

RESUMEN

VIRTUALIZACIÓN Y TARJETAS EMBEBIDAS  
COMO NÚCLEO DE UN LABORATORIO  
DE CÓMPUTO

por

Pedro Robles López

Asesor principal: Carlos Emilio Hernández Rentería

## RESUMEN DE TESIS DE MAESTRÍA

Universidad de Morelos

Facultad de Ingeniería y Tecnología

Título: VIRTUALIZACIÓN Y TARJETAS EMBEBIDAS COMO NÚCLEO DE UN LABORATORIO DE CÓMPUTO

Nombre del investigador: Pedro Robles López

Asesor principal: Carlos Emilio Hernández Rentería, Maestría en Teleinformática

Fecha de terminación: Mayo de 2017

### Problema

Actualmente las Instituciones educativas privadas, no poseen laboratorios de cómputo y herramientas computacionales de calidad, que complementen el desarrollo de las aptitudes intelectuales de los alumnos. Por consiguiente, esta investigación se planteó las siguientes preguntas: ¿Cómo implementar un laboratorio de cómputo basado en tarjetas embebidas y virtualización de escritorio que permita mantener un desempeño óptimo? y ¿Qué tecnología puede ayudar a optimizar el uso del laboratorio de cómputo tradicional?

### Método

El desarrollo del prototipo del laboratorio de cómputo estará conformado por los siguientes pasos:

Comparar tarjetas embebidas. En este paso es indispensable conocer los beneficios que todas las tarjetas ofrecen y se elegirá la que mejor se adapte al modelo del laboratorio.

Evaluar software para implementación en tarjetas embebidas y servidor. Es necesario comparar las diferentes plataformas que ayuden a lograr la máxima compatibilidad de la tarjeta embebida. Los sistemas operativos utilizados son el Raspbian jessie, Raspbian jessie con pixel, fluxbox y Ubuntu mate 16.04. Al obtener una versión estable del procesador entre cliente y servidor, se hizo la instalación del escritorio virtual eyeOS en la Pc utilizada como servidor y se comprobó su desarrollo óptimo. Se examinaron diversos escenarios, teniendo al cliente conectado por medios guiados y no guiados para conocer su funcionamiento.

### Conclusiones

En relación con los temas planteados precedentemente y la importancia de los cambios actuales en ciencia y tecnología, es imprescindible el desarrollo de un prototipo de laboratorio de cómputo que optimice las herramientas computacionales disponibles para afrontar los problemas que enfrentan los laboratorios de cómputo tradicionales.

## DEDICATORIA

Agradezco a Dios por su inmenso amor y protección; sin él, nada de esto fuera posible.

A mis padres, Pedro Robles Cruz y Aracely López Hernández, por su amor, apoyo incondicional y sus consejos en el transcurso de mi vida. Gracias a su ejemplo y dedicación a Dios, soy una persona de bien.

A mi hermano, Abimael Robles López, por el apoyo que me ha otorgado.

A la contador público Flor de Cerezo Ontiveros Ramírez y su esposo, el ingeniero Josué Jair Rivera Palomo, por brindarme el apoyo para poder continuar mi preparación profesional.

Debo agradecer especialmente a las personas que de alguna manera me apoyaron desde el inicio de mis estudios en la maestría y me brindaron su amistad: maestra Ana Silvia Martínez Flores, contador público Madaí Hernández Flores, maestro Jaime Alcántara Quiroga y a los compañeros que tuve en el departamento de nómina.

Gracias a mi compañero y amigo, Hugo César Cadenas Franco, por haberme hablado del plan de la maestría y el apoyo incondicional brindado desde mi llegada a Montemorelos.

Quiero expresar de manera especial mi agradecimiento a mi asesor principal, el maestro Carlos Emilio Hernández Rentería, por haberme apoyado en el transcurso del desarrollo del proyecto; por su tiempo, dedicación y amistad. También al maestro Saulo Hernández Osoria, por su amistad y apoyo.

Universidad de Morelos  
Facultad de Ingeniería y Tecnología

VIRTUALIZACIÓN Y TARJETAS EMBEBIDAS COMO  
NÚCLEO DE UN LABORATORIO  
DE CÓMPUTO

Tesis  
presentada en cumplimiento parcial  
de los requisitos para el grado de  
Maestría en Ciencias Computacionales  
Área de Redes Informáticas

por

Pedro Robles López

Mayo del 2017

VIRTUALIZACIÓN Y TARJETAS EMBEBIDAS  
COMO NÚCLEO DE UN LABORATORIO  
DE CÓMPUTO

Tesis  
presentada en cumplimiento parcial de  
los requisitos para el título de  
Maestría en Ciencias Computacionales  
Área de Redes Informáticas

por

Pedro Robles López

APROBADA POR LA COMISIÓN:



Asesor principal: MT. Carlos Emilio  
Hernández Rentería



Dr. Ramón Andrés Díaz Valladares  
Examinador externo



Miembro: MC. Saulo Hernández Osoria



Dra. Raquel B. de Korniejczuk,  
Directora de Estudios de Posgrado



Miembro: MC. Alejandro Walterio  
García Mendoza

05 de Mayo del 2017  
Fecha de aprobación

## TABLA DE CONTENIDO

LISTA DE FIGURAS .....	vi
Capítulo	
I. INTRODUCCIÓN.....	1
Antecedentes .....	1
Historia de la virtualización .....	1
Virtualización. ....	2
Virtualización de servidores.....	3
Conceptos .....	4
Virtualización .....	4
Virtualización de servidores.....	5
Sistemas embebidos .....	6
Justificación .....	7
Definición del problema.....	8
Declaración del problema .....	8
Objetivos .....	8
Limitaciones .....	9
Delimitaciones .....	9
Materiales, métodos y técnica .....	9
Definición de términos.....	10
Marco filosófico .....	12
Lo que la Biblia enseña acerca de la optimización de los recursos tecnológicos .....	12
Dios da el conocimiento a quien lo pide y necesita.....	13
Optimización de los recursos naturales .....	15
Conservación de los recursos.....	17
La ética en el desarrollo e implementación de un proyecto.....	18
Cuidado con los excesos .....	19
II. MARCO TEÓRICO .....	21
Laboratorios escolares .....	21
Laboratorios de cómputo .....	22
Laboratorios de computación virtuales.....	23
Virtualización.....	24
Máquina virtual.....	25

Hypervisor.....	26
Tipos de virtualización.....	26
Virtualización de plataforma .....	26
Sistemas operativos invitados.....	27
Emuladores .....	28
Virtualización completa .....	28
Paravirtualización.....	29
Virtualización a nivel del sistema operativo.....	29
Virtualización a nivel de kernel.....	29
Virtualización de recursos.....	29
Encapsulación.....	30
Memoria virtual.....	30
Virtualización de almacenamiento.....	30
Virtualización de red.....	31
Unión de interfaces de red .....	31
Virtualización de hardware .....	31
Virtualización de aplicaciones.....	31
Virtualización limitada .....	32
Virtualización completa .....	32
Virtualización de escritorio.....	32
Ventajas y desventajas de la virtualización.....	33
Herramientas de virtualización para un servidor .....	35
Windows Server 2012 Hyper-V .....	35
Plataforma VMware vSphere 6.....	37
VMware workstation .....	38
EyeOS .....	39
Sistemas embebidos.....	41
Plataformas de sistemas embebidos .....	44
Raspberry Pi.....	44
Odroid-C2.....	46
Nvidia Jetson TK1 .....	47
Arduino .....	48
Ventajas y desventajas de los sistemas embebidos .....	50
Medios de transmisión .....	51
Medio de transmisión guiados y no guiados.....	52
III. METODOLOGÍA .....	56
Características del proyecto .....	56
Planificación.....	57
IV. RESULTADOS .....	59
Modelo de fases para el prototipo .....	59
Primera fase: pruebas de sistemas operativos.....	59
Segunda fase: servidor para virtualización .....	64



Tercera fase: comunicación cliente servidor.....	69
Cuarta fase: evaluaciones .....	73
V. CONCLUSIONES, DISCUSIÓN Y RECOMENDACIONES.....	83
Introducción .....	83
Conclusiones .....	84
Discusión. ....	85
Recomendaciones .....	86
Apéndice	
A. COSTOS DE LABORATORIO CONVENCIONAL Y PROTOTIPO .....	88
REFERENCIAS .....	89

## LISTA DE FIGURAS

1. Administración de hardware .....	25
2. Hypervisor bare metal .....	27
3. Hypervisor hosted .....	28
4. Plataforma de virtualización Windows server 2012 con Hyper-V .....	36
5. Plataforma vSphere 6.0.....	38
6. Arquitectura básica de Vmware workstation .....	39
7. Arquitectura eyeOS.....	40
8. Diagrama de bloque de un sistema embebido .....	41
9. Diagrama básico de Raspberry Pi 1 modelo B.....	44
10. Componentes del Raspberry Pi 3 modelo B .....	45
11. Componentes de placa Odroid C-2.....	46
12. Componentes de una placa Jetson TK1 .....	48
13. Componentes de placa Arduino .....	49
14. Escritorio Raspbian jessie en Raspberry Pi 1 modelo B .....	60
15. Barra de dirección y uso del procesador en porcentaje .....	60
16. Escritorio Raspbian jessie lite con Fluxbox en Raspberry Pi 1 modelo B .....	61
17. Escritorio Raspbian jessie en Raspberry Pi modelo B .....	62
18. Escritorio Ubuntu mate en Raspberry Pi 3 modelo B.....	63
19. Escritorio Ubuntu server 16.04 LTS .....	65
20. Permiso para ejecutar el instalador Xampp.....	65

21. Instalador de Xampp en Ubuntu server.....	66
22. Iniciar Xampp desde terminal en Ubuntu server .....	66
23. Ventana para iniciar los gestores necesarios en Ubuntu server .....	66
24. Base de datos en phpMyAdmin para eyeOS .....	68
25. Archivo PHP.ini en Ubuntu server.....	68
26. Ventana de instalación eyeOS 1.9.0.3.....	69
27. Ventana de inicio en eyeOS 1.9.0.3.....	70
28. Escritorio virtual eyeOS 1.9.0.3.....	70
29. Inicio eyeOs en Raspbian jessie desde Raspberry Pi 3 modelo B.....	71
30. Escritorio virtual eyeOs 1.9.0.3 en Raspbian jessie .....	71
31. Escritorio virtual en Raspberry Pi 3 modelo B.....	72
32. Escritorio virtual en Raspberry Pi 3 modelo B con Ubuntu mate.....	73
33. Especificaciones de ordenador utilizado como servidor.....	74
34. Escritorio de Raspbian jessie con pixel.....	74
35. Comando htop en Raspberry Pi 1 modelo B.....	75
36. Comando systemd_analyze en Raspberry Pi 1 modelo B .....	75
37. Comando traceroute en Raspberry Pi 1 modelo B.....	76
38. Comando htop en Raspberry Pi 3 modelo B.....	76
39. Comando systemd_analyze en Raspberry Pi 3 modelo B .....	76
40. Comando traceroute en Raspberry Pi 3 modelo B UTP.....	77
41. Comando traceroute en Raspberry Pi 3 modelo B Wi-Fi .....	77
42. Comando iperf en Raspberry Pi 1 modelo B.....	78
43. Comando iperf en Raspberry Pi 3 modelo B UTP.....	78

44. Comando iperf en Raspberry Pi 3 modelo B Wi-Fi.....	78
45. Comparación entre UTP y Wi-Fi del Raspi 1 y 3.....	79
46. Uso del CPU en conexión Raspi 1 modelo B y ordenador.....	79
47. Uso del CPU en conexión Raspi 3 modelo B y ordenador.....	80
48. Uso del CPU en server al recibir cuatro conexiones.....	80
49. Tiempo de inicio en servidor.....	81

## **CAPÍTULO I**

### **INTRODUCCIÓN**

#### **Antecedentes**

##### Historia de la virtualización

Se ha considerado a la virtualización un tema novedoso pero en realidad su desarrollo es mucho más histórico. El proyecto Atlas en la década de los 60 fue un precursor para solucionar problemas que habían surgido, por el uso de un solo ordenador para realizar procesamiento de información mediante terminales. Atlas consistía en compartir los recursos de procesador y disco para que todos los usuarios conectados pudieran hacer sus procesos. Este método fue el inicio de la virtualización para encontrar una manera más eficiente de compartir los recursos de un ordenador. Estas particiones o máquinas virtuales podían obtener una instancia o imagen del sistema operativo, comunicarse entre ellas por red y utilizar los recursos disponibles del resto de los usuarios (Villar Fernández y Gómez López, 2010).

IBM en la misma década comenzó analizando el concepto de virtualización y desde ese momento el procesamiento de información pasó a ser un factor muy importante para el surgimiento de tecnologías que facilitan este proceso. IBM fortaleció la virtualización de máquinas con la creación de un mainframe system/360 que tenía dos procesadores y hasta 2 MB de memoria RAM, contenía un sistema operativo CP/CMS diseñado por IBM y en esa época se reconoció su rendimiento productivo. Al paso del

tiempo se fue ampliando el desarrollo de máquinas con menor procesamiento y que estaban al alcance del mercado, utilizando un sistema operativo abierto con un procesamiento distribuido, esto a su vez, generó una manera más económica de procesar la información (Augusto Marchionni y Martín Formoso, 2012).

## Virtualización

Cualquier compañía necesita actualmente contar con una excelente estructura tecnológica para poder mantener mayor producción a un buen costo, pero también se necesita que ésta tecnología sea confiable, escalable y fácil de implementar. El crecimiento constante en la infraestructura exige tener acceso a mejores soluciones a nivel de almacenamiento físico, es por este motivo que la virtualización ocupa un lugar importante en las empresas que buscan mayor eficiencia en procesos y una reducción de sus costos, ésta tecnología disminuye el tiempo de respuesta, reduce fallas y el espacio físico (Velásquez, 2009).

Existen muchas ventajas al utilizar la virtualización, tales como el ahorro de energía, la reducción de costos en mantenimiento y una mejor administración de recursos. También ofrece mayor seguridad y mejor acceso a los datos; esta tecnología extrae el software y lo encapsula para su mejor optimización en una máquina virtual (Martín, Marrero, Urbano, Barra y Moreiro, 2011).

Actualmente la tecnología de virtualización es una de las tecnologías importantes que proporcionan amplios beneficios en diferentes campos como la educación y la investigación. En conjunto con las nuevas aplicaciones sobre la nube, la virtualización proporciona un entorno amable para desplegar servicios y una alternativa para las diferentes tecnologías emergentes (Ma, Teng, Du y Zhang, 2014).

Las plataformas de virtualización se han utilizado para el proceso de enseñanza-aprendizaje en el área de educación. A menudo se utiliza la virtualización de hardware o de servidor para ofrecer máquinas virtuales en un entorno aislado y seguro. Con esto se evitan errores que afecten a todas las unidades virtuales. Las diferentes tecnologías combinadas con la virtualización permiten el acceso desde cualquier sitio entre el cliente y el servidor, centralizan los datos y simplifican las tareas administrativas (Misevičienė, Ambrazienė, Tuminauskas y pažereckas, 2012).

### Virtualización de servidores

La virtualización es una técnica que se ha ido mejorando a lo largo del tiempo, ya que desde sus inicios, con el monitor de máquinas virtuales (VMM), llamado después como hipervisor y la necesidad de particionar recursos de una computadora, se ha logrado demostrar la importancia que tiene la virtualización de servidores para conseguir que cualquier empresa pueda utilizar un servidor físico para virtualizarlo y así lograr un balance entre costo y rendimiento.

El concepto de virtualización de servidores no es nuevo, ya que en la década de los 60 se comenzó a investigar sobre su utilidad y su estudio se ha extendido de tal manera que en la actualidad ha generado nuevos campos de investigación. En términos de virtualización se encuentra que ésta tecnología implica hacer que un recurso o servidor físico pueda visualizarse como si fuera uno o múltiples recursos lógicos a la vez, de esta manera los clientes se conectan al servidor principal para obtener archivos o administrar recursos en la red (Velázquez, 2009).

La virtualización de servidores mejora la eficacia y la disponibilidad de recursos y aplicaciones en tecnologías de información (TI), crea un ambiente en donde la velocidad

y la capacidad de ampliación con el que se logra el despliegue de un servicio puede llegar a dividirse tan solo a unos sencillos pasos. Libera a los administradores de sistemas de las tareas administrativas repetitivas que pueden conducir a brechas de seguridad. A causa de los bajos precios de un computador y del aumento en el rendimiento de los servidores, el mercado de soluciones en virtualización se ha extendido en los últimos años haciendo posible el uso de múltiples plataformas de virtualización (Mârgescu y Haller, 2013).

La virtualización de servidores se encuentra a un nivel más alto que los servidores físicos utilizados actualmente, este tipo de virtualización provee una amplia plataforma de ejecución para administrar múltiples aplicaciones utilizando máquinas virtuales (Hao, Fu, Trenkamp y Prapatanant, 2012).

Otro planteamiento en la misma dirección es que la virtualización de servidores es un modelo especialmente diseñado para mostrar un servidor físico particionado, que puede alojar múltiples servidores dedicados o privados virtuales, los cuales pueden ejecutar independientemente un sistema operativo propio y efectuar dentro del mismo los diferentes tipos de servicios que se requieran ofrecer (Villar Fernández y Gómez López, 2010).

## **Conceptos**

### Virtualización

La virtualización es una técnica utilizada para controlar de una manera eficaz las interacciones entre usuarios, aplicaciones, sistemas y recursos informáticos por medio de una especificación física concreta. La virtualización se puede aplicar a los elementos relacionados con los ordenadores y otros recursos de hardware (Ahn et al., 2012).



La virtualización atenúa y estimula innovación al permitir la flexibilidad, diversidad, capacidad de administración y seguridad (Duan y Lu, 2013).

La virtualización es la técnica empleada sobre las características físicas de algunos recursos computacionales, esto implica hacer que un recurso físico simule uno o varios recursos lógicos (Velásquez, 2009).

La virtualización analiza dos rasgos: costo y rendimiento, es por eso que la infraestructura virtual marca el comienzo de una era donde la prioridad es aprovechar los recursos físicos del procesamiento distribuido para optimizar la infraestructura tecnológica (Augusto Marchionni y Martín Formoso, 2012).

### Virtualización de servidores

La virtualización de servidor se encuentra en la parte alta de los servidores físicos, consiste en una o varias máquinas virtuales instaladas en un recurso físico. El servidor proporciona una plataforma para la ejecución de aplicaciones o de cualquier otro servicio necesario (Hao et al., 2012).

Consiste en un computador principal al que cualquier cliente u otros ordenadores se conectan para gestionar archivos o administrar los recursos instalados en el servidor, en la virtualización de servidores es donde se particiona el servidor físico y se divide en pequeños recursos virtuales (Velásquez, 2009).

Según Bonilla Suárez y Carrasco Aguilar (2010), la virtualización se puede clasificar en 3 tipos: virtualización de sistema operativo, virtualización completa y paravirtualización. La primera consiste en virtualización completa a nivel de sistema operativo, la segunda se caracteriza por simular todo el hardware en el sistema huésped y la última técnica provee simulación parcial del hardware.

La virtualización de servidores es un método usado actualmente para la ejecución de múltiples sistemas operativos en un único servidor físico. Las particiones virtuales se asignan de acuerdo a las necesidades de las aplicaciones, el objetivo es obtener una mayor densidad del servidor y optimizar el hardware, la virtualización de servidores es excelente para las aplicaciones que tiene gran demanda en el mercado de las tecnologías de información (Mahesh, Trumbach y Walsh, 2012).

La virtualización actual es capaz de alcanzar una mayor eficiencia de tecnologías de información (TI) y las capacidades que tiene una máquina virtual en un recurso físico ha superado las expectativas de un entorno virtualizado, las plataformas de virtualización son utilizadas para administrar servicios y obtener una mejor escala en la gestión de recursos (Microsoft, 2013).

### Sistemas embebidos

Los sistemas embebidos han logrado combinar hardware y software para ofrecer diversos componentes, en los cuáles se puede instalar un sistema que mejore ampliamente cualquier dispositivo. Entre las ventajas de estos sistemas destaca la flexibilidad ante posibles fallos, debido a que se pueden realizar modificaciones de una manera más fácil y sin tener que reemplazar todo el circuito. Al utilizar esta clase de sistemas embebidos en tiempo real, es posible mantener en control todas las características y lograr el correcto funcionamiento entre el hardware y software. Al trabajar con equipos en tiempo real, es crucial que todas las funcionalidades del sistema sean correctas y es imprescindible una tolerancia a fallos eficiente, es por esta razón que los sistemas embebidos han ido incursionando en diversas áreas del mercado. Al tener un bajo costo, a diferencia de

una PC común, se pueden llegar a optimizar los recursos y reducir el consumo de energía. El sistema operativo que se utiliza para este tipo de sistemas embebidos es el GNU/Linux, ya que ofrece ser una opción libre y confiable para múltiples arquitecturas de hardware (Iglesias Benitez, Toledano Hernández, Toledo Enríquez, Martínez Casado y González Aguilera, 2014).

### **Justificación**

Las instituciones educativas privadas cuentan con sistemas de aprendizaje eficientes, que desarrollan de forma completa las aptitudes de cada uno de los alumnos. La Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO, 2017, párr. 3) declara que

en un mundo que evoluciona con rapidez, la educación científica y tecnológica es un instrumento importante para la consecución del desarrollo sostenible y la reducción de la pobreza. Sin embargo, los sistemas educativos afrontan el problema de que esa enseñanza ha perdido pertinencia, por no haberse adaptado a los cambios actuales en materia de ciencia y tecnología.

De esta manera, para lograr el avance tecnológico en las instituciones, es imprescindible el uso de un laboratorio de cómputo idóneo. Se debe considerar que la utilización de los ordenadores es fundamental para la realización de las tareas, investigaciones y clases de los alumnos. Al implementar la infraestructura precisa del laboratorio de cómputo, se pretende incrementar el desarrollo intelectual de cada alumno e instruirlo con las herramientas más modernas. El no poseer laboratorios de cómputo de calidad, sistemas y herramientas computacionales apropiadas pueden estar limitando el desarrollo de las aptitudes intelectuales de los alumnos (Duarte, Gargiulo y Moreno, 2011).

Es importante conocer todas las tecnologías que existen y si estas pueden ayudar a resolver los problemas actuales. Dado el costo que tiene su implementación, se busca elegir la herramienta que permita usabilidad, ahorro en los costos de mantenimiento y ser el complemento de los recursos físicos y virtuales disponibles. La implementación de un laboratorio virtual y remoto para la enseñanza del diseño y administración de redes de computadoras ha dado como resultado la interacción entre el medio físico y virtual, permitiendo satisfacer las necesidades de alumnos y maestros (Britos, Vargas, Arias, Giraudo y Veneranda, 2013).

### **Definición del problema**

Con el propósito de ser una herramienta para el aprendizaje, el actual proyecto propone desarrollar un prototipo de laboratorio de cómputo mediante el uso de virtualización y tarjetas embebidas, que permita lograr reducir el consumo energético y espacio físico.

### **Declaración del problema**

El presente proyecto plantea el desarrollo de un prototipo de laboratorio de cómputo, utilizando software de virtualización en el ordenador servidor y tarjetas embebidas como clientes, conectados por medio guiado y no guiado.

### **Objetivos**

Los objetivos del presente estudio son los siguientes:

1. Crear un prototipo de laboratorio de cómputo capaz de aumentar la eficiencia de las herramientas computacionales.

2. Desarrollar una infraestructura tecnológica con el propósito de estar a la vanguardia.

### **Limitaciones**

Para fines del presente estudio, se plantearon las siguientes limitaciones:

1. Licenciamiento del software.
2. Falta de recursos económicos.

### **Delimitaciones**

Las delimitaciones del presente estudio, fueron las siguientes:

1. Solo se desarrollará el prototipo.
2. Trabajar con software de virtualización open source.
3. Trabajar con software de tarjetas embebidas open source.

