

TRABAJO DE FINAL DE GRADO

Estudio y evaluación de distribuciones GNU/Linux en la SBC Raspberry Pi 3 Model B

Autor: Daniel Borraz Morón

Consultor: Joaquín López Sánchez-Montañés



Universitat Oberta
de Catalunya

“A ti, Sonia, mi niña, por tu cariño, apoyo, paciencia y comprensión”.

*“A vosotros, mis hijos Nerea y Sergio,
pues no quiero pasar un día más sin dedicaros todo mi tiempo”.*

“A vosotros, mis padres y hermanos, siempre habéis estado ahí”.

“A ti, mi hermano, desde donde estés, me has ayudado a ser mejor persona”.

“Sois mi fuerza, os lo debo todo”.

RESUMEN

Con la llegada a la escena de la versión de Raspberry Pi 3 Model B. en la que se actualiza y produce una notable mejora de recursos y rendimiento, se abre una gran cantidad de posibilidades en las que podemos emplear gran variedad de sistemas operativos GNU/Linux en esta diminuta y simple placa de ordenador como de un ordenador de escritorio se tratara.

Se planificará con minuciosidad y se dividirá el proceso en diversas fases dentro de un ciclo de investigación-acción con realimentación constante, con el fin de mejorar la práctica sobre nuestro propósito indicado.

Y con este propósito nos lanzamos a investigar y esclarecer de entre un abanico de sistemas operativos seleccionados, cual es el que mejor se comporta frente a esta placa de reducidas dimensiones.

Además emplearemos la práctica conocida como *overclocking* con el que intentaremos mejorar el rendimiento dentro de las posibilidades técnicas de ésta versión de placa estudiada. Posteriormente sobre estos sistemas, realizaremos un análisis exhaustivo de las aplicaciones instaladas, recursos consumidos, fluidez y estabilidad. Además se realizarán pruebas de esfuerzo sobre los principales componentes electrónicos como es la memoria, la CPU, la GPU y el disco de sistema en condiciones normales y con *overclock*.

Finalmente con los datos y experiencias obtenidas de todo el proceso, dispondremos de información suficiente para poder realizar una comparativa y demostrar cual o cuales de los sistemas operativos de la plataforma GNU/Linux para Raspberry Pi Model B son los mejores para emplear como una estación de trabajo.

Palabras clave: GNU, Linux, Raspberry, SBC, comparativa, rendimiento, sistema, operativo, núcleo

ABSTRACT

The arrival on the scene of the Raspberry Pi 3 Model B. version, which one it's updated and produces a remarkable improvement in resources and performance, a great amount of possibilities opens up to us in which we can use a great variety of GNU/Linux operating systems in this tiny and simple computer board as a desktop computer.

The process will be carefully planned and divided into several phases within a cycle of research-action with constant feedback, in order to improve practice on our stated purpose.

And with this purpose in mind, we set out to investigate and clarify, from a range of selected operating systems, which is the one that best behaves in front of this small board.

We will also use the practice known as *overclocking* to try to improve performance within the technical possibilities of this version of the studied board. Subsequently on these systems, we will carry out an exhaustive analysis of the installed applications, resources consumed, fluidity and stability. In addition, stress tests will be carried out on the main electronic components such as memory, CPU, GPU and system disk under normal and overclocked conditions.

Finally, with the data and experiences obtained from the whole process, we will have enough information to be able to make a comparison and demonstrate which ones of the operating systems of the GNU/Linux platform for Raspberry Pi Model B are the best to use as a workstation.

Keywords: GNU, Linux, Raspberry, SBC, comparative, performance, system, operating, system, core

Índice de contenidos

1. Definición general del proyecto	5
1.1 Introducción, motivación, contexto y objetivos.....	5
1.2 Investigación, Implementación y Evaluación de la propuesta	6
1.2.1 Planificación e Investigación.....	7
1.2.2 Puesta en Acción: Práctica, Desarrollo y Observación.....	7
1.2.3 Análisis y Reflexión sobre actividad práctica	7
1.2.4 Evaluación y conclusiones obtenidas de la práctica	7
1.3 Estructura del Trabajo Final de Grado	7
1.4 Planificación y temporalización del TFG.....	8
1.4.1 Listado de tareas a realizar	8
1.4.2 Planificación temporal del proyecto.....	9
1.4.3 Diagrama de Gantt.....	11
2. Planificación inicial: Conceptos, prerequisites e investigación	12
2.1 Aproximación conceptual y contextualización de la propuesta	12
2.2 Investigación, Implementación y Evaluación de las distribuciones GNU/Linux para Raspberry Pi 3 Model B ..	15
2.2.1 Materiales y Recursos para el desempeño de las pruebas.....	15
2.2.2 Criterios y características previas de selección de las distribuciones.....	15
2.2.3 Aplicaciones de control y medida de las pruebas de rendimiento	17
2.2.4. Estudio y Selección de Distribuciones de Sistemas Operativos GNU/Linux	19
2.2.4.1 Sistema Operativo Raspbian Stretch with Desktop	20
2.2.4.2 Sistema Operativo Ubuntu Mate 16.04 LTS Final Release Xenial.....	21
2.2.4.3 Sistema Operativo OpenSuse 42.3 Tumbleweed	22
2.2.4.4 Sistema Operativo ArchLinux ARM.....	24
2.2.4.5 Sistema Operativo Fedora 27 ARM	25
3. Puesta en acción: Fase de desarrollo y pruebas de las distribuciones	27
3.1. Aproximación y estrategias a llevar en el análisis de las distribuciones	27
3.1.1. Aproximación real al problema	27
3.1.2. Estrategias a realizar.....	27
3.2. Análisis y pruebas del SO Raspbian Stretch with Desktop de 32 bits.....	28
3.2.1. Pruebas con parámetros de hardware sin modificar.....	29
3.2.2. Pruebas con parámetros de hardware modificados.....	32
3.3 Análisis y pruebas del SO Ubuntu Mate 16.04 LTS Final Release Xenial	34
3.3.1. Pruebas con parámetros de hardware sin modificar.....	36
3.3.2. Pruebas con parámetros de hardware sin modificar.....	38
3.4 Análisis y pruebas del SO OpenSuse 42.3 Tumbleweed de 64 bits	40
3.4.1. Pruebas con parámetros de hardware sin modificar.....	42
3.4.2. Pruebas con parámetros de hardware sin modificar.....	45
3.5 Análisis y pruebas del SO Sistema Operativo ArchLinux ARM de 32 bits.....	46
3.5.1. Pruebas con parámetros de hardware sin modificar.....	48
3.5.2. Pruebas con parámetros de hardware sin modificar.....	50
3.6 Análisis y pruebas del SO Sistema Operativo ArchLinux ARM de 64 bits.....	51
3.7 Análisis y pruebas del SO Sistema Operativo Fedora 27 ARM de 64 bits	53
3.7.1. Pruebas con parámetros de hardware sin modificar.....	56
3.7.2. Pruebas con parámetros de hardware sin modificar.....	57
4. Comparativa, Reflexión y Análisis de los resultados	58
4.1. Comparativa de los resultados obtenidos en la fase de pruebas	59
4.1.1. Comparativa de datos aplicaciones preinstaladas de los SSOO	59
4.1.2. Comparativa de datos de experiencia de usuario de los SSOO	59
4.1.3. Comparativa de datos y variables del entorno de los SSOO.....	60
4.1.4. Comparativa de pruebas de rendimiento de los SSOO	61
4.2. Reflexión y selección de candidatos.....	65
5. Futuras mejoras y conclusiones finales.....	66
5.1. Futuras mejoras y conclusiones finales.....	66
5.2. Conclusiones finales sobre el estudio realizado.....	67
6. Bibliografía y webgrafía de Referencia	68

Índice de Tablas de contenidos

Tabla 1. Tabla de fechas de entrega del TFG	9
Tabla 2. Tabla de planificación temporal del TFG.....	10
Tabla 3. Tabla de características de Raspberry Pi 3 Model B	14
Tabla 4. Tabla de configuración de frecuencias de Raspberry Pi 3 Model B	17
Tabla 5. Tabla de características del SO Raspbian	21
Tabla 6. Tabla de características del SO Ubuntu Mate	22
Tabla 7. Tabla de características del SO OpenSuse	23
Tabla 8. Tabla de características del SO ArchLinux ARM	24
Tabla 9. Tabla de características del SO Fedora 27	26
Tabla 10. Tabla con la lista de aplicaciones preinstaladas en Raspbian	29
Tabla 11. Tabla de datos con las puntuaciones de experiencia de usuario en Raspbian	29
Tabla 12. Tabla de datos y variables del entorno en Raspbian	30
Tabla 13. Tabla de datos obtenidos en las pruebas de rendimiento.....	31
Tabla 14. Tabla de datos con las puntuaciones de experiencia de usuario en Raspbian modificado	32
Tabla 15. Tabla de datos y variables del entorno en Raspbian modificado	32
Tabla 16. Tabla de datos obtenidos en las pruebas de rendimiento en Raspbian modificado	33
Tabla 17. Tabla con la lista de aplicaciones preinstaladas en Ubuntu Mate	35
Tabla 18. Tabla con la lista de Lugares en Ubuntu Mate.....	36
Tabla 19. Tabla con la lista de aplicaciones preinstaladas en Ubuntu Mate	36
Tabla 20. Tabla de datos con las puntuaciones de experiencia de usuario en Ubuntu Mate	37
Tabla 21. Tabla de datos y variables del entorno en Ubuntu Mate	37
Tabla 22. Tabla de datos obtenidos en las pruebas de Ubuntu Mate	37
Tabla 23. Tabla de datos con las puntuaciones de experiencia de usuario en Ubuntu Mate modificado	38
Tabla 24. Tabla de datos y variables del entorno en Ubuntu Mate modificado	38
Tabla 25. Tabla de datos obtenidos en las pruebas de rendimiento en Ubuntu Mate modificado	38
Tabla 26. Tabla con la lista de aplicaciones preinstaladas en OpenSuse	41
Tabla 27. Tabla de datos con las puntuaciones de experiencia de usuario en OpenSuse.....	42
Tabla 28. Tabla de datos y variables del entorno en OpenSuse	42
Tabla 29. Tabla de datos obtenidos en las pruebas de rendimiento en OpenSuse	43
Tabla 30. Tabla de comparativa de pruebas de CPU entre los SSOO OpenSuse y Ubuntu MATE.....	45
Tabla 31. Tabla con la lista de aplicaciones preinstaladas en ArchLinux ARM de 32 bits	47
Tabla 32. Tabla de datos con las puntuaciones de experiencia de usuario en ArchLinux ARM de 32 bits	48
Tabla 33. Tabla de datos y variables del entorno en ArchLinux ARM de 32 bits.....	49
Tabla 34. Tabla de datos obtenidos en las pruebas de rendimiento en ArchLinux ARM de 32 bits	49
Tabla 35. Tabla de datos con las puntuaciones de experiencia de usuario en Ubuntu Mate overclocked.....	50
Tabla 36. Tabla de datos y variables del entorno en ArchLinux ARM de 32 bits overclocked	50
Tabla 37. Tabla de datos obtenidos en las pruebas de rendimiento en ArchLinux ARM de 32 bits overclocked	50
Tabla 38. Tabla con la lista de aplicaciones preinstaladas en Fedora 27 ARM.....	53
Tabla 39. Tabla de datos con las puntuaciones de experiencia de usuario en Fedora 27 ARM.....	56
Tabla 40. Tabla de datos y variables del entorno en Fedora 27 ARM	56
Tabla 41. Tabla de datos obtenidos en las pruebas de rendimiento en Fedora 27 ARM.....	56
Tabla 42. Tabla de datos con las puntuaciones de experiencia de usuario en Fedora 27 ARM overclocked	57
Tabla 43. Tabla de datos y variables del entorno en Fedora 27 ARM overclocked.....	57
Tabla 44. Tabla de datos obtenidos en las pruebas de rendimiento en Fedora 27 ARM overclocked	57

Índice de Figuras

Figura 1. Ciclo de Investigación-Acción	6
Figura 2. Diagrama de Gantt con la temporalización del TFG	11
Figura 3. Placa Raspberry Pi Model 3	13
Figura 4. Ejemplo de captura de la vista principal de RPI-Monitor	18
Figura 5. Ejemplo de captura de la vista de estadísticas de RPI-Monitor	19
Figura 6. Captura de Sysbench CPU con 4 threads.....	34
Figura 7. Captura de Sysbench CPU con 4 threads con Overclock	34
Figura 8. Captura del fichero /boot/config.txt en Ubuntu Mate.....	35
Figura 9. Captura de prueba de rendimiento DD en Ubuntu MATE.....	39
Figura 10. Captura de prueba de rendimiento DD en Ubuntu MATE con overclock.....	39
Figura 11. Captura de prueba de rendimiento HDPARM en Ubuntu MATE.....	39
Figura 12. Captura de prueba de rendimiento HDPARM en Ubuntu MATE con overclock.....	39
Figura 13. Versiones de escritorio para el SO OpenSuse para Raspberry Pi 3.....	40
Figura 14. Captura del Centro de control YaST2 para el SO OpenSuse para Raspberry Pi 3	40
Figura 15. Capturas de Memoria RAM/SWAP consumida en el SO OpenSuse para Raspberry Pi	41
Figura 16. Captura de instalación de herramienta de monitorización en OpenSuse con YaST2.....	41
Figura 17. Captura de resultado de Gmark2 ES en el SO OpenSuse para Raspberry Pi 3.....	44
Figura 18. Captura de resultado de Sysbench con 1 core en el SO OpenSuse para Raspberry Pi 3	44
Figura 19. Captura de resultado de Sysbench con 2 core en el SO OpenSuse para Raspberry Pi 3	44
Figura 20. Captura de resultado de Sysbench con 4 core en el SO OpenSuse para Raspberry Pi 3	45
Figura 21. Captura de pantalla con RPI-Monitor funcionado en Arch Linux de 32 bits	49
Figura 22. Captura con la ejecución del comando uname -mao para mostrar información del SO de 64 bits	52
Figura 23. Captura del escritorio XFCE4 funcionando en Arch Linux de 64 bits.....	52
Figura 24. Captura de aplicación de administración de software dnfdragora para Fedora 27 ARM	54
Figura 25. Extensión de la partición de la tarjeta con GParted en el SO Fedora 27 ARM	55
Figura 26. Captura con la ejecución del comando xxx para la monitorización de la frecuencia de CPU	55
Figura 27. Captura con la ejecución del comando free -m para la monitorización de memoria RAM.....	55
Figura 28. Captura con la ejecución del comando uname -mao para mostrar información del SO Arch Linux	56
Figura 29. Captura con la ejecución del comando uname -mao para mostrar información del SO Fedora 27.....	58
Figura 30. Gráfico comparativo sobre la facilidad de instalación de los Sistemas Operativos estudiados	60
Figura 31. Gráfico comparativo de tiempos de arranque y apagado de los Sistemas Operativos estudiados.	61
Figura 32. Gráfico comparativo de recursos de RAM, Disco de sistema y SWAP de los SSOO estudiados	61
Figura 33. Gráfico comparativo de las pruebas de rendimiento de memoria RAM en Sysbench.....	62
Figura 34. Gráfico comparativo de las pruebas de rendimiento de ficheros con Sysbench.....	62
Figura 35. Gráfico comparativo de las pruebas de rendimiento de CPU con 1 core en Sysbench.....	62
Figura 36. Gráfico comparativo de las pruebas de rendimiento de CPU con 2 cores en Sysbench	62
Figura 37. Gráfico comparativo de las pruebas de rendimiento de CPU con 4 cores en Sysbench.	63
Figura 38. Gráfico comparativo de las temperaturas obtenidas en las pruebas de CPU	63
Figura 39. Gráfico comparativo de las puntuaciones y temperaturas obtenidas con GLMark2.	63
Figura 40. Gráfico comparativo de las pruebas de rendimiento de ficheros con Hdparm (buffer caché).	64
Figura 41. Gráfico comparativo de las pruebas de rendimiento de ficheros con Hdparm (buffer disk).	64
Figura 42. Gráfico comparativo de las pruebas de rendimiento de ficheros con DD.....	64

1. Definición general del proyecto

1.1. Introducción, motivación, contexto y objetivos

En el presente documento presento mi propuesta de Trabajo Final de Grado para el Grado en Ingeniería en Informática, en la que realizaré un estudio y análisis en profundidad de los sistemas operativos dedicados para el modelo de SBC¹ **Raspberry Pi 3 Model B**, persiguiendo como objetivo el encontrar aquellos sistemas que mejor funcionan y que aprovecha al máximo sus cualidades y capacidades para este dispositivo, tanto oficiales como no oficiales.

El hecho y lo que me ha motivado a decantarme por el empleo de este sistema empotrado en concreto, es debido no solamente por la mejora hardware (velocidad de la CPU, aumento de memoria RAM, Wifi y Bluetooth incorporado) que ha tenido sustancialmente a lo largo de todas las versiones que han ido saliendo desde que se lanzó a finales de Febrero de 2012, si no por su “*buen hacer*” en todos los aspectos, y esto lo hace ideal para enfocarlo dentro del análisis y evaluación que deseamos realizar. Cuando nos referimos a su “*buen hacer*”, principalmente se debe a sus características del diseño software basadas en *open source* desarrolladas para éste, lo cual garantiza la gran usabilidad y el éxito tenido a lo largo de los últimos años.

Además es la placa que posee una **grandísima comunidad** de desarrollo de software, desarrollo de experimentos, de *ports* de sistemas operativos y software, con foros que contienen gran cantidad de información en todos los aspectos, con un gran ambiente de colaboración lo cual nos sirve de gran ayuda.

Y como punto extra añadido posee un excelente **soporte técnico** y la **comunidad de desarrollo** detrás de estos sistemas, de vital importancia si nos encontramos problemas debido a fallos en el funcionamiento físico o una gran comunidad de desarrollo si tenemos problemas de cara al uso del sistema operativo en el que trabajemos o el software empleado.

En el desempeño de esta memoria buscaremos como objetivo el analizar el rendimiento de las distintas distribuciones GNU/Linux que están a disposición de esta placa, para lo cual analizaremos previamente las características físicas del dispositivo a explotar, posteriormente investigar y realizar distintas pruebas de rendimiento en estas distribuciones con el fin de observar su eficiencia y posteriormente poder esclarecer la conveniencia de uso que podemos obtener de cada uno de ellos.

Para ello se empleará una placa Raspberry Pi 3 Model B (versión UK), en la que sucesivamente se irá instalando las distintas distribuciones y se procederá a realizar las distintas pruebas de rendimiento para analizar, observar y sacar las conclusiones precisas del proceso.

¹ SBC. Una placa computadora u ordenador de placa reducida (Single Board Computer) es una computadora completa en un sólo circuito.

Continuaremos eligiendo varias de las distribuciones existentes más populares, con arquitectura de 32 y 64 bits (aprovechando que el nuevo SoC² BCM2837 usado para esta versión es de 64 bits), versiones servidor y usuario, con y sin entorno de escritorio, mono-núcleo y multi-núcleo, etc.

También se realizarán pruebas *overclocking*³ para ver qué aumento de rendimiento se obtiene así como ver qué medidas preventivas deberemos tomar así como qué consecuencias puede conllevar su empleo.

1.2. Investigación, Implementación y Evaluación de la propuesta

Para el desarrollo del TFG emplearemos el proceso de investigación llamado ciclo de investigación-acción, en el cual tras una planificación minuciosa, se pasará a la acción para la obtención de datos evaluables que nos permitirán controlar, registrar y observar el proceso, y posteriormente evaluar la acción para la revisión de la práctica llevada lo cual nos llevará a identificar aspectos a mejorar, imaginar e implementar soluciones así como si es preciso tras la reflexión final, modificar o mejorar los resultados obtenidos.

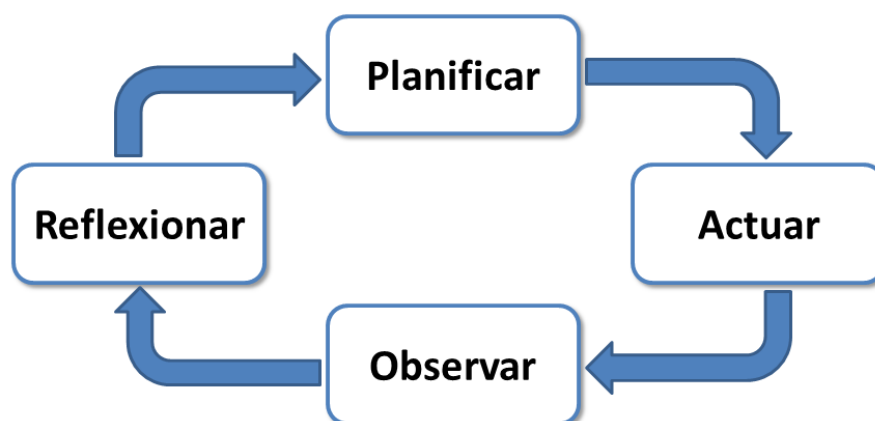


Figura 1 . Ciclo de Investigación-Acción

Buscaremos conseguir una serie de objetivos a alcanzar durante y al final del desarrollo del TFG:

- Obtener una visión general sobre los sistemas operativos que mejor funcionan en este modelo de Raspberry Pi. Nos permitirá orientarnos en la elección de la distribución a elegir en función del rendimiento obtenido para el uso que le querremos dar en un futuro.
- Ampliar competencias y experiencias informáticas en este tipo de sistemas, sobre todo en el análisis global centrado en el rendimiento. El hecho de utilizar estos dispositivos se conseguirá experiencia en su manejo, carga de sistema operativo y configuración, etc.

² SoC. Sistema en chip (del inglés system on a chip o system on chip), describe la tendencia cada vez más frecuente de usar tecnologías de fabricación que integran todos o gran parte de los módulos que componen un computador en un único circuito integrado o chip.

³ Overclocking. La práctica con la que se pretende alcanzar una mayor velocidad de reloj para un componente electrónico (por encima de las especificaciones del fabricante).

1.2.1. Planificación e Investigación

Se buscará información bibliográfica necesaria sobre el todo el tema a tratar, la cual será la base sobre la que se apoyará la investigación, nos acercará a una visión tangible sobre la temática de esta memoria y sobre la que se pondrá posteriormente en práctica y se procederá a la observación y evaluación y finalmente sus consecuentes conclusiones obtenidas de todo el proceso.

1.2.2. Puesta en Acción: Práctica, Desarrollo y Observación

Esta etapa del ciclo será la que probablemente precise de un buen tiempo del desarrollo general del proyecto, pues engloba gran parte del trabajo práctico, como puede ser la descarga de distribuciones, instalación de las imágenes en las tarjetas MicroSD, instalación, configuración y ejecución de diversas aplicaciones de Benchmark⁴ con el que pondremos a prueba su capacidad de cómputo, etc. y todo ello en tiempo real.

Con esto obtendremos una buena cantidad de datos que nos permitirán tener una aproximación real del funcionamiento de los sistemas operativos en nuestra máquina con los que pasaremos a la siguiente fase.

1.2.3. Análisis y Reflexión sobre actividad práctica

El análisis y procesamiento de los datos obtenidos de todas las pruebas realizadas nos permitirán obtener conclusiones claras, las cuales se expondrán de forma esquemática y gráfica para una mejor comprensión. En este se realizarán procedimientos de comparación y evaluación ordinaria y se emplearán aplicaciones para poder presentar los datos de forma clara y concisa.

1.2.4. Evaluación y conclusiones obtenidas de la práctica

Para terminar, una vez realizado el análisis, observaremos como ha resultado la experiencia y tendremos una base que nos proporcionarán la base suficiente para realizar conclusiones finales del trabajo realizado así como posibles propuestas de trabajo futuras.

1.3. Estructura del Trabajo Final de Grado

Siguiendo el esquema de trabajo expuesto en el punto anterior, el TFG se estructurará y adecuará de forma ordenada conforme al ciclo de investigación-acción y su estructura se realizará conforme a su desarrollo, dividiéndolo en cinco capítulos principales.

- **1. Introducción.** Siguiendo la metodología de planificación expuesta, se determinarán que tareas son necesarias para alcanzar los objetivos del proyecto y se estimará el esfuerzo y los recursos necesarios para llevarlas a cabo.

⁴ Benchmark. Prueba de rendimiento o comparativa. Es una técnica utilizada para medir el rendimiento de un sistema o uno de sus componentes.

- **2. Planificación inicial: Conceptos, prerequisites e investigación.** Engloba la conceptualización, contextualización y estudio previo a realizar la comparativa de las distribuciones, se realizará una revisión exhaustiva y explicación de los conceptos básicos que se tratarán en el trabajo. Además se explicará, como y porque se han escogidos los criterios que aparecen en la comparativa, y cómo y por qué se han escogido las diferentes distribuciones que aparecen en la comparativa.
- **3. Puesta en acción.** En esta parte del TFG se realizarán toda la batería de pruebas necesarias sobre la máquina para posteriormente obtener los resultados necesarios para la comparativa de las distribuciones elegidas.
- **4. Análisis y Evaluación de los resultados y Reflexión sobre la práctica.** En este capítulo se procesarán, analizarán y evaluarán los datos para poder sacar conclusiones en la siguiente parte.
- **5. Conclusiones finales y futuras mejoras.** En el capítulo conclusiones, se hace una síntesis de las conclusiones sacadas de la elaboración y resultados del trabajo. Además se incluirán posibles futuras mejoras.
- **6. Biografía y Webgrafía de referencia.** Tendremos el listado de referencias bibliográficas de recursos electrónicos, es decir, de enlaces a webs, blogs o portales de internet, etc.
- **7. Anexos.** Aquí se incluirá de ser necesario, aquella información complementaria que aunque puede haber sido necesaria, no es esencial su exposición en el trabajo. Aquí pueden entrar guías y manuales del funcionamiento de las pruebas o de los *benchmarks*, por ejemplo.

