

UNIVERSIDAD NACIONAL DEL LITORAL

FACULTAD DE INGENIERIA Y CIENCIAS HÍDRICAS

---



**UNL • FACULTAD  
DE INGENIERÍA Y  
CIENCIAS HÍDRICAS**

TRABAJO FINAL

**MIGRACIÓN DE SOFTWARE PRIVATIVO A SOFTWARE  
LIBRE**

*baruja shen adonai elohim*

---

Blanco Emanuel A.

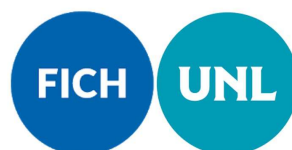
Universidad Nacional Del Litoral.

Facultad de Ingeniería y Ciencias Hídricas.

Tecnicatura Universitaria en Software Libre.

Creative Commons Atribución-CompartirDerivadasIgual 2.5 (Argentina)

julio, 2022



# Índice general

<b>1. Introducción</b>	<b>1</b>
<b>2. Motivación</b>	<b>3</b>
<b>3. Contexto</b>	<b>4</b>
<b>4. Importancia</b>	<b>5</b>
<b>5. Destinatarios</b>	<b>6</b>
<b>6. Conceptos técnicos aplicados</b>	<b>7</b>
<b>7. Licencias involucradas</b>	<b>8</b>
7.1. Tipos de licencia . . . . .	8
7.1.1. Licencias GPL . . . . .	8
7.1.2. Licencias LGPL . . . . .	8
7.1.3. Licencias AGPL . . . . .	8
7.1.4. Licencias Estilo BSD . . . . .	8
7.1.5. Licencia PSFL . . . . .	9
7.1.6. Licencias MPL y derivadas . . . . .	9
7.1.7. Licencia CDDL . . . . .	9
7.1.8. Licencias EPL . . . . .	9
7.1.9. Licencia Apache . . . . .	9
7.1.10. Licencia PHP . . . . .	9
7.1.11. Licencias Creative Commons . . . . .	10
<b>8. Documentación</b>	<b>11</b>
<b>9. Estado del arte</b>	<b>12</b>
<b>10. Objetivos</b>	<b>13</b>
10.1. Generales . . . . .	13
10.2. Específicos . . . . .	13
<b>11. Desarrollo</b>	<b>14</b>
<b>12. Cronograma de trabajo</b>	<b>15</b>
<b>13. Proceso de migración</b>	<b>17</b>
13.1. Aspectos importantes . . . . .	17
13.2. Inventario de Software . . . . .	17
13.2.1. Software Invetariado . . . . .	17
13.3. Inventario de Hardware . . . . .	18
13.3.1. Hardware Inventariado . . . . .	18
13.4. Sistema Operativo . . . . .	18
13.4.1. Distribución GNU/Linux . . . . .	18
13.4.2. Debian . . . . .	19

13.5. Instalación del sistema operativo . . . . .	19
13.5.1. Requisitos . . . . .	19
13.5.2. Proceso de Instalación . . . . .	19
13.6. Repositorios . . . . .	21
13.6.1. ¿Que es un repositorio? . . . . .	21
13.6.2. Configuración de repositorios Debian . . . . .	21
13.7. Instalación de Programas . . . . .	21
<b>14. Configuración red LAN</b>	<b>23</b>
14.1. ¿Que es una red LAN? . . . . .	23
14.2. ¿Que es una dirección IP? . . . . .	23
14.2.1. Direcciones IP dinámicas . . . . .	23
14.2.2. Direcciones IP estáticas . . . . .	23
14.3. Configuración de red . . . . .	23
14.3.1. Diagrama de red . . . . .	24
<b>15. Servidor Local</b>	<b>25</b>
15.1. ¿Que es un Servidor Local . . . . .	25
15.2. Uso del servidor . . . . .	25
15.3. Sistema Operativo . . . . .	25
15.4. Instalación del Sistema Operativo . . . . .	25
15.5. LVM . . . . .	27
15.5.1. ¿Que es LVM? . . . . .	27
15.5.2. Gestión volúmenes lógicos . . . . .	27
15.6. fstab . . . . .	28
15.6.1. Configurar fstab . . . . .	29
15.7. Controlar cambios en el sistema . . . . .	29
15.7.1. Generar Hash . . . . .	30
15.7.2. Comparar Hash . . . . .	30
15.8. Configuración de red . . . . .	31
15.8.1. Asignar una IP estática . . . . .	31
15.8.2. Servidor DNS . . . . .	31
15.9. Firewall . . . . .	32
15.9.1. ¿Que es iptables? . . . . .	32
15.9.2. Reglas iptables . . . . .	32
15.10 Servidor Docker . . . . .	33
15.10.1. ¿Que es Docker? . . . . .	33
15.10.2. Funcionamiento de los contenedores? . . . . .	33
15.10.3. Características . . . . .	33
15.10.4. Herramientas de Docker . . . . .	33
15.10.5. Instalación Docker . . . . .	34
15.10.6. ¿Que es Docker Compose? . . . . .	35
15.10.7. Instalación Docker Compose . . . . .	35
15.11 Contenedor NextCloud . . . . .	35
15.11.1. ¿Que es Nextcloud? . . . . .	35
15.11.2. Características: . . . . .	35
15.11.3. Requisitos: . . . . .	35
15.11.4. Preparación del área de trabajo . . . . .	35
15.11.5. Crear Contenedor . . . . .	37
15.11.6. Configuración de NextCloud . . . . .	37
15.11.7. Conclusión . . . . .	38
15.12 Contenedor Emby . . . . .	39
15.12.1. ¿Que es Emby? . . . . .	39
15.12.2. Características de Emby: . . . . .	39
15.12.3. ¿Qué es un centro multimedia? . . . . .	39
15.12.4. Preparación del área de trabajo . . . . .	39
15.12.5. Crear Contenedor Docker: . . . . .	40
15.12.6. Configuración Emby . . . . .	40

15.12.7.Conclusión . . . . .	41
<b>16.Conclusión</b>	<b>42</b>

# Capítulo 1

## Introducción

Este trabajo es el resultado de un largo periodo formativo, el cual concluyó en Diciembre del 2020. Lo desarrollado en esta página pretende cumplir con los requisitos del Trabajo Final de la Tecnicatura Universitaria en Software Libre.

El trabajo desarrollado tuvo lugar en la sala de informática perteneciente a la UNL, ubicada en el instituto de detención penal N° 2 “Las Flores”. El propósito de esta sala informática es posibilitar el acceso a personas privadas de su libertad a distintas trayectorias educativas/culturales/laborales, mediante el estudio a distancia a través de internet.

El siguiente texto describe la migración de Software Propietario a Software Libre realizada en el Aula Virtual de la UNL, en el penal de las flores.

Los objetivos propuestos y a concretar fueron los siguientes:

- Migrar los equipos informáticos de Software Privativos a Software Libre en todas sus dimensiones.
- Lograr el cambio de una filosofía a otra en la selección y organización de las tecnologías utilizadas en el aula.
- Posibilitar que los estudiantes modifiquen la visión que tienen respecto al Software Libre.

Problemas resueltos:

- Bajo rendimiento del hardware disponible.
- Incompatibilidad del software con el hardware disponible.
- Rendimiento deficitario de la red informática.
- Falta de autonomía tecnológica.

Para elaborar este trabajo fue necesario llevar adelante una investigación previa a la migración, la cual consistió en dos etapas:

- Investigar sobre otros procesos de migración o proyectos similares que hayan tenido éxito, indagar si podría tomarse como modelo alguno de ellos/continuarlo/mejorarlo.
- Diseñar una estrategia que permita alcanzar el objetivo planteado, junto con un cronograma especificando las acciones y tiempo estimado de inicio y finalización.

Todo lo realizado fue posible gracias al apoyo de la gran comunidad del Software Libre. Es el deseo de este estudiante activista que todo lo elaborado pueda servir para motivar y ayudar a otras personas en el proceso de migración de cualquier otra institución con similares características.

## Movimiento Software Libre

En la década del '70', cuando la computación estaba en sus inicios era común que tanto los desarrolladores de software profesionales, como los aficionados, publicaran sus trabajos para que otros puedan utilizarlo, corregir errores y mejorarlo. A partir del avance de las industrias tecnológicas sobre el software en la década del '80', las prácticas colaborativas se vieron afectadas, y en consecuencia muchos desarrolladores de software decidieron dejar de compartir sus trabajos y solamente dejar que otras personas lo utilicen bajo ciertas condiciones, manifestadas en lo que se denomina "licencias restrictivas". Estas licencias no permiten que se pueda compartir el programa sin el consentimiento del desarrollador y mucho menos la posibilidad de poder hacerle modificaciones para corregir errores o agregar mejoras.

Un caso ejemplar lo constituye el episodio protagonizado por la empresa Microsoft, la cual le envió una carta a un grupo de programadores aficionados que utilizaban copias no autorizadas de su programa BASIC. En esta carta, Bill Gates, "General Partner", acusa a esos programadores de que le están robando su programa, argumentando que compartir el software es injusto, ya que su creador no recibe suficiente dinero a cambio. Esta forma de pensar atentaba contra el espíritu de cooperación, solidaridad y reciprocidad que existía en ese entonces en los grupos informáticos.

Para contrarrestar esta tendencia a no compartir el código fuente, surgió el movimiento Software Libre.

El Software Libre es un movimiento ético, político y social, que tiene por objetivo defender la libertad de las personas en un mundo donde las computadoras afectan cada vez más nuestra forma de vivir. Se lo considera como un movimiento político y social, dado que no solo implica defender las "cuatro libertades esenciales", sino que también, el no permitir que el capitalismo tome el poder del conocimiento. Es por ello, que centra su lucha en una mirada política sobre el conocimiento en general y las tecnologías en particular, ya que se cuestiona el concepto de "propiedad privada del conocimiento" y busca promover la libertad de los "usuarios de computadoras", para contribuir en la lucha por los derechos de los ciudadanos en el entorno digital. ( traficante de sueños)

Los artefactos diseñados bajo la filosofía del Software Libre posibilita Satisfacer las necesidades tecnológicas de las comunidades y los individuos, ya que al poder modificarse se puede adaptar a las necesidades existentes.

Ademas, al poder ser redistribuido libremente, (sea la versión original o una con modificaciones) se aporta al desarrollo de la sociedad. De esta manera al compartir el programa y las ideas, se logra generar más conocimiento y el involucra miento de las personas en las decisiones sobre el desarrollo tecnológico de sus comunidades.

## Capítulo 2

# Motivación

Este trabajo fue motivado al reconocer las distintas problemáticas existentes en el aula informática de la UNL.

El bajo rendimiento de los ordenadores por el uso de software privativo, la degradación del rendimiento causada por virus o spyware, la falta de fondos para adquirir licencias de software, son algunos de los problemas que no permitían un uso eficiente de los equipos informáticos, para que cubra las necesidades de los alumnos en ese lugar.

El interés por realizar este trabajo se centro en buscar alternativas viables para superar las limitaciones y mejorar las herramientas de trabajo en el marco de un proyecto que garantiza el acceso a la educación de una población vulnerable.

