

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

Implementación de sistemas multi-asiento
Proyecto de Investigación

Marco Alejandro Borja Recalde

Ingeniería en Sistemas

Trabajo de titulación presentado como requisito
para la obtención del título de
Ingeniero en Sistemas

Quito, 21 de diciembre del 2018

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ
COLEGIO DE CIENCIAS E INGENIERÍAS

**HOJA DE CALIFICACIÓN
DE TRABAJO DE TITULACIÓN**

Implementación de sistemas multi-asiento

Marco Alejandro Borja Recalde

Calificación:

Nombre del profesor, título académico:

Daniel Riofrío, Doctor of Philosophy in
Computer Science

Firma del profesor:

Quito, 21 de diciembre del 2018

Derechos de Autor

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Firma del estudiante: _____

Nombres y apellidos: Marco Alejandro Borja Recalde

Código: 115824

Cédula de Identidad: 1723158356

Lugar y fecha: Quito, 21 de diciembre del 2018

RESUMEN

En el mundo moderno tenemos acceso a tecnologías muy avanzadas como la computadora personal, la cual, si bien es cierto es muy común hoy en día, no por eso se puede menospreciar todo el desarrollo que conllevó su creación. Los usuarios de dichas máquinas, es poco probable que ocupen todo el potencial de ellas, dejando subutilizadas a sus estaciones de trabajo en la mayoría de los casos. Se propone sistemas multi-asiento que permitan la utilización de este potencial latente, y a la vez, una reducción de costos en construcción de laboratorios o entornos en los que se necesite más de una máquina para servir a dos usuarios que no tengan requerimientos mayores a los de oficina y entretenimiento en línea, dejando de lado por el momento la discusión de sistemas “gamer” con configuraciones de multi-asiento.

Palabras clave: multi-asiento, hardware, terminales, laboratorios de computación, bajo costo

ABSTRACT

In the modern world, people have access to advanced technologies such as the personal computer, although very common nowadays, its creation involved many work hours in research and development. PC users will not usually make full use of their work stations, leaving them underutilized most of the times. In this research, I propose to use multi-seat systems to allow better hardware utilization, while achieving at the same time cost reduction when building computing centers, or any scenario in which many terminals are needed but they are within a tight budget. Office work will be considered, and at most, web browsing; gaming multi-seat rigs are not part of the analysis and will not be discussed in this work.

Keywords: multi-seat, hardware, terminals, computing centers, low cost

TABLA DE CONTENIDO

Introducción	9
Motivación	9
Objetivo general.....	9
Objetivos específicos.....	9
Organización	10
Desarrollo	11
Marco teórico.....	11
Sistemas operativos	11
Bases conceptuales de un sistema multi-asiento	11
Estado del arte	12
Soluciones propuestas	13
Elección del OS, ¿por qué Arch Linux?.....	13
Instalación.....	14
Configuración	14
Discusión financiera	29
Descripción del escenario	29
Caso multi-cabecera	30
Caso multi-tarjeta	31
Caso híbrido	31
Comparación financiera multi-cabecera, multi-tarjeta, híbrido.....	32
Caso Raspberry Pi	33
Resumen comparativo	34
Escenario alternativo	34
Conclusiones y trabajo futuro	35
Conclusiones	35
Trabajo futuro	35
Bibliografía	36

ÍNDICE DE TABLAS

Tabla 1: Costo de máquinas para el laboratorio propuesto	32
Tabla 2: Costo de tarjetas para el laboratorio propuesto.....	32
Tabla 3: Costos totales para el laboratorio propuesto	33
Tabla 4: Costo de componentes para sistemas sobre Raspberry Pi	34
Tabla 5: Resumen comparativo	34

ÍNDICE DE FIGURAS

Figura 1: Flujo de un comando como superusuario	15
Figura 2: Contenido del archivo /etc/netctl/enp2s0	15
Figura 3: Contenido de /etc/gdm/custom.conf	16
Figura 4: Funcionamiento de configuración multi-cabecera	17
Figura 5: Funcionamiento de configuración multi-tarjeta	18
Figura 6: Activación de XDMCP	19
Figura 7: Activación de TCP	19
Figura 8: greeter.....	21
Figura 9: Escritorio awesome.....	21
Figura 10: Resultados de xrandr	22
Figura 11: Ventana en Xephyr.....	23
Figura 12: terminator corriendo en Xephyr	23
Figura 13: Resultado de “xinput list”	24
Figura 14: “xinput list” con identificador creado	25
Figura 15: Resultado multi-asiento una tarjeta	25
Figura 16: Programa de UnRAID para creación de un USB arrancable.....	27
Figura 17: Pantalla de inicio del servidor UnRAID	28
Figura 18: Sistema multi-tarjeta	28

INTRODUCCIÓN

Motivación

La Electronic Numerical Integrator and Computer (ENIAC) terminó de ser construida en 1946, y fue comisionada por el gobierno de Estados Unidos en la Segunda Guerra Mundial a un costo de 400,000 USD (Freiberger & Swaine, 2017), lo que, ajustado a la inflación, actualmente es equivalente a 5'572,287.29 USD. Para junio de 1978, nace un estándar internacional; el procesador Intel 8086 fue el primero de 16 bits, pero lo crucial en él yace en la arquitectura que manejaba, la cual era suficientemente flexible y eficiente (Intel Newsroom, 2018) como para dominar el mercado hasta 1999 (Kerner, 2007). A partir de esta fecha AMD será el que, con su nueva arquitectura AMD64, marque el paso de las computadoras que utilizamos hasta hoy en día. El precio del primer procesador Intel 8086 según TechRadar, era de 360 USD (Sheesly, 2008), 1,392.22 USD actualmente, no obstante, representa un gran salto del exorbitante costo de la ENIAC, además el 8088, que es una versión más barata del 8086 (Sheesly, 2008) vendría a formar parte de las computadoras personales vendidas por IBM a partir del 1981 (PCMag, 2018). La frecuencia del primer Intel 8086 era de 5MHz en un solo núcleo (Intel, 1989).

Actualmente las capacidades de hardware han sobrepasado aquellas de 1978, y como ejemplo, el AMD Ryzen Threadripper 2990WX posee 32 núcleos, cada uno trabajando hasta una frecuencia máxima de 4.2GHz (AMD, 2018) y se lo encuentra en Amazon a 1,750 USD (Amazon, 2018), siendo 84,000 veces más rápido por núcleo que el Intel 8086. Al tener esta diferencia tan abrupta, como es lógico, se puede pensar que este procesador soporta cientos de miles de usuarios de 1978, pero esto qué significa en el mundo actual, y, sobre todo, ¿qué implicaciones útiles podemos sacar de este hecho?

Considerando esta aparente sobredimensión del hardware actual y la versatilidad de los sistemas operativos basados en Unix, decidí investigar de más cerca la posibilidad de que dos o más usuarios utilizaran la misma máquina al mismo tiempo y compartieran todos los recursos que una computadora personal tiene para ofrecer, efectivamente transformando una estación de trabajo en dos, tres, o más dependiendo de las limitaciones físicas.

Objetivo General

Investigar alternativas económicas de implementar sistemas multi-asiento.

Objetivos Específicos

- Analizar el estado del arte de los sistemas multi-asiento tanto en hardware como en software.
- Implementar un sistema multi-asiento en configuración multi-cabecera

- Implementar un sistema multi-asiento en configuración multi-tarjeta
- Implementar un sistema multi-asiento en configuración híbrida
- Realizar una comparación técnica y económica entre los diferentes sistemas

Organización

El presente trabajo consiste en cuatro partes principales. Como apertura se presenta definiciones que serán empleadas a lo largo de todo el documento; además, se incluye una breve discusión de sistemas que compiten con el aquí propuesto. En el segundo punto se detalla la solución propuesta. Se desarrolla una guía paso a paso de cómo se puede implementar un sistema multi-cabecera (configuración básica principal), un sistema multi-tarjeta y finalmente, las consideraciones a tener en una implementación híbrida. En el tercer punto se analizará desde una perspectiva económica la factibilidad de los sistemas propuestos. Además, se agrega una opción adicional para discutir la factibilidad de estos sistemas. Finalmente, se cierra el tema con conclusiones basadas en dificultades encontradas a lo largo del camino, resultados obtenidos y opiniones respecto a la eficiencia y uso real de ellos.

DESARROLLO DEL TEMA

Marco Teórico

Sistemas operativos

Windows y Linux son dos sistemas operativos rivales con una larga historia en la industria y en el hogar, siendo Windows el dominante en computadoras personales por su distintivo GUI que ha perdurado por generaciones (Hellard, B., & Hopping, C, 2018). Linux por su parte tiene ventajas significativas sobre Windows en aspectos como la seguridad, bajos requerimientos de hardware en la mayoría de sus distribuciones, y flexibilidad (Hellard, B., & Hopping, C, 2018), pero puede ser un tanto sobrecogedor para usuarios con poca, o sin experiencia en esta línea de sistemas operativos. En lo que respecta al trabajo presente se considera a Arch Linux contra Microsoft Server 2016.

El caso de Arch será discutido más a profundidad en párrafos posteriores, pero a breves rasgos se puede decir que es un sistema minimalista con posibilidades de customización únicamente limitadas por la habilidad del usuario. Para Microsoft Server se decidió comparar contra la versión 2019, ya que a partir de su versión 2016 MultiPoint viene como parte de los servicios incluidos, que es la solución de Microsoft para sistemas multi-asiento (Microsoft, 2016).

Arch y Server 2016 son sistemas operativos complejos, el primero al necesitar una configuración desde cero, y el segundo al ser software profesional, destinado a casos de uso más complejos que servicios domésticos. Una de las principales desventajas de Server 2019 yace en su costo, la que, incluso en su versión más baja (Essentials), su costo es de \$501 (Pricing and licensing for Windows Server 2019, 2018) en comparación a Arch que es gratis.