Cabe decir que pueden producirse cambios y revisiones entre cualquiera de las etapas para poder realizar mejoras en el proceso de investigación, es decir, por ejemplo, si durante la etapa de análisis de los resultados, esos datos no fueran satisfactorios u se observase algún tipo de incoherencia, regresaríamos a la etapa de puesta en acción para repetir alguna prueba en particular.

1.4. Planificación y temporalización del TFG

Siguiendo la metodología de planificación aprendida, se determinarán las tareas necesarias para la consecución de los objetivos del proyecto y se estimará el esfuerzo y los recursos necesarios para llevarlas a cabo.

1.4.1. Listado de tareas a realizar

La siguiente tabla se esquematiza la lista de actividades y tareas a realizar y sus respectivas fechas de inicio y entrega durante el transcurso del TFG.

Trabajo Final de Grado – Fechas de inicio/entrega	
Fecha inicio	21/02/18
Elaboración del Borrador del TFG	21/02/18 – 05/03/18
Plan de trabajo TFG (PEC 1)	26/02/18 – 19/03/18
Diseño e implementación (PEC 2)	20/03/18 – 09/04/18
Resultado y Análisis (PEC 3)	10/04/18 – 07/05/18
Entrega final de la memoria (PEC 4 – Entrega Final)	08/05/18 – 28/05/18
Video Presentación, debate virtual y finalización TFG	16/06/18 – 22/06/18

Tabla 1. Tabla de fechas de entrega del TFG

- **Plan de trabajo TFC (PEC 1)**

La PEC1 (sobre la cual comenzamos la planificación) será la documentación en la que se describe la descripción de la tarea a realizar como TFG, en el que se incluye una planificación temporal que reflejando el progreso del trabajo para alcanzar los objetivos marcados. Esta información se puede ver en el diagrama de Gantt expuesto en el punto 1.4.3.

- **Diseño e implementación (PEC 2)**

Esta PEC se nos indica que ha de incluir un pequeño informe que comente el grado de progreso del trabajo de acuerdo a la planificación realizada en la PEC1. Se comentarán las desviaciones que sean podido producir más importantes y qué medidas paliativas se han tomado para ajustarlas. En esta entrega se debería haber realizado al menos un 30% del trabajo. Este informe ya incluirá un primer esbozo del TFG, con un índice completo y detallado sobre la estructura expuesta en el punto anterior.

- **Resultado y análisis (PEC 3)**

Llegados a esta fase, se deben ya haber realizado las actuaciones y pruebas necesarias y se deberían estar redactando a partir de los datos analizados de la comparativa así como las conclusiones obtenidas del proceso. En cuanto al progreso del trabajo realizado, ya se debería estar en torno al 70% del total del TFG, y con una gran parte de los capítulos ejecutados.

- **Entrega final de la Memoria y Presentación TFG (PEC4 - ENTREGA FINAL)**

Con la realización de esta tarea se finaliza la realización del TFG. Se habrán completado y ajustado la memoria conforme a la planificación y se procederá a su entrega dentro del plazo recogido.

Ya como paso final, nos faltará la video-presentación y debate virtual, colofón final del trabajo realizado.

1.4.2. Planificación temporal del proyecto

La fecha inicial del proyecto es el 21-02-2018, mientras que la fecha límite es el 28-05-2018. Disponemos de 122 días naturales, es decir contando festivos, sin embargo debido a mi disponibilidad horaria y falta de tiempo, sobre todo debido a mi jornada de empleo, vida familiar,

etc. no me resulta posible fijar una jornada de dedicación al TFG de duración fija, ni designar a priori los días laborables de la semana.

Para la estimación de esfuerzos se ha considerado en general una dedicación prevista de entre 6 a 8 horas por cada semana natural del plan (7 días). No obstante, se han realizado ajustes en las duraciones de algunas tareas para tener en cuenta que su periodo de realización incluye fines de semana o festivos, ya que en esos días se pueden concentrar un mayor número de horas de trabajo.

Para la planificación se ha empleado Microsoft Project 2010 para realizar la planificación. Antes, sin embargo, es necesario establecer las precedencias entre las actividades que hay que desarrollar dentro del proyecto, para lo cual utilizaremos una tabla de precedencias.

Id	Nombre de tarea	Duración restante
0	Planificación temporal del TFG	122 días
1	TFC BD	122 días
2	00 - Borrador sobre la temática del TFG	5 días
3	01 - Planificación TFG (PEC 1)	22 días
4	01.01 - Elección de la propuesta del TFG	3 días
5	01.02 - Elaboración Plan de Trabajo TFG	3 días
6	01.03 - Búsqueda de fuentes de información bibliográfica	4 días
7	01.04 - Redacción de la Planificación del TFG (PEC 1)	9 días
8	01.05 - Realización del Diagrama de GANT	2 días
9	01.06 - Entrega Plan de Trabajo TFG (PAC 1)	1 día
10	02 - Diseño e Implementación (PEC 2)	21 días
11	02.01 - Búsqueda de fuentes de información bibliográfica	4 días
12	02.02 - Estudio previo para la selección de distribuciones	4 días
13	02.03 - Búsqueda de aplicaciones de Benchmark	4 días
14	02.04 - Redacción y elaboración de la memoria	6 días
15	02.05 - Revisión Final Documentación	1 día
16	02.06 - Entrega PEC2	1 día
17	03 - Resultados y Análisis (PEC 3)	28 días
18	03.01 - Instalación de las imágenes	6 días
19	03.02 - Pruebas de rendimiento	9 días
20	03.03 - Redacción de los datos obtenidos en las pruebas	5 días
21	03.04 - Análisis y procesado de los datos obtenidos en las pruebas	3 días
22	03.05 - Redacción del proceso de análisis en la memoria	3 días
23	03.06 - Revisión Final Documentación	1 día
24	03.07 - Entrega PEC3	1 día
25	04 - Elaboración Memoria y Presentación TFG (PEC4) - ENTREGA FINAL	21 días
26	04.01 - Elaboración Final de la Memoria (Conclusiones y revisión de toda la documentación)	20 días
27	04.02 - Entrega Memoria y Presentación TFC (ENTREGA FINAL)	1 día
28	05 - Video presentación	7 días
29	05.01 - Entrega de la video presentación	6 días
30	05.02 - Entrega de la video presentación	1 día

Tabla 2. Tabla de planificación temporal del TFG

1.4.3. Diagrama de GANTT

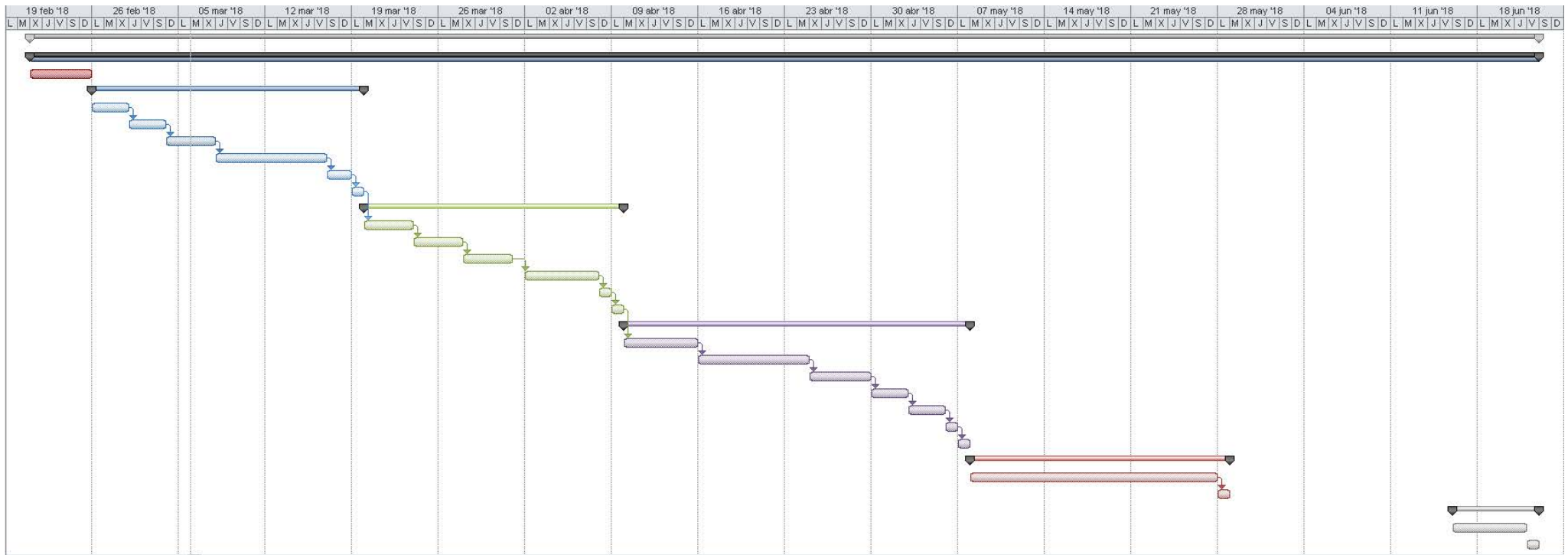


Figura 2. Diagrama de Gantt con la temporalización del TFG

2. Planificación inicial: Conceptos, prerequisites e investigación.

2.1. Aproximación conceptual y contextualización de la propuesta

En los tiempos que corren actualmente, crisis, recortes económicos, hay que mirar más allá y pensar en la adquisición de recursos tecnológicos al mejor precio posible, y por supuesto si éstos son eficientes mejora mucho la situación. Todo esto nos ha permitido obtener una mejor visión global sobre la temática que vamos a tratar y poder enfocar y orientar su empleo desde un punto de vista práctico, como un recurso aplicable a multitud de ámbitos.

Estos dispositivos de pequeño tamaño gracias a su gran evolución, son cada vez más accesibles y sencillos de comprender, simplifican las barreras visuales que asustaban en el pasado a aquellos, retirándose y pensando en no optar a adentrarse en el mundo de la informática debido a la aparente complejidad que va alcanzando su entorno y que cada vez se diversifica más.

Si pudiéramos girar la cabeza y echar una mirada hacia atrás en el pasado, por un precio irrisorio conseguimos una estación de trabajo en Linux con aceleración gráfica 2D/3D, algo que hace años sería impensable de encontrar por un precio tan bajo en un ordenador personal. Tanto la informática como los ordenadores personales son elementos imprescindibles y esenciales en el mundo de la comunicación en el que vivimos. Un mundo de intercomunicación que permite que personas desde distintas partes del mundo puedan colaborar en un proyecto común, algo que en el ámbito educativo es más que básico y sustancial.

La informática está cambiando, y las personas cada vez emplean menos los ordenadores personales. Éstos se están sustituyendo por otros dispositivos mucho más portables como son las tabletas y dispositivos móviles, lo cual está haciendo que el ordenador personal como estación de trabajo en casa vaya desapareciendo, en pro de sistemas mucho más ligeros y de gran rapidez que nos permiten hacer lo mismo.

Esta placa nació en 2012 con un precio de 25\$, de la cual ya se cuenta con su tercera versión, la Raspberry Pi 3 Model B, con la que vamos a trabajar, con unas prestaciones más que aceptables y manteniendo el precio aproximado de sus predecesoras.

Un éxito más que merecido e indiscutible, ya que ha vendido más de 10 millones de unidades en el mundo (a fecha de Septiembre 2016), y del que parten miles de proyectos caseros de bajo presupuesto.

La **Raspberry Pi** (Fig.3), es el gran referente hoy en día de los sistemas SBCs y podemos decir que es la placa que ha realizado un salto exponencial en cuanto al uso de estos sistemas por parte del consumidor en tecnología “de a pie”.



Figura 2. Placa Raspberry Pi Model 3

Describiremos a continuación cuáles son sus principales ventajas para, en primera instancia se pueda observar el gran atractivo de este tipo de dispositivos:

- Un **menor consumo eléctrico**, este tipo de sistemas suelen y pueden estar siempre encendidos (24x7) con una capacidad de disipación calorífica mínima, por lo que un menor consumo supone un ahorro económico importante.
- **Factor de tamaño reducido**, se trata de mini ordenadores personales, conforme está avanzando la tecnología, aumentan cada día su potencia y prestaciones así como sus sistemas de interconexión.
- **Precio reducido**, hace que supongan un costo en ocasiones de lo más irrisorio, debido sobre todo a la fabricación en serie de electrónica de consumo, a la globalización económica y los sistemas de producción actuales.
- Emplean **Sistemas operativos libres**, aunque existen otros sistemas operativos propietarios, la mayoría utilizan versiones libres de Linux adaptadas a los procesadores de estos sistemas (arquitectura ARM⁵ Cortex).

Si tenemos que citar alguna desventaja, la principal es que comparándolos con las arquitecturas típicas cliente/servidor claramente son **mucho menos potentes**, pero esto es algo que está empezando a cambiar.

El actual SBC cuenta con unas características y prestaciones mucho más que aceptables (por ejemplo: 1GB RAM, procesadores Quad Core a 1,2 GHz, WIFI, USB 3.0, etc.), en los cuales se pueden instalar versiones de Linux mucho más complejas (Debian, Suse, etc.) y que se pueden realizar tareas como las que realizan ordenadores personales y los mismísimos servidores. También decir, que los desarrolladores también compilan sus aplicaciones para este tipo de procesadores, lo cual hace que tengamos cada vez más software a nuestro alcance.

⁵ ARM Holdings plc. Multinacional dedicada a los semiconductores y al desarrollo de software.

En la siguiente tabla podemos observar más en detalle sus características principales.

RASPBERRY PI 3 Model B	
Procesador	Broadcom BCM2387 chipset a 1.2GHz Quad-Core ARM Cortex-A53 (64Bit)
GPU	Dual Core VideoCore IV® Multimedia Co-Processor. Compatible con Open GL ES 2.0, Aceleración por hardware OpenVG, y decodificación de video 1080p30 H.264.
Memoria RAM	1 GB LP-DDR2
HDMI-VGA	HDMI rev 1.3 y 1.4, RCA compuesto
Conectividad	BCM43143 WiFi, Bluetooth Low Energy (BLE). Ethernet 10/100 Ethernet socket, Salida de video: HDMI, Compuesto RCA, Salida de audio 3.5mm Jack, USB 4 x USB 2.0, USB Host x 3, Device x 1, Ethernet RJ-45, Headphone Jack
Puertos IO	Conexiones GPIO 40-pin 2.54 mm (100 mil), conector a cámara de 15-MIPI Camera Serial Interface (CSI-2), conector de Display Serial Interfac (DSI)
Almacenamiento	Adaptador de Tarjeta Micro SDIO.
Precio	40€
Sistemas Operativos	Android 5.1, Windows IoT, Ubuntu 17.04, Raspbian, LibreElec, Suse , Slackware, motionEyeOs, etc.

Tabla 3. Tabla de características de Raspberry Pi 3 Model B

La Raspberry Pi se ha ideó con la intención y determinación de incentivar la experimentación, de asombrar por lo coqueto de su reducido tamaño y por supuesto minimizar el coste de rotura accidental y sustitución posterior durante todo el proceso de acceso al sistema y creación de recursos necesarios.

La comunidad de desarrollo que hay detrás de estos dispositivos es tan sumamente inmensa y crece a pasos agigantados, en la que existen miles de proyectos de acceso libre compartidos de forma altruista en todos los campos posibles que podrían despertar el interés de todo tipo de personas y edades. Un ejemplo del este tipo de proyectos de software es Scratch⁶, orientado a alumnado de primaria y secundaria, que se emplea en los colegios para acercar la programación e incentivar su uso a estudiantes a hacer que los niños y adolescentes se interesen en la tecnología.

Y es en esta gran comunidad donde encontraremos recursos en los que apoyarnos para poder desarrollar el trabajo práctico de la presente memoria, y que ésta contenga la información y prácticas necesarias para poder emplearlo de la forma más eficiente posible, sumado a la gran comunidad que hay también por detrás del propio Linux, porque en Raspberry Pi tenemos gran cantidad de distribuciones de todo tipo, la principal es **Raspbian** y actualmente en su versión *Jessie* con y sin el escritorio *PIXEL*. Aunque cabe decir que la cantidad de distribuciones a día

⁶ Scratch. lenguaje de programación visual desarrollado por el MIT, se utiliza por estudiantes, académicos, profesores y padres para crear fácilmente animaciones, juegos (también educativos) e interacciones etc.

de hoy es apabullante, tanto para uso exclusivo, como escritorio personal como con usos específicos.

2.2. Investigación, Implementación y Evaluación de las distribuciones GNU/Linux para Raspberry Pi 3 Model B

Como hemos comentado en el punto anterior, nos centraremos en el estudio de aquellas distribuciones que son para uso genérico y de usuario de escritorio.

Hay que resaltar que tras la mejora que se ha producido de hardware en esta última revisión, la experiencia de escritorio ha mejorado notablemente, lo cual ha propiciado que se produzcan nuevos lanzamientos de distribuciones con mayor contenido y orientado a emplearlos como equipo de escritorio uso común.

En esta etapa, comienza la fase de investigación en sí, estudiaremos las necesidades básicas que serán de utilidad de cara a resolver el estudio que llevaremos a cabo. Ordenaremos, agruparemos, dispondremos y relacionaremos los datos acorde a los objetivos fijados.

Comenzaremos con lo necesario para poder empezar a trabajar.

2.2.1. Materiales y Recursos para el desempeño de las pruebas

Se requiere en principio los siguientes elementos para realizar todas las pruebas necesarias:

- Una SBC Raspberry Pi 3 Model B.
- Una base o carcasa básica fabricada en ABS transparente con ventilación.
- Disipadores de cobre para el procesador y la memoria RAM.
- Ventilador de pequeño formato de 5v y 2A.
- Transformador de 5V y al menos 2.4A.
- Un monitor con entrada HDMI y en su defecto un adaptador de HDMI a VGA/DVI.
- Teclado y Ratón USB.
- Soporte de almacenamiento propio tanto una tarjeta MicroSD de 64 GB Toshiba EXCERIA Clase 10 UHS-I Clase 3 (Velocidad de lectura real de unos 60 MB/s y de escritura de 2 MB/s), totalmente compatible según la web Embedded Linux Wiki.
- Un ordenador personal con conexión a Internet para la descarga e instalación de las imágenes con Linux Ubuntu (comando DD).
- Medidor de consumo eléctrico.

2.2.2. Criterios y características previas de selección de las distribuciones

Comenzaremos con el análisis de los sistemas operativos observando sus características intrínsecas que precisaremos para poder comparar y analizar de las distintas distribuciones:

- Datos del proyecto como la fecha de lanzamiento, el tipo de distribución, tipo de licenciamiento, políticas de lanzamiento y actualizaciones.
- Arquitectura de sistema operativo, Versión del núcleo Linux, si es de 32 o 64 bits, APIs de video como OpenGL ES y OpenVG, bibliotecas principales, etc.

- Sistema de ficheros compatibles (EXT4, XFS, BTRFS).
- Sistema de gestión de paquetes, como APT de Debian (dpkg), Pacman de Arch Linux y Zypper de OpenSuse.
- Entorno de usuario y tipo de escritorio, que puede tener o no interfaz gráfica, o con escritorios Pixel Desktop, LXDE, KDE, etc.
- Tamaño del fichero de imagen descargada.
- Aplicaciones preinstaladas.

Posteriormente en la fase de pruebas en la que se estudiará el comportamiento, también tendremos en cuenta y podremos cuantificar las siguientes variables, de las cuales algunas se modificarán para poder mejorar el rendimiento y que aportarán información en tiempo real del comportamiento del sistema operativo en nuestra SBC. Comenzaremos a con las variables que precisaremos para poder comparar y analizar las distintas distribuciones:

- Tiempo de arranque del sistema operativo, aunque esto depende de la velocidad de lectura de la *microSD* usada, también dependerá de la carga de inicio del sistema operativo, servicios arrancados, aplicaciones instaladas, etc.
- Espacio de disco usado inicialmente.
- RAM usada y disponible.
- Tamaño de la Swap.
- Temperatura de la CPU.
- Espacio de disco usado.
- Consumo eléctrico (emplearemos un medidor de consumo eléctrico).

De cara a cuantificar el rendimiento, ya hemos avanzado que emplearemos algunos *tweaks*⁷ para poder mejorarlo, tomando valores tanto antes y después de los cambios. Cabe decir que se tomarán medidas de seguridad para evitar sustos, proveyendo a nuestra Raspberry con unos disipadores de cobre para la CPU y RAM y un pequeño ventilador de 5V que refrescará el pequeño habitáculo y contribuirá a que baje la temperatura. Algunos de los valores que tocaremos serán:

- Frecuencia Procesador, que por defecto viene a 1.200 MHz, realizaremos pruebas a 1.350 MHz, lo cual nos deberá ofrecer un 10% más de rendimiento.
- Frecuencia de la memoria RAM, la aumentaremos a 500 MHz.
- RAM de video usada. El valor por defecto es de 64 MB, pero se puede aumentar hasta 944 (por ejemplo cambiando la variable *gpu_mem=128*), con lo cual conseguiremos un rendimiento gráfico superior.
- Frecuencia de acceso a la microSD (Overclock bcm2835-sdhost SD/MMC driver (de 50 a 100 MHz), en una tarjeta UHS-1 certificada.

⁷ Tweaks. Modificaciones de los valores por defecto de configuración hardware, en nuestro caso para obtener mejoras de rendimiento,.

La siguiente tabla ilustra los valores de frecuencia de los principales componentes que tiene por defecto de la Pi 3 y sus respectivas modificaciones.

	Por defecto	Overclock
ARM	1200 MHz	1350 MHz
Core	400 MHz	500 MHz
GPU	300 MHz	500 MHz
RAM	450 MHz	500 MHz

Tabla 4. Tabla de configuración de frecuencias de Raspberry Pi 3 Model B

También decir que se analizará el rendimiento general, el tiempo de respuesta del propio sistema operativo a la hora abrir aplicaciones de uso básico como el navegador, aplicaciones de oficina, etc. así como su respuesta multitarea.

2.2.3. Aplicaciones de control y medida de las pruebas de rendimiento

A continuación expondremos que aplicaciones se emplearán para realizar las pruebas de rendimiento. En algunas ocasiones realizaremos scripts sencillos para intentar ahorrar tiempo y automatizar procesos. No obstante, puede que alguna de las aplicaciones que emplearemos para realizar las pruebas de rendimiento no sean compatibles con alguna de las distribuciones.

- **Sysbench**, es una herramienta que realiza secuencias de comandos multi-hilo basada en LuaJIT⁸. Se emplea normalmente para realizar benchmarks para base de datos, pero en nuestro caso particular lo emplearemos para realizar cargas de trabajo arbitrarias para la CPU y la memoria RAM.
- **GLMark2** es otra herramienta muy conocida Open Source para realizar pruebas de rendimiento GPU con OpenGL 2.0 y OpenGLES 2.0, y con el que se medirá el rendimiento gráfico de la VideoCore 4 de nuestra Pi.
- **Hdparm** es una buena utilidad para poder ver lecturas de disco y de la caché del disco (buffer caché), de esta forma podremos medir la velocidad directamente del contenido de la memoria caché del buffer del disco proveniente del propio disco, sin tener que acceder a éste último. Y también podremos medir la velocidad de lectura de información de la propia superficie del disco (buffered disk) y medida en la caché del disco, que como podemos imaginar el valor obtenido será inferior. Decir cabe, que para la obtención de resultados que sean significativos será necesario que se realicen al menos 3 veces las pruebas.
- **DD (Dataset Definition)**. Conocida aplicación para Linux de gran utilidad para realizar copias de seguridad de discos y particiones. Nosotros la emplearemos para poder monitorizar el rendimiento de los procesos de lectura/escritura en la Pi.
- **RPI-Monitor**. Con esta aplicación podremos acceder vía web al estado de nuestro dispositivo, cosa que podremos hacer tanto desde este mismo como desde cualquier otro equipo en la red.

⁸ LuaJIT. Compilador Just-In-Time (JIT) para el lenguaje de programación Lua.

Una vez abierta la interfaz web nos muestra un resumen del estado de nuestra Raspberry Pi 3, en la que podemos encontrar por orden los siguientes parámetros a monitorizar:

- **Versión:** Nos da datos sobre el procesador, la distribución y versión instalada. Igualmente nos indica la versión del núcleo de Linux, y la del firmware. Por último, en este apartado, nos indica si hay paquetes por instalar.
- **Tiempo encendido (Uptime):** En este apartado nos indica la hora del reloj de nuestra Raspberry Pi, y cuánto tiempo lleva en funcionamiento desde el último arranque
- **CPU:** Nos indica la carga de la CPU, y otros parámetros como frecuencia, tensión, etc.
- **Temperatura:** Nos indica la temperatura medida en la CPU.
- **Memoria:** Aquí podemos ver la cantidad de memoria en uso y disponible.
- **Swap:** En este punto encontraremos el uso y estado de la memoria de intercambio.
- **SD Card:** Lo mismo que en el apartado anterior pero para el caso de nuestra tarjeta SD.
- **Red (Network):** Aquí se verán los paquetes enviados y recibidos según tu conexión.
- **Servidores (Servers).** Con información de los servicios habilitados como puede ser Apache, MySQL, PostgreSQL, etc.
- **Wifi.** Con información de los datos enviados/recibidos a través de la conexión inalámbrica.