## Capítulo 3

# Contexto

El Programa “Educación Universitaria en Prisiones”. El cual consiste en alinea con aquellos intentos de transformar la herramienta educativa en un vehículo no ya de “corrección”, ni de “moralización”, sino de resistencia frente a la degradación cotidiana que el encierro supone. Se trata siempre de intentar construir espacios de libertad, gobernados por una lógica sustancialmente distinta de aquella que rige el penal.

El Programa comenzó a funcionar en el año 2004, a partir de la firma de un convenio entre la Universidad Nacional del Litoral y el entonces Ministerio de Gobierno, Justicia y Culto. En función de este convenio, se disponía la instalación de aulas virtuales en las Unidades Penitenciarias N<sup>o</sup> I de la Ciudad de Coronda y N<sup>o</sup> II “Las Flores” de la ciudad de Santa Fe.

Estas aulas se integrarían a la Red de Campus Virtuales a través de los cuales opera el Centro Multimedial de Educación a Distancia de la UNL.

### Propuesta y actividades desarrolladas en las aulas virtuales

La oferta educativa del Programa está compuesta por carreras de pre-grado, denominadas Tecnicaturas, que brindan formación técnica vinculada con demandas del mercado laboral, tienen una duración que oscila entre 5 y 6 cuatrimestres y otorgan título universitario de validez nacional.



## Capítulo 4

# Importancia

La importancia de este trabajo radica en la mejora sustancial de las prestaciones del equipamiento informático del aula de la universidad como condición necesaria para garantizar el acceso a la educación superior de la población de la Unidad Penal N°2, “Las Flores”.

## Capítulo 5

# Destinatarios

Los destinatarios beneficiados por la concreción de este trabajo son alrededor de 30 internos, que se encuentran cursando diferente carreras de la oferta a distancia de la Universidad Nacional del Litoral.

## Capítulo 6

# Conceptos técnicos aplicados

Conocimientos técnicos aplicados en este trabajo:

- Administración de Sistemas Operativos GNU/Linux: Incluye la personalización, instalación y configuración de servicios.
- Administración de redes de datos: configuración de direcciones IP.
- Administración básica de Docker: Creación y Configuración de contenedores.

# Capítulo 7

## Licencias involucradas

### 7.1. Tipos de licencia

#### 7.1.1. Licencias GPL

Una de las más utilizadas es la Licencia Pública General de GNU (GNU GPL). El autor conserva los derechos de autor, y permite la redistribución y modificación bajo algunos términos diseñados para asegurarse de que todas las versiones modificadas del software permanecen bajo la misma licencia GNU GPL. Esto hace que sea imposible crear un producto con partes no licenciadas GPL: La licencia GNU GPL posibilita la modificación y redistribución del software, pero únicamente bajo esa misma licencia.

#### 7.1.2. Licencias LGPL

La Licencia Pública General Reducida de GNU, o más conocida por su nombre en inglés GNU Lesser General Public License, es una licencia creada por la (FSF) que garantiza la libertad de compartir y modificar el software cubierto por ella, asegurando que el software es libre para todos sus usuarios. Esta licencia se aplica a cualquier programa o trabajo que contenga una nota puesta por el propietario de los derechos del trabajo estableciendo que su trabajo puede ser distribuido bajo los términos de esta.

#### 7.1.3. Licencias AGPL

La Licencia Pública General de Affero es una licencia copyleft derivada de la Licencia Pública General de GNU diseñada específicamente para asegurar la cooperación con la comunidad en el caso de software que corra en servidores de red. Se engloba dentro de las licencias destinadas a modificar el derecho de autor derivadas de GNU. La Affero GPL es íntegramente una GNU GPL con una cláusula nueva que añade la obligación de distribuir el software si este se ejecuta para ofrecer servicios a través de una red de ordenadores.

La novedad de AGPL es que, aparte de las cláusulas propias de una GNU GPL, ésta obliga a que se distribuya el software que se destine a dar servicios a través de una red de ordenadores, es decir, si se quiere usar como parte del desarrollo de un nuevo software, este quedaría obligado a su libre distribución.

#### 7.1.4. Licencias Estilo BSD

Llamadas así porque se utilizan en gran cantidad de software distribuido junto a los sistemas operativos BSD. Es una licencia permisiva que casi no impone condiciones sobre lo que un usuario puede hacer con el software. El autor, bajo tales licencias, mantiene la protección de copyright únicamente para la renuncia de garantía y para requerir la adecuada atribución de la autoría en trabajos derivados, pero permite la libre redistribución y modificación, incluso si dichos trabajos tienen propietario. Son muy permisivas, tanto que son fácilmente absorbidas al ser mezcladas con la licencia GNU GPL con quienes son compatibles. También, BSD permite el cobro por la distribución de objetos binarios. Otras opiniones están orientadas a destacar que este tipo de licencia no contribuye al desarrollo de más software libre

(normalmente utilizando la siguiente analogía: una licencia BSD es más libre que una GPL si y solo si se opina también que un país que permita la esclavitud es más libre que otro que no la permite”).

### 7.1.5. Licencia PSFL

La Python Software Foundation License, anteriormente Python License, es una licencia de software libre permisiva, al estilo de la licencia BSD. Cumple con los requisitos OSI para ser declarada licencia de software libre; además, es compatible con la licencia GPL. A diferencia de la licencia GPL, y como la mayoría de licencias tipo BSD, la licencia PSFL no es una licencia copyleft, y permite modificaciones del código fuente, así como la creación de trabajos derivados, sin requerir que ni las modificaciones, ni los trabajos derivados tengan que ser a su vez de código abierto. La licencia PSFL está dentro de las listas de licencias aprobadas tanto por la Free Software Foundation como por la Open Source Initiative.

### 7.1.6. Licencias MPL y derivadas

Esta licencia es de Software Libre y tiene un gran valor porque fue el instrumento que empleó Netscape Communications Corp. para liberar su Netscape Communicator 4.0 y empezar ese proyecto tan importante para el mundo del Software Libre: Mozilla. Se utilizan en gran cantidad de productos de software libre de uso cotidiano en todo tipo de sistemas operativos. La MPL es Software Libre y promueve eficazmente la colaboración evitando el efecto “viral” de la GPL. No obstante la MPL no es tan excesivamente permisiva como las licencias tipo BSD. Estas licencias son denominadas de copyleft débil. La NPL (luego la MPL) fue la primera licencia nueva después de muchos años, que se encargaba de algunos puntos que no fueron tomados en cuenta por las licencias BSD y GNU. En el espectro de las licencias de software libre se la puede considerar adyacente a la licencia estilo BSD, pero perfeccionada.

### 7.1.7. Licencia CDDL

Common Development and Distribution License, también conocida como Sun Public License (SPL) versión 2, es una licencia de código abierto (OSI) y libre, producida por Sun Microsystems, basada en la Mozilla Public License o MPL, versión 1.1. La licencia CDDL fue enviada para su aprobación al Open Source Initiative el 1 de diciembre de 2004, y fue aprobada como una licencia de código abierto a mediados de enero de 2005. En el primer borrador hecho por el comité de divulgación de licencias OSI, la CDDL es una de las nueve licencias más populares, mundialmente usadas o con fuertes comunidades.

### 7.1.8. Licencias EPL

La Licencia Pública Eclipse (EPL) es una licencia utilizada por la Fundación Eclipse para su software. Sustituye a la Licencia Pública Común (CPL) y elimina ciertas condiciones relativas a los litigios sobre patentes. La Licencia Pública de Eclipse está diseñado para ser una licencia de software favorable a los negocios y cuenta con disposiciones más débiles que las licencias copyleft contemporáneas. El receptor de programas licenciados EPL pueden utilizar, modificar, copiar y distribuir el trabajo y las versiones modificadas, en algunos casos están obligados a liberar sus propios cambios.

### 7.1.9. Licencia Apache

La licencia Apache es una licencia de software libre creada por la Apache Software Foundation (ASF). La licencia (con versiones 1.0, 1.1 y 2.0) requiere la conservación del aviso de copyright y el disclaimer, pero no es una licencia copyleft, ya que no requiere la redistribución del código fuente cuando se distribuyen versiones modificadas ni siquiera que se tengan que distribuir como software libre/open source, solo exige que se mantenga una noticia que informe a los receptores que en la distribución se ha usado código con la Licencia Apache.

### 7.1.10. Licencia PHP

La licencia PHP es la licencia bajo la cual se publica el lenguaje de programación PHP. De acuerdo a la Free Software Foundation es una licencia de software libre no copyleft y una licencia de código abierto según la Open Source Initiative. Debido a la restricción en el uso del término “PHP”, no es compatible con la licencia GPL.

Las continuas mejoras y avances dentro del lenguaje resultan de una gran comunidad de desarrolladores que contribuyen, sin obtener réditos comerciales, con:

- Código fuente.
- Soporte a otros usuarios a través de listas de correo.
- Revisión del programa en busca de errores.
- Notificación de fallas de seguridad y más.

Sobre esta base se sostiene una licencia que, justamente, asegura la libertad del lenguaje y no permite bajo concepto alguno que alguien obtenga beneficios comerciales de PHP y sea el dueño del lenguaje: éste es el espíritu de la licencia.

Cuando se desarrolla una aplicación y se la vende a terceros el importe que se cobra no es el lenguaje de programación sino la solución a un problema, el tiempo invertido en el desarrollo, el soporte, u otro particular.

### 7.1.11. Licencias Creative Commons

Las licencias Creative Commons permite a los usuarios usar obras protegidas por derecho de autor sin solicitar el permiso del autor de la obra. Inicialmente, estas licencias se crearon con base en la legislación estadounidense y rápidamente fueron adaptadas a las legislaciones de los diferentes países de todo el mundo.

#### Tipos de licencias de Creative Commons

Todas las licencias Creative Commons conceden ciertos derechos básicos, derecho a reproducir la obra, así como a distribuir la obra sin cargo.

- **Atribucion (BY)** El beneficiario de la licencia tiene el derecho de copiar, distribuir, exhibir y representar la obra y hacer obras derivadas siempre y cuando reconozca y cite la obra de la forma especificada por el autor o el licenciante.
- **No Comercial (NC)** El beneficiario de la licencia tiene el derecho de copiar, distribuir, exhibir y representar la obra y hacer obras derivadas para fines no comerciales.
- **No Derivadas (ND)** El beneficiario de la licencia solamente tiene el derecho de copiar, distribuir, exhibir y representar copias literales de la obra y no tiene el derecho de producir obras derivadas.
- **Compartir Igual (SA)** El beneficiario de la licencia tiene el derecho de distribuir obras derivadas bajo una licencia idéntica a la licencia que regula la obra original.

## Capítulo 8

# Documentación

Este proyecto se publica con licencia **Creative Commons BY-SA**. La cual permite a los usuarios mezclar, transformar y crear a partir del contenido de esta obra, incluso para fines comerciales. Toda obra derivada de esta publicación deberá ser distribuida bajo la misma licencia CC-BY-SA.

La documentación de este proyecto se encuentra en [GitLab](#).

## Capítulo 9

# Estado del arte

Luego de realizar una investigación sobre trabajos que plantearan algo similar a lo propuesto, tome como referencia central la [Tesis](#) de la Universidad Técnica de Manabí “Instalación y Configuración de Equipos Informáticos bajo software Libre”

El objetivo de este documento es ofrecer a la biblioteca de su facultad una alternativa al software privativo y realizar una migración exitosa de los entornos de escritorio a software libre. Este documento es una guía de buenas prácticas. El mismo ofrece una visión acerca de los pasos, procesos y elementos necesarios para migrar los equipos de la biblioteca a entornos que solamente utilicen software libre.

