



# **UNIVERSIDAD NACIONAL DE LOJA**

**ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS  
RECURSOS NATURALES NO RENOVABLES**

## **CARRERA DE INGENIERÍA EN SISTEMAS**

**CREACIÓN DEL REPOSITORIO DE PAQUETES Y  
ACTUALIZACIÓN DEL SISTEMA OPERATIVO UNIX**

**TESIS PREVIA A LA OBTENCIÓN  
DEL TÍTULO DE INGENIERO EN  
SISTEMAS.**

**AUTORES:** César Augusto Silverio Mendieta.  
Víctor Ignacio Cueva Cango.

**DIRECTOR:** Ing. Edison Leonardo Coronel Romero

**LOJA-ECUADOR**

**2011**



## PENSAMIENTOS

“El software libre construye una sociedad mejor”

**Richard M. Stallman**

"El movimiento ético del software libre puede vencer a cualquiera interesado en su libertad"

**Richard M. Salman**

“Hombre concóete a tu mismo y conocerás las leyes del Universo y a los Dioses”

**Platón**

"No hay que confundir nunca el conocimiento con la sabiduría. El primero nos sirve para ganarnos la vida; la sabiduría nos ayuda a vivir".

**Sorcha Carey**



## **CERTIFICACIÓN**

Ing.

Edison Leonardo Coronel Romero

DOCENTE DE LA UNIVERSIDAD NACIONAL DE LOJA, ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y DE LOS RECURSOS NATURALES NO RENOVABLES Y DIRECTOR DE TESIS

### **CERTIFICO:**

Que en calidad de Director del Proyecto de Tesis denominado “**CREACIÓN DEL REPOSITORIO DE PAQUETES Y ACTUALIZACIÓN DEL SISTEMA OPERATIVO UNLIX**” he dado la asesoría requerida a los egresados: Víctor Ignacio Cueva Cango y César Augusto Silverio Mendieta, durante el proceso de desarrollo de esta tesis previa a la obtención del Grado de Ingenieros en Sistemas, la cual cumple con los requisitos para ser aceptada por el Tribunal de grado correspondiente, por esta razón autorizo su presentación.

Loja, junio de 2011

Ing. Edison Leonardo Coronel Romero

**DIRECTOR DE TESIS**



## **DECLARACIÓN DE AUTORÍA**

Nosotros, César Augusto Silverio Mendieta y Víctor Ignacio Cueva Cango, autores de este trabajo de tesis, certificamos la propiedad intelectual a favor de la Universidad Nacional de Loja, las cuales podrán hacer uso del mismo con la finalidad académica que estimen conveniente.

.....  
César Augusto Silverio Mendieta

.....  
Víctor Ignacio Cueva Cango



## **AGRADECIMIENTO**

De manera muy especial a la gran Universidad la de la Vida, dirigida por el creador de todo este sistema Dios, a él porque el día a día fue dándonos las fuerzas necesarias para seguir adelante, especialmente cuando nos encontrábamos en duros pasajes de esta existencia.

A la Universidad Nacional de Loja, al Área de la Energía, las Industrias y de los Recursos Naturales no Renovables, a la carrera de Ingeniería en Sistemas, por permitir terminar con éxito nuestros estudios. A los docentes y personal administrativos por su ayuda, para la culminación del presente trabajo.

También de manera muy particular al Ing. Edison Leonardo Coronel por su tiempo dedicado en el asesoramiento del presente proyecto, a todas las personas que brindaron un criterio del presente trabajo amigos y compañeros que estuvieron siempre con nosotros en diálogos relacionados con el tema, además a los que no son amigos porque gracias a ellos pudimos comprender el equilibrio de algunas cosas.

A nuestros familiares y amigos que colaboraron de forma directa e indirecta para la finalización del presente proyecto.

Los Autores.



## DEDICATORIA

De todo corazón le dedico este trabajo a mis padres José Floresmilo Cueva y Luzbenia Esperanza Cango Cuenca, porque ellos han sabido forjar en mi la personalidad y las ganas de seguir adelante y ser un hombre de bien, por lo que he llegado a culminar este proyecto con la mayor satisfacción del mundo, de igual forma le dedico esto a mis hermanos quienes han sido el soporte en todo sentido y han logrado formar en mí sentimientos buenos y valaderos para la vida cotidiana, además también va dedicado a todas esas personas que creen en mí y están convencidos que no les he defraudado y a aquellos que sin conocerme hacen uso de este documento porque les es útil.

**Víctor Cueva.**

A la vida en sí representada por Jehová, a todas las personas con deseo de superación y cambio, aquellos que ven a la misma en forma desinteresada, de manera especial a mi madre Margarita que día a día estuvo conmigo, a mi padre, hermanos, a mis tíos que siempre estuvieron pendientes de mí, en especial a Eduardo y Patricio, a mis amigos y amigas sinceros que verdaderamente están en los momentos difíciles y a todos los amigos invisibles que de corazón apoyaron a la realización de este proyecto.

**César Silverio.**



## 1. RESUMEN

El propósito del presente proyecto es de elaborar un repositorio de paquetes y actualización del sistema operativo UNLIX, el cual que es un sistema que cubre la filosofía libre y es orientado inicialmente para el Área de la Energía, las Industrias y de los Recursos Naturales no Renovables, pero la perspectiva es que sea, un sistema operativo para la Universidad Nacional de Loja en su versión UNLIX 2.0 GNU/LINUX.

En la fase de introducción se hace referencia a los conceptos teóricos básicos que dan soporte a esta investigación, posteriormente se realiza un estudio preliminar y un análisis del sistema operativo UNLIX en su primera versión, ahí se observó la investigación realizada por Maritza Quezada y Roberto Mora, quienes hicieron un estudio de los requerimientos para los docentes, estudiantes y administrativos del Área de la Energía de la Universidad Nacional de Loja.

Luego se inicia con el desarrollo del proyecto, se crea un ambiente propicio para almacenar el repositorio de paquetes de UNLIX 2.0 en un servidor ubicado en la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja, dicho servidor es el encargado de alojar los paquetes.

Posteriormente, se crea la distribución UNLIX 2.0 GNU/LINUX, el mismo que está basado en la última versión estable de Debian, denominada Squeeze, se instalan los programas de UNLIX 2.0, tomando como referencia el estudio de la primera versión de UNLIX, se la personaliza a la misma y se crea el Live DVD y el Live Usb de UNLIX 2.0. Además se crea un repositorio de paquetes primario en un DVD.

Finalmente se hacen las pruebas del producto final con algunos estudiantes y funcionarios de la Universidad Nacional de Loja, se realizan las conclusiones y recomendaciones del proyecto.



## **2. SUMMARY**

The purpose of this project is to develop a repository of packages and UNLIX operating system update, which is a system that covers the philosophy is free and primarily oriented to the area of Energy, Industry and Non-Renewable Natural Resources but the prospect is that it is, an operating system for the National University of Loja in the version 2.0 UNLIX GNU / Linux.

In the introductory phase refers to the basic theoretical concepts that support this research, then performed a preliminary study and analysis of operating system UNLIX in its first version, there was observed the investigation conducted by Maritza Quezada and Roberto Mora who made a study of the requirements for teachers, students and administrative area of Energy, National University of Loja.

Then it starts with the development of the project, creates an environment to store the package repository UNLIX 2.0 on a server located in the direction of Computing National University of Loja, the server is in charge of hosting packages.

Subsequently, the distribution is created UNLIX 2.0 GNU / Linux, it is based on the latest stable version of Debian, called Squeeze, programs are installed UNLIX 2.0, with reference to the study of the first version of UNLIX, it personalize it and creating the Live DVD and Live Usb UNLIX 2.0. You can create a primary package repository on DVD.

Finally, it makes the end product testing with students and staff of the National University of Loja, made the findings and recommendations of the project.



### 3. ÍNDICE DE CONTENIDOS

<b>CARATULA</b> .....	I
<b>PENSAMIENTOS</b> .....	II
<b>CERTIFICACIÓN</b> .....	III
<b>DECLARACIÓN DE AUTORÍA</b> .....	IV
<b>AGRADECIMIENTO</b> .....	V
<b>DEDICATORIA</b> .....	VI
<b>1. RESUMEN</b> .....	VII
<b>2. SUMMARY</b> .....	VIII
<b>3. ÍNDICE DE CONTENIDOS</b> .....	IX
3.1 ÍNDICE DE FIGURAS.....	XII
3.2 ÍNDICE DE TABLAS.....	XIV
<b>4. INTRODUCCIÓN</b> .....	16
<b>5. METODOLOGÍA</b> .....	18
<b>6. FUNDAMENTACIÓN TEÓRICA</b> .....	20
<b>6.1 PROYECTO GNU/LINUX</b> .....	20
6.1.1 Características Linux y el Proyecto GNU.....	21
6.1.2 Núcleo de Linux como proyecto de software libre.....	23
<b>6.2 SOFTWARE LIBRE (FREE SOFTWARE)</b> .....	23
6.2.1 Características del Software libre.....	24
6.2.2 Ventaja de la Utilización de software libre.....	25
6.2.3 Licencias en el Software Libre.....	26
<b>6.3 DEBIAN</b> .....	31
6.3.1 El proyecto Debian.....	32
6.3.2 Debían como distribución.....	33
<b>6.4 LOS REPOSITORIOS</b> .....	35
6.4.1 ¿Qué son los paquetes?.....	35
6.4.2 ¿Qué es un repositorio de paquetes?.....	36
6.4.3 Características y funcionamiento de un repositorio.....	37
6.4.4 Alcance de un repositorio.....	39



6.4.5 Aspectos necesarios para elaborar repositorio.....	40
6.4.6 El repositorio de Debían.....	40
6.4.7 apt get como herramienta de administración de paquetes.....	41
6.4.8 Synaptic como gestor de Paquetes.....	42
6.4.9 DVD como medio para actualizaciones de paquetes.....	43
6.4.10 Los scripts.....	43
<b>6.5 LAS DISTRIBUCIONES.....</b>	<b>44</b>
6.5.1 ¿Que es una Distro o Distribución?.....	44
6.5.2 Las Metadistros.....	46
6.5.3 Análisis para la creación de una distribución.....	47
6.5.4 Herramientas para la creación de una distro.....	47
6.5.5 El Sistema objetivo.....	50
<b>7. EVALUACIÓN DEL OBJETO DE INVESTIGACIÓN.....</b>	<b>51</b>
<b>8. DESARROLLO DE LA PROPUESTA ALTERNATIVA.....</b>	<b>54</b>
<b>8.1 NOCIONES GENERALES SOBRE UNLIX.....</b>	<b>54</b>
8.1.1 ¿Que es UNLIX? .....	54
8.1.2 Características de UNLIX.....	54
8.1.3 Proyección de UNLIX.....	55
8.1.4 Manejo Básico de UNLIX.....	56
8.1.5 Los paquetes de UNLIX.....	57
<b>8.2 CREACIÓN DEL REPOSITORIO.....</b>	<b>59</b>
8.2.1 Requerimientos del servidor para el repositorio.....	62
8.2.2 Configuración Previa a la Creación del Repositorio.....	65
8.2.3 Configuración de los Scripts para el repositorio.....	70
8.2.4 Ejecución de los scripts y Creación del repositorio.....	74
8.2.5 Actualización automática del repositorio.....	79
<b>8.3 ACTUALIZACIÓN DE UNLIX 2.0.....</b>	<b>82</b>
8.3.1 Partiendo de la versión de Debian.....	85
8.3.2 Configuración básica de la versión de UNLIX 2.0.....	90
8.3.3 Instalación y actualización de las aplicaciones para UNLIX 2.0.....	91
8.3.4 Netbeans en UNLIX 2.0.....	98



8.3.5 Otros paquetes para UNLIX 2.0.....	99
8.3.6 Personalizar la Distribución objetivo UNLIX 2.0.....	100
8.3.7 Experiencias en la elaboración del sistema objetivo.....	101
<b>8.4 CREACIÓN DE LA DISTRIBUCIÓN UNLIX 2.0 CON REMASTERSYS</b> .....	<b>102</b>
8.4.1 Módulos necesarios antes de la instalación de Reamastersys.....	103
8.4.2 Instalación de la aplicación de Remastersys.....	105
8.4.3 Trabajando con Remastersys.....	105
8.4.4 Generación de la imagen en un DVD UNLIX 2.0.....	106
8.4.5 Generación de la imagen en un USB UNLIX 2.0.....	109
8.4.6 Experiencias del a creación de la imagen.....	111
<b>8.5 CREACIÓN DE UN CD/DVD PARA ACTUALIZAR PAQUETES</b> .....	<b>113</b>
<b>8.6 PLAN DE VALIDACIÓN</b> .....	<b>118</b>
8.6.1 Pruebas con Usuarios.....	118
8.6.2 Análisis de los resultados de la pruebas.....	120
8.6.3 Pruebas de Inicio.....	131
8.6.4 Pruebas de hardware y Software .....	131
8.6.5 Pruebas con el Repositorio.....	132
<b>9. VALORACIÓN TÉCNICA ECONÓMICA</b> .....	<b>135</b>
<b>10. CONCLUSIONES</b> .....	<b>138</b>
<b>11. RECOMENDACIONES</b> .....	<b>140</b>
<b>12. BIBLIOGRAFÍA Y DEFENECÍAS</b> .....	<b>142</b>
<b>13. ANEXOS</b> .....	<b>144</b>
ANEXO A: Glosario de Términos.....	145
ANEXO B: Modelo de Encuesta Realizadas en las Pruebas.....	154
ANEXO C: Encuestas Realizadas.....	156
ANEXO D: Certificación de los desarrolladores de UNLIX (Primera versión).....	164
ANEXO E: Licencia de UNLIX 2.0 GNU/LINUX.....	165
ANEXO F: Ley Orgánica de Educación Superior.....	168
ANEXO G: Certificación de la Unidad de Telecomunicaciones e Información.....	169
ANEXO H: Certificaciones de Conferencia de UNLIX 2.0 GNU/LINUX.....	170
ANEXO I: Anteproyecto de Tesis.....	171



### 3.1 ÍNDICE DE FIGURAS

Figura 6.1 Logo GNU.....	20
Figura 6.2 Logo Debian.....	31
Figura 6.3 Gestor Synaptic.....	41
Figura 6.4 Logo Herramienta Remastersys.....	49
Figura 8.1 Netbeans en UNLIX 1.0.....	59
Figura 8.2 Servidor Apache.....	66
Figura 8.3 Servicio Apache.....	67
Figura 8.4 Buscar el paquete mirror .....	67
Figura 8.5 Instalación Debian Mirror.....	68
Figura 8.6 Servidor Apache con el directorio Unix2.....	69
Figura 8.7 Descargando del repositorio Oficial.....	75
Figura 8.8 Descargando del repositorio Backports.....	75
Figura 8.9 Descargando del repositorio Multimedia.....	76
Figura 8.10 Descargando del repositorio Security.....	76
Figura 8.11 Verificando la descarga del repositorio.....	77
Figura 8.12 Estructura de repositorio unix2.....	78
Figura 8.13 Directorio base del repositorio.....	78
Figura 8.14 Creación de los paquetes del repositorio.....	79
Figura 8.15 Crontab de los scripts.....	81
Figura 8.16 Pantalla verificación del kernel de debian.....	90
Figura 8.17 Pantalla administrador de programas de debian.....	92
Figura 8.18 Netbeans en Unix 2.0.....	98
Figura 8.19 Escritorio Unix 2.0.....	101
Figura 8.20 Pantalla de synaptic.....	104
Figura 8.21 Pantalla de configuración de remastersys.....	106
Figura 8.22 Pantalla de remastersys.....	107
Figura 8.23 Pantalla trabajando con remastersys.....	107
Figura 8.24 Pantalla creando la iso con remastersys.....	108
Figura 8.25 Remstersys usb.....	109
Figura 8.26 Selección de la memoria usb.....	110



---

Figura 8.27 Selección de la iso Unix 2.0.....	110
Figura 8.28 Exposición de la Distro en el “Comil”.....	112
Figura 8.29 Inicio AptonCd.....	115
Figura 8.30 Paquetes AptonCd.....	116
Figura 8.31 Creación del CD/DVD con AptonCd.....	117
Figura 8.32 Grabación del CD/DVD con AptonCd.....	117
Figura 8.33 Tiempo de respuesta del sistema.....	121
Figura 8.34 Levantamiento de Unix 2.0.....	122
Figura 8.35 Accesibilidad de Unix 2.0.....	123
Figura 8.36 Personalización de Unix 2.0.....	124
Figura 8.37 Trabajar con Unix 2.0.....	125
Figura 8.38 Conocimientos de Unix 2.0.....	126
Figura 8.39 Calificación de Unix 2.0.....	127
Figura 8.40 Confiabilidad de Unix 2.0.....	128
Figura 8.41 Utilización de Unix 2.0.....	129
Figura 8.42 Expectativa de Unix 2.0.....	130
Figura 8.43 Pantalla inicial para instalar UNLIX 2.0.....	131
Figura 8.44 Trabajando con UNLIX 2.0.....	132
Figura 8.45 Actualización con UNLIX 2.0.....	133
Figura 8.46 Configurando Programas en UNLIX 2.0.....	134



### 3.2 ÍNDICE DE TABLAS

Tabla 8.1 Paquetes Unix 1.0 Multimedia.....	57
Tabla 8.2 Paquetes Unix 1.0 Gráficos.....	57
Tabla 8.3 Paquetes Unix 1.0 Ofinina.....	57
Tabla 8.4 Paquetes Unix 1.0 Desarrollo.....	58
Tabla 8.5 Paquetes Unix 1.0 Diseño.....	58
Tabla 8.6 Paquetes Unix 1.0 Internet.....	58
Tabla 8.7 Características del Servidor Unix 2.0.....	63
Tabla 8.8 Script Repositorio Oficial.....	72
Tabla 8.9 Script Repositorio Security.....	72
Tabla 8.10 Script Repositorio Backports.....	73
Tabla 8.11 Script Repositorio Multimedia.....	73
Tabla 8.12 Característica del crontab.....	80
Tabla 8.13 Programación del crontab.....	81
Tabla 8.14 Comparación en paquetes Unix 1.0 y Unix 2.0.....	83
Tabla 8.15 sources.list Unix 2.0.....	87
Tabla 8.16 sources.list Repositorio Oficial.....	88
Tabla 8.17 Paquetes Multimedia Unix 2.0.....	93
Tabla 8.18 Paquetes Ofimática Unix 2.0.....	93
Tabla 8.19 Paquetes Electrónica Unix 2.0.....	94
Tabla 8.20 Paquetes Internet Unix 2.0.....	94
Tabla 8.21 Paquetes Desarrollo Unix 2.0.....	95
Tabla 8.22 Otros Paquetes Unix 2.0.....	96
Tabla 8.23 Paquetes Diseño Unix 2.0.....	97
Tabla 8.24 Repositorio Remastersys.....	103
Tabla 8.25 Tiempo de respuesta del sistema.....	121
Tabla 8.26 Levantamiento de Unix 2.0.....	122
Tabla 8.27 Accesibilidad de Unix 2.0.....	123
Tabla 8.28 Personalización de Unix 2.0.....	124
Tabla 8.29 Trabajar con Unix 2.0.....	125
Tabla 8.30 Conocimientos de Unix 2.0.....	126



---

Tabla 8.31 Calificación de Unix 2.0.....	127
Tabla 8.32 Confiabilidad de Unix 2.0.....	128
Tabla 8.33 Utilización de Unix 2.0.....	129
Tabla 8.34 Expectativa de Unix 2.0.....	130
Tabla 9.1 Valoración Económica Recursos Humanos.....	135
Tabla 9.2 Valoración Económica Recursos Materiales.....	135
Tabla 9.3 Valoración Económica Recursos Hardware.....	136
Tabla 9.4 Valoración Económica Recursos Software.....	136
Tabla 9.5 Valoración Económica Recursos Comunicaciones.....	136
Tabla 9.6 Valoración Económica Recursos Técnicos y Tecnológicos.....	137
Tabla 9.7 Aproximación del Costo Real del Proyecto.....	137



## 4. INTRODUCCIÓN

La carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja, ha visto la posibilidad de darle impulso al proyecto UNLIX, sistema operativo GNU/LINUX para el Área de la Energía, las Industrias y de los Recursos Naturales no Renovables y crear su actualización, añadiendo un repositorio de paquetes propio para las actualizaciones futuras.

UNLIX 2.0 GNU/LINUX, recoge las experiencias del estudio realizado por los investigadores Maritza Quezada y Roberto Mora, posee la mayoría de paquetes de su primera versión, además contiene otros paquetes necesarios para cubrir los requerimientos de las diferentes carreras de la Universidad Nacional de Loja, estos paquetes se los ha puesto gracias a la asesoría de profesionales en informática y estudiantes de la Universidad Nacional de Loja.

UNLIX 2.0 GNU/LINUX trabaja en modo Live DVD y Live USB, es decir DVD o dispositivo USB que arranca el sistema sin necesidad de instalarlo en un computador, además UNLIX 2.0 GNU/LINUX se lo puede instalar en cualquier computador siempre y cuando el usuario lo requiera.

El objetivo fundamental de UNLIX 2.0 GNU/LINUX, es de dar una ayuda a los funcionarios, docentes y estudiantes de la Universidad Nacional de Loja, que están inmersos en el software libre, y/o aquellos que están empezando a trabajar con software libre, incentivados por la actual Ley de Educación Superior (ver anexo F) que se refiere al uso del software libre como prioridad en las Universidades Públicas.

Una vez instalado UNLIX 2.0 GNU/LINUX, desde cualquier punto de red de la Universidad Nacional de Loja, se lo puede actualizar, se puede agregar o quitar paquetes, debido a que este sistema está conectado a un servidor de paquetes dentro del campus de la Universidad.



UNLIX 2.0 GNU/LINUX nació como proyecto de tesis, así mismo fue mejorado y actualizado en su segunda versión, es multiplataforma y sería una buena alternativa para que la Universidad lo pueda acoger como su sistema oficial, así como el caso de CANAIMA GNU/LINUX que es un sistema usado por las entidades públicas y para proyectos educativos de Venezuela, <http://canaima.softwarelibre.gob.ve/>. Se sabe que la Universidad actualmente está trabajando con software libre, por lo tanto es posible que UNLIX 2.0 GNU/LINUX sea un sistema que a la postre vaya siendo mejorado y adaptado a nuevas necesidades universitarias y del público en general.



## 5. METODOLOGÍA

En el presente proyecto de investigación “**CREACIÓN DEL REPOSITORIO DE PAQUETES Y ACTUALIZACIÓN DEL SISTEMA OPERATIVO UNLIX**”, se utilizó el método experimental basado en la observación, el análisis y la investigación de elementos que conlleven a solucionar los problemas, este método se acopla al que se usa para el desarrollo de sistemas como son el análisis, diseño y la implementación del sistema

Las Fases que involucra el desarrollo del presente proyecto:

La fase uno se realizó un estudio preliminar del sistema operativo UNLIX en su primera versión, se hizo una entrevista con los Ingenieros Maritza Quezada y Roberto Mora creador del proyecto UNLIX para constancia de que se realizó la entrevista ver Anexo D del presente documento, se analizó la motivación que llevo a concretar el proyecto UNLIX, así como el estudio de los paquetes base e indispensables del sistema operativo UNLIX, también las herramientas que utilizó para la creación de la distribución, dicho estudio consta el proyecto de tesis cuyo tema versa “**DESARROLLO DE UNA DISTRIBUCIÓN LINUX BASADA EN DEBIAN PARA EL ÁREA DE ENERGÍA, LAS INDUSTRIAS Y RECURSOS NATURALES NO RENOVABLES DE LA UNL**” realizada por Maritza Quezada y Roberto Mora.

