargo, es muy similar a la salida que se produce cuando directamente escribimos, parece que el coste en tiempo de lectura de trozos más pequeños mientras se esta escribiendo no tiene un impacto muy grande en tiempo por tanto el disco es el más rápido.

Vamos a comenzar con el experimento 2:

- **Escritura en dispositivo:** Para la escritura de muchos archivos de tamaño reducido se produce lo mismo que en el experimento 1, el disco duro tiene una estructura que permite una velocidad mayor de escritura pese a que la salida sea 2.0.

Tiempos:

USB3.0 DISCOEXTRAIBLE USB2.0

586,369 157,098 741,888
 613,674 164,436 790,578
 613,188 160,346 750,954
 597,932 154,524 723,0955082
 612,581 170,969 765,9485

Intervalos de confianza(respectivamente):

[589,585895606, 619,911704394]
 [153,4360187715, 169,5131812285]
 [722,8939242407, 786,0916790393]

Velocidades:

USB3.0 DISCOEXTRAIBLE USB2.0

8,52705 31,8273 6,73956
 8,14765 30,407 6,32449
 8,1541 31,1826 6,65819
 8,36215 32,3575 5,23515
 8,16219 29,2451 5,27145

Intervalos de confianza(respectivamente):

[8,0604695533, 8,4807864467]
 [29,4842152052, 32,5235847948]
 [5,1268603187, 6,9646756813]

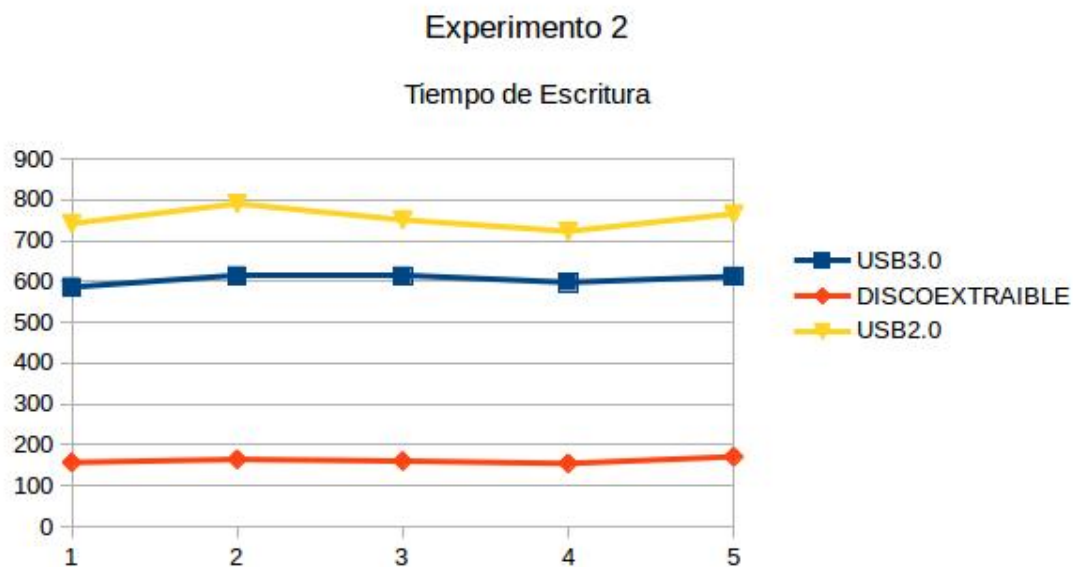


Figura 10: Gráfica de Tiempo de escritura.