### **Materiales, métodos y técnica**

El desarrollo del laboratorio de cómputo estará conformado por los siguientes pasos:

1. comparar tarjetas embebidas. En este paso es indispensable conocer los beneficios que todas las tarjetas ofrecen y elegir la que mejor se adapte al modelo del laboratorio.

2. Evaluar software para implementación en tarjetas embebidas y servidor. Es necesario comparar las diversas plataformas que ayuden a lograr la máxima compatibilidad de la tarjeta embebida y el servidor.

3. Medio de transmisión guiado. Para lograr la transmisión de datos en una red, se debe contar con el medio de transmisión correcto. Es importante que los dispositivos transmitan los datos de manera eficiente y sin ruido. Existen tres tipos de medios alámbricos: Par trenzado, coaxial y fibra óptica.

4. Medio de transmisión no guiado. Para llevar a cabo la transmisión y recepción de datos es imprescindible el uso de antenas. La antena emite energía electromagnética hacia el receptor que recibe los datos por medio de ondas electromagnéticas. El rango de frecuencias se puede clasificar en tres tipos: radiofrecuencias, microondas e infrarrojos.

### **Definición de términos**

Los principales términos usados en el tema se definen en el contexto del estudio de la siguiente manera:

*Aplicaciones*: tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajos.

*GNOME*: entorno de escritorio e infraestructura de desarrollo para sistemas basados en GNU/LINUX.

*Máquina virtual*: software que emula a una computadora y puede ejecutar programas como si fuera una computadora normal.

*Medio de transmisión guiado*: conductor de señales enviadas desde un dispositivo a otro por medio de cables.

*Medio de transmisión no guiado*: transporte de ondas electromagnéticas sin usar un conductor; pueden ir del origen al destino a través de superficie, cielo y en línea de visión.

*Paravirtualización:* técnica de programación informática que permite virtualizar por software sistemas operativos.

*Red de datos escolar:* conjunto de ordenadores conectados entre sí a través de un medio físico (cable o inalámbrico) y por equipos de transmisión de datos (routers), para facilitar compartir archivos, aplicaciones, etc.

*Servidor:* tipo de software que realiza ciertas tareas en nombre de los usuarios.

*Sistema embebido:* dispositivo utilizado para controlar, supervisar o ayudar en la operación de equipos, maquinaria o planta. “incrustado” refleja el hecho de que son parte integral del sistema. En muchos casos su “incrustación” puede ser tal que su presencia no es evidente para un cliente.

*Tecnologías de la información:* cualquier forma de hacer cómputo.

*Virtualización:* técnica empleada sobre las características físicas de algunos recursos computacionales, para ocultarlas de otros sistemas, aplicaciones o usuarios que interactúen con ellos.

*Virtualización de servidores:* utilización de software para crear máquinas virtuales que emulen un host físico.

*Virtualización de sistema operativo:* uso de una aplicación de software para permitir que un mismo sistema operativo administre varias imágenes de los sistemas operativos.

*Virtualización completa:* máquina virtual que simula un hardware suficiente para permitir un sistema operativo “huésped” sin modificarlo para ejecutar de forma aislada.

## **Marco filosófico**

### Lo que la Biblia enseña acerca de la optimización de los recursos tecnológicos

Cuando se habla de optimización de recursos, se indica con frecuencia un manejo adecuado de los recursos o materiales que se tienen a la mano para el desarrollo de actividades específicas. Sin embargo, conlleva dos palabras que es necesario definir: optimización y recursos. La optimización es mayormente utilizada en el área de informática y de sistemas para describir el desarrollo de un sistema de forma óptima. También se utiliza para hacer referencia a una manera más eficiente de optimizar el hardware y conseguir el mejor rendimiento con la plataforma adecuada.

En las diversas áreas del conocimiento, se tiene a disposición un amplio número de recursos; es necesario administrar los recursos de una manera inteligente, con la finalidad de lograr los objetivos propuestos. En el área de la tecnología se puede optimizar un recurso específico para que se ejecute con mayor rapidez y que sus operaciones obtengan un mayor rendimiento de la red. En los últimos años, se ha intensificado el estudio con respecto a las formas de optimizar un recurso de hardware, mediante acciones que generen resultados óptimos en los procesos de ejecución. Esto se debe a que las personas necesitan soluciones inteligentes que les lleve a obtener un balance entre el costo y el rendimiento de los recursos.

En el área de sistemas, los recursos son todos aquellos componentes de hardware y/o software dedicados a mejorar el funcionamiento de ordenadores y periféricos de forma individual o colectiva.



## **Dios da el conocimiento a quien lo pide y necesita**

Después que Jacob engañó a su hermano mayor Esaú para obtener la primogenitura, huyó a la tierra de los orientales para salvar su vida. Al llegar a Harán preguntó a los moradores si conocían a un hombre llamado Labán hijo de Nacor. Labán era hermano de Rebeca, madre de Jacob y Esaú. Al sentarse en el pozo de la ciudad observó a una joven llamada Raquel que se acercaba con su rebaño. Jacob quitó la piedra y abrevó el rebaño del hermano de su madre. Cuando Labán reconoció a Jacob, le ofreció la hospitalidad por el tiempo que deseara quedarse. Después de trabajar un tiempo apacentando las ovejas, Labán le dijo a Jacob que no era justo que trabajara sin recibir algún pago, entonces Jacob decidió trabajar siete años para casarse con Raquel, la hija menor de Labán. Al finalizar los siete años, Labán no cumplió el trato y le dio por esposa a Lea, su hija mayor. Jacob trabajó de nuevo durante otros 7 años para casarse con Raquel. Después de mucho tiempo en la casa de su suegro Labán, Jacob decidió regresar a su tierra natal junto a sus esposas e hijos. Labán, que había sido bendecido a causa de su yerno, no quiso dejarlo ir y le ofreció un nuevo trato. Labán le dijo a Jacob que todas las ovejas con manchas, de color oscuro y salpicadas de color serían su salario y que las demás quedarían en manos de él. Dios prosperó en abundancia a Jacob, pese a que Labán le cambió el salario muchas veces. En momentos le daba como salario las ovejas con manchas y en otro las blancas. De esta manera, Dios le quitó el ganado a Labán y se lo dio a Jacob. Por mandato de Dios Jacob regresó a la tierra de su nacimiento y fue bendecido en abundancia por obedecer lo que Dios le ordenó (Génesis 30: 31-43).

Jacob utilizó eficientemente los recursos que tenía disponibles para lograr alcanzar sus objetivos en el menor tiempo posible, aumentando la calidad al realizar bien lo

que esperaba obtener. En un laboratorio de cómputo, se utilizan los recursos tecnológicos para estimular el aprendizaje y enseñar sobre nuevas tendencias del mundo actual. La tecnología puede influir positivamente sobre lo que se enseña en los colegios, sin embargo, el conocimiento humano es reducido en comparación con lo que Dios puede enseñar al que pide la sabiduría de lo alto, ofreciéndose con un corazón contrito y humillado. Dios desea que todos sus hijos utilicen sabiamente los recursos que tienen a su disposición. Al seguir sus órdenes específicamente, se pueden lograr todas las metas. Cuando Jacob entendió que por sus fuerzas no podría lograr los objetivos que se había propuesto y escuchó la voz del Creador, comenzó a ser bendecido en gran manera hasta obtener mucha riqueza. Dios es la fuente del conocimiento y Él da a sus hijos la ayuda en las proporciones necesarias.

En la actualidad es necesario que las tecnologías sean utilizadas adecuadamente en el entorno educativo. Es esencial facilitar a los educadores las herramientas necesarias para obtener la calidad en la educación. Se crea un ambiente apto para superar los desafíos permitiendo a los estudiantes avanzar con éxito en la carrera de la vida. En la Biblia se encuentra registrado un personaje importante que demostró el verdadero camino para superar los obstáculos, lograr calidad en la optimización de los recursos y poder combinar el conocimiento humano con el que Dios otorga.

Al combinar el entendimiento humano con el divino, es posible obtener resultados espléndidos. Cuando Dios decide dar el conocimiento para lograr un propósito especial, entonces el resultado es una obra digna del Creador. Un ejemplo es la misma construcción del santuario. Dios hizo un llamado a Bezahleel y Aholiab, ambos fueron llenados del Espíritu de Dios en sabiduría, en ciencia, inteligencia y en todo tipo de arte. Aholiab

fue inspirado para tener el don de enseñar sobre los dones que Dios le dio. Los llenó de sabiduría de corazón para hacer toda labor e inventar diseños (Éxodo 35:30-35).

Dios puede llenar de sabiduría a cada educador para que sea ejemplo del educando en proceso de superación. Desde el momento que el educador pide ayuda al Creador, las puertas de los cielos son abiertas y es derramado el Espíritu sobre quien lo necesita. El manantial de la sabiduría fluye para quienes desean en verdad conocer el origen del conocimiento.

### **Optimización de los recursos naturales**

A lo largo de la historia se ha conocido el resultado que ha dejado el ser humano sobre el medio ambiente. Se ha perdido el concepto esencial sobre lo que es la mayoría de la creación. Dios, en su infinito amor, dejó la naturaleza para aprender de ella; es triste ver cómo el ser humano destruye poco a poco cada una de las creaciones del Creador. En los primeros capítulos del Génesis se enseña sobre el origen de la tierra y la finalidad de las plantas, animales, lumbreras, aves, árboles, etc. La naturaleza es el reflejo de su Creador y en ella se demuestra humildad, compromiso y cada uno de los valores estimados del ser humano. “Entonces Dios contempló todo lo que había hecho, y vio que era bueno en gran manera” (Génesis 1:31). La obra del Creador no fue hecha sin planificación; cada una de las partes fue una pieza perfecta del rompecabezas.

El ser humano es responsable de cuidar toda la naturaleza y debe velar para obtener la mejor optimización de los recursos naturales. “Entonces dijo Dios: Hagamos al hombre a nuestra imagen, conforme a nuestra semejanza; y que domine los peces del mar, las aves del cielo, el ganado y todo animal que anda sobre la tierra. Llenad la tierra y gobernadla” (Génesis 1:26-27).

“Del Señor es la tierra y su plenitud, el mundo y los que habitan en él. Porque él la fundó sobre los mares, y la afirmó sobre las aguas” (Salmos 24:1-2). El salmista destaca al Creador del Universo y demuestra que no hay otro como él. “¡Comerás del trabajo de tus manos, serás dichoso, y todo te irá bien!” (Salmos 128:2). “Amado, deseo que prosperes en todo, y tengas salud, así como prosperas espiritualmente” (3 Juan 1:2). El propósito de Dios es que el ser humano utilice la naturaleza para obtener el alimento diario, el resultado de alimentarse sanamente será una vida llena de salud y en comunión estrecha con el Creador.

“Los atributos invisibles de Dios, su eterno poder y su divinidad, se ven claramente desde la creación del mundo, y se entienden por las cosas que han sido creadas; de modo que no tienen excusa” (Romanos 1:20). El ser humano es el único creado a imagen y semejanza del Creador. La naturaleza es parte de su creación y refleja su poder. Se debe respetar a la naturaleza como el reflejo de Dios y aprender a admirar cada una de las maravillas de la naturaleza y sus lecciones para la vida cotidiana. La grandeza del Creador se revela mediante su creación y el ser humano debe conocer el diseño perfecto de la naturaleza y disfrutar de los frutos que otorga. El futuro de la tierra depende del buen uso que el ser humano pueda ofrecer a los recursos de la naturaleza. La naturaleza es la escuela que Dios dejó para que el hombre aprenda sobre la administración de los recursos. Dios como Creador y Maestro demostró que es posible llevar una vida en completa armonía con la naturaleza. Un autor dice que la creación es finita por naturaleza y que solo Dios es infinito. Por lo tanto, Dios es el que sustenta el mundo natural. La naturaleza no posee vida en sí misma. Dios, por medio de Cristo, es el que lo mantiene (Nehemías 9:6; Colosenses 1:17) (¿Los desastres naturales son castigo de Dios?).

## **Conservación de los recursos**

Es importante mantener los recursos naturales, tecnológicos o los que estén a disposición del ser humano, con la finalidad de conservar la tierra.

La creación aguarda con profundo anhelo que los hijos de Dios sean revelados. Porque la creación fue sometida a frustración, no por su propia elección, sino por la voluntad del que la sujetó, con esperanza de que la misma creación será librada de la esclavitud de la corrupción, para participar de la gloriosa libertad de los hijos de Dios. Sabemos que hasta el presente, todas las criaturas gimen a una, y a una sufren dolores como de parto. (Romanos 8:19-22)

Dios ha dotado al ser humano de la capacidad para administrar todos los recursos que se le ofrecen. Lamentablemente, el trabajo no se ha hecho de la mejor manera y Pablo confirma que la creación espera la restauración de la tierra para restituir el orden de la naturaleza.

La Palabra de Dios registra un evento que influye en la conservación de los recursos; la alimentación de los cinco mil. Jesús se encontraba al otro lado del mar de Galilea, en Tiberias. Una gran multitud lo seguía a causa de que veían las señales que hacía en los enfermos. En medio del ajetreo, uno de los discípulos de Jesús le preguntó: ¿dónde compraremos pan para toda esta gente?; otro de ellos contestó: aquí se encuentra un muchacho que tiene cinco panes de cebada y dos pescados, pero ¿qué es esto para tantos? Jesús tenía el poder y sus discípulos aún no habían aprendido a tener fe en medio de los problemas. Luego de hacer una oración sobre los panes y los peces, hubo suficiente alimento para alimentar a toda la multitud. Posteriormente, Jesús les ordenó que recogieran el sobrante de la comida y lo colocaran en cestas. En aquel momento se juntaron doce cestas de pedazos sobrantes de los cinco panes de cebada (Juan 6: 1-13). Jesús era consciente de la conservación de los recursos y en la alimentación de los

cinco mil lo resalta. Dios otorga bendiciones en abundancia para satisfacer todas las necesidades, lo único que pide es que el ser humano administre sabiamente esos recursos. Hay otro evento que resalta el buen uso de los recursos y se trata del alimento que Dios otorgó a su pueblo mientras cruzaban el desierto. Los israelitas se quejaban constantemente de las vicisitudes que tenían cada día y olvidaron que Dios los había sacado de la amarga servidumbre de Egipto. Por este motivo, Dios les dio el maná como alimento que les caía por las mañanas. La Palabra de Dios registra que el maná era igual a hojuelas con miel y se podía preparar como el trigo. Moisés les dijo que caería todas las mañanas con excepción del día sábado y les recomendó que recogieran aproximadamente dos litros por persona cada día. No obstante, el sexto día deberían recoger doble porción para el sábado, porque el sábado no habría en el suelo (Éxodo 16-17). El maná otorgado por Dios es un sabio ejemplo sobre la conservación que el ser humano debe hacer de los recursos. El pueblo de Israel debía obedecer el mandato divino; algunos que hicieron caso omiso se dieron cuenta de que la ley de Dios es absoluta y que Dios es un Dios de orden. Dios quiere que todos sus hijos aprendan a conservar los recursos y puedan llevar una vida próspera.

### **La ética en el desarrollo e implementación de un proyecto**

Aristóteles consideró a la ética como el estudio del comportamiento que puede ser considerado bueno o malo. El código de ética es importante en todas las áreas que desempeña el ser humano. Es lo que hace la diferencia en el momento de conocer las virtudes de una persona. El apóstol Pablo exhorta al ser humano a poner la vista en las cosas de arriba y no en las de la tierra (Colosenses 3:1-4). La Biblia otorga al ser humano

las herramientas necesarias para vivir en armonía con los principios bíblicos; la ética es poner la mirada en las cosas celestiales. Los principios que rigen al ser humano deben ser conforme a la vida que lleva; es un estándar que conduce a los principios morales. “Por eso, desechad la mentira, hablad la verdad cada uno con su prójimo; porque somos miembros los unos de los otros” (Efesios 4:25). Pablo se preocupa por la ética del hombre y ofrece consejos inspirados por el Espíritu para regir la vida del cristiano. “El que robaba, no robe más; antes trabaje y haga algo útil con sus propias manos, para tener con que ayudar al necesitado” (Efesios 4:28). El trabajo útil debe ser parte importante en la formación de la ética, se genera un ambiente libre de malas influencias y se lleva a cabo el uso correcto de los recursos. “Sed, pues, perfectos, como vuestro Padre celestial es perfecto” (Mateo 5:48).

### **Cuidado con los excesos**

El ser humano, por naturaleza, tiende a propasarse en el uso de los recursos tecnológicos; es esencial que todos los recursos obtengan el tiempo necesario para su desarrollo y un periodo de mantenimiento o descanso. La naturaleza es un vivo ejemplo de optimización y la palabra de Dios lo confirma. Jehová dio recomendaciones específicas al pueblo de Israel para conservar la vida de la tierra y mantenerla productiva. Los israelitas podían sembrar la tierra y podar las viñas para recoger sus frutos. No obstante, el séptimo año, la tierra debía tener un reposo para Jehová, libre de siembra y cosecha. Ese año es de completo reposo para la tierra y con el fruto de los años anteriores alimentarán a toda su casa. Y Jehová hizo provisión para que, en el sexto año, la tierra otorgara fruto hasta por tres años (Levítico 25).

La Palabra de Dios muestra el interés que Dios tiene para que el hombre aprenda sobre el uso correcto de los recursos. El hombre es mayordomo de toda la obra maestra que el Creador diseñó. La Biblia tiene mucho que decir acerca de la optimización de los recursos y el empleo apropiado de los recursos informáticos, naturales, espirituales, etc. Jesús lo ejemplificó magistralmente en el relato de la parábola de los talentos.

A uno dio cinco talentos, y a otro dos, y a otro uno, a cada uno conforme a su capacidad, y luego se fue lejos. Después de mucho tiempo vino el señor de aquellos siervos, y arregló cuentas con ellos... Y llegando el que había recibido cinco talentos, trajo otros cinco talentos, diciendo: Señor, cinco talentos me entregaste; aquí tienes, he ganado otros cinco talentos sobre ellos. Y su señor le dijo: Bien, buen siervo y fiel; sobre poco has sido fiel, sobre mucho te pondré; entra en el gozo de tu señor... Pero llegando también el que había recibido un talento, dijo: Señor, te conocía que eres hombre duro, que siegas donde no sembraste y recoges donde no esparciste. Por lo cual tuve miedo, y fui y escondí tu talento en la tierra; aquí tienes lo que es tuyo. Respondiendo su señor, le dijo: Siervo malo y negligente, sabías que siego donde no sembré, y que recojo donde no esparcí... Porque al que tiene, le será dado, y tendrá más; y al que no tiene, aun lo que tiene le será quitado. (Mateo 25:14-30)

Es considerable utilizar sabiamente los recursos que Dios otorga al ser humano. Como mayordomos de toda su creación, es indispensable pedir ayuda divina para administrar sus recursos de una forma eficiente. Dios da el conocimiento a quien en verdad lo solicita con un corazón humilde. Saber utilizar los recursos informáticos es importante, pero no es suficiente. Se necesita poner la voluntad en las manos de Dios para que ese conocimiento se transforme en acciones responsables ante Dios y los hombres.



## **CAPÍTULO II**

### **MARCO TEÓRICO**

En este capítulo se presentan los conceptos del proyecto y los temas imprescindibles para su desarrollo. Se mencionan algunas metodologías sobre los laboratorios escolares, laboratorios de cómputo, laboratorios virtuales, conceptos generales sobre virtualización, sistemas embebidos y el medio de transmisión.

#### **Laboratorios escolares**

La tecnología en las escuelas va retomando importancia con el paso de los años. La integración de los laboratorios de cómputo en las escuelas comenzó desde la década de los 90. En México, con ayuda de algunas empresas privadas y el gobierno, se obtuvo una sala de cómputo que integraba de cinco a 20 ordenadores. Las aulas se utilizaban para enseñar computación, los alumnos y profesores tuvieron la oportunidad de aprender significativamente sobre su función (Heredia Escorza, 2010).

Es imprescindible disponer de un laboratorio para la enseñanza y la investigación; de esta manera se enriquece la teoría con la práctica. Se puede experimentar sobre lo que dicen los libros y aprender mediante los errores que pueden surgir en el transcurso de la práctica. Un laboratorio es un lugar donde se provee de todo lo necesario para experimentar de acuerdo con el área enfocada. Se puede mencionar que las computadoras añaden un amplio conocimiento sobre lo que se aprende en las aulas de clases impartidas por docentes. Con la implementación de estos laboratorios, es posible mejorar

el análisis de resultados y realizar observaciones sobre proyectos. Un laboratorio bien equipado es parte fundamental de la enseñanza y contribuye a reforzar el conocimiento que se obtiene de la teoría (Lugo, 2006).

### **Laboratorios de cómputo**

Al utilizar herramientas de computación desarrolladas en código libre e implementarlas en un laboratorio de cómputo, se puede llegar a optimizar los ordenadores y reducir el consumo de energía sin afectar el rendimiento. Al combinar los recursos del laboratorio con la enseñanza, es posible obtener una mejoría en el aprendizaje de los alumnos (Marco Rivadeneira, 2014).

La forma en la que los estudiantes conciben el laboratorio de cómputo refleja la percepción que tienen sobre su manera de aprender y lograr sus objetivos. Cuando se ofrece un curso de computación, es imprescindible el uso de una sala de cómputo para complementar la teoría. Sin embargo, una mejora en la gestión del laboratorio de cómputo puede facilitar la enseñanza y el aprendizaje (Syazwani y Choong-Yeun, 2012).

Para un usuario avanzado, es necesario complementar la teoría con la práctica y obtener el conocimiento en los errores que puedan surgir en el transcurso de la implementación. Los laboratorios de cómputo proporcionan el ambiente de aprendizaje idóneo para reforzar el estudio por competencias (Ahiatrogah y Adane, 2011).

Los laboratorios de cómputo generalmente tienen un número de ordenadores de acuerdo con la cantidad de alumnos que se tengan en la institución educativa; estos ordenadores se utilizan para realizar prácticas, investigar sobre temas novedosos y otras tareas, con la finalidad de aumentar su aprendizaje. Es importante que los ordenadores se utilicen para mejorar el conocimiento y no con fines de juego. En algunos casos, para

evitar el mal uso de los ordenadores del laboratorio de cómputo, se instalan aplicaciones para monitorear su uso y generar un ambiente propicio para el aumento de las habilidades de cada alumno (Zhixin, 2013).

### **Laboratorios de computación virtuales**

En el transcurso de los años, la tecnología ha tenido diversos cambios; la educación se ha reforzado con sistemas inteligentes mediante el uso de tutoriales sofisticados. En la actualidad se han implementado aulas interactivas y laboratorios virtuales para poder ampliar los horizontes de la enseñanza. Estos laboratorios integran la tecnología con los medios didácticos impartidos en las aulas de clases; de esta forma se estimula la interacción con los dispositivos y sistemas de aprendizaje (Burd, Gaillard, Rooney y Seazzu, 2011).

Los laboratorios virtuales pueden ser utilizados para el intercambio de información mediante el internet, reduciendo el costo de los dispositivos y aumentando la disponibilidad de la red. El propósito principal para lo que se están desarrollados los laboratorios virtuales es para ampliar el aprendizaje en el área de la educación (Krbeček y Schauer, 2015).

Esta tecnología se ha implementado en diversas universidades del mundo; por ejemplo, en la Universidad de Mongolia de Ciencia y Tecnología (MUST), se aceptó un proyecto para crear e implementar los laboratorios virtuales para la enseñanza clásica y un espacio virtual de aprendizaje a distancia. De esta manera, es posible proporcionar un ambiente apto para la enseñanza moderna (Tudevtagva, Ayush y Baatar, 2014).