Bases conceptuales de un sistema multi-asiento

Durante la época de 1960, IBM revoluciona el mundo computacional con su línea Systems/360, el cual es una familia de mainframes (IBM, 2009). Un mainframe es una computadora muy grande, burdamente dicho, la cual puede poseer una o varias unidades de procesamiento, cantidades grandes (dependiendo de las necesidades) de memoria de sistema, varias unidades de almacenaje, así como otros componentes hardware que se adaptarán a los requerimientos del sistema (como por ejemplo tarjetas gráficas, tarjetas de red, tarjetas para caching, etc); además, estos sistemas pueden ser expandibles, dando la posibilidad de aumentar hardware.

La idea de una unidad de una unidad de procesamiento para un sistema no es nada nuevo al mundo tecnológico, cuando los costos de computadoras personales eran demasiado

altos se recurría a la compartición de una sola unidad de procesamiento para varias personas, y por supuesto esto volvía necesario un horario de trabajo para cada grupo, no obstante esto cambiaría con la mejora tecnológica, la ley de Moore y la caída de costos de hardware, así como la creación de nuevas capas de abstracción en el mundo del software.

Estado del arte

Los sistemas multiasiento guardan muchas similitudes con modelos computacionales cliente-servidor, ya que en ambos casos existen varios clientes conectándose a servicios en el servidor. Si los nombramos uno por uno vendrían a ser:

- Servicio de autenticación
- Greeter
- Servicio de interfaz gráfica
- Servicio de almacenamiento

Existen más tipos de servicios, pero los básicos para el tema en cuestión se reducen a los antes mencionados.

Con la venida del procesador de Intel 8086 y su versión “barata”, el 8088 (Kerner & Padgett, 2007) se sentarían las bases para la computación personal. Para el año 2013, según el Instituto Nacional de Estadísticas y Censos (INEC), 27.5% posee una computadora de escritorio en su hogar, y el 18.1% una portátil (2013), ambas, parte de la categoría de computadoras personales (PC).

A nivel industrial existen varias soluciones que brindan a la empresa la posibilidad de tener un control centralizado de soluciones tecnológicas, y muchos de estos se pueden resumir al pensar en una unidad de procesamiento fuerte, a la que se conectan varias terminales “tontas”. No obstante, el motivo de este trabajo es encontrar soluciones de bajo costo con miras a micro, pequeña y mediana empresa; instituciones educativas y gubernamentales; y varios otros clientes que se encontrarían interesados en aprovechar de manera más eficiente sus soluciones tecnológicas en pie, o en planes de compra, y es así que para el caso se comparará los sistemas multi-asiento con varias metodologías que pueden brindar un servicio similar como son:

- Escritorio Remoto
- Escritorio Virtual
- Thin Clients
- Zero Clients (Ultra Thin Clients)

En lo que respecta a la academia, hay dos casos en los que hacen referencia explícita a sistemas multi-asiento. El primer documento viene de la Universidad de Michigan, pero su tema principal es la interconexión entre escuelas de Tanzania y al internet, utilizando la red satelital y radiofrecuencia (García et al., 2010), pero entre los elementos ya existentes en

estas escuelas se encuentran sistemas multi-asiento, y aunque no se profundiza en su implementación ni hay mayor discusión acerca de ellos, es un buen punto de referencia para uno de los casos de uso: escuelas de bajos recursos.

El segundo paper, Hernández y Soler presentan un trabajo específicamente destinado a estaciones “de trabajo multi-asiento” (2012), y los autores discuten varias implementaciones existentes al momento de su investigación. Uno de los sistemas mencionados es Useful Multiplier, que hasta el 2010 se encontraba en su versión 4.0 (Hernandes & Soler, 2012); en su página oficial Useful Multiplier se cataloga como discontinuado (Online Manuals, 2013) en favor de Useful Multiseat, sistema operativo que según su página fue lanzado, su última versión, en abril del 2012 y el sistema operativo sobre el que trabaja es Ubuntu/Edubuntu 10.06 (Useful Multiseat Release Notes, 2015). Ubuntu actualmente se encuentra en su versión 18.04 LTS (Ubuntu, 2018) por lo que resulta evidente que este sistema operativo también se encuentra discontinuado, aunque no se da razones explícitas de por qué se tomó esta decisión.

Soluciones propuestas

Elección de OS, ¿por qué Arch Linux?

Arch Linux comenzó a ser desarrollado en el 2001, por Judd Vinet (Arch Linux, 2018), y los objetivos de dicho sistema como lo define la Wiki de Arch son: simplicidad, modernidad, pragmatismo, centrado en el usuario, y versatilidad (2018). En una reseña hecha por Fabio Akita, el autor menciona como, una de las principales ventajas, el nivel de dificultad que Arch tiene para el usuario, que si bien es cierto no es amigable para aquellos completamente nuevos a Linux, tampoco requiere un conocimiento extenso de su arquitectura para lograr una mínima funcionalidad del sistema (Akita, 2017).

Arch Linux se considera a sí mismo un sistema que baila en el filo de la espada (Arch Linux, 2018), por lo que el software que se encuentra en los repositorios oficiales de ellos generalmente serán los últimos en ser liberados, pero como también Akita lo menciona, existe un gracioso equilibrio entre “bleeding edge” y estable, por lo que es poco probable que el sistema quede inútil después de una actualización utilizando los repositorios oficiales. Pero no todo es color de rosa con este OS, y hay problemas.

Arch, si bien es cierto, tiene objetivos coherentes con un mundo en el que no todos son expertos tecnológicos, tampoco es completamente amigable y cierto grado de conocimiento es necesario, por lo tanto aunque hay información extensa del sistema, en el campo de Wikis, Ubuntu reina por mucho sobre él y muchas otras distribuciones Linux, lo que nos lleva a cómo enfrentar el problema de traducir un sistema multi-asiento de algunos “How to” en Ubuntu, a Arch.

Instalación

Para Arch Linux es de mucha ayuda que en su sitio oficial se encuentren páginas amigables con el usuario que señalan una serie de pasos para levantar una máquina desde cero; desde cómo configurar correctamente la fecha; crear, formatear, y montar las particiones de disco necesarias; crear el `fstab` con los identificadores de discos; hasta la instalación del `bootloader` (Installation Guide, 2018), dando en cada la posibilidad de customizar el sistema de acuerdo con las necesidades del usuario.

Entre las decisiones de qué paquetes usar, tenemos a GRUB como *bootloader* en conjunción con *efibootmgr* para manejar sistemas UEFI, y se realizó la instalación de *sudo* para manejo del nivel de acceso de usuarios; *w3m* como navegador en línea; *tmux* como contenedor de consolas; *vim* como editor de textos; *Xorg* como *display server*; *Xephyr* como *nested display server*; y finalmente *Intel-ucode* para actualización de microcódigo en procesadores Intel; todos de los repositorios oficiales de Arch. Cabe mencionar que el manejador de paquetes de Arch es *pacman*, y está basado en `tar`, que a su vez está programado en C (pacman, 2018); la instrucción para dicha instalación es la siguiente:

```
pacman -S grub efibootmgr w3m tmux vim xorg-server xorg-server-xephyr intel-ucode
```

El argumento “-S” es para sincronizar esos paquetes de los repositorios al sistema.

Configuración

1. Inicial

Se procede a la creación de un usuario administrador junto con una carpeta de este, se lo puede realizar con el comando:

```
sudo useradd -m {nombre_de_usuario}
```

Como se mencionó en un principio, es necesaria la instalación del paquete *sudo*, y para configurarlo se emplea el comando “*visudo*” que utilizará *vi* como editor por defecto. En el archivo de configuración se añade la línea:

```
Default editor=/usr/bin/vim
```

Para utilizar *vim* como editor y se añade una entrada debajo de la de `root` con el nombre del usuario que tendrá privilegios de superusuario. Cuando se corre un comando con `sudo` se tiene un flujo como en la Figura 1.

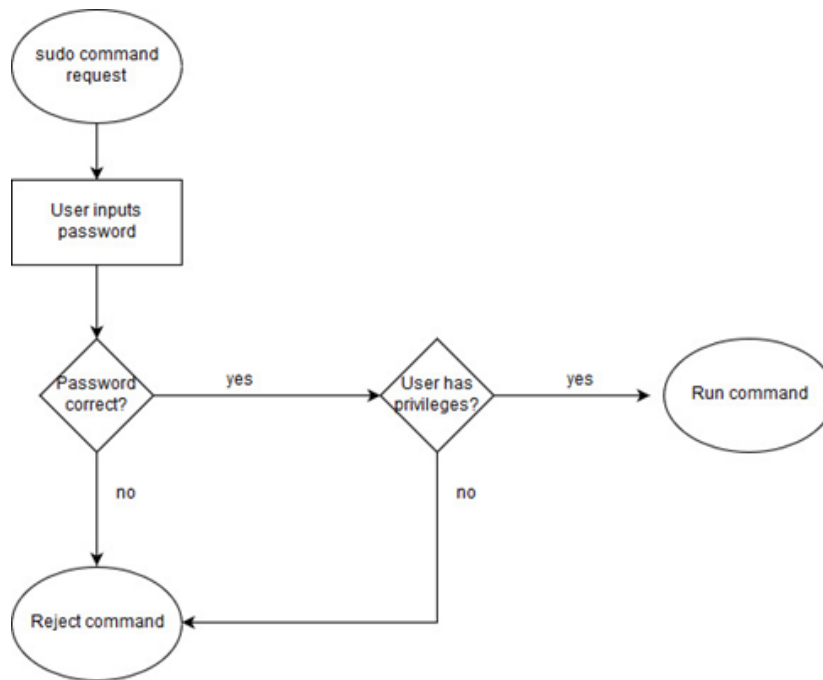


Figura 1: Flujo de un comando como superusuario

Posteriormente se procede a deshabilitar el inicio de sesión de root con el siguiente comando:

`passwd root -l`

Donde el argumento “-l” sirve para bloquear la contraseña de dicha cuenta como lo dice el manual de “passwd” (die.net, 2017).