Este trabajo cubre por una parte lo planteado con la desventaja que nunca fue actualizado, es por ello, que decidí realizar un trabajo nuevo, detallando la migración de una red de computadoras y que aplicativos de software utilizar para que sirva como referencia para futuras migraciones de establecimientos con las mismas características.

El Software que se utilizara en esta migración fue seleccionado por poseer licencias de Software Libre y sus respectivas ventajas practicas con sus contra partes privativas.



# Capítulo 10

## Objetivos

### 10.1. Generales

Como objetivo general, lo que se pretende lograr es la migración completa a Software Libre de una red de computadoras y optimizar el rendimiento de toda la red informática. Pero que también este trabajo pueda ser replicado en diferentes de instituciones con similares características.

### 10.2. Específicos

Objetivos específicos:

- Mejorar la seguridad, funcionalidad y productividad de toda la red informática.
- Reducción de costos en la adquisición de hardware y licencias.
- Reutilización de hardware desechados por no cumplir con los requisitos exigidos por los diferentes software privativos.
- Hacer que la migración a Software Libre sea lo mas transparente y fácil de asimilar para cualquier persona que intente implementara este trabajo.

## Capítulo 11

# Desarrollo

El presente trabajo contempla la migración de la red informática del Aula Virtual, la cual cuenta con 9 estaciones de trabajo, de las cuales una será utilizada como un Servidor Docker donde se correrán los servicios que se ofrecerán en el presente/futuro a los estudiantes del espacio. Al resto de los equipos se les instalará un Sistema Operativo GNU/Linux orientado a usuario final, con las aplicaciones requeridas para el trabajo diario:

- Navegador de Internet.
- Cliente de Correo Electrónico.
- Suite de Oficina.
- Cliente FTP.
- TexStudio.

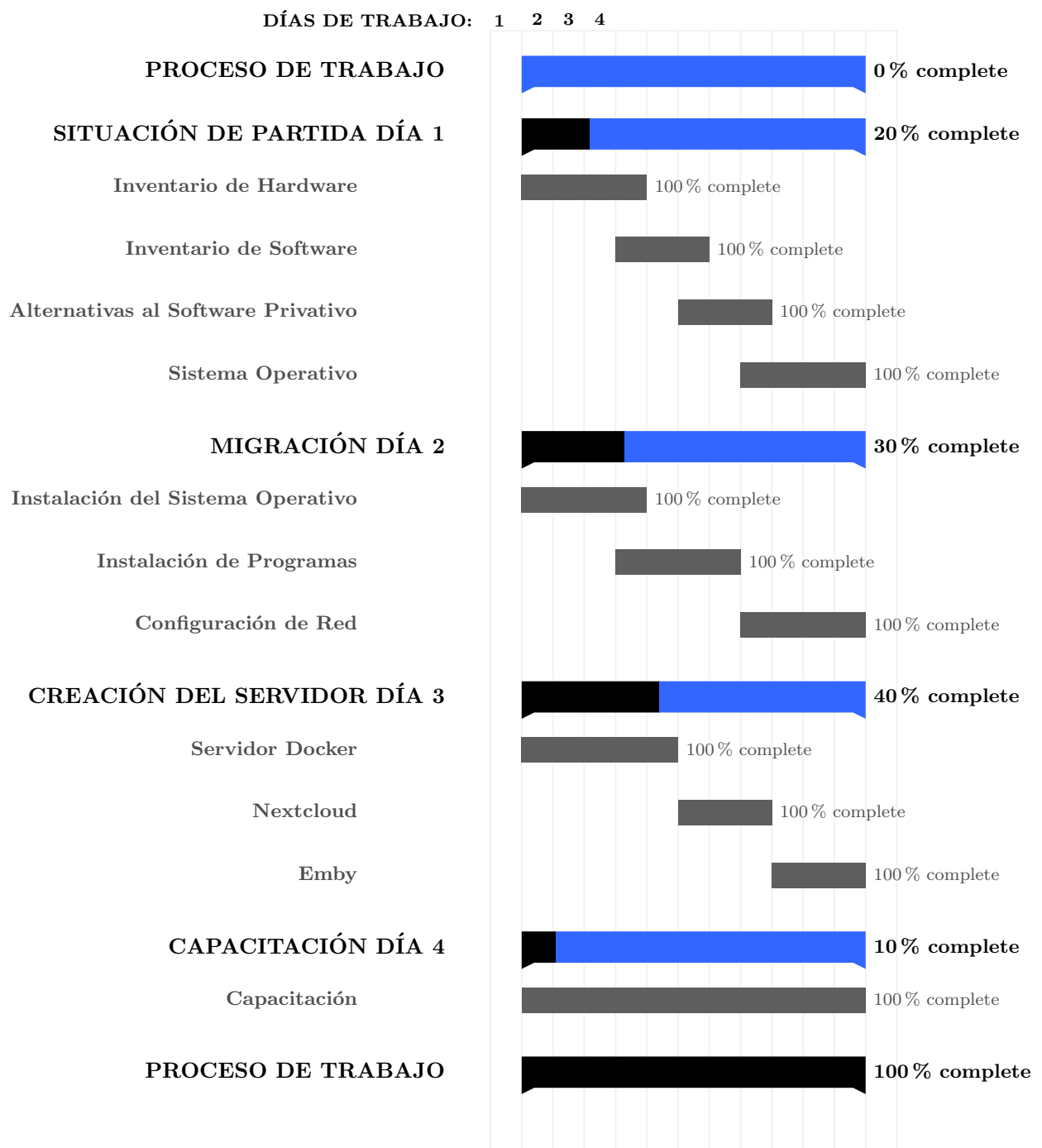
Desde comienzo se tuvo en cuenta los diferentes formatos de archivos para realizar la migración y así evitar uno de los grandes problemas que se presentan al momento de realizar las migraciones a software libre.

Las aplicaciones de software libre que se usaran soportan formatos privativos y permiten importarlos a formatos libres. No obstante, en los documentos al momento de exportarlos al formato libre se pueden perder algunas características propias del formato privativo, lo cual lleva a tener cuidado al momento de la conversión de formato.



## Capítulo 12

# Cronograma de trabajo



# Capítulo 13

## Proceso de migración

En este punto realizare un análisis de la situación de partida para tener conocimiento detallado de las arquitecturas de los hardware disponibles, documentos/aplicaciones/tipos de archivos, etc. Esto me ayudará a evitar que realice ajustes imprevistos durante la migración y establecer un plan de actuación con suficiente antelación.

Con este análisis pretendo identificar los requisitos funcionales que debe cumplir el nuevo sistema operativo.

### 13.1. Aspectos importantes

- Documentos y sus formatos.
- Archivos de audio/vídeos y sus formatos.
- Aplicaciones y sus interfaces.
- Bases de datos y estructura de datos.
- Disponibilidad de datos y aplicaciones.
- Hardware Disponible y driver necesarios.

### 13.2. Inventario de Software

El inventario de software consiste en realizar un listado de todos los programas (Aplicaciones, servicios y configuraciones) que se utilizan en los equipos que se deben migrar.

#### 13.2.1. Software Invetariado

- Windows 10 64 bits
- Navegador Chrome
- Microsoft Outlook,
- Adobe After Effect
- Adobe Acrobat
- Adobe Photoshop
- atube gatcher
- uTorrent
- Microsoft Office 2016

- Panda Cloud Antivirus
- Windows Media
- WinRAR
- CCleaner

### 13.3. Inventario de Hardware

El inventario de hardware consiste en conocer en detalle el hardware de los ordenadores que se planean migrar. Esto me ayudara a saber si estarán soportado por las distribuciones de software libre de manera nativa e identificar si hay componentes a actualizar o cualquier incidencia con el soporte de hardware.

#### 13.3.1. Hardware Inventariado

##### Desktop

Cuatro computadoras Intel con las siguientes características:

Hardware	Modelo
Mother:	Asrock H55M-LE
Procesador:	Intel Core i3 3.0GHz.
Disco Duro	500GB.
Memoria RAM	DDR3 2GB
Arquitectura	64bits.

Tres computadoras AMD con las siguientes características:

Hardware	Modelo
Mother:	MSI
Procesador:	Amd athlon2 x2 3.00GHz.
Disco Duro	500GB.
Memoria RAM	DDR3 2GB
Arquitectura	64bits.

##### Servidor

Una computadora AMD con las siguientes características:

Hardware	Modelo
Mother:	FM2
Procesador:	Amd 3.5ghz.
Disco Duro	500GB.
Memoria RAM	DDR3 8GB
Arquitectura	64bits.

### 13.4. Sistema Operativo

#### 13.4.1. Distribución GNU/Linux

Una distribución GNU/Linux es una distribución de software basada en el núcleo Linux que incluye determinados paquetes de software. Dependiendo el tipo de software que incluya surgen ediciones domésticas, empresariales y para servidores. Por lo general están compuestas, total o mayoritariamente, de software libre, aunque a menudo incorporan aplicaciones o controladores propietarios.

Además del núcleo Linux, las distribuciones incluyen habitualmente las bibliotecas y herramientas del proyecto GNU y el sistema de ventanas X Window System. Dependiendo del tipo de usuarios a los que la distribución esté dirigida se incluye también otro tipo de software como procesadores de texto, hoja

de cálculo, reproductores multimedia, herramientas administrativas, etc. En el caso de incluir paquetes de código del proyecto GNU, se denomina distribución GNU/Linux.

### 13.4.2. Debian

Seleccione Debían como mi sistema Desktop personal y para instalar en todos los ordenadores del aula universitaria porque es una distribución mantenida por la comunidad, tienen una muy buena estabilidad y a la hora de actualización de paquetes o de la propia distribución se puede lograr de una manera muy sencilla.

Debian funciona en numerosas arquitecturas y dispositivos y ofrece soporte a largo plazo (LTS).

#### Características de Debian

- **Debian es software libre:** Está hecho de Software Libre y siempre será 100 % libre.
- **Debian es estable y seguro:** Es un sistema operativo basado en Linux, y soporta una gran variedad de dispositivos, portátiles, ordenadores de escritorio y servidores. Proporciona una configuración predeterminada para cada paquete, así como actualizaciones de seguridad con regularidad durante su ciclo de vida.
- **Debian tiene un soporte de hardware extenso** La mayoría del hardware ya está soportado por el núcleo Linux, lo que significa que está soportado en Debian también. Hay disponibles controladores no libres para el hardware si fueran necesarios.

## 13.5. Instalación del sistema operativo

Antes de comenzar con la instalación del Sistema Operativo en los ordenadores, realizare el respaldo de toda la información de los alumnos en un disco externo.

Este respaldo se puede realizar de varias maneras, siempre es recomendable que se haga en otro medio físico de almacenamiento.



### 13.5.1. Requisitos

Los requisitos mínimos necesarios para la instalación de **Debían Cinnamon** son los siguientes:

- 1 GB de RAM (se recomiendan 2 GB para un uso cómodo).
- 15 GB de espacio en disco (se recomiendan 20 GB).
- Resolución de 1024×768.
- Procesador 2Ghz, doble núcleo.