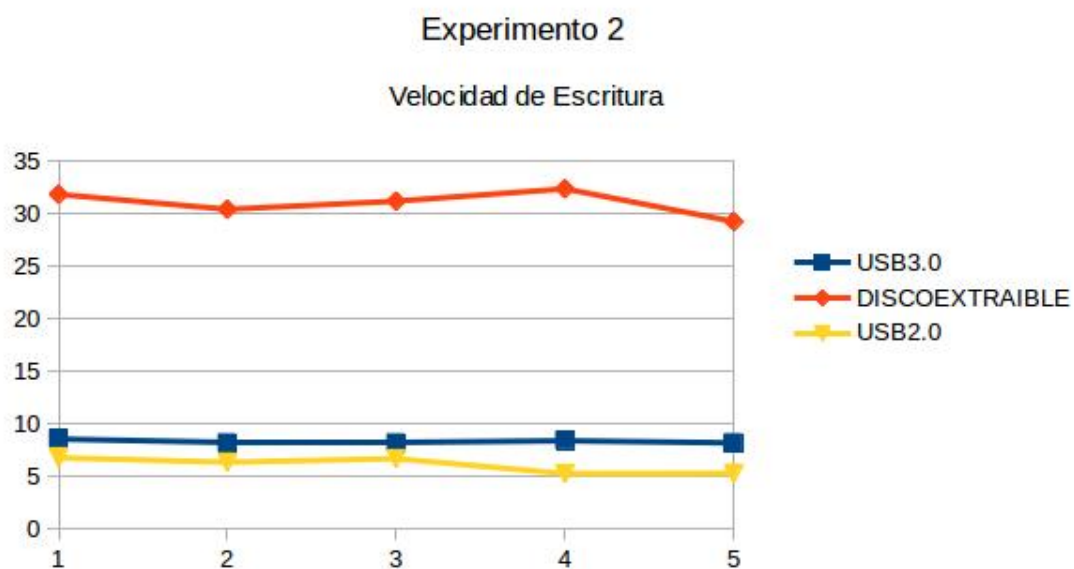


Figura 11: Gráfica de Velocidad de escritura.

- **Lectura en dispositivo:** Ocurre exactamente lo mismo que el experimento 1 el USB 3.0 tiene la mejor velocidad y tiempo de lectura. (figura 12)

Tiempos:

USB3.0 DISCOEXTRAIBLE USB2.0

88,3328 215,326 355,336

91,1757 216,5 358,154

88,687 205,875 275,722

88,0768 215,787 295,82826
87,8435 205,924 314,9563

Intervalos de confianza(respectivamente):

[87,1445900944, 90,5017299056]
[205,0810190598, 218,6837809402]
[274,9104340972, 365,0881899028]

Velocidades:

USB3.0 DISCOEXTRAIBLE USB2.0

56,6041 23,2206 14,0712
54,8392 23,0947 13,9605
56,378 24,2866 18,1342
56,7686 23,171 16,03675
56,9194 24,2808 17,2674

Intervalos de confianza(respectivamente):

[55,2564035274, 57,3473164726]
[22,8459201016, 24,3755598984]
[13,5722208951, 18,2157991049]

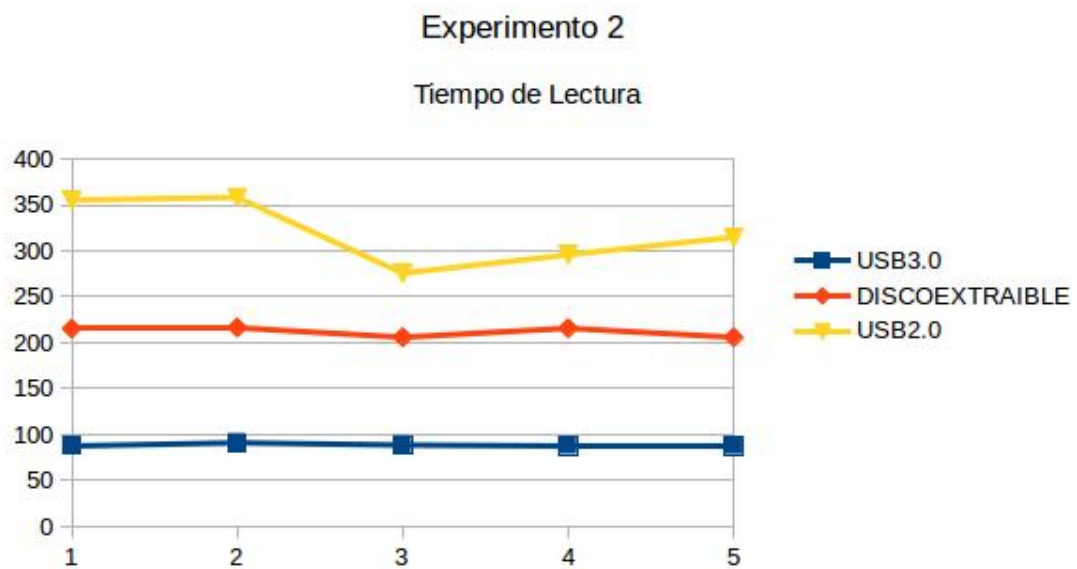


Figura 12: Gráfica de Tiempo de lectura.