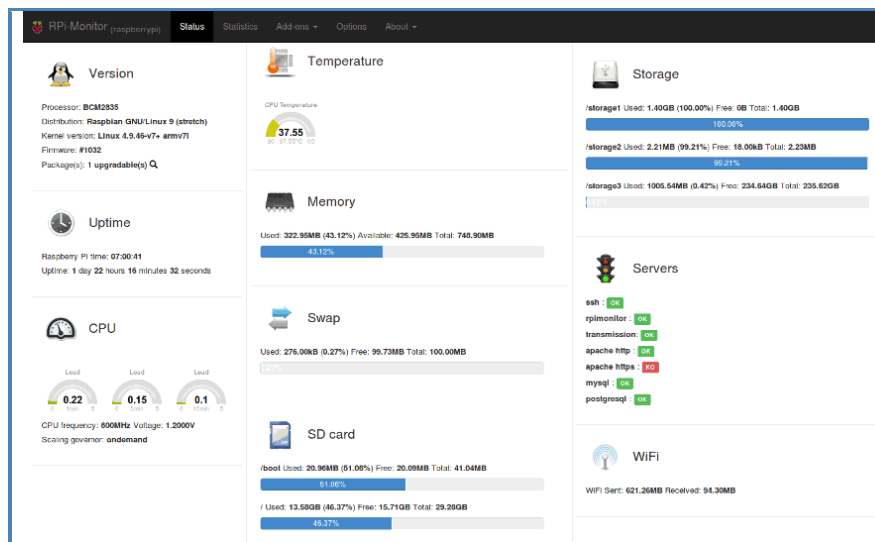


Figura 4. Ejemplo de captura de la vista principal de RPI-Monitor

Otra utilidad muy interesante de esta aplicación es la posibilidad que tiene para el muestreo de estadísticas de forma sencilla. De este modo nos ofrecerá información gráfica mucho más precisa y detallada de las variables a cuantificar a lo largo del tiempo de duración de las pruebas. Los parámetros que nos ofrece son los siguientes:

- Carga de la CPU.
- Tiempo de uso de la máquina.
- Temperatura de la CPU.

- Uso de la Memoria RAM.
- Memoria de intercambio (SWAP).
- Uso de la tarjeta MicroSD.
- Tráfico de la red: Paquetes enviados y recibidos.



Figura 5. Ejemplo de captura de la vista de estadísticas de RPI-Monitor

2.2.4. Estudio y Selección de Distribuciones de Sistemas Operativos GNU/Linux

Para comenzar el estudio y selección de las distribuciones de SSOO GNU/Linux partiremos de una visión general de los sistemas operativos recomendados para su descarga en la web oficial de Raspberry Pi, donde por supuesto está **Raspbian Stretch** con escritorio, que será, como es lógico, la primera con la que comenzaremos a realizar pruebas. Otras que aparecen son *Ubuntu Mate*, *Snappy Ubuntu Core*, *Windows 10 IoT Core*, etc. De todas éstas como nuestro análisis principalmente se basará en el uso como Sistema Operativo de Escritorio, he optado por seleccionar también la muy amigable distribución **Ubuntu Mate**, aunque se trata de otra versión *Debian*, mi experiencia con este sistema operativo en un PC convencional a lo largo de los años siempre ha sido muy satisfactoria, además parece que Canonical ha hecho un gran esfuerzo por portar el sistema a nuestra pequeña placa, lo cual merece una buena oportunidad.

La siguiente distribución que procederemos a analizar, será la *OpenSuse Tumbleweed*, también con un gran trayecto de su desarrollo buscando satisfacer a la mayoría de sus usuarios de menos experiencia, con interfaces gráficas muy cuidadas y con sistemas de auto-configuración que simplifican el mantenimiento del propio sistema. Cabe decir que posee versiones de 32 y 64 bits. Con lo que realizaremos pruebas con ambas, con el fin de constatar el aprovechamiento de la tecnología de 64 bits para la que fue diseñado el procesador.

Otra distribución que también cuenta con una filosofía muy interesante es **Arch Linux**, denominada “*The Arch Way*”, emplea el acrónimo *KISS* (“*Keep It Simple, Stupid*”), es decir la Simplicidad es su principal objetivo. Esta simpleza no está orientada al uso en sí del propio sistema como lo hace *OpenSuse*, se basa en mantener una estructura base UNIX sin modificaciones innecesarias, mejorando las demandas de recursos del sistema y dándole

ligereza en su funcionamiento, lo cual es algo primordial en nuestra SBC. También cabe decir que posee una versión de 64 bits lo cual nos vendrá fenomenal para nuestras pruebas de rendimiento.

Para terminar con la selección, nuestra última opción es para **Fedora 27**, la cual es compatible desde la versión 25, dejando a un lado su versión anterior de *Pidora*⁹, una versión modificada para versiones menos potentes de Raspberry Pi. También posee compatibilidad para sistemas de 32 y 64 bits, y cuenta con varias versiones de escritorio instalado, encabezadas por su versión Workstation con *GNOME 3.26* como escritorio de referencia. Bien conocida es la estabilidad de este sistema, y con un ritmo frenético de actualizaciones (normalmente cada 9 meses), tanto de parches de seguridad, librerías, etc. Normalmente en PC goza de una fluidez, que comprobaremos que tal se comporta en nuestra placa, y por la que merece una oportunidad dentro de nuestra comparativa.

Cabe decir además que el hecho de que haya desestimado otras versiones, que en ordenador personal gozan de mayor acogida, es que muchas de ellas no están bien soportados todos los controladores y además poseen pocas actualizaciones en sus versiones, de parches de seguridad, etc. como por ejemplo ocurre con *FreeBSD ARM*. Otras versiones como *Slackware ARM Project*, en las que se nota bastante abandono en su soporte y desarrollo a la comunidad.

Y con las versiones expuestas podemos comenzar, ya que en un principio no seleccionaré ninguna más, puesto que partimos de cinco distribuciones pero tres de ellas tienen versiones para arquitecturas de 32 y 64 bits, con lo que pasan a ser ocho distribuciones en total.

A continuación, en el siguiente apartado expondremos las distribuciones seleccionadas con sus características básicas mucho más en detalle, para poder obtener información más detallada tras haber realizado una primera aproximación.

2.2.4.1. Sistema Operativo Raspbian Stretch with Desktop. Como es bien sabido, esta versión está basada en Debian, específicamente en **Debian 9**, con las últimas versiones de las aplicaciones y paquetes. Versión oficial para esta placa, en la que ya se añadido soporte para la nueva versión Plus recién salida al mercado. Incluye el escritorio de acceso ligero llamado *PIXEL (Pi Improved Xwindows Environment Lightweight)*. También decir que incluye el administrador de ficheros *PCManFM*, la aplicación *PiServer* que permite a los usuarios configurar un sistema Raspbian y compartirlo entre varios clientes de la misma red, herramientas de desarrollo como lo es *IDLE* para Python, el más que conocido *Scratch*, indicado para enseñar programación de videojuegos, *LibreOffice*, *Chromium*, *Pi Store*, etc.

A continuación podemos ver más en detalle sus características principales en la siguiente tabla.

⁹ *Pidora*. Versión de Fedora Remix para Raspberry Pi. La última versión que salió de esta distribución es de finales de Abril de 2014.

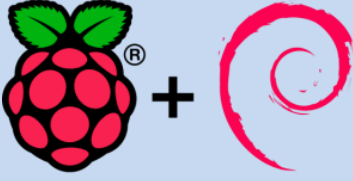
 Raspbian	
Año de creación del proyecto	2012
Distribución nativa	Debian 9
Versión del núcleo	Linux Kernel 4.9.80+.
Arquitectura del SO	ARM 32 bits.
Última modificación	13 de Marzo de 2018.
Nombre del fichero	2018-03-13-raspbian-stretch.zip
Tamaño del fichero	1,7 GB
Gestión de ficheros	PCManFM.
Gestor de ventanas	Openbox 3.6.1-4+rpi3.
Colección de compiladores GNU (gcc)	6.3.
Entorno de usuario	PIXEL (LXDE) versión 0.99.2-3 (core 9+rpi1)
Administrador de paquetes	Dpkg.
Método de actualización	APT (Software Updater, Ubuntu Software center)
Sistemas de ficheros soportados	EXT2/ EXT3/ EXT4(Nativo).
Enlace de descarga	https://downloads.raspberrypi.org/raspbian_lite_latest

Tabla 5. Tabla de configuración de frecuencias de Raspberry Pi 3 Model B

2.2.4.2. Sistema Operativo Ubuntu Mate 16.04 LTS Final Release Xenial. Bien conocida y popular entre los usuarios de escritorio Linux es esta distribución de Ubuntu basada en el entorno de escritorio ligero MATE derivado del código de *GNOME 2*, compatible con innumerables versiones de Linux, cosa que lo hace ideal para plataformas de ordenador personal de bajas y modestas prestaciones, y algo que además nos viene fenomenal para trabajar en nuestra SBC, que aunque en esta versión se nota su incremento de potencia sigue estando lejos de las plataformas de escritorio de hoy en día.

Esta distribución introduce como novedad *MATE Tweak* utilizando herramienta de personalización para las opciones de inicio fácilmente, como por ejemplo cambiar el tamaño de los iconos, agregar paneles, etc. También cuenta con *Firefox* como navegador web, *Pluma* como editor de texto, *Thunderbird* como cliente de correo con filtro anti spam y *LibreOffice*. En la siguiente tabla podemos apreciar más en detalle sus características.


 ubuntu MATE	
Año de creación del proyecto	2014
Distribución nativa	Debian (Linux Ubuntu 16.04)
Versión del núcleo	Linux Kernel 4.4.0.38.v7
Arquitectura del SO	ARM 32 bits (ARMv7).
Última modificación	21 de Abril de 2016.
Nombre del fichero	ubuntu-mate-16.04.2-desktop-armhf-raspberry-pi.img
Tamaño del fichero	1,2 GB.
Gestión de ficheros	Caja (versión de Nautilus)
Gestor de ventanas	Mate's Windows Manager 1.12.1
Colección de compiladores GNU (gcc)	5.4.0
Entorno de usuario	MATE Desktop con MATE Menu 5.7.1 (Xfce)
Administrador de paquetes	Dpkg
Método de actualización	APT (Software Updater, Ubuntu Software center)
Sistemas de ficheros soportados	EXT2/ EXT3/ EXT4(Nativo).
Enlace de descarga	https://ubuntu-mate.org/raspberry-pi/ubuntu-mate-16.04.2-desktop-armhf-raspberry-pi.img.xz

Tabla 6. Tabla de configuración de frecuencias de Ubuntu Mate

2.2.4.3. Sistema Operativo OpenSuse 42.3 Tumbleweed. El sistema operativo *OpenSuse*, basado en la distribución histórica y empresarial *Suse Linux*, disfruta de una gran estabilidad y tradicionalmente con una gran comunidad de soporte tras él.

La versión *Tumbleweed* posee un sistema emplea actualizaciones de forma continua, sin saltarse una versión a otra (Rolling Distro¹⁰), de esto último le viene el apodo que posee (tumbleweed es en inglés es un estepicursor¹¹).

Respecto a sus características principales, una de las herramientas más atractivas de esta distribución que posee es el Centro del Control *Yast*, que nos permite de forma muy sencilla la instalación y administración del sistema, tanto el hardware como configurar la red, servicios de sistema y la configuración de seguridad, de una forma muy similar a como lo hace el Panel de Control en Windows.

¹⁰ Rolling distro. Método de distribución en Linux que implementa actualizaciones frecuentes.

¹¹ Estepicursor. También llamada nube del desierto, son plantas rodadoras típicas de zonas desérticas e icono de las películas de género Western.

Decir también que introduce grandes novedades, además de la versión de 64 bits para nuestra SBC, lo cual lo hace más interesante como habíamos comentado anteriormente de cara a nuestro estudio, así como características muy importantes como lo es el soporte nativo para el sistema de ficheros *BTRFS*, proporcionándole gran estabilidad.

Otra interesante novedad que cabe destacar es la inclusión de *JeOS*¹², un soporte que facilita una plataforma idónea para agilizar el entorno en aquellos entornos actuales, como lo son las nubes (cloud) o centros de datos virtualizados. Un ejemplo de estas tecnologías es los software empresariales de virtualización *KVM/Xen*, *Xen* parcialmente virtualizado, *Microsoft Hyper-V*, *VMware* o *OpenStack Cloud*.

Con todo lo introducido, queda claro que seguramente OpenSuse es una distribución de propósito general y con alta implicación empresarial, quizás la más potente en un principio, con este tipo de funciones específicas como características principales del entorno, y que será una gran e interesante apuesta para nuestra particular investigación en la que podremos comprobar que tal se comporta en nuestra pequeña placa.

Como punto flaco de este tipo de distribuciones, sobretodo basado en experiencias personales, es la falta de paquetes y versiones de aplicaciones que suelen aparecer en otras distribuciones de Linux como son las basadas en Debian, pero que en ocasiones se suelen solventar en un plazo razonable o por el contrario nos toca añadir repositorios “no oficiales” y realizar pruebas, con los problemas de estabilidad que puede conllevar este tipo de acciones.

A continuación podemos observar en la siguiente tabla sus propiedades principales.

 openSUSE	
Año de creación del proyecto	1994
Distribución nativa	Suse Linux
Versión del núcleo	Linux Kernel 4.4.21-69-default.
Arquitectura del SO	ARM 32 bits y 64 bits.
Última modificación	2 de Febrero de 2018.
Nombre del fichero	openSUSE-Tumbleweed-ARM-XFCE-raspberrypi3.aarch64-2018.02.02-Build1.2.raw.xz
Tamaño del fichero	696 MB.
Gestión de ficheros	Dolphin file manager
Gestor de ventanas	JeOS y E20.

¹² JeOS. Abreviación del concepto Just Enough Operating System, una versión minimizada de un Sistema Operativo que satisface una necesidad concreta de un aplicativo en particular.

Colección de compiladores GNU (gcc)	7.3+r258313
Entorno de usuario	KDE, GNOME, XFCE y LXDE.
Administrador de paquetes	RPM Package Manager
Método de actualización	Zypper (YaST2) Ready For Action.
Sistemas de ficheros soportados	BTRFS/ EXT2/EXT3/EXT4.
Enlace de descarga	http://download.opensuse.org/ports/aarch64/tumbleweed/images/openSUSE-Tumbleweed-ARM-XFCE-raspberrypi3.aarch64-Current.raw.xz

Tabla 7. Tabla de configuración de frecuencias de OpenSuse

2.2.4.4. Sistema Operativo ArchLinux ARM. Como ya habíamos adelantado, uno de los puntos más atractivos de esta distribución es la de la capacidad intrínseca de construir y modificar tu propio sistema operativo a tu gusto, cosa que es plausible en la mayoría de distribuciones Linux, pero que particularmente en este SO es su razón de ser, con una gran comunidad específica trabajando detrás y una ligereza que permite centrarnos en otros aspectos más específicos, punto a favor, sobretodo sabiendo que vamos a trabajar sobre una placa de limitada potencia pero de gran versatilidad.

Otro aspecto que me gusta también de esta versión es *Arch Linux* no parchea nada, cuando instalas y empleas un paquete no ha sido modificado por el equipo de desarrollo, nos de la manera tal cual fue liberada, cosa que no genera gastos de recursos extras. Emplea el sistema propio *ABS (Arch Build System)* que compilar paquetes *tar* y posteriormente se instalan con *pacman*.

Respecto al repositorio oficial detrás *AUR (Arch User's Repository)*, tampoco existen objeciones, como puede pasar con otras distribuciones, la comunidad trabaja para tener paquetes preparados para instalarse y cuenta con una sistema de búsqueda centralizada.

Quizás como punto no favorable, especialmente en usuarios no muy avanzados, posee cierto grado de dificultad en su instalación, lo cual nos permite ampliar competencias y aprender cómo funciona un sistema operativo, incluyendo la satisfacción que produce el esfuerzo y tiempo empleado para conseguirlo. Seguidamente, en la siguiente tabla podemos analizar cuáles son sus principales características.

	
Año de creación del proyecto	2009
Distribución nativa	Ninguna (Inspirado en CRUX ¹³)
Versión del núcleo	Linux Kernel 4.4.x.

¹³ CRUX. Distribución de Linux liviana y simple optimizada para la arquitectura x86/x64.

Arquitectura del SO	ARM 32 bits y 64 bits.
Última modificación	3 de Abril de 2018.
Nombre del fichero	ArchLinuxARM-rpi-3-latest.tar.gz
Tamaño del fichero	1,2 GB.
Gestión de ficheros	PCManFM.
Gestor de ventanas	Xfwm (XFCE Windows manager)
Colección de compiladores GNU (gcc)	7.3.0-1.
Entorno de usuario	XFCE4 1.6.1-1
Administrador de paquetes	Pacman
Método de actualización	Pacman
Sistemas de ficheros soportados	F2FS/BTRFS/EXT2/ EXT3/ EXT4.
Enlace de descarga	http://os.archlinuxarm.org/os/ArchLinuxARM-rpi-2-latest.tar.gz http://os.archlinuxarm.org/os/ArchLinuxARM-rpi-3-latest.tar.gz

Tabla 8. Tabla de configuración de frecuencias de ArchLinux ARM

2.2.4.5. Sistema Operativo Fedora 27 ARM. El Proyecto Fedora es otro viejo conocido por los *linuxeros* desde hace ya muchos años, cuyo origen parte de una gran colaboración *Open Source* llevada de la mano de la comunidad de usuarios de *Red Hat Inc.*

Las ediciones que ha tenido esta distribución de Linux, se han originado partiendo desde un conjunto de paquetes base comunes para todas las versiones, nosotros nos basaremos en la versión de escritorio. La versión base posee diversas versiones de escritorio, siendo el entorno predeterminado *GNOME 3.26*, pero también da la posibilidad de emplear *KDE Plasma*, *Xfce*, *MATE*, *Cinnamon*, *LXDE*, *Sugar on a Stick* y *LXQt*.

Esta última versión sigue sorprendiendo con su gran estabilidad, quizás por su tendencia conservadora, con actualizaciones constantes (casi a diario) permitiendo tener un núcleo lo más actualizado y seguro posible.

Como sucede con *ArchLinux*, *Fedora* es una distribución no apta para todos los públicos, centrándose en usuarios avanzados y fundamentándose en mi experiencia en particular, al basarse en una solución *Red Hat* (competencia empresarial de *Suse Linux Enterprise* o *CentOS*), que siempre ha tenido una orientación hacia un público experto en sistemas servidores más que en usuarios de escritorio. Aun así, el entorno que proporciona es claro y sólido, con un enfoque que la comunidad de soporte aprueba.

Otro aspecto, similar a *OpenSuse TW (Tumbleweed)*, es que es una distribución de tipo *Rolling Release*, con frecuencias de actualizaciones muy rápidas, que quizás las mentes más conservadoras no aceptarán, pero que a la larga demuestran la fiabilidad que ofrece esta

distribución y el esfuerzo que hay detrás. Estas actualizaciones suelen ser parches de seguridad, revisiones de Kernel, librerías y versiones de aplicaciones, lo cual hace parecer que este sistema operativo se comporta como un ente vivo.

Un siguiente punto que será vital respecto al empleo de esta distribución, es el consumo de recursos, que aunque dependerá sobre todo inicialmente del contundente escritorio en GNOME, que nos hará plantearnos si empleamos otra opción de escritorio para poder alcanzar un balance correcto respecto a tener una correcta funcionalidad del escritorio con un rendimiento razonable (adelantando que quizás tengamos que pasarnos a *LXQt Desktop Panel*).

Y como novedad para nuestra Pi, incluye una utilidad llamada *Fedora Writer* que ahora permite crear tarjetas SD de arranque para dispositivos con procesador ARM, lo cual es una gran noticia para los que trabajamos con este tipo de dispositivos.

Y finalmente, respecto al software instalado en *Fedora 27*, encontraremos como en la mayoría de distribuciones con la versión de *LibreOffice 5.4*, también con el navegador web *Firefox 57*.

Concluyendo con la muestra de la última distribución a analizar, mostramos a continuación la tabla de características del sistema operativo.

	
Año de creación del proyecto	Diciembre de 2014
Distribución nativa	Desarrollado por la comunidad Red Hat
Versión del núcleo	Linux Kernel 4.13.9.
Arquitectura del SO	ARM 32 bits y 64 bits.
Última modificación	12 de Febrero de 2018.
Nombre del fichero	Fedora-Workstation-armhfp-27-1.6-sda.raw.xz
Tamaño del fichero	1,3 GB.
Gestión de ficheros	GNOME Commander. PCManFM (LXQt).
Gestor de ventanas	Metacity
Colección de compiladores GNU (gcc)	7.2.1
Entorno de usuario	GNOME, KDE, Xfce, LXDE, MATE-Compiz, LXQt
Administrador de paquetes	RPM Package Manager
Método de actualización	DNF (Fork de YUM)
Sistemas de ficheros soportados	EXT2/ EXT3/ EXT4 (Nativo).
Enlace de descarga	https://arm.fedoraproject.org/es/

Tabla 9. Tabla de configuración de frecuencias de Fedora 27

3. Puesta en acción: Fase de pruebas y análisis de las distribuciones

3.1. Aproximación y estrategias a llevar en el análisis de las distribuciones

En este punto comenzaremos y pondremos en práctica todo el trabajo de investigación y de búsqueda de fuentes de información sobre la que parte nuestro estudio particular.

Puesto que el tiempo que se empleará para realizar todo la labor de configuración de los sistemas operativos va a ser bastante extenso, estableceremos una serie de pautas y estrategias para poder optimizar al máximo el trabajo realizado y poder portarlo de una distribución a otra, siempre que nos sea factible y de este modo aprovecharemos recursos, ahorraremos tiempo y garantizaremos que podamos alcanzar la totalidad de pruebas y objetivos previstos.

3.1.1. Aproximación real al problema

Debido a los motivos expresados anteriormente, los medios que disponemos son limitados, solamente dispongo de una placa Raspberry Pi 3 Model B., y para que el trabajo desarrollado sobre esta sea equitativo en las distribuciones también se usará el mismo soporte físico para todas las distribuciones. Esto como es normal, limita bastante el poder realizar muchas pruebas en paralelo y nos exige que tengamos que nuestra metodología sea minuciosa para poder progresar adecuadamente.

Todo este proceso nos va a suponer mucho tiempo de trabajo delante de nuestra SBC, instalar las imágenes en el soporte físico, instalar y configurar Linux, realizar toda la batería de pruebas, crear y modificar scripts de configuración, instalar repositorios y paquetes, configurar aplicaciones, capturas, etc. y todo esto multiplicarlo por todas las distribuciones a estudiar.

3.1.2. Estrategias a realizar

A continuación expondremos que estrategias prácticas realizaremos para poder optimizar tiempo y recursos.

- Para comenzar, emplearemos el software libre **Win32diskimager** para Windows o el comando **DD** en Linux, para realizar copias de seguridad de las imágenes conforme se vayan instalando aplicaciones, realizando configuraciones, etc. lo que nos garantizará que de cara a posibles contingencias, ya sean deterioro del contenido de la tarjeta o de su hardware en sí, podamos tener un punto de partida sin haber perdido parte o la totalidad del trabajo desarrollado, lo cual es de vital importancia.

Cada vez que se trabaje con un sistema operativo se realizará una copia al menos diaria en un disco duro externo con gran capacidad, en la que guardaremos varias versiones de cada sistema operativo por si fuera necesario volver a un estado concreto.

- Se realizarán scripts de configuración para todos los procesos que precisen labores

repetitivas, como por ejemplo serán las pruebas de rendimiento de memoria, de mono y multi-núcleo del procesador, pruebas de lectura/escritura de memoria, disco, etc.

- En caso de que se produjera un error de hardware que afecte a la propia placa, el suministro está garantizado en varias tiendas *on-line* en las que ya tenemos registro, con un plazo de entrega previsto bastante reducido.
- Como opción, aunque no es viable para las pruebas y análisis de rendimiento, existe un par de proyectos libres que permiten ejecutar el Kernel de la Raspberry Pi en *QEMU*¹⁴.

Se podría emplear como alternativa para realizar pruebas y configurar las imágenes antes de realizar las pruebas de rendimiento sobre el hardware real, como acaba empleando de base los ficheros de imagen, posteriormente se pueden copiar en la tarjeta *MicroSD* y ejecutar en nuestra Pi.

Y en un principio no sería necesaria ninguna acción preventiva más de cara a asegurar que el transcurso y finalización exitosa del proyecto.

3.2. Análisis y pruebas del SO Raspbian Stretch with Desktop de 32 bits

Empezaremos nuestras primeras pruebas cargando el fichero de imagen *2018-03-13-raspbian-stretch.img*, descargada de la web oficial en la tarjeta *MicroSD* de características expuestas en el punto 2.2.1 y posteriormente la introduciremos en nuestra Pi.

Nada más arrancar el sistema operativo Raspbian, observamos que emplea la particularidad y funcionalidad de software llamada “*Out Of The Box*” (OOTB¹⁵), la cual agiliza, tras transferir la imagen a la tarjeta SD con el software de copia de imágenes, se ejecute el sistema sin ningún tipo de instalación previa, configuración y modificación alguna, lo cual garantiza la empleabilidad del sistema operativo para todo tipo de usuarios.