Con la finalidad de mejorar el aprendizaje de los estudiantes, se han implementado laboratorios virtuales equipados con la más alta tecnología que brinda una alta estabilidad

entre dispositivos de red y ordenadores para facilitar las prácticas de lo aprendido en la teoría. Un laboratorio emplea recursos físicos para que los alumnos interactúen y aprendan sobre su implementación (Hwang, Kongcharoen y Guinea, 2014).

### **Virtualización**

En este proceso, se utilizan diversas características del sistema físico para visualizar un modelo virtual y aplicarlo a ordenadores, redes, sistemas operativos, entre otras aplicaciones. Las empresas afrontan un reto al tratar de optimizar las características de los servidores x86, ya que están desarrollados para trabajar con un solo sistema operativo. Por este motivo la virtualización se está convirtiendo en un alternativa viable en el desarrollo de proyectos y para simplificar exceso de procesos (Velásquez, 2009).

La noción de virtualización puede ser utilizado para proporcionar una capa de software en la máquina o servidor físico para implementar una máquina virtual. Sin embargo, existen diversos tipos de máquinas virtuales que tienen la finalidad de ofrecer al usuario el acceso a un entorno de trabajo virtual. Las máquinas virtuales pueden lograr simular entornos de sistemas operativos y lograr optimizar las aplicaciones instaladas en el servidor u ordenador principal. Al virtualizar un sistema se puede visualizar una interfaz similar al del dispositivo o servidor físico (González, 2010).

Lo primordial de la virtualización es simular el hardware y software de un sistema operativo, para obtener múltiples sistemas trabajando como uno. Esto se puede lograr al implementar un gestor de máquina virtual (VMM) capaz de crear una interacción entre el servidor físico y el cliente. En la Figura 1 se muestra la administración de los diversos elementos del hardware y la forma en la que se encuentran distribuidos para lograr la virtualización del sistema (Sotaminga Reyes, Guerrero Vallarezo y Abad Eras, 2011).



*Figura 1.* Administración de hardware (Baigorria, 2017).

La virtualización permite agrupar todos los recursos necesarios con la finalidad de aumentar el rendimiento del hardware y reducir el tiempo de procesamiento. Sin embargo, es imprescindible gestionar la ejecución del sistema operativo y las aplicaciones que se lanzarán. Dicha gestión la lleva a cabo el hypervisor, totalmente capaz de administrar los recursos físicos y el de las máquinas virtuales con el propósito de aprovechar el hardware (Rosero Vinuesa, 2012).

### **Máquina virtual**

Un ordenador o servidor físico no puede diferenciar entre una máquina física o una máquina virtual. Las principales características de una máquina virtual son aislamiento, compatibilidad, encapsulamiento e independencia del hardware. Una máquina virtual puede albergar un sistema operativo y las aplicaciones necesarias para su función. Pueden permanecer aisladas entre sí y reunir un grupo entero de recursos virtuales

para lograr una independencia del hardware físico (Márquez, 2011).

## **Hypervisor**

De acuerdo con Revelle (2011) y Villar Fernández y Gómez López (2010), la tarea principal de los hypervisores es el de administrar diversos sistemas operativos mediante algoritmos. Trabaja utilizando poco uso de recursos físicos y supervisa toda la máquina virtual. Por lo general inicia junto con el sistema operativo principal y permite la interacción entre las diversas aplicaciones en una sola máquina. Existen dos tipos de hypervisores para administrar las máquinas virtuales. En la Figura 2 se muestra un ejemplo del hypervisor llamado bare metal (sobre el metal desnudo), native o unhosted, el cual se inicia sobre el hardware.

La Figura 3 muestra el hypervisor de tipo hosted, llamado así porque requiere tener instalado previamente el sistema operativo. Este último tiene una mayor compatibilidad con el sistema al permitir gestionar todos los drivers directamente.

## **Tipos de virtualización**

### **Virtualización de plataforma**

Implica la presentación de una máquina virtual en el hardware, que a su vez se encuentra montada en el software del sistema invitado. Se crea un entorno de trabajo similar al original, permitiendo el acceso a las aplicaciones necesarias para su control. Se pueden poseer múltiples máquinas virtuales corriendo en el sistema de una máquina real (Aragundi Lucas, Cedeño Zamora, Quiroz Cedeño y Zambrano Veliz, 2012).

González (2010) señala a la consolidación de servidores como otra manera de llamar a la virtualización de servidores y hacer uso compartido de aplicaciones instaladas

en el sistema real. La virtualización de plataforma se puede lograr a través de sistemas operativos invitados, emulación de hardware, virtualización completa, paravirtualización, virtualización a nivel de sistema operativo y virtualización a nivel de kernel. Estos seis modelos se explican brevemente para conocer los beneficios de su implementación.



Figura 2. Hypervisor bare metal (López Medina, 2010).

### **Sistemas operativos invitados**

Este modelo no hace uso de un hypervisor para su implementación; trabaja directamente en el sistema real. Permite trabajar con sistemas operativos independientes bajo el control de una máquina virtual. Ejemplos representativos de lo anterior son VirtualBox, vmwareWorkstation 12 Pro, Parallels desktop y Windows Virtual PC.



Figura 3. Hypervisor hosted (López Medina, 2010).

## Emuladores

Tienen como principal característica duplicar completamente el hardware del sistema real; aceptan la ejecución de sistemas y herramientas diferentes a las instaladas en el host. Entre algunos ejemplos se encuentran Bochs x86 PC emulator, Windows Virtual PC, VirtualBox, Multiple Arcade Machine Emulator (MAME), DOSBox, Hercules, Multi Emulator Super System (MESS) y QEMU.

## Virtualización completa

En este modelo el hipervisor es el encargado de gestionar las aplicaciones entre el sistema real y el invitado. Permiten llevar a cabo la ejecución de diversos sistemas operativos en el hardware inmediato y mejorar su rendimiento. A modo de ejemplo, se pueden utilizar Vmware server, XenServer, Oracle Virtual Machine, Sun xVM Server, Vmware vSphere, Hyper-V y Proxmox.



## **Paravirtualización**

Es similar a la virtualización completa ya que, de igual forma, el hypervisor gestiona el enlace entre el sistema real y el invitado. Es importante mencionar que no emula el hardware desde el hypervisor y es capaz de realizar modificaciones en el sistema invitado. Como muestra, se pueden utilizar Logical Domains (LDom), Oracle Virtual Machine, Vmware Server y Sun xVM Server.

## **Virtualización a nivel del sistema operativo**

Este modelo se basa en un servidor virtual. Es imprescindible resaltar la diferencia entre una máquina virtual y un servidor virtual; el primero aloja un sistema operativo y el segundo ofrece una arquitectura compartida con el sistema real. Algunos ejemplos de este modelo son Linux-Vserver, Virtuozzo, Free VPS, Oracle Solaris Zones y OpenVZ.

## **Virtualización a nivel de kernel**

Al no utilizar el hypervisor para ser mediador entre el sistema real y el invitado, transforma el núcleo de Linux mediante un módulo capaz de crear máquinas virtuales disponibles para ejecutar procesos desde el área del usuario. De esta forma todos los sistemas operativos y aplicaciones corren sobre la máquina real sin utilizar virtualización por hardware. Como ejemplo se pueden mencionar Kernel-based Virtual Machine (KVM) y User Mode Linux.

## Virtualización de recursos

Sotaminga Reyes et al. (2011) afirman que, para esta versión, lo que se virtualiza son los recursos. Generalmente, todos los asociados con el hardware del sistema real

pueden ser tarjetas de red, memoria, disco duro, etc. A continuación se muestra una clasificación de los diferentes tipos de virtualización de recursos, su aplicación y sus variantes más destacables.

### **Encapsulación**

Es encubrir implementaciones de recursos complicados por medio de una interface sencilla. Se puede utilizar en discos duros, módems, dispositivos inteligentes, entre otros; por ejemplo, en Redundant Array of Independent Disks (RAID), Logical Volume Manager (LVM), Network-Attached Storage (NAS) y en la virtualización de red mediante Vmware NSX.

### **Memoria virtual**

En la actualidad, se encuentra incluido en la mayor parte de los sistemas operativos. Su finalidad es hacer creer al sistema real que existe espacio utilizable en la memoria, cuando en verdad se está empleando el espacio secundario. Algunos ejemplos son el espacio de intercambio o swapping en Linux y la paginación de memoria.

### **Virtualización de almacenamiento**

En términos generales, se refiere a la abstracción del almacenamiento real, de modo que se muestra un almacén intermedio que procede como interface de usuario para el almacenamiento real. Al deslocalizar el almacenamiento de información, es posible confundir al sistema real, ya que al tener almacenes conectados a la red pueden dar aspecto de trabajar localmente. Ejemplos característicos son el Network Attached Storage (NAS), el Storage Area Network (SAN) y el Internet Small Computer System Interface (iSCSI).

## **Virtualización de red**

Se fundamenta en la creación de zonas de red virtuales dentro de las reales, mediante el uso de abstracción en la red original. Permite que exista intercambio de información en redes compartidas o públicas, como si se tratara de una red privada. Por ejemplo, los dispositivos de subneteo propio de las redes TCP/IP, Virtual Private Network (VPN) y OpenSwarm.

## **Unión de interfaces de red**

Combina enlaces de red configurados por zonas y proporciona abstracción de recursos en una red virtual. Este modelo se compone por tarjetas de interface de red virtual, NIC virtuales (VNIC), virtual host bus adapter (vHBA) y Oracle Solaris.

## **Virtualizaciones de hardware**

Relacionado con la abstracción de los dispositivos de entrada/salida y memoria, al realizar la unión de ambos recursos, se logra virtualizar los sistemas en red. Como ejemplo, surgen Xsigo Systems, Cisco Unified Computing System (UCS) y Brocade VDX.

### Virtualización de aplicaciones

De acuerdo con Fernández Romero y García Pombo (2011), este modelo aumenta la particularidad de las aplicaciones para ejecutarse en múltiples plataformas y facilita la administración para obtener una mejor compatibilidad. Esto se logra mediante la encapsulación de las aplicaciones en el sistema que gestiona su inicio. Sin embargo, es fundamental tener un software que funcione como intermediario entre la capa de virtualización y el sistema operativo. De esta manera se interrumpen las operaciones realizadas por las aplicaciones, para permitir que el software administre todas las solicitudes.

Se pueden clasificar en virtualización limitada y completa. A manera de ejemplo, se pueden utilizar Wine y Java Virtual Machine (JVM).

### **Virtualización limitada**

Se refiere a las aplicaciones que son alteradas para evitar en lo más mínimo su conexión con el sistema operativo. Es ordinario entre las aplicaciones que en general requieren integrarse con el sistema y, al aplicar este tipo de virtualización, es posible lograr su dependencia, mediante el uso de un medio de almacenamiento que puede ser extraído del ordenador sin apagarse.

### **Virtualización completa**

Es necesario tener el software que funcione como gestor entre la aplicación virtualizada y el hardware del sistema real. Esto se puede llevar a cabo mediante portabilidad multiplataforma y simulación. El primer concepto admite la instalación de aplicaciones en múltiples entornos operativos; originalmente estas aplicaciones están desarrolladas para uso exclusivo de un sistema operativo. Se pueden utilizar JVM y VMware ThinApp, entre otros. El segundo concepto permite emular un programa mediante la abstracción de la interface de programación en aplicaciones o API (Application Program Interfaces, por sus siglas en inglés). Para lograr la emulación se pueden utilizar Wine, CrossOver y Zebra.

### Virtualización de escritorio

Conforme a Villar Fernández y Gómez López (2010), consiste en separar el entorno de escritorio del hardware y el software en el que se ejecuta. Se basa principalmente en la arquitectura cliente-servidor. Un cliente manda una petición al proveedor de

servicios o servidor y este le responde. Mediante el uso de máquinas virtuales que se guardan en el servidor principal, se obtiene un escritorio virtual dotado de todas las aplicaciones y programas necesarios para su ejecución. Los clientes que pretenden acceder al escritorio virtual, deben estar conectados en red con el servidor principal.

Todos los procesos ocurren directamente en el servidor, permitiendo al cliente disfrutar un entorno virtual con todos los beneficios de un ordenador físico. La virtualización de escritorio utiliza gestores ligeros para la interacción con el escritorio virtual; de esta forma se garantiza compartir recursos en red y la optimización de toda la infraestructura de cómputo. Algunas de los beneficios que se obtienen al implementar este modelo son los siguientes: gestión más eficiente de la seguridad al centralizar todas las políticas de seguridad en el servidor principal, reducción del coste de soporte al guardar todos los datos de los clientes, disminuir la inversión en el hardware, realizar copias de seguridad de todos los clientes almacenados en el servidor y el beneficio de poder utilizar múltiples escritorios. Entre los ejemplos de soluciones para implementar virtualización de escritorio sobresalen eyeOS, Dell Wyse, Vmware view, Oracle Virtual Desktop Infrastructure (VDI), XenDesktop VDI y ThinLinc de Cendio.

### **Ventajas y desventajas de la virtualización**

Es necesario conocer las ventajas vinculadas a la virtualización, el manejo que se hace al utilizar las máquinas virtuales en los diferentes entornos y el desempeño que se obtiene en la implementación en una red de máquinas virtuales y servidores reales (Ali y Meghanathan, 2011).

Martín et al. (2011) destacan múltiples ventajas de implementar la virtualización en un entorno de servidores; algunas de ellas se mencionan a continuación, junto con

una breve descripción:

1. Ahorro energético. Se puede reducir significativamente el uso de energía en los centros de procesamiento de datos (CPD), al consolidar múltiples servidores reales en uno.

2. Ahorro de espacio. Al administrar los espacios en un solo servidor real, es posible sustituir los CPD antiguos por racks compactos.

3. Mantenimiento de servidores. Es posible realizar una planificación a largo plazo con nuevos servidores, que más adelante podrán ser utilizados de forma eficiente para la optimización de hardware y reducción de equipo.

4. Facilidad de gestión. Al disponer de un centro virtual, es posible administrar todos los entornos desde el servidor central, ver las alertas y estar atento ante posibles fallos.

5. Transportable. Permite migrar una máquina virtual (VM) desde un host con destino a otro sin perder servicios, mediante tecnología apta para este proceso.

6. Recuperación del entorno. Otorga facilidad para recuperar entornos físicos en cuestión de minutos/horas.

7. Reconocimiento de múltiples entornos. Admite diferentes sistemas operativos virtualizados sin excesivas exclusiones.

8. Escritorio virtualizado. Posibilita a los usuarios gestionar escritorios locales o remotos mediante la tecnología de Virtual Desktop Infrastructure (VDI).

De acuerdo con Gimeno Martínez (2008), existen algunos inconvenientes al implementar la virtualización en general. En seguida se enlistan determinados conceptos detallados.

1. Descenso del rendimiento. Al tener virtualizado un sistema real, existe la posibilidad de que se reduzca su rendimiento, ya que no está instalado directamente en el hardware del servidor real. Sin embargo, no todas las opciones de virtualización generan este tipo de inconvenientes.

2. Incompatibilidad de hardware. No es posible el uso de hardware que no esté administrado y que sea compatible con el sistema real. Esto ocurre cuando se le implantan a la máquina virtual dispositivos que no podrá modificar.

3. Aceleración de video. Carece de aceleración de video por hardware, tal como el 3D. Sin embargo, algunas de las nuevas tecnologías ya permiten esta característica.

4. Hardware del servidor. Existe el peligro constante de que, si se cae el servidor real, las VM que se encuentran alojadas se ven afectadas. Ante posibles caídas de servicio, es importante implementar soluciones de disponibilidad o de cloud computing.

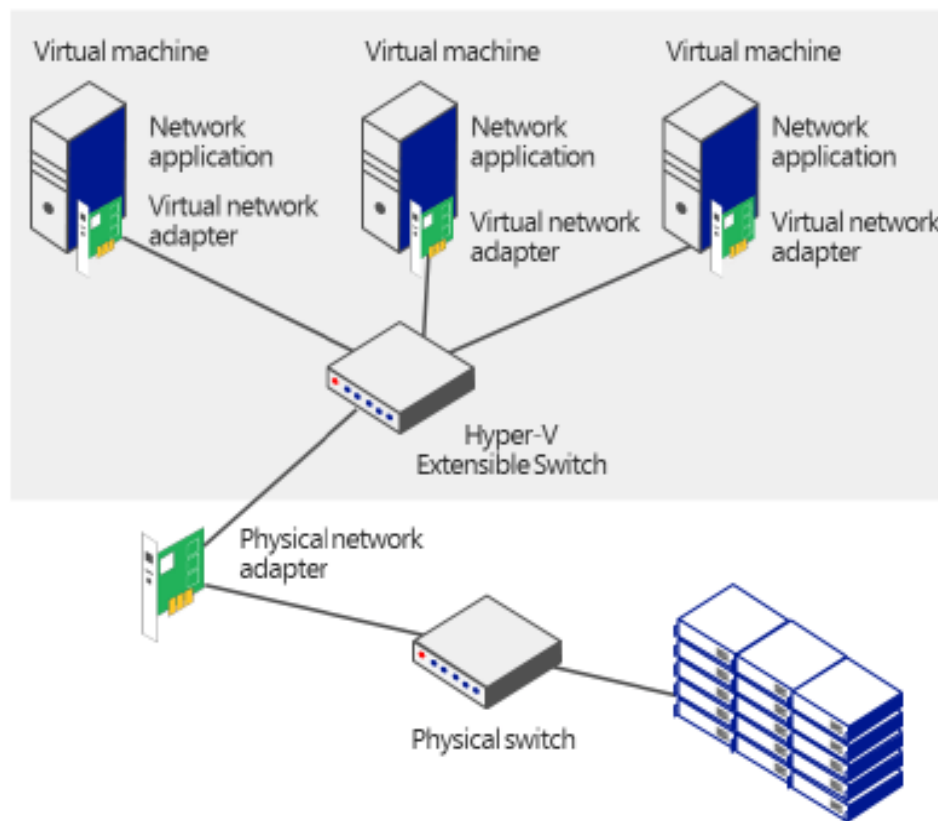
### **Herramientas de virtualización para un servidor**

#### Windows Server 2012 Hyper-V

De acuerdo con Microsoft (2012), la virtualización con este tipo de herramientas ha generado nuevas posibilidades para gestionar diversos centros de datos. Al implementar esta plataforma es posible reducir costos de producción, optimizar el hardware del servidor y crear máquinas virtuales multipropósito independientes. Entre los sistemas operativos soportados se encuentran Windows y Linux. Algunas de sus ventajas son el nivel de aislamiento de las cargas de trabajo y la flexibilidad en su infraestructura al permitir deslocalizar los servidores según sea la necesidad. Incorpora la tecnología de VLANs con Hyper-V, virtualización de red para colocar máquinas virtuales en los nodos

necesarios, migración e importación de máquinas virtuales y reducción de carga administrativa.

Al tener la capacidad de adaptarse de acuerdo con los cambios en el hardware y los dispositivos de almacenamiento, es posible lograr un amplio rendimiento de las máquinas virtuales. Esta herramienta permite poseer alta escalabilidad en el servidor y un entorno disponible ante posibles fallos, redundancia de datos y soporte para los clientes virtuales. Entre sus características destaca el conmutador de red virtual Hyper-V Extensible Switch (ver Figura 4).



*Figura 4.* Plataforma de virtualización Windows server 2012 con Hyper-V (Noguiera, 2017).



El switch permite la interconexión de máquinas virtuales a la red, ubicando en cada una de ellas las políticas de seguridad, aislamiento de las cargas de trabajo y especificaciones necesarias para su funcionamiento.

### Plataforma VMware vSphere 6

Según VMware (2016), esta es una plataforma de virtualización que permite construir infraestructuras de servidores estables; mediante la consolidación de servidores es posible tener 15 o más máquinas virtuales en un servidor real. El software vSphere está configurado para disminuir la pérdida de datos y la sobrecarga de operaciones. Esta herramienta utiliza el hypervisor vSphere ESXi para permitir la administración de recursos entre diversas máquinas virtuales; incluye máquinas de hasta 32 núcleos por CPU (unidad central de procesamiento), un TeraByte de ram, interfaces USB 3.0, almacenamiento compartido mediante canal de fibra, automatización de la administración de energía, implementación de switches virtuales capaces de distribuir múltiples entornos, migración de máquinas virtuales entre diversos tipos de servidores, tolerancia a fallos efectiva al permitir reiniciar aplicaciones si existe un fallo de hardware, posibilita el respaldo para realizar las restauraciones de sistema cuando se necesite, ofrece gestionar la memoria y el uso de CPU en cada una de las máquinas virtuales. La arquitectura del VMware vSphere (ver Figura 5) está constituida por la tolerancia a fallos mediante la creación de una copia exacta de la máquina virtual; de esta manera, si se cae un servidor físico, es posible levantar otra máquina virtual desde otro host para evitar posibles fallos, pérdida de datos y tiempos de inactividad.

El resultado se basa en crear máquinas virtuales protegidas primarias y secundarias, cada una implementada en servidores con hypervisor ESXi diferentes. Cuando un

servidor real falla, una de las copias se levanta junto con todas las conexiones de red, asegurando la migración exitosa de las máquinas virtuales.

### VMware workstation

De acuerdo con Bugnion, Devine, Rosenblum, Sugerman y Wang (2012), se ejecuta bajo el conjunto de instrucciones x86 y permite insertar máquinas virtuales en el sistema real. Su arquitectura está basada principalmente en la encapsulación independiente del hardware de las máquinas virtuales. Al optimizar la arquitectura x86, VMware workstation puede soportar múltiples sistemas operativos y permitir la abstracción de las máquinas virtuales con la finalidad de que éstas se ejecuten en instancias independientes del sistema real.

El modelo de VMware permite la emulación del hardware con el mismo procesador del sistema real, Bios, tarjeta gráfica, memoria Ram, etc. Virtualizar mediante el software ayuda en la optimización de los recursos y proporciona un entorno de trabajo similar al de un ordenador real. Su funcionamiento se detalla en la Figura 6.

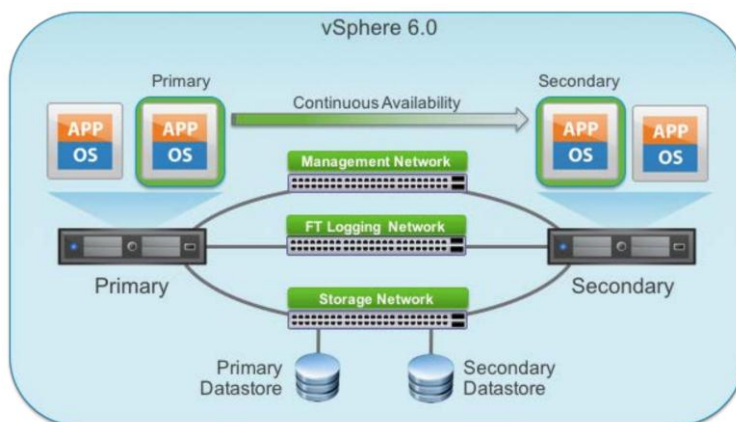


Figura 5. Plataforma vSphere 6.0 (VMware, 2016).

Algunas de sus características sobresalientes son el soporte de procesadores a 32 y 64 bits, creación de máquinas virtuales con nivel de procesamiento a cuatro núcleos, expansión de discos virtuales, gestión de carpetas compartidas, capturas de pantalla de las actividades realizadas en la máquina virtual y acceso a la red (López Medina, 2010).

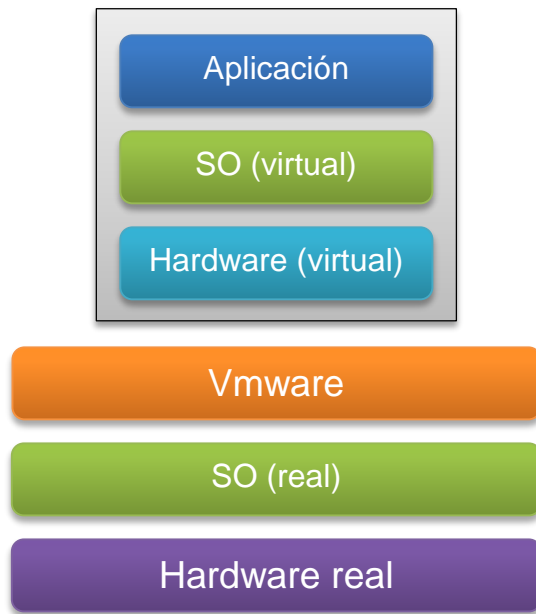


Figura 6. Arquitectura básica de VMware workstation (Periáñez Gómez, 2017).