2. Red

Debido a que es un servidor, es necesario que en todo momento se conozca la dirección IP de este, por lo que se necesita tener una dirección estática conocida, esto se lo puede realizar de diferentes maneras, pero se decidió utilizar “netctl”, y su configuración consiste en la creación de un archivo en la carpeta /etc/netctl/ con el contenido de la Figura 2.

```

Description='Static address on ethernet adapter'
Interface=enp2s0
Connection=ethernet
IP=static
Address=('192.168.100.94/24')
Gateway=('192.168.100.1')
DNS=('192.168.100.1' '1.1.1.1')
  
```

Figura 2: Contenido del archivo /etc/netctl/enp2s0

Para activarlo en el arranque de la computadora se crea una entrada en los servicios de inicio con el siguiente comando:

```
netctl enable enp2s0
```

Finalmente se habilita el servicio de OpenSSH para consolas remotas con el comando:

```
systemctl enable sshd
```

y se procede a reiniciar la máquina.

3. Opciones de Display Manager

Gnome Display Manager: Para instalarlo se utiliza el comando:

```
pacman -S gnome
```

Gnome viene por defecto activado para utilizar *Wayland* como *display server*, no obstante, este protocolo no tiene la posibilidad de realizar configuraciones multi-sesión así que es necesario cambiar a que utilice Xorg, se lo puede hacer en los archivos de configuración ubicados en `/etc/gdm/custom.conf` y habilitar la línea que habilita o deshabilita *Wayland*.

```
# GDM configuration storage

[daemon]
# Uncomment the line below to force the login screen to use Xorg
WaylandEnable=false

[security]

[xdmcp]

[chooser]

[debug]
# Uncomment the line below to turn on debugging
#Enable=true

/etc/gdm/custom.conf (END)
```

Figura 3: Contenido de `/etc/gdm/custom.conf`

Para comprobar si la modificación fue exitosa se inicia el entorno gráfico con:

```
sudo systemctl start gdm
```

en la pantalla de inicio de sesión se pone las credenciales del usuario administrador; se inicia una consola en el entorno gráfico y se tipea:

```
echo $XDG_SESSION_TYPE
```


se debe obtener como resultado “x11”.

LightDM: Es uno de los Display Managers (DM) más recomendados para realizar este tipo de configuraciones, usa por defecto Xorg por lo que no requiere configuración adicional, no obstante, hay que tener en cuenta que es necesario instalar un Greeter y un Window Manager (WM). El primero entra en efecto al iniciar la máquina y presenta al usuario una pantalla de login; el segundo, una vez realizado un login exitoso se activa y toma el control de X server, presentando al usuario con un entorno gráfico en el que se pueden crear ventanas de los diferentes programas que se abran.

4. Opciones para sistemas multi-asiento:

Para el proyecto se consideraron 2 configuraciones. La primera utiliza cualquier tarjeta de video o incluso una placa con GPU integrado y varias cabeceras de salida de video; la segunda consiste en una configuración con una tarjeta de video por usuario. Se discute a continuación algunas ventajas y desventajas de cada configuración.

Para la primera opción las ventajas económicas son evidentes ya que se puede hacer un uso eficiente de todos los puertos y el poder de procesamiento de video que el GPU pudiera tener al repartirlo entre los múltiples asientos, pero este beneficio no viene sin su costo. Actualmente esta configuración es compleja técnicamente ya que se necesita una modificación pesada de *Xorg* y combinarlo con *Xephyr* para poder lograrlo. El funcionamiento de dicho sistema se resume en la Figura 4.

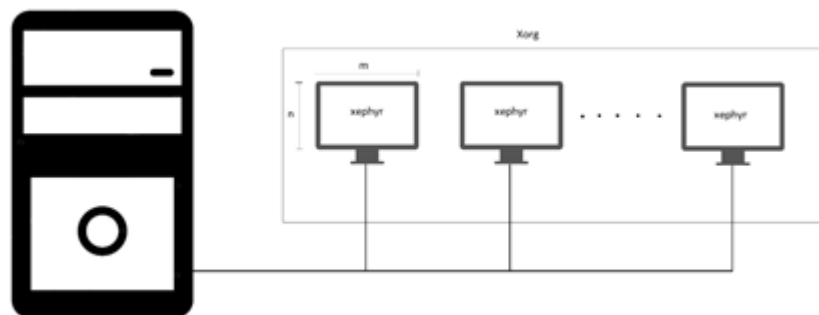


Figura 4: Funcionamiento de configuración multi-cabecera

Xorg toma la longitud total de las pantallas; y de la pantalla más grande, la distancia vertical, y se realiza una configuración de escritorio extendido; para mejor comprensión se puede considerar una pantalla de 1920x1080 junto con otra de 1440x900, entonces el tamaño de pantalla horizontalmente será de $1920+1440=3360$, mientras que verticalmente tendremos que es $\max(1080,900)=1080$.

Xephyr es un servidor X anidado (Xephyr, 2018), y se utilizará para crear cada asiento como una ventana dentro de *Xorg*, además de esto, *xephyr* nos permitirá tomar control independiente de cada dispositivo de entrada (ratón y teclado) para cada sesión/ventana.

La segunda configuración es mucho más sencilla técnicamente, pero es considerablemente más costosa dependiendo del hardware adquirido. El motivo por el que esta configuración es más sencilla es que *Xorg* tiene nativamente protocolos para manipular múltiples tarjetas y vincular dispositivos I/O a ellas; se inician servidores X en las diferentes tarjetas y se le da el control de la pantalla conectada. A diferencia de la primera opción, ésta tiene un stack como la Figura 5.

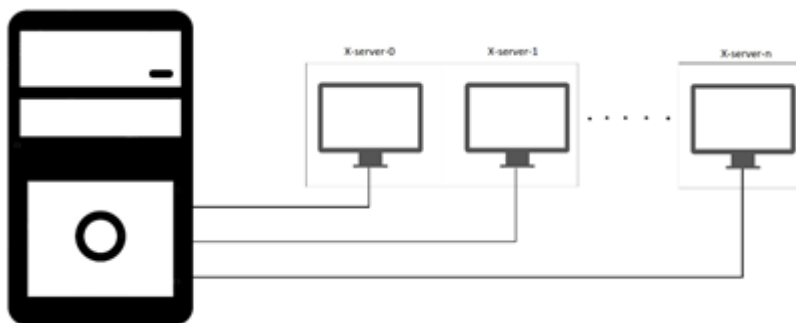


Figura 5: Funcionamiento de configuración multi-tarjeta

5. Instalación multi-cabecera

De acuerdo con el esquema antes mencionado se sabe que en un inicio es necesario que el servidor Xorg tome control de todas las pantallas en modo extendido, de modo que posteriormente Xephyr venga a anidarse dentro de él creando sesiones independientes. Para lograr este objetivo existen dos cosas que se deben realizar:

- Habilitar XDMCP
- Habilitar conexión TCP

XDMCP (por sus siglas en inglés X Display Manager Control Protocol) permite el acceso remoto a los servicios de X, este es un protocolo muy usado para el inicio de sesiones remotas y es el equivalente a Microsoft Remote Desktop Protocol (Cygwin/X, 2018). El propósito de habilitar XDMCP es justamente permitir a Xephyr trabajar sobre la interfaz loopback del sistema.

Transfer Control Protocol (TCP) por otro lado, da la posibilidad de conexión, incluso para conexiones sobre loopback, si este protocolo no se ha habilitado, a pesar de que X esté esperando conexiones remotas, no habrá manera de llegar a él sin TCP.

En el sistema de prueba se decidió utilizar LightDM como DM, GTK como greeter y awesome como WM. Lo primero que se debe hacer es lo que se mencionó anteriormente, activar XDMCP y TCP en el archivo de configuración del DM, el cual se encuentra (para LightDM) en `/etc/lightdm/lightdm.conf`

```

#
# XDMCP Server configuration
#
# enabled = True if XDMCP connections should be allowed
# port = UDP/IP port to listen for connections on
# listen-address = Host/address to listen for XDMCP connections (use all addresses if not present)
# key = Authentication key to use for XDM-AUTHENTICATION-1 or blank to not use authentication (store
# in keys.conf)
# hostname = Hostname to report to XDMCP clients (defaults to system hostname if unset)
#
# The authentication key is a 56 bit DES key specified in hex as 0xxxxxxxxxxxxxxx. Alternatively
# it can be a word and the first 7 characters are used as the key.
#
[XDMCP:Server]
enabled=true
port=177
#listen-address=
#key=
#hostname=

```

Figura 6: Activación de XDMCP

Para habilitar TCP se lo puede hacer para todos los asientos o para uno en específico; cada asiento se identifica con la cabecera “[Seat:x]”, donde x es el número de asiento, para habilitarlo para todos se pone un asterisco, y así tenemos la Figura 7.

```

[Seat:*)
#type=local
#pan-service=lightdm
#pan-auto-login-service=lightdm-auto-login
#pan-greeter-service=lightdm-greeter
#xserver-backend=
#xserver-command=X
#xmir-command=Xmir
#xserver-config=
#xserver-layout=
xserver-allow-tcp=true
#xserver-share=true
#xserver-hostname=
#xserver-display-number=
#xdmcp-manager=
#xdmcp-port=177
#xdmcp-key=
#unity-compositor-command=unity-system-compositor
#unity-compositor-timeout=60
greeter-session=lightdm-gtk-greeter
#greeter-hide-users=false
#greeter-allow-guest=true
#greeter-show-manual-login=false
#greeter-show-remote-login=true
#user-session=default
#allow-user-switching=true
#allow-guest=true
#guest-session=
session-wrapper=/etc/lightdm/Xsession
#greeter-wrapper=
#guest-wrapper=
#display-setup-script=
#display-stopped-script=
#greeter-setup-script=
#session-setup-script=
#session-cleanup-script=

```

Figura 7: Activación de TCP

Se puede ver 4 líneas activadas (descomentadas), la primera es el nombre del asiento al que afecta (en este caso es general), la segunda permite conexión TCP, la tercera define el greeter a utilizarse, y finalmente la cuarta es el archivo ejecutable que sigue LightDM para iniciarse y funcionar.