Podemos examinar en un principio que el sistema arranca con una rapidez relativa, sobre todo conociendo las características y teniendo en cuenta la velocidad de lectura que nos permite nuestra tarjeta de memoria. Además, la mayoría de aplicaciones se ejecutan también en un tiempo aceptable, y el intercambio y paso entre tareas lo hace casi instantáneamente, lo cual le otorga una suavidad, ligereza y soltura que es de bastante agradecer, sobretodo conociendo a priori las capacidades técnicas de la placa.

A continuación seguimos mostrando la relación de las aplicaciones que vienen instaladas con esta versión de Raspbian.

¹⁴ QEMU. Software de código abierto virtualizador y emulador de ordenadores personales.

¹⁵ OOTB. Funcionalidad software que agiliza su instalación y configuración sin intervención por parte del usuario.

Aplicaciones preinstaladas	
Programación	BlueJ Java IDE, Geany, Greenfoot Java IDE, Mathematica, Node-RED, Python 2 (IDLE), Python 3 (IDLE), Scratch, Scratch 2, Sense HAT Emulator, Sonic Pi, Thonny Python IDE, Thony (Simple Mode), Wolfram
Oficina	LibreOffice Base, LibreOffice Calc, LibreOffice Draw, LibreOffice Impress, LibreOffice Math, LibreOffice Writer.
Internet	Chromium, Claws Mail, Raspbery Pi Resources, The MagPi, VNC Viewer.
Juegos	Minecraft Pi, Python Games.
Accesorios	Administrador de tareas, Calculator, Gestor de archivos, LXTerminal, PDF Viewer, SD Card Copier, Text Editor, Visor de imágenes, Xarchiver.
Preferencias	Add/ Remove Software, Appaerence Settings, Audio Device Settings, Configuración de Raspbery Pi, Main Menu Editor, Teclado y Ratón.

Tabla 10. Tabla con la lista de aplicaciones preinstaladas en Raspbian

3.3.1. Datos, variables y pruebas de rendimiento con parámetros de configuración de hardware nativos

Iniciaremos esta parte del primero de los análisis de los sistemas cuantificando la experiencia de uso que nos transmite personalmente, valorando y puntuando la facilidad de instalación, la fluidez y la estabilidad con variables enteras del 1 al 10 (por ejemplo, tomando como referencia la Fluidez, del 1 a 2,5 Muy Fluido, de 2,5 a 5 Poco Fluido, de 5 a 7,5 Fluido de 7,5 a 10 Muy Fluido). Además revisaremos que sea compatible todo el hardware, y en caso de que alguno de los componentes en función de su importancia lo cuantificaremos mediante un porcentaje. En la siguiente tabla podemos ver las primeras valoraciones datos cuantificados para Raspbian de la experiencia de uso desde el punto de vista personal.

Puntuaciones de experiencia de usuario	
Facilidad de instalación	10 (Out of the box)
Fluidez	7
Estabilidad	9
Compatibilidad con el hardware	100%

Tabla 11. Tabla de datos con las puntuaciones de experiencia de usuario en Raspbian

Una vez instalado el monitor de rendimiento *RPiMonitor* ya tenemos de una forma sencilla acceso a cierta cantidad de datos para poder cuantificar el rendimiento. Observaremos los primeros datos accediendo tanto en *localhost* como a través de la IP con un equipo de la misma red. La información sobre APIs de video la logramos gracias al comando *glxinfo*. En la tabla que se expone continuación podemos ver los resultados.

Datos y variables del entorno	
Tiempo de arranque	23 segundos
Tiempo de apagado	25 segundos
Memoria RAM usada/total	419 / 927 MB (45,19%)
Espacio de disco usado/total	1,3 GB de 62 GB (2,09%)
Tamaño SWAP usado/disponible	10 / 99 MB (9,9%)
Temperatura CPU Media	38 ° C
Gestión de energía (KW consumida en 1 hora)	Potencia consumida 4,5 W/hora
Compatibilidad con APIs de video y 3D	EGL,OpenGL ES 3.0, OpenVG, OpenMAX

Tabla 12. Tabla de datos y variables del entorno en Raspbian

Y para concluir con la última toma de datos pasaremos a realizar la batería de pruebas de rendimiento. Comenzando con **Sysbench**, en el que parametrizaremos cada test con los siguientes valores, lo cuales se repetirán en todas las versiones de sistemas operativos que vayamos a probar.

- Para las pruebas de Memoria RAM, estableceremos el tamaño de búfer (*memory-block-size*) de un Megabyte, y un total de memoria de 10 Gigabytes. Este proceso se repetirá hasta alcanzar el máximo fijado.
- Para las pruebas de CPU emplearemos un test de cálculo de números primos y le estableceremos como valor 20.000 unidades. Aquí también obtendremos las medidas de temperatura que se obtienen durante las pruebas de carga.
- Para las pruebas de almacenamiento (IO), se empleará un tamaño de 10 Gigas, espacio que previamente habrá que preparar para posteriormente ejecutar la prueba de lectura y escritura aleatoria en disco (*random read/write*) y con una duración máxima de 300 segundos, y finalmente habrá que limpiar los ficheros utilizados.

Respecto a la aplicación de *benchmarks* de video **GLMark2** no tiene particularidad ninguna a la hora de parametrizar la prueba, tiene más complicación sobretodo en sistemas débiles, ya que no se incluye en los repositorios oficiales, y es preciso descargar los fuentes del *git* oficial y compilarlos en nuestra máquina. A partir de ahí, solamente con ejecutar el comando `glmark2`, se realizará la prueba de video. Esta prueba la repetiremos en al menos un par de ocasiones, para tener datos de resultado claros y objetivos, además en los anexos se añadirá un pequeño manual de cómo se puede realizar este proceso.

Para la herramienta **Hdparm** como hemos comentado anteriores puntos, la emplearemos para realizar pruebas y mediciones de velocidad de lectura. Introduciremos dos parámetros, el primero es el parámetro **t** (*Timing buffered disk*), para ordenar que se mida la velocidad real de lectura del contenido almacenado en la caché de disco (buffer caché) sin que se almacene nada previamente es la misma buffer caché, es decir, toda la información provendrá del contenido almacenado en el disco. El segundo parámetro que emplearemos será **T** (*Timing chache reads*),

con el que mediremos la velocidad de lectura de los datos de la buffer caché, sin tener que acceder como ocurría de forma análoga con el otro parámetro, al contenido del disco. De aquí obtendremos los dos resultados solicitados por los parámetros, *Timing caché reads* y *Timing buffered disk*.

También decir, que para que los datos obtenidos de las pruebas de lectura sean significativos, repetiremos el proceso en tres ocasiones. Automatizaremos este proceso mediante un pequeño script con el siguiente comando el cual meteremos en un script:

```
for a in 1 2 3;
do sudo hdparm -tT /dev/mmcblk0;
done
```

Y para concluir con las herramientas de pruebas de rendimiento, con el comando **DD** (*duplicate data*), realizaremos las pruebas de medición de velocidad escritura de la *MicroSD*, empleando el comando que se expone a continuación:

```
dd if=/dev/zero of=/tmp/prueba.dat bs=1024 count=2000000 conv=fdatasync
```

Se especificará como origen el archivo *if =/dev/zero*, que generará una cadena de ceros con las características que definidas mediante *count=2000000*, en el que se escribirán 2.000.000 de bloques del tamaño definido por el otro parámetro *bs=1024*, es decir, de 1024 bytes. Como fichero de salida emplearemos el fichero *of=/tmp/prueba.dat*. Y para terminar con el parámetro *conv=fdatasync*, obtendremos los resultados de escritura que sean lo más próximos posibles a la velocidad de escritura real del disco. De no emplear este último, solo se obtendría el tiempo en volcar la totalidad de información a la cache de disco.

Ahora para finalizar la prueba, solamente nos quedaría borrar el fichero generado, simplemente con emplear el comando que ponemos a continuación:

```
sudo rm -f /tmp/prueba.dat
```

Como nota final a todo lo expuesto, todas las pruebas se realizarán sin ningún proceso en segundo plano o tarea en ejecución, mantenido nuestra Pi en completo reposo, de cara a garantizar que todos los recursos hardware de la máquina están disponibles para realizar las pruebas, y así los resultados sean lo más fieles posibles en todos los entornos con los que vamos a trabajar.

Seguidamente podemos ya observar los primeros resultados obtenidos de las pruebas de rendimiento de todas las herramientas en la siguiente tabla:

Puntuaciones obtenidas en las Pruebas de rendimiento	
Tiempo total Sysbench memoria RAM	1.0520 segundos
Tiempo total Sysbench IO	374.7373 segundos
Tiempo total Sysbench 1 core	373.2903 segundos
Tiempo total Sysbench 2 cores	188.4181 segundos

Tiempo total Sysbench 4 cores	94.8028 segundos
Temperatura CPU Max obtenida durante las pruebas(1 core / 2 cores / 4 cores)	41,8 °C / 46 °C / 53,6 °
GLMark2	129 puntos (46,3 °C)
Hdparm (buffer cache)	642.75 MB/seg
Hdparm (buffer disk)	20,59 MB/seg
Dataset Definition (DD) escritura	2 GB/ 186,825 seg (11 MB/seg)

Tabla 13. Tabla de datos obtenidos en las pruebas de rendimiento

Comentar de los resultados, como partimos de los primeros obtenidos, simplemente que se han realizado las pruebas sin ningún tipo de problema, y se han obtenido los datos con un tiempo de espera bastante razonable. Respecto a la temperatura de la CPU, significativamente ha aumentado en las pruebas específicas para el procesador, y las rendimiento de video (47,018 °C Max. Alcanzados) y ha aumentado dentro de unos parámetros moderados en tal que se aumentaban los núcleos de proceso puestos a prueba, sin alcanzar valores que pudieran poner en riesgo nuestra maquinita.

De parte de las pruebas realizadas, aquellas más significativas, anexaremos al final del trabajo algunas de las capturas realizadas, para que se pueda ver las pautas de trabajo desarrolladas y sobre las que hemos obtenido y seleccionado los datos a comparar.

3.3.2. Datos, variables y pruebas de rendimiento con parámetros de configuración de hardware modificados.

Repetiremos el proceso con los cambios de parámetros de configuración hardware modificado indicado en la tabla nº 4 del punto 2.2.2 de esta memoria. Nada más arrancarlo el tiempo de arranque se ha reducido mínimamente y la fluidez ha aumentado, hay menor tiempo de respuesta a la hora de realizar las acciones, y tras un tiempo de uso y prueba de algunas de las aplicaciones aparentemente la estabilidad no se ha visto comprometida, pero la pondremos a prueba con las pruebas de rendimiento que realizaremos posteriormente.

Puntuaciones de experiencia de usuario	
Fluidez	7,5
Estabilidad	9

Tabla 14. Tabla de datos con las puntuaciones de experiencia de usuario en Raspbian modificado

Estas son las primeros datos de las variables, en las que como se puede observar han variado notoriamente debido a las modificaciones de configuración realizadas.

Datos y variables del entorno	
Tiempo de arranque	21 segundos
Tiempo de apagado	23 segundos

Temperatura CPU Media	39 ° C
Gestión de energía (KW consumida en 1 hora)	Potencia consumida 5 W/hora

Tabla 15. Tabla de datos y variables del entorno en Raspbian modificado

Y tras hacer las pruebas de rendimiento los datos obtenidos son los siguientes:

Puntuaciones obtenidas en las Pruebas de rendimiento	
Tiempo total Sysbench memoria RAM	1,0069 segundos
Tiempo total Sysbench IO	300,3306 segundos
Tiempo total Sysbench 1 core	333,4337 segundos
Tiempo total Sysbench 2 cores	173,4210 segundos
Tiempo total Sysbench 4 cores	153,4990 segundos
Temperatura CPU Max obtenida durante las pruebas(1 core / 2 cores / 4 cores)	45,1 °C / 53,7 °C / 49,4 °C
GLMark2	133 puntos (52,4 °C)
Hdparm (buffer cache)	720,88 MB/segundo
Hdparm (buffer disk)	20,60 MB/segundo
Dataset Definition (DD) escritura	2 GB/ 159,925 seg (12MB/seg)

Tabla 16. Tabla de datos obtenidos en las pruebas de rendimiento en Raspbian modificado

Los datos obtenidos de las pruebas finales han sido más que concluyentes, **casi todos los resultados han mejorado como cabía esperar**. El aumento de todos los parámetros de configuración hardware para mejorar el rendimiento le ha sentado fenomenal, los datos lo confirman. La temperatura del procesador ha aumentado simplemente un grado centígrado de media respecto a las anteriores pruebas sin los *tweaks* aplicados, con lo que las medidas paliativas para refrigerar la placa (ventilador y disipadores) han sido efectivas, además ha sido de vital importancia de cara a controlar que no se desestabilice el sistema por sobrecalentamiento.

Solamente se ha producido un par de situaciones en las pruebas en la que no se han obtenido datos superiores y cabe remarcar, y ha sido exactamente en la prueba de *sysbench* con 4 núcleos, en los que ha sacado inferior calificación (153,4990 segundos). Esto se debe a que probablemente hemos superado la capacidad de rendimiento termal del procesador de la Raspberry Pi 3 (CPU thermal throttling¹⁶), haciendo que el rendimiento sea peor al alcanzar unos límites que lo activen. Esto podríamos solucionarlo o aumentando la refrigeración, con disipadores de mayor tamaño o forzando el modo turbo (*force_turbo*) y bajar el voltaje del procesador hasta el mínimo que continúe siendo estable. Eso sí, esto debería de hacerse

¹⁶ CPU thermal throttling. También llamada Dynamic frequency scaling, es una técnica en arquitectura de computadores en el que la frecuencia del procesador se ajusta automáticamente al vuelo, dependiendo de las necesidades en el momento, con el propósito de conservar energía y reducir la cantidad de calor generado en el chip.

teniendo en cuenta que estaremos vulnerando la garantía de nuestra placa. En las siguientes figuras podemos observar la pérdida de rendimiento obtenida durante la realización de las pruebas.

```
Running the test with following options:
Number of threads: 4
Doing CPU performance benchmark
Threads started!
Done.
Maximum prime number checked in CPU test: 20000

Test execution summary:
total time: 94.8028s
total number of events: 10000
total time taken by event execution: 379.1344
per-request statistics:
  min: 36.89ms
  avg: 37.91ms
  max: 95.73ms
  approx. 95 percentile: 37.55ms

Threads fairness:
events (avg/stddev): 2500.0000/13.98
execution time (avg/stddev): 94.7836/0.00
```

Figura 6. Captura de Sysbench CPU con 4 threads

```
Test execution summary:
total time: 153.4990s
total number of events: 10000
total time taken by event execution: 613.9002
per-request statistics:
  min: 32.79ms
  avg: 61.39ms
  max: 183.87ms
  approx. 95 percentile: 75.89ms

Threads fairness:
events (avg/stddev): 2500.0000/15.41
execution time (avg/stddev): 153.4750/0.01
```

Figura 7. Captura de Sysbench CPU con 4 threads con Overclock

La segunda situación es que no ha sido posible en Raspbian aumentar los parámetros de frecuencia de acceso a la tarjeta de memoria, ni con frecuencias inferiores a los 100 MHz, en los que tras habilitarlo en el fichero *config.txt* el sistema operativo no arrancaba. Volveremos a realizar pruebas en los siguientes sistemas operativos a analizar, para intentar concluir si el problema tiene que ver más con la tarjeta o del propio lector.

3.3. Análisis y pruebas del SO Ubuntu Mate 16.04 LTS Final Release Xenial

Continuamos las pruebas con otra de las grandes distribuciones para la Pi. Puesto que seguimos bajo un sistema *Debian*, pero esta vez en un sistema con las funcionalidades y operatividad completas de cualquier PC de escritorio y con el optimizado escritorio *MATE*.

Una vez instalada la imagen en la tarjeta, tras arrancarlo por primera vez, nos sale un asistente muy sencillo en el que podremos realizar los siguientes ajustes: *Configuración del sistema* ⇒ *Asistente de idioma* ⇒ *Configuración de la red Wifi* ⇒ *Ubicación* ⇒ *Teclado* ⇒ *Nombre de usuario*, *Nombre de equipo* y *contraseña* y para finalizar se pondrá a instalar el sistema y desempaquetar las aplicaciones preinstaladas. Tras terminar el asistente realiza las autoconfiguraciones necesarias y lanzará el escritorio.

Esta vez el tiempo de arranque ha sido muy superior al visto en Raspbian, casi el triple de tiempo, pero una vez nos situamos sobre el escritorio muchísimo más completo, simplista y

eficaz, todo se mueve con buena fluidez parecida a la que encontramos en el anterior sistema. Eso sí, nos encontramos ante un entorno de escritorio mucho más potente, con muchísimas más aplicaciones instaladas, con mayor posibilidades visuales y accesibles que nos dan un abanico de opciones mayor, recordándonos lo visto sobre un ordenador personal. El escritorio se divide en tres zonas, la barra superior de aplicaciones y configuración con la fecha y hora del sistema y accesos rápidos de configuración, el espacio de escritorio, y la barra de tareas inferior, en la que además tenemos a disposición de cuatro espacios de escritorio para desplegar nuestras ventanas.

A parte de lo comentado, un aspecto que me gustaría remarcar que tiene Ubuntu MATE 16.04 para Raspberry Pi 3, es el cuidado y trabajo que han realizado en el fichero de configuración de arranque, explicando cada variable minuciosamente, con gran cantidad de opciones y comentario exquisitamente.

```

root@pi-desktop: /home/pi
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.5.3 File: /boot/config.txt
##
## The GPU core, h264, v3d and isp share a PLL, so need to have related
## frequencies.
## pll_freq = floor(2400 / (2 * core_freq)) * (2 * core_freq)
## gpu_freq = pll_freq / [even number]
##
## The effective gpu_freq is automatically rounded to nearest even integer, so
## asking for core_freq = 500 and gpu_freq = 300 will result in divisor of
## 2000/300 = 6.666 => 6 and so 333.33MHz.
##
##
## Standard Profiles:
##          arm_freq  core_freq  sdram_freq  over_voltage
## -----
## None       700        250         400         0
## Modest     900        300         400         0
## Medium     950        333         450         2
## High       950        450         450         6
## Turbo     1000       500         500         6
##
#####
    
```

Figura 8. Captura del fichero /boot/config.txt en Ubuntu Mate

A continuación mostramos la relación de las *Aplicaciones, Lugares y Accesos de configuraciones del Sistema* que tenemos en el lanzador (barra superior del escritorio) que vienen instaladas con esta versión de Ubuntu MATE.

Aplicaciones preinstaladas	
Accesorios	Backups, Capturar pantalla, Character Map, Disks, Editor de texto Pluma, Calculator, Gestor de archivadores Engrampa, Herramientas de búsqueda MATE, Passwords and Keys, Plank y Synapse.
Acceso universal	Onboard, Screen Magnify, Screen Reader.
Educación	Scratch.
Gráficos	Selección de color MATE, Shotwell, Simple Scan, Visor de imágenes Eye of MATE
Herramientas del sistema	Analizador de uso de disco MATE, Caja, Editor de dconf, Estadísticas de energía, GDebi Package Installer, Monitor del sistema de MATE, Navegador Zeroconf de Avahi, Terminal de MATE.

Internet	Cliente de correo Thunderbird, Cliente de mensajería de Internet Pidgin, HexChat, Navegador web Firefox, Transmission, YouTube DL GUI.
Juegos	Minecraft Pi
Oficina	Atril Document Viewer, LibreOffice, LibreOffice Calc, LibreOffice Draw, LibreOffice Impress, LibreOffice Writer, MATE Dictionary.
Programación	IDLE, IDLE 3, IDLE (using Python-2.7), IDLE (using Python-3.5), Scratch, Sense HAT Emulator, Sonic Pi.
Sonido y video	Brasero, Cheese, QjackCtl, Reproductor multimedia VLC, Rhythmbox, Sonido.

Tabla 17. Tabla con la lista de aplicaciones preinstaladas en Ubuntu Mate

Lugares
Carpeta personal
Escritorio
Equipo
Red
Conectar con el servidor...
Herramienta de búsqueda de MATE
Documentos recientes

Tabla 18. Tabla con la lista de Lugares en Ubuntu Mate

Accesos de configuraciones del Sistema	
Accesorios	Bienvenido, Configuración de la interfaz GTK+ de LightDM, Hora y fecha, Printers, Red, Software Boutique, Software Updater, Software & Updates, Usuarios y grupos.
Acceso universal	Hardware, Internet y red, Personal, Visualización y comportamiento.
Bienvenido	
Centro de control	
Ayuda	
Acerca de MATE	
Bloquear la pantalla	
Cerrar la sesión de pi	
Apagar	

Tabla 19. Tabla con la lista de aplicaciones preinstaladas en Ubuntu Mate

3.3.1. Datos, variables y pruebas de rendimiento con parámetros de configuración de hardware nativos

Proseguimos con el análisis personal de la experiencia de uso del sistema *Ubuntu MATE*.

Puntuaciones de experiencia de usuario	
Facilidad de instalación	9
Fluidez	7
Estabilidad	9
Compatibilidad con el hardware	100%

Tabla 20. Tabla de datos con las puntuaciones de experiencia de usuario en Ubuntu Mate

Por lo que respecta a los datos y variables de entorno, volvemos a instalar el monitor de rendimiento *RPIMonitor* con el propósito de poder acceder a los datos que se muestran a continuación en la siguiente tabla.

Datos y variables del entorno	
Tiempo de arranque	1 minuto y 12 segundos
Tiempo de apagado	28 segundos
Memoria RAM usada/total	366,64 / 925.52 MB (39,61%)
Espacio de disco usado/total	4,38 GB de 62 GB (7,06%)
Tamaño SWAP usado/disponible	No soportada para discos Micro SD
Temperatura CPU Media	40 ° C
Gestión de energía (KW consumida en 1 hora)	Potencia consumida 5,5 W/hora
Compatibilidad con APIs de video y 3D	EGL,OpenGL ES 3.0, OpenVG, OpenMAX

Tabla 21. Tabla de datos y variables del entorno en Ubuntu Mate

Después procedemos a realizar las pruebas de rendimiento tras instalar de nuevo todas las aplicaciones de *benchmarking*. Estos han sido los resultados obtenidos:

Puntuaciones obtenidas en las Pruebas de rendimiento	
Tiempo total Sysbench memoria RAM	1,1275 segundos
Tiempo total Sysbench IO	330,0873 segundos
Tiempo total Sysbench 1 core	422,5605 segundos
Tiempo total Sysbench 2 cores	209,6975 segundos
Tiempo total Sysbench 4 cores	113,9309 segundos
Temperatura CPU Max obtenida durante las pruebas(1 core / 2 cores / 4 cores)	40,6 °C / 44 °C / 50,5 °
GLMark2	114 puntos (47,2 °C)
Hdparm (buffer cache)	570,60 MB/segundo
Hdparm (buffer disk)	20,86 MB/segundo
Dataset Definition (DD) escritura	2 GB/ 134,443 seg (15,2MB/seg)

Tabla 22. Tabla de datos obtenidos en las pruebas de rendimiento en Ubuntu Mate

3.3.2. Datos, variables y pruebas de rendimiento con parámetros de configuración de hardware modificados

Tras realizar los cambios de parámetros de rendimiento que van a ser la constante para el resto de pruebas sobre el fichero *config.txt*, comentamos de nuevo como ha sido la experiencia de usuario reflejado en la siguiente tabla.

Puntuaciones de experiencia de usuario	
Fluidez	7,5
Estabilidad	9

Tabla 23. Tabla de datos con las puntuaciones de experiencia de usuario en Ubuntu Mate overclocked

Tras ejecutar las pruebas estos son los datos de las variables obtenidas, en las que una vez más se puede observar el cambio y mejora debido a las modificaciones de configuración realizadas.

Datos y variables del entorno	
Tiempo de arranque	1 minuto
Tiempo de apagado	24 segundos
Temperatura CPU Media	41 ° C
Gestión de energía (KW consumida en 1 hora)	Potencia consumida 5 W/hora

Tabla 24. Tabla de datos y variables del entorno en Raspbian Ubuntu Mate overclocked

Una vez hechas las pruebas de rendimiento los datos obtenidos podemos verlos a continuación:

Puntuaciones obtenidas en las Pruebas de rendimiento	
Tiempo total Sysbench memoria RAM	0,9608 segundos
Tiempo total Sysbench IO	358,3077 segundos
Tiempo total Sysbench 1 core	383,3813 segundos
Tiempo total Sysbench 2 cores	200,2323 segundos
Tiempo total Sysbench 4 cores	164,9995 segundos
Temperatura CPU Max obtenida durante las pruebas(1 core / 2 cores / 4 cores)	47,8 °C / 52 °C / 49,6 °C
GLMark2	122 puntos (51 °C)
Hdparm (buffer cache)	636,53 MB/segundo
Hdparm (buffer disk)	20,87 MB/segundo
Dataset Definition (DD) escritura	2 GB/ 141,433 seg (14,5MB/seg)

Tabla 25. Tabla de datos obtenidos en las pruebas de rendimiento en Ubuntu Mate overclocked

Una vez más, las pruebas finales nos han dado resultados positivos en la mayoría de pruebas de rendimiento, repitiéndose el descenso de rendimiento en la prueba con cuatro *threads*, en la

que esta vez la disminución de rendimiento ha sido menos acusada que con las realizadas en *Raspbian*, además hemos observado y certificado otro problema que ya se veía en la anterior sistema operativo que comentaremos a continuación.