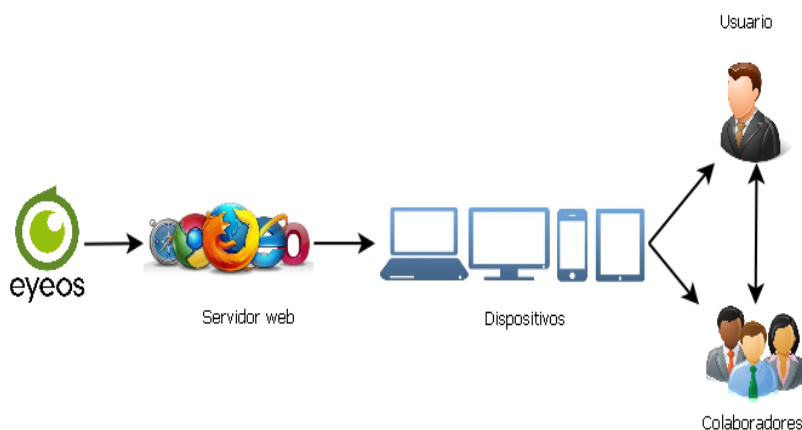
## EyeOS

De acuerdo con Ebberts et al. (2010), se trata de una plataforma de nube privada dedicada para aplicaciones web; permite la gestión de documentos y aplicaciones desde cualquier dispositivo conectado a internet. Integra el sistema real a la web, dando como resultado el acceso a las aplicaciones y servicios alojadas en el escritorio personal. Su estructura está basada en la de cliente-servidor, en donde el cliente es el navegador

web. La plataforma eyeOS permite la instalación en un servidor privado; de esta manera, crea un entorno de fácil administración. El servidor privado debe tener previamente instalado Apache y PHP; al tener ambos, se garantiza un control más organizado de todas las aplicaciones y acceso e interacción con todos los usuarios en todo momento (ver Figura 7).

Los pilares de eyeOs son Apache, PHP, AJAX y DHTML. También acepta la unificación de servicios por medio de su kernel, funciones de bajo y alto nivel, entre otras. Contiene MMAP (Message Mapping) para gestionar los mensajes entre el cliente y la aplicación, VFS como sistema de seguridad, eyeX para administrar los mensajes en XML del navegador, Proc en la gestión de procesos, entre algunos servicios.

Esta plataforma (eyeOS) mantiene la facilidad de uso en el escritorio y es posible utilizarlo de forma local en el servidor privado; de esta manera, el usuario puede tener acceso sin depender de la conexión a internet, permite disponer de un amplio número de aplicaciones y la gestión de archivos entre usuarios (Norte y Fiestas, 2015).



*Figura 7.* Arquitectura eyeOS (Jiménez Ortega, Fernández Valverde y López Pino, 2017).

## Sistemas embebidos

De acuerdo con Patwardhan, Nair y Paranjape (2014), un sistema embebido (SE) es un dispositivo que ayuda en la gestión de múltiples equipos, el término “incrustado o embedded” se refiere a que es parte esencial de un sistema. En la Figura 8 se muestra un diagrama básico sobre un sistema embebido.

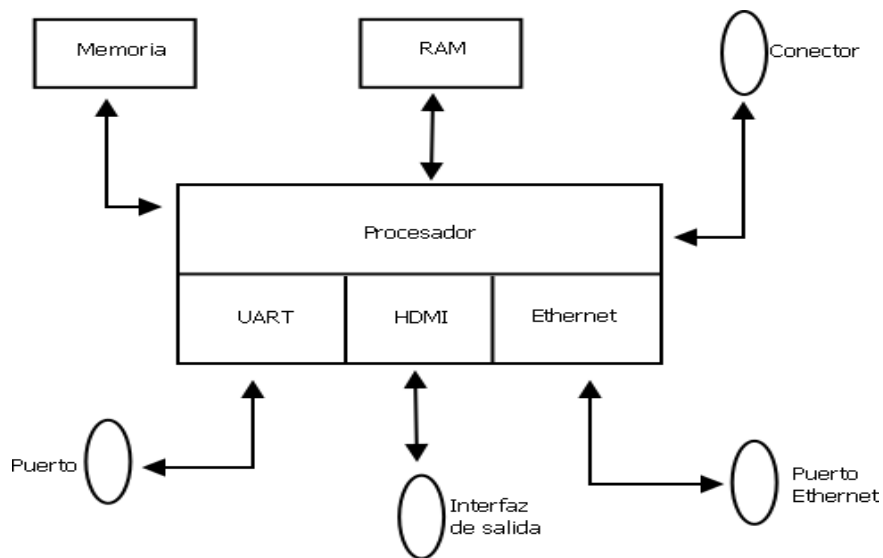


Figura 8. Diagrama de bloque de un sistema embebido (López, 2017).

Los sistemas embebidos (SE) ofrecen ejecución de aplicaciones rápidas y están creados para optimizar energía y tiempo de ejecución. El SE está diseñado para aumentar la eficiencia de los procesos en tiempo real; si una respuesta llega tarde, entonces está equivocada. Al ser un sistema reactivo, el SE se mantiene en constante interacción con el entorno haciendo uso de sensores y actuadores. Algunos sistemas embebidos carecen de interface de usuario; sin embargo, es posible instalar una para obtener la

interacción y programación del SE. Una de las aplicaciones de un SE está en la robótica. Al trabajar en un robot autónomo con ruedas, es posible detectar áreas completas por medio de sensores para evitar obstáculos y obtener movimiento inteligente. Esta clase de robot lo forman sistemas embebidos reutilizables. La aplicación que tiene en el campo de la robótica es muy extenso y es ideal para desarrollar habilidades en los estudiantes.

Los sistemas embebidos también se encuentran en la implementación de protocolos de nueva generación. El objetivo principal es la prueba y demostración de conexiones por medio de IPv6. Sin embargo, en todo el proceso es imprescindible comprender cada uno de los requisitos necesarios para su implementación en un servidor web. Para llevar a cabo esta prueba, se utilizó un microcontrolador para la conexión con Ethernet, con FreeRTOS para su gestión y con un stack de protocolos. Al realizar la implementación, se obtuvo que es necesario el uso de un algoritmo extra para la gestión de la capa de enlace y considerar el manejo apropiado del microcontrolador. De esta manera, se puede observar con seguridad el desempeño de las capas y el hardware necesario para su implementación (López Sarmiento y Garzón Romero, 2013).

A lo largo del tiempo, se ha ido incursionando en nuevas tecnologías con el propósito de facilitar la vida diaria. La comunicación en redes por medio de sistemas embebidos está desarrollada para tener el control de acceso entre el medio inalámbrico y el enrutamiento. Es posible mejorar el rendimiento en redes de sensores, VoIp, entre otros. De esta manera, se utiliza el SE para gestionar los diferentes entornos y reducir el consumo de energía (Friedrich et al., 2013).

El desarrollo de tecnologías como los sistemas embebidos permite el desarrollo y aplicación en diversos entornos, tanto en el software como en el hardware. Sin embargo,

es imprescindible el desarrollo de proyectos para conocer los beneficios que se pueden lograr en diversas áreas. De esta manera, es posible analizar los resultados que se obtienen al optimizar recursos, sistemas en tiempo real y sistemas de hardware (Tosini et al., 2013).

Chouza, Erusalimsky, Lanzelotti y Rodríguez (2012) destacan la importancia de trabajar con sistemas embebidos al desarrollar un prototipo de UAV (vehículo aéreo no tripulado), llamado Quadpector por tener aterrizaje y despegue vertical mediante el uso de cuatro hélices en forma de cuadricóptero. Los resultados del proyecto demuestran el avance en la tecnología de sistemas embebidos, la facilidad en la manipulación de dispositivos pequeños, la optimización de los recursos y el bajo costo para el desarrollo.

Lizondo, Agüero, Uriz, Tulli y Gonzalez (2012) señalan otra forma de utilizar un SE. El proyecto está fundado en el reconocimiento automático por voz; este modelo aplica una máquina para realizar reconocimiento e identificación del usuario mediante una frase pronunciada. Al implementar el SE, es posible utilizarlo en entornos de electrónica, cerraduras de puerta, etc. El sistema está construido en un modelo de microchip con características de los microcontroladores en un núcleo de 16 bits de alto rendimiento.

Al emplear un SE en la agricultura, es posible lograr un beneficio en la producción. El proyecto está basado en la construcción de un invernadero robotizado; esto quiere decir, el uso de un robot móvil que recorre cada uno de los pasillos del invernadero. Para desarrollar el SE fue necesario contar con el hardware y software diseñados a medida para satisfacer cada una de las características necesarias. El resultado de las simulaciones servirá como base para futuros proyectos (Scasso, Garro, Ordinez y Alimenti, 2012).

## Plataformas de sistemas embebidos

### Raspberry Pi

De acuerdo con Salcedo Tovar (2015), se trata de un miniordenador de bajo costo con la capacidad de funcionar como herramienta de aprendizaje y para desarrollo de proyectos en menor escala. Desde sus inicios en el 2008, como versión beta hasta su lanzamiento oficial en el año 2012, el Raspberry Pi (RPi) ha sido uno de los ordenadores británicos más vendidos en poco tiempo. En la actualidad, existen tres modelos, cada uno con versiones diferentes de acuerdo con sus especificaciones. En la Figura 9 se muestra una placa básica de un RPi.

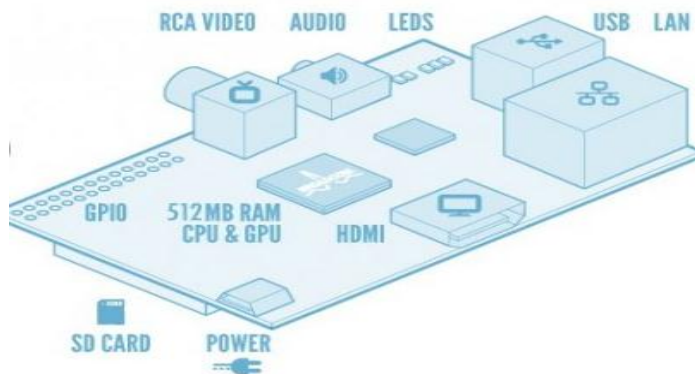


Figura 9. Diagrama básico de Raspberry Pi 1 modelo B (Sevilla Rodríguez, 2017).

El nuevo RPi 3 modelo B consta de una arquitectura ARMv8 a 1.2 GHz en cuatro núcleos a 64 bits, cuatro puertos USB, GPIO de 40 pines, puerto HDMI, puerto Ethernet, jack de 3.5 mm, interface CSI para cámara, interface DSI para display, Slot de Micro SD, VideoCore IV para gráficos en 3D, módulo Wireless 802.11n y Bluetooth 4.1, integrados a la placa base.



El sistema operativo recomendado para el RPi es el Raspbian, del que existen diversas versiones. Sin embargo, actualmente se encuentra disponible el Raspbian Jessie. Es posible instalar otros sistemas operativos con licencia de código libre en el RPi; de esta manera, se pueden adaptar para el uso específico que requiera la plataforma. Entre sus características principales destacan la facilidad de adaptarse para cualquier proyecto, el costo del hardware, la documentación abundante en foros, el soporte constante para los usuarios y la factibilidad para desarrollar proyectos educativos y experimentación, entre otros múltiples usos que se le puede dar. En la Figura 10 se puede observar el uso específico de cada uno de los módulos disponibles en el RPi 3 Modelo B.

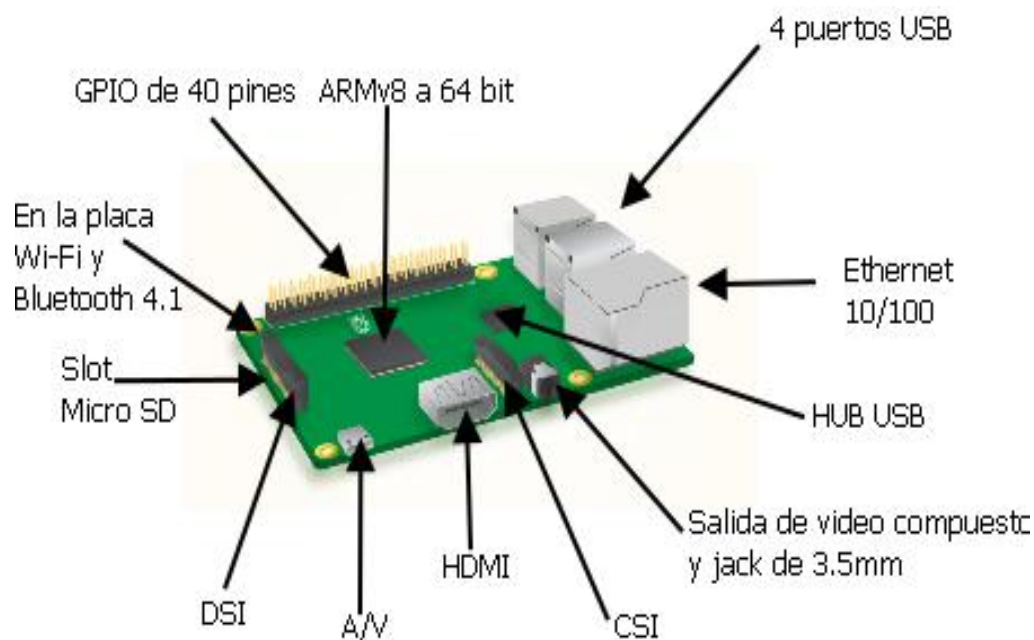


Figura 10. Componentes del Raspberry Pi 3 modelo B (Thomsen, 2017).

## Odroid-C2

Lee (2016) destaca las características principales de esta placa que puede ser utilizada para el desarrollo de videojuegos, trabajos de oficina, proyectos educativos, experimentación en laboratorios, creación de prototipos, como estación de trabajo, etc. Desde sus inicios, Odroid fue creada como una placa para desarrollar aplicaciones en android, con el propósito de ser apoyo en áreas de robótica, domótica, programación y cualquier uso que se le pudiera dar. Es un dispositivo que tiene incluida una placa de depuración para desarrolladores con múltiples plataformas de software libre para su experimentación. En la Figura 11 se muestran los componentes básicos de la placa Odroid.

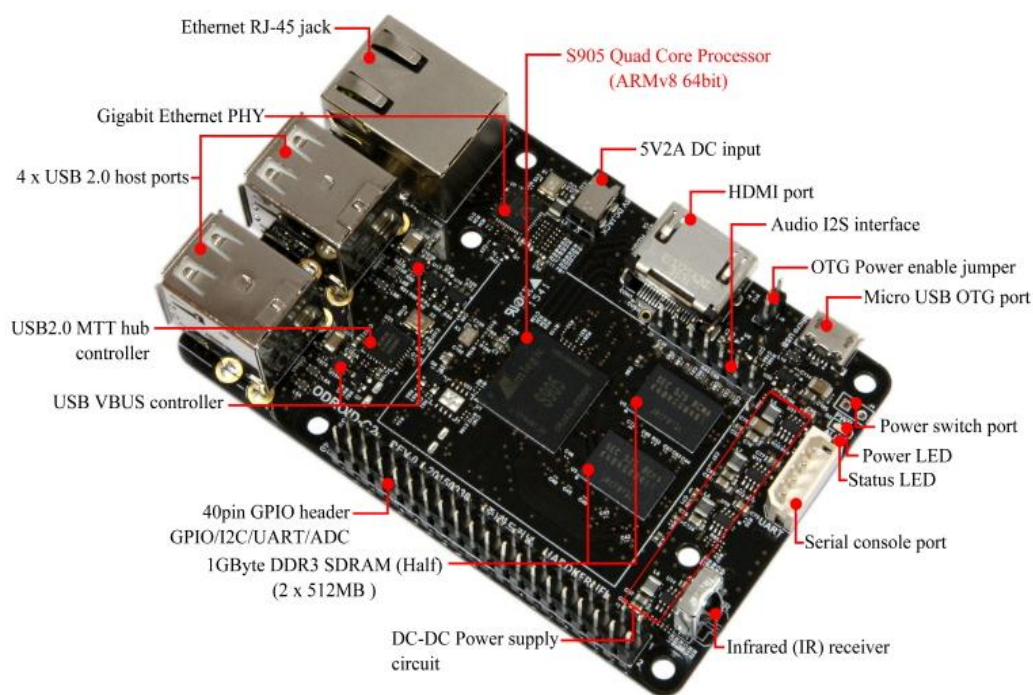


Figura 11. Componentes de placa Odroid C-2 (Roy y Bommakanti, 2017).

El OdroidC-2 posee CPU amlogic s905 de cuatro núcleos a 2GHz, 2GB de memoria RAM, GPU que puede soportar videos en Full HD hasta 4K, HDMI 2.0 4K/60 Hz, slot de memoria para eMMC, slot para tarjeta MicroSD, puerto Ethernet Gigabit, cuatro puertos USB, un puerto 2.0 OTG (On the Go), receptor infrarrojo, GPIO de 40+7 pines, interface de audio 12s y puerto serial para tener acceso a la consola de Linux. El sistema operativo que tiene preinstalado es el Android 5.1, con posibilidad de compatibilidad con Ubuntu 16.04 LTS en versiones para dispositivos con arquitectura ARM64, Arch Linux y Debian. Su procesador ARM permite obtener resultados con bajo consumo de energía para una mejor optimización del hardware. A pesar de la robustez en su familia de procesadores, Odroid mantiene una comunidad en desarrollo para mejorar su potencia y rendimiento.

### Nvidia Jetson TK1

De acuerdo con Nvidia (2015), esta es una plataforma apta para el desarrollo que integra lo necesario para funcionar como un superordenador móvil. Es útil en diversos entornos de experimentación; de esta manera, puede ser utilizada para crear prototipos en visión e inteligencia artificial, robótica, seguridad de redes, etc. Jetson TK1 (JTK1) permite el rendimiento significativo en la GPU (Unidad de procesamiento gráfico) y obtiene una mayor eficiencia del consumo energético para las diversas aplicaciones embebidas que se integren. En la Figura 12 se muestran los componentes de una placa JTK1.

La JTK1 está basada específicamente en el procesador de 192 núcleos de Nvidia Tegra móvil K1; su GPU se fundamenta en la Nvidia Kepler, destinado para el desarrollo de supercomputadores y sistemas de alto rendimiento. Integra una CPU (Unidad central

de procesamiento) ARM cortex-A15 de cuatro núcleos, motor de visualización en pantalla local con soporte para 4K, 2 GB de memoria RAM, almacenamiento externo por medio de una tarjeta Micro SD, entrada para periféricos, USB 3.0, puerto Gigabit ethernet, slot para adaptador PCIe, interface CSI, puerto serial para comunicaciones COM y un puerto SATA (Serial Advanced Technology Attachment) para la transferencia de datos.

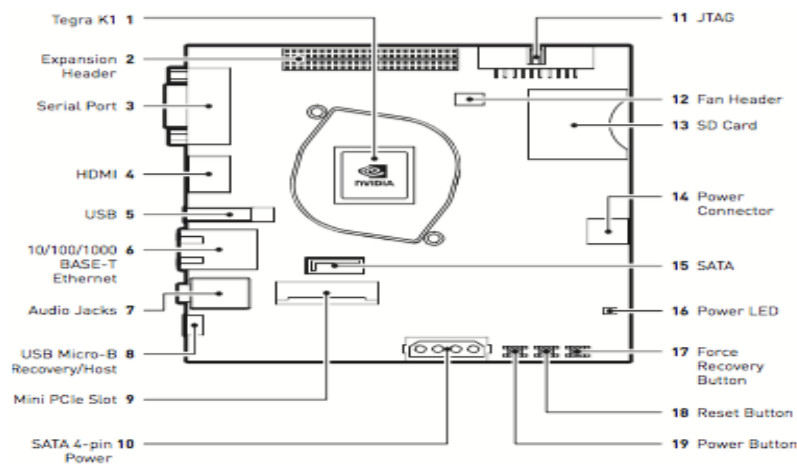


Figura 12. Componentes de una placa Jetson TK1 (Hildenbrand, 2017).

El software incluido en la JTK1 es el Ubuntu versión 14.04, pero pueden ejecutarse distribuciones de GNU/Linux. Los entornos donde se puede aplicar la JTK1 son amplios; en un futuro es posible que sean apoyo en la realidad aumentada, en la fotografía computacional y en sistemas de asistencia.

## Arduino

Enríquez Herrador (2009) destaca la placa Arduino como una herramienta que permite la interacción entre el ordenador y el mundo real por medio de un ordenador

externo. Es utilizada para desarrollar programas en código abierto, mediante la programación del microcontrolador en la placa base para su uso en robótica, creación de objetos interactivos, control de sensores, luces, motores, etc. Los diversos proyectos creados con Arduino pueden ser implementados para funcionar de forma autónoma o ser gestionados mediante un programa instalado en un ordenador. El lenguaje de Arduino se encuentra basado principalmente en wiring, una plataforma que se origina en el entorno de programación multimedia Processing. El software y el hardware funcionan en diversas plataformas de sistemas operativos y permiten la facilidad del trabajo sobre el microcontrolador. En la Figura 13 se muestran los componentes generales de la placa Arduino.

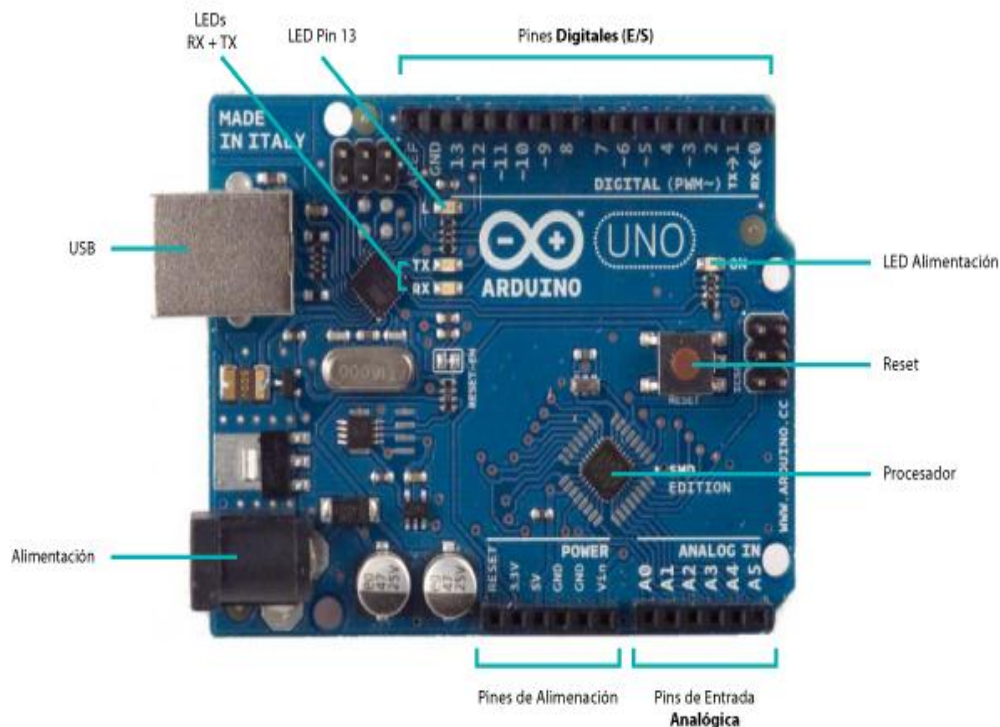


Figura 13. Componentes de placa Arduino (Fitzgerald y Shiloh, 2017).