Finalmente se crea un usuario administrador que será el encargado de iniciar la sesión master, crear las sesiones adicionales correspondientes y asignar los dispositivos periféricos necesarios. Para el caso se creó el usuario session-manager con el siguiente comando:

```
useradd -m session_manager
```

adicionalmente a esto se debe dar permisos de superusuario modificando el archivo de "*sudoers*" con el comando:

```
visudo
```

se abrirá el archivo y se busca la ubicación de "*root ALL=(ALL) ALL*" y debajo de esta línea se aumenta para el caso:

```
session_manager ALL=(ALL) ALL
```

para probar si el usuario se creó correctamente se realiza un comando *sudo* usando el usuario creado, y se sigue una serie de comandos como los siguientes:

```
sudo su session_manager
```

```
sudo pacman -Syyu
```

que es lo que se utilizaría para una actualización completa del sistema. Para terminar, se le asigna una contraseña con el comando

```
passwd session_manager
```

Realizadas estas configuraciones se procede a iniciar el servicio completo con el siguiente comando:

```
systemctl start lightdm
```

se presentará el greeter (GTK), se inicia sesión con el usuario antes creado y se verifica que todo esté corriendo normalmente.

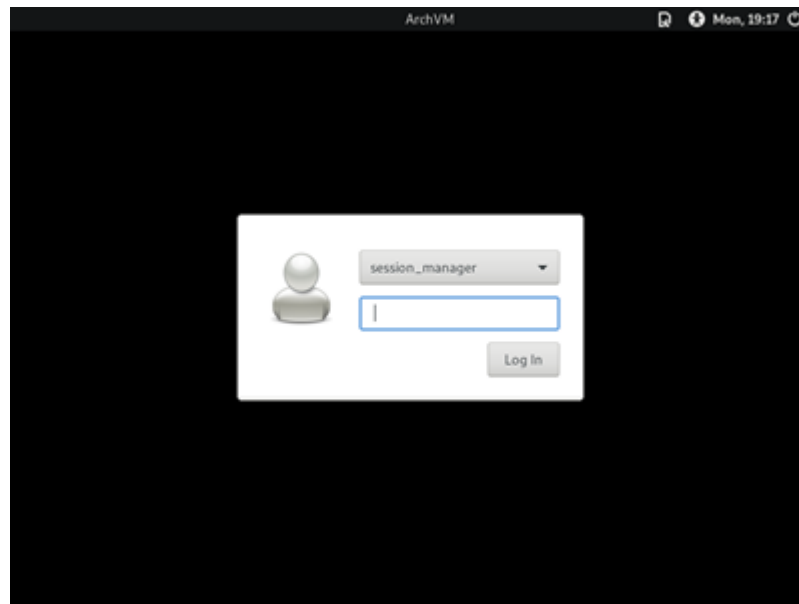


Figura 8: greeter

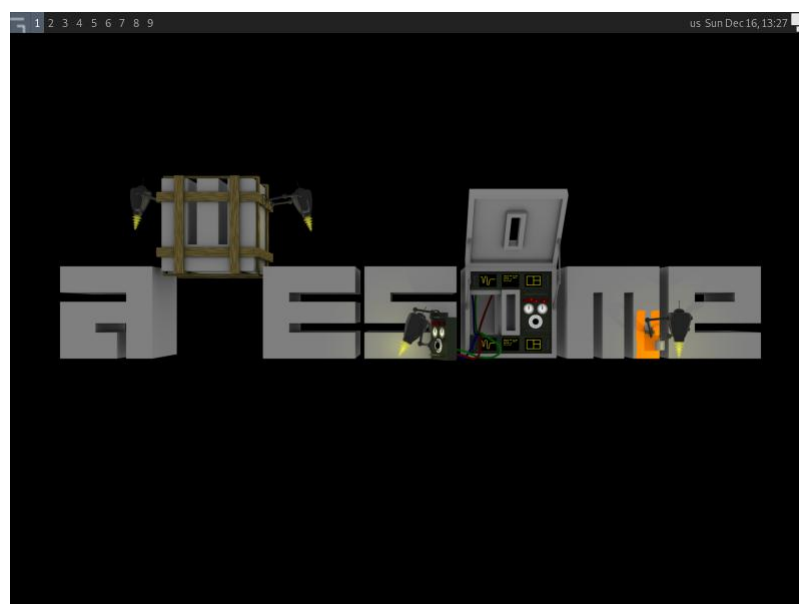


Figura 9: Escritorio awesome

Se podría realizar el inicio de sesión y configuración con el usuario administrador antes creado, pero como buena práctica se recomienda crear un usuario específico para administrar el entorno gráfico de los asientos.

Una vez en la interfaz gráfica, se procede a obtener información de las pantallas conectadas y su geometría con el comando `xrandr` como se muestra en la Figura 10:

```

Screen 0: minimum 320 x 200, current 800 x 600, maximum 16384 x 16384
VGA-1 connected primary 800x600+0+0 (normal left inverted right x axis y axis) 0mm x 0
mm
800x600 59.96*+ 60.32 56.25
2560x1600 59.99 59.97
1920x1440 60.00
1856x1392 60.00
1792x1344 60.00
2048x1152 60.00
1920x1200 59.88 59.95
1920x1080 60.00
1600x1200 60.00
1680x1050 59.95 59.88
1400x1050 59.98 59.95
1600x900 60.00
1280x1024 60.02
1440x900 59.89 59.90
1280x960 60.00
1366x768 59.79 60.00
1360x768 60.02
1280x800 59.81 59.91
1280x768 59.87 59.99
1280x720 60.00
1024x768 60.00

```

Figura 10: Resultados de xrandr

La primera línea nos da información relacionada a la pantalla principal, y como se puede ver en términos generales consiste de la resolución a la que se está trabajando en ese momento seguido de dos coordenadas más:

$$\{ancho\}x\{largo\}+x+y$$

esto es importante ya que los asientos que creemos tienen que ser desplazados el ancho de las pantallas que le preceden, entonces si tenemos una pantalla de 1920x1080 como principal, una de 1440x900 y otra de 3280x2160, entonces la geometría y ubicación de la secundaria donde estaría el primer asiento vendría a ser:

$$1440x900+1920+0$$

y la del segundo asiento:

$$3280x2160+3360+0$$

con estas consideraciones podemos probar cómo funciona Xephyr con una serie de comandos:

$$Xephyr :1 -screen 320x200+0+0$$

dando un resultado demostrado en la Figura 11:

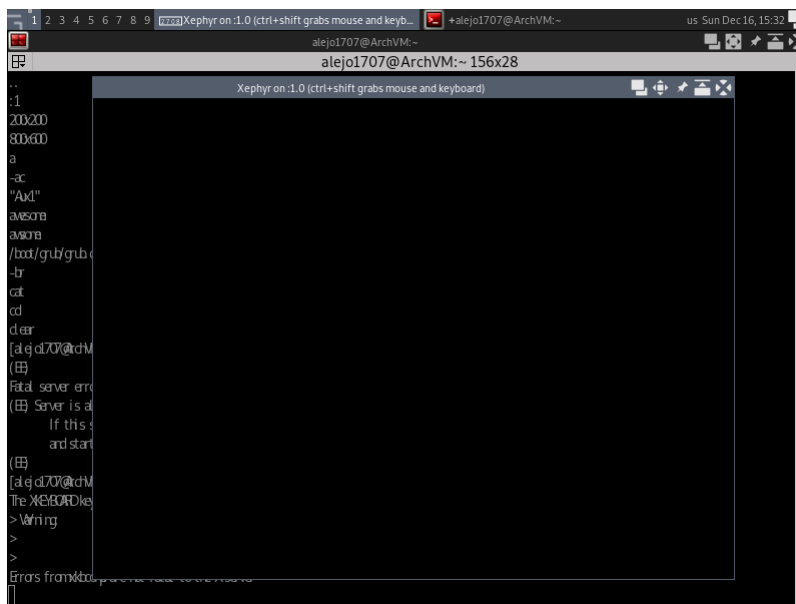


Figura 11: Ventana en Xephyr

ahora, en otra consola se escribe:

DISPLAY :1 terminator

y su resultado debe ser:

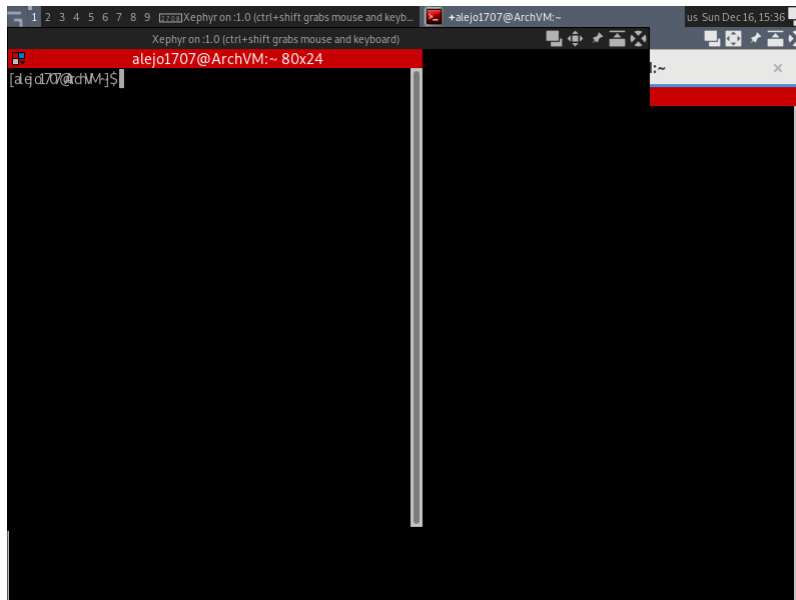


Figura 12: terminator corriendo en Xephyr

En este momento queda claro lo que un servidor X anidado significa, y, por ende, el funcionamiento de Xephyr sobre Xorg. Se procede a cerrar la ventana con lo que terminará el comando que inició este servidor anidado y terminator.