La fase dos se hizo el diseño del prototipo de UNLIX 2.0 GNU/LINUX, con la información y base de la fase uno, se procedió a establecer que paquetes se debería quitar o agregar en UNLIX 2.0 GNU/LINUX, esto se concretó tomando como referencia otros sistemas operativos GNU/LINUX que están siendo usados en la actualidad y también se buscó información de paquetes de software libre para otras carreras (medicina, agrícola, contabilidad banca y finanzas entre otras) y estos sean agregados a UNLIX 2.0 GNU/LINUX.



La fase tres con la implementación, se empieza con la creación del repositorio de paquetes, esto es muy importante ya este repositorio es la base de los paquetes de UNLIX 2.0 GNU/LINUX, de ahí se procedió con la instalación de los paquetes de software en la distribución final de UNLIX 2.0 GNU/LINUX y se realiza la personalización de la distribución.

Posterior a esta etapa se procede con la creación de la distribución, esta se la realiza una vez que se tiene todos los programas base de UNLIX 2.0 GNU/LINUX.

Dentro de la fase tres, también se consideran las pruebas, las mismas que se las realizó con personal docente, administrativo y estudiantes de la Universidad Nacional de Loja, además se probó la distribución en el evento del festival latinoamericano de software libre FLISOL, que se realizó el día sábado 9 de abril de 2011 en el hall del municipio de la ciudad de Loja. Cabe señalar que UNLIX 2.0 GNU/LINUX, también fue expuesto en el ciclo de conferencias de software libre 2011 de la carrera de Informática Educativa de la Universidad Nacional de Loja.

Dentro de las pruebas que se realizó se recogieron sugerencias de los encuestados las mismas que fueron aceptadas y adaptadas en UNLIX 2.0 GNU/LINUX, además se creó un blog [<http://unix.wordpress.com>] para dar soporte a los usuarios de UNLIX 2.0 GNU/LINUX, ya que la visión de este sistemas es para que sea usado no solo dentro del campus de la Universidad Nacional de Loja, sino también que lo usen todas las personas en general.



## 6. FUNDAMENTACIÓN TEÓRICA

### 6.1 PROYECTO GNU/LINUX

El proyecto GNU fue iniciado por Richard Stallman con el objetivo de crear un sistema operativo completamente libre: el sistema GNU.

GNU es un acrónimo recursivo que significa GNU No es Unix (GNU is Not Unix). Puesto que en inglés "gnu" (en español "ñu") se pronuncia igual que "new", Richard Stallman recomienda pronunciarlo "guh-noo". Lo que el fundador recomienda es pronunciarlo como uno quiera pero no decir que este significado es UNIX.

El 27 de septiembre de 1983 se anunció públicamente el proyecto por primera vez, esta idea de Stallman fue seguida por muchos usuarios a nivel del mundo, y en la actualidad el proyecto sigue creciendo día a día.

“**GNU/Linux** es uno de los términos empleados para referirse a la combinación del núcleo o kernel libre similar a Unix denominado **Linux**, que es usado con herramientas de sistema GNU. Su desarrollo es uno de los ejemplos más prominentes de software libre; todo su código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la GPL (**L**icencia **P**ública **G**eneral de GNU) y otra serie de licencias libres”<sup>1</sup>



Figura 6.1 Logo GNU

---

<sup>1</sup> WIKIPEDIA, Enciclopedia Libre. 2010. Filosofía GNU/LINUX. [en línea], [http://es.wikipedia.org/wiki/GNU/Linux], [Consulta: 2 agosto 2010].



### 6.1.1 Características Linux y el Proyecto GNU.

El proyecto GNU (GNU is Not Unix), nace por la década de los 80, fundado por Richard Stallman, con el objetivo de desarrollar un sistema operativo parecido al UNIX, pero con la característica de que maneje la filosofía del software libre, ya que Unix tenía un costo en aquel entonces, apoyado de muchos voluntarios están intentando crear un sistema operativo libre, recreando todos los componentes necesarios para tener un sistema operativo funcional que se convertiría en el sistema operativo GNU. En el comienzo de los años 1990, después de algunos años, GNU tenía muchas herramientas importantes, como compiladores, depuradores, intérpretes de órdenes etc., excepto por el componente central: el núcleo.

Más adelante aparece un estudiante llamado Linus Torvalds en la Universidad de Helsinki, en Finlandia, desarrolla un kernel similar al de Unix, y liberando la versión 0.11, entonces se completa el proyecto de Richard Stallman, es decir se ha creado, el sistema operativo completo con el nombre de GNU/LINUX.

Algo que hace reflexionar que gracias al compañerismo, de muchas personas y a la filosofía de libertad, hoy por hoy se tiene un sistema operativo libre, estable, incluso elegante, el mismo que sirve como objeto de estudio a los desarrolladores.

*“Linux es un sistema operativo, la gran pieza de software que maneja un computador. Es similar a Microsoft Windows, pero es completamente libre. El nombre correcto es GNU/Linux, pero Linux también se admite.*

*Linux no es el producto de una sola compañía, es el resultado de la contribución de un gran número de compañías y grupos de personas. De hecho, el sistema GNU/Linux es un componente central, el cual se transforma en muchos productos diferentes: las llamadas distribuciones.”<sup>2</sup>*

---

<sup>2</sup> LINUX, Soporte Linux. 2010. Linux Básico. [en línea], [<http://www.obtengalinux.org/linux/>], [Consulta: 15 agosto 2010].



Richard Salmen manifiesta, *“Los programadores generalmente saben que Linux es un núcleo. Pero como generalmente han oído llamar también «Linux» a todo el sistema, frecuentemente se imaginan una historia que justifique que todo el sistema se llame como el núcleo. Por ejemplo, muchos creen que una vez que Linus Torvalds terminó de escribir Linux, el núcleo, sus usuarios buscaron más software libre que lo acompañara, y encontraron que (sin ninguna razón en particular casi todo lo necesario para hacer un sistema de tipo Unix estaba ya disponible.”*<sup>3</sup>

Las características de GNU/LINUX son las siguientes:

- Multitarea, es posible ejecutar varios programas a la vez sin necesidad de tener que parar la ejecución de cada aplicación.
- Multiusuario, varios usuarios pueden acceder a las aplicaciones y recursos del sistema Linux al mismo tiempo. Y por supuesto, cada uno de ellos puede ejecutar varios programas a la vez.
- Shells programables, un shell conecta las ordenes de un usuario con el kernel y al ser programables se puede modificar para adaptarlo a las necesidades.
- Independencia de dispositivos Linux posee una gran adaptabilidad y no se encuentra limitado como otros sistemas operativos.
- Comunicaciones Linux es el sistema más flexible para poder conectarse a cualquier ordenador del mundo.
- Lo principal que **es libre** y cumple con las cuatro libertades manifestadas por la comunidad.
- El ambiente gráfico que las últimas distribuciones traen.

---

<sup>3</sup> Linux y el Proyecto GNU. 2010. GNU/LINUX. [en línea], [<http://www.gnu.org/gnu/linux-and-gnu.es.html>], [Consulta: 7 de agosto 2010]



### 6.1.2 Núcleo de Linux como proyecto de software libre

El núcleo Linux es un sistema operativo libre tipo UNIX. Es uno de los principales ejemplos de software libre y código abierto. Linux está licenciado bajo la GPL v2 y está desarrollado por colaboradores de todo el mundo. El desarrollo del día a día tiene lugar en la *Linux Kernel Mailing List*.

Es ahí donde sale el proyecto GNU/Linux, que incorpora el núcleo original de Linux (kernel) y el GNU que representa la libertad. Este kernel ha ido evolucionando con el pasar de los tiempos gracias a la ayuda de muchos colaboradores.

*“El núcleo Linux puede correr sobre muchas arquitecturas de máquina virtual, tanto como host del sistema operativo o como cliente. La máquina virtual usualmente emula la familia de procesadores Intel x86, aunque en algunos casos también son emulados procesadores de PowerPC o AMD.”<sup>4</sup>*

## 6.2 SOFTWARE LIBRE (FREE SOFTWARE)

El Software libre se refiere a un asunto de libertad y no de precios, en la actualidad muchos gobiernos, empresas e instituciones en general están migrando a la utilización de Linux, debido a sus múltiples beneficios y sobre todo costos, Según la Free Software Foundation, *“El software libre es una cuestión de la libertad de los usuarios de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. Más precisamente, se refiere a cuatro tipos de libertades para los usuarios del software:*

- *La libertad de ejecutar el programa, para cualquier propósito (libertad 0).*
- *La libertad de estudiar cómo trabaja el programa, y adaptarlo a sus necesidades (libertad 1). El acceso al código fuente es una condición necesaria.*
- *La libertad de redistribuir copias para que pueda ayudar al prójimo (libertad 2).*

---

<sup>4</sup> WIKIPEDIA, Enciclopedia Libre 2010. Núcleo de Linux. [[http://es.wikipedia.org/wiki/N%C3%BAcleo\\_Linux](http://es.wikipedia.org/wiki/N%C3%BAcleo_Linux) ], [en línea] [Consulta: 25 agosto 2010].



- *La libertad de mejorar el programa y publicar sus mejoras, y versiones modificadas en general, para que se beneficie toda la comunidad (libertad 3). El acceso al código fuente es una condición necesaria.”<sup>5</sup>*

Entonces cuando se habla del software libre es la libertad para poder trabajar, modificar, redistribuir y mejorar el software.

### 6.2.1 Características del software libre

Entre las principales características podemos citar las siguientes:

- Los usuarios de software libre esperan licencias abiertas y la inexistencia de métodos de activación
- Los usuarios de software libre esperan actualizaciones y parches regulares
- Los usuarios de software libre esperan poder trabajar de la manera que ellos decidan
- Los usuarios de software libre quieren el control de sus propios sistemas
- Los usuarios de software libre exploran.
- Los usuarios de software libre esperan poder auto ayudarse.
- Los usuarios de software libre no tienen miedo a la línea de mandatos.
- Los usuarios de software libre aprenden categorías de software, no programas.
- Los usuarios de software libre esperan poder acceder a los desarrolladores y a otros miembros de la comunidad.

Vistas así de esta manera, estas 9 expectativas podrían invertirse. Imagínate un usuario que opera con licencias propietarias y métodos de activación, sin actualizaciones ni parches regulares para sus sistemas, que trabajan con sistemas no como ellos quisieran, sino como dicta el sistema por sus limitaciones para ser modificado y porque no tienen el control sobre los mismos. Adicionalmente, imagina que ese usuario no explora

---

<sup>5</sup> Proyecto GNU. 2010. GNU/LINUX. [en línea], [<http://www.gnu.org/philosophy/free-sw.es.html>], [Consulta: 12 de septiembre 2010].



soluciones para sus problemas, no tiene capacidad para auto asistirse, no conoce la línea de comandos, no es capaz de agrupar por categorías los programas, sino que sólo conoce soluciones específicas y particulares, no tiene manera humana de acceder directamente a los desarrolladores y su círculo de colaboradores.

### 6.2.2 Ventaja de la Utilización de Software libre

Las ventajas más notables podemos mencionar las siguientes

- Más económico: porque muchos de los sistemas basados en LINUX son gratuitos
- Más fácil de actualizar y más seguros
- Linux es muy robusto, estable y rápido: Ideal para servidores y aplicaciones distribuidas.
- Linux es libre: Esto implica no sólo la gratuidad del software, sino también que Linux es modificable y que Linux tiene una gran cantidad de aplicaciones libres en Internet. Todo ello arropado por la inmensa documentación de Linux que se puede encontrar en la red.
- Linux ya no está restringido a personas con grandes conocimientos de informática.

Que no llame la atención si esta filosofía está siendo ya comercializada, por lo que existen versiones de Linux que están exigiendo que se les pague por el uso de estas licencias, claro que el precio es mucho menor que el que proporciona Microsoft, pero la idea de software libre es la que tiene que cumplir las cuatro libertades. Richard Stallman, personaje fundador de esta filosofía, manifiesta que la idea que software libre aún está latente en el mundo, su organización existen muchos desarrolladores que se mantienen con sus ideales y apoyando para que esta filosofía se expanda lo más pronto posible por el mundo. Hay que tomar en cuenta que Open Source y el software libre comparten muchas de sus licencias, pero la filosofía Open Source se enfoca más a la excelencia y el perfeccionamiento técnico como objetivo principal, algo que la filosofía del software



libre no está muy de acuerdo.

Es por esto que las universidades estatales deberían ser las primeras en empezar a migrar a esta tendencia y más aun los que están inmersos dentro del desarrollo de sistemas, solo así se podrá seguir avanzando en el progreso informático dejando a un lado muchos egoísmos, teniendo el código ahí para poder estudiarlo y adaptarlo a las necesidades, solo así se llegará muy lejos y se dejará ser dependientes de Microsoft.

### **6.2.3 Licencias en el Software Libre**

Como ya se ha visto en todo proceso de creación y uso de algo realizado, puede ser esto un trabajo, una obra, un cuadro de pintura, una obra literaria respectivamente. En todos estos modelos de creación y posterior uso se tiene un inventor, un creador, un autor, por esta razón es que cuando alguien más va a usar el producto creado generalmente se lo atribuye a su creador o se hace referencia de él, porque es su originalidad puesta en escena para obtener la obra o el trabajo, por esto cuando alguien usa el trabajo de otros debe referirse a su creador para darle así el origen, en si esto llevado a los negocios de hoy en día desencadena lo que comúnmente se llama licencias esta es la indicación de lo que se puede hacer con el trabajo en referencia a esa licencia y también posee referencias del autor o autores.

Por este razón las licencias son muy usadas en el mundo de los negocios y no podía ser la excepción que en el ambiente de la tecnología no hubiera esto de las licencias, ya que ellas permiten el mejor uso de las creaciones de un software o de un hardware, es por ello que se tiene licencias privativas y no privativas o libres, en el caso de las licencias privativas, son aquellas que no permiten el uso indebido, por ejemplo el caso del sistema operativo Windows, este exige un pago previo y su licencia es restringida.

Para el caso de las licencias no privativas o libres pues aquí hay varias y de diferente tipo, las hay para el open source y para el software libre. Para entender mejor el uso de las licencias, se tiene que tener bien claro lo que es el software libre, básicamente no es



un software gratuito por su traducción al español de free software, sino más bien “*es un software disponible para cualquiera que desee utilizarlo, copiarlo y distribuirlo, ya sea en su forma original o con modificaciones. Es importante no confundir software libre con software gratis, porque la libertad asociada al software libre de copiar, modificar y redistribuir, no significa gratuidad. Existen programas gratuitos que no pueden ser modificados ni redistribuidos.*”<sup>6</sup>

**Licencia Copyleft**, esta es una licencia del software libre que permite copiar, modificar y distribuir el software que esta licenciado con esta modalidad, como el copyright es una licencia que restringe el uso del software, pues así como hay una licencia copyright para el software privativo, también hay la licencia copyright para el software libre y esta licencia permite todas las libertades, entonces a esta se la denomina copyleft.

**Licencia GPL**, “*es la Licencia Pública General GNU (GNU General Public License GPL) es la licencia que acompaña los paquetes distribuidos por el Proyecto GNU, más una gran variedad de software que incluye el núcleo del sistema operativo Linux*”<sup>7</sup>. Con una licencia GPL, se tiene la seguridad de que este software no va a formar parte de un software privativo o propietario por la misma legislación de la licencia, ya que esta permite todas las libertades del software libre y por lo tanto tiene todas las atribuciones que un copyright internacionalmente, es decir es muy reconocida.

**La licencia Debian** “*es parte del contrato realizado entre Debian y la comunidad de usuarios de software libre, y se denomina Debian Free Software Guidelines (DFSG). En esencia, esta licencia contiene criterios para la distribución que incluyen, además de la exigencia de publicación del código fuente:*

---

<sup>6</sup> Licencias Software libre 2011 [en línea], [<http://www.informatica-hoy.com.ar/software-libre-gnu/Tipos-de-licencia-de-Software.php>], [Consulta: 20 de mayo del 2011]

<sup>7</sup> Licencias Software libre 2011 [en línea], [<http://www.informatica-hoy.com.ar/software-libre-gnu/Tipos-de-licencia-de-Software.php>], [Consulta: 20 de mayo del 2011]



1. *La redistribución libre.*
2. *El código fuente debe ser incluido y debe poder ser redistribuido*
3. *Todo trabajo derivado debe poder ser redistribuido bajo la misma licencia del original*
4. *Puede haber restricciones en cuanto a la redistribución del código fuente, si el original fue modificado*
5. *La licencia no puede discriminar a ninguna persona o grupo de personas, así como tampoco ninguna forma de utilización del software*
6. *Los derechos otorgados no dependen del sitio en el que el software se encuentra;*
7. *La licencia no puede 'contaminar' a otro software”<sup>8</sup>*

**Licencia OpenSource**, es una licencia que está basada en la licencia de Debian, porque el principal creador del DFSG, tomó como base este manifiesto para crear los lineamientos del openSource, que en si es el término con el que se conoce al software distribuido y desarrollado libremente.

**Licencia CCO.** *Es la licencia Creative Commons de publicación de dominio es decir con esta licencia se publica los documentos realizados para que el usuario haga uso de ellos haciendo referencia al autor y es compatible con la licencia GNU GPL”<sup>9</sup>*

A continuación se considera algunas definiciones importantes que se debe tomar en cuenta a la hora de usar un software, como por ejemplo

**“Software Propietario.** *El Software propietario es aquel cuya copia, redistribución o modificación están, en alguna medida, prohibidos por su propietario. Para usar, copiar o redistribuir, se debe solicitar permiso al propietario o pagar.*

---

<sup>8</sup> Licencias Software libre 2011 [en línea], [<http://www.informatica-hoy.com.ar/software-libre-gnu/Tipos-de-licencia-de-Software.php>], [Consulta: 20 de mayo del 2011]

<sup>9</sup>Softwarelicenses.2011.[en línea],[<http://www.gnu.org/licenses/icenselist.html#GPLCompatibleLicense>], [Consulta: 23 de mayo del 2011]



*Software Comercial. El Software comercial es el software desarrollado por una empresa con el objetivo de lucrar con su utilización. Nótese que "comercial" y "propietario" no son lo mismo. La mayor parte del software comercial es propietario, pero existe software libre que es comercial, y existe software no-libre que no es comercial*"<sup>10</sup>.

Además de estos conceptos, hay que tomar en cuenta las directrices del software libre de Debian (DFSG), las cuales constan en la licencia de Debian que toma como base estas directrices:

1. *“Libre redistribución, la licencia de un componente de Debian no puede restringir a un tercero el vender o entregar el programa como parte de una distribución mayor que contiene programas de diferentes fuentes. La licencia no debe solicitar «royalties» u otras comisiones para su venta.*
2. *Código fuente, el programa debe incluir el código fuente completo, y debe permitir la distribución en forma de código fuente y en forma compilada (binario).*
3. *Trabajos derivados, la licencia debe permitir modificaciones y trabajos derivados y debe permitir que estos se distribuyan bajo los mismos términos que la licencia del programa original.*
4. *Integridad del código fuente del autor, la licencia puede restringir la distribución del código fuente en forma modificada sólo si la licencia permite la distribución de parches (patch files) para poder modificar el código fuente original del programa en el momento de compilarlo. La licencia debe permitir explícitamente la distribución de software a partir de código fuente modificado. La licencia puede obligar a los trabajos derivados a llevar un nombre o número de versión diferentes del programa original. Esto es un compromiso. El grupo de Debian anima a todos los autores a no restringir ningún fichero, fuente o compilado, de ser modificado.*

---

<sup>10</sup> Licencias Software libre 2011 [en línea], [<http://www.informatica-hoy.com.ar/software-libre-gnu/Tipos-de-licencia-de-Software.php>], [Consulta: 20 de mayo del 2011]



5. *No discriminación contra personas o grupos, la licencia no debe discriminar a ninguna persona o grupo de personas.*
6. *No discriminación en función de la finalidad perseguida, la licencia no puede restringir el uso del programa para una finalidad determinada. Por ejemplo, no puede restringir el uso del programa a empresas con fines comerciales, o en investigación.*
7. *Distribución de la licencia, los derechos y libertades de uso asociados al programa deben aplicarse en la misma forma a todos aquellos a los que se redistribuya el programa, sin necesidad de pedir una licencia adicional para estas terceras partes.*
8. *La licencia no ha de ser específica para Debian, los derechos asociados al programa no deben depender de que el programa sea parte o no del sistema Debian. Si el programa es extraído de Debian y usado o distribuido sin Debian, pero manteniendo el resto de las condiciones de la licencia, todos aquellos a los que el programa se redistribuya deben tener los mismos derechos que los dados cuando forma parte de Debian.*
9. *La licencia no debe contaminar a otros programas. la licencia no debe poner restricciones sobre otros programas que se distribuyan junto con el programa licenciado. Por ejemplo, la licencia no puede insistir que todos los demás programas distribuidos sobre el mismo medio deben ser software libre.*
10. *Ejemplos de licencias, las licencias GPL, BSD, y Artística son ejemplos de licencias que se considera libres.*

*La expresión contrato social con la comunidad de software libre fue sugerida por Ean Schuessler. El primer borrador del documento de las directrices de debian fue escrito por Bruce Perens, fue modificado por los demás desarrolladores de Debian durante una conferencia por correo electrónico que duró un mes (junio de 1997) y finalmente aceptado como la normativa pública del proyecto Debian.*



*Posteriormente, Bruce Perens eliminó las referencias específicas a Debian en las directrices de Debian para el Software Libre (DFSG), creando la definición de Open Source.”<sup>11</sup>*

### 6.3 DEBIAN

*“Debian es un sistema operativo (S.O.) libre, para su computadora. El sistema operativo es el conjunto de programas básicos y utilidades que hacen que funcione su computadora. Debian utiliza el núcleo Linux (el corazón del sistema operativo), pero la mayor parte de las herramientas básicas vienen del Proyecto GNU; de ahí el nombre GNU/Linux.*

*Debian GNU/Linux ofrece más que un S.O. puro; viene con 25000 paquetes, programas precompilados distribuidos en un formato que hace más fácil la instalación en su computadora.”<sup>12</sup>*

La figura 6.2 muestra el logo del sistema operativo Debian GNU/LINUX.



**Figura 6.2 Logo Debian**

<sup>11</sup> Documentación Oficial del Proyecto Debian. 2011. [en línea],  
[[http://www.debian.org/social\\_contract.es.html/](http://www.debian.org/social_contract.es.html/)], [Consulta: 25 de mayo del 2011]

<sup>12</sup> DEBIAN 2010. Página oficial de debían. Información general de Debian. [en línea],  
[<http://www.debian.org/index.es.html>], [Consulta: 20 de septiembre 2010].



### 6.3.1 El proyecto Debían

El proyecto Debian, es una organización voluntaria con tres documentos iniciales, por ejemplo, el **Contrato Social de Debian**, define las bases por las cuales el proyecto y sus desarrolladores tratan los asuntos. Además las **Directrices de software libre de Debian**, definen los criterios del Software libre y dictan qué software es aceptable para la distribución, según lo referido al contrato social. Estas pautas también se han adoptado como base de la definición del Open Source. Y la **Constitución de Debian**, describe la estructura de la organización para la toma de decisiones de manera formal dentro del proyecto.

Actualmente, el proyecto incluye más de mil desarrolladores. Cada uno de ellos posee algún lugar en el proyecto ya sea relacionado con los paquetes: mantenimiento, documentación, control de calidad o relacionado con la infraestructura del proyecto: coordinación de lanzamientos, traducciones de web, etc.

El proyecto mantiene listas de correo y un sistema que permite a toda la comunidad leer y reportar bugs (erros del sistema, que son enviados a la comunidad para sean corregidos). De esta forma, trabajan de manera conjunta usuarios y desarrolladores.

Existen otros medios de comunicación con colaboradores y usuarios, principalmente por medio del IRC (Protocolo de comunicación en tiempo real basado en texto chatear) y particularmente en el servidor Freenode (red de servidores IRC). Todos estos medios poseen una gran concurrencia, siendo frecuentados por multitud de usuarios y desarrolladores.