El software de dicha placa, al ser de licencia libre, puede ser modificado mediante librerías de C++. De igual manera, los microcontroladores pueden ser diseñados a medida de acuerdo con las necesidades de los proyectos. La placa de Arduino incluye 14 entradas digitales que pueden ser configuradas de acuerdo con las necesidades como entradas y salidas; cada una opera con cinco voltios. Sin embargo, para obtener una mayor gestión sobre el dispositivo, es imprescindible instalar un entorno de desarrollo o, por sus siglas en inglés, un IDE (Integrated Development Environment). El IDE proporciona el ambiente necesario para desarrollar aplicaciones, cargarlas a la placa y ejecutarlas, todo esto desde el ordenador personal. Arduino ofrece usabilidad en una placa de bajo costo que permite a los usuarios desarrollar todo tipo de proyectos que sea posible imaginar.

### **Ventajas y desventajas de los sistemas embebidos**

Los sistemas embebidos tienen características específicas que han permitido a los usuarios tener el control sobre el hardware y el software para desarrollar proyectos multidisciplinarios. Al implementar una microarquitectura en el diseño del sistema, es posible obtener una amplia optimización de todos sus componentes (Ashford Lee y Arunkumar Seshia, 2011).

Lister (2016) señala algunas ventajas de los sistemas embebidos; algunas de ellas se describen a continuación.

1. Beneficios físicos. Al ser utilizado para un tarea específica, el sistema embebido (SE) no necesita cambios de hardware o software constantes; esto resulta en un dispositivo diseñado para ofrecer múltiples servicios al usuario.

2. Tareas dedicadas. Una tarea en el SE puede ser desarrollada sin interrupciones; esto permite un funcionamiento continuo y correcto.

3. Sistema operativo. Los requisitos del sistema operativo son menos costosos en comparación al de los ordenadores convencionales; un SE puede ejecutar su sistema sin demasiada complejidad y con el mínimo de actualizaciones.

4. Especificaciones y costo. El costo del hardware de un SE es relativamente bajo y una tarea puede ser ejecutada sin necesidad de utilizar mucho su capacidad de procesamiento. Dependiendo de la utilidad, el SE puede optimizar cada uno de los recursos necesarios para los proyectos.

Otras características de un SE son la simplicidad para realizar procesos, durabilidad en diversas condiciones ambientales, usabilidad, tamaño del hardware, eficiencia energética, confiabilidad en sus acciones y facilidad en reparaciones (Pearson, 2016).

### **Medios de transmisión**

Al obtener una conexión de red, un dispositivo puede intercambiar información a través de cable o mediante conexiones inalámbricas. De esta manera, es posible ampliar el rango de conexión al utilizar los protocolos de red correctos. Un SE puede mantenerse en comunicación directa con otro sistema mediante el uso de la red y cualquier ordenador integrado en ella puede ser utilizado para monitorear o controlar la ejecución de las solicitudes de los SE. Es necesario que las redes posean dispositivos físicos habilitados para el intercambio de información. De esta forma, se puede lograr que la información llegue íntegra hacia el dispositivo destino. De acuerdo con las necesidades de la red, los dispositivos físicos pueden integrarse mediante el uso de un medio de transmisión guiado y no guiado (Ruiz Molina, 2014).

## Medios de transmisión guiados y no guiados

De acuerdo con Tanenbaum y Wetherall (2012), es posible utilizar diferentes tipos de medios para transmitir datos de un dispositivo de red a otro. A continuación se describe brevemente cada uno de los componentes de los medios de transmisión guiados y no guiados.

1. Par trenzado. Es el medio utilizado con mayor frecuencia en diversos tipos de instalaciones, debido a su bajo costo y por las cualidades que posee; consta de hilos de cobre trenzados entre sí, rodeados por una capa aislante para evitar problemas de interferencia, atenuación, ruido, etc. El par trenzado puede ser STP (par trenzado blindado) y UTP (par trenzado no blindado); el primero está limitado a una distancia máxima de 90 con una impedancia de 120 a 150 Ohm. El segundo tiene una longitud máxima de 100 metros y es mayormente utilizado para realizar conexiones con una impedancia de 100 Ohm; el UTP y el STP pueden ser empleados para transmitir señales analógicas y digitales. Existen diversas categorías para el par trenzado; los más comunes son el Cat5e y el Cat6.

2. Coaxial. A diferencia del medio UTP cat5e, este medio contiene un mejor blindaje y ancho de banda mayor. De esta manera, es posible cubrir distancias más amplias a mayor velocidad. Generalmente, se utilizan dos tipos de cable coaxial: el de 50 Ohms y el de 75 Ohms. Los componentes de un medio coaxial es un alambre de cobre rígido en el centro, envuelto por una capa de material aislante que, a la vez, está constituido por una malla para evitar interferencias.

3. Líneas eléctricas. Normalmente estas líneas transportan energía a las casas y el cableado se encarga de distribuir la energía en cada una de las tomas de corriente. Al



utilizar las líneas eléctricas para montar una LAN (red de área local), es imprescindible conocer el diseño sobre la distribución de las señales eléctricas; estas señales se envían por el medio que atenúa las señales de frecuencia más elevadas. El cableado eléctrico se encarga de agrupar las señales externas y emitir las señales correspondientes.

4. Fibra óptica. Consiste en fibras hechas de vidrio que transmiten luz, las cuales son ideales para distancias largas en redes LAN de alta velocidad. Se compone de una fuente de luz y de un detector. El detector emite un pulso eléctrico, cuando el haz de luz ingresa en el medio desde un extremo a otro, y la señal eléctrica se convierte y transmite desde el receptor. Esto genera un sistema de transmisión unidireccional. Para evitar atenuación en el medio de transmisión, es imprescindible tener una longitud de onda correcta y un medio físico íntegro.

En la siguiente sección se hace un análisis breve sobre los conceptos básicos de un medio de transmisión no guiado o inalámbrico.

1. Espectro electromagnético. Se refiere a la distribución energética de un grupo de ondas electromagnéticas que emiten o absorben radiación del mismo elemento. Para difundir de manera eficiente los datos, es necesario tener conectado un dispositivo transmisor y receptor a un circuito eléctrico adecuado. De esta manera, el principio básico de la comunicación inalámbrica depende de la velocidad de la luz en el vacío, sin implicar la frecuencia.

2. Radiotransmisión. Utiliza como medio de transmisión ondas de radiofrecuencia, por lo cual es posible cubrir distancias largas entre edificios. Las ondas de radio se desplazan por todas las direcciones en su totalidad. De esta manera, los dispositivos de

transmisión y recepción no necesitan estar colocados en línea verdaderamente. Las ondas de radio necesitan de la frecuencia para conocer el impacto que puede generar la pérdida de trayectoria; esto se produce cuando existe un déficit en la frecuencia y ocasiona que las ondas de radio no lleguen a su destino íntegramente. Al transmitir en alta frecuencia, las ondas de radio se pueden perder debido a lluvias y otros impedimentos mayores, tales como interferencias eléctricas.

3. Microondas. Las microondas, a diferencia de las ondas de radio, no pueden atravesar edificios, debido a que realizan su comunicación en línea recta. Si las torres de transmisión se encuentran separadas por un rango muy grande, será necesario implementar repetidores en lugares específicos. Cuando hay divergencia en el espacio se genera un desvanecimiento por multitrayectorias; esto ocasiona que las ondas se desfasen y cancelen la señal. Para tratar de evitar estos problemas, los operadores utilizan espectros electromagnéticos más amplios y ondas de radio con frecuencias altas. Las microondas son utilizadas frecuentemente en telefonía, distribución de televisión, etc.

4. Infrarroja. Es utilizada regularmente para transmitir datos en distancias cortas; por ejemplo, para los controles remotos, estéreos, etc. Las ondas infrarrojas no atraviesan objetos sólidos; esto permite que no exista interferencia con otros dispositivos que utilizan ondas de radio para transmitir datos.

5. Onda de luz. Es comúnmente conocida como óptica de espacio libre, debido a que es posible conectar dos redes LAN entre dos edificios mediante el uso de láser instalado en las azoteas. La señalización suele ser unidireccional y en cada extremo debe poseer un fotodetector con láser. El haz de luz es estrecho y ofrece un ancho de banda

alto a un precio accesible. Sin embargo, al ser estrecho, es necesario tener precisión para apuntar a un objetivo que se encuentre a 500 metros de distancia.

6. Satélite. Dispositivo de comunicación capaz de repetir microondas a través del espacio; está equipado en su interior por transpondedores encargados de escuchar una parte del espectro, amplificar y retransmitir la señal en una frecuencia diferente a la entrante. Es posible realizar procesamiento de señales digitales para transmitir las hacia el dispositivo receptor, ya que este método permite mejorar el desempeño y disminución de ruido en la señal. De acuerdo con altitud, tiempo de retardo, posición, radio de la órbita terrestre y cantidad de satélites necesarios para cobertura global, existen satélites geoestacionarios de órbita terrestre media y de órbita terrestre baja.

## **CAPÍTULO III**

### **METODOLOGÍA**

#### **Características del proyecto**

En este capítulo se describe la metodología y los procesos empleados para la creación del prototipo de laboratorio de cómputo mediante el uso de virtualización de escritorio y tarjetas embebidas. Se presentan las diferentes fases, problemas y desafíos encontrados en el proceso, así como las soluciones aplicadas.

La placa del Raspberry Pi (Raspi) ha sido utilizada en proyectos que involucran lo que tiene que ver con educación, música, internet de las cosas, computación en la nube, medicina, videojuegos interactivos, miniordenadores, lectores de códigos de barra, etc. Se han encontrado diversas ventajas al poner en práctica esta tecnología para sistemas embebidos, tal como la reducción de costos en el consumo energético.

El propósito primordial del proyecto de investigación es el correcto desarrollo de un prototipo de laboratorio de cómputo, que involucre virtualización de escritorio y sistemas embebidos para lograr un modelo cliente-servidor. Al principio de este proyecto, se hizo un análisis sobre los diferentes sistemas embebidos y software de virtualización aptos para el prototipo. Sin embargo, como resultado de las limitaciones antes mencionadas, en el proyecto se utilizó un miniordenador Raspi 1 modelo B, que entre sus características principales se resalta un SoC Broadcom BCM2835, CPU ARM 1176JZFS a

700 MHz, GPU Videocore 4, 512 en SDRAM, salidas de video en HDMI y RCA, resolución a 1080p, salidas de audio HDMI y 3.5 mm, dos puertos USB 2.0, puerto Ethernet 10/100, almacenamiento integrado con ranura para SD, consumo energético de 700 mA (3.5 W) y su fuente de alimentación es mediante micro-USB de 5V.

Debido a la necesidad de velocidad de procesamiento, se optó por conseguir el prototipo más nuevo en el mercado. El Raspi 3 modelo B cuenta con un SoC Broadcom BCM2837, CPU ARMv8 de 1.2 GHz 64 bit quad-core, GPU Videocore IV, un GB en memoria SDRAM compartida con la GPU, cuatro puertos USB 2.0, Conector HDMI, RCA e interfaz DSI para panel LCD, audio por HDMI y 3.5 mm, almacenamiento integrado por MicroSD, consumo energético de 800 mA (4.0 W), conectividad mediante Ethernet 10/100, Wi-Fi 802.11n, Bluetooth 4.1 y fuente de alimentación vía MicroUSB de 5V. Estos dos modelos llamados clientes se utilizaron para hacer pruebas de conectividad con el ordenador utilizado como servidor.

Para el desarrollo del prototipo, se utilizaron los siguientes componentes tecnológicos: (a) miniordenador Raspberry Pi 1 modelo B y Raspberry Pi 3 modelo B, (b) monitor con entrada HDMI, (c) dispositivos periféricos de entrada, (d) PC de escritorio con Ubuntu Server 16.04 LTS 64-bits, (e) software de virtualización eyeOS versión 1.9.0.3 y (f) medio de transmisión guiado y no guiado.

### **Planificación**

El objetivo de este proyecto es que, por medio de virtualización de escritorio y sistemas embebidos, se logre obtener el prototipo de un laboratorio de cómputo, mediante el uso de un modelo cliente-servidor, donde el cliente obtenga acceso al servidor

desde un Raspberry Pi (Raspi), utilizando un navegador compatible con HTML5. Al principio una de las limitantes que se presentó fue al trabajar con el Raspi 1 modelo B, ya que su velocidad de procesamiento no es muy alta y fue necesario probar con diversas versiones del sistema operativo Debian. El otro problema era que el porcentaje del trabajo en el procesador aumentaba tan solamente al ingresar al explorador y realizar una búsqueda. Al ingresar al servidor mediante el explorador Epiphany, el procesador tuvo un aumento del 100% únicamente al abrir el explorador y, posteriormente, fue disminuyendo al normalizarse la carga.

Por este motivo, se utilizaron y ratificaron diversos sistemas operativos compatibles en la Raspberry Pi 1 modelo B y posteriormente, con la adquisición del Raspberry Pi 3 modelo B, con la finalidad de optimizar el procesador. Los sistemas operativos utilizados fueron el Raspbian jessie, Fluxbox y Ubuntu Mate 16.04. Al obtener una versión estable del procesador entre cliente y servidor, se hizo la instalación del servidor virtual eyeOS en la PC utilizada como servidor y se comprobó su funcionamiento óptimo. Se examinaron diversos escenarios, teniendo al cliente conectado por medios guiados y no guiados, para conocer su funcionamiento. En el capítulo IV se habla ampliamente sobre el modelo de fases para el desarrollo del prototipo.

## **CAPÍTULO IV**

### **RESULTADOS**

#### **Modelo de fases para el prototipo**

Para el desarrollo del prototipo, se propone un modelo de cuatro fases: (a) pruebas de sistemas operativos ligeros para optimizar el procesador del Raspi, (b) montar un servidor para la virtualización con eyeOS, (c) comunicación local cliente-servidor y (d) evaluaciones sobre uso de CPU, memoria, medio de transmisión y consumo energético.

#### **Primera fase: pruebas de sistemas operativos**

En la Figura 14 se muestra la fase inicial del proyecto; se instaló el sistema operativo Raspbian en la Raspi 1 para conocer su plataforma de funcionamiento.

Al instalar y analizar el escritorio del Debian jessie, se observa su facilidad de acceso y su contenido de herramientas básicas. En la Figura 15 se examina lo que pasa cuando se ingresa al explorador web. Se observó un cambio significativo en el uso del procesador mediante el marcador de porcentaje local, ubicado en la parte superior derecha del escritorio. Del 1% pasó al 100% con tan solo esta actividad. Al realizar una conexión a un servidor externo o tan simple como una búsqueda en la barra de dirección, se genera un cambio constante en el uso del procesador. Esto quiere decir que el hardware no es suficiente para soportar la carga de procesos en los que estará bajo demanda constante.

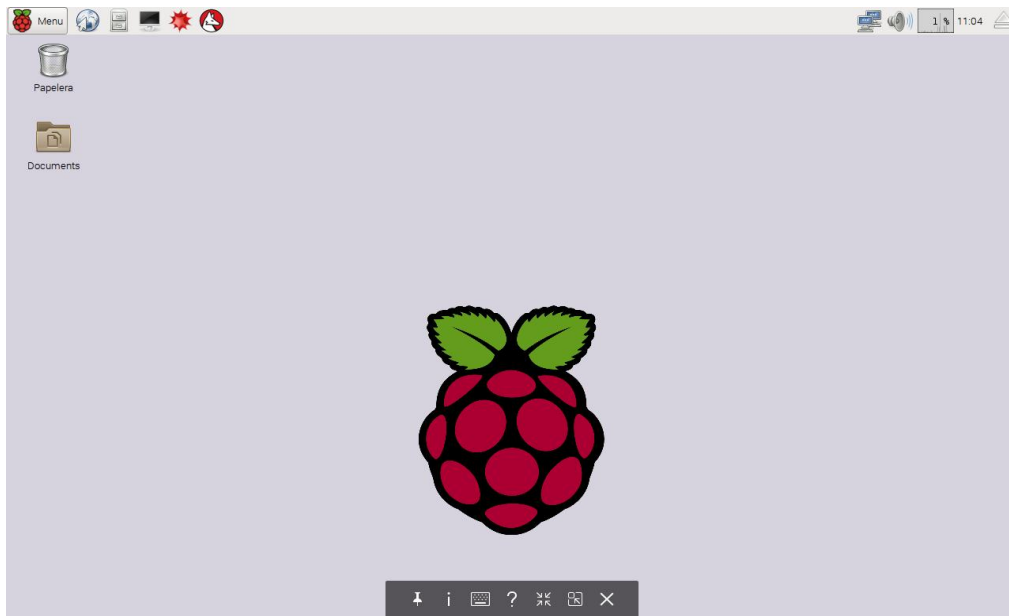


Figura 14. Escritorio Raspbian jessie en Raspberry Pi 1 modelo B.

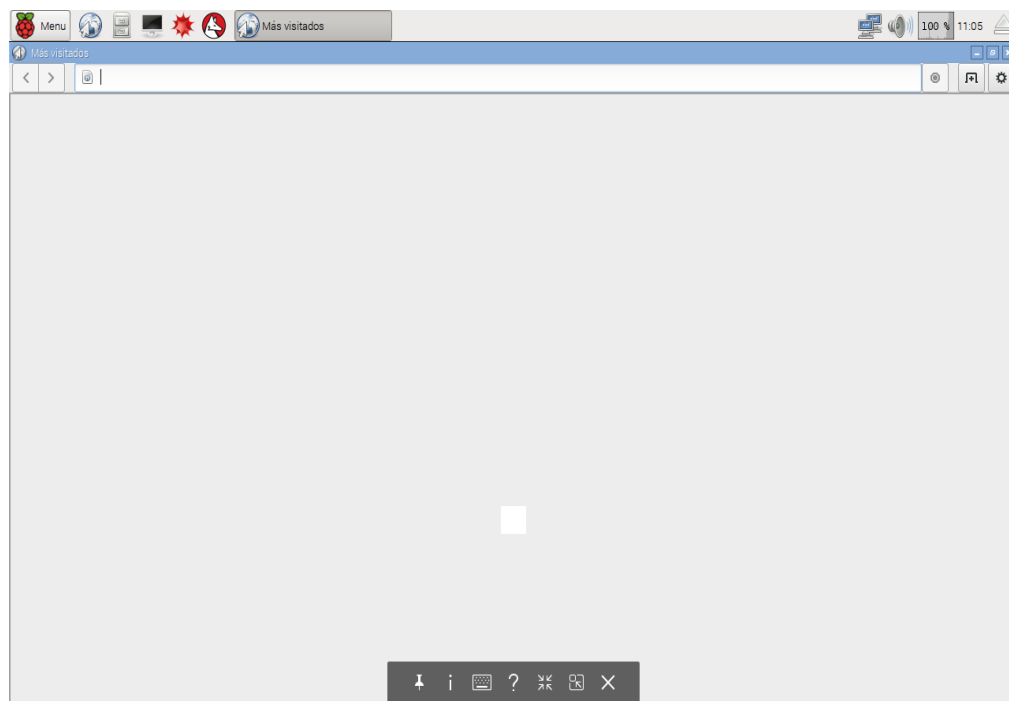


Figura 15. Barra de dirección y uso del procesador en porcentaje



Al examinar este problema con el hardware del Raspi 1 Modelo B, se exploró otra solución y se escogió utilizar jessie Lite, una versión a la cual es posible instalar desde línea de comando todos los complementos básicos que se necesiten para hacer más ligera la carga en el procesador. Se hizo un análisis sobre los sistemas aptos para esta versión de sistema operativo y se encontró que el Fluxbox era idóneo para los requerimientos. Sin embargo, como se muestra en la Figura 16, contiene una interfaz poco amigable; para ingresar al menú principal es imprescindible dar clic derecho con el mouse y seleccionar la aplicación que requiere el acceso. Se analizó el uso del procesador y se observó una disminución significativa del procesador; también se observó un inconveniente con la instalación del navegador web. Pese a que se instalaron las aplicaciones necesarias para el buen funcionamiento del hardware, el acceso no fue tan amigable como se esperaba.

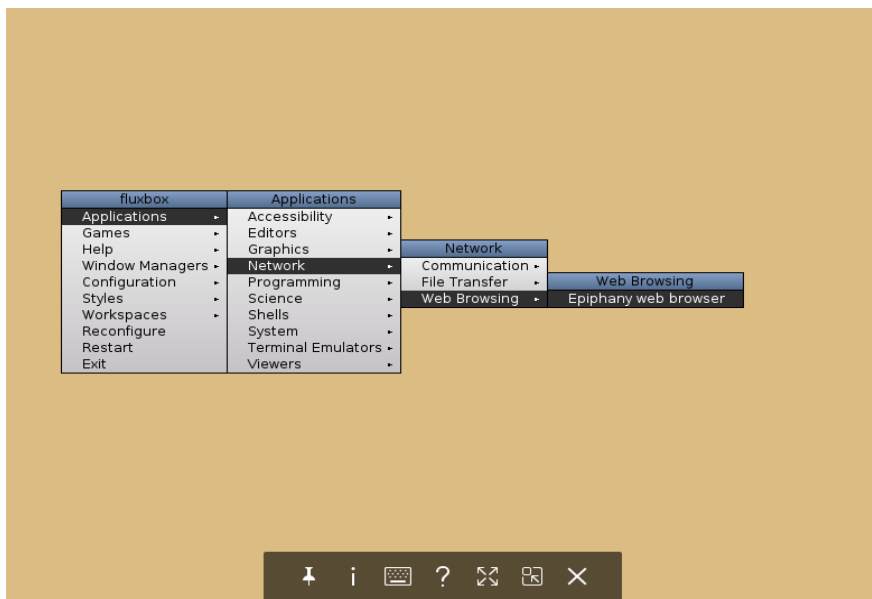
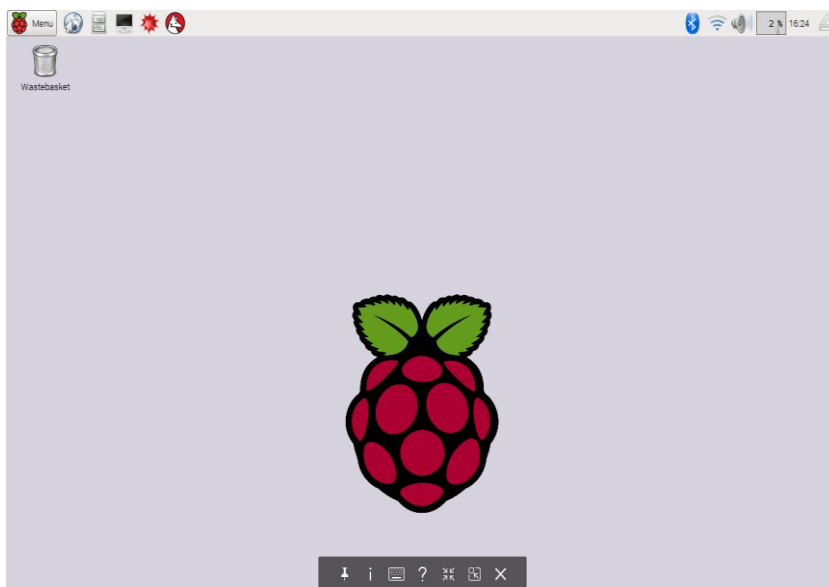


Figura 16. Escritorio Raspbian jessie Lite con Fluxbox en Raspberry Pi 1 modelo B.