### 13.5.2. Proceso de Instalación

#### Primeras configuraciones

Para el proceso de instalación seguiré una serie de pasos que me servirán para ir configurando el sistema operativo. Este proceso se puede repetir en cada uno de los ordenadores o clonar el disco con alguna herramienta.

- **Seleccionar idioma:** Seleccionar el idioma del proceso de instalación.
- **Seleccionar su Ubicación:** Seleccionar la ubicación geográfica. Así Debian podrá configurar automáticamente la zona horaria y la localización.
- **Configure el Teclado:** Seleccionar el idioma de nuestro teclado.

Ahora que el idioma y el teclado de la distro esta configurado, Debian se encargara de realizar las demás configuraciones.

### Detectar y configurar el hardware

Después de las primeras configuracionesq Debian dedicará un tiempo a volcar los datos del medio de instalación (USB o DVD) al ordenador para poder acceder a ellos más rápidamente. También aprovechará para detectar todo el hardware del ordenador, configurarlo y conectarlo a Internet.

Este proceso puede tardar varios minutos dependiendo de la potencia del PC.

- **Configurar red:** El nombre que se le coloque al PC sera con el cual se identificara en la red local.
- **Configurar usuarios y contraseña:** Lo primero que pedirá Debian es introducir una contraseña para el superusuario. Esta contraseña debe ser larga y robusta, para proteger el equipo.

Después de haber creado la cuenta de superusuario es hora de crear el primer usuario. Debian solicitara que se introduzca en primer lugar el nombre completo del mismo y luego la contraseña que se le asignara.

- **Configurar Reloj:** A continuación,se debe seleccionar la ubicación de la zona horaria para que se ajuste el reloj automáticamente.

### Particionado

Ha llegado el turno al disco duro. Debian, como cualquier otro Linux utiliza una serie de particiones y puntos de montaje necesarios para poder guardar todos los datos. En este caso se puede usar todo el disco y usar el método guiado para que el proceso de creación de las particiones sea lo más sencillo posible.

Por fines prácticos creare un LVM y seleccionare el **método guiado** para usar todo el disco.

- **Paso N° 1:** Seleccionar el disco duro donde se va a instalar la distribución
- **Paso N° 2:** Seleccionar la estructura de particiones que se va a utilizar.

El asistente de instalación de Debian hará los cálculos correspondientes y mostrara un resumen con todos los cambios que se van a realizar.

Si está todo correcto se debe aceptar los cambios y así comenzará la instalación como tal.

### Comenzando la copia de archivos para instalar Debian

- **Instalando el sistema base:** El sistema se instala en el PC. Durante este proceso se descargarán los paquetes que faltan y se configurará la distribución para poder funcionar.
- **Configurar el gestor de paquetes:** Al acabar la copia de archivos, el asistente preguntará si hay otro CD o DVD de software para utilizar e instalar al mismo tiempo. Esto era bastante frecuente en el pasado, pero ahora ya está en desuso. Por tanto, como no se dispone otro medio de instalación, seleccionare la opción «No».
- **Configurar el gestor de paquetes:** Ahora Debian solicitara la ubicación del servidor de repositorios, desde donde se descargarán los paquetes y las actualizaciones. Es recomendable que este servidor esté lo más cerca posible para que la velocidad sea lo mayor posible. También se debe seleccionar el servidor desde donde bajar todo. Generalmente, el que viene por defecto (deb.debian.org).

Con todo esto el asistente de instalación de Debian se conectará a los repositorios y actualizará la base de datos de software, descargando otras dependencias o listas que no se encuentren en el sistema.

- **Configurar el popularity-contest:** Configurar si se compartirán datos anónimos de uso. Por defecto, no.



### Instalando software adicional

- **Selección de programas:** Seleccionar el entorno de escritorio.

Como ultimo paso se debe instalar el gestor de arranque **GRUB**, e indicar en qué unidad queremos instalarlo.

Guía de Instalación

Manual de Debían

## 13.6. Repositorios

### 13.6.1. ¿Que es un repositorio?

Un repositorio es un servidor web que se encarga de almacenar paquetes y programas para que los usuario puedan descargarlos e instalarlos en su distribución GNU/Linux. Además de los repositorios pre-determinados del sistema operativo, se puedes disponer y utilizar repositorios de terceros, que contengan actualizaciones de software o programas que no han sido incluidos en los repositorios iniciales.

### 13.6.2. Configuración de repositorios Debian

La lista a editar se encuentra alojada en `/etc/apt/source.list`. Para obtener información sobre los repositorios disponibles por parte de [Debian.org](https://www.debian.org), se puede hacer uso de una [Wiki](#) que cuenta con información detallada de cada repositorio.

Mi lista de repositorios.

```

1  deb http://deb.debian.org/debian/ bullseye main
2
3  deb-src http://deb.debian.org/debian/ bullseye main
4
5  deb http://security.debian.org/debian-security bullseye-security main contrib
6
7  deb-src http://security.debian.org/debian-security bullseye-security main contrib
8
9  bullseye-updates, to get updates before a point release is made;
10
11  # see https://www.debian.org/doc/manuals/debian-reference/ch02.en.html#
12  _updates_and_backports
13
14  deb http://deb.debian.org/debian/ bullseye-updates main contrib
15
16  deb-src http://deb.debian.org/debian/ bullseye-updates main contrib
17
18
```

Código 13.1: Repositorios

## 13.7. Instalación de Programas

La mayoría de los programas que necesita el estudiante ya se instalan de forma automática y otros los tendré que instalar de forma manual por medio de la terminal.

Los programas que debo instalar en los equipos son los siguientes:

- **Brave:** Brave es un navegador totalmente gratuito y de código abierto para ordenador o para teléfonos móviles que destaca por la privacidad, por la velocidad.
- **Chromium:** Chromium es una versión de código abierto de Google Chrome, pero sin todos los códecs exclusivos y otros elementos con los que Google pretende diferenciar Chrome de otros navegadores.

- **Cliente mail Thunderbird:** Thunderbird es el cliente de correo electrónico de la Fundación Mozilla, la misma responsable del navegador Firefox. Thunderbird fue creado para cubrir las necesidades de tener un gestor de correo ligero y gratuito.
- **Evince:** Evince es un visor de documentos para el entorno de escritorio GNOME. Se pueden ver los archivos en formato PDF y PostScript.
- **Gimp:** GIMP (GNU Image Manipulation Program) es un programa de edición de imágenes. Se trata de un software libre, englobado dentro del proyecto GNU y disponible bajo esta licencia pública y la de GNU Lesser General Public License.
- **Inkscape:** Inkscape es un editor de gráficos vectoriales que permite: Diseñar imágenes de calidad, básicas o complejas. Crear y editar diagramas, líneas, gráficos, logotipos, cómics, folletos, entre otros. Utilizar formas, trazos, texto, marcadores, clones, y otras herramientas de diseño.
- **HandBrake:** HandBrake es un programa Software Libre que permite editar archivos de audio y videos.
- **qBittorrent:** qBittorrent es un cliente P2P, Software Libre y se utiliza para la transferencia de archivos grandes.
- **FileZilla:** FileZilla es un programa software libre y funciona a nivel cliente/servidor. Con esta herramienta podemos conectarnos a un servidor para consultar, adquirir y manipular contenido del mismo.
- **VLC:** VLC es un reproductor y framework de vídeo y música software libre compatible con un gran número de formatos de archivo multimedia. Es capaz de reproducir la mayoría de códecs sin tener que descargar paquetes adicionales.
- **BleachBit:** BleachBit es una herramienta software libre que se encarga de eliminar: Cache, Cookies, Archivos temporales, Historiales, Registros de chats, Thumbnails (miniaturas), Historial de descargas, Accesos directos inválidos, Registros de depuración.
- **TeXstudio** TeXstudio es un editor de  $\text{\LaTeX}$  multiplataforma software libre.

Entre sus funciones se destacan: Marcadores, autocompletado de comandos, coloreado de sintaxis, menús, soporte de arrastrar imágenes, asistente para la creación de tablas, fórmulas, etc, soporte integrado de diversos compiladores de  $\text{\LaTeX}$ .

Para instalar el navegador Brave es necesario que agregue el repositorio donde se encuentra el navegador

```

1  #herramientas necesarias para agregar el repositorio.
2  apt install apt-transport-https curl
3
4  #repositorio de Brave
5  curl -fsSLo /usr/share/keyrings/brave-browser-archive-keyring.gpg https://brave-
6  browser-apt-release.s3.brave.com/brave-browser-archive-keyring.gpg
7
8  echo "deb [signed-by=/usr/share/keyrings/brave-browser-archive-keyring.gpg arch=
9  amd64] https://brave-browser-apt-release.s3.brave.com/ stable main" | tee /etc/apt
10 /sources.list.d/brave-browser-release.list
11
12 #actualizar la lista de paquetes disponibles.
13 apt update
14
15 #instalar programas.
16 apt install brave-browser texstudio bleachbit vlc filezilla qbittorrent handbrake
17 inkscape gimp evince thunderbird chromium

```

Código 13.2: Instalación de programas

## Capítulo 14

# Configuración red LAN

### 14.1. ¿Que es una red LAN?

Una red LAN es un grupo de dispositivos (impresoras, sistemas, dispositivos móviles, consolas de juegos) interconectados mediante alguna tecnología, y que comparten datos entre sí. Una LAN puede ser una entidad autónoma e independiente, así como una parte de redes más grandes y extensas.

### 14.2. ¿Que es una dirección IP?

una dirección IP (Internet Protocol que corresponde al nivel de red del modelo TCP/IP) es un número que identifica, de manera lógica y jerárquica, a una Interfaz red de un dispositivo. En un paquete IP, el encabezado contiene una dirección de 32 bits (IPv4) o 128 bits (IPv6). Este identificador es único y no se repite en ningún otro equipo en el mundo y tiene la tarea de registrar el equipo a la red global. Pero una dirección IP no solo la poseen los equipos de cómputo, también los módems, routers, sitios web, etc.

Entendiendo el papel de una dirección IP es necesario saber que existen dos tipos de direcciones IP y que se manejan dos protocolos para éstas. Éstos son el protocolo IPv4, pero con la demanda cada día mayor de solicitudes de direcciones IP se está implementando el protocolo IPv6. El cual ofrece mayor direcciones.

En el mundo del direccionamiento IP se encuentran dos tipos de direcciones IP, estática y dinámica.

#### 14.2.1. Direcciones IP dinámicas

Son direcciones variables, entregadas y administradas por un servidor DHCP y su funcionamiento radica en el arrendamiento de esta dirección por un tiempo específico, luego de este periodo de tiempo la dirección se renovará modificando su sintaxis.

#### 14.2.2. Direcciones IP estáticas

Como su nombre lo indica, son direcciones IP que permanecerán fijas, sin ningún tipo de variación. Son usadas en servidores, máquinas de producción conectadas a la red y en general todo usuario que no requiere que su IP sea modificada, ya que de él dependen otros servicios.

### 14.3. Configuración de red

Por simplicidad y dado que la red del aula es una red pequeña las direcciones IP para los ordenadores de escritorios serán dinámicas. El servidor recibirá una dirección IP estática para que las estaciones de trabajo puedan encontrarlo en la red.