Si todos los pasos anteriores funcionaron bien, se puede avanzar a crear una sesión completa anidada, y esto se logra mediante el comando:

```
Xephyr -output {pantalla} -query localhost :1
```

El argumento “-query localhost” sirve justamente para solicitar una conexión XDMCP a la interfaz loopback de la máquina, y es esto por lo que se realizaron las primeras configuraciones en el archivo de LightDM. Para completar la configuración es necesario asignar dispositivos periféricos al o los distintos asientos, para poder hacerlo se necesita instalar el paquete “xinput” con el comando

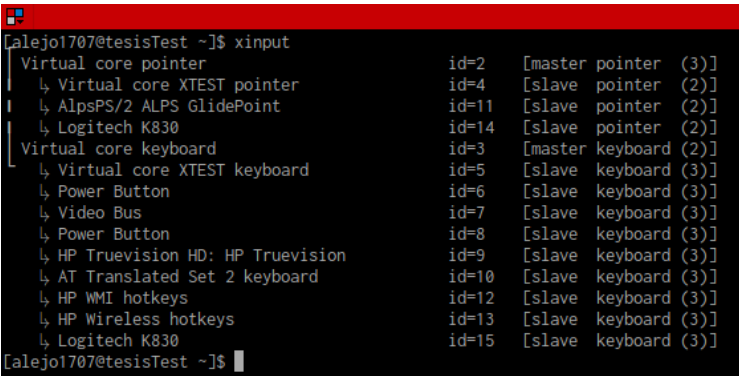
```
pacman -S xorg-xinput
```

“xinput” es una herramienta que permite manejar dispositivos al anclarlos a un identificador maestro, generando efectivamente un entorno de multi-entrada (varios ratones y teclados), necesario para la configuración multi-asiento.

Para poder identificar los dispositivos conectados corremos:

```
xinput list
```

lo que nos dará una lista como la que se muestra en la Figura 13:



```

calejo1707@tesisTest ~]$ xinput
Virtual core pointer                id=2    [master pointer (3)]
↳ Virtual core XTEST pointer        id=4    [slave pointer (2)]
↳ AlpsPS/2 ALPS GlidePoint          id=11   [slave pointer (2)]
↳ Logitech K830                      id=14   [slave pointer (2)]
Virtual core keyboard               id=3    [master keyboard (2)]
↳ Virtual core XTEST keyboard        id=5    [slave keyboard (3)]
↳ Power Button                       id=6    [slave keyboard (3)]
↳ Video Bus                           id=7    [slave keyboard (3)]
↳ Power Button                       id=8    [slave keyboard (3)]
↳ HP Truevision HD: HP Truevision     id=9    [slave keyboard (3)]
↳ AT Translated Set 2 keyboard        id=10   [slave keyboard (3)]
↳ HP WMI hotkeys                     id=12   [slave keyboard (3)]
↳ HP Wireless hotkeys                id=13   [slave keyboard (3)]
↳ Logitech K830                      id=15   [slave keyboard (3)]
calejo1707@tesisTest ~]$

```

Figura 13: Resultado de “xinput list”

Las entradas con el nombre de “Virtual Core XTEST” se pueden ignorar, y si hay algún dispositivo que no se pueda identificar se puede lo puede desconectar, ver la lista de *xinput* y volver a conectar para detectar el cambio.

Una vez identificado cada par de periféricos (ratón y teclado), se procede a crear un grupo para cada uno, esto se consigue con:

```
xinput create-master {nombre-master}
```

donde *nombre-master* es el identificador del par. Finalmente, para anclar los dispositivos se utiliza el comando:

xinput reattach {id-periférico} {nombre-master tipo}

donde tipo es “*keyboard*” o “*pointer*” dependiendo del caso. Con esto ya tenemos un teclado que responderá al foco de su respectivo ratón, y *xinput* se verá ahora de acuerdo a la Figura 14.

```
[alejo1707@tesisTest ~]$ xinput
Virtual core pointer
├─ Virtual core XTEST pointer          id=2   [master pointer (3)]
├─ AlpsPS/2 ALPS GlidePoint           id=4   [slave pointer (2)]
└─ AlpsPS/2 ALPS GlidePoint           id=11  [slave pointer (2)]
Virtual core keyboard
├─ Virtual core XTEST keyboard        id=3   [master keyboard (2)]
├─ Virtual core XTEST keyboard        id=5   [slave keyboard (3)]
├─ Power Button                       id=6   [slave keyboard (3)]
├─ Video Bus                           id=7   [slave keyboard (3)]
├─ Power Button                       id=8   [slave keyboard (3)]
├─ HP Truevision HD: HP Truevision     id=9   [slave keyboard (3)]
├─ AT Translated Set 2 keyboard       id=10  [slave keyboard (3)]
├─ HP WMI hotkeys                     id=12  [slave keyboard (3)]
├─ HP Wireless hotkeys               id=13  [slave keyboard (3)]
└─ Seat1 pointer                      id=16  [master pointer (17)]
    ├─ Logitech K830                  id=14  [slave pointer (16)]
    └─ Seat1 XTEST pointer            id=18  [slave pointer (16)]
Seat1 keyboard
├─ Logitech K830                      id=17  [master keyboard (16)]
├─ Logitech K830                      id=15  [slave keyboard (17)]
└─ Seat1 XTEST keyboard              id=19  [slave keyboard (17)]
[alejo1707@tesisTest ~]$
```

Figura 14: “*xinput list*” con identificador creado

Con esto es termina la configuración completa de Arch para sistemas multi-asiento, y el resultado final se presenta a continuación en la Figura 15.

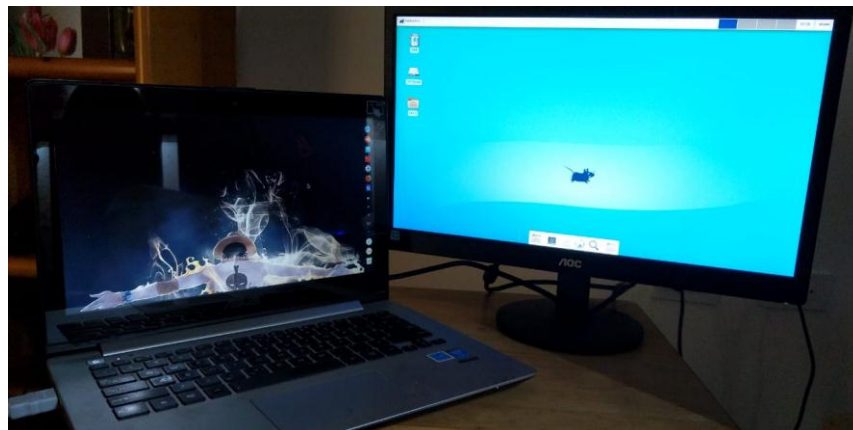


Figura 15: Resultado multi-asiento una tarjeta

6. Configuración multi-tarjeta

El sistema empleado es una computadora construida personalmente, consiste en un procesador Ryzen 7 2700X con un conteo de 8 núcleos (16 hilos) cada uno trabajando a 4.3GHz como máximo, 24GiB de RAM DDR4 a 2400MHz, y una tarjeta de video Nvidia GTX 1070, además se utilizó un par de tarjetas de video, una Nvidia 9500

y otra 8400 ya que el host necesita una salida de video en el sistema operativo UnRAID, por lo que se pierde un asiento.

El Input/Output Memory Management Unit (IOMMU) es una tecnología destinada a virtualización, y lo que busca es reducir el overhead que genera la comunicación en *system calls* entre sistema host y virtual, al dar *address translation*, *system memory access protection* en transferencias DMA(Direct memory access), y filtrado y remapeo de *interrupts* (AMD, 2018). Dicho de otra manera, aumenta la eficiencia de comunicación con el hardware de la máquina host, dando una conexión casi directa entre hardware y el OS virtualizado.

La decisión de utilizar UnRAID yace en el deseo de demostrar el poder de IOMMU y cómo su utilización permite la creación no sólo de máquinas virtuales, si no de poderosas estaciones de trabajo con una baja pérdida de rendimiento, ya que, según AMD, esta tecnología da rendimiento casi nativo en máquinas virtuales (AMD, 2016).

La instalación de UnRAID consiste en 2 pasos.

1. Creación de un USB de arranque
2. Obtención y activación de la licencia

La primera parte actualmente es muy sencilla ya que de la página oficial se puede descargar un “.exe” que reconocerá los dispositivos USB conectados de los cuales se puede seleccionar el deseado; una lista de opciones especializadas que incluye el nombre del servidor, la posibilidad de dar una IP estática y decidir si esta USB trabajará sobre sistemas UEFI.

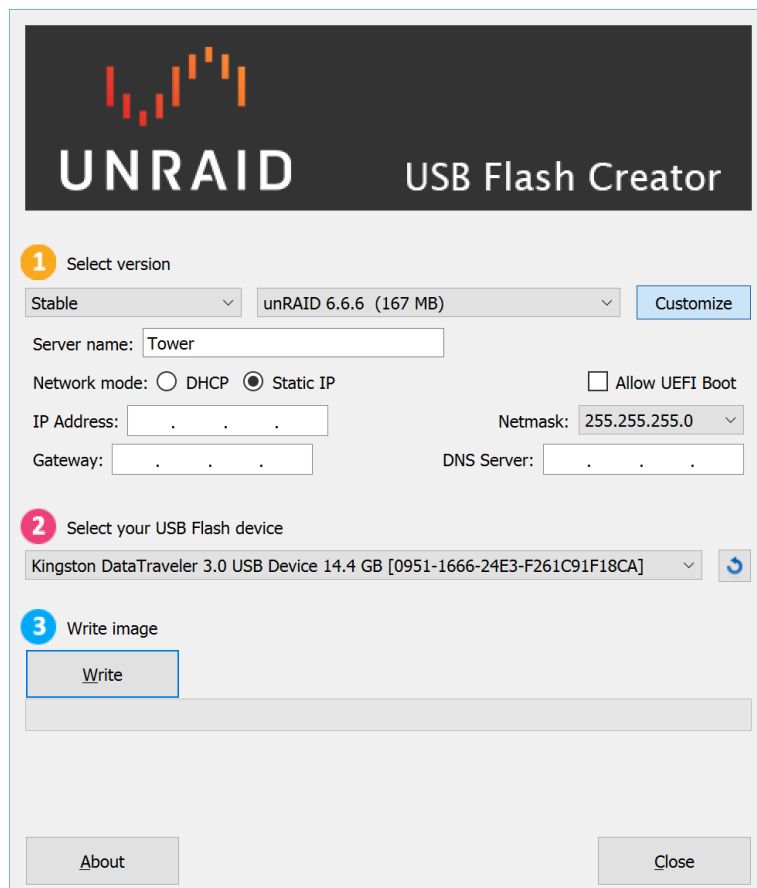


Figura 16: Programa de UnRAID para creación de un USB arrancable

Cuando el proceso termine se desconecta el USB y se lo conecta a la máquina que se utilizará de servidor, se la arranca, y cuando todo haya iniciado, desde otra máquina se procede a conectarse utilizando la IP provista anteriormente y se tendrá una pantalla como la mostrada en la Figura 17.