A pesar de que el escritorio de Fluxbox es flexible, al ingresar al explorador no se muestra un ambiente que sea sencillo y es posible que a los usuarios se les pueda complicar su funcionamiento e ingreso. No obstante, se observó un cambio significativo en el rendimiento del procesador al ser un sistema operativo con el cual es posible tener aplicaciones instaladas de acuerdo con las necesidades del usuario. Al ser Fluxbox ligero, se incrementa la optimización en cada una de las cargas de proceso en el procesador. Sin embargo, para el usuario final puede ser complicado manipular cada una de las aplicaciones sin tener un previo conocimiento sobre el uso e instalación del sistema.

En el transcurso del desarrollo, surgió en el mercado la nueva versión del sistema embebido, el Raspberry Pi 3 modelo B (Raspi 3).

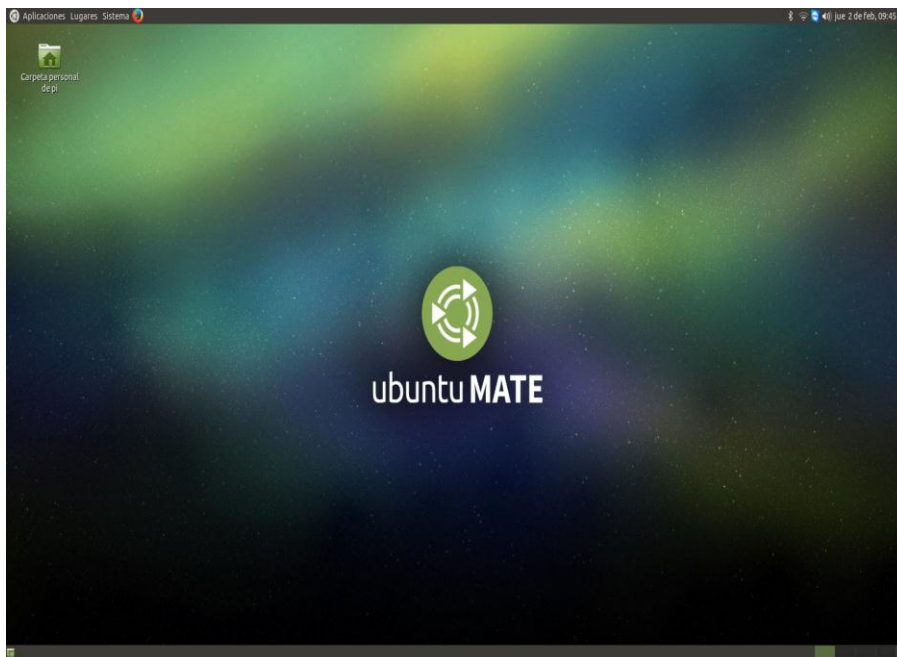
En la Figura 17 se muestra el escritorio del Raspbian jessie en su versión para Raspi 3.



*Figura 17.* Escritorio Raspbian jessie en Raspberry Pi 3 modelo B.

Es una buena alternativa para los clientes Raspi 3, de esta manera se puede combinar el medio guiado y no guiado para su conexión con el servidor. Utiliza una interfaz gráfica amigable y tiene diversidad de aplicaciones que pueden ser utilizadas por el usuario. El explorador Epiphany basado en el escritorio GNOME facilita el acceso y permite obtener eficiencia en los procesos. La sinergia entre el sistema operativo montado en el Raspi 3 y la interfaz de usuario proporciona un medio por el cual obtener un prototipo factible.

Se realizó la última prueba de sistemas operativos para el Raspi 3, utilizando el Ubuntu Mate; esta versión permite presentar al usuario un entorno más amigable con respecto a las interfaces utilizadas habitualmente. En la Figura 18 se observa un ejemplo del escritorio anteriormente mencionado.



*Figura 18.* Escritorio Ubuntu mate en Raspberry Pi 3 modelo B.

Entre las características principales del Raspi 3 con Ubuntu Mate destacan las aplicaciones que ofrece de manera nativa, su interface, la optimización del procesador, la flexibilidad de su explorador, etc. En cada prueba de los diversos sistemas operativos antes mencionados se buscaron tres cosas esenciales: optimización del procesador, integración de la plataforma con el usuario y facilidad de acceso de su explorador para hacer la conexión con la PC utilizada como servidor. Se observó que este sistema operativo junta las características esenciales para el funcionamiento cliente-servidor. Se destaca de manera significativa la optimización del servidor, ya que es importante que los procesos que son llamados por el cliente, puedan ser respondidos de manera rápida y eficiente por el servidor; de esta manera se puede obtener un prototipo funcional.

#### Segunda fase: servidor para virtualización

Para montar el servidor, se utilizó una PC que permitiera instalar lo necesario y obtener el escritorio virtual e interactuar con el sistema embebido desde el explorador a través de la dirección de red asignada al servidor. En la Figura 19 se observa el escritorio del sistema operativo Ubuntu Server.

Teniendo el sistema operativo funcional en el ordenador, es imprescindible hacer la instalación de cada uno de los componentes principales para la ejecución del escritorio virtual. Algunos de los elementos que lo integran son los siguientes: Apache, PHP, Ajax y DHTML. Cada uno de estos complementos tiene una función importante en la ejecución del software para la administración en los sistemas embebidos Raspberry Pi. Para montar los elementos principales, se instaló la herramienta Xampp, la cual es un paquete que incluye los componentes importantes para la instalación del software eyeOS 1.9.0.3. En la Figura 20 se muestra la primera parte del proceso, que consistió en otorgar el permiso

al instalador descargado Xampp. Después de obtener el permiso, se inició la instalación del paquete Xampp, que entre sus componentes incluye el sistema de gestión de base de datos MySQL, importante como requisito para el buen funcionamiento del escritorio virtual eyeOS.



Figura 19. Escritorio Ubuntu server 16.04 LTS.

```
peter@servidor-proyecto:~$ sudo rm -rf /opt/lampp
[sudo] password for peter:
peter@servidor-proyecto:~$ sudo chmod 755 xampp-linux-5.5.34-0-installer.run
peter@servidor-proyecto:~$
```

Figura 20. Permiso para ejecutar el instalador Xampp.

En la Figura 21 se observa una parte del proceso de instalación de Xampp. Al finalizar la instalación, se muestra una ventana en el explorador predeterminado; con esta acción se puede observar que el paquete ha sido instalado con éxito. Ahora el siguiente paso es iniciar Xampp en Ubuntu Server; en la Figura 22 se muestra su inicio mediante línea de comando. La Figura 23 muestra el otro modo de inicio para Xampp; con la ventana se pueden iniciar de igual modo sus componentes y, por consiguiente, realizar la configuración de la base de datos.

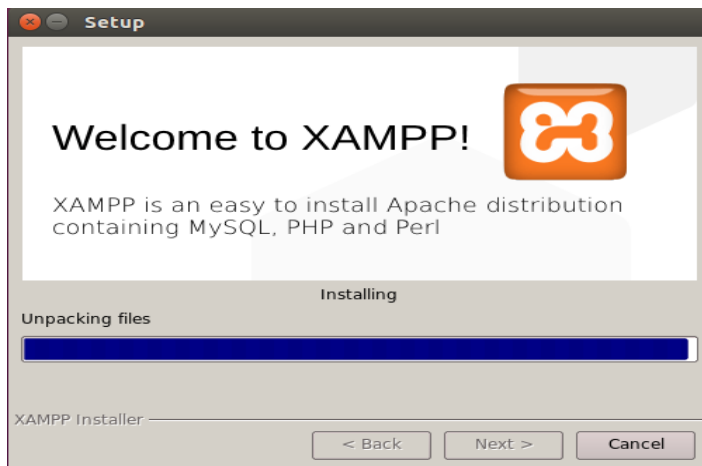


Figura 21. Instalador de Xampp en Ubuntu server.

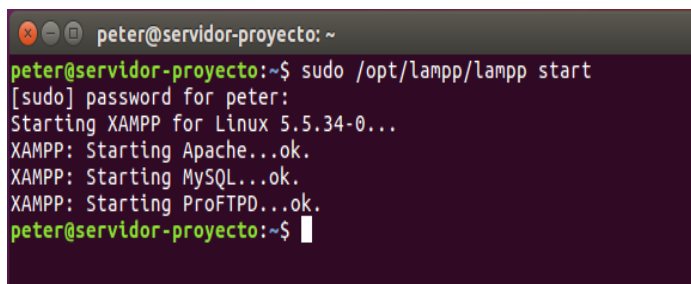


Figura 22. Iniciar Xampp desde terminal en Ubuntu server.

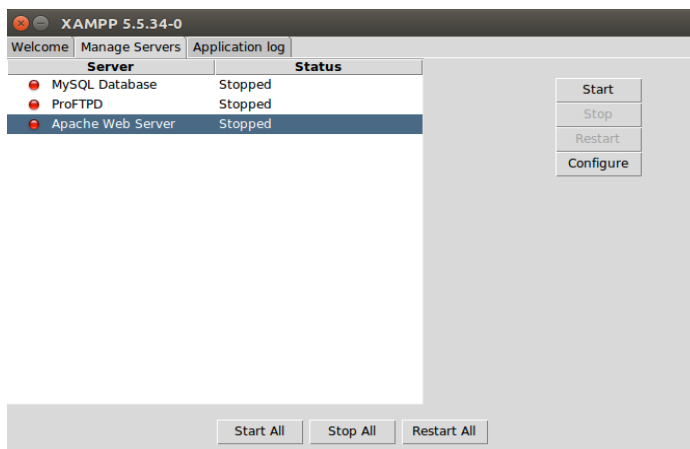


Figura 23. Ventana para iniciar los gestores necesarios en Ubuntu server.

Los tres elementos principales están ejecutándose perfectamente en el sistema operativo. Sin embargo, para el correcto funcionamiento de eyeOS es imprescindible crear una base de datos y cambiar algunos parámetros en el archivo PHP.ini. La Figura 24 muestra la creación de la base de datos para el escritorio virtual.

Para continuar con la configuración de los requerimientos principales del escritorio virtual, es imprescindible realizar la modificación de algunos parámetros del archivo PHP.ini, si bien, varios elementos tienen que ser modificados específicamente y quitar punto y coma definidos como comentarios en líneas de código para su funcionamiento, tales como `extensión=php_curl.dll`, `extensión=php_shmod.dd`, `display_errors=enable`, `memory_limit=512M`, `file_uploads=On`, habilitar extensiones de Mysql, Sqlite y PDO. Las modificaciones del archivo deben ser hechas de acuerdo a los requerimientos antes mencionados, ya que, si no son correctos los parámetros, el escritorio virtual no podrá ser instalado. En la Figura 25 se puede observar una parte del encabezado del archivo PHP.ini y un breve resumen sobre el contenido del archivo.

Una vez hechas las configuraciones pertinentes en PHP.ini, se ingresa al explorador web en Ubuntu Server y se ingresa al `localhost/eyeos/installer/`, tal y como se muestra en la Figura 26. No obstante, se establece la configuración de la base de datos anteriormente creada y se asigna una contraseña para el usuario por default root.

Posteriormente, se hace la instalación del escritorio virtual y se hace el inicio de igual forma por el explorador web, pero ahora en la dirección `localhost/eyeos/index.php`. En la Figura 27 se puede observar el primer inicio de eyeOS, en el cual se pide el usuario por default root y también su contraseña. Después de ingresar correctamente los parámetros solicitados, en la Figura 28 se muestra el escritorio virtual.

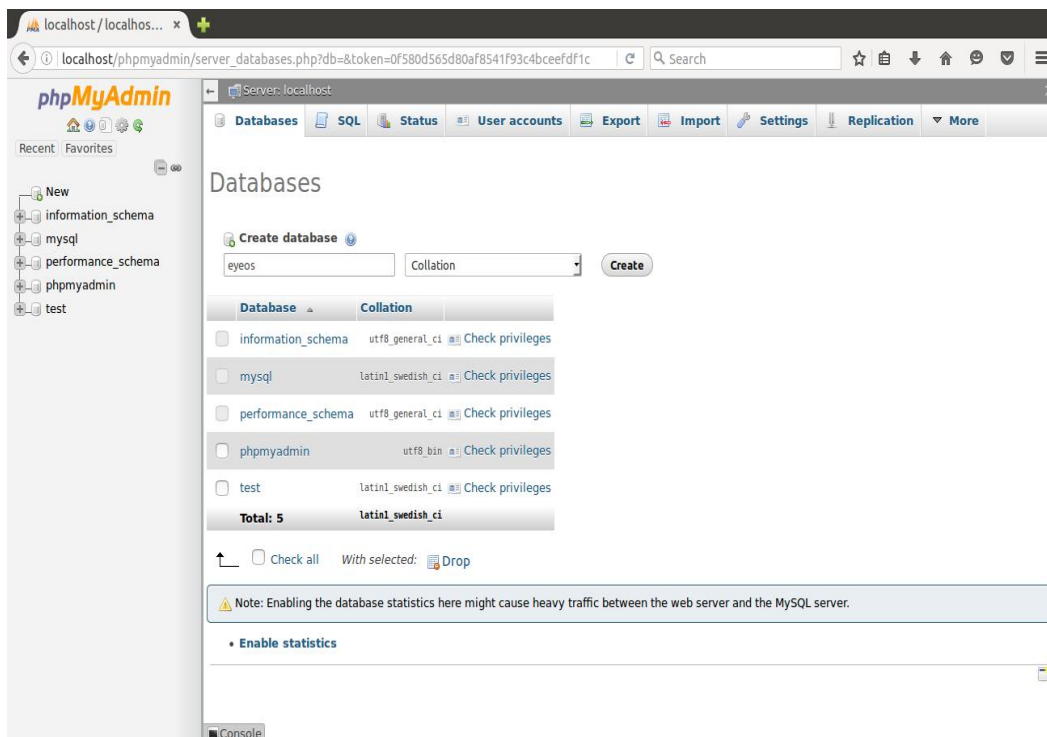


Figura 24. Base de datos en phpMyAdmin para eyeOS.

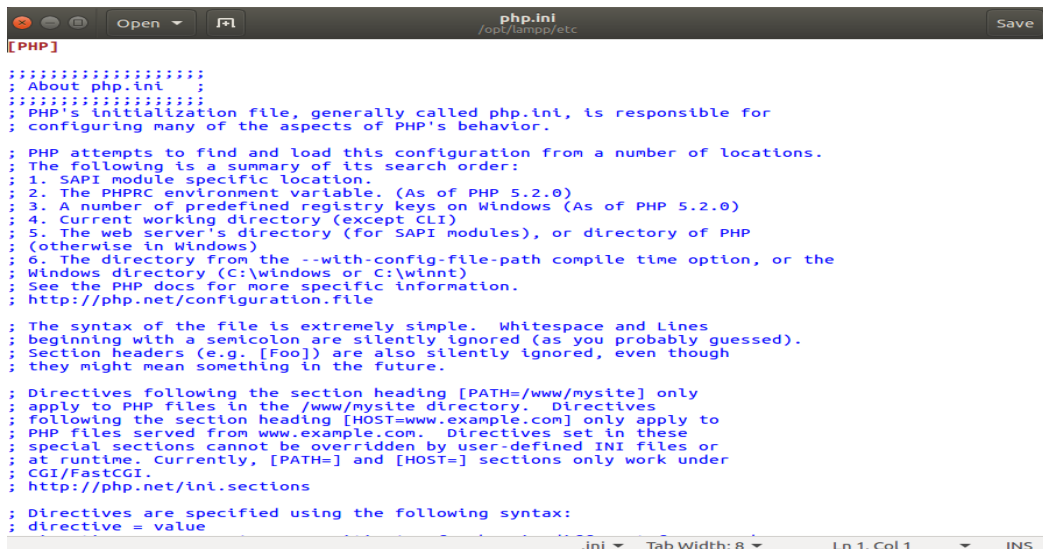


Figura 25. Archivo PHP.ini en Ubuntu server.



El escritorio virtual eyeOS incluye diversas aplicaciones de código libre para la educación, programación, creación y edición de texto. Es posible realizar transferencia de archivos para una mejor interacción entre el cliente y el servidor. Al estar instalado el escritorio virtual en el servidor, es posible lograr una reducción significativa en la carga de procesos de los clientes conectados. El desarrollo e implementación de aplicaciones nativas en eyeOS puede permitir la interacción con un prototipo que proporcione las herramientas idóneas para cada uno de los usuarios.

### Tercera fase: comunicación cliente servidor

Posteriormente, es imprescindible realizar la interacción remota entre el Raspberry Pi (Raspi) determinado como cliente y el ordenador designado como servidor. Para lograr esta conexión, es necesario poseer la dirección ip asignada por la red, en el caso del prototipo es 10.4.215.185. No obstante, se recomienda para una mejor conexión entre el cliente y el servidor, poseer una ip fija asignada en ambos extremos. De esta manera se mantiene una conexión más estable.

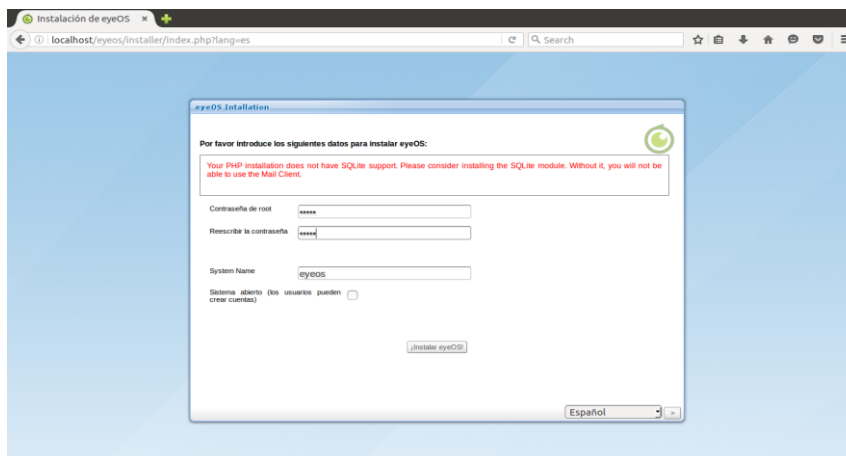


Figura 26. Ventana de instalación eyeOS 1.9.0.3.

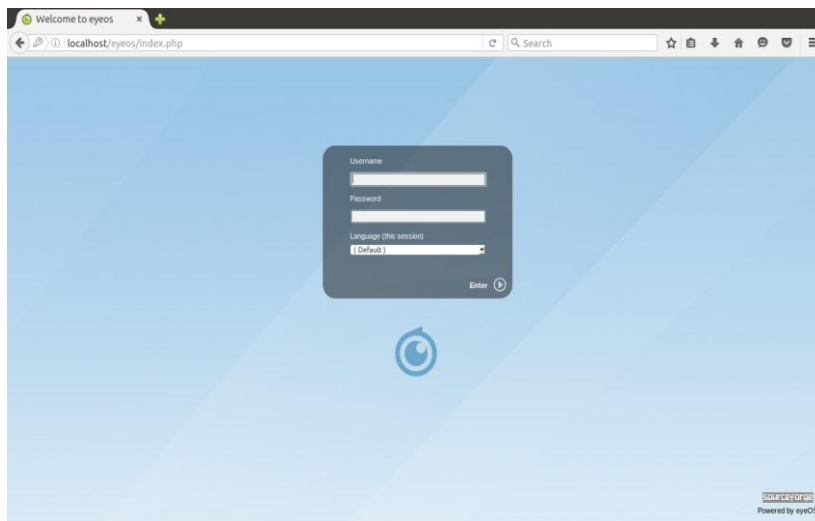


Figura 27. Ventana de inicio en eyeOS 1.9.0.3.

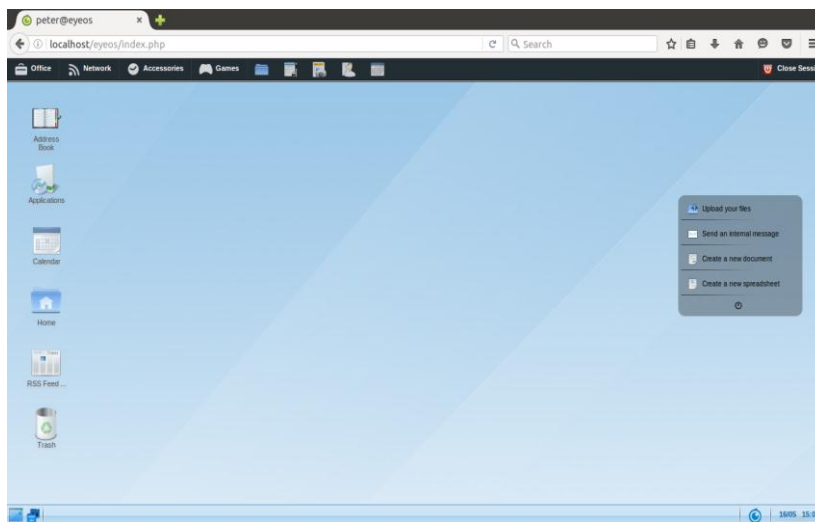


Figura 28. Escritorio virtual eyeOS 1.9.0.3.

Para el ingreso desde el sistema embebido se propone crear los usuarios necesarios para cada uno de los clientes desde el escritorio virtual instalado en el servidor, con el propósito de tener mayor control de cada uno de los sistemas embebidos asignados como clientes. En la Figura 29 se observa la primera pantalla desde el explorador

web. Se agrega el usuario y la contraseña designados por el administrador para poder ingresar a las aplicaciones que ofrece el escritorio virtual de eyeOS.

En la Figura 30 se muestra la interacción entre el explorador web Epiphany del Raspi 1 modelo B, el cual se incorpora con el sistema operativo Raspbian jessie mediante una conexión por medio guiado. El cliente realiza la petición de conexión al servidor utilizando la dirección de red asignada y enseguida se realiza el enlace.

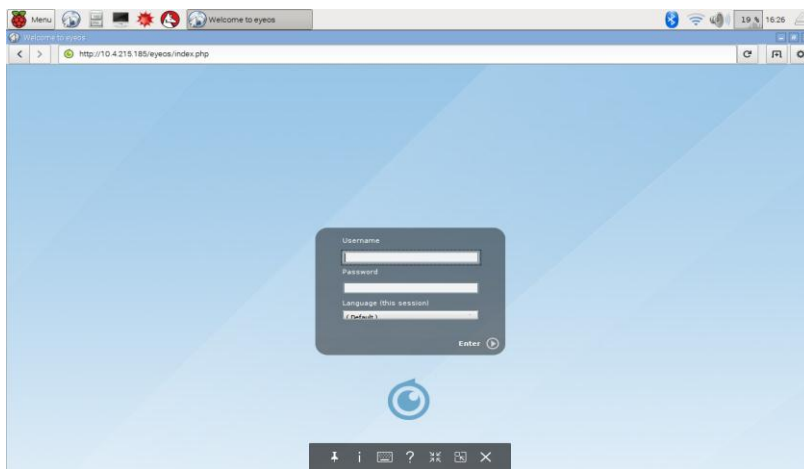


Figura 29. Inicio eyeOs en Raspbian jessie desde Raspberry Pi 3 modelo B.

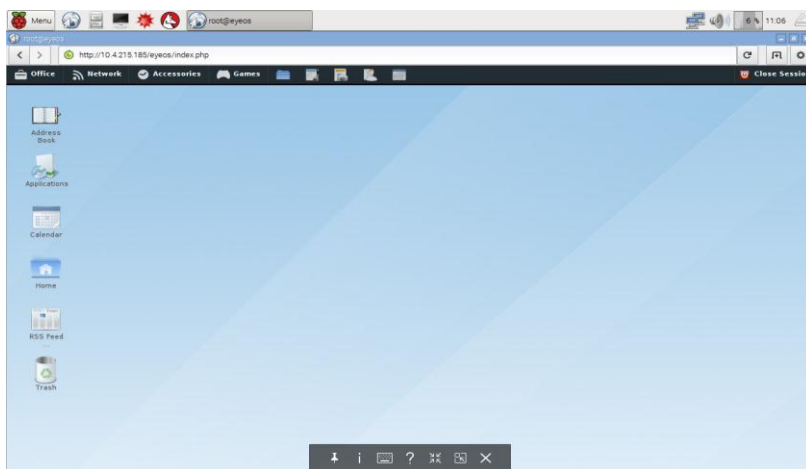


Figura 30. Escritorio virtual eyeOS 1.9.0.3 en Raspbian jessie.