Además, los desarrolladores pueden tomar decisiones generales obligatorias para una resolución o una elección general. Toda votación es realizada por el método Schulze, que un sistema de votación usado por la comunidad Debian.



### 6.3.2 Debian como distribución

Hay una serie de distribuciones basadas en Debian, algunos usuarios podrían echar un vistazo a estas distribuciones, además de los comunicados oficiales de Debian. Esto se hace por varias razones (mejor soporte de localización, soporte de hardware específico, simplifica la instalación, etc.)

Debian celebra y alienta a las organizaciones que quieran desarrollar nuevas distribuciones basadas en Debian, sin embargo, en el espíritu de Debian contrato social, les pide que contribuyen con su trabajo a la distribución principal, de modo que, todos los usuarios pueden beneficiarse de las mejoras.

Algunas distribuciones basadas en debían

- COLLAX
- DAMN SMALL LINUX
- DEBIAN JP
- EMBEDDED DEBIAN, <http://www.emdebian.org/>
- EURONODE, <http://euronode.org/>
- FLOPPIX, <http://floppix.ccai.com/>
- GIBRALTAR
- GNUstep LIVE CD, <http://io.debian.net/~tar/gnustep/>
- KANOTIX, <http://www.kanotix.com/>
- KNOPPIX, <http://www.knopper.net/knoppix/>
- LINEX
- LINSPIRE, <http://www.linspire.com/>
- LINUX ADVANCED, <http://www.linuxadvanced.at/> [in German]
- MEPIS, <http://www.mepis.org/>
- M.N.I.S. OCERA, <http://www.mnis.fr/en/products/>
- PUREOS, <http://pureos.org/>
- RAYS LX



- SIDUX, <http://sidux.com/>
- STONEGATE
- UBUNTU
- UNIVENTION CORPORATE SERVER
- XANDROS

A continuación se va hacer referencia de algunas distribuciones basadas en Debian. El proyecto **Adamantix** tiene como objetivo crear una plataforma Linux de alta seguridad, pero utilizable. Para lograr esto, el proyecto utilizará las soluciones de seguridad disponibles actualmente para Linux (como parches del kernel, parches del compilador, los programas relacionados con la seguridad y técnicas) y tejer estas junto a una plataforma Linux de alta seguridad.

**Collax** (antes BenHur) es un dispositivo de servidor basado en Debian. Se proporciona un firewall, servidor de correo electrónico con antivirus, servidor de fax, puerta de enlace VPN, servidor SMS, web proxy y funcionalidades de acceso remoto (RADIUS).

**Debian JP** es un esfuerzo voluntario impulsado por la intención de proporcionar una distribución basada en Debian orientada a Japón los usuarios finales. Las mejoras incluyen la internacionalización de la distribución Debian, con todos los cambios que más tarde contribuyó al proyecto principal

**Ubuntu** es una distribución Linux que se inicia con la amplitud de Debian y añade lanzamientos regulares (cada seis meses), un claro enfoque en el usuario y la usabilidad y un compromiso con las actualizaciones de seguridad con 18 meses de apoyo para cada lanzamiento, Ubuntu trae las últimas versiones de GNOME y KDE, así como una selección de servidor y del software de escritorio que hace que una experiencia de escritorio cómoda.



## 6.4 LOS REPOSITARIOS

En la vida cotidiana, se debe tener siempre algún dinero para imprevistos o para ocasiones que ameriten el gasto inmediato del dinero guardado para este fin. También en la vida se tiene alimentos guardados para momentos de escases, esos alimentos están almacenados en la alacena, donde reposan hasta que se los ocupa en un instante dado, como la comida que nos alimenta diariamente, bueno estos son ejemplos analógicos de repositorio, ya que repositorio en el campo informático se define como un depósito, archivo o un sitio centralizado, donde se almacena y mantiene información digital, habitualmente bases de datos o archivos informáticos.

### 6.4.1 ¿Qué son los paquetes?

Un paquete de software es una serie de programas que se distribuyen conjuntamente. Algunas de las razones suelen ser que el funcionamiento de cada uno complementa o requiere de otros, además de que sus objetivos están relacionados como estrategia de mercadotecnia.

Muchos sistemas operativos modernos emplean sistemas de gestión de paquetes que permiten que el administrador del sistema instale o desinstale paquetes, sin que en ningún momento queden paquetes instalados que no funcionen por falta de otros incluidos. El sistema de gestión de paquetes usualmente también se ocupa de mantener las dependencias entre paquetes: si un paquete se requiere de otro, el sistema se encarga de instalar éste primero.



## 6.4.2 ¿Qué es un repositorio de paquetes?

*“El origen de la palabra repositorio deriva del latín repositorium que significa armario, alacena. Este término es recogido del Diccionario de la Real Academia (DRAE) como: lugar donde se guarda algo”<sup>13</sup>.*

Los repositorios o depósitos como se los puede llamar también, son preparados para ser repartidos o accedidos mediante internet o por un medio físico como CDs, DVDs o también por medio de una red local, por esta razón pueden necesitar permisos de ingreso o autenticación o pueden ser de carácter público, entonces no necesitarían de ninguna clave para su acceso, los más reconocidos suelen ser los repositorios académicos e institucionales.

En el mundo actual las computadoras de escritorio sirven para hacer las tareas que comúnmente se hacen en una oficina, pero estas computadoras al momento que se dañan o sufren alguna avería por motivo de virus o algún desperfecto de hardware, la información ahí contenida queda inutilizable a diferencia de los sistemas de mainframes, súper computadores, sistemas online, sistemas montados en alguna red local, estos sistemas son con backups o respaldos, que permiten que los datos ahí almacenados puedan ser utilizables por el usuario o a su vez por otro programa y de esta manera no perder el acceso y el uso. Entonces aquí se ve la utilidad de los repositorios que almacenan información. ya sea como respaldo o como información de primera mano, pero bueno los repositorios en cambio en el área del software libre o en los sistemas GNU/LINUX, son depósitos de datos o paquetes que se usan en una manera intensa en el manejo de estos sistemas libres, estos repositorios son almacenes de software que están listos para ser instalados en las computadoras con ayuda de un gestor de paquetes, estos almacenes de software o paquetes de computadoras están unidos en un solo contenedor manejados por alguna aplicación que se llama gestor de paquetes, su nombre se debe a que direcciona los paquetes de software para que cada paquete tenga su dirección propia dentro del repositorio y pueda ser usado o instalado como si existiese

---

<sup>13</sup> WIKIPEDIA, Enciclopedia Libre. 2010. Información general de repositorios. [<http://es.wikipedia.org/wiki/Repositorio>], [en línea] [Consulta: 15 octubre 2010].



solo ese paquete y este pueda ser utilizado por el usuario.

De esta manera los paquetes tratados por el gestor, se unen de uno, en uno y se crea un conjunto de paquetes, donde cada paquete lleva el nombre, la arquitectura, el creador y datos de interés del paquete o software, esto conlleva a que exista una relación de paquetes dentro del conjunto de paquetes y de esta manera se va creando el repositorio de paquetes, que en si es una agrupación de paquetes listos para instalarse en el ámbito de sistemas GNU/LINUX.

### 6.4.3 Características y funcionamiento de un repositorio

Como los repositorios en GNU/LINUX, son un conjunto de paquetes listos para ser ejecutados, de ahora en adelante se llamara repositorio, este se basa justamente en encontrar la manera de trabajar con uno de ellos, es decir tener un repositorio de paquetes para la distribución UNLIX, estos repositorios manejan paquetes y cada uno es hecho por un conjunto de programadores, los mismo que con el pasar del tiempo sacan nuevas versiones para que sean utilizados en sus computadores o simplemente para ser más actuales en su uso, porque los paquetes implementan nuevas funcionalidades. El termino versión es conceptualizada como: *“una versión, revisión o edición de producto, es el estado en que se encuentra dicho producto en un momento dado de su desarrollo o modificación”*<sup>14</sup>. Hay que tener en cuenta, esto de las versiones de los paquetes, porque un repositorio siempre debe estar actualizado, con ello se quiere decir que sus paquetes deben estar actualizados.

Por ello en GNU/LINUX, el kernel es el núcleo de un sistema y es la parte principal con la cual un sistema operativo puede funcionar a la perfección, sin este kernel el sistema en si no existiría, ya que no habría comunicación entre el hardware y las diferentes partes de software, de esta manera la maquina entiende las ordenes y sus funciones.

---

<sup>14</sup> WIKIPEDIA, Enciclopedia Libre. 2010. Información Control de versiones. [[http://es.wikipedia.org/wiki/Control\\_de\\_versiones](http://es.wikipedia.org/wiki/Control_de_versiones)], [en línea] [Consulta: 23 octubre 2010].



El núcleo es Linux en GNU, debido a que sobre este kernel se adiciona muchos programas complementarios que en si llegan a formar la distribución Linux, esta distribución se la conoce como GNU/LINUX y es conformada por paquetes, *“un paquete viene a ser un pieza de software que cumple una determinada funcionalidad, así pues un paquete puede ser una biblioteca un conjunto de plugins o incluso un conjunto de iconos, etc.”*<sup>15</sup>. En definitiva una distribución puede verse como un rompecabezas formado por piezas de software que llegan a interactuar una con otra para funcionar bien, entonces un repositorio es un almacén donde se guardan los paquetes o las piezas de software para que sean instalados, es decir es la caja donde se guardan las piezas de la distribución y puede estar en el internet, en un medio extraíble o en el disco duro. Hay variedad repositorios como distribuciones que existen en el internet cada una con su funcionalidad y uso.

Luego para manejar estos repositorios existen los llamados sistemas de gestión de paquetes, que son los encargados de la instalación actualización y desinstalación de los paquetes de la distribución. El gestor de paquetes pone en evidencia todas las instrucciones de instalación que contiene cada paquete, como son información, función, ficheros que contiene, dependencias, etc. Estos datos hacen que el gestor se comporte como el constructor del repositorio, permitiendo que se ordene y se instale cada paquete según su dependencia y su ubicación dentro de repositorio organizándolo todo, por eso es muy importante. Entre los más conocidos se tiene el gestor de paquetes urpmi de mandriva Linux, el dpkg para Debian, entre otros.

Los sistemas de gestión de paquetes son diferentes a los formatos de paquetes, el sistema gestor de paquetes maneja paquetes que tienen sus propios formatos, es decir el sistema es *“un conjunto de reglas propias de una distribución que le indican donde se ubican los programas, como se instalan demonios en el sistema, que ficheros de configuración son accesibles (o modificables) por los programas, así como las distintas interacciones entre los paquetes, indicando por ejemplo, si dos programas tienen*

---

<sup>15</sup> BOGDRAKE. 2010. Información sobre repositorios. [<http://blogdrake.net/que-es-un-repositorio>, [en línea] [Consulta: 16 noviembre 2010].



*incompatibilidades y no pueden coexistir en el mismo sistema.*<sup>16</sup>

Ahora se tratará en dpkg, que es el gestor de bajo nivel de Debian, es donde se puede instalar, ubicar paquetes, actualizar, pero no se puede filtrar paquetes que no sean dependientes o que están siendo necesitados por otros paquetes para su instalación, es decir dpkg, no los identifica bien para eso existe apt, que es un herramienta que permite la instalación de paquetes con dependencias.

#### **6.4.4 Alcance de un repositorio**

Un repositorio es una herramienta que sirve para actualizar el sistema operativo, ya sea en caliente desde internet o en frío desde los DVDs, por eso el alcance que tiene un repositorio es general, porque sirve para renovar el software del sistema, ya que contiene los paquetes necesarios que el usuario desee instalar y como está en un servidor publicado desde internet es accesible desde cualquier lugar y muy útil a la hora de usar el sistema.

Las herramientas que manejan los repositorios son: apt, synaptic entre otras. Estas herramientas permiten la gestión de los paquetes del repositorio teniendo para esto la ayuda de comandos que hacen que se cargue tal o cual paquete en el sistema.

En si el repositorio es un conjunto de programas listo para usarse y que están presentes en el internet, desde donde se puede tener acceso por todos los usuarios y obtener algún paquete cargado en él, para poder agregarlo a nuestro sistema.

---

<sup>16</sup> DEBIAN. 2010. Página oficial de debían. Información general de Debian. [<http://www.debian.org/international/spanish/contrib/paqifaz.html>], [en línea] [Consulta: 23 noviembre 2010].



#### **6.4.5 Aspectos necesarios para elaborar repositorio.**

Para elaborar un repositorio se debe tener una lista de los paquetes que van a conformar el repositorio, luego de eso hay que ver que los paquetes que son de Debian se actualicen directamente del repositorio de Debian y los que no son de Debian conformen un grupo todos, para que esta lista se actualice directamente de los sitios correspondientes.

Luego con todos estos paquetes cargados en un solo grupo se los agrega a un disco local y se lo pone como servidor espejo para publicarlo hacia el internet y de esta manera queda creado el repositorio.

#### **6.4.6 El repositorio de Debian**

El repositorio de Debian, es un almacén de programas de casi 80 gigas de paquetes aproximadamente (considerando dos arquitecturas, la amd64 y la i386), los mismos que están disponibles para los usuarios de todo el mundo gracias a la internet y como están presentes para todos, se los tiene que a nivel mundial. Para aminorar el tiempo de descarga de un paquete, el repositorio de Debian tiene servidores espejos para las diferentes regiones del mundo y estos permiten tener un acceso más rápido para sus descargas, por ejemplo servidores espejos en Chile, en Estados Unidos, en Europa etc.

Estos servidores espejos además de dar el servicio de descargas sirven como respaldo de los paquetes a nivel mundial, ya que debían es de todos. Por este motivo el presente proyecto, va a tomar los paquetes del proyecto Debian para ser actualizados en un espejo pero con la diferencia de que va a llevar el nombre del proyecto UNLIX no de Debian y de esta manera dejar publicado un repositorio tan robusto como lo es el de Debian.



## 6.4.7 apt get como herramienta de administración de paquetes

El apt get es el comando de la herramienta apt, esta sentencia seguida de la palabra install (instalar), sirve para instalar paquetes por ejemplo:

```
# apt-get install k3b
```

En el ejemplo se instala el programa de grabador de CDs/DVDs k3b, de esa forma apt gestiona los paquetes dentro del sistema, teniendo en cuenta también que manejar apt es cómo trabajar cualquier programa de gestión que tiene sus propias directrices y comandos.

Synaptic es una interfaz gráfica para apt, ya que apt solo funciona en tipo comandos de consola una de las tantas interfaces gráficas para apt es Synaptic, que gestiona los paquetes de forma visual con botones y atributos que poseen los entornos gráficos. La siguiente figura 6.3 muestra la interfaz gráfica de Synaptic

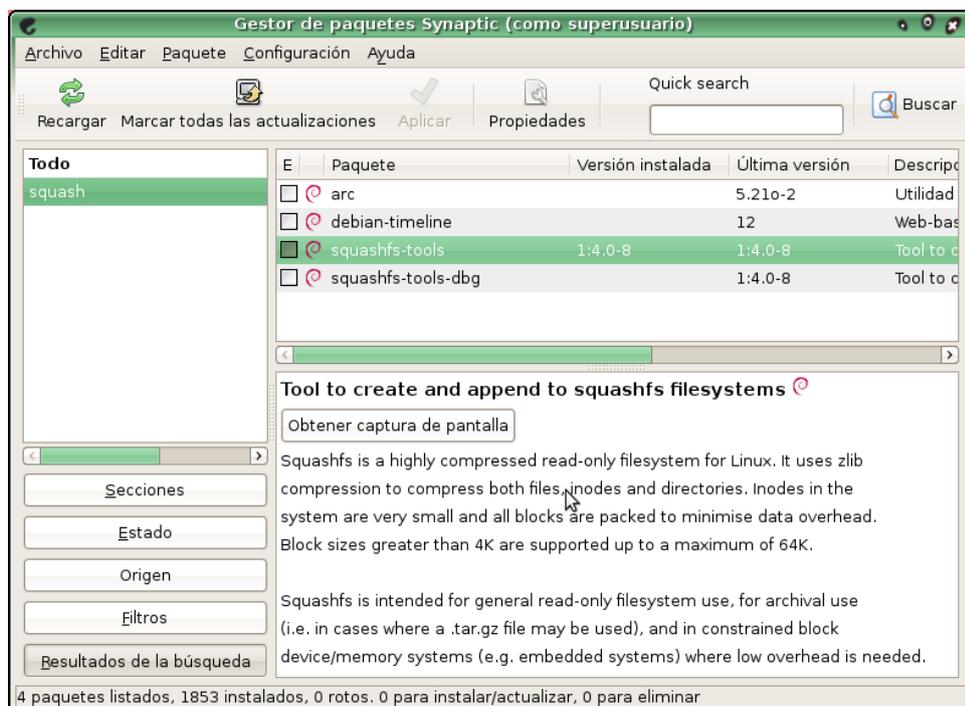


Figura 6.3 Gestor Synaptic



Apt es un conjunto de herramientas básicas dentro de Debian, permite

- Instalación de aplicaciones
- Eliminación de aplicaciones
- Actualizar aplicaciones.

Apt, básicamente resuelve problemas de dependencias, se ocupa de la instalación real y eliminación de paquetes (aplicaciones). Apt es muy poderoso, y se utiliza principalmente en la línea de comandos (consola / terminal). Sin embargo, muchos herramientas graficas que le permiten utilizar apt sin tener que tocar la línea de comandos.

Por otro lado, también se encuentra la opción aptitude, que es la herramienta recomendada para la interacción con el conjunto de la apt.

#### **6.4.8 Synaptic como gestor de paquetes**

El paquete de Synaptic utiliza el repositorio de Debian y permite su gestión mediante una interfaz amigable.

Haciendo un clic, es posible actualizar la lista de paquetes disponibles o marcar automáticamente todas las actualizaciones de los paquetes ya instalados.

Dispone de un avanzado filtro de búsqueda, es capaz de reparar las dependencias rotas de paquetes y permite deshacer y rehacer las últimas selecciones de paquetes.

Su interfaz gráfica, además, es personalizable mediante el menú de configuración.



*“Synaptic es un programa informático que es una interfaz gráfica GTK+ de APT, para el sistema de gestión de paquetes de Debian GNU/LINUX. Generalmente se utiliza Synaptic para sistemas basados en paquetes .deb pero también puede ser usado en sistemas”<sup>17</sup>*

#### **6.4.9 DVD como medio para actualizaciones de paquetes**

Una ayuda importante en la instalación de paquetes, es el uso del DVD/CD, este posee un repositorio de paquetes para poder trabajar desde un sistema operativo. Debian como Ubuntu direccionan su repositorio al DVD/CD, de no existir el paquete en dicho medio lo busca en otro medio como por ejemplo el internet o algún otro repositorio de paquetes que este configurado en el archivo `/etc/apt/sources.list`

#### **6.4.10 Los Scripts**

Un script básicamente es un conjunto de instrucciones, es decir un pequeño programa que necesariamente tiene que ser interpretado. Cuando se trabaja con Linux Equivale a un archivo \*.bat de MSDOS, con la diferencia de que no tiene que terminar con la extensión .bat, tiene que tener permisos de ejecución y puede ser mucho más complejo. Además en su forma más simple, es un archivo en modo texto, con una lista de comandos, que se ejecutan en ese orden de forma automática del mismo modo que si se lo escribe uno a uno en una consola de texto o una terminal.

En Linux se considera a script como un conjunto de instrucciones las cuales se ejecutan en forma secuencial por el bash de Linux.

Ejemplo: (usuario.sh)

---

<sup>17</sup> WIKIPEDIA, Enciclopedia Libre. 2010. Información de la herramienta synaptic. [<http://es.wikipedia.org/wiki/Synaptic>], [en línea] [Consulta: 12 noviembre 2010].



```
#!/bin/bash
echo 'Script para crear un usuario nuevo en el sistema'
adduser juan
passwd juan
exit
```

Para ejecutarlo, se abre la terminal y se va al directorio donde se encuentra el archivo por ejemplo: usuario.sh

La ejecución de este script se lo hace:

```
# sh usuario.sh
```

El sistema desvuelve las órdenes que posea este script, según como se encuentre programado.

## 6.5 LAS DISTRIBUCIONES

Una distribución GNU/LINUX, contiene un Kernel Linux, bibliotecas y paquetes de software. Es un conjunto de aplicaciones Linux preparadas para que el usuario las pueda instalar (o ejecutar) de forma sencilla.

La mayor parte de las distribuciones son fáciles de manejar. El usuario acostumbrado a utilizar Windows no tendrá grandes problemas para adaptarse a los sistemas GNU/LINUX.

### 6.5.1 ¿Que es una Distro o Distribución?

Una distribución Linux conocida también como “distro”, es una distribución de software con el núcleo Linux y entorno GNU formando así un sistema operativo, que incluye determinados paquetes de software para satisfacer las necesidades de un grupo



específico de usuarios, dando así origen a ediciones domésticas, empresariales y para servidores. Estos paquetes están compuestos en su totalidad por software libre.

*“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, etcétera. En el caso de incluir herramientas del proyecto GNU, también se utiliza el término distribución GNU/Linux.”<sup>18</sup>*

Cuando se habla de GNU/LINUX, que en si es software libre y cumple la licencia GPL, viene a la mente que a nivel mundial existen muchas entidades que usan este núcleo abierto de Linux para crear distribuciones, por ejemplo: Mandrake se creó en 1998 para hacer Linux fácil de usar y de hecho se considera una de las distribuciones más amigables para usuarios principiantes, más adelante en el año 2005 Mandrake se unió con Conectiva, creando Mandriva <http://www.mandriva.com/es/>. También se encuentra la distro conocida como Redhat, SuSe, que son plataformas apoyadas por grandes empresas en el mundo de la informática.

Debian es la más o una de las distribuciones más extendidas por el mundo, cabe destacar su herramienta apt, permite descargar paquetes de forma muy sencilla.

Gentoo es una distribución que sigue la política de las antiguas distribuciones BSD. Su comportamiento es similar a otras distribuciones como Debian, aunque se consigue algo más de optimización, ya que la filosofía del proyecto de Gentoo es hacer que el software sea específicamente compilado para la computadora del usuario. Obviamente eso deriva en tener que compilar los paquetes en un tiempo muy largo.

---

<sup>18</sup> WIKIPEDIA, Enciclopedia Libre. 2010. Información sobre distribuciones de Linux. [[http://es.wikipedia.org/wiki/Distribuci%C3%B3n\\_Linux](http://es.wikipedia.org/wiki/Distribuci%C3%B3n_Linux)], [en línea] [Consulta: 20 noviembre 2010].



En fin estos son unos ejemplos de distribuciones, existen muchas más como Ubuntu, entre otras, pero lo que hay que tomar en cuenta que las diferentes distribuciones son muy parecidas en su interfaz gráfica, se puede dar cuenta que los escritorios Gnome por ejemplo en diferentes distribuciones no varía mucho y no se diga con los KDE. En fin el mundo GNU/LINUX resulta una experiencia muy agradable, no solo para programadores o desarrolladores, sino también para el usuario final.