The screenshot displays the UnRAID web interface. At the top, the server status is shown as 'Tower • 192.168.1.10' with 'Medija server' description, '6.1.6' version, and '21 minutes' uptime. The main navigation bar includes 'Dashboard', 'Main', 'Shares', 'Users', 'Settings', 'Plugins', and 'Tools'. The 'Array Devices' section shows a RAID array with the following data:

Device	Identification	Temp.	Reads	Writes	Errors	FS	Size	Used	Free	View
Parity	ST4000DM000-1F2168_Z300P821 - 4 TB (sdb)	24 C	501	299,201	0					
Disk 1	WDC_WD20EURS-63S48Y0_WD-WCAZAH211356 - 2 TB (sde)	31 C	299,857	536	0	xfs	2 TB	321 GB	1.68 TB	
Disk 2	WDC_WD40EFRX-68WT0N0_WD-WCC4E3YFK3TT - 4 TB (sdc)	33 C	308,812	217	0	xfs	4 TB	2.01 TB	1.99 TB	
Disk 3	WDC_WD40EFRX-68WT0N0_WD-WCC4E7HR0Z40 - 4 TB (sdd)	31 C	309,322	1233	0	xfs	4 TB	2 TB	2 TB	
Total	Array of four devices	29.8 C	918,492	301,187	0		10 TB	4.33 TB	5.67 TB	

The 'Boot Device' section shows a 'Flash' device 'DataTraveler_3.0 - 7.83 GB (sda)' with 'vfat' filesystem, '7.81 GB' size, '156 MB' used, and '7.65 GB' free. The 'Array Operation' section includes controls for 'Stop', 'Cancel', 'Spin Down', 'Spin Up', and 'Clear Statistics', each with a brief description of its function.

Figura 17: Pantalla de inicio del servidor UnRAID

A partir de este punto se detendrá la descripción de la creación de máquinas virtuales en UnRAID ya que esto se encuentra bien documentado en internet y además sale de los límites de este trabajo, y como tal se presenta el trabajo final en la Figura 18.

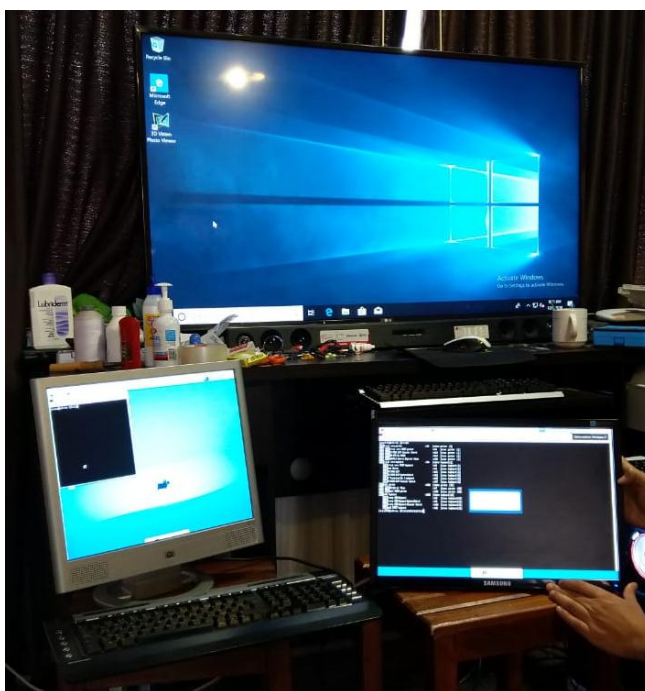


Figura 18: Sistema multi-tarjeta

7. Consideraciones configuración multi-cabecera

- El entorno gráfico más simplista que se pueda encontrar funcionará bien para el asiento maestro, ya que no se espera trabajar sobre él más que para la creación de los demás.
- Durante la explicación de LightDM se menciona secciones del archivo de configuración conocidas como “Seats”, esto no debe ser confundido con los asientos aquí creados, de modo que se podría agregar una sección “Seat 2”, pero no afectará en absoluto al segundo asiento, ya que lo creado en la configuración multi-cabecera se podría considerar como asientos falsos, y en un mismo asiento (“Seat 0”) se lanzan una serie de servidores X que no cuentan dentro del sistema como asientos en sí.
- Se puede crear un usuario sin privilegios *root* para el manejo de asientos, ya que todos los comandos usados a partir de la sesión gráfica iniciada corren sin tal necesidad, no obstante, por cuestiones administrativas es recomendable dar al menos un grado medio de permisos a tal usuario.
- Se tiene que considerar las implicaciones de seguridad que dicha configuración involucra, ya que al trabajar sobre un host es posible lanzar el terminal al aplastar la combinación de teclas *Ctrl+Alt+F#* donde # es el número de terminal deseado

8. Consideraciones multi-tarjeta

- Con el uso exclusivo de una tarjeta para cada host, los costos por asiento se elevan considerablemente
- Si bien es cierto el entorno se encuentra autocontenido, y se evita el acceso a carpetas personales de otros usuarios, se sigue corriendo sobre la misma máquina y sobre el mismo sistema operativo, lo que causa que una falla crítica en un asiento tiene la posibilidad de botar al resto de ellos
- Linux tiene una amplia gama de tarjetas soportadas, no obstante hay excepciones que se deben tomar en cuenta antes de adquirir hardware que puede ser incompatible con la versión de Linux corriendo en el servidor

Discusión financiera

1. Descripción del escenario

El programa quiteño, conocido como Guambra Centros, fue lanzado hace 25 años, y que actualmente se encuentra bajo la dirección del Patronato San José, se enfoca en brindar “espacios educativos, seguros y de calidad” para niños menores de 5 años (Guambra Centros, 2018), y actualmente cuentan con 14 sedes en todo el Distrito Metropolitano de Quito (GUAMBRA CENTROS, 2018). Éste es el escenario elegido a ser analizado en un principio, centros educativos infantiles para niños y niñas de bajos recursos.

El hardware elegido es muy dependiente de la capacidad adquisitiva de cada institución, pero dado que se apunta a centros de bajos recursos se buscará la tecnología más barata existente en el país; adicionalmente se debe considerar los

requisitos mínimos del sistema operativo elegido, según su página oficial (Installation Guide, 2018):

- Cualquier procesador compatible con la arquitectura x86_64
- 512 MiB de RAM
- 800 MiB de espacio en disco duro

Pensando en el programa educativo social antes mencionado, la presidenta de la Fundación Patronato San José, María Fernanda Pacheco, en una entrevista con El Comercio mencionó que se espera atender a 20.000 niños para el 2019 (2017), lo que considerando una distribución uniforme entre los 14 centros da 1430 estudiantes por cada uno (redondeado al múltiplo de 5 más cercano), para 5 días laborales tenemos 286, y finalmente asumiendo un periodo de estudio de 8 horas se deberá atender en una clase de introducción a las herramientas tecnológicas un total de 47 alumnos (redondeados al entero inferior). Para resumir:

- 47 máquinas
- Servicio a 286 estudiantes diarios
- Una hora de clase a la semana por estudiante
- 20.000 estudiantes en 14 Guambracentros

La computadora utilizada para la implementación multicabecera es una Dell OptiPlex 3060 SFF, que en el mercado estadounidense se valora en 568.68 USD (Dell, 2018), y consiste en una computadora con procesador Intel Core i3-8100, 4 GiB de RAM DDR4 y 500 GiB de espacio en disco duro (Dell, 2018). Para el caso interesa la cantidad de puertos disponibles, que resultan ser:

- 1 HDMI
- 1 VGA
- 1 DisplayPort

En teoría hay la posibilidad de conectar 3 pantallas diferentes con 2 entornos para ser usados para asientos y uno para el host.

Para el caso de multi-tarjeta se utilizó una computadora diferente, con características que se mencionarán más adelante, pero para asentar una base comparativa se puede decir que la máquina utilizada en el anterior sistema posee además 1 ranura PCIe x16 (OptiPlex 3060 SFF, 2018), lo que permitiría a aquella computadora dar servicio multi-tarjeta, e híbrido también.

2. Caso multi-cabecera

La configuración de un sistema multi-asiento con una sola tarjeta gráfica trae su propia carga de compromisos, uno de ellos como se mencionó anteriormente es la seguridad y la facilidad para hacer caer al sistema si se tiene un conocimiento previo

de Linux, no obstante, dependiendo del caso de uso este tipo de problemas no tendrían mayor impacto e incluso pueden tener sus ventajas, como en el caso que el tutor requiera presentar a todos los alumnos una diapositiva y pueda tener control de todas las máquinas.

Considerando la cantidad de salidas de video antes mencionadas, y cuantos niños se deben servir por hora, se necesita 24 máquinas si se deja un asiento de host o 16 si se ocupa todas las pantallas sin dejar ningún host habilitado. Si se desea más horas de enseñanza en los centros, sería necesario duplicar o triplicar la cantidad de máquinas para tener el mismo efecto sobre la cantidad de horas por estudiante.