La red tendrá la dirección IP 192.168.1.0/24 y se establecerá la dirección IP en 192.168.1.222 para el servidor Docker .

### 14.3.1. Diagrama de red

El diagrama de red es la representación visual de los ordenadores que compondrán la red de trabajo y su interconexión.

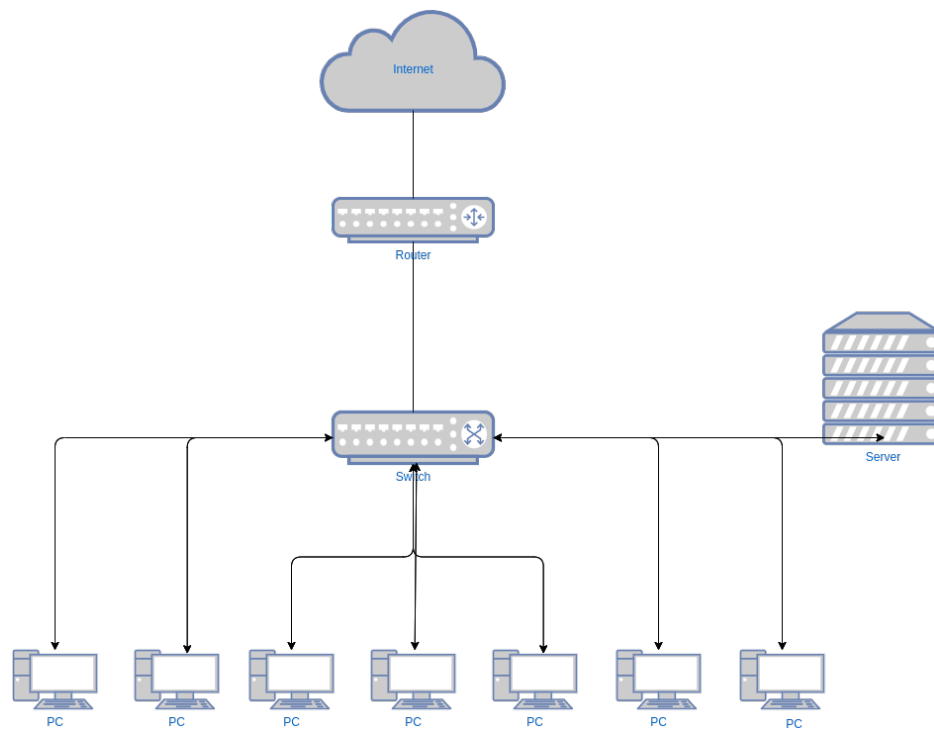


Figura 14.1: Diagrama de red

# Capítulo 15

## Servidor Local

### 15.1. ¿Que es un Servidor Local

Definiré a un servidor local como un ordenador que estará permanentemente encendido y conectado a la red LAN. Este ordenador proporcionara una serie de servicios y contenidos sin importar el dispositivo que se utilice. Al no ser consultado directamente no sera necesario tener un monitor y teclado conectado.

### 15.2. Uso del servidor

El servidor local proporcionara servicio de alojamiento de archivos y Streaming.

- **Servidor de archivos:** Permite alojar, sincroniza y compartir archivos con otros usuarios y dispositivos que se encuentren en la misma red. Al almacenarse localmente permitirá tener el control total de los datos.
- **Servidor Streaming:** gestiona el contenido multimedia (música, películas, series, documentales, etc), otorgando la posibilidad de reproducirlo en cualquier dispositivos que se encuentren en la misma red.

### 15.3. Sistema Operativo

Por las características propias de Debian no necesitare buscar un sistema operativo orientado a servidores, dado que me permitirá crear un potente servidor, al poseer estabilidad/escalabilidad y buenos repositorios.

### 15.4. Instalación del Sistema Operativo

Una vez que comience el proceso de instalación, seguiré una serie de pasos que me permitirán instalar y configurar el sistema operativo :

- **Graphical Install**
- **Localización:** Seleccionar el idioma de instalación que será el idioma utilizado por el sistema. (Por motivos de compatibilidad, se recomienda seleccionar English) Después indicar la localización geográfica del servidor.
- **Configure Locale:** Seleccionar “United Estates” para evitar conflictos de compatibilidad.
- **Configuración del teclado:** Seleccionar Spanish o Latin America.
- **Conexión a Internet:** Para conectarse a Internet se necesita la asignación de una dirección IP a la interfaz de red. La dirección IP y los demás parámetros de la red pueden obtenerse de forma automática a partir de un servidor DHCP o se pueden configurar manualmente.
- **Nombre del sistema:** Nombre con el que se conocerá al sistema en la red.

- **Cuentas de usuario y passwords:** El programa instalador solicita la creación y configuración de dos cuentas de usuario del sistema o logins. La primera es la cuenta de root y la segunda es una cuenta de usuario ‘normal’, con permisos limitados.
- **Reloj del sistema y huso horario:** El instalador intentará sincronizar el reloj del sistema con uno de los servidores que establecen la hora oficial en el Internet.
- **Particionado:** Por fines prácticos creare un LVM y seleccionare el método guiado para usar todo el disco. Luego Gestionare el GV de la siguiente manera:

GNU/Linux tiene un nivel de personalización muy amplio, es por ello, que seguramente no todos verán necesario lo que realizare, así como quizás haya alguien que piense que hay muchos otros directorios más susceptibles que no se mencionan y deben alojarse en un (LV).

La gestión de volúmenes consistirá en organizar la raíz del sistema en varios volúmenes. Cada uno de estos pueden tener un objetivo y/o un tipo de sistema de archivos distinto y específico.

A continuación mencionare los directorios que tendrán su propio (LV), dado el uso que le daré al servidor.

- **Directorio /var:** En este directorio se encontraran todo tipo de archivos, de los cuales se espera que en una ejecución normal del sistema estén cambiando continuamente. Estos podrían ser “logs, bases de datos, spool, cachés, e-mail, etc”.
  - **Directorio /tmp:** En este directorio es donde las aplicaciones almacenarán archivos temporales. Tiene la particularidad de que siempre será borrado tras un reinicio del sistema.
  - **Directorio /usr:** Es con frecuencia grande, debido a que todos los programas están instalados aquí.
  - **Directorio /srv:** Sirve para almacenar archivos y directorios relativos a servidores.
  - **/var/lib/docker** Los volúmenes se almacenan en /var/lib/docker/volumes/. Docker almacena los datos dentro de este directorio.
  - **Directorio /home:** Es donde se crean los directorios de los usuarios normales.
  - **Directorio /opt:** En este directorio se instalan algunos programas de terceros, además es el directorio donde suelo guardar ejecutables.
- **Instalación del sistema base:** En este paso el instalador comenzará la instalación de los paquetes de aplicaciones necesarios para crear un sistema base. Este proceso puede tardar algún tiempo.
  - **Configuración del gestor de paquetes apt:** Seleccionar el repositorio geográficamente más cercano al equipo que se está instalando. En primer lugar se debe elegir el país y luego escoger el mirror más próximo.
  - **Concurso de popularidad:** La comunidad Debian mantiene un concurso de popularidad interno, con el fin de obtener estadísticas sobre los sistemas instalados. La instalación de este paquete implica la instalación de otros paquetes. No se recomienda instalar el paquete de popularidad para que el sistema sea lo más ligero posible.
  - **Finalizar instalación:** El instalador permite la instalación automática de diversas configuraciones del sistema. Como se busca personalizar totalmente el sistema, se anulará cualquier selección existente. De esta forma se instalará un sistema con un mínimo de funcionalidades.
  - **Instalación del gestor de arranque grub:** En este punto el sistema ya está casi completamente instalado. Sin embargo, para que el sistema pueda arrancar hay que instalar el gestor de arran que “grub” en el master boot record (mbr) del disco. Una vez instalado el grub se debe seleccionar continuar, y el equipo se reiniciara automática mente para el primer login.

## 15.5. LVM

### 15.5.1. ¿Que es LVM?

LVM es una capa de abstracción entre un dispositivo de almacenamiento y un sistema de ficheros. Las ventajas que tienen son múltiples, pero la inicial y por la cual lo uso, es por la flexibilidad frente al particionado tradicional. Con LVM las limitaciones en el redimensionado y particionado desaparecen. Se puede aumentar el tamaño de los volúmenes lógicos independientemente de que no haya espacio libre contiguo.

- **Volumen físico o Physical Volume (PV):** Un volumen físico es un dispositivo de bloque que se le entregará al LVM para que este lo gestione.
- **Grupo de volúmenes o Volume Group (VG):** Para poder usar el espacio de almacenamiento de un PV, éste debe pertenecer a un Grupo de volúmenes. Un VG es una especie de disco duro virtual compuesto de uno o más PVs.
- **Volumen Lógico o Logical Volume (LV):** Los volúmenes lógicos son dispositivos en los cuales se crea sistemas de archivos. Por seguir con la analogía del «*disco duro virtual*» que es el VG, los LVs serían las particiones.

### 15.5.2. Gestión volúmenes lógicos

Los volúmenes lógicos son donde todos los datos se almacenan en un LVM. Para crear los nuevos volúmenes lógicos, utilizare los siguiente comandos.

```

1  lvcreate -L 10G -n home ema
2  lvcreate -L 10G -n docker ema
3  lvcreate -L 100G -n srv ema
4  lvcreate -L 5G -n opt ema
5  lvcreate -L 1G -n tmp ema
6  lvcreate -L 15G -n usr ema
7  lvcreate -L 1G -n var-tmp ema
8  lvcreate -L 1G -n var-log ema
9  lvcreate -L 2G -n swap ema
10
11
12
```

Código 15.1: Creación de (LV)

La sintaxis básica para crear volúmenes lógicos es:

`lvcreate -L «tamaño»G -n «nombre del volumen» «nombre del grupo»`

- **-L** Tamaño en GB o MB.
- **-n** Nombre que tendrá el (LV) Nombre del VG con el que se trabajara.

Luego de que los (LVs) estén disponibles, les creare un sistema de archivo:

```

1  mkfs.ext4 /dev/mapper/ema-var--log
2  mkfs.ext4 /dev/mapper/ema-var--tmp
3  mkfs.ext4 /dev/mapper/ema-tmp
4  mkfs.ext4 /dev/mapper/ema-opt
5  mkfs.ext4 /dev/mapper/ema-home
6  mkfs.ext4 /dev/mapper/ema-docker
7  mkfs.ext4 /dev/mapper/ema-usr
8  mkfs.ext4 /dev/mapper/ema-srv
9  mkfs.ext4 /dev/mapper/ema-srv
10 mkswap /dev/mapper/ema-swap
11
12
```

Código 15.2: Crear Sistema de Archivo

Una vez creados los sistemas de archivo es hora montar los (LVs) y mover el contenido de los directorios al (LV) correspondiente.

```

1      mv /mnt
2
3
4      mkdir {var-log,opt,home,usr,srv}
5
6      mount /dev/mapper/ema-var--log var-log
7      mount /dev/mapper/ema-opt opt
8      mount /dev/mapper/ema-home home
9      mount /dev/mapper/ema-usr usr
10     mount /dev/mapper/ema-srv srv
11
12

```

Código 15.3: Crear directorios y montar los (LVs)

Mover todo el contenido de los directorios a su (LV) correspondiente.

```

1
2      #Mover el contenido de los directorios
3      mv -f var/log/* var-log
4      mv opt/* opt
5      mv -f home/* home
6      mv usr/* usr
7      mv srv/* srv
8      #Eliminar archivos temporales
9      rm -rf var/tmp/*
10     rm -rf tmp/*
11
12

```

Código 15.4: Mover directorios

## 15.6. fstab

El archivo `/etc/fstab` es el archivo de configuraciones que une dispositivos con punto de montaje. Le indica al sistema cómo montar cada dispositivo y qué configuración utilizar.