Algo importante de señalar, que el actual gobierno del Ecuador tiene un proyecto grande, el cual consiste en trabajar con el sistema Ubuntu en las entidades educativas, además los docentes ya están siendo capacitados para trabajar con esta distribución y más aún las llamadas escuelas del milenio que vienen incorporados sus equipos informáticos con el sistema operativo Ubuntu GNU/LINUX como el caso de la UNIDAD EDUCATIVA EXPERIMENTAL DEL MILENIO 5 DE JUNIO del cantón Macará

### 6.5.2 Las Metadistros

*“Se considera metadistro al conjunto de herramientas para la creación de distribuciones del sistema operativo GNU/Linux.*

*Una metadistro es una distribución hecha a medida por un grupo determinado de usuarios con un objetivo concreto. El fin último de una metadistro es que sea instalada, ya sea durante el arranque como desde el escritorio por medio de un instalador. El hecho de funcionar en modo Live cd no es más que un paso intermedio para el fin, por lo cual un metadistro no es lo mismo que live cd.*

*El objetivo del proyecto metadistros es ser un espacio de colaboración entre gente que desarrolla distribuciones (para un público determinado, con un objetivo determinado como universitarios, estudiantes, empresas entre otros) y no duplicar esfuerzos.*



*Una metadistro es la plataforma para personalizar una distribución GNU/Linux a un entorno de trabajo concreto.*<sup>19</sup>

### 6.5.3 Análisis para la creación de una distribución

Una metadistro consta de dos partes: el calzador es un conjunto de scripts, módulos del kernel y aplicaciones (busybox, por ejemplo) que preparan el entorno, montan el sistema de archivos en RAM y autoconfiguran el sistema para que el usuario final no tenga más que poner el cdrom en su equipo y esperar a que cargue.

La otra parte es una copia del sistema de archivos GNU/LINUX comprimido o Imagen squashfs (sistema de archivos comprimidos). Para hacer un Live Cd, se puede remasterizar uno que ya exista (knoppix, morphix, Ubuntu, Debian, etc..) o crearlo desde cero. La decisión de uno u otro depende de la pericia que se tenga sobre la distribución que se use como base, así como unos conocimientos básicos de scripting (trabajar con scripts) y la facilidad para remasterizar la distro objetivo.

### 6.5.4 Herramientas para la creación de una distro

Una de las más populares es Garfio es un proyecto basado en metadistros, que sirve para armar distribuciones personalizadas muy fácilmente, de una manera estructurada y organizada.

Dependiendo la distribución, tras la instalación se necesita un tiempo extra para dejarla a punto, instalando los programas, herramientas y configuraciones de interés del usuario, por suerte, en GNU/LINUX existen herramientas que ayudan a la creación de distros personalizadas, esto de hacer distribuciones ahorra trabajo al momento de implementar un sistema GNU/LINUX en varias computadoras. A continuación se citan algunas herramientas para construir distribuciones.

---

<sup>19</sup> WIKIPEDIA, Enciclopedia Libre. 2010. Información sobre meta distros de Linux. [[http://es.wikipedia.org/wiki/Metadistros\\_%28herramienta\\_GNU/Linux%29](http://es.wikipedia.org/wiki/Metadistros_%28herramienta_GNU/Linux%29)], [en línea] [Consulta: 28 noviembre 2010].



- LinuxAdmin, es una lista de estas aplicaciones que te ayudarán a crear tu propia distro.
- Remastersys, para distros basadas en Debian y Ubuntu.
- UCK, como su nombre indica, kit de customización de Ubuntu, toma como base una iso del sistema operativo.
- Reconstructor, toma como base Debian 5 y Ubuntu 9.04 y 9.10, se hace a partir del navegador web donde pide que se registre el usuario previamente, es muy parecido a SuseStudio.
- Revisor, para customizar distros basadas en Fedora.
- SuseStudio, para distros basadas en Suse.
- Pungi, esta herramienta va por línea de comandos para crear distros basadas en Fedora
- Builder, son scripts que usa gNewSense para crear su distro. Para
- Linux-Live, herramienta bastante avanzada para gente con conocimientos amplios del sistema.
- Linux From Scratch, permite hacer paso a paso los procesos para construir una “distro” a partir del código fuente.

**Garfio.-** es un proyecto basado en metadistros, que sirve para armar distribuciones personalizadas muy fácilmente, de una manera estructurada y organizada. Entre las características principales se considera:

- *“Su funcionamiento es modulado, lo cual facilita su desarrollo para todo tipo de personas (con algunos conocimientos básicos de Linux), ya que solo deberían agregar o quitar módulos.*
- *Se encuentra organizado por directorios que lo hace más visible y expansible.*
- *Trabaja con variables globales que se pasan como parámetro desde el booteo.*
- *Dispone de múltiples opciones para la detección de hardware y configuraciones del sistema.*
- *Arranque detallado y con debugger (depuración) en tiempo real.*



- *Garfio es actualmente la base de la distribución Tuquito GNU/Linux.* <sup>20</sup>

**Reconstructor.-** *“es un programa que permite a cualquier usuario personalizar una imagen iso de Debian y derivados”*<sup>21</sup>

**Remastersys.-** es un paquete muy potente para la creación de distribuciones para Ubuntu y Debian, *“es una herramienta que puede utilizar para hacer dos cosas con una ya existente de Debian, Ubuntu o derivados de instalación”*.<sup>22</sup>

- Puede hacer una copia de seguridad completa del sistema, incluidos los datos personales a un live cd o dvd que se puede utilizar en cualquier lugar e instalar.
- Puede hacer una copia distribuible que puede compartir con amigos. Esto no tiene ninguno de los datos personales del usuario en el mismo.

La figura 6.4 es el logo de la herramienta Remastersys



**Figura 6.4 Logo Herramienta Remastersys**

<sup>20</sup> GARFIO, Página oficial de garfio. 2010. Información de la herramienta garfio. [<http://www.garfio.org.ar/>], [en línea] [Consulta: 2 diciembre 2010].

<sup>21</sup> WIKIPEDIA, Enciclopedia Libre. 2010. Información sobre herramienta reconstructor. [<http://es.wikipedia.org/wiki/Reconstructor>], [en línea] [Consulta: 5 diciembre 2010].

<sup>22</sup> REMASTERSYS, Página oficial de remastersys. 2010. Información de la herramienta remastersys. [<http://remastersys.sourceforge.net/>], [en línea] [Consulta: 10 diciembre 2010].



### **6.5.5 El sistema Objeto**

Se considera sistema objetivo el resultado de la distribución que se va a crear, es decir el sistema operativo con todas las aplicaciones necesarias para la distribución. En base al sistema objetivo se construirá la imagen de este sistema el mismo que se convertirá en Live CD o Live DVD. La imagen recoge todo lo que contenga el sistema objetivo, programas, la personalización, incluso archivos que el usuario pueda tener en un computador.



## 7. EVALUACIÓN DEL OBJETO DE INVESTIGACIÓN

Una vez realizado el presente trabajo de investigación, es oportuno llevar a cabo la evaluación de cada uno de los objetivos planteados al inicio del mismo, tal y como se detalla a continuación:

**Objetivo Específico 1: Realizar un análisis de los paquetes de UNLIX para determinar posibles actualizaciones e inclusiones.**

Para el logro de este objetivo, se realizó una entrevista con los Ingenieros Maritza Quezada y Roberto Mora creadores del proyecto UNLIX en su primera versión, la constancia de la entrevista está en el Anexo D del presente documento, se analizó la motivación que llevo a concretar el proyecto UNLIX, así como el estudio de los paquetes base e indispensables del sistema operativo UNLIX, también las herramientas que utilizó para la creación de la distribución, dicho estudio consta en su proyecto de tesis cuyo tema versa “DESARROLLO DE UNA DISTRIBUCIÓN LINUX BASADA EN DEBIAN PARA EL ÁREA DE ENERGÍA, LAS INDUSTRIAS Y RECURSOS NATURALES NO RENOVABLES DE LA UNL”. Gracias a esta entrevista se pudo tener una idea más clara de los paquetes indispensables para UNLIX 2.0 GNU/LINUX, así como la inclusión de otros paquetes de software orientados al uso en otras carreras de la Universidad Nacional de Loja, los mismos que constan en el **capítulo 8.3.3 Instalación y actualización de las aplicaciones para UNLIX 2.0** de este documento.

**Objetivo Específico 2: Crear un repositorio general de paquetes para el sistema operativo UNLIX utilizando la versión estable de debian Lenny.**

Para lograr este objetivo se buscó un computador básico para ponerlo de servidor (**ver Tabla 8.7 características del servidor UNLIX 2.0**), al mismo que se le instaló la última versión de Debian, se comprobó el funcionamiento del servidor web apache, también el servicio ssh para acceder remotamente, además se instaló el paquete debmirror para que genere el espejo de los paquetes del repositorio oficial de Debian



(ver capítulo 8.2.2 Configuración previa a la Creación del Repositorio) y se procedió hacer el espejo de los repositorios de paquetes del Debian en su última actualización (ver capítulo 8.2.3 Configuración de los Scripts para el repositorio).

**Objetivo Específico 3: Montar el repositorio estándar de paquetes para UNLIX en un servidor del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables.**

Para el logro de este objetivo se lo hizo gracias a la ayuda de la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja, quienes manifestaron su interés por el proyecto UNLIX 2.0 y permitieron que se colocara un servidor dentro de la red de la Universidad, además permitieron que el servidor tenga todos los permisos necesarios para realizar descargas y réplicas desde el internet y también concedieron una dirección de red propia para el servidor. Para constancia de lo expuesto ver el **Anexo G** del presente documento de investigación.

**Objetivo Específico 4: Almacenar un repositorio primario en DVD para actualizar el sistema sin necesidad de conexión.**

Este objetivo se lo realizó gracias a una herramienta llamada “aptoncd” esta permite crear un repositorio de paquetes propias de Debian en una Unidad de CD/DVD, esta unidad se la puede usar en otro computador con UNLIX 2.0 o algún sistema con Debian o derivado del mismo, ya que los paquetes que constan en este CD/DVD son con la extensión .deb. Este CD/DVD se lo usaría en el caso de que el usuario quiera agregar un paquete que posea el CD/DVD y este fuera del campus de la Universidad Nacional de Loja o no posea Internet, **ver capítulo 8.5 CREACIÓN DE UN CD/DVD PARA ACTUALIZAR PAQUETES** de este trabajo investigativo. Además este CD/DVD viene adjunto en el presente proyecto de tesis.



**Objetivo Específico 5: Realización del plan de validación del repositorio de paquetes utilizando diferentes arquitecturas.**

Este objetivo se lo estableció cuando se hicieron las pruebas del proyecto, se pudo instalar UNLIX 2.0 GNU/LINUX en diferentes arquitecturas tanto en laptops como en computadoras de escritorio, además se agregó, actualizó y elimino paquetes de software de estas computadoras, las cuales se encuentran direccionadas al servidor de UNLIX 2.0 que está alojado en la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja. El servidor se lo denomina unixServer y posee la siguiente dirección IP **172.16.32.29** interna.



## **8. DESARROLLO DE LA PROPUESTA ALTERNATIVA**

### **8.1 NOCIONES GENERALES SOBRE UNLIX**

#### **8.1.1 Que es UNLIX?**

Unlix es un sistema operativo GNU/LINUX que por ser basado en Debian adquiere las características y licencias del software libre, Unlix fue pensado en sus inicios para satisfacer las necesidades de los alumnos y docentes del área de Energía Industrias y Recursos Naturales No Renovables de la Universidad Nacional de Loja, dando como resultado a un gran sistema operativo llamado con el acrónimo UNLIX formado por las siglas de “UNL” de la Universidad Nacional de Loja más “IX”, por ser GNU/Linux. Este sistema puede no solo ser usado por los estudiantes sino por el público en general gracias a sus mejorías en la adición de software útil para diferentes tareas dentro de una oficina, aula o en el hogar.

#### **8.1.2 Características de UNLIX.**

UNLIX posee buenas características como sistema operativo que las hereda de esos grandes sistemas como lo es Debian por ejemplo es un sistema libre, estable, posee un kernel 2.6.32, posee sistema de arranque basado en dependencias, tiene variedad de entornos gráficos como el gnome, kde entre otros, entre los principales estándares de UNLIX es que trabaja con paquetes de software .deb, además es de libre distribución, incluye código fuente, permite trabajos derivados. También se puede agregar que Unlix posee las siguientes características:

- Posee software para utilización tanto de oficina aula el hogar o empresarialmente.
- Tiene un grupo de herramientas para actualizar e instalar nuevo software estas herramientas son apt, synaptic y kpackage que agarran los datos de los repositorios puestos en `/etc/apt/sources.list`.



- Otra característica es que cumple con todos los estándares del software libre en todos sus estamentos.
- Posee un DVD Live, es decir este sistema está presente en un DVD vivo listo para usarse o instalarse si así lo deseara el usuario

### 8.1.3 Proyección de UNLIX

La perspectiva a futuro inmediato de UNLIX es tener su propio repositorio para que los usuarios de este buen sistema tengan un lugar de la red, donde descargar todas las actualizaciones de software que en la línea del tiempo aparecen para que los clientes finales actualicen sus computadoras, también UNLIX tendrá un DVD con un repositorio primario para que con este medio de almacenamiento se pueda actualizar los ordenadores sin necesidad de conectarse al internet.

En esencia UNLIX posee un robusto sistema ya que es basado en software libre más directamente basado en Debian, UNLIX se proyecta en su versión 2.0 a ser cimentado en la última versión estable de Debian hasta el momento.

### 8.1.4 Manejo Básico de UNLIX

UNLIX (primera versión) desde el momento que bootea (proceso que inicia al sistema operativo cuando el usuario enciende un computador) dentro de la computador, permite la opción instalar o a correr el sistema, se dice correr el sistema porque tiene la opción de un Live Cd puede ser arrancado desde la RAM como lo hacen los CD Live o DVD Live de otros sistemas, desde ese momento este sistema presenta facilidad de uso teniendo una guía estándar a seguir, puesto que como todos los programas computacionales tiene pasos similares en la iniciación del sistema y su posible instalación dependiendo de lo que decida el usuario como menú se tiene las siguientes opciones:



- **Arranque normal** este permite arrancar de manera normal el sistema en un modo Live Cd o Live DVD.
- **Arranque failsafe** este nos da la opción de arrancarlo en modo a prueba de errores
- **Arranque para portátil** aquí pues pone todo lo necesario para que inicie el sistema en las laptops.
- **Instalar** esta opción permite instalar el sistema en la computadora.

Teniendo estas opciones de booteo de UNLIX se puede decir que este sistema operativo es flexible y útil, debido a que UNLIX cuando se inicializa en el computador, es como cualquier otro sistema operativo en modo Live DVD con la diferencia de que viene cargado con la mayoría de herramientas de software necesarias para satisfacer a los usuarios en las tareas que ellos requieran dentro del aula, de la oficina, de la empresa o del hogar.

Para poder iniciar en el entorno de UNLIX se tiene que digitar el nombre de usuario y la contraseña que es: para usuario “unlix” y para la contraseña “unlix”, y si se quiere trabajar dentro del entorno grafico gnome de UNLIX o hacer algo que requiera cosas del sistema administrativo, pues pedirá la contraseña de superusuario que es “sistemas” eso en el modo de Live DVD, si el usuario lo instala en la su computador, pues se podrá poner restricciones o se podrá usar las mismas que tiene.

UNLIX, como sistema montado en la máquina sirve para manejar todo el hardware conectado a ella, además permite levantar todo el software previamente instalado y el que a futuro se instale, en conclusión UNLIX es un robusto sistema operativo GNU/LINUX basado en Debian Etch.



### 8.1.5 Los paquetes de UNLIX.

Los paquetes de software que incluye UNLIX en su primera versión, es el resultado de un estudio hecho a los usuarios en este caso docentes y estudiantes del Área de la Energía, las Industrias y de los Recursos Naturales No Renovables, de la Universidad Nacional de Loja y se tiene los siguientes principales paquetes que muestran las tablas 8.1, 8.2, 8.3, 8.4, 8.5 y 8.6

En multimedia	Versión
Sound Juicer	2.14.6
Rhythmbox	0.9.6
GnomeBaker	0.6.0
K3B	0.12.17
Totem-xine	2.16.5
Juk	2.3.2

**Tabla 8.1 Paquetes Unix 1.0 Multimedia**

En Gráficos	Versión
Draw	2.0.4
Capturar pantallas (Yelp)	2.14.3
Visor Imágenes (Eye)	2.16.3
KolourPaint	1.4.5
Gimp	2.2.13

**Tabla 8.2 Paquetes Unix 1.0 Gráficos**

En Ofimática	Versión
Dia	0.95
KPDF	0.5.5
OpenOffice	2.0.4
Planner	0.14.2

**Tabla 8.3 Paquetes Unix 1.0 Oficina**



En Desarrollo	Versión
Nvu	1.0
PHP	5.2.0
Apache	1.3.34
Nasm	0.98.38
Java	5.0
MonoDevelop	0.11
Scilab	4.0.12
Eclipse	3.2.1
NetBeans	6.0.1
Ruby	1.8.2
MySQL	5.0.32
Phyton	2.4.4

**Tabla 8.4 Paquetes Unix 1.0 Desarrollo**

En diseño	Versión
Geda	20060123
Grass	6.0.2
Blender	2.42 <sup>a</sup>
Qcad	2.0.5

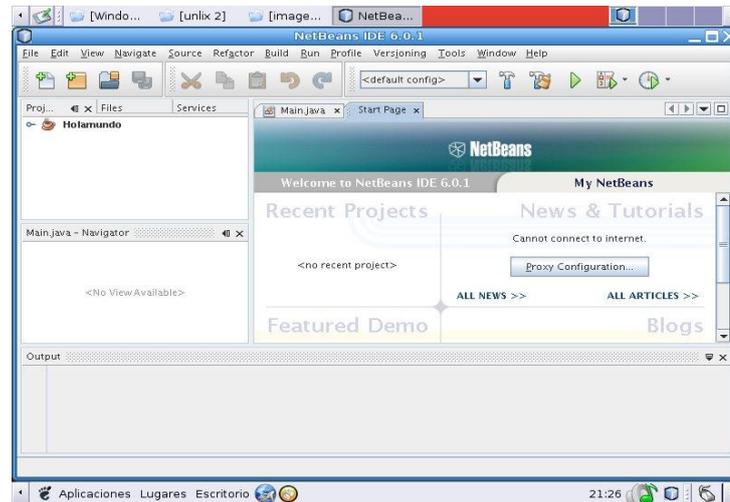
**Tabla 8.5 Paquetes Unix 1.0 Diseño**

En Internet	Versión
Gnome BitTorrent	3.4.2
Navegador Epiphany	2.14.3
Softphone Ekiga	2.0.3
Mozilla	1.7.8
IceWeasel	2.0.0
Evolution	2.6.3
Amsn	0.95

**Tabla 8.6 Paquetes Unix 1.0 Internet**



De la misma forma como ejemplo de la potencia de UNLIX, se tiene la iniciación de netbeans que es un poderoso programa para programación de java, se ejecuta sin problemas y está listo para usarse dentro de su entorno gráfico como muestra la figura 8.1



**Figura 8.1 Netbeans en UNLIX 1.0**

## 8.2 CREACIÓN DEL REPOSITORIO

Un repositorio se considera como un almacén de datos y por esta razón es una fuente al cual se puede acceder para utilizar esos datos.

Los repositorios o depósitos de datos, también son preparados para ser repartidos o accedidos mediante internet o por un medio físico como CDs, DVDs o también por medio de una red local, por esta razón pueden necesitar permisos de ingreso o autenticación o pueden ser de carácter público, entonces no necesitarían de ninguna clave para su acceso los más reconocidos suelen ser los repositorios académicos, institucionales o sitios donde se trabaja bajo la filosofía libre.

En el mundo actual las computadoras de escritorio nos sirven para hacer las tareas que comúnmente se hacen en una oficina pero estas computadoras al momento que se dañan o sufren alguna avería por motivo de virus o algún desperfecto de hardware la



información ahí contenida queda inutilizable, en cambio de los sistemas de mainframes, súper computadores, los sistemas online montados en la internet o en alguna red local estos sistemas son con backups o respaldos que permiten que los datos ahí almacenados o generados puedan ser utilizables por el usuario o a su vez por otro programa y de esta manera no perder el acceso y el uso, entonces aquí se ve la utilidad de los repositorios que almacenan información, ya sea como respaldo o como información de primera mano.

Los repositorios en cambio en el área del software libre o en los sistemas GNU/LINUX son depósitos de datos o programas que se usan en una manera intensa en el manejo de estos sistemas libres, estos repositorios son almacenes de software que están listos para ser instalados en las computadoras con ayuda de un gestor de paquetes, estos almacenes de software o programas de computadoras están unidos en un solo contenedor manejados por un solo programa o software que se llama gestor de paquetes su nombre se debe a que direcciona paquetes de software para que cada programa tenga su dirección propia dentro del repositorio y pueda ser usado o instalado como si existiese solo ese programa y pueda ser utilizado así por el usuario, de esta manera los programas tratados por el gestor se unen de uno en uno y se crea un paquete cuando se adiciona el programa y el nombre del programa, la arquitectura el creador del programa y demás datos de interés de este programa o software y así de paquete en paquete se hace un conjunto de paquetes o almacén de software formando de esta manera el repositorio de paquetes, que en si es un conjunto de programas listos para instalar, eso en el ámbito de GNU/LINUX.

En GNU/LINUX la idea de repositorio es tener un conjunto de paquetes o programas listos para ser usados y el presente proyecto está encaminado a utilizar un repositorio a partir de otro primario como lo es el repositorio de Debian Squeeze para la distribución de UNLIX 2.0 que es basada de Debian, en si estos repositorios manejan programas y cada programa es hecho por uno o un conjunto de programadores los mismo que con el pasar del tiempo sacan nuevas versiones para que sean utilizados, en muchos de los casos las nuevas versiones de programas incorporan nuevos módulos, por lo que



muchos usuarios optan por usar siempre las últimas versiones de los programas. Hay que tener en cuenta esto de las versiones de los programas, porque un repositorio siempre debe estar actualizado con ello se quiere decir que sus paquetes deben tener las últimas versiones, por ello es importante que el repositorio este en permanente actualización de sus paquetes.

Por ello en GNU/LINUX el Kernel es el núcleo del sistema y es la parte principal con la cual un sistema operativo puede funcionar a la perfección, sin este Kernel el sistema en si no existiría, no habría comunicación entre el hardware y el software y de esta manera la computadora no entendería las órdenes para que funcione la computadora, el núcleo LINUX es el Kernel a este se adiciona el conjunto de aplicaciones de GNU y se complementa variedad de programas complementarios que en si llegan a formar la distribución LINUX, esta distribución se la conoce como GNU/LINUX y es conformada por paquetes de software y cada paquete viene a ser un pieza de software que cumple una determinada funcionalidad. En definitiva una distribución puede verse como un rompecabezas formado por piezas de software que llegan a interactuar una con otra para funcionar todas de forma correcta, entonces un repositorio es un almacén donde se guardan los paquetes o las piezas de software para que sean instalados, es decir es la caja donde se guardan las piezas de la distribución y puede estar en el internet en un medio extraíble o en el disco duro. Hay muchos repositorios como distribuciones y existen en el internet cada una con su funcionalidad y uso.

Luego para manejar estos repositorios existen los llamados sistemas de gestión de paquetes que son los encargados de la instalación actualización y desinstalación de los paquetes de la distribución. El gestor de paquetes pone en evidencia todas las instrucciones de instalación que contiene cada paquete, como son información de cada paquete su servicio, ficheros que contiene, dependencias, etc. Estos datos hacen que el gestor se comporte como el constructor del repositorio permitiendo que se ordene y se instale cada paquete según se dependencia y su ubicación dentro de repositorio organizándolo todo, por eso es muy importante. Entre los más conocidos el urpmi de mandriva Linux, dpkg para Debian, Synaptic entre otros.



Para tener una idea del manejo visual de los paquetes se tiene la herramienta Synaptic que es una de tantas en el mercado del software, pero esta herramienta tiene la diferencia que es la oficial usada por Debian y que permite la instalación visual de los paquetes porque tiene interfaz gráfica intuitiva. Todos estos programas son un apoyo principal para utilizar un repositorio y a la hora de usarlos son fáciles de aprender y operar.

Los sistemas de gestión de paquetes son diferentes a los formatos de paquetes, el sistema gestor de paquetes maneja paquetes que tienen sus propios formatos es decir para el caso de Debian maneja paquetes .deb (es la extensión del formato de paquetes de software de Debian y derivadas).