El aumento de las frecuencias para trabajo en CPU y gráfica, penalizan las pruebas de rendimiento de lectura y escritura en disco, en las de lectura más exactamente realizando las lecturas de disco con empleo de la memoria caché (*timing cached reads*) y en escritura en general, con una pérdida entorno al 12% y un 5% respectivamente.

Es algo más que evidente que se había observado en las pruebas realizadas con *Raspbian*, pero esta vez la caída de rendimiento ha sido de mayor importancia.

Aunque en las tablas anteriores ya se puede observar las diferencias, en las siguientes instantáneas podemos ver en detalle las capturas con el comando de copia *DD* de las pruebas realizadas al respecto.

```
root@pi-desktop:/home/pi# dd if=/dev/zero of=/tmp/prueba.dat bs=1024 count=2000000 conv=
fdatasync
2000000+0 records in
2000000+0 records out
2048000000 bytes (2,0 GB, 1,9 GiB) copied, 134,443 s, 15,2 MB/s
```

Figura 9. Captura de prueba de rendimiento DD en Ubuntu MATE

```
root@pi-desktop:/home/pi# dd if=/dev/zero of=/tmp/prueba.dat bs=1024 count=2000000 conv=fdatasync
2000000+0 records in
2000000+0 records out
2048000000 bytes (2,0 GB, 1,9 GiB) copied, 141,433 s, 14,5 MB/s
```

Figura 10. Captura de prueba de rendimiento DD en Ubuntu MATE con overclock

Y las capturas realizadas durante las pruebas con el script con el comando *hdparm*.

```
root@pi-desktop:/home/pi# ./hdparm_test.sh
/dev/mmcblk0:
Timing cached reads: 1126 MB in 2.00 seconds = 562.37 MB/sec
Timing buffered disk reads: 64 MB in 3.07 seconds = 20.88 MB/sec
/dev/mmcblk0:
Timing cached reads: 1142 MB in 2.00 seconds = 570.60 MB/sec
Timing buffered disk reads: 64 MB in 3.07 seconds = 20.86 MB/sec
/dev/mmcblk0:
Timing cached reads: 1144 MB in 2.00 seconds = 571.48 MB/sec
Timing buffered disk reads: 64 MB in 3.07 seconds = 20.86 MB/sec
```

Figura 11. Captura de prueba de rendimiento HDPARM en Ubuntu MATE

```
• root@pi-desktop:/home/pi
Archivo Editar Ver Buscar Terminal Ayuda
root@pi-desktop:/home/pi# ./hdparm_test.sh
/dev/mmcblk0:
Timing cached reads: 1274 MB in 2.00 seconds = 636.53 MB/sec
Timing buffered disk reads: 64 MB in 3.08 seconds = 20.81 MB/sec
/dev/mmcblk0:
Timing cached reads: 1290 MB in 2.00 seconds = 644.88 MB/sec
Timing buffered disk reads: 64 MB in 3.07 seconds = 20.87 MB/sec
/dev/mmcblk0:
Timing cached reads: 1262 MB in 2.00 seconds = 631.02 MB/sec
Timing buffered disk reads: 64 MB in 3.07 seconds = 20.84 MB/sec
root@pi-desktop:/home/pi#
```

Figura 12. Captura de prueba de rendimiento HDPARM en Ubuntu MATE con overclock

El que suceda este fenómeno se debe probablemente a posibles inconsistencias por el aumento de las diferentes frecuencias, lo cual puede estar causando cierta inestabilidad y esperemos que no genere otro tipo de consecuencias no deseables, pero es bueno conocer este

comportamiento si el aumento de rendimiento por parte de la CPU y RAM en el sistema nos permite asumir esa mínima pérdida de rendimiento por parte del disco.

Una última cosa que me gustaría comentar además, es que *Ubuntu MATE* no soporta partición de intercambio (swap) en tarjetas Micro SD debido a la velocidad de transferencia que soportan este tipo de medios, con lo que la memoria RAM realiza todo el trabajo.

3.4. Análisis y pruebas del SO OpenSuse 42.3 Tumbleweed de 64 bits

Antes de comenzar *OpenSuse* nos ofrece varias versiones de escritorio para su sistema operativo, deseable sería probarlas todas, pero nuestro tiempo es más que limitado.

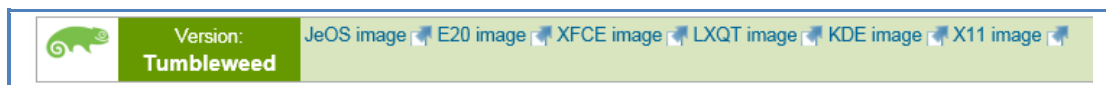


Figura 13. Versiones de escritorio para el SO OpenSuse para Raspberry Pi 3

Conociendo que *KDE* es un gran escritorio, pero que consume muchos más recursos, con lo que será fácil desestimarlos de primeras y bueno tanto *JeOS* y *X11* son demasiado básicos como para considerarlo un sistema de escritorio. *XFCE* lo habíamos visto funcionar en el sistema operativo *Xubuntu* es tremendamente ligero y muy potente, y nos es familiar a *GNOME*. *LXQT* es algo más ligero que éste último, y también lo he visto funcionar en equipos antiguos y se movía con bastante soltura. Pero Finalmente y por experiencia personal una vez más, nos decantamos por la versión *XFCE*. Si, posteriormente se dispone de más tiempo se probarán otras imágenes.

La imagen viene en formato *RAW* comprimido, emplearemos el comando *dd* de Linux para transferirlo a la tarjeta. Una vez colocada la tarjeta en la Pi, se observa que el método de arranque es diferente al empleado en las versiones *Debian*, sin preguntarnos nada y sin ningún tipo de asistente y tras un largo tiempo de autoconfiguración directamente nos sale el login de máquina, empleando la característica de “*Out Of The Box*”, anteriormente mencionada.

Tras entrar en el escritorio, podemos observar que nos encontramos ante un entorno bastante potente, muy parecido en características a lo visto en *Ubuntu MATE*, e integrando la aplicación de administración *YaST*¹⁷, que nos muestra un panel de control y de administración excelente.

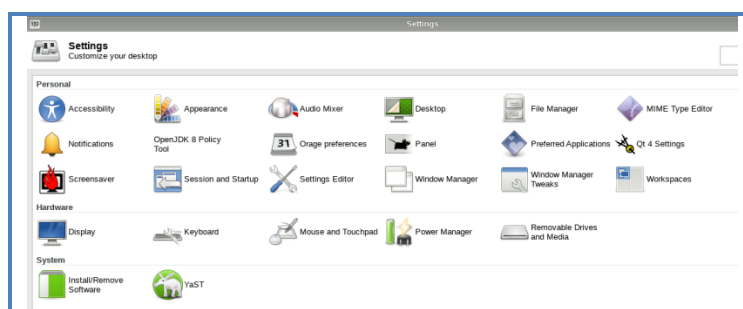


Figura 14. Captura del Centro de control YaST2 para el SO OpenSuse para Raspberry Pi 3

El entorno de escritorio con *Xfce*, en primera instancia se mueve con agilidad, pero el entorno

¹⁷ YaST. YaST, acrónimo de Yet another Setup Tool (Otra Herramienta de Configuración Más) es una aplicación de openSUSE, que facilita la administración del sistema y la instalación de software.

no responde con la fluidez vista en los otros sistemas una vez se comienza a trabajar con varias tareas en éste, pese a que estamos en una versión que se supone aprovecha las características del procesador de 64 bits. Ciertas tareas y el intercambio de ventanas no responde con la soltura esperada, de hecho en ocasiones ha sido una acción desesperante, y eso que emplea una SWAP de mayor tamaño (500 MB), pero el entorno de escritorio consume gran cantidad de RAM, y eso que hemos seleccionado uno de los menos pesados.

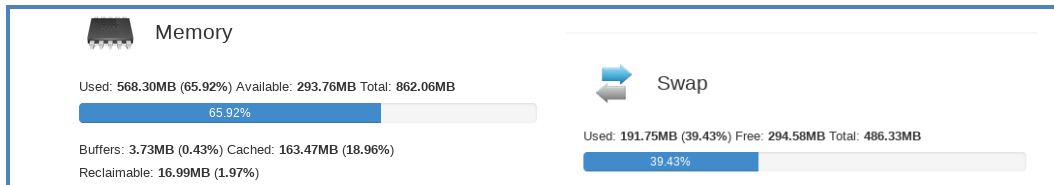


Figura 15. Capturas de Memoria RAM/SWAP consumida en el SO OpenSuse para Raspberry Pi 3

Además que pese a que integra todos los drivers del hardware de la Pi, he tenido problemas de conexión, de hecho en un par de ocasiones se me ha congelado el sistema y ha dejado de responder empleando la tarjeta de red inalámbrica, cosa que se ha solventado utilizando la conexión por Ethernet.

Por comentar no solo los problemas encontrados, hemos empleado la herramienta YaST2 para instalar tanto las herramientas para las pruebas de rendimiento Sysbench y GImark2 como el monitor RPI-Monitor y es una auténtica gozada, lo bien implementado que está la función de búsqueda de aplicaciones y el agregado de repositorios.

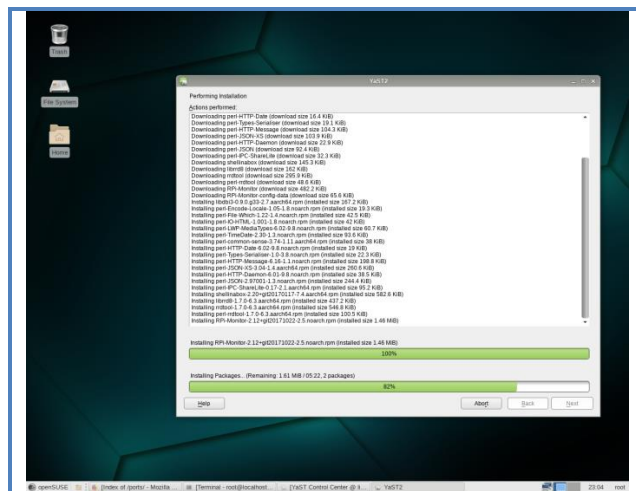


Figura 16. Captura de instalación de herramienta de monitorización en OpenSuse con YaST2

Pasaremos seguidamente a mostrar la relación de las aplicaciones preinstaladas que muestra el menú de inicio.

Aplicaciones preinstaladas	
Favoritos	
Recientemente usados	
Todos	

Accesorios	Archive Manager, Files, Calculator, GNOME Character Map, Leafpad, Main Menu, Notes, Orage Globaltime, Passwords and Keys, Screenshots, Xfce Application Finder.
Juegos	Chess, Five or More, Four-in-a-row, Klotski, Lights Off, Mahjongg, Mines, Nibbles, Quadrapassel, Robots, Sudoku, Swell Foop, Tali, Tetravex.
Gráficos	Document Viewer, LibreOffice Draw, Ristretto Image Viewer, Shotwell, Simple Scan.
Internet	Chromium Web Browser, Firefox, Pidgin, Remmina, Thunderbird, TigerVNC Viewer, Transmission.
Multimedia	Brasero, Pragma, PulseAudio Volume Control, Sound Juicer, Videos
Office	Dictionary, Document Viewer, LibreOffice, LibreOffice Base, LibreOffice Calc, LibreOffice Draw, LibreOffice Impress, LibreOffice Writer, Orage Calendar.
Configuraciones	Instalar/Desinstalar Software, Administrador de configuraciones, YaST.
Sistema	Bulk Rename, Network Connections, OpenJDK 1.8.0 Policy Tool, Print Settings, Task Manager, Thunar File Manager, Xfce Terminal.

Tabla 26. Tabla con la lista de aplicaciones preinstaladas en OpenSuse

3.4.1. Datos, variables y pruebas de rendimiento con parámetros de configuración de hardware nativos

Continuamos con el análisis personal de la experiencia de uso del sistema OpenSuse.

Puntuaciones de experiencia de usuario	
Facilidad de instalación	9
Fluidez	5
Estabilidad	7
Compatibilidad con el hardware	85%

Tabla 27. Tabla de datos con las puntuaciones de experiencia de usuario en OpenSuse

Continuamos con las tablas de datos y variables de entorno, una vez mas basándonos en el uso del monitor de rendimiento *RPiMonitor* para poder acceder a los datos que se muestran en la siguiente tabla.

Datos y variables del entorno	
Tiempo de arranque	2 minutos
Tiempo de apagado	35 segundos
Memoria RAM usada/total	568,30 / 862,06 MB (65,92%)
Espacio de disco usado/total	3,11 GB de 56,1 GB (5,55%)

Tamaño SWAP usado/disponible	191,75 / 486,33 MB (39,43%)
Temperatura CPU Media	-- ° C
Gestión de energía (KWs consumidos en 1 hora)	Potencia consumida 3,5 W/hora
Compatibilidad con APIs de video y 3D	EGL,OpenGL ES 3.0, OpenVG (no compatible con hardware acelerado)

Tabla 28. Tabla de datos y variables del entorno en OpenSuse

Una vez más, habilitamos el modulo del kernel para que tenga compatibilidad con el software de *mesa VC4*, en nuestro fichero de configuración *config.txt* con el parámetro *dtoverlay=vc4-kms-v3d* y viendo que el rendimiento no ha sido lo óptimo que cabía esperar, también aumentaremos la memoria de la *GPU* a *128 MBs* empleando el parámetro *gpu_mem=128*. Una vez realizadas las pruebas de rendimiento éstos han sido los resultados obtenidos, posteriormente comentaremos la serie de problemas encontrados.

Puntuaciones obtenidas en las Pruebas de rendimiento	
Tiempo total Sysbench memoria RAM	segundos
Tiempo total Sysbench IO	-- segundos
Tiempo total Sysbench 1 core	119,08 eventos/segundo
Tiempo total Sysbench 2 cores	230,46 eventos/segundo
Tiempo total Sysbench 4 cores	468,56 eventos/segundo
Temperatura CPU Max obtenida durante las pruebas(1 core / 2 cores / 4 cores)	-- ° C / -- ° C / -- ° C
GLMark2	13 puntos (-- ° C)
Hdparm (buffer cache)	461,90 MBs/segundo
Hdparm (buffer disk)	21,70 MBs/segundo
Dataset Definition (DD) escritura	2 GB/ 127,606 segs. (16 MBs/seg)

Tabla 29. Tabla de datos obtenidos en las pruebas de rendimiento en OpenSuse

Para comenzar con los problemas encontrados a la hora de realizar las pruebas de estrés, no nos ha sido posible monitorizar la temperatura, seguramente por algún problema o bug del kernel que no nos permite acceder a la constante.

```
/sys/devices/virtual/thermal/thermal_zone0/temp:Invalid argument
```

Lo cual ya nos podemos adelantar a que las pruebas de redimiento con overclock probablemente no va ser posible de realizar, aun así lo intentaremos.

Otro de los problemas encontrados es que pese a poner la parametrización para poder utilizar la aceleración por hardware, no funciona en el sistema operativo, ignorando dicho parámetro. Tras realizar las pruebas de rendimiento, observamos que no *OpenSuse* no es del todo compatible con *OpenGL*, generando errores a la hora de realizar las pruebas, pero sí que lo es

con *OpenGL ES*, con lo cual en vez de emplear el ejecutable de `glmark2` usaremos `glmark2-es2`. En la siguiente captura podemos observar los resultados obtenidos.

```
[loop] fragment-steps=5:fragment-uniform=true:vertex-steps=5: FPS: 16 FrameTime:
62.500 ms
=====
glmark2 Score: 13
=====
```

Figura 17. Captura de resultado de GImark2 ES en el SO OpenSuse para Raspberry Pi 3

También comentar que la versión de *Sysbench* que hay para *AArch64*¹⁸ en *OpenSuse* es la *1.0.11*, superior a las versiones empleadas para realizar las anteriores pruebas (*versión 0.4.12*) y al realizar dichas pruebas de ficheros, con ésta se han obtenido datos erróneos y no han podido realizarse correctamente, con lo que descartamos también los resultados para la comparativa.

Pero no todo van a ser problemas, gracias al trabajo desarrollado y las pruebas realizadas en de este sistema operativo, hemos podido comprobar y demostrar tras la observación de los resultados obtenidos en los test de rendimiento de CPU en *Sysbench*, que la compatibilidad con el procesador de 64 Bits de *OpenSuse* avala que emplear un sistema compatible con la arquitectura para la que está diseñada el procesador exprime mejor sus características.

Para las pruebas de CPU realizadas con uno, dos y cuatro núcleos se han obtenido resultados hasta casi ocho veces superiores a los sacados en cualquiera de los sistemas operativos de 32 bits, lo cual resulta realmente sorprendente. También comentar que esta versión de *Sysbench* superior ejecuta las pruebas con mayor rapidez y muestra los eventos por segundo calculados.

A continuación mostramos capturas de los resultados obtenidos.

```
Running the test with following options:
Number of threads: 1
Initializing random number generator from current time

Prime numbers limit: 20000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 119.08
```

Figura 18. Captura de resultado de Sysbench con 1 core en el SO OpenSuse para Raspberry Pi 3

```
Running the test with following options:
Number of threads: 2
Initializing random number generator from current time

Prime numbers limit: 20000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 230.46
```

Figura 19. Captura de resultado de Sysbench con 2 cores en el SO OpenSuse para Raspberry Pi 3

¹⁸ AArch64. Terminología empleada para las arquitecturas ARM versión 8 con procesadores de 64 bits.

```
Running the test with following options:
Number of threads: 4
Initializing random number generator from current time

Prime numbers limit: 20000
Initializing worker threads...
Threads started!

CPU speed:
events per second: 468.56
```

Figura 20. Captura de resultado de Sysbench con 4 cores en el SO OpenSuse para Raspberry Pi 3

Compararemos los en la siguiente tabla los resultados obtenidos con *Ubuntu MATE*, de los que teníamos los datos en segundos, pero como sabemos los eventos que se han realizado en total (10.000 eventos) dividiendo entre el total de segundos que ha tardado podemos sacar fácilmente la misma medida de eventos por segundo.

Puntuaciones obtenidas en las Pruebas de rendimiento		
	SO. OpenSuse	SO. Ubuntu MATE
Sysbench 1 core	119,08 eventos/segundo	26,08 eventos/segundo
Sysbench 2 cores	230,46 eventos/segundo	49,94 eventos/segundo
Sysbench 4 cores	468,56 eventos/segundo	60,60 eventos/segundo

Tabla 30. Tabla de comparativa de pruebas de CPU entre los SSOO OpenSuse y Ubuntu MATE

3.4.2. Datos, variables y pruebas de rendimiento con parámetros de configuración de hardware modificados

Como colofón final, aunque era predecible lo que voy a comentar se produjese, **no ha sido posible tampoco realizar los cambios de los parámetros de frecuencia para realizar overclocking en OpenSuse**, pese a cambiarlo como hicimos anteriormente para activar la aceleración por hardware en el fichero *config.txt*, al iniciar el sistema operativo ignora nuestros cambios realizados y deja el parámetro de configuración por defecto.

Pese a las grandes expectativas que teníamos en este sistema, una vez nos hemos encontrado frente a *OpenSuse Tumbleweed* para Raspberry Pi, nos hallamos ante un entorno al que le queda mucho trabajo que hacer y sobre todo pulir, pese a actualizarse al día, algo clave en una *Rolling Distro*, no se ha conseguido que la experiencia de usuario mejore en absoluto, de hecho las actualizaciones de sistema han provocado fallos de estabilidad y que no arranque en el siguiente inicio, teniendo que emplear nuestro plan de contingencias previsto para volver y recuperar la última imagen hecha del sistema.

Se me ha quedado un pequeño sabor de boca amargo, como he comentado antes, puede que no esté todo lo que debiera estar optimizada para trabajar con la arquitectura de 64 bits, o quizás habría que probar *OpenSuse* con un entorno gráfico básico o sin él, empleando *JeOS* o *X11* para que la experiencia sea mucho más enriquecedora y agradable a la hora de trabajar con él, pero como nuestro propósito es el emplear nuestra Pi como un sistema de escritorio esta premisa no es válida.

De otra manera, probablemente también se trate de que tengamos que probar con otro entorno de escritorio (*LXQT*), o una versión de la distribución en arquitectura de 32 bits o tal vez tratar de usar la versión estable de *OpenSuse Leap* para encontrar una experiencia de uso mucho más parecidas a las encontradas en Raspbian o Ubuntu MATE. Por supuesto, si contamos con tiempo extra para realizar más pruebas, *OpenSuse* merece una oportunidad más.

3.5. Análisis y pruebas del SO Sistema Operativo ArchLinux ARM de 32 bits

Hemos llegado al sistema operativo que seguramente suponga el mayor desafío a la hora de desplegarlo, ya que por la filosofía minimalista que emplea ArchLinux hace que seguramente tengamos que partir desde el sistema de comandos, y acostumbrados a versiones con configuraciones de inicio minimalistas, nos tocará realizar gran cantidad de esfuerzo extra para poder situarnos ante un sistema de escritorio similar a los anteriores estudiados.

Una vez descargada la imagen del fichero en formato *TAR* comprimido (*tar.gz*), las indicaciones de división de la partición que nos dan desde la página oficial ya implican tener conocimientos avanzados en sistemas operativos Linux, algo que de tratarse de un usuario inexperto, ya le haría retroceder en el intento, y eso que simplemente es particionar la tarjeta en dos, la primera en *FAT16* con un espacio reducido que contiene el sistema de arranque y el resto en formato *EXT4* para ubicar el sistema operativo. En los archivos adjuntos de esta memoria también incluiremos un pequeño manual traducido para poder instalar la imagen en Linux fácilmente con este sistema.

Una vez que arranca la imagen en unos pocos segundos, nos muestra un prompt en Linux para introducir el usuario, como cabía esperar no existe escritorio. Con lo que tendremos que emplear su sistema de gestión de paquetes y actualizaciones *pacman* para comenzar con la tarea. Si le echamos un vistazo a los drivers implementados, tenemos red, la tarjeta de red wifi también está instalada.

Esta vez la estrategia a tomar es diferente, deberemos de seguir la filosofía “*Arch Way*”, basado en la simplicidad como sofisticación última y fundamento base de funcionamiento de éste sistema, con lo que tomaremos la libertad que nos ofrece junto al control total e instalaremos por nuestra cuenta libremente el sistema de escritorio que deseemos.

Esta vez optaremos por *LXDE*, que tan buenos resultados ha dado en Raspbian. Antes de comenzar actualizaremos *pacman* con el comando `pacman -Syu`, reiniciaremos la y emplearemos de nuevo el comando: `pacman -S lxde` para que comience la instalación del entorno de escritorio, una vez se completa la instalación, listaremos y buscaremos si existe el fichero *.xinitrc*: `ls -la | grep .xinitrc`, de existir (en nuestro caso no existe), lo borrarémos con el comando: `rm .xinitrc`.

Ahora instalaremos *sddm* (*Simple Desktop Display Manager*) como administrador de pantalla (como programa de login) para sistemas *X11* y *sistemas de ventanas Wayland*.

```
sudo pacman -S sddm
sudo sh -c "sddm --example-config > /etc/sddm.conf"
```

A continuación instalamos el tema SDDM: `packer -S archlinux-themes-sddm`

y lo configuraremos para que se inicie automáticamente:

```
sudo systemctl enable sddm
sudo systemctl start sddm
```

y lo configuraremos para que se inicie automáticamente modificando el fichero de configuración `sudo nano /etc/sddm.conf`

Y agregando las siguientes líneas.

```
[Autologin]
Session=xfce.desktop
User=alarm
.....
Current=archlinux-simplyblack
```

Ya tan solo deberemos reiniciar el sistema o simplemente con ejecutar el escritorio escribiendo en el terminal `startx` y simplemente esperar a que LXDE se lance.

Ya, a partir de este momento podemos mostrar las aplicaciones preinstaladas que vienen con el sistema de escritorio elegido.

Aplicaciones preinstaladas	
Gráficos	Image Viewer.
Internet	Avahi SSH Server Browser, Avahi VNC Server Browser.
Sonido y video	LXMusic simple music player
Herramientas del Sistema	Avahi Zeroconf Browser, File Manager PCManFM, LXTerminal, Task Manager.
Preferencias	Customize Look and Feel, Default applications for LXSession, Desktop Preferences, Desktop Session Settings, Keyboard and Mouse, Monitor Settings, Preferred Applications, Setup Hot Keys.

Tabla 31. Tabla con la lista de aplicaciones preinstaladas en ArchLinux ARM de 32 bits

Como podemos observar no tenemos apenas aplicaciones, no hay navegador web, no tiene herramientas de oficina, etc. Con lo que añadiríamos aquellas que precisemos y que sean de nuestro gusto.

Ésta es la filosofía minimalista de *Arch Linux*, además que contiene todos los drivers integrados, se ha comprobado la conectividad inalámbrica y funciona a la perfección, pero tampoco existe un plugin en el escritorio, hay que hacerlo levantando la interfaz con el comando `ifconfig wlan0 up` y he instalado `netctl`¹⁹(sucesor de `netcfg`) y fácilmente hemos conseguido

¹⁹ Netctl. Herramienta basada en CLI (Interprete de línea de comandos a través de la consola), que se emplea para configurar y gestionar la conexiones de red mediante perfiles.

crear nuestro perfil de configuración de red, y ha funcionado a la perfección.

Ya pasaremos sin más dilación a instalar las aplicaciones que precisamos para realizar las pruebas de rendimiento.

3.5.1. Datos, variables y pruebas de rendimiento con parámetros de configuración de hardware nativos

A continuamos con la tabla de datos basada en el análisis personal de la experiencia de uso del sistema *ArchLinuxARM de 32 bits*.