Tabla de configuración del archivo `fstab`:

```

1      # <file system> <mount point> <type> <options> <dump> <pass>
2
3
4

```

Código 15.5: `fstab`

- **file system:** Esta primera columna guarda el dispositivo a montar.
- **mount point:** Esta opción permite definir el directorio que se va a asociar con el (LV) definido en la primera columna.
- **Tipo:** Este campo se refiere al tipo de filesystem, por ejemplo, `ext4`, `xf`s, entre otros.
- **Opciones:** Hace referencia a las opciones de montaje. Deben ir separadas por comas.

Algunas de las opciones que se pueden usar son:

- **auto:** para que la partición se monte al arrancar.
- **noauto:** opción que impide que la partición se monte durante el arranque.
- **user:** los usuarios tienen permitido montar la partición.
- **nouser:** solo el usuario `root` tiene permitido realizar el montaje de la partición.
- **ro:** partición que solo permite la lectura.
- **rw:** opción que permite la lectura escritura.
- **exec:** es posible ejecutar los binarios pertenecientes a esa partición.



- **async:** esta opción permite que el sistema continúe trabajando luego de una petición de escritura del equipo, aunque no haya recibido la confirmación.
  - **suid:** esta opción permite las operaciones con los bits suid y sgid. Es utilizado para permitir a los usuarios diferentes del root, ejecutar binarios con ciertos privilegios otorgados temporalmente para que realicen una labor determinada.
  - **nosui:** se encarga de impedir el funcionamiento de los bits suid y sgid.
  - **noatime:** esta opción no actualiza el nodo-i de los ficheros con el tiempo de acceso. Además, permite aumentar las prestaciones del sistema, debido a que accede menos al disco.
  - **nodiratime:** con esta opción se impide la actualización del nodo-i de los directorios con el tiempo de acceso. Al igual que la opción noatime, también puede aumentar las prestaciones del sistema.
  - **defaults:** establece que las opciones sean asignadas por defecto gracias al sistema operativo. Estas opciones predeterminadas son rw, suid, dev, exec, auto, nouser y async.
- **Soporte a Dump:** Este campo es requerido por algunas soluciones de backup. Además, determina la frecuencia con la que debe realizarse la copia de seguridad.
  - **Chequeo automático:** Es el campo encargado de especificar si el sistema de ficheros debe ser revisado durante el arranque, si el formato es correcto, entre otros. Normalmente ese campo se deshabilita para todas las particiones, a excepción del /.

### 15.6.1. Configurar fstab

Es necesario que defina que los (LVs) se monten automáticamente cuando se está iniciando el sistema operativo. La forma correcta para resolver este problema es modificando el archivo fstab.

Finalizada la configuración mi archivo fstab quedara de la siguiente manera:

```
# <file system> <mount point> <type> <options> <dump> <pass>
/dev/mapper/ema-raiz / ext4 errors=remount-ro 0 1
# /boot/efi was on /dev/mmcblkOp1 during installation
UUID=2321-DD09 /boot/efi vfat umask=0077 0 1
/dev/mapper/ema-home /home ext4 rw,noexec,auto,nodev,nosuid,noatime,async,nouser 0 2
/dev/mapper/ema-opt /opt ext4 rw,exec,auto,nodev,nosuid,noatime,async,nouser 0 0
/dev/mapper/ema-srv /srv ext4 defaults 0 0
/dev/mapper/ema-docker /var/lib/docker ext4 defaults 0 0
/dev/mapper/ema-var--log /var/log ext4 rw,noexec,auto,nodev,nosuid,noatime,async,nouser 0 0
/dev/mapper/ema-var--tmp /var/tmp ext4 rw,noexec,auto,nodev,nosuid,noatime,async,nouser 0 0
/dev/mapper/ema-tmp /tmp ext4 rw,noexec,auto,nodev,nosuid,noatime,async,nouser 0 0
/dev/mapper/ema-usr /usr ext4 defaults 0 0
/dev/mapper/ema-swap none swap sw 0 0
```

## 15.7. Controlar cambios en el sistema

Por cuestiones de seguridad tomo como buena practica generar HASH de los archivos que se encuentran en algunos directorios que considero importante, por si en algún momento tengo la duda de que se generaron cambios en el sistema, poder realizar las comparaciones de HASH.

### 15.7.1. Generar Hash

md5sum es un algoritmo de reducción criptográfico de 128-bits desarrollado por MIT. Se utiliza con frecuencia como herramienta para validar un archivo.

```
#!/bin/bash

workdir=$PWD/seguridad

echo "Crear Base de datos del HASH de cada archivo"

echo ""

find /usr -type f -exec md5sum {} \; > $workdir/usr.txt
find /boot -type f -exec md5sum {} \; > $workdir/boot.txt
find /opt -type f -exec md5sum {} \; > $workdir/opt.txt
find /etc -type f -exec md5sum {} \; > $workdir/etc.txt
find /var -type f -not -path "/var/pool/*" -not -path "/var/log/*" \
    -not -path "/var/tmp/*" -exec md5sum {} \; > $workdir/var.txt

echo ""

echo "Fin..."
```

### 15.7.2. Comparar Hash

Si por alguna razón necesito validar la integridad de los archivos del sistema, utilizo el siguiente script en bash.

```
#!/bin/bash
#crear directorio llamado "seguridad"
workdir=$PWD/seguridad
echo "Crear Base de datos del HASH de cada archivo"
echo ""

find /usr -type f -exec md5sum {} \; > $workdir/usr.tmp
find /boot -type f -exec md5sum {} \; > $workdir/boot.tmp
find /opt -type f -exec md5sum {} \; > $workdir/opt.tmp
find /etc -type f -exec md5sum {} \; > $workdir/etc.tmp
find /var -type f -not -path "/var/pool/*" -not -path "/var/log/*" \
    -not -path "/var/tmp/*" -exec md5sum {} \; > $workdir/var.tmp

echo ""
echo "Diferencias..."

diff $workdir/usr.txt $workdir/usr.tmp
diff $workdir/boot.txt $workdir/boot.tmp
diff $workdir/opt.txt $workdir/opt.tmp
diff $workdir/etc.txt $workdir/etc.tmp
diff $workdir/var.txt $workdir/var.tmp

echo ""
echo "Limpiar"

rm -f $workdir/usr.tmp $workdir/boot.tmp $workdir/opt.tmp \
    $workdir/etc.tmp $workdir/etc.tmp $workdir/var.tmp
```

## 15.8. Configuración de red

### 15.8.1. Asignar una IP estática

El servidor tendrá una dirección IP estática, la cual se la asignare editando el archivo *interfaces*”.

```
1 nvim /etc/network/interfaces
2
```

Código 15.6: Interfaces

A continuación presento el contenido del archivo */etc/network/interfaces* con una configuración de IP estática:

```
1 # This file describes the network interfaces available on your system
2 # and how to activate them. For more information, see interfaces(5).
3 source /etc/network/interfaces.d/*
4 # The loopback network interface
5 auto lo
6 iface lo inet loopback
7 # Interfaz de red enp2s0
8 auto enp2s0
9 allow-hotplug enp2s0
10 iface enp2s0 inet static
11     address    192.168.1.222
12     netmask    255.255.255.0
13     network    192.168.1.0
14     broadcast  192.168.1.255
15     gateway    192.168.1.1
16
```

Código 15.7: Configuración de IP estática

- **iface enp2s0 inet static:** define el nombre lógico de la interfaz de red enp2s0, la *address\_family* que es **inet** y el método de configuración de la interfaz que es **static**.
- **address:** define la IP que tendrá el servidor.
- **netmask:** define la máscara de subred.
- **network:** define la dirección de red.
- **broadcast:** define la IP de difusión (broadcast).
- **gateway:** define la IP de la puerta de enlace (gateway).

Una vez finalizada la edición del archivo reiniciare el servicio de red para aplicar los cambios.

```
1 systemctl restart networking.service
2
```

Código 15.8: Reiniciar servicio de RED

### 15.8.2. Servidor DNS

Es necesario que tambien configure los servidores DNS. Para configurarlos editare el archivo */etc/resolv.conf*:

```
1 nvim /etc/resolv.conf
2
```

Código 15.9: Editar archivo resolv

A continuación presento el contenido del archivo */etc/resolv.conf* con dos servidores DNS:

```
1 # Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
2 #     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
3 nameserver 1.1.1.1
4 nameserver 8.8.8.8
5
```

Código 15.10: Servidores DNS

- **1.1.1.1:** Es un DNS público operado por Cloudflare que ofrece una forma rápida y privada de navegar por Internet. A diferencia de la mayoría de los servidores de DNS, 1.1.1.1 no vende los datos de los usuarios a los anunciantes.
- **8.8.8.8:** Como una forma de hacer más rápida la web Google tiene, Google Public DNS. Que es una forma de navegar fuera de los límites del proveedor de servicios de internet.

## 15.9. Firewall

### 15.9.1. ¿Que es iptables?

Iptables es una herramienta de filtrado de paquetes en Linux. Se encarga de analizar cada uno de los paquetes del tráfico de red que entra en una máquina y decidir, en función de un conjunto de reglas, qué hacer con ese paquete.

Gracias a la capacidad de analizar el tráfico y ver las características del paquete, se puede implementar un cortafuegos que controle qué paquetes puede llegar al equipo.

#### Características de iptables:

- Es el sucesor de ipfwadm e ipchains, un software previo que existía en sistemas Linux.
- Se encuentra disponible desde la rama 2.4 del kernel de Linux, que se lanzó el año 2001.
- Está desarrollado por el proyecto netfilter. Este mismo proyecto ha desarrollado también el sucesor de iptables, que se denomina nftables, aunque en la actualidad todavía está mucho más extendido el uso de iptables que el de nftables.

### 15.9.2. Reglas iptables

El siguiente script en bash me permitirá configurar el el firewall de linux.

```
#!/bin/bash
#-----LIMPIAR TODAS LAS REGLAS-----
iptables -F
iptables -X
iptables -Z
iptables -t nat -F
#-----POLITICA POR DEFECTO-----
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP
#-----PERMITIR RESPUESTA A TODAS LAS SESIONES ESTABLECIDAS-----
#PAQUETES QUE LLEGAN AL FIREWALL ESTABLISHED
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
#PAQUES QUE SALEN ESTABLISHED
iptables -A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
#-----PERMITIR SSH AL FIREWALL -----
iptables -A INPUT -s 192.168.1.4 -p tcp --dport 2222 -j ACCEPT #MI PC
#-----PERMITO SALIDAS DEL FIREWALL POR LOS PUERTOS-----
iptables -A OUTPUT -o enp2s0 -p tcp --dport 80 -j ACCEPT #HTTP
iptables -A OUTPUT -o enp2s0 -p tcp --dport 443 -j ACCEPT #HTTPS
iptables -A OUTPUT -o enp2s0 -p tcp --dport 53 -j ACCEPT #DNS
#-----PERMITO QUE LA RED LAN USE EL SERVIDOR NEXTCLOUD Y EMBY -----
iptables -A INPUT -s 192.168.1.0/24 -p tcp --dport 8080 -j ACCEPT
iptables -A INPUT -s 192.168.1.0/24 -p tcp --dport 8096 -j ACCEPT
```

## 15.10. Servidor Docker

### 15.10.1. ¿Que es Docker?

Docker es una plataforma de contenerización de código abierto. Permite a los desarrolladores empaquetar aplicaciones en contenedores. Los contenedores combinan el código fuente de la aplicación con las bibliotecas del sistema operativo y las dependencias necesarias para ejecutar el código en cualquier entorno. Los contenedores simplifican la entrega de aplicaciones distribuidas.