Cuando se habla de herramientas de paquete, no se debe descuidar la opción dpkg que es el gestor de bajo nivel de Debian, ahí es donde también se puede instalar, ubicar paquetes actualizar pero no se puede filtrar paquetes que no son dependientes o que están siendo necesitados por otros paquetes para su instalación, es decir dpkg no los identifica bien para eso existe apt, que es un herramienta que permite la instalación de paquetes con dependencias, por esta razón apt es una herramienta superior a dpkg para gestionar paquetes y sirve mucho a la hora de instalar paquetes, una herramienta similar que esta al mismo nivel de apt, es aptitude que también sirve para instalar paquetes y manejarlos.

### **8.2.1 Requerimientos del Servidor para el Repositorio**

Para poder alojar un repositorio en una máquina se necesita que esta máquina tenga espacio libre de 80 GB aproximadamente que es lo más primordial para crear el repositorio, lo recomendable es que el disco duro sea de 120 GB o superior (si se toma en cuenta las dos arquitecturas i386 y amd64), el resto de características de hardware puede ser un CPU con mínimo de 512 de RAM, Pentium IV. Debian Squeeze trabaja bien en computadoras con estas características. El CPU servidor que se está utilizando



para el alojamiento del repositorio de paquetes de UNLIX 2.0 posee las siguientes características, como indica la tabla 8.7.

Hardware	Software
<ul style="list-style-type: none"> <li>• Mainboard 845 intel</li> <li>• Procesador Pentium IV 2.6 GHZ</li> <li>• Ram 512 MB</li> <li>• Disco duro 160 GB</li> <li>• CD-Rom</li> <li>• Video, sonido.</li> <li>• Tarjeta de red Ethernet</li> </ul>	<ul style="list-style-type: none"> <li>• Debian Squeeze 6.0.1</li> <li>• Servidor web Apache</li> <li>• Aplicación DebianMirror</li> <li>• ssh openssh-server</li> <li>• firewall iptables</li> </ul>

**Tabla 8.7 Características del Servidor Unix 2.0**

La mejor manera de gestionar un servidor es con un acceso remoto. Pero, el acceso remoto debe ser hecho de forma segura, para garantizar que la comunicación no sea interceptada por terceros.

El servicio ssh, es un protocolo de comunicaciones que encripta los datos intercambiados, haciendo virtualmente imposible la violación de la privacidad de la comunicación. El acrónimo ssh viene del inglés: Secure SHell.

El protocolo ssh es muy versátil, cuenta con un software cliente que posibilita el acceso a la línea de comandos, la transferencia de archivos y la creación de túneles seguros con soportes de comunicaciones para otros protocolos.

Por lo general Debian ya viene con el servicio ssh instalado, pero no está demás realiza una comprobación, esto se hace con el siguiente comando.

**# apt-get install ssh openssh-server**



El servidor de UNLIX 2.0 usa como firewall los iptables los cuales viene integrado con el kernel, lo que se hace con un iptables es aplicar reglas. Se ejecuta el comando iptables, ahí se puede añadir, borrar, o crear reglas. Por ello un firewall de iptables no es sino un simple script de shell en el que se van ejecutando las reglas de firewall.

El servidor “unlixServer”, no posee reglas dentro de iptables, ya que la Unidad de Telecomunicaciones e Información posee un servidor propio con el firewall configurado. Por lo tanto no hay necesidad de crear las reglas de iptables dentro del servidor “unlixServer” porque posee una sola IP de la red.

Dicha máquina posee características básicas para configurarla como servidor, porque cuando se instala un sistema operativo GNU/LINUX como lo es Debian Squeeze no se necesita muchos requerimientos de hardware para ejecutar un servidor, debido a que la tecnología de GNU/LINUX optimiza bastante la memoria RAM de un computador. Por esta razón se ha escogido trabajar con Debian Squeeze como sistema operativo servidor de UNLIX 2.0, porque más allá de la potencia de hardware del servidor lo que se necesita es un buen ancho de banda de conexión (256 Kbps dedicado), el caso de la Universidad se tiene 50 Mbps sin segmentación, es decir que pueden llegar a una computadora final todo el ancho de banda, si alguien no está usando el internet desde otro punto de red. En el apartado 8.3.1 se explica paso a paso como se realiza la instalación de Debian Squeeze.

Cabe destacar que la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja, encabezada por el Lic. Jamil Ramón, permitió un espacio físico para poder alojar el servidor, además proporcionó una dirección IP (172.16.32.29), para la conexión en red del servidor. También se destaca la buena voluntad dada por los señores representantes del departamento de redes de la Unidad de Telecomunicaciones e Información que permitieron que la IP asignada pueda descargar sin restricciones los paquetes del repositorio de UNLIX 2.0. Cabe destacar que la dirección IP proporcionada no es pública, por lo que no se puede acceder desde fuera del campus de la Universidad Nacional de Loja, las razones por las que no brindaron la IP pública, es



porque no está dentro de la política del departamento de redes de la Unidad de Telecomunicaciones e Información.

Fue muy importante el apoyo dado por Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja, ya que el repositorio en su totalidad incluyendo las dos arquitecturas posee una capacidad de 80 GB aproximadamente, el amplio ancho de banda que se trabaja en la Unidad de Telecomunicaciones e Información hizo de que el número de días de la descarga del repositorio sea de 3 días aproximadamente. El disco duro del servidor es de 160 GB, capacidad suficiente para alojar el repositorio en su totalidad considerando las dos arquitecturas (i386 y amd64), además no se necesita otro disco duro, ya que este repositorio se va actualizando semanalmente y los paquetes de software viejos son reemplazados por los nuevos lo que significa que la cantidad de GB se mantiene.

### **8.2.2 Configuración previa a la Creación del Repositorio**

Para tener el repositorio de UNLIX 2.0, es haciendo un mirror de Debian Squeeze, teniendo el mirror creado los usuarios comunes puedan acceder vía http a la dirección de red del servidor (172.16.32.29) y actualizar cualquier paquete que necesiten. El mirror como es su nombre en inglés, en español es un espejo, es una réplica del repositorio oficial de Debian Squeeze, esta replica contiene todas las ramas creadas a imagen y semejanza del repositorio de Debian oficial lo cual significa que este mirror perfectamente puede ser actualizado con todos los paquetes nuevos que Debian Squeeze haya creado en su debido momento, de esta manera con conexión al repositorio espejo creado en el servidor local se pueda actualizar sin problemas desde cualquier punto de red dentro de la Universidad Nacional de Loja siempre y cuando se tenga todas las configuraciones correspondientes en cada computador (Distribuciones de UNLIX 2.0 ya las tiene).

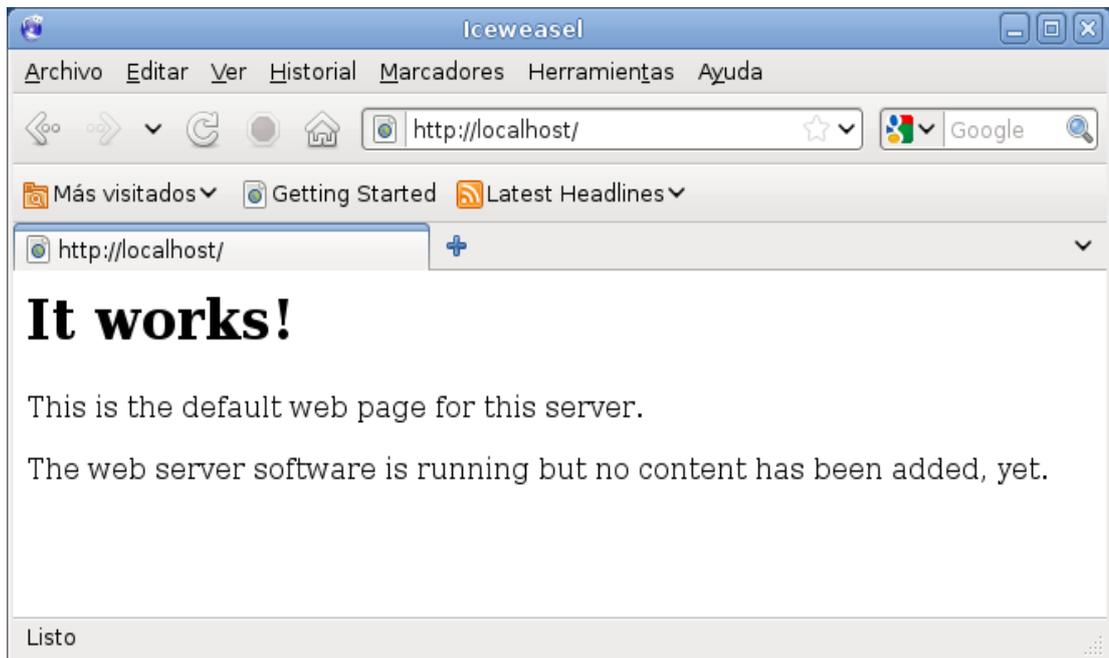


Para crear el repositorio se debe hacer lo siguiente:

1. Se debe verificar que se tenga instalado el servidor web apache, esto se lo hace abriendo una ventana del navegador de internet que en Debian Squeeze es Iceweasel (Navegador basado en Mozilla Firefox). Una vez dentro se escribe en la barra de navegación la palabra **localhost** y si se despliega una ventana de listo con la palabra **it works** que significa que apache está levantado normalmente sin problemas. La figura 8.2 muestra el servicio apache. Si se está trabajando en modo texto se puede ver el servicio levantado con siguiente comando:

```
# ps -eda
```

La figura 8.3 muestra el servicio apache levantado en el servidor



**Figura 8.2 Servidor Apache**



```
Terminal (como superusuario)
Archivo  Editar  Ver  Terminal  Ayuda
2398 ?      00:00:00 polkitd
2400 ?      00:00:00 upowerd
2401 ?      00:00:00 cron
2454 ?      00:00:00 cupsd
2479 ?      00:00:00 apache2
2480 ?      00:00:00 apache2
2481 ?      00:00:00 apache2
2482 ?      00:00:00 apache2
2483 ?      00:00:00 apache2
2741 ?      00:00:00 exim4
2752 ?      00:00:00 kerneloops
2765 ?      00:00:00 nmbd
2768 ?      00:00:00 smbd
```

**Figura 8.3 Servicio Apache**

2. Para saber con qué herramienta se puede trabajar en la creación del repositorio espejo se debe escribir en consola de root el siguiente comando

**# aptitude search mirror**

Aquí va a desplegar información de los diferentes paquetes derivados del mirror, en este caso se va a instalar el paquete debianmirror. La figura 8.4 muestra la salida de la ejecución del presente comando.

```
Terminal (como superusuario)
Archivo  Editar  Ver  Terminal  Ayuda
root@unixServer:/home/unix# aptitude search mirror
p  apt-mirror          - APT sources mirroring tool
p  debmirror           - Debian partial mirror script, with ftp and
p  debpartial-mirror   - tools to create partial Debian mirrors
p  ftpmirror           - Sincronización de jerarquías de directorio
p  liborlite-mirror-perl - ORLite extension to use remote SQLite data
p  libsvn-notify-mirror-perl - Keep a mirrored working copy of a reposito
p  mirror              - keeps FTP archives up-to-date
p  mirrormagic         - Dispare esquivando los obstáculos para rec
p  mirrormagic-data    - Archivos de datos de mirrormagic
root@unixServer:/home/unix#
```

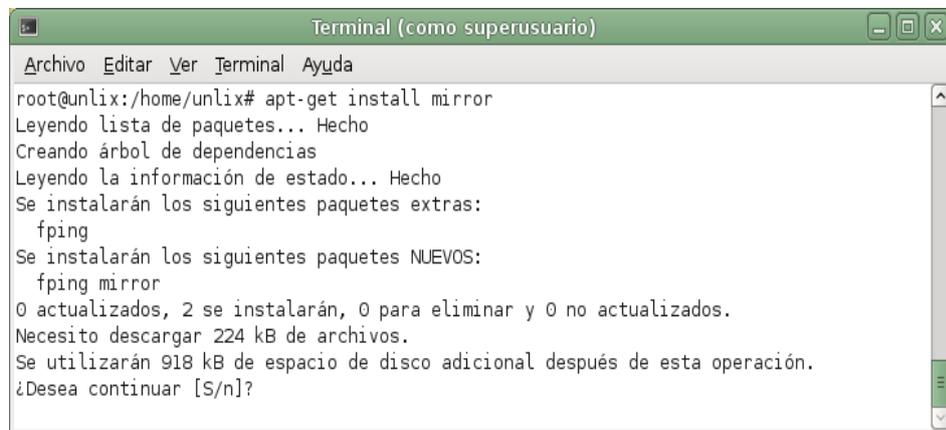
**Figura 8.4 Buscar el paquete mirror**



- Después de tener todo a punto con apache se debe instalar el paquete `debianmirror`, porque este paquete es el que va a gestionar la creación del repositorio espejo en el servidor local, para instalar este paquete se debe escribir el siguiente comando en consola de root

### # aptitude install debmirror

La figura 8.5 muestra la instalación del paquete `debianmirror`



```
Terminal (como superusuario)
Archivo Editar Ver Terminal Ayuda
root@unlix:/home/unlix# apt-get install mirror
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes extras:
  fping
Se instalarán los siguientes paquetes NUEVOS:
  fping mirror
0 actualizados, 2 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 224 kB de archivos.
Se utilizarán 918 kB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]?
```

**Figura 8.5 Instalación DebianMirror**

- Teniendo ya la herramienta instalada solo queda preparar el sistema para que aloje al repositorio como es el servidor apache entonces se debe crear una carpeta con el nombre `unlix2` en la dirección `/var/www/`. Esta carpeta almacenara al repositorio que se crea (multimedia, oficial, security y backports) y como está contenida en la carpeta `www` todos los clientes tendrán acceso a los datos de la carpeta `unlix2` y los directorios que se crean dentro.
- Con la carpeta creada para verificar que se tiene un acceso normal a la carpeta `unlix2` se debe poner en una ventana del navegador de internet la palabra **`http://localhost/unlix2`** y se despliega la siguiente información que presenta la figura 8.6 indicando de esta manera que se tiene total acceso de cliente servidor a la carpeta `unlix2`.



**Figura 8.6 Servidor Apache con el directorio Unix2**

Con estos pasos, está listo el servidor para ejecutar los scripts que se encuentran en el directorio /etc/scriptsUnix

Para la ejecución de los scripts se lo hace anteponiendo el comando:

**# sh nombredelscript.sh**

Y se ejecuta. Por ejemplo se ubica en el directorio donde se encuentra cualquier script para este caso el nombre “debianMirror.sh”, al mismo que se lo ejecuta de la siguiente manera.

**# sh debianMirror.sh**



### 8.2.3 Configuración de los Scripts para el repositorio

Una vez puesto en marcha el servidor, al mismo que se le asignó el nombre de “serverUnlix”, con las configuraciones básicas de Debian y conectado al internet, se procede con la establecer del repositorio, para llegar a esto se lo hace con la creación de scripts, los cuales se deben configurar para que se ejecuten y se cree el repositorio, este script prácticamente es el corazón del repositorio porque con estas líneas de código en bash se logra construir toda la estructura del repositorio para UNLIX 2.0.

Para trabajar con un script, se lo puede hacer usando cualquier editor de texto, propio de los sistemas GNU/LINUX, el editor que se usó fue el gedit y los scripts se los guarda con la extensión .sh.

Estos scripts que son para crear espejos o réplicas de repositorios, deben poseer algunos paramentos indispensables como por ejemplo:

- **debmirror** es la herramienta que se utiliza para la creación del repositorio.
- **debug** indica la ejecución del script.
- **verbose** tipo de respuesta en base a la lectura del script.
- **diff** expone la diferencia entre dos repositorios o entre un repositorio y la version anterior al mismo presentando los cambios realizados en sus paquetes.
- **host** desde donde se va a hacer la descarga del repositorio es aquí donde se debe poner la dirección de repo oficial.
- **method** es el modo de descarga e intercambio de paquetes en este caso http.
- **root** se le debe poner Debian ya que es la raíz del sistema.
- **dist** se pone la distribución que se desea replicar. Ejemplo Squeeze
- **arch** se coloca la arquitectura para la cual va dirigido el repositorio en este caso es i386, amd64.
- **section** se escribe las ramas que se desea crear en este caso son main contrib non-free.



- **ignore** va por defecto para ignorar ciertos errores de descargas que puedan generarse.
- Y la dirección donde se va a guardar el repositorio en el servidor local para este caso es **`/var/www/unlix2/oficial/`**

El repositorio de UNLIX 2.0 es una mirror del repositorio de Debian, su estructura es similar en tamaño y forma. Debido a que el repositorio UNLIX 2.0 se forma de varios pequeños repositorios o subrepositorios que tienen o cumplen una función específica dentro de la aplicación global ya sea como un mecanismo de actualización de la distribución en este caso UNLIX 2.0 GNU/LINUX o como almacén de datos para contener paquetes específicos que no pueden estar contenidos o mezclados entre sí, se tiene un conjunto de repositorios como: Oficial, Seguridad o Security, Backports y Multimedia.

El Repositorio Oficial es el conjunto de paquetes fuentes y binarios que son oficiales dentro de la rama estable de Debian Squeeze eso se indica para binarios se tiene el **.deb** y para fuentes **deb-src**, esto puesto en el archivo sources.list dice que se va a alojar los dos tipos de datos fuentes y binarios en el repositorio oficial y se lo va a hacer con el protocolo http apuntando a un servidor en específico en este caso apunta al servidor ftp.us.debian.org lo que permite obtener todos los paquetes que contienen este servidor para replicarlo en el servidor local creado para UNLIX 2.0 y de esa manera conseguir tener a disposición los programas oficiales de la rama estable de Debian. El script para este repositorio queda de la siguiente manera: (ver tabla 8.8).



dMirror.sh
<pre>#!/bin/bash  ## Debian Mirror  debmirror --debug --progress --verbose --diff=none --host=ftp.us.debian.org -- root=debian --method=http --dist=squeeze --arch=i386,amd64 -- section=main,contrib,non-free --ignore-release-gpg /var/www/unlix2/oficial</pre>

**Tabla 8.8 Script Repositorio Oficial**

El repositorio de Seguridad o Security, contiene software de verificación para el repositorio oficial, en caso de que falle algún paquete en el sistema este se reemplaza por un paquete del repositorio de seguridad hasta que se solucione el inconveniente y luego si retome el paquete del repositorio oficial y entonces siga funcionando como antes todo el sistema para el cual se ha creado el paquete. Su script es: (ver tabla 8.9)

dSecurityMirror.sh
<pre>#!/bin/bash  ## Debian Security  debmirror --debug --progress --verbose --diff=none --host=security.debian.org -- root=/ --method=http --dist=squeeze/updates --arch=i386,amd64 -- section=main,contrib,non-free --ignore-release-gpg /var/www/unlix2/security</pre>

**Tabla 8.9 Script Repositorio Security**

Los Repositorio backports, almacena los paquetes que son más actuales que no están presentes en la rama estable pero tienen un inconveniente que si se los usa no se va a poder actualizar a una versión de la rama estable porque puede que de error de dependencia ya que se está utilizando otra manera de indexación de paquetes para poder utilizar los programas más actuales dentro de la distribución que se está ocupando, así como son útiles para el momento pero para futuro no es aconsejable utilizarlo porque es mejor esperar que la última versión del programa requerido llegue directamente desde el



repositorio oficial que tiene todo configurado para que sea utilizado como un paquete oficial y no desde el repositorio backports. Su script es el siguiente: (ver tabla 8.10).

dBackportsMirror.sh
<pre>#!/bin/bash  ## Debian Backports debmirror --debug --progress --verbose --diff=none -- host=backports.debian.org/debian-backports --root=/ --method=http -- dist=squeeze-backports --arch=i386,amd64 --section=main,contrib,non-free -- ignore-release-gpg /var/www/unlix2/backports</pre>

**Tabla 8.10 Script Repositorio Backports**

Y el repositorio Multimedia, es un repositorio no oficial, que contiene software que no es en su mayoría totalmente libre es decir que no cumple con la filosofía de software libre es decir es privativo pero que se puede utilizar dentro de los sistemas GNU/LINUX o son software que no son totalmente libres de acuerdo a las normas DFSG Debian Free Software Guidelines que es la licencia libre de Debian. También este repositorio tiene software comercial Multimedia de ahí su nombre pero que puede ser usado en la distribución estable de Debian como audacious2 entre otros, audacious2 es un atractivo reproductor de música. (Ver tabla 8.11)

dMultimediaMirror.sh
<pre>#!/bin/bash  ## Debian Multimedia debmirror --debug --progress --verbose --diff=none --host=www.debian- multimedia.org --root=/ --method=http --dist=squeeze --arch=i386,amd64 -- section=main --ignore-release-gpg /var/www/unlix2/multimedia</pre>

**Tabla 8.11 Script Repositorio Multimedia**



Los scripts están listos para ser ejecutados, pero antes se los deben guardar en una carpeta que puede llevar cualquier nombre, para este proyecto la carpeta se llama “scriptsUnlix” que se crea en el directorio `/etc/`, ahí se guardan los scripts, en total se va a trabajar con cuatro scripts, un script para el repositorio oficial, para el repositorio multimedia, para el repositorio security y para el repositorio backports, todos llevan la extensión `.sh`.

De esta manera quedan listos los scripts para ejecutarlos y se empieza con la descarga de cada uno de ellos y estos se vayan almacenando en el servidor.