Volviendo a las consideraciones de seguridad, se puede argumentar que no resulta de mayor preocupación en ambientes controlados con usuario no expertos, lo que justamente son los Guambracentros de Quito. Otra de las ventajas de esto es, no obstante, la baja redundancia de datos compartida entre todas las computadoras; la instalación de un paquete en el entorno host dará el mismo programa a todos.

3. *Caso multi-tarjeta*

Esta configuración, a diferencia de la primera, tiene un campo de acción más amplio, ya que Xorg permite nativamente la creación de asientos y asignación de dispositivos de entrada, se evita la complicación de la creación de un escritorio extendido con un servidor X anidado, y en su lugar cada sesión es autocontenida y cerrada, los usuarios poseen su propio identificador y contraseña, así como un nivel de privilegios y archivos personales privados si así se lo deseara.

Como se mencionó anteriormente, la máquina de prueba posee una ranura extra, en donde se podría colocar una tarjeta de video adicional, dando un total de 2 asientos independientes. Una tarjeta Nvidia Gigabyte GT210 se encuentra a 59.99 USD en el mercado nacional (MTEC-EC, 2018), para el caso de los Guambracentros ahora tendríamos que la cantidad de máquinas necesarias sería 24.

3. *Caso híbrido*

Así como se puede utilizar las múltiples cabeceras de una tarjeta o utilizar las múltiples ranuras de una placa madre, también es posible realizar una combinación de las dos, lo que daría un uso máximo de la computadora. En el caso de la OptiPlex 3060 con una tarjeta de video instalada como la mencionada antes se tendría en total:

- x2 VGA
- x2 HDMI
- x1 DVI-I
- x1 DisplayPort

Haciendo uso completo de estas salidas tendríamos en total 6 asientos por computadora, lo que reduce el número de máquinas, de 47 a 8 para máquinas sin

host, y a 12 con host, pero además cada máquina necesitaría su adaptador extra de video.

4. Comparación financiera multi-cabecera, multi-tarjeta, híbrido

Se resume brevemente cada configuración para los centros de cómputo propuestos por facilidad:

- Computadora Personal: 47 máquinas, 0 tarjetas gráficas
- Multi-cabecera host: 24 máquinas, 0 tarjetas gráficas
- Multi-cabecera sin host: 16 máquinas, 0 tarjetas gráficas
- Multi-tarjeta: 24 máquinas, 24 tarjetas gráficas
- Híbrido host: 12 máquinas, 12 tarjetas gráficas
- Híbrido sin host: 8 máquinas, 8 tarjetas gráficas

De acuerdo a los costos antes discutidos para una Dell OptiPlex 3060 SFF y una tarjeta gráfica Nvidia 210 tenemos para empezar la Tabla 1.

Configuración	Costo unitario	Unidades	Total
Computadora personal	\$568.68	47	\$26,727.96
Multi-cabecera host	\$568.68	24	\$13,648.32
Multi-cabecera sin host	\$568.68	16	\$9,098.88
Multi-tarjeta	\$568.68	24	\$13,648.32
Híbrido host	\$568.68	12	\$6,824.16
Híbrido sin host	\$568.68	8	\$4,549.44

Tabla 1: Costo de máquinas para el laboratorio propuesto

Ahora para las tarjetas tenemos la Tabla 2:

Configuración	Costo unitario	Unidades	Total
Computadora personal	\$59.99	0	\$0
Multi-cabecera host	\$59.99	0	\$0
Multi-cabecera sin host	\$59.99	0	\$0
Multi-tarjeta	\$59.99	24	\$1,439.76
Híbrido host	\$59.99	12	\$719.88
Híbrido sin host	\$59.99	8	\$479.92

Tabla 2: Costo de tarjetas para el laboratorio propuesto

Adicionalmente se debe considerar periféricos como pantalla, mouse y teclado para cada asiento, que considerando que el total de niños servidos será 47, y de acuerdo a precios en el mercado nacional tenemos:

- Combo teclado y mouse: \$8.56 (Cartimex, 2018)
- Pantalla: \$90.00 (Cartimex, 2018)

Con un total de \$4,632.32, por los 3 dispositivos multiplicados los 47 asientos, unido con los costos anteriores, finalmente obtenemos el precio total del laboratorio de computación propuesto, resumido en la Tabla 3:

Configuración	Costo de máquinas	Costo de tarjetas	Costo de periféricos	Total
Computadora personal	\$26,727.96	\$0	\$4,632.32	\$31,360.28
Multi-cabecera host	\$13,648.32	\$0	\$4,632.32	\$18,280.64
Multi-cabecera sin host	\$9,098.88	\$0	\$4,632.32	\$13,731.2
Multi-tarjeta	\$13,648.32	\$1,439.76	\$4,632.32	\$19,720.4
Híbrido host	\$6,824.16	\$719.88	\$4,632.32	\$12,176.36
Híbrido sin host	\$4,549.44	\$479.92	\$4,632.32	\$9,661.68

Tabla 3: Costos totales para el laboratorio propuesto

5. Caso Raspberry Pi

Una opción adicional existe, y es la creación de dicho laboratorio utilizando Raspberry Pi, por lo que, para comenzar se da una descripción técnica de las características de su modelo 3 B+ directamente tomada de su página web (2018):

- Cortex-A53 (ARMv8) 64bit SoC @ 1.4GHz
- 1GB LPDDR2 SDRAM
- Full-size HDMI

Un procesador ARM, o Advanced RISC Machine, utiliza una arquitectura diferente a la de los procesadores i386 y amd64, por lo que para instalar un sistema operativo se debe utilizar una versión específica para ARM.

Los componentes necesarios, aparte de la placa en sí (Raspberry Pi RASPBerry PI 3 MODEL B+, 2018), serán una memoria microSD de 32GB (MEMORIA MICRO SD SANDISK 32GB C / A (5A), 2018), ya que no viene con memoria integrada; un case (NEW AGE ENCLOSURES P4P-382612, 2018), y los periféricos utilizados anteriormente, para tener lo descrito en la Tabla 4:

Componente	Costo unitario	Unidades	Total
Raspberry Pi 3 Model B+	\$35.00	47	\$1,645
Memoria microSD 32GB	\$18.71	47	\$879.37
Carcasa para Raspberry Pi 3	\$6.63	47	\$311.61
Monitor	\$90.00	47	\$4,230

Combo mouse y teclado	\$8.56	47	\$402.32
Total	\$7,468.3		

Tabla 4: Costo de componentes para sistemas sobre Raspberry Pi

6. Resumen comparativo

Después de analizar los 4 diferentes escenarios tenemos los resultados en términos de costo por máquina demostrados en la Tabla 5:

Configuración	Costo total	Costo por asiento
Computadora personal	\$31,360.28	\$667.24
Multi-cabecera host	\$18,280.64	\$388.95
Multi-cabecera sin host	\$13,731.2	\$292.15
Multi-tarjeta	\$19,720.4	\$419.58
Híbrido host	\$12,176.36	\$259.07
Híbrido sin host	\$9,661.68	\$205.57
Raspberry Pi	\$7,468.3	\$158.9

Tabla 5: Resumen comparativo

Se puede considerar que la opción de utilizar Raspberry Pi para la construcción del laboratorio es la más eficiente en tema de costos, no obstante, hay consideraciones que tener en cuenta como las siguientes:

- Velocidad de transferencia, memoria SD vs SATA: La memoria seleccionada, es una clase 4, lo que significa que la mínima tasa de transferencia es de 4Mbps (Kingston, 2018), mientras que SATA I (la primera revisión del estándar) es de 1.5Gbps (Sandisk, 2018).
- Arquitectura: Los procesadores ARM están limitados por sus instrucciones, y no pueden correr aplicaciones destinadas a arquitecturas CISC, y si bien es cierto hay bastante mercado de aplicaciones RISC, es importante tener en cuenta tal limitación.
- Desgaste de componentes: Las memorias microSD es un tipo de memoria flash, y estas se encuentran destinadas a sufrir desgaste por múltiples escrituras (Perdue, K., 2018). Eventualmente las memorias flash fallarán en el tiempo que un disco duro seguiría trabajando normalmente.

Después de todos estos puntos realmente queda a discreción del usuario la decisión de construir un laboratorio de bajo costo utilizando sistemas multi-asiento, zero/thin clients, o Raspberry Pi; se debe tener en cuenta que cada una de estas opciones trae compromisos y que no existe una solución para todas las necesidades.

7. Escenario alterno

Como se ha podido ver, los sistemas multi-asiento son una alternativa a sistemas de bajo costo para instituciones benéficas o con miras al trabajo social, pero en menor grado se los puede considerar como una manera barata de añadir una máquina al equipo informático disponible, como ejemplo se propone un hogar con

dos niños y una máquina familiar. Se puede hacer uso de sus salidas de video extra y servir a dos niños en sus tareas escolares y ocio (limitado) comprando únicamente los periféricos necesarios. Una de las dificultades de este escenario yace en la pericia técnica necesaria para la configuración, por lo que hasta que no haya una manera sencilla de instalación de estos sistemas queda como posibilidad a futuro.

CONCLUSIONES Y TRABAJO FUTURO

Conclusiones

Después de haber implementado el sistema propuesto, investigar opciones al problema planteado y compararlas unas contra otras, he llegado a las siguientes conclusiones:

1. Los sistemas multi-asiento son viables para ciertos entornos.
2. La falta de desarrollo a nivel profesional de ellos también dice que puede no ser un modelo económico efectivo, lo que ha llevado a varias empresas a alejarse del desarrollo de éstos.
3. Se puede reducir la producción de desechos tecnológicos utilizando sistemas de este tipo; mucho más, si se lo empareja con campañas de reciclaje destinadas a obtener máquinas buenas, o piezas para armar una, que están a punto de ser descartadas.
4. Después de analizar el costo de implementación de cada una de las opciones multi-asiento, me resulta evidente su viabilidad para instituciones o proyectos que busquen proveer a muchos usuarios de terminales de trabajo, a un costo muy reducido.