Puntuaciones de experiencia de usuario	
Facilidad de instalación	6
Fluidez	7
Estabilidad	9
Compatibilidad con el hardware	100%

Tabla 32. Tabla de datos con las puntuaciones de experiencia de usuario en ArchLinux ARM de 32 bits

Antes de pasar con la tabla de datos y variables de entorno, no ha sido posible instalar la herramienta de monitorización y nos hemos encontrado algunos problemas con varias de las aplicaciones de rendimiento que estamos empleando, sobre todo empleando *pacman*, con lo que hemos tirado de dos aplicaciones de gestión y actualización de paquetes alternativos, *packer*²⁰ y *yaourt*²¹.

De hecho para poder instalar *glmark2* se ha tenido que emplear *yaourt* y actualizar el script para compilarlo. Con el comando `yaourt -S glmark2-git`, añadiendo en el fichero `PKGBUILD` el parámetro de la arquitectura `'armv7h'` y modificando con *nano* el segmento de código siguiente, hemos conseguido intarlo.

```
Fichero PKGBUILD - File
...
build() {
cd "${srcdir}/${pkgname}"
python2 ./waf configure --prefix=/usr --with-flavors=${GM2_FLAVORS}
python2 ./waf -j4
}
...
```

Y lo mismo nos ha pasado con el monitor *RPI-Monitor*, pero también al final lo hemos conseguido, esta vez empleando el comando `yaourt -S glmark2-git`, modificando un fallo de cadena en el fichero `PKGBUILD` una vez más (simplemente cambiar la parametrización de la url con comillas, y ha compilado. Ejecutamos el servicio con el comando `rpimonitor` y ya lo tenemos en funcionamiento. En la siguiente captura podemos apreciarlo.

²⁰ Packer. Administrador de paquetes para repositorios *pacman* y *AUR* (Arch User's Repository).

²¹ Yaourt. Administrador de paquetes complementario para repositorios *pacman* y *AUR*.

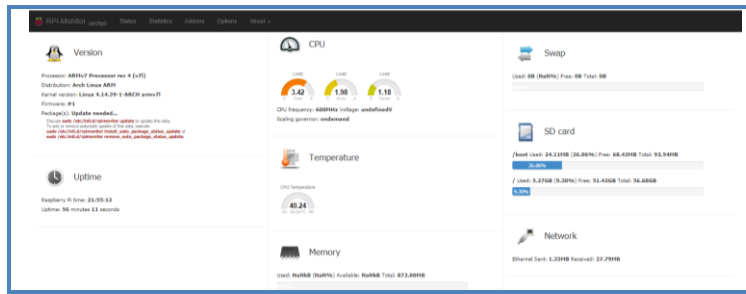


Figura 21. Captura de pantalla con RPI-Monitor funcionando en Arch Linux de 32 bits

De nuevo, mostramos los datos y variables del entorno obtenidos en un vistazo de *RPI-Monitor* que se exponen en la siguiente tabla.

Datos y variables del entorno	
Tiempo de arranque	25 segundos
Tiempo de apagado	15 segundos
Memoria RAM usada/total	161 / 935 MB (17,20%)
Espacio de disco usado/total	3,44 GB de 56,1 GB (5,55%)
Tamaño SWAP usado/disponible	No soportada para discos Micro SD
Temperatura CPU Media	37 ° C
Gestión de energía (KW media consumida en 1 hora)	Potencia consumida 3,5 W/hora
Compatibilidad con APIs de video y 3D	EGL,OpenGL ES 3.0, OpenVG

Tabla 33. Tabla de datos y variables del entorno en ArchLinux ARM de 32 bits

Una vez más se habilita el módulo del kernel para que funcione la aceleración por hardware 3D y además como hemos hecho en otras ocasiones, aumentaremos la memoria de la *GPU* a 128 MBs (*gpu_mem=128*).

Un último problema que quería comentar, antes de mostrar los resultados de las pruebas de rendimiento obtenidas en la siguiente tabla. A la hora de ponernos manos a la obra con las pruebas de rendimiento, también hemos tenido ciertos problemas para instalar *Sysbench*, se ha tenido que instalar los paquetes *autoconf* y *autogen* con *yaourt* para poder instalar finalmente con *packer Sysbench*. El resto de aplicaciones no ha dado problemas al instalarse con *pacman*.

Puntuaciones obtenidas en las Pruebas de rendimiento	
Tiempo total Sysbench memoria RAM	1402,92 eventos/segundo (10240 eventos / 7,26 segs)
Tiempo total Sysbench IO	305,3631 segundos
Tiempo total Sysbench 1 core	21,39 eventos/segundo (10 segs)
Tiempo total Sysbench 2 cores	28,95 eventos/segundo (10 segs)
Tiempo total Sysbench 4 cores	71,22 eventos/segundo (10 segs)

Temperatura CPU Max obtenida durante las pruebas(1 core / 2 cores / 4 cores)	46,8 °C / 49,8 °C / 52,7 °C
GLMark2	114 puntos (43,7 ° C)
Hdparm (buffer cache)	591,16 MB/segundo
Hdparm (buffer disk)	21,95 MB/segundo
Dataset Definition (DD) escritura	2 GB/ 148,383 seg (13,8 MB/seg)

Tabla 34. Tabla de datos obtenidos en las pruebas de rendimiento en ArchLinux ARM de 32 bits

3.5.2. Datos, variables y pruebas de rendimiento con parámetros de configuración de hardware modificados

Una vez cambiados los parámetros de configuración de hardware, retomamos exponer la puntuación otorgada a la experiencia de usuario reflejada en la siguiente tabla.

Puntuaciones de experiencia de usuario	
Fluidez	7,5
Estabilidad	9

Tabla 35. Tabla de datos con las puntuaciones de experiencia de usuario en Ubuntu Mate overclocked

Proseguimos con los datos y variables del entorno en el que algo se ha producido mejora.

Datos y variables del entorno	
Tiempo de arranque	24 segundos
Tiempo de apagado	13 segundos
Temperatura CPU Media	37,7 ° C
Gestión de energía (KWs consumidos en 1 hora)	Potencia consumida 3,8 W/hora

Tabla 36. Tabla de datos y variables del entorno en ArchLinux ARM de 32 bits overclocked

Un último problema que quería comentar, antes de mostrar los resultados de las pruebas de rendimiento obtenidas en la siguiente tabla. A la hora de ponernos a realizar todas las pruebas de rendimiento, también hemos tenido ciertos problemas para descargar *Sysbench*, se ha tenido que instalar los paquetes *autoconf* y *autogen* con *yaourt* para poder instalar finalmente con la herramienta de gestión de paquetes *pacman* la aplicación *Sysbench*. El resto de aplicaciones no ha dado problemas al instalarse con *pacman*.

Puntuaciones obtenidas en las Pruebas de rendimiento	
Tiempo total Sysbench memoria RAM	955,2 event./seg. (9970 event./10 segs)
Tiempo total Sysbench IO	315,5832 segundos
Tiempo total Sysbench 1 core	24,35 eventos/segundo (10 segs)
Tiempo total Sysbench 2 cores	25,64 eventos/segundo (10 segs)
Tiempo total Sysbench 4 cores	43,78 eventos/segundo (10 segs)

Temperatura CPU Max obtenida durante las pruebas(1 core / 2 cores / 4 cores)	47 °C / 50,3 °C / 53,4 °C
GLMark2	116 puntos (44,9 ° C)
Hdparm (buffer cache)	725,89 MB/segundo
Hdparm (buffer disk)	22,05 MB/segundo
Dataset Definition (DD) escritura	2 GB/ 138,398 seg (14,8 MB/seg)

Tabla 37. Tabla de datos obtenidos en las pruebas de rendimiento en ArchLinux ARM de 32 bits overclocked

Como hemos estado observando en el resto de pruebas, en las realizadas con *overclocking* en éste sistema operativo han mejorado los resultados como cabía esperar, con un aumento no muy significativo del rendimiento general, pero sin llegar a ser escandalosa y en el que una vez más se produce la pérdida de potencia por protección ante el sobre calentamiento en las pruebas de procesador con 4 núcleos.

Para finalizar, las impresiones últimas tras el uso de este sistema operativo han sido del todo positivas y las más satisfactorias y gratificantes de los por ahora estudiados, también debido a que ha sido de los sistemas que más tiempo de desarrollo y trabajo me han llevado. Muchas de las tareas ya las hemos comentado anteriormente, ya que partimos de un sistema base con compatibilidad completa de todos los recursos de la placa, y a partir de ahí nos ha tocado personalizar y construir el sistema que deseamos, instalando el escritorio, aplicaciones, gestores de descarga de paquetes de repositorios personales, de los cuales tras descargarlos y no funcionar, los hemos modificado y añadiéndoles las variables necesarias para posibilitar la compatibilidad con el procesador, compilarlos y ejecutarlos y si no terminan de funcionar del todo, modificarlos a posteriori.

Toda esta labor hace que se sienta que el sistema operativo es parte de uno mismo, la filosofía de hacer las cosas por uno mismo, da como resultado un sistema con una usabilidad y rendimiento excelente, unido a una gran estabilidad y consistencia. Lo que más siento, es no disponer de tiempo ilimitado en el desarrollo de esta memoria para haberlo pulido más, quedando pendiente como una tarea que aplazaremos para realizar a posteriori por gusto personal.

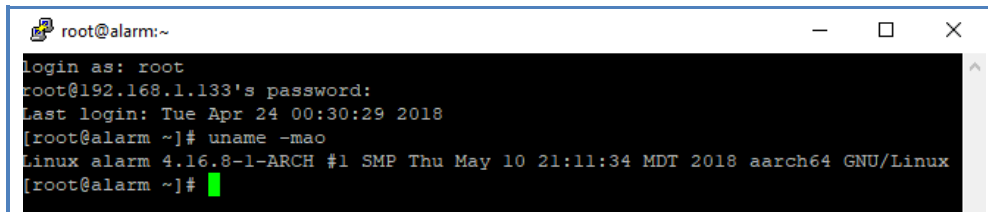
3.6. Análisis y pruebas del SO Sistema Operativo ArchLinux ARM de 64 bits

Para comenzar, la estrategia que emplearemos para el análisis de este sistema operativo es la de repetir todo el trabajo realizado en ésta versión tal y como lo hicimos con la versión previa *Arch Linux de 32 bits*, esperando obtener las mismas garantías de éxito alcanzadas en dicho sistema.

Decir que puesto que ha sido de los sistemas operativos que más trabajo y tiempo ha costado realizar las tareas, trasladaremos todos los apuntes y experiencias realizadas. Comentaremos a continuación como nos ha ido.

El sistema arranca con igual celeridad como ocurría con la versión previa, y en principio parece que hay compatibilidad con la mayoría de drivers de la placa.

Si sacamos la información del sistema podemos comprobar que nos encontramos con un núcleo adaptado al hardware del procesador de 64 bits.

A terminal window titled 'root@alarm:~' showing the output of the 'uname -mao' command. The output indicates the system is running Linux alarm 4.16.8-1-ARCH #1 SMP Thu May 10 21:11:34 MDT 2018 aarch64 GNU/Linux.

```
root@alarm:~  
login as: root  
root@192.168.1.133's password:  
Last login: Tue Apr 24 00:30:29 2018  
[root@alarm ~]# uname -mao  
Linux alarm 4.16.8-1-ARCH #1 SMP Thu May 10 21:11:34 MDT 2018 aarch64 GNU/Linux  
[root@alarm ~]#
```

Figura 22. Captura con la ejecución del comando `uname -mao` para mostrar información del SO de 64 bits

Para comenzar no levantan las interfaces de red de forma automática, nos toca hacerlo a mano con el comando `dhcpcd eth0`, actualizaremos `pacman` y al intentar instalar el escritorio `LXDE` con `SDDM` tal como hicimos con la versión previa, no funciona correctamente para esta versión de sistema operativo, con lo que probaremos una alternativa el sistema de escritorio `XFCE`.

```
pacman -S xfce4 xfce4-goodies xarchiver pavucontrol
```

Tras arrancar `xfce`, se muestra impecable, ligero, minimalista, con un diseño atractivo y con gran cantidad de opciones, con una barra de accesos directos, pero todo ésto resulta una “*mera ilusión*”. El sistema de escritorio se cuelga cada dos por tres, se hace irrecuperable, el sistema `Linux` aguanta por detrás, de hecho se puede cambiar de terminal de acceso con `Ctrl + Alt + F2`, por ejemplo, y seguir trabajando sobre línea de comandos, pero para volver al escritorio hay que reiniciar a la fuerza.

También decir que no ha sido posible instalar otras aplicaciones que funcionan correctamente en la versión de 32 bits, como lo son `yaourt`, `glmark2`, `sysbench`, `rpimonitor`, etc. Con lo que no será posible realizar algunas de las pruebas de rendimiento,

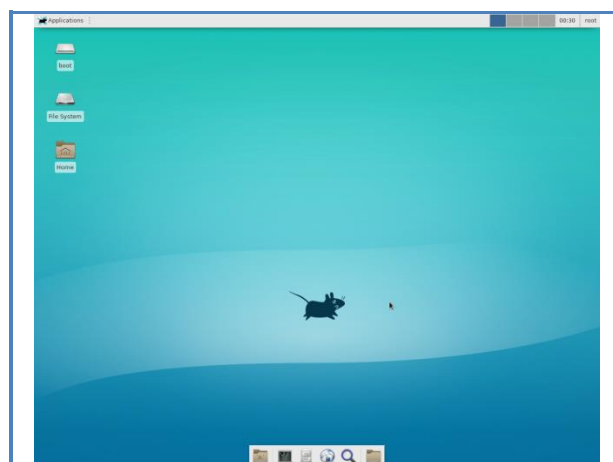


Figura 23. Captura del escritorio XFCE4 funcionando en Arch Linux de 64 bits

Se ha intentado descargar de repositorios de usuarios `git` y `de AUR`, y no ha habido manera, los problemas de compatibilidad con este sistema son más que obvios.

No existen apenas repositorios con paquetes actualizados al kernel con compatibilidad para procesadores *ARM* en *aarch64* y esto hace que demos por desestimado el análisis de este sistema operativo y no seguir perdiendo tiempo en él, por ahora.

A modo de conclusión, hemos podido ver y comprobar que aunque el desarrollo de este sistema está bastante avanzado, ciertas tareas las realiza con gran soltura, pero nos ha sido imposible cuantificar el rendimiento y además no cumple con la mínima estabilidad necesaria para poder trabajar más con él, puesto aún le queda mucho desarrollo por delante. Quizás dentro de un tiempo le daremos otra oportunidad.

3.7. Análisis y pruebas del SO Sistema Operativo Fedora 27 ARM de 32 bits

Hemos llegado al último de los SSOO a analizar, del cual poseo grandes expectativas. Tras la experiencia pasada con *OpenSuse Tumbleweed*, he optado por evitar descargar la versión *Workstation* con *GNOME* como escritorio de trabajo y bajar la versión con el escritorio ligero *LXQt*, para que el desarrollo de la práctica sea lo más ágil y estable posible. El hecho que me haya decantado y seleccionado este escritorio es para emplear y funcionar con otro tipo de escritorio que aún no habíamos probado en las otras distribuciones.

Tras descargar y descomprimir el fichero de imagen en formato RAW, volvemos a instalarlo sobre nuestra tarjeta con el comando DD. Tras arrancar el sistema, nos muestra un asistente de inicio de configuración muy parecido al visto en *Ubuntu MATE*, en el que nos permite cambiar los parámetros de localización (zona horaria, fecha y hora), sistema (nombre de equipo y configuración de la red), establecer la contraseña del usuario *root* y añadir nuevos usuarios.

Añadir a lo expuesto anteriormente que en el asistente de configuración de red no permite conectarse mediante conexión inalámbrica, una vez más hemos optado por conectar el cable Ethernet y ha detectado la configuración de red por DHCP al instante, posteriormente si disponemos de tiempo, comprobaremos si hay compatibilidad con el driver de red inalámbrica y configuraremos su acceso.

Tras realizar los cambios reinicia el sistema y nos permite acceder al escritorio. En un principio la experiencia es bastante fluida, *LXQt* tiene un escritorio muy sencillo parecido a los vistos en los sistemas Windows, con una barra de tareas con dos escritorios, que nos permite añadir accesos directos que deseemos y al final de la barra los típicos iconos con información de hora, fecha, dispositivos, etc. El menú de inicio incluye la siguiente lista de aplicaciones.

Aplicaciones preinstaladas	
Accesorios	Ark, FeatherPad, KCalc, PCManFM File Manager, Qlipper
Administración	Date and Time, dnfdragora, dndragora-updater, Firewall, Users and Groups
Gráficos	LXImage, Screenshot

Internet	Qtransmission Bitorrent Client, Quassel IRC, QupZilla
Office	Qpdview (Qt5)
Sonido y Video	Dragon Player, PulseAudio Volume Control (Qt), Yarock.
Herramientas del sistema	Fedora Release Notes, Gparted, QTerminal, Qterminal drop down, Sensor Viewer.
Preferencias	LXQt settings→LXQt Configuration Center, Desktop, Openbox Settings, Appearance, Customize Look and Feel, Desktop Notifications, File Associations, Keyboard and Mouse, Locale, Monitor Settings, Network Connections, Power Management, Screensaver, Session Settings, Shortcut Keys.
About LXQt	
Salir	Hibernar, Deslogear, Reiniciar, Apagar, Suspende.
Bloquear pantalla	
Búsqueda...	

Tabla 38. Tabla con la lista de aplicaciones preinstaladas en Fedora 27 ARM

Tras comprobar cómo se mueve el escritorio, comprobar las aplicaciones instaladas y que tal intercambia tareas y demás procedemos a instalar las aplicaciones de pruebas de rendimiento. Se han podido instalar todas, a excepción del monitor de Raspberry Pi, he intentado buscar repositorios no oficiales, o alguna forma de compilar e instalarlo, pero no hay nada para *Redhat/Fedora*, con lo que tendremos que emplear alternativas para poder monitorizar las variables que precisamos para el estudio.

Una de las herramientas interesantes que trae Fedora 27 es el *Software Management* – *dnfdragora*, un gestor de descargas con *dnf* al más puro estilo de *YaST* de *Suse*. Además de gestionar automáticamente las actualizaciones, te permite realizar las búsquedas e instalaciones de software de forma muy sencilla, y con una interfaz gráfica intuitiva.

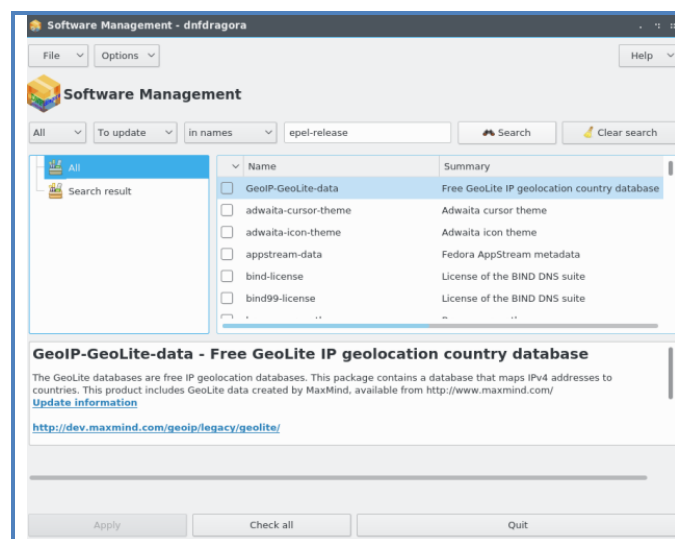


Figura 24. Captura de aplicación de administración de software dnfdragora para Fedora 27 ARM

Para poder observar gráficamente el tamaño de las particiones emplearemos *GParted*, el cual es un conocido y fantástico software de gestión de particiones que tras instalarse, podemos ver que el sistema no ha redimensionado el tamaño de partición de la tarjeta para la partición de sistema operativo, por lo que aprovecharemos de la instalación de éste para realizar dicha acción.

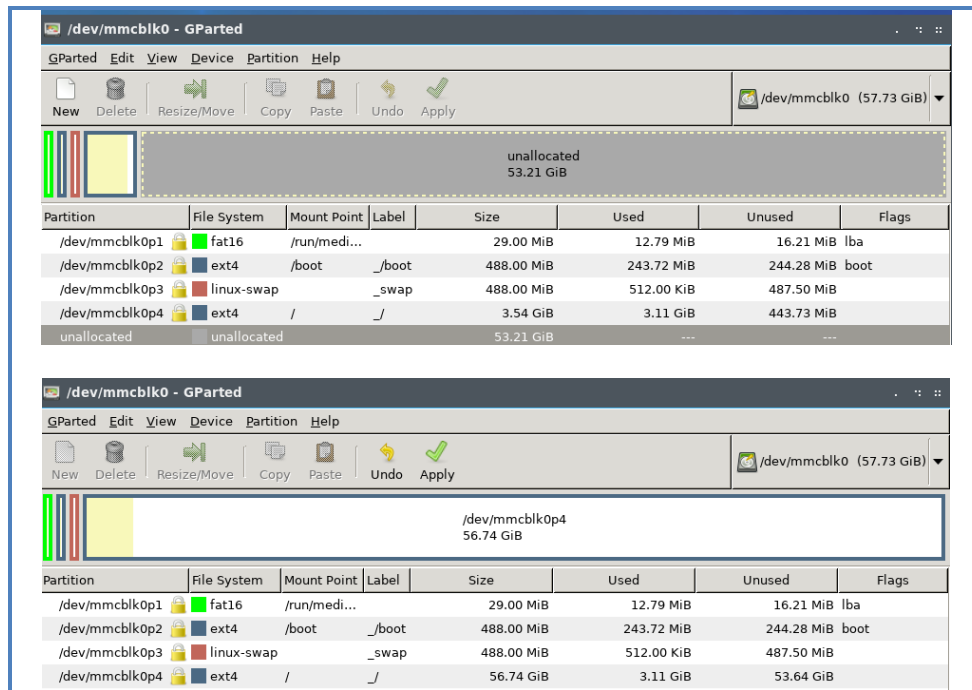


Figura 25. Extensión de la partición de la tarjeta con GParted en el SO Fedora 27 ARM

Emplearemos el comando `yum search sensors` y `yum search cpufreq` para que nos busque qué aplicaciones existen en el repositorio de *Fedora 27* compatibles con monitorización de temperatura y nos muestre la frecuencia e información del procesador. Emplearemos el `xfce4-sensors-plugin.armv7hl` para la monitorización de temperatura y `xfce4-cpufreq-plugin.armv7hl` para las características del procesador. La memoria RAM usada y disponible con el comando `free -m`. Y finalmente para obtener la información del sistema emplearemos el comando `uname -mao` para que nos muestre dicha información. Podemos ver en las siguientes capturas como monitorizaremos dichas variables.

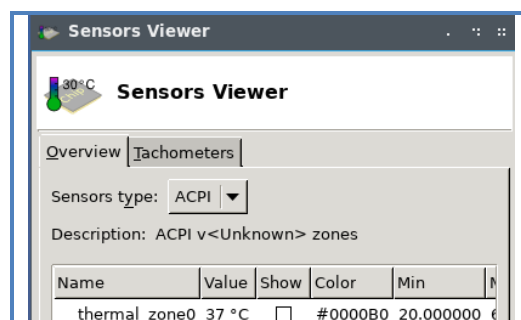


Figura 26. Captura con la ejecución del comando xxx para la monitorización de la frecuencia de CPU

```
[root@FEDORA27 borraz]# free -m
              total        used         free       shared    buff/cache   available
Mem:           974         255          182           14           536         650
Swap:          487           0           487
```

Figura 27. Captura con la ejecución del comando `free -m` para la monitorización de memoria RAM

```
[root@FEDORA27 borraz]# uname -m
Linux FEDORA27 4.13.9-300.fc27.armv7hl #1 SMP Mon Oct 23 15:0
2:20 UTC 2017 armv7l armv7l armv7l GNU/Linux
[root@FEDORA27 borraz]#
```

Figura 28. Captura con la ejecución del comando `uname -m` para mostrar información del SO Arch Linux

3.7.1. Datos, variables y pruebas de rendimiento con parámetros de configuración de hardware sin modificar

Seguimos a continuación con el análisis personal de la experiencia de uso del sistema *Fedora 27 ARM*, el cual exponemos en su ya típica tabla de resultados.

Puntuaciones de experiencia de usuario	
Facilidad de instalación	9
Fluidez	7
Estabilidad	9
Compatibilidad con el hardware	95%

Tabla 39. Tabla de datos con las puntuaciones de experiencia de usuario en *Fedora 27 ARM*

Es una pena que no exista una versión de *RPI-Monitor* para *Fedora*, nos facilita mucho la labor de monitorización, pero por lo menos se ha podido suplir con los comandos y herramientas antes citadas, con lo que mostramos a continuación los datos obtenidos en la siguiente tabla.