#### **8.2.4 Ejecución de los scripts y Creación del repositorio**

Configurado el server o servidor y teniendo los scripts, se procede con la ejecución de los mismos, para ello se trabaja mediante el servicio ssh o usando el programa putty para acceder al servidor desde cualquier punto de red del campus de la Universidad Nacional de Loja, para lo cual, se tiene que ingresar a la consola y en modo root o súper usuario, al directorio `/etc/scriptUnlix/` y se procede a ejecutar cada uno de los cuatro scripts: `dMirror.sh`, `dMultimediaMirror.sh`, `dSecurityMirror.sh` y `dBackportsMirror.sh`

Cabe señalar que para la descarga de repositorio Oficial demora alrededor de 2 días (80 GB aproximadamente), mientras que los otros no demora más de una hora (2 GB aproximadamente de cada uno). La velocidad de transferencia de datos es de 0.5 Mbps aproximadamente, cabe destacar que es un conexión rápida, teniendo en cuenta que se tiene 50 Mbps de ancho de banda dentro de la Universidad. Los siguientes gráficos 8.7, 8.8, 8.9 y 8.10 muestran cómo se procede la descarga de los cuatro repositorios que va a usar UNLIX 2.0



```
Terminal (como superusuario)
Archivo Editar Ver Terminal Ayuda
99%] Getting: dists/squeeze/contrib/binary-amd64/Release... 200 OK
ok
dists/squeeze/non-free/binary-i386/Packages.gz needs fetch
http://ftp.us.debian.org/debian/dists/squeeze/non-free/binary-i386/Packages.gz =
> [ 99%] Getting: dists/squeeze/non-free/binary-i386/Packages.gz... 200 OK
ok
dists/squeeze/non-free/binary-i386/Release needs fetch
http://ftp.us.debian.org/debian/dists/squeeze/non-free/binary-i386/Release => [
99%] Getting: dists/squeeze/non-free/binary-i386/Release... 200 OK
ok
dists/squeeze/non-free/binary-amd64/Packages.gz needs fetch
http://ftp.us.debian.org/debian/dists/squeeze/non-free/binary-amd64/Packages.gz
=> [ 99%] Getting: dists/squeeze/non-free/binary-amd64/Packages.gz... 200 OK
ok
dists/squeeze/non-free/binary-amd64/Release needs fetch
http://ftp.us.debian.org/debian/dists/squeeze/non-free/binary-amd64/Release => [
100%] Getting: dists/squeeze/non-free/binary-amd64/Release... 200 OK
ok
Parse Packages and Sources files and add to the file list everything therein.
Download all files that we need to get (49066 MiB).
Get package files.
http://ftp.us.debian.org/debian/pool/contrib/c/cyadela/cyadela-data_1.0.1-1_all
l.deb => [ 0%] Getting: pool/contrib/c/cyadela/cyadela-data_1.0.1-1_all.deb..
.
```

Figura 8.7 Descargando del repositorio Oficial

```
Terminal (como superusuario)
Archivo Editar Ver Terminal Ayuda
200 OK
ok
Parsing Packages and Sources files ...
Files to download: 336 MiB
/usr/bin/debmirror: Created directory: pool/
/usr/bin/debmirror: Created directory: pool/main/
/usr/bin/debmirror: Created directory: pool/main/a/
/usr/bin/debmirror: Created directory: pool/main/a/amarok/
http://backports.debian.org/debian-backports//pool/main/a/amarok/amarok-common_2
.4.0-1~bpo60+1_all.deb => [ 0%] Getting: pool/main/a/amarok/amarok-common_2.4.0
-1~bpo60+1_all.deb *** GET http://backports.debian.org/debian-backports//p
ool/main/a/amarok/amarok-common_2.4.0-1~bpo60+1_all.deb ==> 200 OK (11s)
200 OK
ok
http://backports.debian.org/debian-backports//pool/main/a/amarok/amarok-dbg_2.4.
0-1~bpo60+1_amd64.deb => [ 3%] Getting: pool/main/a/amarok/amarok-dbg_2.4.0-1~b
po60+1_amd64.deb *** GET http://backports.debian.org/debian-backports//p
ool/main/a/amarok/amarok-dbg_2.4.0-1~bpo60+1_amd64.deb ==> 200 OK (38s)
200 OK
ok
http://backports.debian.org/debian-backports//pool/main/a/amarok/amarok-dbg_2.4.
0-1~bpo60+1_i386.deb => [ 18%] Getting: pool/main/a/amarok/amarok-dbg_2.4.0-1~bp
o60+1_i386.deb *** GET http://backports.debian.org/debian-backports//pool/main
/a/amarok/amarok-dbg_2.4.0-1~bpo60+1_i386.deb ==> 1%
```

Figura 8.8 Descargando del repositorio Backports



```
Terminal (como superusuario)
Archivo Editar Ver Terminal Ayuda
ok
dists/squeeze/main/binary-amd64/Packages.gz needs fetch
http://www.debian-multimedia.org//dists/squeeze/main/binary-amd64/Packages.gz =>
[ 51%] Getting: dists/squeeze/main/binary-amd64/Packages.gz... 200 OK
ok
Parse Packages and Sources files and add to the file list everything therein.
Download all files that we need to get (624 MiB).
Get package files.
http://www.debian-multimedia.org//pool/main/b/blender/blender_2.49.2-dfsg-2.1_i386.deb => [ 0%] Getting: pool/main/b/blender/blender_2.49.2-dfsg-2.1_i386.deb...
. 200 OK
ok
/usr/bin/debmirror: Created directory: pool/main/b/bombono-dvd/
http://www.debian-multimedia.org//pool/main/b/bombono-dvd/bombono-dvd-data_1.0.0-0.0squeezel_all.deb => [ 2%] Getting: pool/main/b/bombono-dvd/bombono-dvd-data_1.0.0-0.0squeezel_all.deb... 200 OK
ok
http://www.debian-multimedia.org//pool/main/b/bombono-dvd/bombono-dvd_1.0.0-0.0squeezel_amd64.deb => [ 2%] Getting: pool/main/b/bombono-dvd/bombono-dvd_1.0.0-0.0squeezel_amd64.deb... 200 OK
ok
http://www.debian-multimedia.org//pool/main/b/bombono-dvd/bombono-dvd_1.0.0-0.0squeezel_i386.deb => [ 2%] Getting: pool/main/b/bombono-dvd/bombono-dvd_1.0.0-0.0squeezel_i386.deb...
```

Figura 8.9 Descargando del repositorio Multimedia

```
Terminal (como superusuario)
Archivo Editar Ver Terminal Ayuda
ok
dists/squeeze/updates/contrib/binary-i386/Release needs fetch
http://security.debian.org//dists/squeeze/updates/contrib/binary-i386/Release =>
[100%] Getting: dists/squeeze/updates/contrib/binary-i386/Release... 200 OK
ok
dists/squeeze/updates/contrib/binary-amd64/Release needs fetch
http://security.debian.org//dists/squeeze/updates/contrib/binary-amd64/Release =>
[100%] Getting: dists/squeeze/updates/contrib/binary-amd64/Release... 200 OK
ok
dists/squeeze/updates/non-free/binary-i386/Release needs fetch
http://security.debian.org//dists/squeeze/updates/non-free/binary-i386/Release =>
[100%] Getting: dists/squeeze/updates/non-free/binary-i386/Release... 200 OK
ok
dists/squeeze/updates/non-free/binary-amd64/Release needs fetch
http://security.debian.org//dists/squeeze/updates/non-free/binary-amd64/Release =>
[100%] Getting: dists/squeeze/updates/non-free/binary-amd64/Release... 200 OK
ok
Parse Packages and Sources files and add to the file list everything therein.
Download all files that we need to get (2125 MiB).
Get package files.
/usr/bin/debmirror: Created directory: pool/updates/main/a/apache2/
http://security.debian.org//pool/updates/main/a/apache2/apache2-dbg_2.2.16-6+squeezel_amd64.deb => [ 0%] Getting: pool/updates/main/a/apache2/apache2-dbg_2.2.16-6+squeezel_amd64.deb...
```

Figura 8.10 Descargando del repositorio Security

A continuación la figura 8.11 muestra la transferencia de paquetes desde el servidor oficial de Debian al Servidor local unixServer. La velocidad es de aproximadamente 500 Kbps.



Terminal (como superusuario)						
Archivo Editar Ver Terminal Ayuda						
	1.91Mb	3.81Mb	5.72Mb	7.63Mb	9.54Mb	
unixServer.local	=>	unixServer-2.local	2.33Kb	2.44Kb	2.52Kb	
	<=		208b	208b	208b	
255.255.255.255	=>	*	0b	0b	0b	
	<=		2.59Kb	2.33Kb	2.75Kb	
255.255.255.255	=>	172.16.60.147	0b	0b	0b	
	<=		3.66Kb	1.47Kb	1.83Kb	
255.255.255.255	=>	172.16.60.58	0b	0b	0b	
	<=		3.66Kb	1.47Kb	1.10Kb	
169.254.255.255	=>	Usuario-PC-3.local	0b	0b	0b	
	<=		936b	1.10Kb	858b	
169.254.255.255	=>	169.254.86.120	0b	0b	0b	
	<=		0b	1.08Kb	277b	
172.16.63.255	=>	172.16.34.214	0b	0b	0b	
	<=		624b	1.02Kb	262b	
224.0.0.251	=>	169.254.80.91	0b	0b	0b	
	<=		964b	1.01Kb	259b	
169.254.255.255	=>	169.254.95.137	0b	0b	0b	
	<=		624b	989b	247b	
TX: cumm: 495KB peak: 4.14Kb rates: 3.17Kb 3.39Kb 3.39Kb						
RX: 21.3MB 61.5Kb 44.3Kb 37.8Kb 40.5Kb						
TOTAL: 21.8MB 65.0Kb 47.5Kb 41.2Kb 43.9Kb						

**Figura 8.11 Verificando la descarga del repositorio**

Una vez que se han descargado los diferentes repositorios, se procede a verificar los paquetes desde el servidor web apache, para lo cual se abre la ventana de un navegador y se coloca la dirección de red del servidor <http://172.16.32.29/unlix2/> y el nombre de la carpeta unlix2, se aclara se está trabajando desde un servidor local. La figura 8.12 muestra que se han generado los repositorios



Figura 8.12 Estructura de repositorio unix2

Importante señalar que dentro de cada directorio creado se genera las carpetas o directorios contrib, main y non-free, tal como se muestra en la figura 8.13.



Figura 8.13 Directorio base del repositorio



La siguiente figura 8.14 muestra cómo se va generando los paquetes .deb y sus dependencias, algo importante de la creación de los repositorios, es que los paquetes se van generando en orden alfabético, es decir los programas que empiezan con la letra a lo crea el directorio a y así sucesivamente.

Name	Last modified	Size	Description
Parent Directory	-	-	-
<a href="#">alien-arena-common_7.40-2_amd64.deb</a>	27-Jul-2010 11:32	201K	
<a href="#">alien-arena-common_7.40-2_i386.deb</a>	27-Jul-2010 14:02	183K	
<a href="#">alien-arena-server_7.40-2_amd64.deb</a>	27-Jul-2010 11:32	145K	
<a href="#">alien-arena-server_7.40-2_i386.deb</a>	27-Jul-2010 14:02	131K	
<a href="#">alien-arena_7.40-2_amd64.deb</a>	27-Jul-2010 11:32	506K	
<a href="#">alien-arena_7.40-2_i386.deb</a>	27-Jul-2010 14:02	453K	

Figura 8.14 Creación de los paquetes del repositorio

### 8.2.5 Actualización automática del repositorio

Debido a que un repositorio contiene muchos programas o paquetes conformando su estructura esto hace que sea un almacén de datos muy extenso y por ende necesita de actualización cuando ya salen nuevas versiones de los programas contenidos dentro del repositorio. Si el repositorio de UNLIX 2.0 fuera creado solamente con el apoyo de desarrolladores la actualización sería muy difícil pero como el repositorio de UNLIX 2.0 es una réplica del repositorio de Debian Squeeze la actualización se convierte en una tarea sencilla que consiste en sacar una réplica de las actualizaciones que va teniendo el repositorio oficial de Debian, para ellos se necesita programarse en el demonio **cron**.

Para realizar esta tarea se debe ejecutar el siguiente comando en consola de root

```
# crontab -e
```



Escribiendo este comando se despliega la siguiente pantalla de edición. Para trabajar con el cron se debe considerar lo siguiente:

- El símbolo almohadilla «#» es un comentario, todo lo que se encuentre después de ese carácter no será ejecutado por *cron*.
- El momento de ejecución se especifica de acuerdo con la siguiente tabla 8.12.

1	Minutos: (0-59)
2	Horas: (0-23)
3	Días: (1-31)
4	Mes: (1-12)
5	Día de la semana: (0-6), siendo 1=lunes, 2=martes,... 6=sábado y 0=domingo (a veces también 7=domingo)

**Tabla 8.12 Característica del crontab**

- Nota si se quiere por ejemplo que el scrip se ejecute todos los meses del año se escribe \*, que quiere decir todos.
- Por ejemplo: **01 22 \* \* 1 sh /etc/scriptUnix/dMirror.sh**, dice que el script se va a ejecutar todos los lunes a las 22:01 horas.

El cron para todos los scripts en el servidorUnix quedará como indica la siguiente figura 8.15.



```

Terminal (como superusuario)
Archivo Editar Ver Terminal Ayuda
GNU nano 2.2.4 Fichero: /tmp/crontab.ho3ui2/crontab

# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
01 22 * * 1 sh /etc/scriptUnix/dMirror.sh
01 22 * * 2 sh /etc/scriptUnix/dSecurityMirror.sh
01 22 * * 3 sh /etc/scriptUnix/dMultimediaMirror.sh
01 22 * * 4 sh /etc/scriptUnix/dBackportsMirror.sh

^G Ver ayuda ^O Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTxt ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^L PegarTxt ^T Ortografía
  
```

**Figura 8.15 Crontab de los scripts**

La tabla 8.13 muestra que los scripts se van a ejecutar de la siguiente forma:

Día	Hora	Comando	Script
Todos Lunes	22:01	01 22 * * 1	dMirror.sh
Todos Martes	22:01	01 22 * * 2	dSecurityMirror.sh
Todos Miércoles	22:01	01 22 * * 3	dMultimediaMirror.sh
Todos Jueves	22:01	01 22 * * 4	dBackportsMirror.sh

**Tabla 8.13 Programación del crontab**

Este análisis se lo hizo considerando que son 4 scripts y estos necesitan de tiempo para la actualización de su repositorio, por ello se vio en la necesidad que dejarlos que se actualicen cada día, ya que no se recomienda hacer la actualización de dos scripts al mismo tiempo, debido a que la descarga de cada script es de forma individual, y de esta manera no se satura el ancho de banda.



### 8.3 ACTUALIZACIÓN DE UNLIX 2.0

La primera versión de UNLIX fue creada justamente para satisfacer necesidades de docentes y alumnos de la Universidad Nacional de Loja en especial orientado al Área de la Energía, las Industrias y los Recursos Naturales no Renovables, es una versión bastante buena y confiable, su uso se extendió hasta en la comunidad local pero ahora la nueva versión mejorada y actualizada denominada “UNLIX 2.0”, es un sistema basado en la versión estable de Debian Squeeze 6.0.1 con un kernel 2.6.32, donde se incorpora ciertas actualizaciones de seguridad y de compatibilidad del sistema.

Es por esto, que UNLIX 2.0 a más de tener la estructura de Debian Squeeze, tiene el adicional de que posee paquetes actualizados (jdk, mysql, postgresql, entre otros) con ciertas mejoras en sus versiones, ya que la primera versión de UNLIX (creada en el año 2008) posee paquetes propios del periodo 2008, además UNLIX en su primera versión fue hecha basándose en Debian etch (versión estable de Debian anterior a Lenny). Por lo tanto UNLIX 2.0, fue pensado para que este incorpore en sus paquetes principales las últimas actualizaciones de los mismos, es decir versiones más recientes. Por ejemplo la tabla 8.14 hace una comparación de los principales paquetes de las versiones de UNLIX.



Paquete	UNLIX 1.0	UNLIX 2.0
Apache	1.3.34	2.2.16
Gimp	2.2.13	2.6.11
MySQL	5.0.32	5.1.49
Php	5.2.0	5.3.3
PostgreSQL	8.1.15	8.4.6
Python	2.4.4	2.6.6
jdk	5.0.0	1.6.22
OpenOffice	2.0.4	3.2.1
K3b	0.12.17	2.0.1
KoloarPaint	1.4.5	4.4.5
Dia	0.95	0.97.1
Planner	0.14.2	0.14.4
MonoDevelop	0.11	2.4
Eclipse	3.2.1	3.5.2
Netbeans	6.0.1	7.0
Blender	2.42	2.49
Qcad	2.0.5	2.0.5
IceWeasel	2.0	3.5.16
Grass	1.6.0.2	1.6.1.20
Tomcat	5.5.20	6.0.18

**Tabla 8.14 Comparación en paquetes Unlix 1.0 y Unlix 2.0**

Unlix 2.0, además incorpora:

- kernel 2.6.32-5
- entorno de escritorio GNOME 2.30
- entorno de escritorio Xfce 4.6
- entorno de escritorio LXDE 0.5.0



UNLIX 2.0 está basado en Debian Squeeze 6.0.1, porque es la última versión estable lanzada el 22 de marzo del 2011 en la página oficial de Debian <http://www.debian.org/>, por esta razón se eligió la presente versión.

Es conveniente tomar la última versión, porque siempre se corrigen bugs que otras versiones anteriores no los corrigen, por eso se ha basado UNLIX 2.0 en esta versión, teniendo en cuenta este antecedente, se puede decir que UNLIX 2.0, es un sistema robusto seguro confiable igual que Debian Squeeze y con las mejoras adicionales, que son el contar con los paquetes a las medidas y necesidades de los docentes y alumnos del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables de la Universidad Nacional de Loja, en especial.

Algo que se debe destacar es que UNLIX en su primera versión considera ciertos paquetes base como se explica en el apartado 8.1.8 Los paquetes de UNLIX del presente documento, además posee tres proyectos de tesis como son: Sistema de Fondo de Jubilación, Sistema de Gestión Administrativa y Sistema de Generación de Herramientas multimedia. Para la versión de UNLIX 2.0 GNU/LINUX, estos sistemas no están considerados debido a que la idea central de UNLIX 2.0 es para que sea usado por toda la Universidad Nacional de Loja y usuarios en general y no solamente por los usuarios del Área de la Energía, las Industrias y los de los Recursos Naturales no Renovables.

Además, se suprimieron paquetes como el GnomeBaker por un paquete para grabar Cd/Dvds por el k3B que hace lo mismo, así como el Juk paquete para escuchar archivos de audio por el Rhythmbox que hace exactamente lo mismo, se eliminó el paquete Firefox que es un navegador web y este fue reemplazado por el Iceweasel que un paquete de software que usa los mismos componentes del Firefox y se quitó Nvu el cual fue reemplazado por el Kompozer que es una aplicación para desarrollo web.



### 8.3.1 Partiendo de la versión de Debian Squeeze 6.0.1

En Debian Squeeze las configuraciones y uso del sistema en sí, es similar a cualquier sistema basado en GNU/LINUX, se conoce que Debian es el padre de los sistemas GNU/LINUX, por ejemplo Ubuntu, Kanoplix, Mint, entre otros

UNLIX en su primera versión y en la versión 2.0, se considera sistema GNU/LINUX, derivado de Debian, por lo que su funcionamiento y configuración es similar a cualquier sistema GNU/LINUX.

Cuando se trabaja con UNLIX, el ambiente es parecido al de Debian, por esta razón para crear la versión UNLIX 2.0, se tomó en cuenta el hacer ciertas configuraciones necesarias al Debian base, con el objetivo de que el resultado sea que UNLIX 2.0 sea lo mas estable posible. Para ello se partió desde la instalación de Debian Squeeze 6.0.1<sup>a</sup> (Ver capítulo 2 Instalación de Debian, manual del administrador)

Una vez terminada la instalación de Debian, se tiene el sistema operativo funcionando con sus múltiples opciones. En el caso de tener otro sistema operativo (Windows, Ubuntu) en el computador, se necesita actualizar el GRUB del sistema, para ello se debe poner el siguiente comando en la consola de root que es:

```
# update-grub2
```

Con esto se tiene el nuevo sistema Debian Squeeze instalado y el reconocimiento de Windows o Ubuntu en el GRUB, para que se pueda acceder a los mismos en el caso de que el usuario lo requiera.

Con Debian Squeeze en el computador, queda la opción de configurarlo a gusto y así comenzar a construir la distribución de UNLIX 2.0 GNU/LINUX. Algo que se recomienda es que la instalación se la haga con conexión a Internet, debido a que Debian desde la instalación actualiza los paquetes más recientes que tiene en su



repositorio.

Ya trabajando con Debian, se considera importante realizar ciertas configuraciones adicionales, como el editar el archivo **/etc/apt/sources.list** que es donde se almacena las direcciones de los repositorios de descarga de los paquetes de Debian Squeeze para las diferentes actualizaciones.

Para editar el archivo **/etc/apt/sources.list**, se debe seguir los siguientes pasos, se abre una consola de root como administrador y se coloca el siguiente comando:

**# gedit /etc/apt/sources.list**

Este comando lo que hace es abrir el editor gedit, pero si alguien trabaja con otros editores puede usar el vi, vim, nano, en este caso se ha usado gedit para realizar la respectiva configuración del texto. Después de esto, es importante configurar los repositorios en el sources.list.

La siguiente tabla 8.15 muestra el contenido del archivo sources.list



sources.list
<pre># Repositorios Servidor UNLIX 2.0 # Repositorio DVD # deb cdrom:[Debian GNU/Linux 6.0.1a _Squeeze_ - Official i386 DVD Binary-1 20110205-17:27]/ squeeze contrib main #Repositorios oficiales deb http://172.16.32.29/unlix2/ squeeze main contrib non-free deb-src http://172.16.32.29/unlix2/ squeeze main contrib non-free # Repositorios de seguridad deb http://172.16.32.29/unlix2/security/ squeeze/updates main contrib non-free deb-src http://172.16.32.29/unlix2/security/ squeeze/updates main contrib non-free # Repositorios multimedia deb http://172.16.32.29/unlix2/multimedia/ squeeze main non-free deb-src http://172.16.32.29/unlix2/multimedia/ squeeze main non-free # Repositorios Backports deb http://172.16.32.29/unlix2/backports/ squeeze-backports main contrib non-free deb-src http://172.16.32.29/unlix2/backports/ squeeze-backports main contrib non-free</pre>

**Tabla 8.15 sources.list Unlix 2.0**

Esto se lo hace si se encuentra dentro de la red del campus de la Universidad Nacional de Loja. Pero la idea de UNLIX 2.0 GNU/LINUX es que sea usado por cualquier usuario, es decir alguien que por diferentes motivos no pueda trabajar con UNLIX 2.0 desde la Universidad Nacional de Loja, para este caso se puede añadir al `/etc/apt/sources.list` las direcciones del repositorio oficial y comentar las que apuntan al `http://172.16.32.29/unlix2`, por ejemplo: (ver tabla 8.16).



sources.list
<pre># Repositorios Oficiales de debian # Repositorio DVD # deb cdrom:[Debian GNU/Linux 6.0.1a _Squeeze_ - Official i386 DVD Binary-1 20110205-17:27]/ squeeze contrib main # Repositorios oficiales deb http://http.us.debian.org/debian squeeze main contrib non-free deb-src http://http.us.debian.org/debian squeeze main contrib non-free # Repositorios seguridad deb http://security.debian.org/ squeeze/updates main contrib non-free deb-src http://security.debian.org/ squeeze/updates main contrib non-free # Repositorios multimedia deb http://www.debian-multimedia.org/ squeeze main non-free deb-src http://www.debian-multimedia.org/ squeeze main non-free # Repositorios Backports deb http://backports.debian.org/debian-backports squeeze-backports main contrib non-free deb-src http://backports.debian.org/debian-backports squeeze-backports main contrib non-free</pre>

**Tabla 8.16 sources.list Repositorio Oficial**

Los repositorios oficiales sirven para bajar todos los programas o paquetes que estén disponibles para la instalación en Debian estable que es Squeeze.

Los repositorios de seguridad son esenciales a la hora de descargar paquetes para que se descarguen en forma segura y confiable.

Los repositorios multimedia son necesarios para bajar paquetes que sean multimedia, para utilizar correctamente este repositorio es necesario instalarle la clave de acceso.



Los repositorios backports contienen paquetes de la versión de prueba de Debian, pero que están testeados para ser instalados en la versión estable de Squeeze, obteniendo así una mejor estabilidad si se instala paquetes que todavía están en la versión testing de Debian, por ejemplo si se quiere la última versión de postgresql 9 se la puede encontrar para Debian Squeeze que es estable, en los repositorios oficiales de backports de Squeeze y desde ahí se puede instalar postgresql 9 o postgresql en su última versión y el mismo proceso se lo puede realizar con los demás paquetes que se desee actualizar.

El siguiente paso es poner el comando `apt-get update` y va a salir un error del repositorio de multimedia, entonces se debe poner la llave del repositorio multimedia que se lo hace con el siguiente comando:

```
# apt-get install debian-multimedia-keyring
```

Luego para actualizar los repositorios se escribe en consola el comando:

```
# apt-get update
```

Se espera que se corra el proceso, el mismo que a demorar algunos minutos según el tráfico de red que esté en ese momento.