Trabajo futuro

La comparación de sistemas multi-cabecera e híbridos sin host se encuentra fundamentada en el desarrollo de un sistema operativo que permita al usuario nativamente crear un sistema multi-asiento, restringiendo las sesiones anidadas y las combinaciones de teclas que podrían afectar al entorno en general, por lo que como trabajo futuro se puede considerar lo siguiente:

1. Modificación del kernel de Arch Linux para volver las sesiones anidadas autocontenidas.
2. Creación de un manejador de ventanas que permita levantar servidores X anidados autocontenidos.
3. Programación de un asistente de instalación que permita al administrador configurar el sistema de acuerdo con sus necesidades.
4. Automatizar procesos de actualización y mantenimiento del software para evitar la necesidad de un experto constantemente.

BIBLIOGRAFÍA

- Akita, F. (2017). Arch Linux – Best distro ever? Recuperado de <http://www.akitaonrails.com/2017/01/10/arch-linux-best-distro-ever> el 7 de octubre del 2018
- Allied. (2018). NEW AGE ENCLOSURES P4P-382612. Recuperado de <https://www.alliedelec.com/new-age-enclosures-p4p-382612/71356508/> el 17 de diciembre del 2018
- Allied. (2018). Raspberry Pi RASPBERRY PI 3 MODEL B+. Recuperado de <https://www.alliedelec.com/raspberry-pi-raspberry-pi-3-model-b-/71131895/?src=raspberrypi> el 17 de diciembre del 2018
- Amazon. (2018). AMD Ryzen Threadripper 2990WX Processor (YD299XAZAFWOF). Recuperado de <https://www.amazon.com/AMD-Threadripper-2990WX-Processor-YD299XAZAFWOF/dp/B07G25SD1P> el 25 de septiembre del 2018
- AMD. (2016). AMD I/O Virtualization Technology (IOMMU) Specification. Recuperado de https://www.amd.com/system/files/TechDocs/48882_IOMMU.pdf el 27 de noviembre del 2018
- AMD. (2018). AMD Ryzen Threadripper 2990WX Processor. Recuperado de <https://www.amd.com/en/products/cpu/amd-ryzen-threadripper-2990wx> el 25 de septiembre del 2018
- AMD. (2016). VIRTUALIZING IO THROUGH THE IO MEMORY MANAGEMENT UNIT (IOMMU). Recuperado de http://pages.cs.wisc.edu/~basu/isca_iommu_tutorial/IOMMU_TUTORIAL_ASPLOS_2016.pdf el 16 de diciembre del 2018
- Arch Linux. (2018). About Arch Linux. Recuperado de <https://www.archlinux.org/about/> el 7 de octubre del 2018
- Arch Linux. (2018). Arch Linux. Recuperado de https://wiki.archlinux.org/index.php/Arch_Linux el 7 de octubre del 2018
- Arch Linux. (2018). Installation Guide. Recuperado de https://wiki.archlinux.org/index.php/installation_guide el 7 de octubre del 2018
- Arch Linux. (2018). Pacman. Recuperado de <https://wiki.archlinux.org/index.php/pacman> el 8 de octubre del 2018
- die.net. (2017). passwd(1) – Linux man page. Recuperado de <https://linux.die.net/man/1/passwd> el 8 de octubre del 2018
- Arch Linux. (2018). Xephyr. Recuperado de <https://wiki.archlinux.org/index.php/Xephyr> el 11 de octubre del 2018

Arch Linux. (2018). Xinput. Multi-pointer X. Recuperado de https://wiki.archlinux.org/index.php/Multi-pointer_X el 13 de noviembre del 2018

CARTIMEX. (2018). FLAT PANEL 15.6" LED 1366X768 RGB (VGA). Recuperado de <http://www2.cartimex.com/profiles/VistaProducto/33659/1074> el 16 de diciembre del 2018

CARTIMEX. (2018). MEMORIA MICRO SD SANDISK 32GB C / A (5A). Recuperado de <http://www2.cartimex.com/profiles/VistaProducto/29598/1074> el 17 de diciembre del 2018

CARTIMEX. (2018). TECLADO Y MOUSE XTECH ALAMBRICO USB NEGRO. Recuperado de <http://www2.cartimex.com/profiles/VistaProducto/38665/1074> el 16 de diciembre del 2018

Cygwin/X. (2018). Glossary. Recuperado de <https://x.cygwin.com/docs/faq/glossary.html#gloss-xdmc> el 12 de noviembre del 2018

Dell. (2018). New OptiPlex 3060 Small Form Factor. Recuperado de <https://www.dell.com/ec/p/desktops-n-workstations?c=ec&l=es&s=bsd&~ck=mn#!dlpgid=optiplex-desktops> el 16 de noviembre del 2018

Dell. (2018). OptiPlex 3060 Small Form Factor Service Manual. Recuperado de https://topics-cdn.dell.com/pdf/optiplex-3060-desktop_service-manual2_en-us.pdf el 6 de diciembre del 2018

El Comercio. (2017). Padres y niños de centros de desarrollo infantil del Municipio festejaron el fin del año lectivo. Recuperado de <https://www.elcomercio.com/tendencias/padres-ninos-centrosdedesarrolloinfantil-patronatosanjose-municipiodequito.html> el 16 de noviembre del 2018

Freiberger, P. A., & Swaine, M. R. (2017, Junio 28). ENIAC. Recuperado de <https://www.britannica.com/technology/ENIAC> el 25 de septiembre del 2018

García, L., Hatch, E., Larsen, J., Moulton, M., Ocampo, R., & Singleton, R. (2010, Abril 28). "Connect-on-Demand" Satellite/Radio Link for Tanzanian Schools. Recuperado de https://www.egr.msu.edu/classes/ece480/capstone/spring10/group03/documents/Final_Report.pdf el 19 de diciembre del 2018

Hellard, B., & Hopping, C. (2018). Windows vs Linux: what's the best operating system? Recuperado de <https://www.itpro.co.uk/operating-systems/24841/windows-vs-linux-whats-the-best-operating-system> el 19 de diciembre del 2018

Hernández D., & Soler Y. (2012). Estaciones de trabajo multi-asiento. Recuperado de <https://www.redalyc.org/html/3783/378343675008/> el 19 de diciembre del 2018

IBM. (2009). A HISTORY OF PROGRESS. Recuperado de https://www-03.ibm.com/ibm/history/interactive/ibm_history.pdf el 7 de octubre del 2018

Intel. (1989). M80C86/M80C86-2 16 BIT CMOS MICROPROCESSOR. Recuperado de <http://datasheets.chipdb.org/Intel/x86/808x/datashts/8086/27105805.pdf> el 25 de septiembre del 2018

Intel Newsroom. (2018, Mayo 31). Intel at 50: The 8086 and Operation Crush. Recuperado de <https://newsroom.intel.com/news/intel-50-8086-operation-crush/> el 25 de septiembre del 2018

Kerner, M. (2007, Febrero). A History of Modern 64-bit Computing. Recuperado de <https://courses.cs.washington.edu/courses/csep590/06au/projects/history-64-bit.pdf> el 25 de septiembre del 2018

Kingston. (2018). MicroSD Cards. Recuperado de https://www.kingston.com/us/flash/microsd_cards/sdc4 el 17 de diciembre del 2018

Microsoft. (2016). Introducing MultiPoint Services. Recuperado de <https://docs.microsoft.com/en-us/windows-server/remote/multipoint-services/introducing-multipoint-services> el 19 de diciembre del 2018

Microsoft. (2018). Pricing and licensing for Windows Server 2019. Recuperado de <https://www.microsoft.com/en-us/cloud-platform/windows-server-pricing> el 19 de diciembre del 2018

MTEC-EC. (2018). GIGABYTE GT210 2GB DDR3 64BITS. Recuperado de <https://www.mtec-ec.com/producto/gigabyte-gt210-2gb-ddr3-64bits/> el 6 de diciembre del 2018

Patronato San José. (2018). Desarrollo Infantil. Recuperado de <https://www.patronato.quito.gob.ec/desarrollo-infantil.html> el 15 de noviembre del 2018

Patronato San José. (2018). GUAMBRA CENTROS. Recuperado de https://www.patronato.quito.gob.ec/images/pdf/listado_guambra_centros.pdf el 15 de noviembre del 2018

PCMag. (2018). Definition of: x86. Recuperado de <https://www.pcmag.com/encyclopedia/term/54979/x86> el 25 de septiembre del 2018

Perdue, K. (2018). Wear Leveling. Recuperado de <http://www.cypress.com/file/202541/download> el 17 de diciembre del 2018

Raspberry Pi. (2018). Raspberry Pi 3 Model B+. Recuperado de <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/> el 17 de diciembre del 2018

Sandisk. (2018). Difference between SATA I, SATA II and SATA III. Recuperado de https://kb.sandisk.com/app/answers/detail/a_id/8142/~/-/difference-between-sata-i%2C-sata-ii-and-sata-iii el 17 de diciembre del 2018

Sheesly, J. (2008, Junio 16). Intel's 8086 passes the bit 3-0. Recuperado de <https://www.techrepublic.com/blog/classics-rock/intels-8086-passes-the-big-3-0/> el 25 de septiembre del 2018

Ubuntu. (2018). Download Ubuntu Desktop. Recuperado de <https://www.ubuntu.com/download/desktop> el 19 de diciembre del 2018

Userful. (2013). Online Manuals. Recuperado de <https://support.userful.com/Manuals> el 19 de diciembre del 2018

Userful. (2015). Userful Multiseat™ Release Notes. Recuperado de <https://support.userful.com/UMs/Manuals/ReadMe/UMs> el 19 de diciembre del 2018

Walters, P., Youge, A., Kang, D., Yao, K., Kang, M., Crago, S., Fox, G. (2014). GPU Passthrough Performance: A comparison of KVM, Xen, VMWare ESXi, and LXC for CUDA and OpenCL Applications. Recuperado de https://www.isi.edu/sites/default/files/users/jwalters/papers/Cloud_2014.pdf el 27 de noviembre del 2018