Datos y variables del entorno	
Tiempo de arranque	1 minuto y 59 segundos
Tiempo de apagado	29 segundos
Memoria RAM usada/total	255 / 974 MB (26,18%)
Espacio de disco usado/total	3,11 GB de 56,74 GB (5,48 %)
Tamaño SWAP usado/disponible	0,5 / 487 MB (100%)
Temperatura CPU Media	40 °C
Gestión de energía (KW consumida en 1 hora)	Potencia consumida 3,5 W/hora
Compatibilidad con APIs de video y 3D	EGL, OpenGL ES 3.0, OpenVG, OpenMAX

Tabla 40. Tabla de datos y variables del entorno en *Fedora 27 ARM*

Por defecto *Fedora 27* viene con la aceleración por hardware activada, lo cual se agradece, de hecho se puede comprobar que agiliza la navegación web y la reproducción de video y a la hora de realizar las pruebas con *Gmark2* no ha sido preciso hacer el cambio en el fichero *config.txt*. A continuación podemos ver los resultados:

Puntuaciones obtenidas en las Pruebas de rendimiento	
Tiempo total Sysbench memoria RAM	717,43 eventos/segundo (10 segs)
Tiempo total Sysbench IO	310,0814 segundos

Tiempo total Sysbench 1 core	10,70 eventos/segundo (10 segs)
Tiempo total Sysbench 2 cores	21,39 eventos/segundo (10 segs)
Tiempo total Sysbench 4 cores	41,78 eventos/segundo (10 segs)
Temperatura CPU Max obtenida durante las pruebas(1 core / 2 cores / 4 cores)	41,5 °C / 45,3 °C / 52,3 °
GLMark2	120 puntos (43,2 °C)
Hdparm (buffer cache)	378,42 MB/segundo
Hdparm (buffer disk)	21,80 MB/segundo
Dataset Definition (DD) escritura	2 GB/ 140,586 seg (14,6 MB/seg)

Tabla 41. Tabla de datos obtenidos en las pruebas de rendimiento en Fedora 27 ARM

3.7.2. Datos, variables y pruebas de rendimiento con parámetros de configuración de hardware modificados

Tras realizar los cambios de parámetros de rendimiento que van a ser la constante para el resto de pruebas sobre el fichero *config.txt*, comentamos de nuevo como ha sido la experiencia de usuario reflejado en la siguiente tabla.

Puntuaciones de experiencia de usuario	
Fluidez	7,5
Estabilidad	9

Tabla 42. Tabla de datos con las puntuaciones de experiencia de usuario en Fedora 27 ARM overclocked

Estas son las primeros datos de las variables, en las que como se puede observar han variado notoriamente debido a las modificaciones de configuración realizadas.

Datos y variables del entorno	
Tiempo de arranque	1 minuto 43 segundos
Tiempo de apagado	26 segundos
Temperatura CPU Media	41 ° C
Gestión de energía (KW consumidos en 1 hora)	Potencia consumida 4 W/hora

Tabla 43. Tabla de datos y variables del entorno en Fedora 27 ARM overclocked

Y para concluir con las pruebas de rendimiento realizadas, éstos son los datos obtenidos:

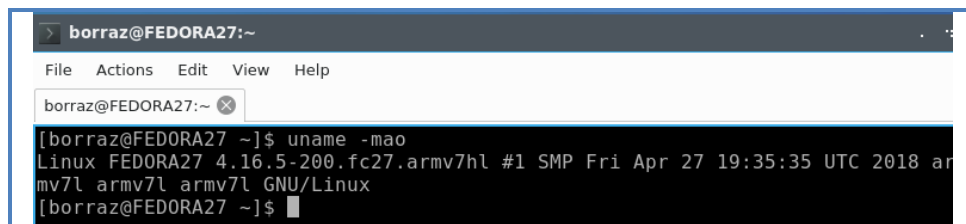
Puntuaciones obtenidas en las Pruebas de rendimiento	
Tiempo total Sysbench memoria RAM	714,53 eventos/segundo (10 segs)
Tiempo total Sysbench IO	304,0245 segundos
Tiempo total Sysbench 1 core	10,03 eventos/segundo (10 segs)
Tiempo total Sysbench 2 cores	20,30 eventos/segundo (10 segs)
Tiempo total Sysbench 4 cores	37,43 eventos/segundo (10 segs)

Temperatura CPU Max obtenida durante las pruebas(1 core / 2 cores / 4 cores)	41,7 °C / 42,4 °C / 42,9 °
GLMark2	129 puntos (44,2 °C)
Hdparm (buffer cache)	405,17 MB/segundo
Hdparm (buffer disk)	21,86 MB/segundo
Dataset Definition (DD) escritura	2 GB/ 149,932 seg (13,7 MB/seg)

Tabla 44. Tabla de datos obtenidos en las pruebas de rendimiento en Fedora 27 ARM overclocked

Comentar que es evidente que la mejora se ha producido en la mayoría de los test como ha sucedido en otras ocasiones, con pautas de comportamiento parecidas a las vistas.

La experiencia final con *Fedora 27* ha sido más que grata, se nota la consistencia del sistema operativo con experiencia de muchos años detrás por la gente de *Red Hat*. Aunque no es perfecta del todo, posee una usabilidad bastante buena, te permite realizar la mayoría de tareas de escritorio con una soltura considerable, y se han instalado 1225 actualizaciones y el sistema ha seguido funcionando sin problema alguno. De hecho, se ha producido una actualización del núcleo de *Linux*, encontrándonos ahora en la versión del *Kernel 4.16.5-200.fc27.armv7hl*.



```

borraz@FEDORA27:~
File Actions Edit View Help
borraz@FEDORA27:~
[borraz@FEDORA27 ~]$ uname -mao
Linux FEDORA27 4.16.5-200.fc27.armv7hl #1 SMP Fri Apr 27 19:35:35 UTC 2018 armv7l armv7l armv7l GNU/Linux
[borraz@FEDORA27 ~]$

```

Figura 29. Captura con la ejecución del comando `uname -mao` para mostrar información del SO Fedora 27

Recapitulando, ya hemos obtenido todos los datos necesarios para poder enfrentarnos a la comparativa en el siguiente apartado. Este capítulo ha resultado mucho más extenso y ha llevado bastante más trabajo del que se había planificado en un principio, pero pese a encontrarnos con varios tipos de problemas, se han solventado la mayoría sin mayor inconveniente.

4. Comparativa, Reflexión y Análisis de los resultados

Nos encontramos en la fase que clausura el ciclo de investigación-acción y dará paso a la elaboración del informe final y quizás, de ser necesario, un replanteamiento del problema, y de encontrarnos en este paso, reiniciar el ciclo de esta espiral auto-reflexiva.

Hemos alcanzado uno de los puntos más importantes del proceso de investigación, ya que tendremos que centrarnos en qué hacer y cómo tratar los datos, como vamos a interpretar la información recabada e imaginar de qué manera podemos interpretarlos. Este proceso nos permitirá enriquecer la práctica y elaborar propuestas de futuro y medidas paliativas para una propuesta de mejora en caso de ser necesaria.

4.1. Comparativa de los resultados obtenidos en la fase de pruebas

Tras el largo trabajo de análisis desarrollado de todos los sistemas operativos, hemos extraído información más que suficiente y experiencias de uso personales para poder cotejar y confrontar los datos conseguidos y poder sacar conclusiones. Sin más dilación pasamos al estudio comparativo.

Para comenzar desestimaremos las pruebas para las versiones de sistema operativo de 64 bits, las cuales no están aún a la altura en cuanto a compatibilidad, estabilidad y recursos disponibles comparándolos con los sistemas de arquitectura inferior, con lo que nos quedan cuatro sistemas de todos los estudiados. También desestimaremos algunas de las variables datadas, debido a que no han aportado información relevante o no han mostrado diferencias significativas para poder comparar unas con otras, lo haremos en cada uno de los siguientes sub-apartados.

4.1.1. Comparativa de datos aplicaciones preinstaladas de los SSOO

Todos incorporan, a excepción de *Arch Linux* una suite de software de escritorio suficiente, y en los casos de *Ubuntu Mate* y *Fedora 27* es excelente, comparando con la que incorpora *Raspbian*, pero por el contrario se puede observar que afecta negativamente en los tiempos de arranque y consumo de recursos de RAM, CPU y disco, pero no por ello afectan a la fluidez del sistema.

Personalmente, **en este aspecto el gran ganador es *Ubuntu Mate***. Pese a que la inclusión de software siempre es un factor que puede mejorar, como es normal agregándolo *a posteriori*, tanto la organización y el conjunto de software es excelente, y las aplicaciones se abren en un tiempo bastante aceptable, permitiendo el intercambio entre algunas de las tareas con una suavidad bastante cercanas a las que podemos tener en un ordenador de escritorio. Añadir que los menús de lugares y configuraciones del sistema, también han sido un punto a destacar, teniendo el acceso a ubicaciones típicas al alcance de un clic, mejorando más aún si cabe la experiencia de usuario.

Para acabar, también decir que el escritorio de *Mate* es posible que como ocurre en los equipos de escritorios, pueda instalarse y usarse en otras versiones de sistema operativo, como en *Raspbian* u *OpenSuse*, con lo que tomamos nota para futuras pruebas a realizar.

4.1.2. Comparativa de datos de experiencia de usuario de los SSOO

Antes de comparar los datos de la experiencia de usuario, desestimaremos varias de las variables como hemos comentado anteriormente, como por ejemplo la **fluidez** y la **estabilidad**, las cuales se han mostrado dentro de un espectro bastante regular, con una fluidez notable y con una estabilidad muy alta (puntuación de 9 sobre 10). Y la fluidez mejora realizando los cambios de frecuencia, pero no de forma significativa, manteniendo la estabilidad del sistema intacta.

Respecto a la compatibilidad con el hardware, se puede decir que es del 100% inicialmente en los sistemas *Debian* y en *Arch Linux*. En *Fedora 27* inicialmente no funciona la conexión *Wifi*, pero se soluciona tras actualizar el sistema.

Solamente nos queda comparar en cuanto a la experiencia de usuario la Facilidad de instalación del sistema operativo, las cuales en los sistemas *Debian* y *Fedora* han sido muy sencillas y cercanas para que se puedan desenvolver cualquier usuario de escritorio sin experiencia y conocimientos informáticos avanzados, podemos ver la comparación en detalle en el siguiente gráfico.

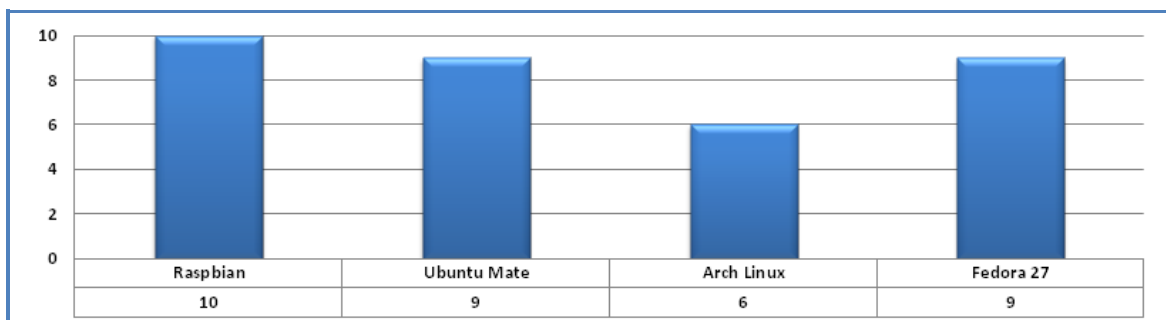


Figura 30. Gráfico comparativo sobre la facilidad de instalación de los Sistemas Operativos estudiados

En el que hay un claro vencedor en este aspecto, *Raspbian*. Con su característica “*Out Of The Box*”, hace que el sistema arranque velozmente, y presente el escritorio. Aunque por el contra, los asistentes de inicio que presentan *Ubuntu Mate* y *Fedora 27* permiten ciertos detalles que son imprescindibles a *priori*, como es la conectividad, idioma, zona horaria, etc. antes de comenzar con el sistema y lo que no supone problema para un usuario inexperto.

Resumiendo y en pocas palabras, en este **apartado los cuatro sistemas se mueven con gran soltura y son altamente estables.**

4.1.3. Comparativa de datos y variables del entorno de los SSOO

Volviendo a las desestimaciones de variables, tanto la **temperatura de CPU Media**, como la **gestión de energía**, se mantienen de forma casi constante, con una temperatura de unos 38° a 39° de media y 4 vatios por hora respectivamente. Y en los que tras realizar los *tweaks* de rendimiento, apenas ha aumentado un par de grados de temperatura y medio vatio. Lo mismo ocurre con la compatibilidad con lo que respecta al **hardware acelerado (APIs de video y 3D)**, en el que todos los sistemas operativos a excepción de los excluidos son compatibles con esta especificación de software sobre el hardware estudiado.

Respecto a los **tiempos de arranque** los gráficos mostrados más abajo son más que concluyentes, aquí el sistema que más rápido inicia es *Raspbian* con diferencia. En cuanto al apagado se puede ver que este caso el ganador es *Arch Linux*, aunque hay que destacar que todos se apagan con cierta rapidez.

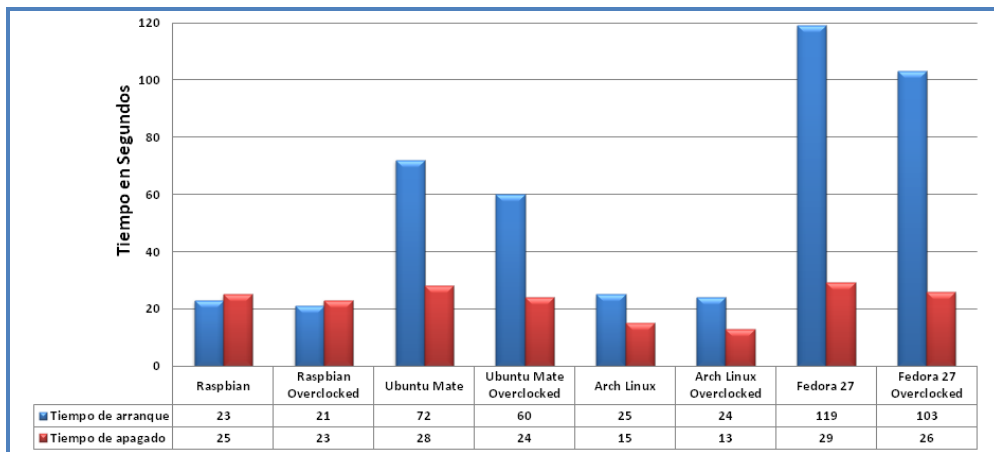


Figura 31. Gráfico comparativo de tiempos de arranque y apagado de los Sistemas Operativos estudiados

Y en cuanto al sistema que consume más **memoria RAM** en el inicio, también es *Raspbian* y por el contrario encontramos a *Arch Linux*, lo cual es normal puesto que partimos de una base excesivamente simple. En cuanto a espacio en disco, también es *Raspbian* el que **menos espacio ocupa inicialmente**, con lo que la combinación de consumo de RAM y escaso espacio de disco usado tiene como consecuencia esos tiempos de arranque sean tan sumamente reducidos.

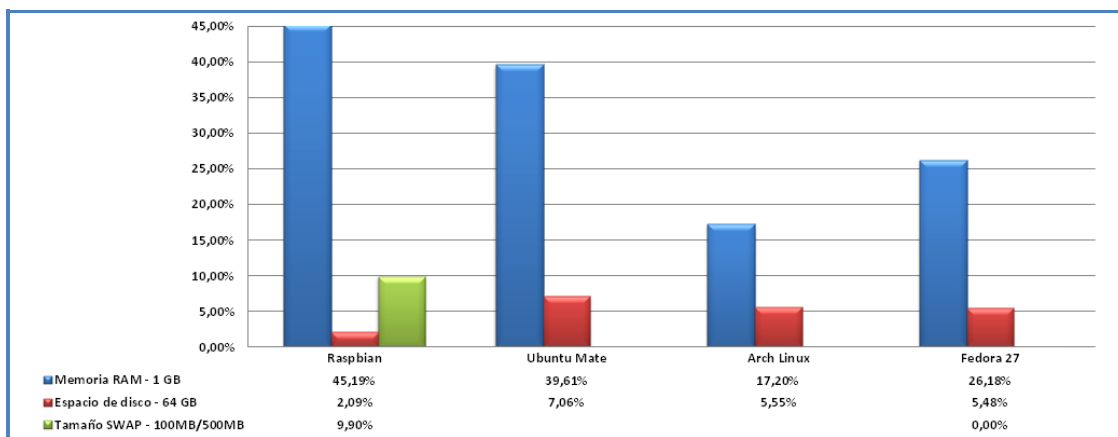


Figura 32. Gráfico comparativo de recursos de memoria RAM, Disco de sistema y SWAP de los SSOO estudiados

4.1.4. Comparativa de pruebas de rendimiento de los SSOO

Finalmente mostraremos los gráficos comparativos de las pruebas de rendimiento realizadas, en las que cabe destacar **la superioridad en las pruebas de rendimiento en los sistemas operativos Debian** respecto a los otros estudiados, seguramente debido a que son sistemas con más tiempo de desarrollo detrás y con un nivel de optimización superior. Además que se traduce también en un aumento de la carga y temperatura del procesador en estos, hecho que también ha quedado constatado con la captura de medidas de las variables de temperatura con *RPI-Monitor*, pero siempre dentro de unos niveles aceptables y seguros y sin vulnerar en ningún momento la garantía de nuestra plaquita. A continuación podemos ver los resultados comparados.

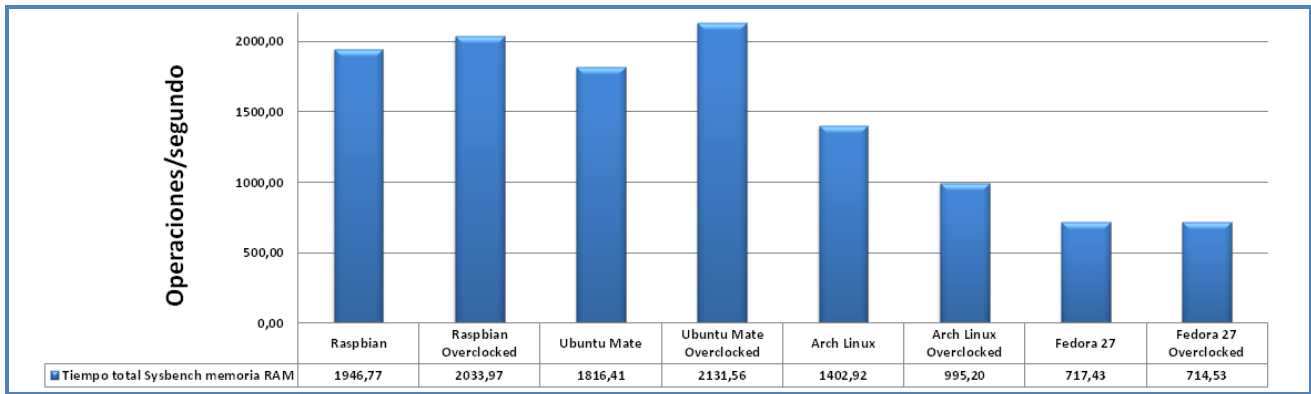


Figura 33. Gráfico comparativo de las pruebas de rendimiento de memoria RAM con Sysbench de los SSOO estudiados

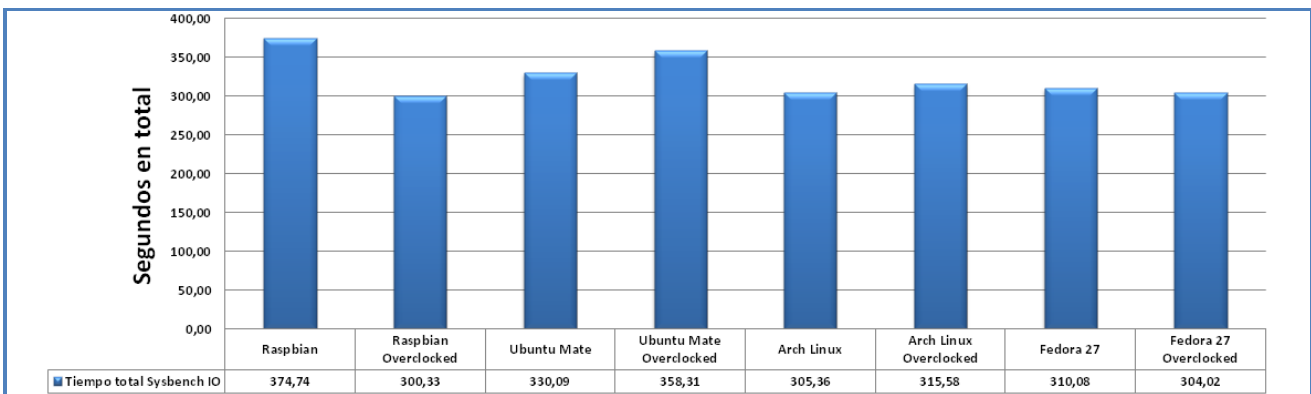


Figura 34. Gráfico comparativo de las pruebas de rendimiento de ficheros con Sysbench de los SSOO estudiados

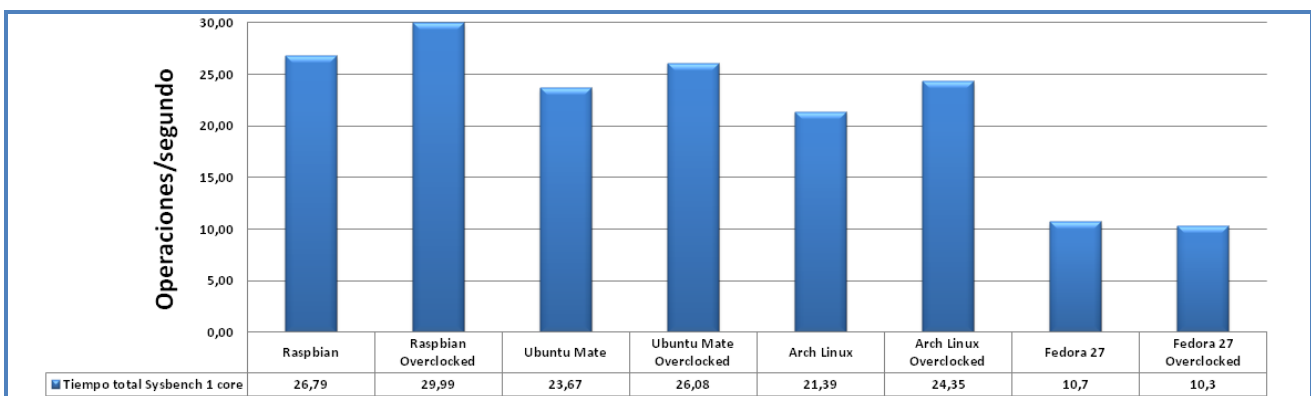


Figura 35. Gráfico comparativo de las pruebas de rendimiento de CPU con 1 core en Sysbench de los SSOO estudiados

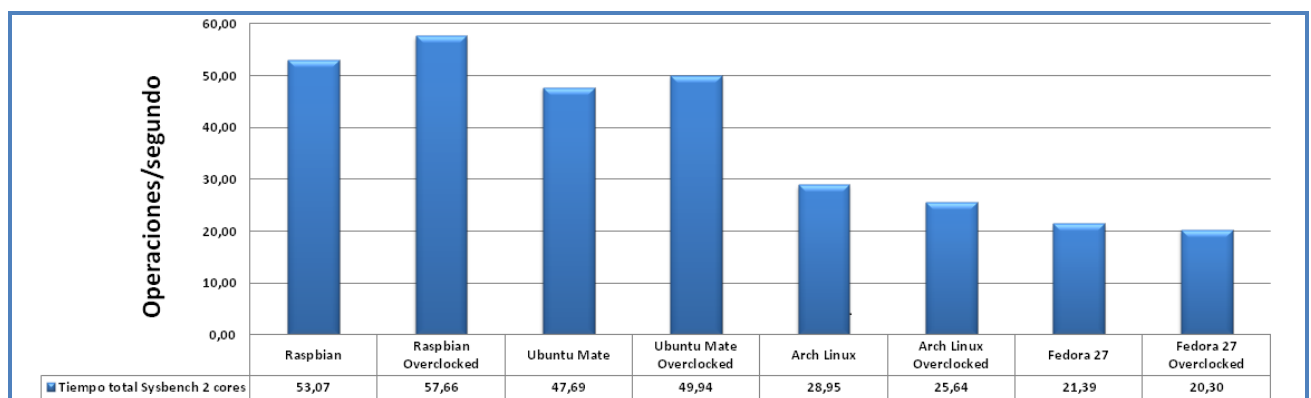


Figura 36. Gráfico comparativo de las pruebas de rendimiento de CPU con 2 cores en Sysbench de los SSOO estudiados

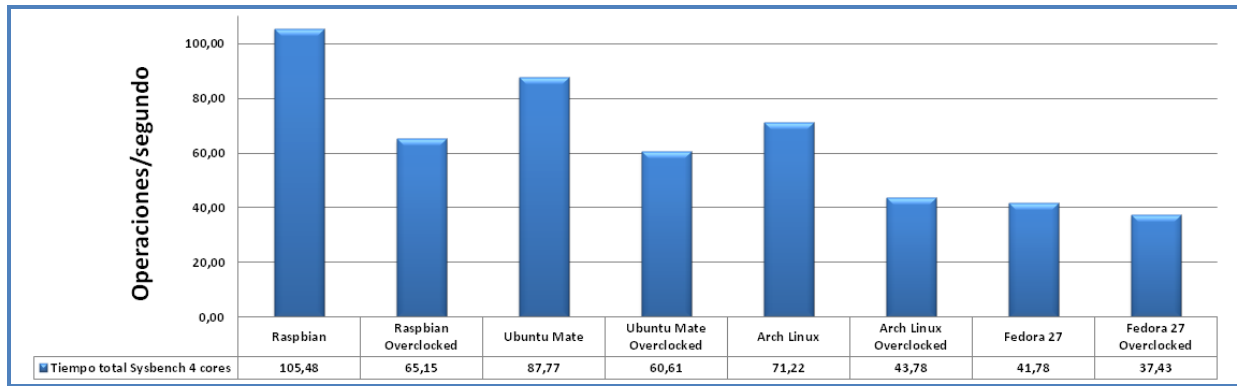


Figura 37. Gráfico comparativo de las pruebas de rendimiento de CPU con 4 cores en Sysbench de los SSOO estudiados

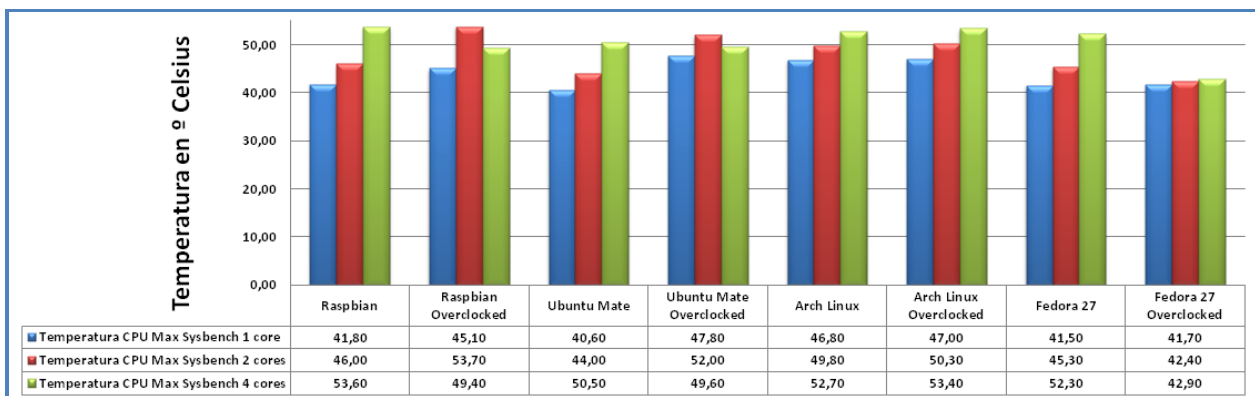


Figura 38. Gráfico comparativo de las temperaturas obtenidas en las pruebas de CPU de los SSOO estudiados

En cuanto a las pruebas de **GLMark2**, no es que destaquen mucho unos de otros, pero sí que **Raspbian es el que mejor rendimiento ha obtenido en las pruebas de gráficos 3D**, sobre todo cuando realizamos las modificaciones para hacer *overclocking*.