Luego teniendo listo todo, se pone el siguiente comando en consola para que se refresque la distribución con los últimos paquetes y actualizaciones disponibles, los comandos son:

```
# apt-get dist-upgrade
```

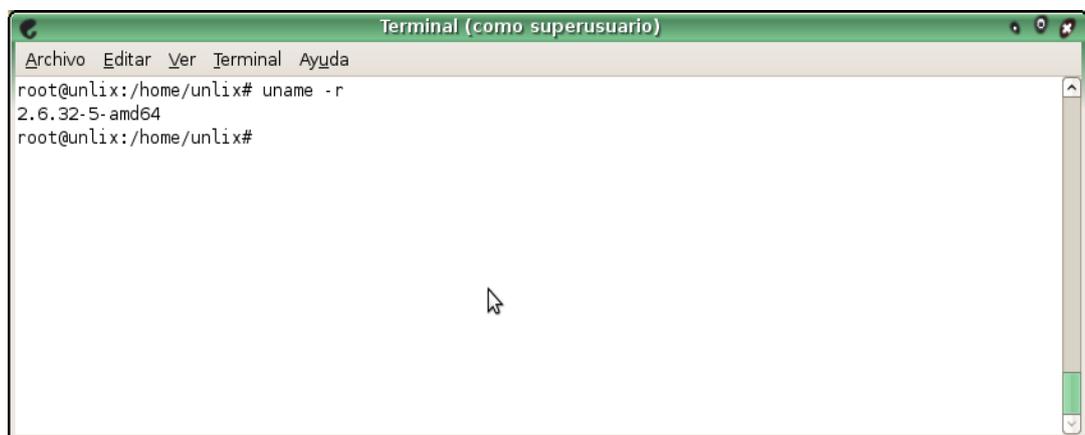
Actualizado y listo para empezar a personalizar e instalar paquetes necesarios en la versión de Debian básica.



### 8.3.2 Configuración básica de la versión UNLIX 2.0

UNLIX en su primera versión, se tomaba como base a Debían etch pero ahora para esta nueva versión de UNLIX 2.0, se toma como base a Debian Squeeze, que es la nueva y actualizada versión estable de Debian hasta febrero del 2011, por esta razón se utiliza Debian Squeeze como sistema base de desarrollo, y así construir UNLIX 2.0, luego de la instalación del sistema base, se debe personalizar todo para que quede listo y así tener la distribución de UNLIX 2.0

Pero antes de remasterizar la distribución (distro), hay que tener otras cosas muy en cuenta como por ejemplo, que tipo de kernel se tiene instalado, si los controladores funcionan con este kernel, si el kernel tiene los modulo necesarios para ser remasterizado. Bueno para conocer con que kernel se está trabajando, se escribe desde la consola el siguiente comando (ver figura 8.16).



```
Terminal (como superusuario)
Archivo Editar Ver Terminal Ayuda
root@unlix:/home/unlix# uname -r
2.6.32-5-amd64
root@unlix:/home/unlix#
```

**Figura 8.16 Pantalla verificación del kernel de debian**

La figura 8.43 muestra el kernel para un computador de 64 bits, porque se trabaja con ese kernel, sale con la extensión “**2.6.32-5-amd64**” propio para este tipo de computadoras (aunque si funciona con el kernel 2.6.32-5-i386). Por lo general el kernel que viene en el DVD de Debian, si posee los módulos necesarios para crear alguna distribución, el problema puede generarse si se utiliza algún otro kernel superior al que viene en el DVD installer de Debian Squeeze 6.0.1



### 8.3.3 Instalación y actualización de las aplicaciones para UNLIX 2.0

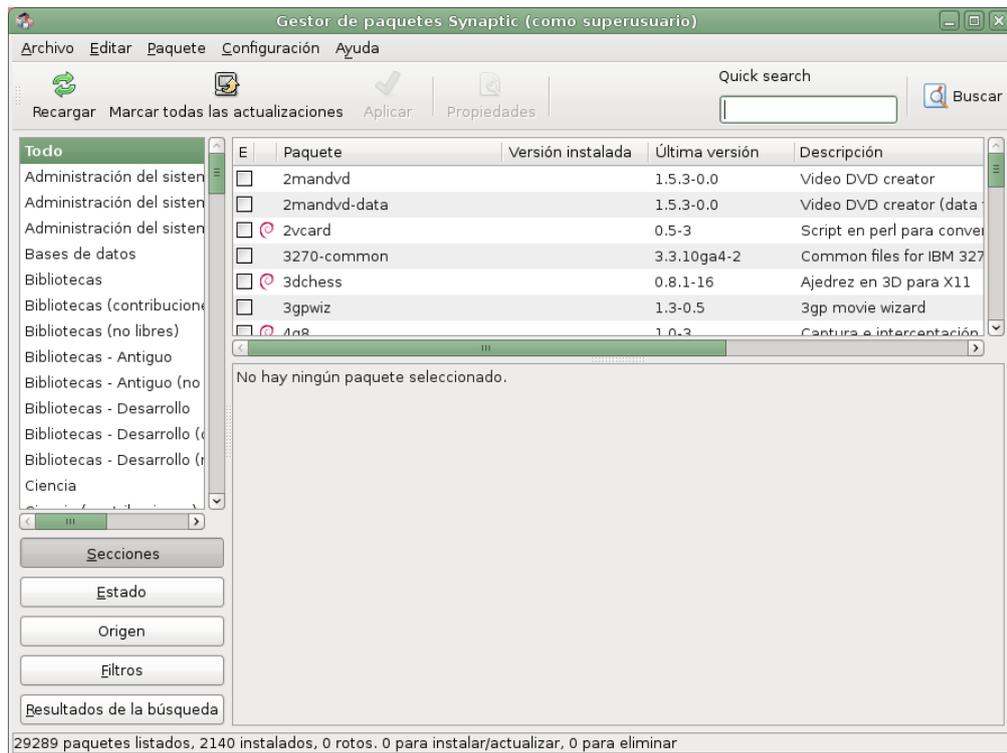
Una vez configurado los respectivos repositorios de Debian y estos actualizados desde los orígenes, se procede a la instalación y actualización de las aplicaciones propias de UNLIX 2.0. Como la versión estable de Debian Squeeze 6.0.1 (última hasta el mes de febrero de 2011), se procede instalar las aplicaciones, algunas de ellas están en el mismo DVD como por ejemplo el paquete `mysql`, también se puede encontrar ciertas aplicaciones desde el repositorio de Debian (en la web), esto sucede cuando no se encuentra algún paquete en el DVD y el sistema lo busca en la red, este proceso se lo hace con el gestor de paquetes Synaptic o desde la consola de root usando el `apt-get`, por ejemplo: **`apt-get install mysql-server`**. Cabe destacar que el archivo `/etc/apt/sources.list` está configurado para que las actualizaciones se las haga desde el servidor `unixServer` que aloja los repositos para UNLIX 2.0. (Ver tabla 8.11)

Por ejemplo la instalación de `mysql-server` se la realiza usando el comando `apt-get`.

**# `apt-get install mysql-server`**

Esta gestor de paquetes (`apt-get`), clasifica los paquetes por categorías, ahí se puede personalizar la distribución agregando o quitando paquetes de software, una particularidad de esta herramienta, es que busca un paquete desde el DVD, siempre y cuando este habilitada esta opción (sin el signo #) en el archivo `/etc/apt/sources.list`, de no existir el paquete, procede a buscarlo desde los repositorios que estén configurados en el `/etc/apt/sources.list`.

Otra forma de instalar aplicaciones en un sistema GNU/LINUX, es mediante el gestor de paquetes Synaptic, también se puede usar otras vías para la personalización del sistema UNLIX 2.0 o de sistemas operativos derivados de Debian como `kpackage`. La figura 8.17 representa un ejemplo usado el gestor de paquetes Synaptic.



**Figura 8.17** Pantalla administrador de programas de debian

En fin UNLIX (en su primea versión) posee aplicaciones generales propias de sistemas GNU/LINUX y aplicaciones propias que usan los estudiantes del Área de Energía, Industrias y Recursos Naturales no Renovables de la Universidad Nacional de Loja, para UNLIX 2.0 se ha tomado con referencia la mayoría de los paquetes, claro tomando en cuenta las versiones más actuales, debido a que el software libre se actualiza muy rápido, el plan de actualización de los paquetes de UNLIX 2.0 debe ser cada inicio de año académico (cada septiembre), debido a que los paquetes de software en los sistemas libres como los GNU/LINUX, se actualizan constantemente (por lo general cada seis meses), dicha actualización se la debe realizar desde un servidor de paquetes para UNLIX 2.0 el cual se encuentra alojado en la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja. (Ver capítulo 8.2.1 Requerimientos del Servidor para el Repositorio)



A continuación las tablas 8.17, 8.18, 8.19, 8.20, 8.21, 8.22 y 8.23, muestran los principales paquetes de software que posee el sistema operativo UNLIX 2.0 GNU/LINUX.

<b>MULTIMEDIA</b>		
<b>Paquete</b>	<b>Función</b>	<b>Versión</b>
<b>Sound Juicer</b>	Extractor de Sonido de CD	2.28.2
<b>Rhythmbox</b>	Reproductor de música	0.12.8
<b>Brasero</b>	grabación de CD/DVD	2.30.3
<b>K3B</b>	grabación de CD/DVD	2.0.1
<b>Toten-xine</b>	Reproductor de películas	2.30.2
<b>Sound-recorder</b>	Grabador de sonidos	2.30.0

**Tabla 8.17 Paquetes Multimedia Unlix 2.0**

<b>OFIMÁTICA</b>		
<b>Paquete</b>	<b>Función</b>	<b>Versión</b>
<b>OpenOffice.Base</b>	Herramienta para base de datos	3.2.1
<b>OpenOffice.Calc</b>	Herramienta para hojas de calculo	3.2.1
<b>OpenOffice.Draw</b>	Herramienta de para Dibujos y diagramas	3.2.1
<b>OpenOffice.Impress</b>	Herramienta para presentación de diapositivas	3.2.1
<b>OpenOffice.Math</b>	Herramienta para fórmulas matemáticas	3.2.1
<b>OpenOffice.Write</b>	Herramienta de procesador o editor de textos	3.2.1
<b>Planner</b>	Herramienta de gestión de proyectos	0.14.4

**Tabla 8.18 Paquetes Ofimática Unlix 2.0**



<b>ELECTRÓNICA</b>		
<b>Paquete</b>	<b>Función</b>	<b>Versión</b>
<b>Geda</b>	Herramienta para diseños eléctricos	1.6.1.20
<b>Electric</b>	Programa para dibujar sistemas eléctricos	8.1

**Tabla 8.19 Paquetes Electrónica Unix 2.0**

<b>INTERNET</b>		
<b>Paquete</b>	<b>Función</b>	<b>Versión</b>
<b>Amsn</b>	cliente de mensajería instantánea	0.98.3
<b>BitTorrent</b>	Descargas de Archivos	2.03
<b>Evolution</b>	Cliente de Correo electrónico	2.30.3
<b>Pidgin</b>	cliente de mensajería instantánea	2.7.3
<b>Empathy</b>	cliente de mensajería instantánea	2.30.3
<b>FileZilla</b>	Transferencia de archivos con ssh con servidores remotos	3.3
<b>Chromium</b>	Navegador Web	6.0.1
<b>Iceweasel</b>	Navegador Web parecido al Firefox	3.5.16
<b>Putty</b>	Para trabajar con servidores remotos (ssh)	V. 2010
<b>ekiga</b>	Aplicación para vídeo conferencia	3,2.7

**Tabla 8.20 Paquetes Internet Unix 2.0**



<b>DESARROLLO</b>		
<b>Paquete</b>	<b>Función</b>	<b>Versión</b>
<b>Python</b>	Lenguaje de alto nivel, orientado a objetos	2.6,6
<b>Php</b>	Lenguaje para creación de páginas dinámicas	5.3.3-7
<b>Apache</b>	Servidor HTTP	2.2.16-6
<b>OpenJdk</b>	Herramienta para programas con java	1.6.0.18
<b>Java</b>	Lenguaje de programación orientado a objetos	1.6.0.22
<b>MonoDevelop</b>	Herramienta programación al estilo .net (C#)	2.4
<b>Eclipse</b>	Herramienta para programas con java	3.5.2
<b>NetBeans</b>	Herramienta para programas con java	7.0
<b>Ruby</b>	Lenguaje interpretado de scripts	4.5
<b>MySQL</b>	Gestor de Base de Datos	5.1.49
<b>Tomcat</b>	Servidor Web para JSP	6.0.28-9
<b>PostgreSQL</b>	Gestor de bases de datos objeto-relacionales	8.4.7
<b>Scilab</b>	Herramienta para el análisis matemático	5.2.2
<b>Jython</b>	Python en Java, es un lenguaje de programación de alto nivel, dinámico y orientado a objetos	2.5.1
<b>Drupal</b>	Desarrollo Web	6.0
<b>Moodle</b>	Desarrollo Web para el aprendizaje	1.1.1
<b>Kompozer</b>	Editor gráfico para páginas webs	0.8b3
<b>Nasm</b>	Herramienta para programar modo ensamblador	2.08.01
<b>Glade</b>	Diseño de interfaces para aplicaciones GTK	3.6.7

**Tabla 8.21 Paquetes Desarrollo Unix 2.0**



<b>OTROS</b>		
<b>Paquete</b>	<b>Función</b>	<b>Versión</b>
<b>Grass</b>	Sistemas de información geográfica.	6.4.0
<b>GpsPrune</b>	Programa para cargar, mostrar y editar datos de receptores GPS.	10
<b>GnuCash</b>	Software para hacer un seguimiento de las finanzas.	2.2.9
<b>HomeBank</b>	Software de contabilidad personal libre y fácil	4.2.1
<b>Grisbi</b>	Paquete de gestión de finanzas personales.	0.6.0
<b>Geogebra</b>	Geométrica analítica en las matemáticas	3.2.44
<b>Kalgebra</b>	Programa para trabajar ecuaciones algebraicas	4.4.5
<b>Cycle</b>	Para la medicina gestiona los periodos en el embarazo de la mujer	0.3.1
<b>Aeskulap</b>	Visor de imágenes medicas	0.2.2
<b>FreeDiams</b>	Software de soporte para fármacos	0.4.2
<b>Viewer</b>	Visor de Documentos PDF	2.30.3
<b>VirtualBox</b>	Levanta entornos virtuales	3.2.10

**Tabla 8.22 Otros Paquetes Unlix 2.0**



<b>DISEÑO</b>		
<b>Paquete</b>	<b>Función</b>	<b>Versión</b>
<b>Qcad</b>	Herramienta para sistemas CAD en 2D	2.0.5.0
<b>Blender</b>	Herramienta para Trabajos 3D	2.49
<b>Dia</b>	Herramienta para diagramas estructurados	0.97.1
<b>Koloar Paint</b>	Herramienta para dibujos tipo Paint	4.4.5
<b>Gimp</b>	Herramienta para manipular imágenes	2.6,10
<b>Xsane</b>	Programa para trabajar con Scanners	1
<b>Shotwell</b>	Gestor de Fotografías	0.6.1
<b>Simple Scan</b>	Programa para escanear documentos	2.31.1

**Tabla 8.23 Paquetes Diseño Unlix 2.0**

Aquí se tiene los principales programas de UNLIX 2.0, como se puede ver está orientado a múltiples disciplinas, es decir cubre algunos requerimientos de software usado por los estudiantes y profesores de todas las áreas de la Universidad Nacional de Loja, esta es la idea central de UNLIX 2.0 GNU/LINUX, que su orientación sea para el uso de todas la áreas de la Universidad Nacional de Loja. Cabe destacar que UNLIX en su primera versión está orientado al uso específico del Área de la Energía, las Industrias y de los Recursos Naturales no Renovables.

Este análisis entre UNLIX en su primera versión y UNLIX 2.0, no restringe su uso por lo que todos los usuarios amantes del software libre lo pueden tener como su sistema GNU/LINUX y lo puedan explorar a su máximo nivel como software de escritorio o como servidor, ya que está basado en Debian.

Además UNLIX 2.0 incorpora algunos documentos propios para estudiantes de la carrera de derecho como las leyes que están en vigencia en la actualidad, también algunos documentos de distribución libre para estudiantes de las demás Áreas, por ejemplo del Área de la Salud, Área Jurídica, Área Agropecuaria y Área Educativa.



### 8.3.4 Netbeans en UNLIX 2.0

UNLIX 2.0 GNU/LINUX, está basado en Debian 6.0.1 recoge las características y configuraciones básicas de Debian, los paquetes adicionales que se han instalado fueron tomados del repositorio del servidor de UNLIX 2.0. Se conoce que existen programas gratuitos pero que no son libres en su totalidad que no están en el repositorio de Debian por lo tanto no constan en el repositorio de UNLIX 2.0, un claro ejemplo es netbeans (herramienta de desarrollo con java, jsf, php, uml), si se ingresa a la página de netbeans desde un navegador <http://www.netbeans.org>, se encuentra el paquete para descargarlo e instalarlo en el equipo que se esté trabajando. Si se trabaja en la red de la Universidad Nacional de Loja, se puede descargar el programa desde el servidor UNLIX 2.0, para ellos se usa cualquier navegador y se escribe la siguiente dirección: <http://172.16.32.29/unlix2/descargas/> (ver figura 8.18).



Figura 8.18 Netbeans en Unix 2.0



El archivo lleva el nombre “netbeans-7.0-ml-linux.sh”, se observa que la extensión no es .deb (paquetes propios de Debian) sino es .sh (paquetes ejecutables para sistemas GNU/LINUX) este tipo de archivos se los puede instalar con el siguiente comando:

```
# sh netbeans-7.0-ml-linux.sh
```

Posteriormente se levanta la interfaz de instalación y se siguen los pasos de una instalación normal para este caso de netbeans. Es importante aclarar que en el manual del administrador de UNLIX 2.0 (ver capítulo 2.3 Subir Aplicaciones al Servidor Unix 2.0), ahí se explica cómo subir alguna aplicación para que conste en el repositorio de UNLIX 2.0, y para que cualquier usuario lo pueda descargar e instalar en algún computador con UNLIX 2.0 o Debian.

### **8.3.5 Otros paquetes para UNLIX 2.0**

Como se explicó en el apartado anterior con el caso del netbeans, para tener una idea más clara de los .sh, se diría que son ejecutables, así como un instalador .exe de Windows, lo bueno de estos paquetes es, que se los puede instalar en cualquier sistema LINUX, ya que no hay la restricción como ciertos paquetes .deb (para Debian) o .rpm (para fedora o redhat). Los .sh no necesitan hacer conversiones adicionales (transformarlo a .deb) para instalar en UNLIX 2.0 y en otros sistemas GNU/LINUX.

Va existir el caso de que un usuario quiera agregar un paquete, por ejemplo .rpm en UNLIX 2.0, para ello se va a necesitar de un paquete que viene en UNLIX 2.0 llamado alien, al mismo que se lo puede instalar desde la consola e modo root, usando el comando:

```
# apt-get install alien
```



Esto ayuda a ahorrar los pasos cuando se desea instalar un paquete por código en el caso de los tar.gz. Cuando se trabaja con un programa en tar.gz y en rpm, se dice que fácil sería instalarlo de forma gráfica (o por consola con una simple línea de código sin peticiones de rutas, etc.) con RedHat o Fedora, pero resulta que se requiere para UNLIX 2.0 el cual es basado en Debian, en ese caso se haría lo siguiente, primero se debe ubicar en el directorio donde se tenga el paquete y luego se trabaja con el siguiente comando:

**alien --to-deb nombredelpaquete.rpm**

Saldrá una secuencia en la consola, y hasta que no diga que esta creado aunque aparece el paquete .deb en el directorio que se está trabajando, ahora si el paquete con extensión .deb generado se puede instalar usando el siguiente comando:

**dpkg -i nombredelpaquete.deb**

Este sería el procedimiento adecuado cuando se trata de trabajar con paquetes .rpm o paquetes comprimidos como los .tar.gz.

### 8.3.6 Personalizar la Distribución objetivo UNLIX 2.0

Como UNLIX 2.0 es una distro basada en Debian, se debe configurar la distribución objetivo, se entiende por distribución objetivo al sistema Debian instalado en la computadora, que se lo ha configurado para ser utilizado como base para UNLIX 2.0, a parte de las aplicaciones, los fondos de pantalla son fundamentales en una distribución, los colores, los menús, etc.

Las imágenes personalizadas para la distribución de UNLIX 2.0 GNU/LINUX, es necesario que vayan en el directorio **/usr/share/images/desktop-base/**, para que las mismas estén en el directorio correcto propio de Debian 6.0.1. La figura 8.19, muestra la imagen personalizada de la distribución de UNLIX 2.0



Figura 8.19 Escritorio unix 2.0

Seguidamente se inserta la imagen “**login-background.png**” en el directorio **/usr/share/images/desktop-base/** para que este lista al momento de arrancar el sistema. Otro detalle importante para que la distribución quede mejor presentada es el cambio de la imagen del GRUB, esto se realiza remplazando la imagen “**spacefun-grub.png**” que se encuentra en el directorio **/usr/share/images/desktop-base/** por una personalizada de UNLIX 2.0.

Trabajando con estas sugerencias, se puede tener el sistema objetivo listo para empezar a remasterizar la distro UNLIX 2.0.

### 8.3.7 Experiencias en la elaboración del sistema objetivo

Entre las experiencias encontradas en la elaboración del sistema objetivo, se tiene las siguientes: que en la escritura de los repositorios en el archivo **/etc/apt/sources.list**, se debe poner tal y como vienen los repositorios con su espacio incluido, así como muestra el siguiente ejemplo: (los espacios son muy importantes).



# Repositorios oficiales

```
deb http://172.16.32.29/unlix2/security/ squeeze-updates main contrib non-free
```

```
deb-src http://172.16.32.29/unlix2/security/ squeeze-updates main contrib non-free
```

Caso que exista alguna falla en el texto del archivo del **/etc/apt/sources.list**, el sistema no se actualiza. Importante destacar que la velocidad de descarga de alguna actualización depende de la velocidad de conexión de la red.

Algo importante de Debian 6.0.1 posee un kernel 2.6.32, este incorpora mayor compatibilidad con drivers de laptops modernas.

Además, existió el interés de instalar le SGA (Sistema de Gestión Académica de la Universidad Nacional de Loja), pero no se pudo concretar este anhelo, debido a que proyecto SGA pertenece a la Universidad y aún no ha sido liberado.

## 8.4 CREACIÓN DE LA DISTRIBUCIÓN UNLIX 2.0 CON REMASTERSYS

Es otra herramienta para la creación de Live CDs, la misma se puede utilizar desde Ubuntu o desde Debian, pero existe una pequeña diferencia en Ubuntu incorpora el paquete installer propio del sistema Ubuntu, mientras que en Debian, el installer se lo levanta una vez que se ha ingresado al modo Live CD.

Una de las particularidades de esta herramienta es, por un lado la de crear una copia de seguridad completa del sistema, incluyendo todos los datos personales a un Live CD/DVD y este se puede utilizar en cualquier computador, incluso podemos instalar la copia generada en el Live CD/DVD. Por otra lado, la característica de esta herramienta es que se puede crear una copia distribuible, es decir simplemente un Live CD/DVD sin datos personales de usuario, el mismo que puede ser compartidos con otros usuarios, además permite distribuir para ciertas instituciones que quieran incorporar software libre en sus computadoras, por ejemplo es el Área de la Energía, de las Industrias y de



los Recursos Naturales no Renovables de la Universidad Nacional de Loja, concretamente en las carrera de Ingeniería en Sistemas, se ha probado el trabajar con un Live CD/DVD basado en Debian el mismo que lleva en nombre de UNLIX en su primera versión.