Hasta aquí se pueden observar los cambios en la carga del procesador y es posible realizar una buena conexión con el servidor para realizar las peticiones; sin embargo, el hardware puede limitar el desarrollo óptimo del escritorio virtual. Así que, al obtener el modelo más reciente del Raspi 3 modelo B, puede conocerse de manera más amplia el trabajo en tiempo real de cada uno de los clientes conectados. En la Figura 31 se puede observar el sistema embebido y su integración con el sistema operativo Raspbian jessie.

La conexión con el servidor se hace a través de un medio inalámbrico o no guiado; de esta manera, se puede conocer el rendimiento entre ambos medios de transmisión. Se observó una diferencia relevante entre ambos modelos de Raspberry Pi; esto ocurre debido al hardware que posee cada uno. Pese a que la conexión al servidor por medio del explorador web puede ser más eficaz, el escritorio puede ser manipulado de acuerdo con los requisitos que el usuario requiera. En la Figura 32 se muestra la interacción del sistema operativo Ubuntu Mate instalado en el Raspi 3 modelo B y el escritorio virtual.

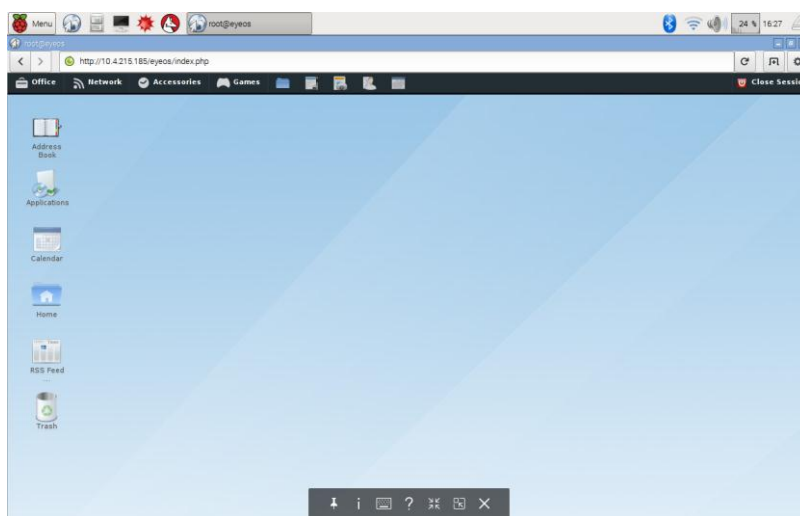


Figura 31. Escritorio virtual en Raspberry Pi 3 modelo B.

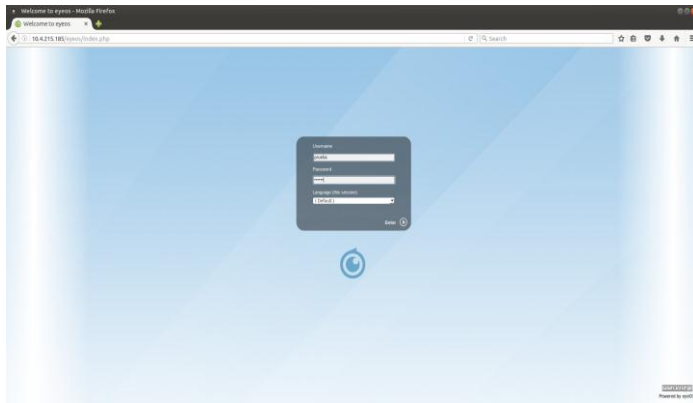


Figura 32. Escritorio virtual en Raspberry Pi 3 modelo B con Ubuntu mate.

La incorporación de este tipo de sistema operativo hizo posible mantener estable el uso del procesador durante la conexión con el servidor. Es imprescindible resaltar que el prototipo puede obtener mejor rendimiento si se posee el hardware correcto para el sistema embebido. Se probó el acceso al eyeOS desde una PC de escritorio y un ordenador portátil; cada uno de ellos mostraron estabilidad en la interacción con el servidor.

Las nuevas tecnologías emergentes hacen posible la experimentación en diversas condiciones para poder obtener un modelo práctico y eficiente.

#### Cuarta fase: evaluaciones

Para conocer el funcionamiento general al realizar las conexiones cliente-servidor, es imprescindible hacer pruebas de rendimiento del CPU, memoria, velocidad de transmisión de los datos y consumo de energía. Para lograr datos confiables, se utilizaron diversos programas disponibles para los sistemas con base en Linux y se tomaron los resultados más confiables. Para hacer pruebas del prototipo, se utilizó un ordenador, un router y un medio guiado y no guiado. En la Figura 33 se pueden observar las especificaciones del ordenador utilizado como servidor.

Para lograr resultados más completos, se instaló el mismo sistema operativo en los dispositivos utilizados como clientes; sin embargo, se limitó al sistema operativo que el Raspberry 1 modelo B podía soportar al poseer hardware limitado en comparación con el Raspberry 3 modelo B. La Figura 34 muestra el escritorio de ambos clientes.

Es necesario conocer datos de los dispositivos clientes y cómo cambian al hacer pruebas de conexión con el servidor. La Figura 35 muestra el comportamiento del CPU y memoria utilizando el comando htop en Raspberry Pi 1 modelo B.

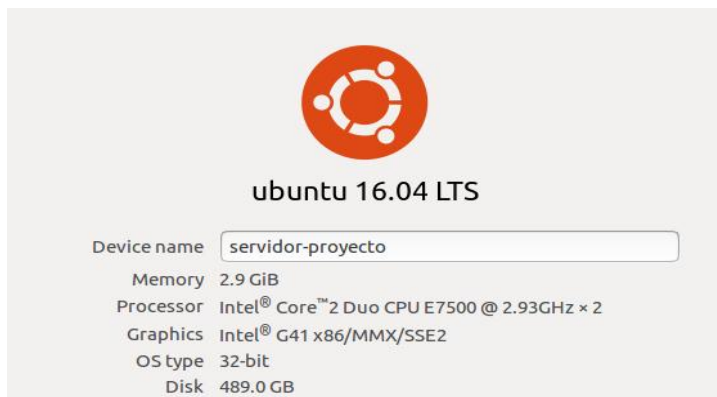


Figura 33. Especificaciones de ordenador utilizado como servidor.

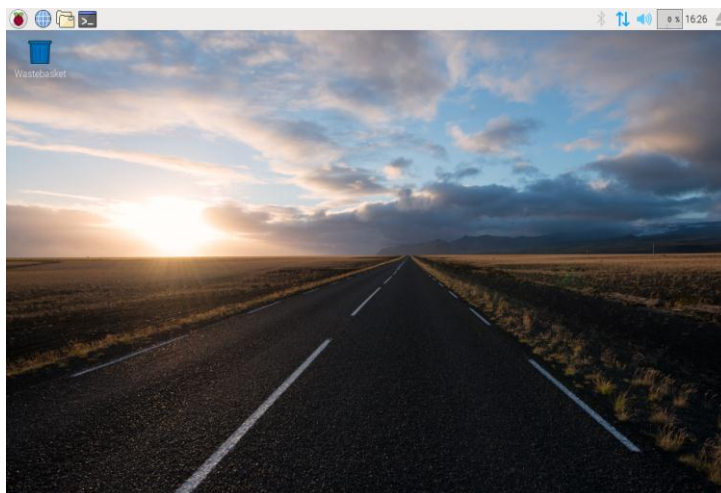


Figura 34. Escritorio de Raspbian jessie with pixel.

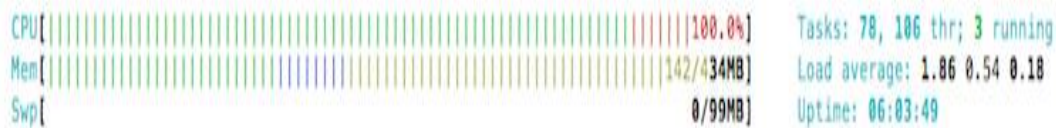


Figura 35. Comando htop en Raspberry Pi 1 modelo B.

Se observa que el Raspberry 1 modelo B consta de procesador con un núcleo y que, al momento de realizar la prueba de conexión al escritorio virtual en el servidor, utiliza el 100% de su uso. De los 434MB disponibles en memoria, utiliza 142 al momento de hacer la conexión con el escritorio virtual. Es fundamental conocer el tiempo de inicio y de conexión para conocer las diferencias con el modelo más reciente. La Figura 36 muestra el tiempo que tarda en iniciar el cliente.

```

pi@raspberrypi:~ $ systemd-analyze
Startup finished in 4.637s (kernel) + 20.751s (userspace) = 25.389s

```

Figura 36. Comando systemd\_analyze en Raspberry Pi 1 modelo B.

Posteriormente, en la Figura 37 se puede observar el traceroute entre el cliente y el servidor.

Al tener montada una red local entre los clientes y el servidor sin conexión con el ISP (Proveedor de servicios de internet), es posible conocer en detalle el tiempo de respuesta. El router permite realizar la conexión a través de medio guiado y no guiado; en tal caso, se utiliza el medio cableado para realizar la prueba en el Raspberry Pi 1 modelo B. El resultado se muestra en ms o milisegundos para saber la latencia en la red. La

siguiente evaluación se realizó en el Raspberry Pi 3 modelo B a través de medio guiado y no guiado. En la Figura 38 se observa el comando htop en el modelo mencionado.

El Raspberry Pi 3 modelo B consta de un procesador con cuatro núcleos. Esto le permite distribuir la carga de los procesos, asignando consecutivamente 50.9%, 41.9%, 50.7% y 42.9%. El total de la memoria utilizable es de 925MB y para la conexión con eyeOS emplea 202MB. En breve conclusión sobre el uso del CPU y la memoria, se puede decir que el hardware tiene un rol fundamental para lograr la optimización del procesador. El tiempo de inicio en el cliente se observa en la Figura 39. El tiempo de inicio es inferior al del modelo anterior, puesto que el hardware del Raspberry Pi 3 modelo B es más potente. En la Figura 40 se observan los resultados del traceroute a través de medio guiado.

```
pi@raspberrypi:~ $ traceroute 192.168.1.19
traceroute to 192.168.1.19 (192.168.1.19), 30 hops max, 60 byte packets
 1 192.168.1.19 (192.168.1.19) 0.347 ms 0.291 ms 0.252 ms
```

Figura 37. Comando traceroute en Raspberry Pi 1 modelo B.

```
1 [|||||||||||||||||||||||||||||||||] 50.9% Tasks: 87, 137 thr; 1 running
2 [|||||||||||||||||||||||||||||||] 41.9% Load average: 0.82 0.39 0.15
3 [|||||||||||||||||||||||||||||||] 50.7% Uptime: 05:58:57
4 [|||||||||||||||||||||||||||||||] 42.9%
Mem[|||||||||||||||||||||||||||||] 202/925MB
Swp[|||||||||||||||||||||||||||] 0/99MB
```

Figura 38. Comando htop en Raspberry Pi 3 modelo B.

```
pi@raspberrypi:~ $ systemd-analyze
Startup finished in 10.166s (kernel) + 9.277s (userspace) = 19.444s
```

Figura 39. Comando systemd-analyze en Raspberry Pi 3 modelo B.



```
pi@raspberrypi:~ $ traceroute 192.168.1.19
traceroute to 192.168.1.19 (192.168.1.19), 30 hops max, 60 byte packets
 1 192.168.1.19 (192.168.1.19) 0.327 ms 0.213 ms 0.169 ms
```

*Figura 40.* Comando traceroute en Raspberry Pi 3 modelo B UTP.

Los paquetes enviados para hacer la prueba son para conocer la duración promedio del cliente y el servidor. El resultado del paquete enviado genera tres diferentes valores en milisegundos para tener mayor confiabilidad. No obstante, es necesario conocer la ruta trazada a través del medio inalámbrico. En la Figura 41 se observa el comando traceroute desde Raspberry Pi 3 modelo B con Wi-Fi. Es importante resaltar que, cuanto mayor latencia se genere en la red, habrá más demora en la transmisión de los paquetes.

```
pi@raspberrypi:~ $ traceroute 192.168.1.19
traceroute to 192.168.1.19 (192.168.1.19), 30 hops max, 60 byte packets
 1 192.168.1.19 (192.168.1.19) 21.414 ms 21.534 ms 21.545 ms
```

*Figura 41.* Comando traceroute en Raspberry Pi 3 modelo B Wi-Fi.

Para conocer la evaluación del medio de transmisión entre el cliente y el servidor, se instaló el programa iperf, el cual permite medir la velocidad que alcanzan dos nodos conectados mediante una red local. Para realizar el intercambio de paquetes, se habilita un socket que permite la comunicación entre el cliente y el servidor, proporcionando un flujo de datos infalible y estructurado. El servidor se mantiene en escucha para poder responder a las peticiones del cliente. En la Figura 42 se muestra el resultado de la

petición del Raspberry Pi 1 modelo B a través de medio guiado. La Figura 43 muestra el resultado al evaluar el Raspberry Pi 3 modelo B a partir de un medio guiado. En la Figura 44 se puede observar el resultado de iperf en Raspberry 3 modelo B utilizando el medio no guiado o inalámbrico. En la Figura 45 se puede observar la comparación gráfica entre el medio de transmisión por cable y el inalámbrico de los clientes Raspberry Pi 1 modelo B y Raspberry Pi 3 modelo B.

```
pi@raspberrypi:~ $ iperf -c 192.168.1.19
-----
Client connecting to 192.168.1.19, TCP port 5001
TCP window size: 43.8 KByte (default)
-----
[ 3] local 192.168.1.46 port 35986 connected with 192.168.1.19 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec  64.6 MBytes 54.1 Mbits/sec
```

*Figura 42.* Comando iperf en Raspberry Pi 1 modelo B.

```
pi@raspberrypi:~ $ iperf -c 192.168.1.19
-----
Client connecting to 192.168.1.19, TCP port 5001
TCP window size: 43.8 KByte (default)
-----
[ 3] local 192.168.1.29 port 53012 connected with 192.168.1.19 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec  112 MBytes 94.2 Mbits/sec
```

*Figura 43.* Comando iperf en Raspberry Pi 3 modelo B UTP.

```
pi@raspberrypi:~ $ iperf -c 192.168.1.19
-----
Client connecting to 192.168.1.19, TCP port 5001
TCP window size: 43.8 KByte (default)
-----
[ 3] local 192.168.1.4 port 43650 connected with 192.168.1.19 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.2 sec  5.50 MBytes 4.53 Mbits/sec
```

*Figura 44.* Comando iperf en Raspberry Pi 3 modelo B Wi-Fi.

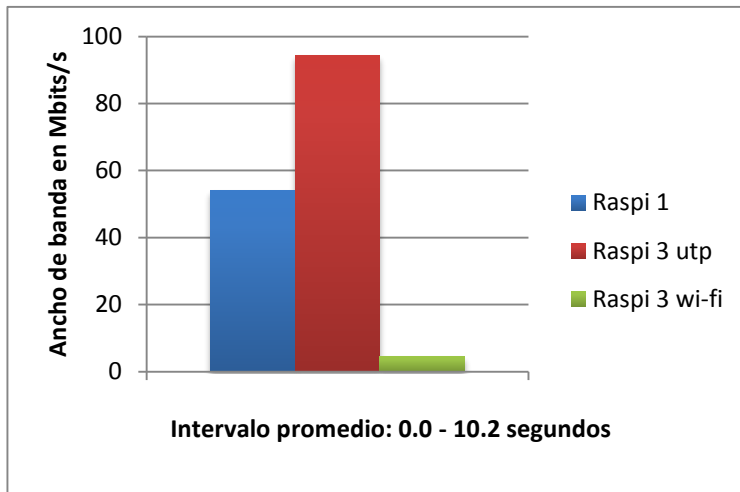


Figura 45. Comparación entre UTP y Wi-Fi del Raspi 1 y 3.

Posteriormente, se realizó la evaluación de múltiples usuarios conectados a través de los clientes Raspberry Pi, ordenador de escritorio con Linux y MacBook Pro, con la finalidad de obtener resultados sobre el rendimiento del CPU en el servidor. En la Figura 46 se observa la conexión a través del ordenador con Linux y el Raspberry Pi 1 modelo B.

La Figura 47 muestra la conexión del Raspberry Pi 3 modelo B y el ordenador con Linux.

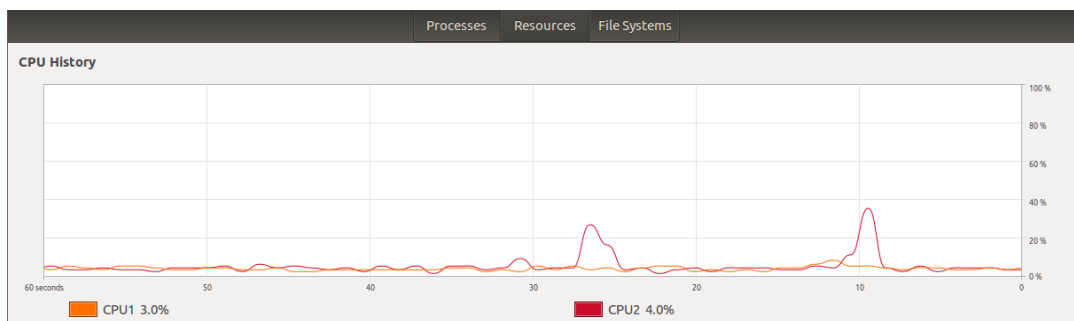


Figura 46. Uso del CPU en conexión Raspi 1 modelo B y ordenador.

En la Figura 48 se puede observar el uso del CPU al realizar la conexión a través de los clientes Raspberry Pi, el ordenador con Linux y el portátil.

A pesar de ser un servidor improvisado en un ordenador de escritorio, se mantiene estable y sin problemas. No obstante, de igual manera que como se hizo con los clientes, se realizó la prueba para conocer el tiempo de inicio en el servidor. La Figura 49 muestra el resultado al ingresar el comando `systemd-analyze`.

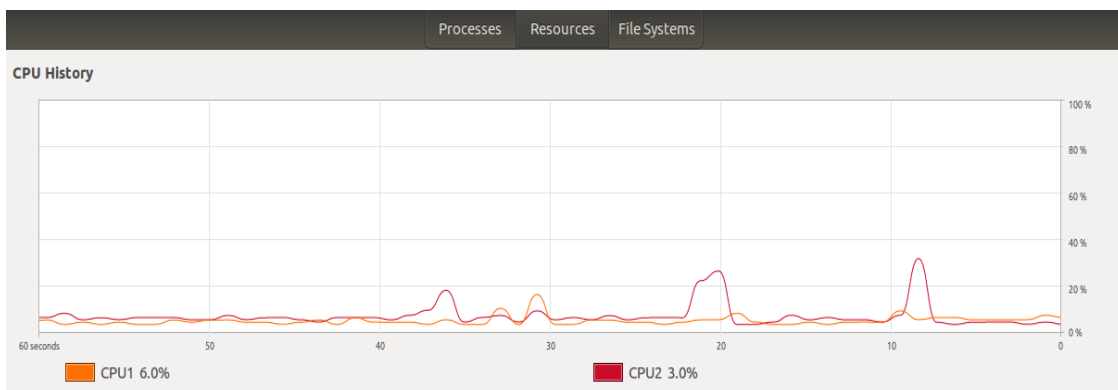


Figura 47. Uso del CPU en conexión Raspi 3 modelo B y ordenador.

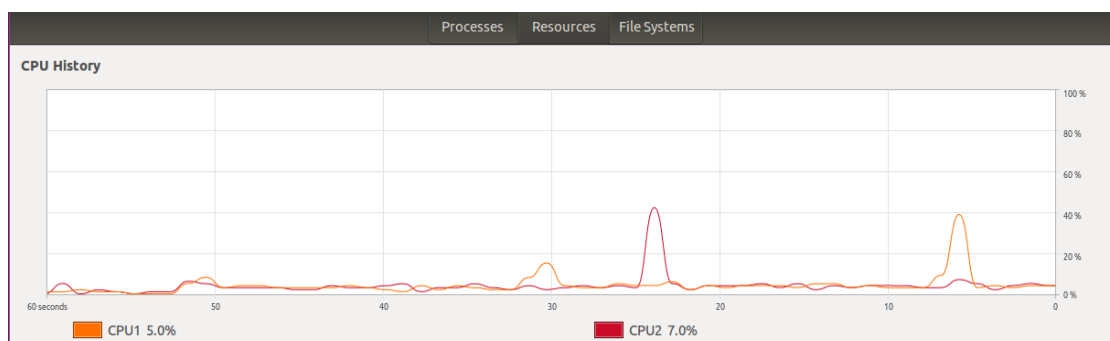


Figura 48. Uso del CPU en server al recibir cuatro conexiones.

```
peter@servidor-proyecto:~$ systemd-analyze
Startup finished in 17.691s (kernel) + 31.471s (userspace) = 49.162s
```

Figura 49. Tiempo de inicio en servidor.

Con la finalidad de obtener el modelo de un laboratorio de cómputo eficiente, es imprescindible tener en cuenta el consumo de energía del servidor y los clientes. Los datos que se muestran a continuación son aproximaciones y pueden variar de acuerdo con el hardware que se implemente. Por ejemplo, la fuente de poder al utilizar un ordenador de escritorio como servidor, el monitor a utilizar y la cantidad de potencia eléctrica producida por los clientes Raspberry Pi. Un ordenador, incluyendo el monitor LCD y los periféricos básicos, puede llegar a consumir aproximadamente de 0.25 a 0.28 kW o el equivalente de 250 a 280 Watts (W). Sin embargo, la fuente de poder puede ser diferente y tener un consumo mayor o menor al proporcionado en el análisis.

El Raspberry Pi 1 modelo B tiene un consumo energético de 3.5 W; incluyendo el consumo del monitor LED, genera 16.5 W o 0.0165 kW. El Raspberry Pi 3 modelo B tiene un consumo de 4.0 W; incluyendo el monitor, resulta en 17 W o 0.017 kW aproximadamente.

Se realizó el cálculo del total en consumo energético respecto de las tarifas aplicadas en territorio mexicano en el año 2017. De acuerdo con la Comisión Federal de Electricidad (CFE), el costo básico para consumidores es de \$0.793 MXN por los primeros 75kWh; el consumo intermedio es de \$0.956 MXN en los siguientes 65kWh y en consumos mayores a los anteriores es de \$2.802 MXN por kWh adicional. La estimación aproximada es en base a un ordenador convencional y a un dispositivo Raspberry Pi 3 modelo B, cada uno con el equipo necesario para su implementación; teniendo en cuenta

el consumo energético del switch TP-Link de 16 puertos 10/100/1000 a 13.3W o 0.0133kW, monitor LCD 30W o 0.03kW y PC de escritorio con fuente de poder de 250W o 0.25kW, además de considerar un ambiente de trabajo de 24 horas diarias durante un mes. El total aproximado de un equipo en el laboratorio convencional es de 293.3W, 7.04kW – 24 horas, 211kW – 1 mes, equivalente a \$ 321 MXN en un mes.

Para el dispositivo cliente del modelo y el equipo necesario para su funcionamiento, los resultados equivalen a 13.3W del switch, monitor LED de 18.5" 13W o 0.013kW, consumo del servidor Lenovo Thinkserver TS150 de 250W o 0.25kW y Raspberry Pi 3 modelo B a 4W. El total aproximado es de 276.3W o 0.2763kW, 6.63kW – 24 horas, 199kW – 1 mes, equivalente a \$ 287 MXN.