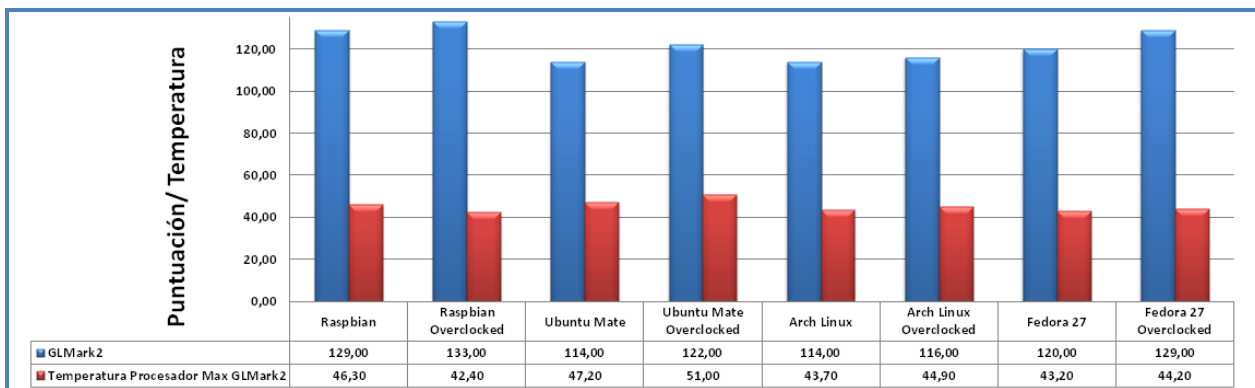


Figura 39. Gráfico comparativo de las puntuaciones y temperaturas obtenidas con GLMark2 de los SSOO estudiados

Y concluyendo con las pruebas de ficheros con **HDparm** y con **Dataset Definition**, son bastante similares, como cabía esperar, pero con **Arch Linux**, las pruebas de lectura con **buffer caché como en buffer disk han sido superiores**, seguido muy de cerca **Raspbian**, pero como hemos comentado con anterioridad esto sea debido a la menor cantidad de recursos en uso.

En cuanto a las **pruebas finales de escritura con Dataset Definition**, son muy similares también (con rangos de escritura que oscilan entre los 11 MBs por segundo a los 15,20 MB) pero

destaca claramente *Ubuntu Mate* con velocidades de más de un MB por segundo respecto a sus competidores.

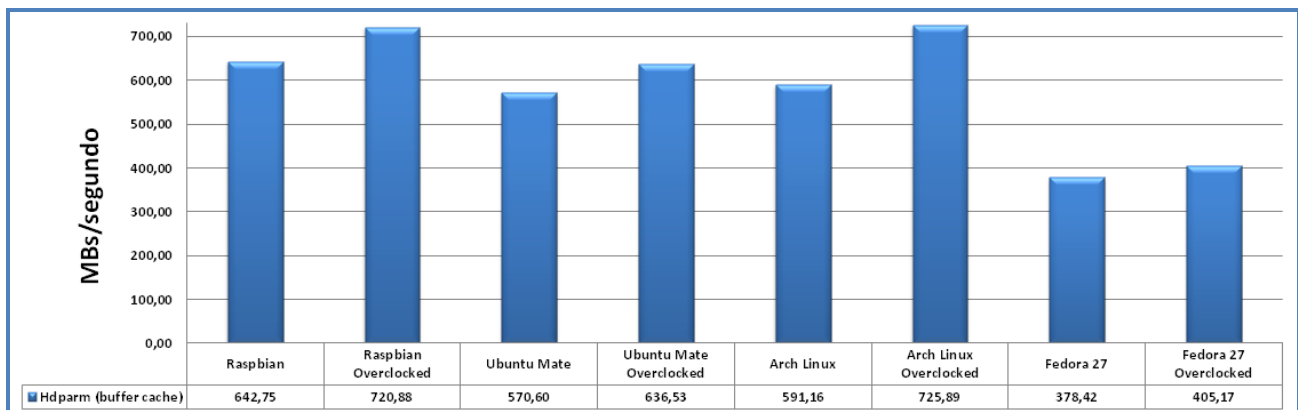


Figura 40. Gráfico comparativo de las pruebas de rendimiento de ficheros con Hdparm de los SSOO estudiados

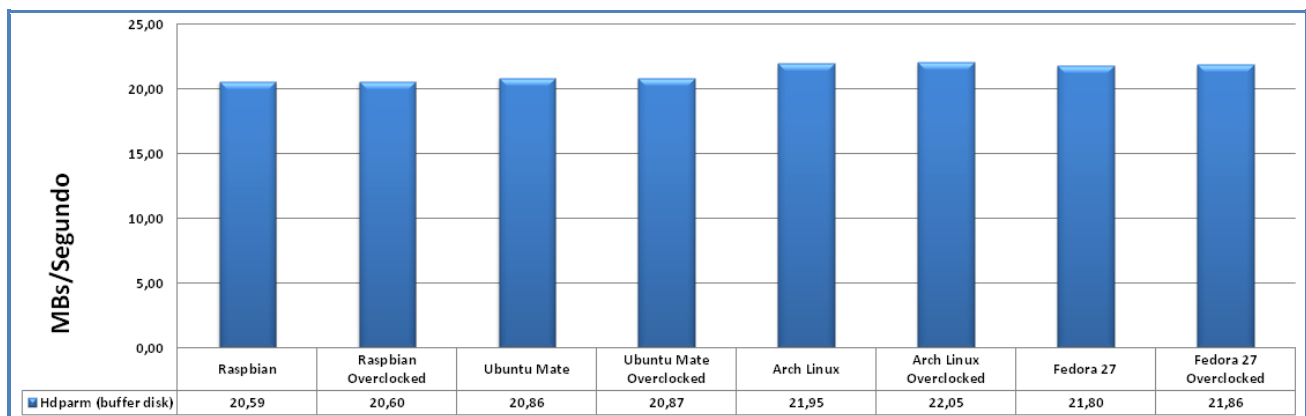


Figura 41. Gráfico comparativo de las pruebas de rendimiento de ficheros con Hdparm de los SSOO estudiados

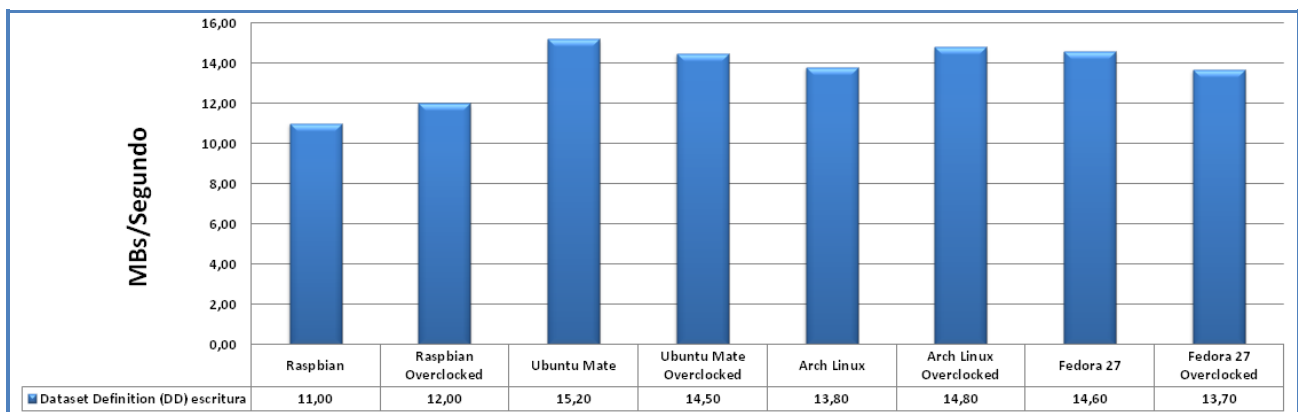


Figura 42. Gráfico comparativo de las pruebas de rendimiento de ficheros con DD de los SSOO estudiados

Ya con todos los datos cotejados y confrontados, pasamos al proceso final de selección.

4.2. Reflexión y selección de candidatos

Hemos alcanzado el punto definido como propósito principal de todo el trabajo realizado en esta memoria, el cual es dilucidar qué sistema operativo se comporta en mejor medida con nuestra *Raspberry Pi 3* ante un sistema Linux de escritorio, y aunque hay que decir que en general casi todos los sistemas operativos obran de una manera más que aceptable, a la vista de los resultados y experiencias transmitidas en esta memoria, ha quedado claro cuál es el que mejor se adapta a las funcionalidades que necesita un usuario de escritorio.

Y el sistema elegido es Ubuntu Mate 16.04 LTS Final Release Xenial, el cual mantiene intactas las propiedades vistas en los sistemas de escritorio basadas en ordenadores personales con arquitecturas *x86* y *x64*, muy similar al aspecto visto en *Gnome2*, sin apenas eliminar elementos ni programas del entorno y conservando la misma esencia y experiencia. Comportándose con un gran rendimiento, además de una fluidez más que aceptable, y con una gran estabilidad, y pese a no ser tan rápido en el arranque como lo es *Raspbian* o *Arch Linux*, tampoco tiene un tiempo excesivamente largo (1 minuto y 12 segundos) y no afecta en absoluto a la navegación fluida en el entorno de escritorio.

Tiene grandes características diferenciadoras respecto a sus otros competidores, aplicando una lógica simplista, muy configurable, con un *Centro de control* unificado que nos facilita en gran medida y de forma visual todas las configuraciones del sistema, una completa suite de aplicaciones e incluyendo desde el principio el navegador *Firefox* que personalmente tanto me gusta por su forma diferente de funcionar y siempre con espíritu libre y grandes posibilidades de expandirlo a bajo nuestros deseos.

Y respecto a la integración con el funcionamiento y sistemas de parametrización con nuestro hardware, también lo bonito y bien escrito que está el fichero de configuración de arranque, que nos posibilita todas las pruebas de mejorar el rendimiento, han sido un punto positivo para este sistema.

Finalmente apostillar que el resto de sistemas tienen sus ventajas que a ciertos tipos de usuarios podrán atraer en mayor medida, con características mejores que las vistas en *Ubuntu Mate*, como que por ejemplo, *Raspbian* es de todos ellos el que posee el núcleo de Linux más actualizado, y que se mueve a gran velocidad respecto al resto, pero carece de la profundidad de los elementos que tiene la versión de su compañero de distribución elegido. Y lo sorprendente que es ver unos sistemas profesionales de escritorio con una base *Red Hat* como lo es *Fedora 27* y como *OpenSuse Tumbleweed de 64 bits* correr en nuestra maquina con gran soltura.

En definitiva, ***Ubuntu Mate*** es un gran sistema operativo y una gran alternativa para todos los usuarios que decidan que quieran dar el salto al software libre y apostar por una opción de bajo coste en una SBC como es la *Raspberry Pi 3 Model B*.

5. Futuras mejoras y conclusiones finales.

5.1. Futuras pruebas y mejoras del proyecto

Las futuras pruebas y mejoras que planteamos sobre el estudio realizado, son principalmente debidas al límite temporal que requiere el transcurso y al largo desarrollo de éste trabajo en sí, así como de la falta de recursos y medios personales (más tarjetas SD, más placas, etc.) para poder realizar muchas más pruebas. Algunas de estas pruebas son las siguientes:

- Más ensayos y análisis de más sistemas operativos y versiones diferentes de escritorio, como por ejemplo con *OpenSuse* de 32 bits, tanto con las versiones *Leap* y *Tumbleweed*, *Kali Linux*, *CentOS*, *Debian*, *Mandriva*, etc. Esto enriquecería tanto la comparativa final como resultado, al tener un número de sistemas mayor.
- Realizar más cambios de configuración en los parámetros de rendimiento del fichero de configuración de arranque, e incluso saltándonos los aquellos que vulneran la garantía para sobrepasar los límites impuestos en cuanto al voltaje y frecuencia con el fin de poder obtener, por ejemplo teóricas mejoras en las pruebas con cuatro núcleos de proceso.
- Emplear sistemas simples sin escritorio, como lo que se hizo con *Arch Linux*, partiendo desde cero y crear uno propio, con más recursos o empleando otras versiones de escritorio e incluso, desarrollar nuestra propia distribución.

De hecho, el análisis de *Arch Linux* ha sido uno de los más enriquecedores de todos los tratados, porque hemos podido crear desde casi cero nuestro propio sistema, añadiendo sistemas de descarga de paquetes alternativas que acceden a repositorios personales de usuarios, en los que hemos modificado y adaptado el código, para que se pudiera compilar y ejecutar en nuestra pequeña Pi. De hecho es el análisis que más curiosidad nos ha hecho despertar en la temática de personalización y desarrollo de los sistemas operativos.

- Emplear otros sistemas de almacenamiento, como por ejemplo un disco duro externo USB. Otros modelos y marcas de tarjetas con mayor categoría para obtener mejoras de lectura y escritura de disco, ya que con los medios y tarjeta empelados, no nos ha sido posible realizar *overclocking* de tarjeta SD.

Y una última prueba que realizaré nada más terminar el curso, que será instalarle las distribuciones *RetroPie*²² o *Recalbox*²³, con la que no solamente disfrutaremos de las grandes posibilidades que se ha demostrado ofrece *Linux*, sino que además disfrutaremos un buen rato de los clásicos *arcade* en nuestra pequeña y maravillosa maquineta.

²² RetroPie. Distribución para *Raspberry Pi* (también adaptada a ordenador personal) basado en *Raspbian* que permite convertir nuestro sistema en una máquina de juegos retro. Emplea *EmulationStation*, *RetroArch* y muchos otros proyectos para simplificar el acceso a los sistemas de emulación con la configuración mínima.

²³ RecalboxOS. Sistema operativo del proyecto *Recalbox*, similar al anterior, para convertir la *Raspberry Pi* en una consola de emulación lista para jugar .

5.2. Conclusiones finales sobre el estudio realizado

En síntesis, mediante el desarrollo de este Trabajo Final de Grado, no solamente hemos analizado y comparado algunos de los mejores sistemas operativos para *Raspberry Pi 3 Model B*, sino que además durante todo el proceso, hemos indagado y ampliado en gran medida muchísimas competencias sobre sistemas operativos basados en *Linux* y hemos adquirido gran cantidad conocimientos trabajando en éstos distintos sistemas, como por ejemplo, emplear las distintas aplicaciones de gestión de actualizaciones (*apt-get*, *pacman*, *yum*, *zypper*, etc.), y en caso de que éstas fallaran, buscar alternativas descargando por nuestra cuenta los ficheros fuentes de repositorios, modificarlos y compilarlos desde nuestra propia máquina con nuestra colección de compiladores y librerías.

También hemos observado una gran variedad de recursos de muy diferentes tipos, las distintas topologías de los diferentes escritorios para *Linux*, poniendo a prueba los límites y las capacidades reales del hardware de la SBC analizada, comprobado cómo se comporta la arquitectura del procesador *Broadcom BCM2837* de 64 bits en sistemas operativos de 32 y 64 bits, y realizando cambios en sus frecuencias de velocidad de *CPU*, *RAM* y *GPU* para lograr mejorar el rendimiento e inspeccionarlo mediante sistemas de monitorización, alarma y control.

Al margen de lo dicho, se ha evidenciado que los sistemas de 64 bits para la Pi 3 están aún “*muy verdes*”, pero van progresando por muy buen camino y probablemente en un tiempo futuro y próximo obtengan la estabilidad y consistencias necesarias para sustituir a los ya anticuados sistemas de 32 bits. Del mismo modo decir que se ha podido constatar mediante las pruebas de rendimiento realizadas de CPU en *OpenSuse* de 64 bits, el más avanzado de los sistemas en este tipo de arquitectura de los analizados, han sido muy superiores a todas las realizadas, signo de que se aprovecha el rendimiento real del procesador.

Derivado de toda esta gran labor realizada, se ha elegido el sistema operativo Ubuntu Mate 16.04 LTS Final Release Xenial como el mejor de los candidatos estudiados.

Esto es *Linux*, un gran sistema operativo que vale tanto para los mejores y mayores servidores que existen actualmente en la faz de la tierra como para los más pequeños sistemas móviles, y precisamente en todo ésto reside su grandeza, la universalidad.

“Gracias Señor Torvalds, su sistema es la esencia y el centro neurálgico de los sistemas operativos actuales de código abierto.”

“Su gran generosidad ha sentado precedentes no solo en el universo computacional, si no en todos los campos, y mostrando la dirección por donde deben ir todos los descubrimientos científicos, por las sendas del acceso libre y universal. “

6. Bibliografía y Webgrafía de Referencia

- Arch Linux. Página de información en Wikipedia. https://es.wikipedia.org/wiki/Arch_Linux
- Arch Linux ARM. Página web oficial del Proyecto ArchLinux ARM para Raspberry Pi 3. <https://archlinuxarm.org/platforms/armv8/broadcom/raspberry-pi-3>
- AArch64. Página de información de características en Wikipedia https://en.wikipedia.org/wiki/ARM_architecture#AArch64_features
- Benchmark. Página de información en Wikipedia. [https://es.wikipedia.org/wiki/Benchmark_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Benchmark_(inform%C3%A1tica))
- Ciclo de la investigación científica. Página de información en Wikipedia. https://es.wikipedia.org/wiki/Ciclo_de_la_investigaci%C3%B3n_cient%C3%ADfica
- CPU thermal throttling. Página de información en Wikipedia. https://en.wikipedia.org/wiki/Dynamic_frequency_scaling
- CRUX. Página de información en Wikipedia. <https://es.wikipedia.org/wiki/CRUX>
- Directorio de cambios de variables en ficheros de configuración para Raspberry Pi. Página de la plataforma de desarrollo colaborativo GitHub. <https://github.com/raspberrypi/linux/blob/rpi-4.4.y/arch/arm/boot/dts/overlays/README>
- Fedora ARM. Página web oficial del Proyecto. <https://arm.fedoraproject.org/>
- Fedora ARM para Raspberry Pi. Página web oficial del Proyecto para Raspberry Pi. https://fedoraproject.org/wiki/Architectures/ARM/Raspberry_Pi
- FreeBSD ARM. Página web oficial del Proyecto FreeBSD ARM. <https://wiki.freebsd.org/FreeBSD/arm/Raspberry%20Pi>
- GLMark2. Página de la plataforma de desarrollo colaborativo GitHub. <https://github.com/glmark2/glmark2>
- GParted. Página web oficial de GNOME Partition Editor. <https://gparted.org/>
- Instalación de Glmark2 en Debian. Blog de informática y Linux. <https://www.pcsuggest.com/install-glmark2-debian/>
- JeOS. Página de información oficial del entorno del software específico. <https://www.suse.com/es-es/products/server/jeos/>
- JeOS. Página de información en Wikipedia. <https://es.wikipedia.org/wiki/JeOS>
- LuaJIT es un compilador Just-In-Time (JIT) para el lenguaje de programación Lua. <http://luajit.org/>
- LXDE. Página oficial de configuración del entorno de escritorio LXDE para ArchLinux. <https://wiki.archlinux.org/index.php/LXDE>

- LXQt Desktop Panel. Página de la plataforma de desarrollo colaborativo GitHub.
<https://github.com/lxqt/lxqt-panel>
- Mate Desktop. Página de información oficial del entorno de escritorio.
<https://mate-desktop.org/es/>
- Netctl. Página oficial del cliente de perfiles de red para Arch Linux.
<https://wiki.archlinux.org/index.php/Netctl>
- Open-Source-Hardware. Página de información en Wikipedia.
https://en.wikipedia.org/wiki/Open-source_hardware
- Overclocking. Página de información en Wikipedia.
<https://es.wikipedia.org/wiki/Overclocking>
- Out of the box (funcionalidad). Página web de información en Wikipedia.
[https://en.wikipedia.org/wiki/Out_of_the_box_\(feature\)](https://en.wikipedia.org/wiki/Out_of_the_box_(feature))
- Packer. Página de la plataforma de desarrollo colaborativo GitHub.
<https://github.com/keenerd/packer>
- Pidora. Página web oficial de Pidora. <http://pidora.ca/>
- QEMU Raspberry Pi Kernel. Página de la plataforma de desarrollo colaborativo GitHub.
<https://github.com/dhruvvyas90/qemu-rpi-kernel>
- QEMU Raspberry Pi 3 Kernel. Página de la plataforma de desarrollo colaborativo GitHub.
<https://github.com/bztsrc/qemu-raspi3>
- QEMU. Página oficial del software virtualizador de código abierto. <https://www.qemu.org/>
- Raspberry Pi. Página web oficial de la Raspberry Pi. <https://www.raspberrypi.org/>
- Raspberry Pi. Página web de información en Wikipedia.
https://es.wikipedia.org/wiki/Raspberry_Pi
- Raspbian Stretch with Desktop. Página web oficial de Raspbian Stretch.
<https://www.raspberrypi.org/blog/raspbian-stretch/>
- Recalbox. Página de la plataforma de desarrollo colaborativo GitHub.
[https://github.com/recalbox/recalbox-os/wiki/Home-\(ES\)](https://github.com/recalbox/recalbox-os/wiki/Home-(ES))
- RetroPie. Página de la plataforma de desarrollo colaborativo GitHub.
<https://github.com/retropie/retropie-setup/wiki>
- *Rolling Distro*. Página web de información en Wikipedia.
https://en.wikipedia.org/wiki/Rolling_distribution
- Rpi-Monitor. Página de la plataforma de desarrollo colaborativo GitHub.
<https://github.com/XavierBerger/RPi-Monitor>

- SBC (Single-Board Computer). Página web de información en Wikipedia.
https://es.wikipedia.org/wiki/Placa_computadora
- Scratch. Página web oficial del Proyecto Scratch. <https://scratch.mit.edu/>
- SD Cards para Raspberry Pi. Página web oficial de Embedded Linux Wiki.
https://elinux.org/RPi_SD_cards
- SDDM. Página oficial de ArchLinux de Simple Desktop Display Manager.
<https://wiki.archlinux.org/index.php/SDDM>
- Sistemas Empotrados. Página de información de Wikipedía sobre sistemas embebidos.
https://es.wikipedia.org/wiki/Sistema_embebido
- Slackware ARM Project. Página web oficial del Proyecto.
<http://arm.slackware.com/supportedplatforms/>
- SoC. Sistema en Chip. Página de información de Wikipedía.
https://es.wikipedia.org/wiki/System_on_a_chip
- Sybench. Página de la plataforma de desarrollo colaborativo GitHub.
<https://github.com/akopytov/sysbench>
- Uso del comando DD. Página web Blog de Linux.
<https://blog.desdelinux.net/uso-del-comando-dd/>
- Victorhck in the free world. Blog especializado en GNU/Linux.
<https://victorhckinthefreeworld.com/2017/12/05/monitorizando-mi-raspberrypi-con-rpimonitor-en-opensuse/>
- Win32diskimager. Página web oficial de descarga del Proyecto.
<https://sourceforge.net/projects/win32diskimager/>
- YaST. Página de información de Wikipedía.
<https://es.wikipedia.org/wiki/YaST>