Docker es esencialmente un kit de herramientas que permite a los desarrolladores crear, implementar, ejecutar, actualizar y detener contenedores utilizando comandos a través de una única API.

### 15.10.2. Funcionamiento de los contenedores?

Los contenedores son posibles gracias al aislamiento de procesos y a las capacidades de virtualización integradas en el kernel de Linux. Estas funcionan como grupos de control (Cgroups) para asignar recursos entre procesos, y espacios de nombres para restringir un acceso de procesos o visibilidad a otros recursos o áreas del sistema.

Como resultado, los contenedores ofrecen toda la funcionalidad y beneficios de las máquinas virtuales, incluyendo aislamiento de aplicaciones, escalabilidad y capacidad de disposición. Sus ventajas son las siguientes:

- **Ligero:** Los contenedores sólo incluyen los procesos y dependencias del sistema operativo necesarias para ejecutar el código.
- **Mejor manejo de los recursos:** Con contenedores se puede ejecutar varias veces las mismas copias de una aplicación en el mismo hardware.
- **Mayor productividad de los desarrolladores:** En comparación con las VM, los contenedores son más rápidos y fáciles de implementar, suministrar y reiniciar.

### 15.10.3. Características

- **Portabilidad:** Los contenedores LXC suelen hacer referencia a configuraciones específicas de la máquina, mientras que los contenedores Docker se ejecutan sin modificaciones en cualquier entorno de escritorios, centro de datos y nube.
- **Ligeros:** Con los contenedores Docker solo se puede ejecutar un proceso en cada contenedor. Esto permite crear una aplicación que puede continuar ejecutándose mientras una de sus partes se desactiva.
- **Creación automática de contenedores:** Docker puede crear automáticamente un contenedor basado en el código de origen de la aplicación.
- **Control de versiones:** Docker puede rastrear versiones de una imagen de contenedor, retroceder a versiones anteriores y rastrear quién creó una versión.
- **Reutilizar contenedores:** Los contenedores existentes se pueden utilizar como imágenes base, esencialmente como plantillas para crear nuevos contenedores.
- **Repositorio de contenedores:** Los desarrolladores pueden acceder a un registro de código abierto que contiene miles de contenedores aportados por los usuarios.

### 15.10.4. Herramientas de Docker

- **DockerFile:** Cada contenedor Docker se inicia con un archivo de texto simple que contiene instrucciones para crear la imagen del contenedor Docker. El DockerFile es una lista de instrucciones de interfaz de línea de comandos que Docker Engine ejecutará para ensamblar la imagen.
- **Imágenes Docker:** Las imágenes Docker contienen el código de origen de la aplicación, herramientas, bibliotecas y dependencias que el código de la aplicación debe ejecutar como contenedor. Las imágenes Docker son archivos de sólo lectura.

- **Contenedores Docker:** Los contenedores Docker son las instancias activas en ejecución de imágenes Docker.

Los contenedores son contenido en vivo, efímero y ejecutable. Se pueden interactuar con ellos utilizando comandos Docker.

- **Docker Hub:** Es el repositorio público de imágenes Docker. Las imágenes de contenedor provienen de proveedores de software comercial, proyectos de código abierto y desarrolladores individuales. Además de las imágenes que han sido producidas por Docker, Inc.
- **Daemon Docker:** El daemon Docker es un servicio que se ejecuta en el sistema operativo. Este servicio crea y gestiona las imágenes Docker.

### 15.10.5. Instalación Docker

Antes de comenzar con la instalación es necesario que se instalen unas herramientas:

- **apt-transport-https:** permite que el administrador de paquetes transfiera datos a través de https.
- **ca-certificates:** permite que el navegador web y el sistema verifiquen los certificados de seguridad.
- **curl:** transfiere datos.
- **software-properties-common:** agrega scripts para administrar el software.

```
1 #actualizar lista de paquetes
2 apt update
3 #instalar herramientas
4 apt install curl apt-transport-https ca-certificates software-properties-common
5
6
```

Código 15.11: Paquetes necesarios

Clave GPG:

```
1 #crear directorio
2 mkdir -p /etc/apt/keyrings
3 #descargar key
4 curl -fsSL https://download.docker.com/linux/debian/gpg | gpg --dearmor -o /etc/apt/
   keyrings/docker.gpg
5
```

Código 15.12: Clave GPG

Agrega el repositorio:

```
1
2 #agregar repositorio
3 echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
   https://download.docker.com/linux/debian $(lsb_release -cs) stable" | sudo tee /etc/
   apt/sources.list.d/docker.list > /dev/null
4
```

Código 15.13: Repositorio

Instalar Docker Engine.

```
1 #actualizar lista de paquetes
2 apt update
3 #instalar docker
4 apt install docker-ce docker-ce-cli containerd.io docker-compose-plugins
5
```

Código 15.14: docker

Para verificar que docker se esta ejecutando correctamente ejecutare el contenedor de Hello-World, el cual da una bienvenida.

```
1 #descargar y ejecutar contenedor de bienvenida
2 docker run hello-world
3
```

Código 15.15: Hello-World

### 15.10.6. ¿Que es Docker Compose?

Docker Compose es una herramienta que permite simplificar el uso de Docker a partir de un archivo YAML. De esta manera es mas sencillo crear contenedores, conectarlos, habilitar puertos, volúmenes, etc.

### 15.10.7. Instalación Docker Compose

Hay varias versiones de Docker Compose, aunque siempre es preferible la versión estable.

```

1  #descargar binario de docker-compose
2  curl -L "https://github.com/docker/compose/releases/download/1.26.0/docker-compose-$(
    uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
3  #cambiar permisos
4  chmod +x /usr/local/bin/docker-compose
5  #controlar la version de docker-compose
6  docker-compose --version
7

```

Código 15.16: Docker Compose

## 15.11. Contenedor NextCloud

### 15.11.1. ¿Que es Nextcloud?

Nextcloud es un servidor para el intercambio de archivos que permite almacenar contenido personal (documentos e imágenes, videos, etc) en una ubicación centralizada. Al ser sus funciones de código abierto, ofrece una gran versatilidad, seguridad y cumplimiento a las normas de privacidad. También permite aplicar diferentes políticas a los datos, como cifrado, administración de usuarios y auditoría.

### 15.11.2. Características:

- Los archivos Nextcloud son almacenados en estructuras de directorio convencionales y se pueden acceder vía WebDAV.
- Los archivos son cifrados en la transmisión y opcionalmente durante el almacenamiento.
- Los usuarios pueden manejar calendarios CalDAV, contactos CardDAV, tareas programadas y reproducir contenido multimedia Ampache.
- Permite la administración de usuarios y grupos de usuarios, vía OpenID o LDAP y definir permisos de acceso.
- Posibilidad de añadir aplicaciones de un solo clic y conexiones con Dropbox, Google Drive y Amazon S3.
- Disponibilidad de acceso a diferentes bases de datos, mediante SQLite, MariaDB, MySQL, Oracle Database, y PostgreSQL
- Cuenta con aplicaciones clientes para Windows, MAC OSX, Linux, Android y IOS.

### 15.11.3. Requisitos:

Al ejecutarse dentro de un contenedor todas sus dependencias se encuentran resueltas, solo se debera considerar la cantidad de memoria RAM.

La documentación de Nextcloud indica que la cantidad de memoria que se necesita para ejecutar un servidor Nextcloud es muy variable, y dependerá de la cantidad de usuarios, aplicaciones, archivos y volumen de actividad del servidor.

### 15.11.4. Preparación del área de trabajo

Mi directorio de trabajo sera `/srv` donde creare un directorio que almacenara toda las configuraciones de Nextcloud, así como la configuración de la base de datos.

```

1  #cambiar al directorio /srv
2  cd /srv
3  #crear directorio Nextcloud
4  mkdir nextcloud
5  #entrar al directorio Nextcloud
6  cd nextcloud
7

```

Código 15.17: Directorio de trabajo NextCloud

Dentro del directorio creare un documento llamado *docker-compose.yaml* y dentro de el colocare el siguiente código:

```

version: "3"
services:
  nextcloud:
    image: nextcloud
    container_name: nextcloud
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=America/Argentina/Buenos_Aires
    volumes:
      - ./nextcloud/config:/config
      - ./nextcloud/data:/data
    ports:
      - 8080:80
  mariadb:
    image: linuxserver/mariadb
    container_name: mariadb
    environment:
      - PUID=1000
      - PGID=1000
      - MYSQL_ROOT_PASSWORD=123456
      - TZ=America/Argentina/Buenos_Aires
      - MYSQL_DATABASE=nextcloud
      - MYSQL_USER=nextcloud
      - MYSQL_PASSWORD=123456
    volumes:
      - ./mariadb/config:/config
    ports:
      - 3306:3306
    restart: unless-stopped

```

### Estructura del docker-compose.yml

- **version:** Un archivo de docker-compose comienza especificando la versión de docker compose que se utilizará.
- **services:** Después de la versión viene anidada la sección de *services*. Puede haber tantos servicios como se necesiten, y cada servicio contará con sus propia estructura.
- **image:** La configuración *image* establece la imagen a partir de la cual se generará el servicio.
- **container\_name:** El nombre que tendrá el contenedor cuando se este ejecutando.
- **environment:** La configuración *environment* permite establecer una lista de variables de entorno que estarán disponibles en el servicio.
- **volumes:** Con volúmenes se puede compartir partes del sistema operativo con un servicio.
- **ports:** Los puertos que se expondrán al exterior y a cual puerto de la máquina se vincularán.
- **restart:** Con *restart* se especifica la política de reinicio a los servicios.



La opción `restart` puede tomar varios valores:

- **no:** nunca reinicia el contenedor.
- **always:** siempre lo reinicia.
- **on-failure:** lo reinicia si el contenedor devuelve un estado de error.
- **unless-stopped:** lo reinicia en todos los casos excepto cuando se detiene.

### 15.11.5. Crear Contenedor

Para crear el contenedor de Nextcloud es necesario que se utilice el siguiente comando, el cual descargara las imágenes necesarias y realizara las configuraciones especificadas dentro del archivo YAML.

```
1 docker-compose up -d
2
3
```

Código 15.18: Crear Contenedor Docker

#### Explicación del comando anterior:

- **docker-compose:** Define y ejecuta aplicaciones de varios contenedores con Docker.
- **up:** Crea e inicia el contenedor.
- **-d:** Ejecuta contenedores en segundo plano.