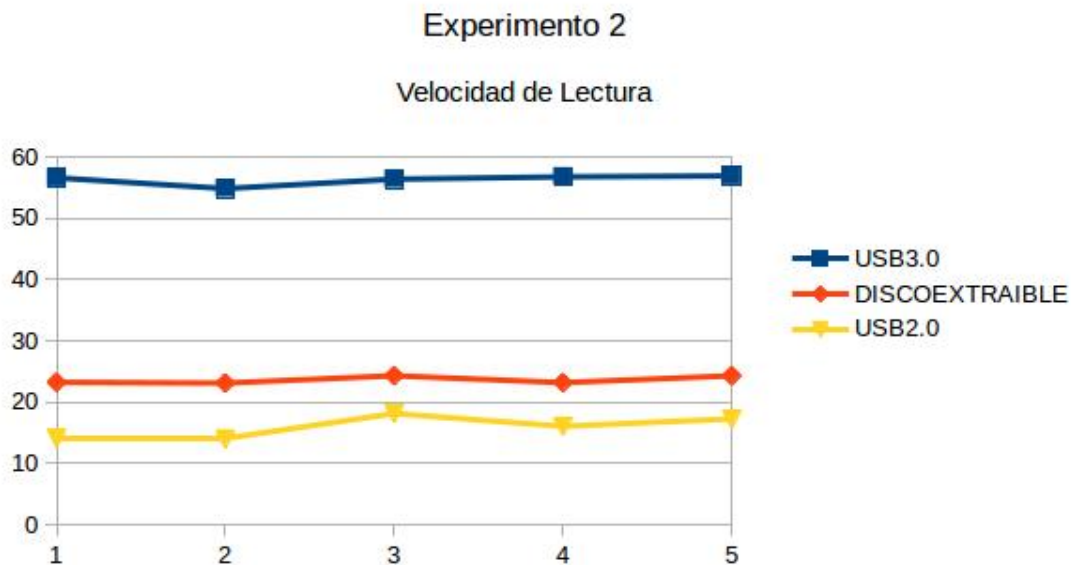


Figura 13: Gráfica de Velocidad de lectura.

- **Escritura/Lectura en dispositivo:** Ocurre exactamente lo mismo que en la escritura debido a que se reduce el tiempo de lectura demasiado por lo que no se produce un impacto muy fuerte en el tiempo total de escritura y lectura.

Conclusión: En mi opinion hemos probado la hipótesis en parte, ya que es cierto que para estos tres dispositivos el disco duro tiene una mayor velocidad de escritura (menor tiempo) que el resto de dispositivos, sin embargo el usb 3.0 lo supera en la velocidad de lectura. Aunque finalmente cuando vamos a usar un disco duro extraible nos interesa normalmente que se produzcan operaciones de lectura y escritura entremezcladas por lo que un disco duro extraible es mejor opción que un usb3.0 por esa misma razón, ya que como se ha indicado antes el impacto de tiempo de la lectura sobre las operaciones de escritura es muy pequeño normalmente y permite mejores tiempo al disco duro extraible.

- b) La programación del Benchmark se llevará a cabo haciendo uso del lenguaje c++, implementando una clase que se dedique a hacer el análisis de unidades de almacenamiento. El análisis de los resultados se llevará a cabo de manera manual por motivos de tiempo, por defecto el programa implementado aporta una versión de los datos en formato cvs, para que pueda ser modificado con un editor estadístico sin embargo es muy probable que tengan que cambiarse las preferencias de lenguaje. El funcionamiento del benchmark es como se describe en la figura 6, en primer lugar se iniciará una instancia de la clase benchmark que será la que ejecute los tests, introducimos el número del experimento que queremos realizar en el caso de introducir uno que no exista se volverá a preguntar, es importante mencionar que este diagrama se producirá tantas veces como niveles se quieran hacer, en nuestro caso 3, después de haber seleccionado el experimento se preguntará cual es el nombre del dispositivo conectado al usb y posteriormente se realizarán los tres tests de escritura, lectura y una alternada de ambas operaciones. Estos tests se realizarán 5 veces y por último se calculará la velocidad a partir del tiempo y se copiarán los datos a un archivo cvs. El benchmark implementado actúa de forma independiente, tan solo se pide al usuario

introducir el nombre del dispositivo a analizar cada cierto tiempo.