Los resultados anteriormente mencionados destacan la diferencia significativa del consumo energético entre el ordenador convencional y el Raspberry Pi; sin embargo, la implementación inicial al adquirir un servidor puede elevar el precio. El costo del hardware del dispositivo cliente es bajo en comparación con el PC. No obstante, si se ensambla un gabinete con el hardware necesario como alternativa del servidor, es posible que el costo disminuya.

## **CAPÍTULO V**

### **CONCLUSIONES, DISCUSIÓN Y RECOMENDACIONES**

#### **Introducción**

El Capítulo I detalló el avance que ha tenido a lo largo del tiempo el uso de la virtualización, la historia de sus inicios y la importancia de su implementación. Se analiza el planteamiento del problema, la justificación y las ventajas de incorporar la virtualización con los sistemas embebidos. Se realiza la propuesta de objetivos: (a) crear un prototipo de laboratorio de cómputo capaz de aumentar la eficiencia de las herramientas computacionales y (b) desarrollar una infraestructura tecnológica con el propósito de estar a la vanguardia. Se hace un análisis de las limitaciones y delimitaciones que pueden acontecer en el transcurso del proyecto; materiales, métodos y técnica a utilizar y definición de términos. Se realiza el marco filosófico sobre la optimización de los recursos tecnológicos.

El Capítulo II proporcionó un concepto general y profundo sobre el uso de los laboratorios escolares, laboratorios de cómputo, sistemas embebidos y medios de transmisión. Plantea una comparación entre los diferentes tipos de virtualización; ejemplos, ventajas, desventajas y conceptos. Además, se realiza un estudio relativo a las diferentes herramientas de virtualización para un servidor, su arquitectura y los servicios que proporcionan. Se profundiza en las características principales de los sistemas embebidos en el mercado tecnológico, ventajas y desventajas que se obtienen al trabajar con su plataforma. Por

último, se efectúa un análisis acerca de los distintos medios de transmisión guiados y no guiados. No obstante, la elección del hardware y software indicados se realiza teniendo en cuenta las limitaciones y delimitaciones mencionadas en el Capítulo I.

El Capítulo III propuso las características referentes a la metodología del proyecto, la planificación y las fases en las que se divide el proyecto.

El Capítulo IV presentó los resultados de las fases en las que se divide el proyecto y los desafíos encontrados. Se describen las especificaciones técnicas del software y hardware utilizados para el desarrollo del prototipo. Se explica la planificación para lograr los objetivos propuestos por medio de la combinación de virtualización y los sistemas embebidos. El modelo de fases para el prototipo consta de cuatro etapas: (a) pruebas de sistemas operativos ligeros para optimizar el procesador del Raspberry Pi, (b) montar un servidor para la virtualización con eyeOS, (c) comunicación local entre el cliente y el servidor y (d) evaluaciones sobre uso de CPU, memoria, red y consumo energético. En virtud de la afabilidad de la tecnología acerca de la virtualización y los sistemas embebidos, es posible obtener la base de una infraestructura tecnológica eficiente y de bajo consumo energético.

El Capítulo V ofrece las conclusiones, la discusión y las recomendaciones para trabajos a futuro.

## **Conclusiones**

El documento presenta el prototipo de un laboratorio de cómputo, en el cual tanto el hardware como el software convergen para efectuar el objetivo de aumentar la eficiencia de las herramientas computacionales. El resultado ha sido un modelo que permite la interacción entre virtualización y sistemas embebidos, en el cual los clientes son



dispositivos con sistemas embebidos. El Raspberry Pi utilizado como cliente es usado en sus dos diferentes modelos, cada uno con entrada de 4.5 y 5 Volts, respectivamente, para su alimentación de energía, de consumo energético de 3.5 W y 4.0 W. El cliente se conecta a un servidor dedicado para el prototipo mediante un explorador con HTML5 y la dirección ip del servidor, el servidor contiene la plataforma de virtualización eyeOS. El escritorio virtual posee aplicaciones nativas y es posible hacer transferencia de archivos entre clientes y el servidor. La comunicación entre el cliente y el servidor puede ser a través de medio guiado o no guiado. Sin embargo, de acuerdo con las evaluaciones realizadas, se recomienda utilizar el medio guiado, ya que permite incrementar la velocidad de transferencia en los paquetes. Es posible observar las diferencias entre el costo de los equipos necesarios para la implementación de un laboratorio de cómputo tradicional y el prototipo referido; el modelo puede llegar a lograr marcar la diferencia en el consumo energético y el espacio físico. La culminación del prototipo puede dar paso al cumplimiento de los objetivos descritos en el documento.

### **Discusión**

Ya teniendo los resultados, se posee el marco general para el desarrollo del prototipo funcional de un laboratorio de cómputo, utilizando tarjetas embebidas y virtualización de escritorio. Es posible utilizar el diseño de la plataforma para implementar aplicaciones desarrolladas a medida, de forma nativa. El acceso a través del explorador web permite la interacción remota entre el cliente y el servidor, permitiendo tener cohesión con el escritorio virtual. Asimismo, al conseguir administrar usuarios, es posible asignar un usuario para cada dispositivo cliente. El medio de transmisión guiado permite obtener comunicación estable entre el cliente y el servidor, logrando optimizar el tiempo de

respuesta. El consumo energético es representativo entre el ordenador convencional y el Raspberry Pi. Se alcanzó un abundante entendimiento sobre las diferentes plataformas de hardware y software open source disponibles en el mercado y la trascendencia que tienen para conseguir un modelo eficiente.

### **Recomendaciones**

Para trabajos futuros, se plantean las siguientes recomendaciones:

1. Desarrollar aplicaciones a medida para eyeOS.
2. Implementar el sistema PiNet para centralizar y administrar grupos en instituciones de nivel medio.
3. Adquirir software de licencia para obtener una versión completa del escritorio virtual.
4. Obtener el servidor completo o una PC ensamblada para alojar el software necesario.
5. Adquirir dispositivos suficientes Raspberry Pi para su implementación en instituciones de nivel medio.
6. Implementar el prototipo en instituciones educativas de nivel medio para conocer la aceptación del hardware y el software.

## **APÉNDICE A**

### **COSTOS DE LABORATORIO CONVENCIONAL Y PROTOTIPO**

Equipo convencional	Precio M.N.	Equipo para prototipo	Precio M.N.
Gabinete Slim con procesador Intel Core i3 a 3.2 GHz, DD de 500GB a 7200 RPM, S/unidad DVD, Windows 7 Pro, Monitor LED WIDE de 19". Mouse y Teclado alámbricos.	\$4,874	Kit Raspberry Pi 3 modelo B, Monitor LED Samsung LS19D300HY/ZX de 18.5". Mouse y Teclado alámbricos.	\$1,720 + \$1,649 + \$289= \$ 3,658
Switch TP-Link de 16 puertos 10/100/1000 Mbps.	\$1,799	Switch TP-Link de 16 puertos 10/100/1000 Mbps.	\$1,799
Patch Panel de red, 16 puertos Cat5e.	\$355	Patch Panel de red, 16 puertos Cat5e.	\$355
Regulador de voltaje, 8 contactos 1400VA.	\$269	Regulador de voltaje, 8 contactos 1400VA.	\$269
<b>Total (16 equipos)</b>	\$80,720	Servidor Lenovo ThinkServer TS150	\$6,499
		<b>Total (16 equipos)</b>	\$67,719

## REFERENCIAS

- Ahn, S., Lee, S., Yoo, S., Park, D., Kim, D. y Yoo, C. (2012). Isolation schemes of virtual network platform for cloud computing. *Kssii Transactions on Internet and Information Systems*, 6(11), 2764-2782. doi:10.3837/tiis.2012.11.001
- Ahiatrogah, P. D. y Adane, M. (2011). High school student's perception of computer laboratory learning environments in Ghana. *Ife Psychologia*, 19(1), 342-359. doi:10.4314/ifep.v19i1.64607
- Ali, I. y Meghanathan, N. (2011). Virtual machines and networks installation, performance, study, advantages and virtualization options. *International Journal of Network Security and its Applications*, 3(1), 1-15. doi:10.5121/ijnsa.2011.3101
- Aragundi Lucas, A. A., Cedeño Zamora, I. A., Quiroz Cedeño, R. Y. y Zambrano Veliz, J. D. (2012). *Implementación de un laboratorio para la virtualización de sistemas operativos mediante la instalación y configuración de ordenadores y servidores bajo plataformas GNU/LINUX y Windows server 2008 aplicando tecnología SCSI* (Tesis de pregrado). Universidad Técnica de Manabí, Manabí, Ecuador.
- Ashford Lee, E. y Arunkumar Seshia, S. (2011). *Introduction to embedded systems. A cyber-physical systems approach*, Berkeley, CA: UC Berkeley.
- Augusto Marchionni, E. y Martín Formoso, O. (2012). *Virtualización con VMware: lo mejor de la computación en la nube*, Buenos Aires: Fox Andina.
- Baigorria, L. (2017). *¿Qué es virtualización?* Recuperado de <https://uialberto.com/2010/09/26/%C2%BFque-es-virtualizacion/>
- Bonilla Suárez, J. J. y Carrasco Aguilar, D. S. (2010). *Análisis e implementación de un prototipo de servidor virtualizado sobre una distribución de Linux para el uso en PyMES* (Tesis de pregrado). Escuela Politécnica Nacional, Quito, Ecuador.
- Britos, D., Vargas, L., Arias, S., Giraudó, N. y Veneranda, G. (2013, junio). *Laboratorio virtual y remoto para la enseñanza de diseño y administración de redes de computadoras*. Memorias presentadas en el XV Workshop de Investigadores en Ciencias de la Computación, Paraná, Argentina.
- Bugnion, E., Devine, S., Rosenblum, M., Sugerman, J. y Wang, E. Y. (2012). Bringing virtualization to the x86 architecture with the original wmware workstation. *ACM Transactions on Computer Systems*, 30 (4), 1-51. doi:10.1145/2382553 .2382554

- Burd, S. D., Gaillard, G., Rooney, E. y Seazzu A. F. (2011, enero). *Virtual computing laboratories using vmware lab manager*. Conferencia presentada en 44<sup>th</sup> Hawaii International Conference on System Sciencies, Kahuai, HI, USA.
- Cardozo Lugo, N. (2014). Tecnologías de virtualización en los sistemas informáticos de las organizaciones empresariales del estado Zulia. *Revista Electrónica de Estudios Telemáticos*, 13(2), 49-67. Recuperado de <http://www.redalyc.org/articulo.oa?id=78431770004>
- Chouza, F., Erusalimsky, M., Lanzelotti, E. y Rodríguez, A. (2012, agosto). *Quadpector: un uav de bajo costo para observación remota*. Ponencia presentada en el Congreso Argentino de Sistemas Embebidos, Buenos Aires, Argentina.
- Duan, Q. y Lu, Y. (2013). Service oriented network discovery and selection in virtualization based mobile internet. *Journal of Computer Information Systems*, 1, 38-46. doi:10.1080/08874417.2013.11645630
- Duarte, J., Gargiulo, C. y Moreno, M. (2011). *Infraestructura escolar y aprendizajes en la educación básica latinoamericana: un análisis a partir del serce*. BID. Recuperado de <http://idbdocs.iadb.org/wsdocs/getdocument.aspx?docnum=36201660>
- Ebbers, M., Buchanan, T. S., Greggo, A., Joseph, D., Langer, J., Ong, E. y Wisniewski, M. (2010). *Performance test of virtual linux desktop cloud services on system z*. Recuperado de <http://www.redbooks.ibm.com/redpapers/pdfs/redp4593.pdf>
- Edwar, A., Susandri y Rahmaddeni. (2015). Optimizing server resource by using virtualization technology. *Procedia Computer Science*, 59, 320-325. doi:10.1016/j.procs.2015.07572
- Enríquez Herrador, R. (2009). *Guía de usuario de Arduino*. Recuperado de [http://www.uco.es/aulasoftwarelibre/wp-content/uploads/2010/05/Arduino\\_user\\_manual\\_es.pdf](http://www.uco.es/aulasoftwarelibre/wp-content/uploads/2010/05/Arduino_user_manual_es.pdf)
- Fernández Romero, Y. y García Pombo, K. (2011). Virtualización. *Revista Telemática*, 10(3), 61-73.
- Fitzgerald, S. y Shiloh, M. (2017). *Arduino projects book*. Recuperado de <http://electronicaytelecomunicaciones-jc.blogspot.mx/2016/02/arduino-projects-book.html>
- Friedrich, G., Reggiani, G., Cayssials, R., Galasso, C., Pellegrino, S., Cofre, L. y Velasquez, G. (2013, junio). *Sistemas embebidos en red con requerimientos de tiempo real*. Memorias presentadas en el XV Workshop de Investigadores en Ciencias de la Computación, Paraná, Argentina.

- Gimeno Martínez, S. (2008). *Evaluación de plataformas virtuales: estudio comparativo* (Tesis de maestría). Universidad Politécnica de Valencia, Valencia, España.
- González, E. B. (2010). *Gestor de máquinas virtuales* (Tesis de maestría). Universidad de Mendoza, Mendoza, Argentina.
- Hao, W., Fu, J., Trenkamp, C. y Prapatanant, P. (2012). Achieving better cloud Access experience with server virtualization. *Journal of Computational Methods in Sciences and Engineering*, 12(1), 29-37. doi:10.3233/JCM-2012-0434
- Heredia Escorza, Y. (2010, junio). *Incorporación de tecnología educativa en educación básica: dos escenarios escolares en México*. Ponencia presentada en el XI Encuentro Internacional Virtual Educa, Santo Domingo, República Dominicana.
- Hildenbrand, J. (2017). *A look at nvidia's jetson tk1*. Recuperado de <http://www.androidcentral.com/look-nvidias-jetson-tk1>
- Hwang, W., Kongcharoen, C. y Ghinea, G. (2014). To enhance collaborative learning and practice network knowledge with a virtualization laboratory and online synchronous discussion. *The International Review of Research in Open and Distance Learning*, 15(4), 113-137. doi:10.19173/irrodl.v15i4.1805
- Iglesias Benitez, A., Toledano Hernández, A., Toledo Enríquez, D., Martínez Casado, J. G. y González Aguilera, D. J. (2014). Desarrollo de un driver GNU/Linux para sistemas de adquisición de datos embebidos. *Revista Cubana de Ciencias Informáticas*, 8(2), 35-51. Recuperado de: <http://www.redalyc.org/articulo.oa?id=378334194003>
- Jiménez Ortega, R., Fernández Valverde, N., López Pino, J. L. (2017). *Eyeos: un sistema operativo web*. Recuperado de <https://es.slideshare.net/jlpino/eyeos-arquitectura-y-desarrollo-de-una-aplicacin>
- Krbeček, M. y Schauer, F. (2015). Communication and diagnostic interfaces in remote laboratory management systems. *International Journal of Online Engineering*, 11(5), 43-49. doi:10.3991/ijoe.v11i5.4926
- Lee, J. (2016). A new generation of ARM: 64-bit processing. *Odroid Magazine*, 27(3), 23-24.
- Lister, J. (2016). The advantages of an embedded system. Recuperado de [http://www.ehow.com/facts\\_5526848\\_advantages-embedded-system.html](http://www.ehow.com/facts_5526848_advantages-embedded-system.html)
- Lizondo, M., Agüero, P. D., Uriz, A. J., Tulli, J. C. y Gonzalez, E. L. (2012, agosto). *Embedded speaker verification in low cost microcontroller*. Ponencia presentada en el Congreso Argentino de Sistemas Embebidos, Buenos Aires, Argentina.

- López, E. P. (2017). *Sistemas embebidos*. Recuperado de <http://linuxemb.wikidot.com/tesis-c2>
- López Medina, A. (2010). *Análisis de la virtualización de sistemas operativos* (Tesis de pregrado). Universidad de Barcelona, Barcelona, España.
- López Sarmiento, D. A. y Garzón Romero, H. A. (2013). Diseño e implementación de ipv6 en un sistema embebido. *Revista Tecnura*, 17(37), 167-176.
- Lugo, G. (2006). La importancia de los laboratorios. *Construcción y Tecnología*, 12(223), 20-22.
- Ma, K., Teng, H., Du, L. y Zhang, K. (2014). Project-driven learning-by-doing method for teaching software engineering using virtualización technology. *Blended Learning*, 9(9), 26-31. doi:10.3991/ijet.v9i9.4006
- Mahesh, S., Trumbach, C. C. y Walsh, K. R. (2012). Visualizing technology mining results on life cycle axes: a study of server virtualization. *Journal Information- knowledge Systems Management*, 11(3), 321-343. doi:10.3233/IKS-2012-0213
- Marco Rivadeneira, F. (2014, abril). *Implementation of computer laboratories in schools at low-cost with environmental perspective*. Ponencia presentada en la First International Conference on eDemocracy and eGovernment (ICEDEG), Quito, Ecuador. doi:10.1109/ICEDEG.2014.6819942
- Mârgescu, C. N. y Haller, P. (2013). Virtualization, the next step for online services. *Scientific Bulletin of the Petru Maior*, 10(1), 34-38.
- Márquez, A. (2011). *Virtualización de servidores* (Tesis de pregrado). Universidad Politécnica de Catalunya, Barcelona, España.
- Martín, D., Marrero, M., Urbano, J., Barra, E. y Moreiro, J. A. (2011). Virtualización, una solución para la eficiencia, seguridad y administración de intranets. *El Profesional de la Información*, 20(3), 348-354. doi:10.3145/epi.2011.may.16
- Microsoft. (2012). *Windows server 2012: Server virtualization*. Recuperado de [https://www.thomas-krenn.com/redx/tools/mb\\_download.php/mid.y5762c57b1eebd93f/WS\\_2012\\_White\\_Paper\\_Hyper-V.pdf](https://www.thomas-krenn.com/redx/tools/mb_download.php/mid.y5762c57b1eebd93f/WS_2012_White_Paper_Hyper-V.pdf)
- Microsoft. (2013). *Best practices for virtualizing and managing sql server*. Recuperado de [http://download.microsoft.com/download/6/1/D/61DDE9B6-AB46-48CA-8380-D7714C9CB1AB/Best\\_Practices\\_for\\_Virtualizing\\_and\\_Managing\\_SQL\\_Server\\_2012.pdf](http://download.microsoft.com/download/6/1/D/61DDE9B6-AB46-48CA-8380-D7714C9CB1AB/Best_Practices_for_Virtualizing_and_Managing_SQL_Server_2012.pdf)



- Misevičienė, R., Ambrazienė, D., Tuminauskas, R. y Pažereckas, N. (2012). Educational infrastructure using virtualization technologies: Experience at Kaunas University of technology. *Informatics in Education*, 11(2), 227-240.
- Noguiera, M. (2017). *Multitenant security and isolation with Hyper 2012*. Recuperado de <https://mdnoga.wordpress.com/category/hyper-v-server-2012/>
- Norte, J. C. y Fiestas, A. (2015). *Eyeos developer manual*. Recuperado de <http://documents.mx/documents/eyeos-developer-manual.html>
- Nvidia. (2015). *Nvidia jetson tk1 development kit: bringing gpu-accelerated computing to embedded systems*. Recuperado de [http://www.embedsolution.com/download/Jetson\\_platform\\_brief\\_Datasheet.pdf](http://www.embedsolution.com/download/Jetson_platform_brief_Datasheet.pdf)
- Patwardhan, S., Nair, J. y Paranjape, N. (2014). Embedded systems: scope in engineering education. *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(3), 712-714.
- Pearson, C. (2016). *Embedded system requirements*. Recuperado de [http://www.ehow.com/facts\\_5526848\\_advantages-embedded-system.html](http://www.ehow.com/facts_5526848_advantages-embedded-system.html)
- Periáñez Gómez, F. (2017). *VirtualBox*. Recuperado de [http://fpg.x10host.com/Virtual-Box/hipervisor\\_de\\_tipo\\_2.html](http://fpg.x10host.com/Virtual-Box/hipervisor_de_tipo_2.html)
- Revelle, D. (2011). Hypervisors and virtual machines: Implementation insights on the x86 architecture. *Usenix, The Advanced Computing Systems Association*, 36(5), 17-22.
- Rosero Vinueza, V. A. (2012). *Estudio de tecnologías informáticas para asegurar la continuidad de servicios de sistemas computacionales mediante virtualización* (Tesis de pregrado). Universidad Técnica del Norte, Ibarra, Ecuador.
- Roy, R. y Bommakanti, V. (2017). *Odroid-C2: User manual*. Recuperado de <http://magazine.odroid.com/category/manuals/>
- Ruiz Molina, F. A. (2014). *Sistema autónomo para la adquisición de variables analógicas basado en crio 9012 y con comunicación ethernet* (Tesis de pregrado). Universidad Nacional Autónoma de México, México.
- Salcedo Tovar, M. L. (2015). Minicomputador educacional de bajo costo raspberry pi: primera parte. *Revista Ethos Venezolana*, 7(1), 28-45.
- Scasso, M., Garro, R., Ordinez, L. y Alimenti, O. (2012, agosto). *Diseño e implementación de un sistema embebido de control moderno: una experiencia práctica*. Ponencia presentada en el Congreso Argentino de Sistemas Embebidos, Buenos Aires, Argentina.

- Schuurman, D. C. (2015). Introducing open source and the raspberry pi to schools in developing nations. *Perspectives on Science y Christian Faith*, 67(1), 50-53.
- Sevilla Rodríguez, J. L. (2017). *Instalando un sistema de detección de intrusos inalámbrico (WIDS) en raspbian - 1*. Recuperado de <https://revista.seguridad.unam.mx/node/2207>
- Sotaminga Reyes, M. B., Guerrero Vallarezo, C. L. y Abad Eras, A. E. (2011). *Implementación de un ambiente de virtualización para el manejo de múltiples servidores de VoIP sobre una plataforma común de hardware* (Tesis de pregrado). Escuela Superior Politécnica del Litoral, Guayaquil, Ecuador.
- Syazwani S. y Choong-Yeun, L. (2012). Perception of students on services at the computer laboratory: A case study at the school of mathematical sciences, University Kebangsaan Malaysia. *Procedia Social and Behavioral Sciences*, 59, 117-124. doi: 10.1016/j.sbspro.2012.09.254
- Tanenbaum, A. S. y Wetherall, D. J. (2012). *Redes de computadoras*. México: Pearson Educación.
- Thomsen, A. (2017). *Primeiros passos com o raspberry pi*. Recuperado de <http://blog.filipeflop.com/embarcados/tutorial-raspberry-pi-linux.html>
- Tosini, M. A., Todorovich, E., Vázquez, M., Leiva, L., Aciti, C., Marone, J., . . . Kornuta, C. (2013, junio). *Metodologías de diseño para sistemas embebidos*. Memorias presentadas en el XV Workshop de Investigadores en Ciencias de la Computación, Paraná, Argentina.
- Tudevtagva, U., Ayush, Y. y Baatar, B. (2014). The virtual laboratories case study in traditional teaching and e-learning for engineering sciences, *Ubi-media Computing and Workshops*, 281-285. doi:10.1109/U-MEDIA.2014.66
- UNESCO. (2015). *Enseñanza de la ciencia y la tecnología*. Recuperado de <http://www.unesco.org/new/es/education/themes/strengthening-education-systems/science-and-technology/>
- Velásquez, A. (2009). *Tecnología Pyme. ¿Qué es la virtualización?*. Recuperado de <http://publicaciones.urbe.edu/index.php/telematique/article/viewArticle/3224/html>
- Villar Fernández, E. E. y Gómez López, J. (2010). *Virtualización de servidores de telefonía ip en GNU/Linux* (Tesis de pregrado). Universidad de Almería, Almería, España.

VMware, Inc. (2016). *Vmware vsphere 6 fault tolerance: Architecture and performance*. Recuperado de <http://www.vmware.com/files/pdf/techpaper/VMware-vSphere6-FT-arch-perf.pdf>

Zhixin, T. (2013). Mobile-C based agent system for detecting improper computer usage at computer laboratories. *Journal of Software*, 2262-2271. doi:10.4304/jsw.8.9.2262-2271