En caso que Docker deba descargar las imágenes, este proceso puede demorar varios minutos, dependiendo de la velocidad de la conexión al Internet con la que se cuente.

Para visualizar si los contenedores se están ejecutando correctamente Docker permite ver el estado de los contenedores con el siguiente comando:

```
1 docker ps -a
2
3
```

Código 15.19: Estado de los contenedores

### 15.11.6. Configuración de NextCloud

Una vez que el contenedor de Nextcloud se encuentre corriendo, desde el navegador web terminare de realizar las configuraciones:

- Crear usuario administrador.
- Especificar el directorio donde se guardan los datos que se almacenen en Nextcloud.
- Configurar la base de datos.

Figura 15.1: Configurar Nextcloud

### Nextcloud

Para acceder a NextCloud en el navegador se debe colocar la dirección IP del servidor Docker y el puerto donde escucha el contenedor. En mi caso sería <http://192.168.1.222:8080>

Finalizada la configuración y ya dentro del dashboard me dirigiré al perfil del usuario administrador, donde podre crear los usuarios que tendrán acceso al servicio de Nextcloud.

Figura 15.2: Creación de Usuarios

Los alumnos podrán hacer uso de este servicio solamente desde la red lan a través de un navegador web. Donde cada uno accederá con su usuario y contraseña.

### 15.11.7. Conclusión

Con Nextcloud la información se almacena de forma segura en un lugar controlable. NextCloud replicar las capacidades de servicios externos de almacenamiento en la nube, permitiendo compartir contenido entre usuarios de la misma red local.

## 15.12. Contenedor Emby

### 15.12.1. ¿Que es Emby?

Emby es un centro de multimedia multiplataforma que permite visualizar contenido desde cualquier tipo de dispositivo via Web, App o DLNA.

Emby trabaja con el contenido multimedia almacenado en servidor local, esto lo hace ideal dadas las características del Aula Virtual.

### 15.12.2. Características de Emby:

- El servidor de Emby convierte y envía automáticamente los vídeos a cualquier dispositivo.
- Cuenta con canales de vídeo en directo.
- Permite sincronizar librerías con los clientes móviles.
- La organización está muy cuidada para ser sencilla de ver y entender.
- Cuenta con controles parentales.
- Servidor DLNA para reproducir el contenido multimedia.
- Compatible con Chromecast.

### 15.12.3. ¿Qué es un centro multimedia?

Un centro multimedia es una aplicación que gestiona el contenido multimedia (fotografías, musica y vídeos), para que se pueda reproducir en el mismo dispositivo donde esta almacenado o en otros dispositivos, en este caso, conectados a la red LAN.

Emby, cuentas con la aplicación servidor que gestiona el contenido y lo transmite por streaming, y por otro lado la aplicación que recibe el contenido. La cual puede ser el cliente que se encuentra disponible para la plataforma de Windows, Mac, Linux, FreeBSD, dispositivos NAS, Android, iOS, Windows Phone, Android TV, Fire TV, Apple TV, Kodi, Xbox, etc. O cualquier navegador compatible con HTML5.

### 15.12.4. Preparación del área de trabajo

Mi directorio de trabajo sera /srv donde creare un directorio que almacenara toda las configuraciones de Emby.

```
1 #cambiar al directorio srv
2 cd /srv
3 #crear directorio
4 mkdir emby
5 #cambiar directorio
6 cd emby
7
```

Código 15.20: Directorio de trabajo Emby

Dentro del directorio creare un documento docker-compose.yaml el cual contendra el siguiente codigo:

```
version: "3.3"
services:
  emby:
    image: linuxserver/emby
    container_name: emby
    restart: unless-stopped
    ports:
      - "8096:8096"
    environment:
      - PUID=998
      - PGID=1000
      - TZ=America/Argentina/Buenos_Aires
    volumes:
      - ./library:/config
      - ./series:/data/tvshows
      - ./peliculas:/data/movies
      - ./musica:/data/music
      - ./tutoriales:/data/tutorials
      - ./imagenes:/data/images
```

#### Estructura del documento

Descripción del código 15.11.4

### 15.12.5. Crear Contenedor Docker:

```
1
2  docker-compose up -d
3
```

Código 15.21: Crear contenedor Docker

#### Explicación del comando anterior:

- **docker-compose:** Define y ejecute aplicaciones de varios contenedores con Docker.
- **up:** Crea e inicia el contenedor.
- **-d:** Ejecuta contenedores en segundo plano.

Este proceso puede demorar varios minutos, dependiendo de la velocidad de la conexión al Internet con la que se cuente. Una vez finalizada la descarga de la imagen se ejecutara el contenedor.

### 15.12.6. Configuración Emby

Una vez que el contenedor se encuentra en ejecución me dirigiré al navegador web para terminar de realizar las configuraciones. Al igual que en NextCloud, para acceder al servicio de Emby solo se debe colocar en el navegador web la dirección IP del servidor Docker y el puerto donde escucha. En mi caso seria <http://192.168.1.222:8096>

- **Idioma:** Seleccionar el lenguaje de la Interfaz Emby.
- **Crear usuario y contraseña:** Creación del usuario para acceder a la interfaz web de Emby.
- **Agregar Contenido Multimedia:** Seleccionar los directorios donde se encuentran todo el contenido multimedia que se compartirá con la red lan.
- **Lenguaje de las Bibliotecas:** Seleccionar el Lenguaje con el que se mostrara la información del contenido multimedia en la plataforma web.
- **Términos y Condiciones:** Aceptar los términos y condiciones de Emby.

- **Final de Configuración:** Ahora Emby se encargara de realizar las demás configuraciones, mientras ofrece los clientes de todas las plataformas soportadas.
- **Login:** Ingresar al sistema web.

### Panel de control de Emby

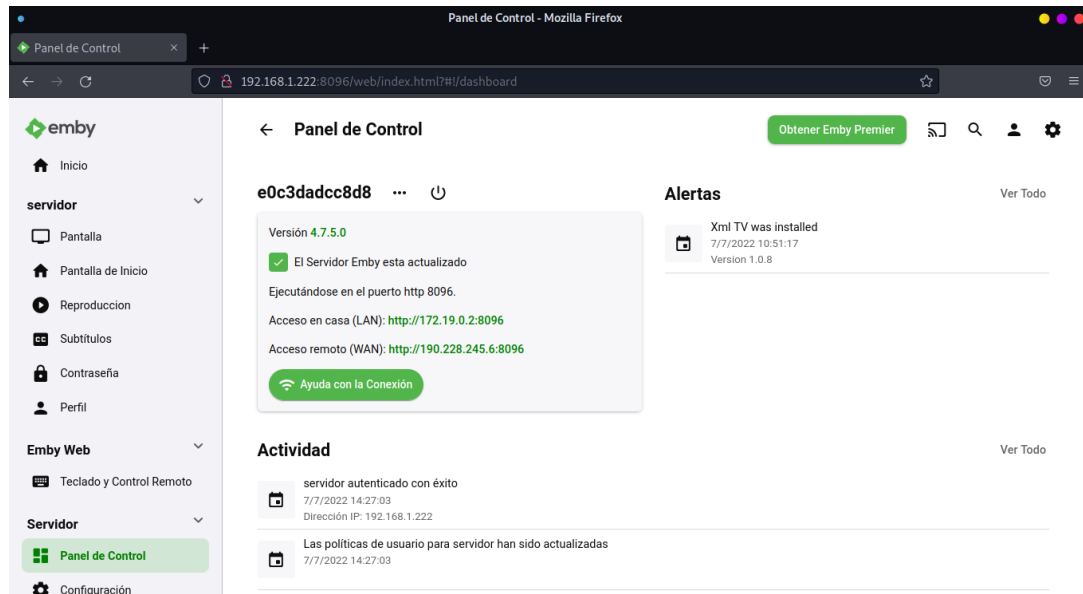


Figura 15.3: Dashboard

### 15.12.7. Conclusión

Uno de los problemas que tenemos en el Aula virtual es la velocidad de conexión al Internet (2MB). Lo que dificulta en gran manera la reproducción de Vídeos Tutoriales o contenido multimedia recomendado por los docentes.

Ahora se descarga el contenido multimedia y se almacena de forma segura en un lugar controlable dentro del Servidor Local y con Emby por medio de streaming se comparte con toda la red lan.

## Capítulo 16

# Conclusión

En este plan de migración he considerado los programas privativos que se utilizan actualmente y sus respectivas alternativas en software libre. Según los resultados obtenidos a partir de la investigación y la búsqueda de alternativas libres, se puede concluir que la migración a software libre es factible si los alumnos están de acuerdo.

Cabe destacar que uno de los factores más importantes en la búsqueda de información fue la comunidad de Software Libre. La cual está conformada en su mayoría por usuarios finales. Estos mismos usuarios me sirvieron de guía y ayuda para enfocarme en la elaboración de una estrategia que me permita afrontar el problema propuesto. Con la ayuda de sus recomendaciones concluí que se requiere una capacitación y formación adecuada para trabajar en el nuevo entorno de trabajo. En mi caso, como persona a cargo de la migración e implementación debo asumir el tiempo asociados al entrenamiento, capacitación, formación y soporte a los alumnos.

Como trabajo futuro se propone la ampliación de este proyecto, la generación de una política de seguridad, administración centralizada y una planificación más detallada del proceso de migración.

Como dice Stallman, es importante que en escuelas y universidades se utilice software libre, ya que estas instituciones educativas están decidiendo el futuro de la sociedad. Por lo tanto, no se debe aceptar que en un espacio perteneciente a la universidad se utilice y enseñe a utilizar software privativo, sabiendo que el deber de una universidad es la creación y difusión del conocimiento. Los alumnos como parte de ella, también deben defender la libertad de las personas para compartir el conocimiento y el software.

«Las escuelas deben enseñar a sus alumnos a ser ciudadanos de una sociedad fuerte, capaz,  
independiente y libre».

The Free Software Foundation.

# Bibliografía

- [1] By Santiago Becerra Carrillo. Lista de licencias con comentarios. January.
- [2] Free Software Foundation. Historia de gnu. December.
- [3] Free Software Foundation. Why Educational Institutions Should Use and Teach Free Software. page 1, June.
- [4] Nextcloud GmbH. Nextcloud Server Administration Manual. pages 1–351, Jul 09.
- [5] By William Henry Gates III. An Open Letter to Hobbyists. page 1, February.
- [6] Bailón Giler Cristina Lizeth, Delgado Castro Nexar Javier, Resabala Córdova Mayra Alejandra, and Resabala Córdova Olga Isabel. Instalación y Configuración de Equipos Informáticos bajo Software Libre para la Biblioteca de la Facultad de Ciencias Informáticas de la Universidad Técnica de Manabí.
- [7] Emby LLC. Emby Server for Docker. 3 days ago.
- [8] Alejandro Lopez. Administración GNU/Linux 2. page x.
- [9] Debian org . Guía Debian GNU/Linux de instalación. pages 1–x, January 29.
- [10] By Fernando Ribeiro. Cómo Instalar un Servidor Linux con Debian 10 'Buster'. pages 1–7, April.
- [11] By Richard Stallman. ¿qué es el software libre? June.
- [12] Celeste Weidman. Redes. page x.