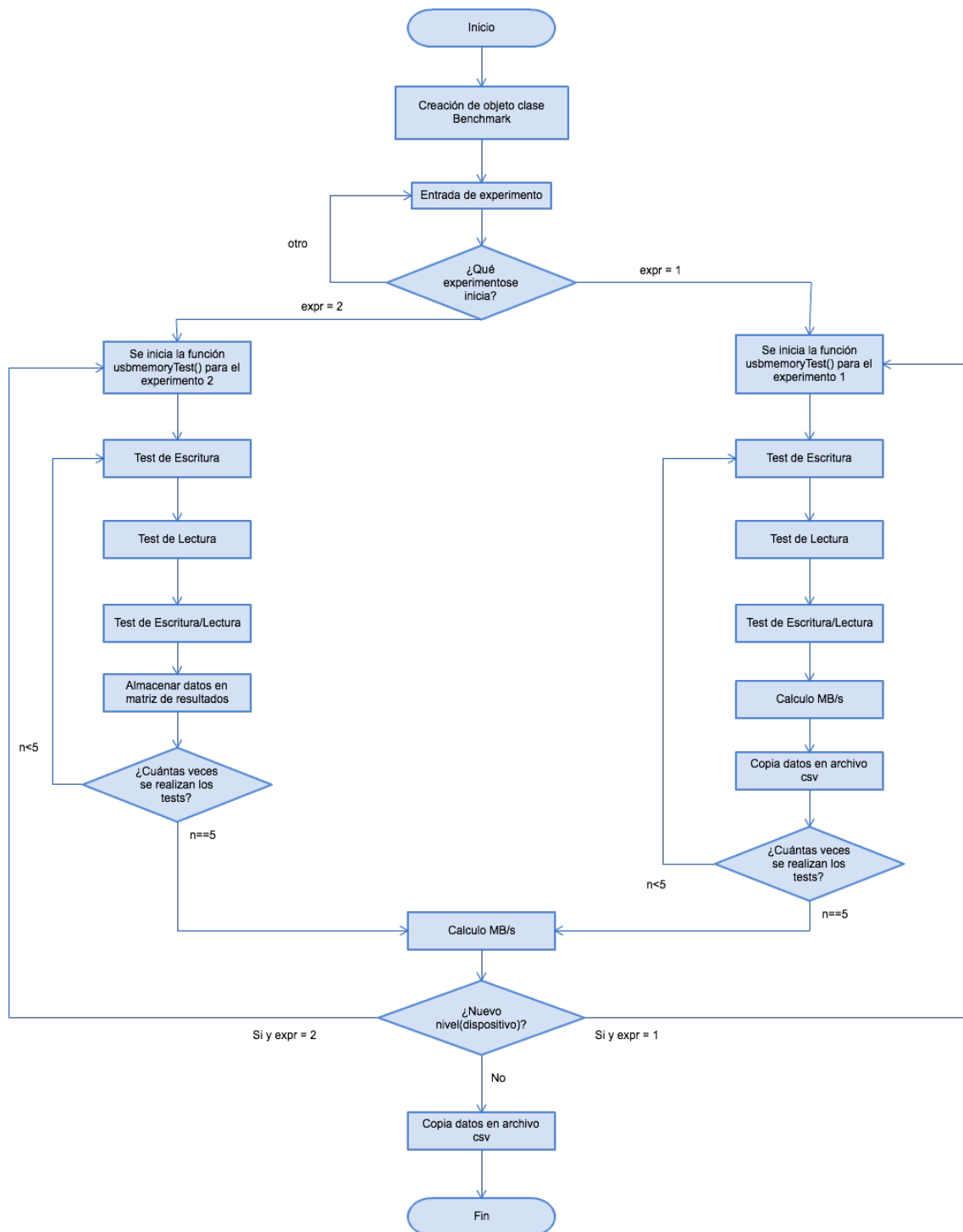


Figura 14: Diagrama de Flujo del programa que ejecutará la prueba de rendimiento.

c) Esquema de subdirectorios del archivo comprimido:

- o benchmark

- ◇ Análisis
 - Exp1.ods
 - Exp2.ods
 - Exp1.csv
 - Exp2.csv
- ◇ bin
 - Bench
- ◇ include
 - benchmark.h
- ◇ object
- ◇ src
 - benchmark.cpp
 - main.cpp
- ◇ data.csv
- ◇ infousb

La salida del programa se realiza en data.csv, los archivos de la carpeta Análisis contienen el análisis teórico de los datos resultantes del benchmark en los archivos Exp1.ods y Exp2.ods, la carpeta bin contiene el archivo binario tal y como se espera, include las cabeceras y src el código del programa.

Referencias

1. <http://www.phoronix-test-suite.com/?k=downloads>
2. www.aida64.com/downloads