#### 8.4.1 Módulos necesarios antes de la instalación de Remastersys

Para trabajar con remastersys, se debe tener en cuenta que remastersys no es un paquete que se encuentra en el repositorio oficial de Debian estable, se necesita agregar el repositorio correspondiente al archivo `source.list` el mismo que lo encontramos en la siguiente dirección `/etc/apt/sources.list`, se lo abre en modo root (súper usuario) desde el terminal con cualquier editor, por ejemplo con el editor gedit o el editor nano y se procede a agregar en el archivo lo siguiente: (ver tabla 8.20).

sources.list
<pre># Remastersys Squeeze deb http://www.geekconnection.org/remastersys/repository squeeze/</pre>

**Tabla 8.24 Repositorio Remastersys**

A continuación se procede a guardar los cambios en el archivo y se comienza actualizar usando el siguiente comando:

```
# apt-get update
```

Importante tener presente que, antes de proceder con la instalación del paquete de remastersys, en especial si estamos trabajando con Debian, se debe tomar en cuenta los módulos, como es el caso del **squashfs-tools**, paquete de Debian que permite la compresión de archivos, el **aufs-tools** y el **unionfs**, paquetes de Debian que permiten combinar prácticamente el contenido de varios directorios en uno solo. Estos se los instala usado el gestor de paquetes synaptic, tal como lo muestra la figura 8.20

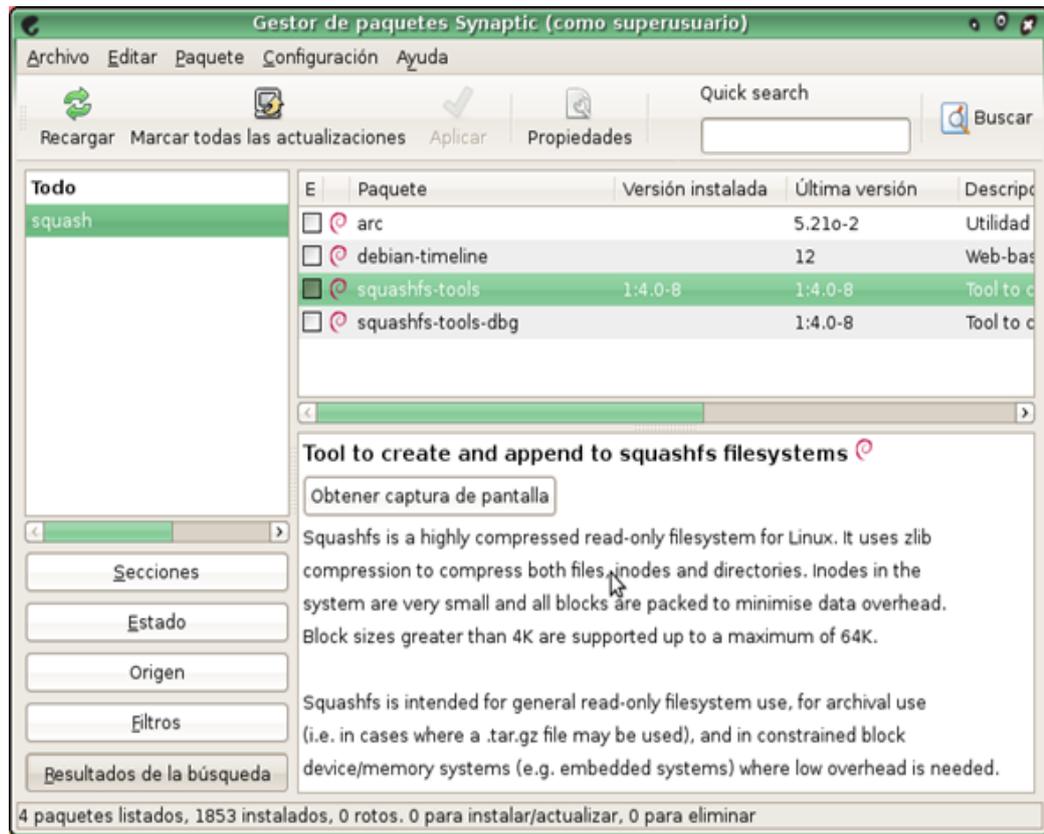


Figura 8.20 Pantalla de synaptic

Continuando con el proceso, se trabaja desde el terminal en modo súper usuario (root) y se va a instalar algunos complementos necesarios para estos paquetes. Estos complementos se hacen por seguridad que es posible que ya estén instalados en el sistema cuando se instaló el squashfs-tools.

Por ejemplo:

```
# apt-get install live-initramfs
```

Este paquete de Debian, hace que el Live CD/DVD tenga su boot y pueda trabajar sin problemas. Una vez instalados estos paquetes se procede a reiniciar el computador.



## 8.4.2 Instalación de la aplicación de Remastersys

Todos los paquetes explicados en el capítulo anterior son muy importantes, ya que de no existir alguno de ellos la distribución no puede funcionar, claro que remastersys se puede instalar (en el caso de que faltare algún paquete anteriormente explicado), pero una vez que se genere el Live CD/DVD este puede tener errores de booteo. Por esto es importante seguir paso a paso el procedimiento de la pre instalación de reamstersys (ver capítulo 8.4.1 Módulos necesarios antes de la instalación de Remastersys).

Ahora sí, está listo el escenario para instalar la herramienta, por lo tanto se procede con la instalación de remastersys, con el comando `apt-get install remastersys` desde la consola en modo root.

**# apt-get install remastersys**

La instalación la hace desde el repositorio de remastersys, por lo tanto el equipo debe estar conectado a internet.

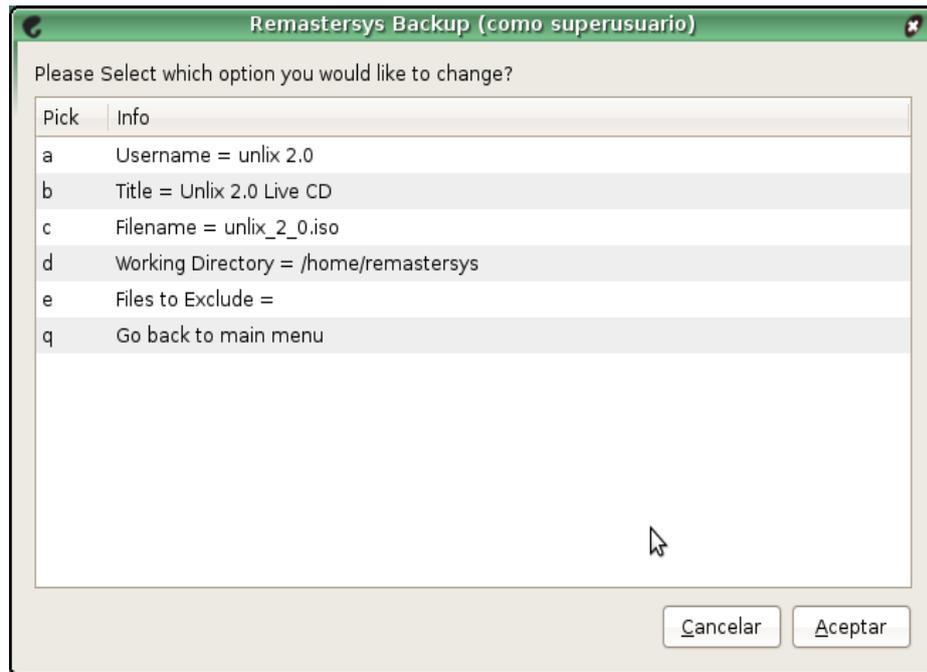
## 8.4.3 Trabajando con Remastersys

Una vez instalado remastersys, ya se puede hacer la distribución, claro que la misma debe estar personalizada y con los programas necesarios según el uso que se le vaya a dar a la distribución.

Para empezar con remastersys se trabaja desde el menú sistema, aplicaciones y remastersys backup, esta opción permite hacer un espejo de todo el sistema, incluidos archivos que se tenga en la carpeta personal de unix, además trabaja con los passwords de usuarios del sistema, para este caso usuario “**unix**”, con clave “**unix**” y de súper usuario (root), con clave “**sistemas**”.



Al realizar este paso, el programa solicita la clave de root, se ingresa la misma y aparece la ventana de remastersys. Cabe señalar que en la ventana principal del programa hay una opción “Modify the remastersys config file to customize options” (ver figura 8.21), ahí se procede asignar nombre a la distribución.

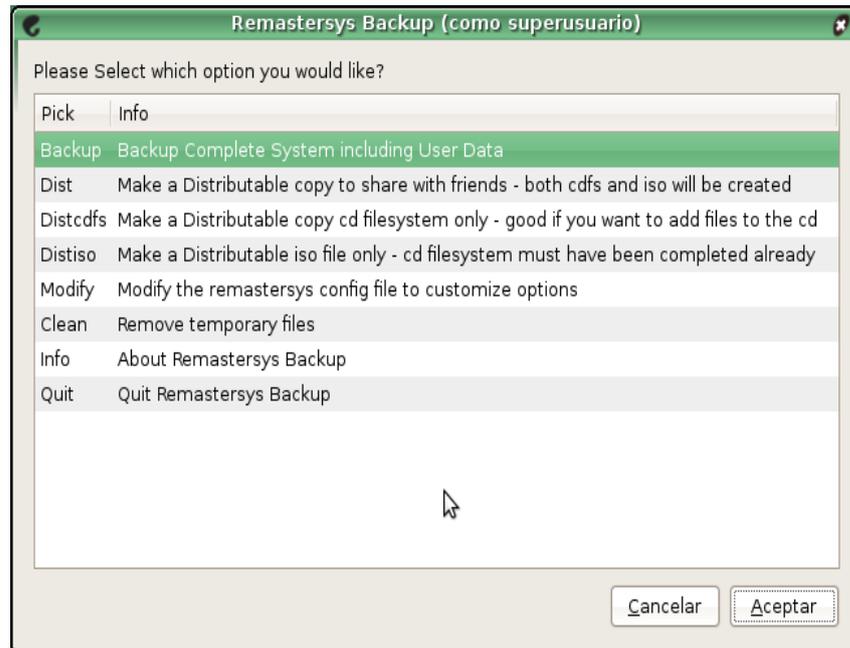


**Figura 8.21** Pantalla de configuración de remastersys

La ventana además indica el lugar donde se va a generar la iso. Ahora ya todo está listo para empezar a generar el proyecto.

#### 8.4.4 Generación de la imagen en un DVD UNLIX 2.0

Configurado la herramienta de remastersys, personalizada la distribución y con el software necesario, se procede a realizar la distribución, eligiendo la opción backup. (ver figura 8.22).



**Figura 8.22 Pantalla de remastersys**

Posterior a la elección, comienza en forma automática a generarse la iso. (Ver figura 8.23)

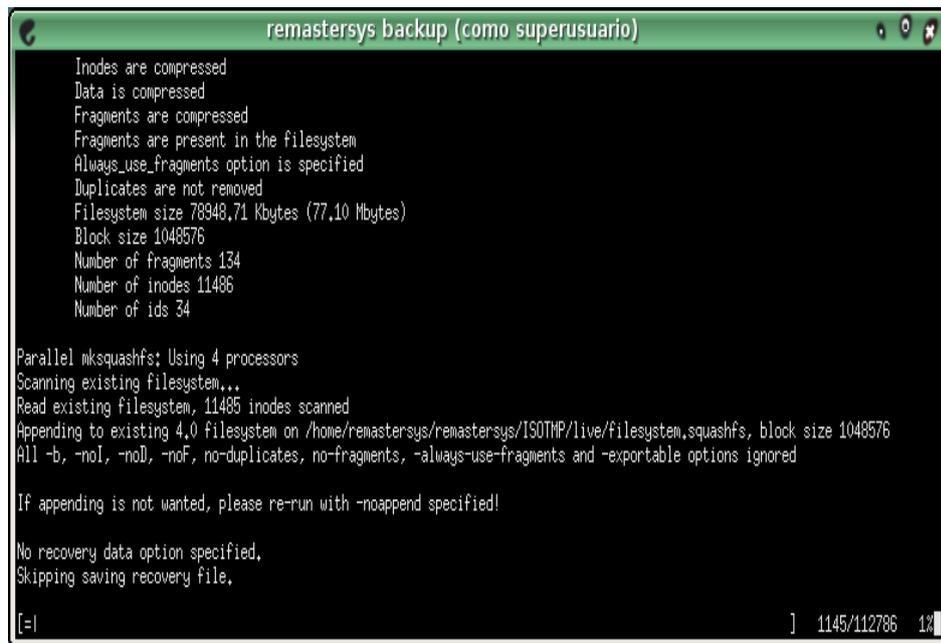


**Figura 8.23 Pantalla trabajando con remastersys**



El trabajo es lento, se observa que solicita los módulos que previamente se instaló, como es el squashfs y el aufs.

En este punto comienza hacer un cálculo de todos los archivos del computador y así proceder a enjaular de modo que toda la información entre en un archivo .iso menor al tamaño de un DVD (ver figura 8.24)



```
remastersys backup (como superusuario)
Inodes are compressed
Data is compressed
Fragments are compressed
Fragments are present in the filesystem
Always_use_fragments option is specified
Duplicates are not removed
Filesystem size 78948,71 Kbytes (77,10 Mbytes)
Block size 1048576
Number of fragments 134
Number of inodes 11486
Number of ids 34

Parallel mksquashfs: Using 4 processors
Scanning existing filesystem...
Read existing filesystem, 11486 inodes scanned
Appending to existing 4,0 filesystem on /home/remastersys/remastersys/ISOTMP/live/filesystem.squashfs, block size 1048576
All -b, -noI, -noF, no-duplicates, no-fragments, -always-use-fragments and -exportable options ignored

If appending is not wanted, please re-run with -noappend specified!

No recovery data option specified.
Skipping saving recovery file.

[=] ] 1145/112786 1%
```

**Figura 8.24 Pantalla creando la iso con remastersys**

Terminando el análisis comienza a generar el producto final que es un archivo .iso, aquí es cuando se va a demorar entre unos 20 minutos a una hora dependiendo de la velocidad del computador que se esté trabajando.

Al finalizar el programa emite un mensaje señalando que la iso ha sido generada, lo que ahora corresponde hacer es grabar la iso en un DVD y para ello se va usar el programa K3B.



Cabe anotar que el programa K3b tiene todas las opciones para grabar un CD o DVD, como cualquier otro programa profesional de grabación posee, para este caso se elige la opción de grabar imagen iso, se va al directorio `/home/remastersys/unlix_2_0.iso`, que es donde se encuentra la iso y se procede a grabar en el DVD.

#### 8.4.5 Generación de la imagen en un USB UNLIX 2.0

UNLIX 2.0 GNU/LINUX, además de funcionar en modo Live DVD, también puede generarse en un Live USB, es decir también se puede crear la imagen Live en un flash memory de capacidad de 4 GB, ya que la distribución de UNLIX 2.0 GNU/LINUX requiere 2.5 GB de almacenamiento aproximadamente.

Para empezar con esta opción de remastersys se trabaja desde el menú sistema, aplicaciones y remastersys USB Startup Disk Tool, esta opción permite hacer del iso (`unlix_2_0.iso`) creado en el apartado anterior con la opción remastersys backup (ver 8.4.4 Generación de la imagen en un DVD UNLIX 2.0). un Live USB, la siguiente imagen muestra el entorno de esta opción. (Ver figura 8.25).

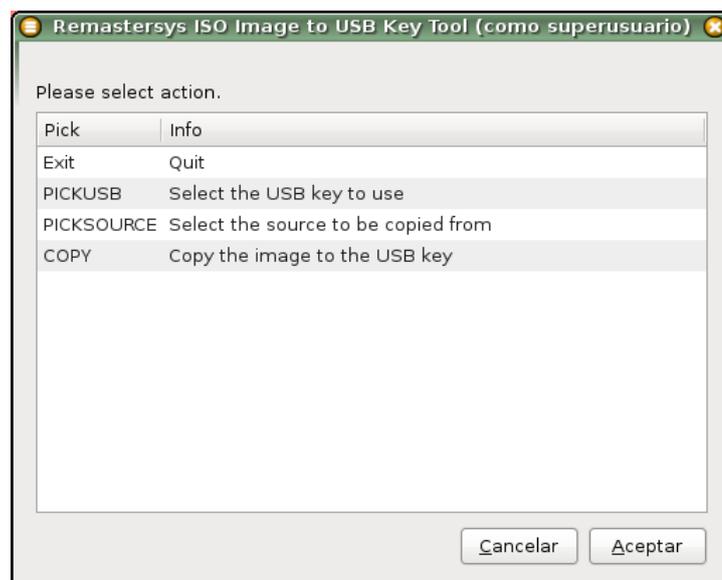


Figura 8.25 Remstersys Usb

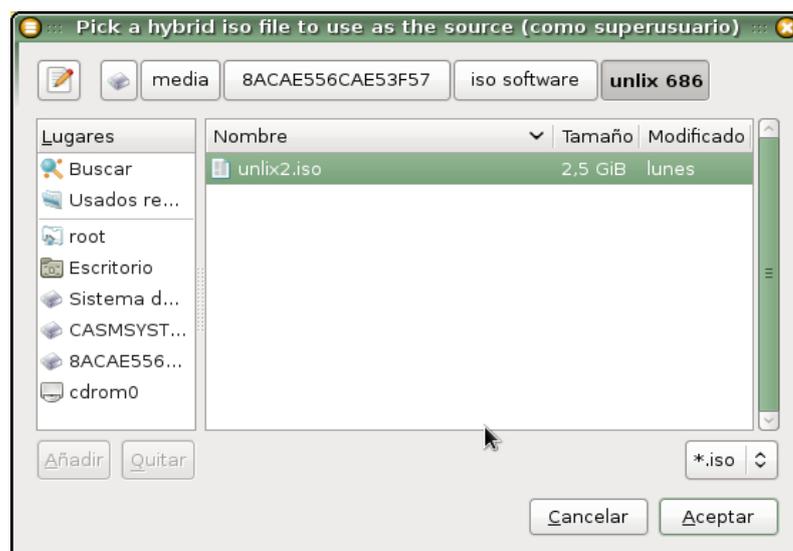


La opción PICKUSB, permite seleccionar el dispositivo Usb (memoria flash) que se va usar, cabe señalar que este dispositivo debe estar formateado, es ahí donde se va a generar el Live USB UNLIX 2.0 GNU/LINUX, tal como muestra la figura 8.26



**Figura 8.26 Selección de la memoria usb**

Posterior a la selección se escoge la opción siguiente de la herramienta, esta es PICKSOURCE, esta permite ir al lugar donde se generó la iso de UNLIX 2.0 y se elige la opción aceptar. (Ver figura 8.27).



**Figura 8.27 Selección de la iso Unlix 2.0**



Finalmente se elige la opción COPY del menú para que se empiece a generar el Live USB UNLIX 2.0 GNU/LINUX, a continuación hay que esperar unos minutos para que construya el USB Live.

Este Live Usb puede ser usado en cualquier computador, incluso se ejecuta más rápido que el Live DVD, sería una opción importante para aquellos que quieran usar UNLIX 2.0 y no se animen a instalarlo en su computador.

#### **8.4.6 Experiencias de la creación de la imagen**

Entre las experiencias que se tuvo al trabajar con esta herramienta se tiene las siguientes, cuando no se instala primero los módulos squashfs-tools y aufs-tools, es decir si se instala remastersys y luego estos módulos, al momento de hacer correr el DVD muestra un error de tipo “initrams no found” y no arranca la aplicación. Por eso se recomienda hacer en el orden correcto y siguiendo los pasos que se indica en el (capítulo 8.4.1 Módulos necesarios antes de la instalación de Remastersys) de este trabajo de investigación.

Para los que trabajan con Ubuntu y desean hacer un proyecto similar, el procedimiento es el mismo para la instalación de remastersys, solo debe cambiar el repositorio que sea para Ubuntu en el archivo **/etc/apt/sources.list**. También es necesario saber que Ubuntu incorpora la opción installer en el booteo y en cambio Debian no, para trabajar con el installer se debe iniciar primero en el modo Live Cd y luego se realiza la instalación.

Remastersys, también se lo usa para hacer copias de seguridad del sistema en su totalidad, la opción backup como se explicó genera una copia de toda los archivos que se tenga en el equipo.

Entre muchas herramientas para trabajar Live CD/DVDs que existen, se observa que hay problemas con Debian, por ejemplo garfio es solo para Ubuntu, así mismo reconstructor igual para Ubuntu. Hay que tener cuidado con las dependencias de los



paquetes si se está trabajando con Debian, por esta razón remastersys exige un conjunto de pasos para que pueda funcionar de forma correcta.

Además es necesario destacar cuando se trabaja con Ubuntu y se quiere hacer una distribución los paquetes explicados en el apartado 8.4.1 “Módulos necesarios antes de la instalación de Remastersys” del presente documento ya no se los toma en cuenta, simplemente se debe agregar el repositorio correspondiente de Remastersys en el archivo `/etc/apt/sources.list` y proceder con la instalación de la herramienta.

Claro que esto se lo hace una vez personalizada la distribución, esto con los fondos respectivos, colores y paquetes de software correspondientes.

Gracias a esto se pudo hacer una distribución para el Colegio Militar Lauro Guerrero “Comil”, la misma que fue expuesta en la feria tecnológica que realizó el colegio en el mes de mayo del 2011, donde la distribución hecha bajo Ubuntu y Remastersys funcionó con éxito. Esta distribución fue realizada gracias a que Ubuntu es derivado de Debian y por ende el proceso que se sigue para la construcción de la misma es similar al que se utilizó para generar UNLIX 2.0. (Ver figura 8.28).



**Figura 8.28** Exposición de la Distro en el “Comil”



## 8.5 CREACIÓN DE UN CD/DVD PARA ACTUALIZAR PAQUETES

En un sistema cualquiera que este sea, la utilización de herramientas que ayuden a la facilidad de uso o de manejo de ciertas cosas dentro de este sistema es primordial para que el resultado sea de lo más versátil y exitoso, ya sea en una actividad normal o en una empresa. De igual forma sucede en los sistemas computacionales, ya que estos tienen software incorporado para que se efectúe una tarea de la forma más sencilla posible, es así el caso de los sistemas operativos que traen paquetes de software que se adicionan o se instalan al sistema base para que sirvan de ayuda en la solución de ciertas necesidades, estos paquetes se instalan y se usan de manera directa, pero en los sistemas GNU/LINUX hay dependencias entre paquetes de software unos necesitan de otros que estén previamente instalados en el sistema, otros necesitan algunos módulos de uso de otro paquete y así son dependientes de sus archivos, por eso es que en los sistemas GNU/LINUX las dependencias son primordiales a la hora de instalar una aplicación.

Es por esta razón, es que existen los repositorios donde se alojan todas la aplicaciones listan para instalar y que son dependientes entre ellas y de esta manera pueden ser instaladas en el sistema con la ayuda de las herramientas synaptic, atp-get y aptitude y por lo general estos repositorios son alojados en la red de redes.

Es así, cuando se instala por primera vez un sistemas GNU/LINUX en un computador se actualiza con las aplicaciones que faltan desde el internet, accediendo hacia algún repositorio disponible en la red global y de esta forma se instalan las aplicaciones restantes o necesarias de acuerdo al gusto del usuario, pero que se lo hace sin ningún problema de dependencias porque ya las resuelve la herramienta conectada al repositorio.

Es el mismo caso con el cual funciona UNLIX 2.0 porque es un sistema GNU/LINUX, pero este nuevo sistema basado en Debian va a incorporar un CD/DVD como repositorio primario, para que sea fuente de instalación de software y que sirva como actualizador de otro sistema, sin la necesidad de utilizar internet, pero que va a estar



cargado con los paquetes de software básicos.

Para hacer o construir este CD/DVD se tiene la ayuda de la herramienta APTonCD, que es una herramienta que se puede instalar en los sistemas GNU/LINUX y que es de fácil uso porque es gráfica, la herramienta en si es una software que genera un CD o DVD pero con la solución de que resuelve las dependencias de los paquetes instalados desde synaptic, aptitude y apt-get , esta herramienta copia los paquetes que se alojan en **/var/cache/apt/archives** y con ellos crea una iso totalmente consolidada para que este lista a grabarse en un medio extraíble como es un CD/DVD, esta iso se va a alojar en **/home /usuario/APTonCD/**. Una vez puesta la iso en un DVD o CD se puede llevar a otro computador y con solo insertar el disco en ese computador se tiene todos los paquetes listos para ser actualizados sin la necesidad de internet o de estar conectado a un repositorio.

A continuación se presentan los pasos a seguir para construir el CD/DVD de un repositorio primario de UNLIX 2.0

Primero se la busca el paquete desde el repositorio con el siguiente comando en consola de root,

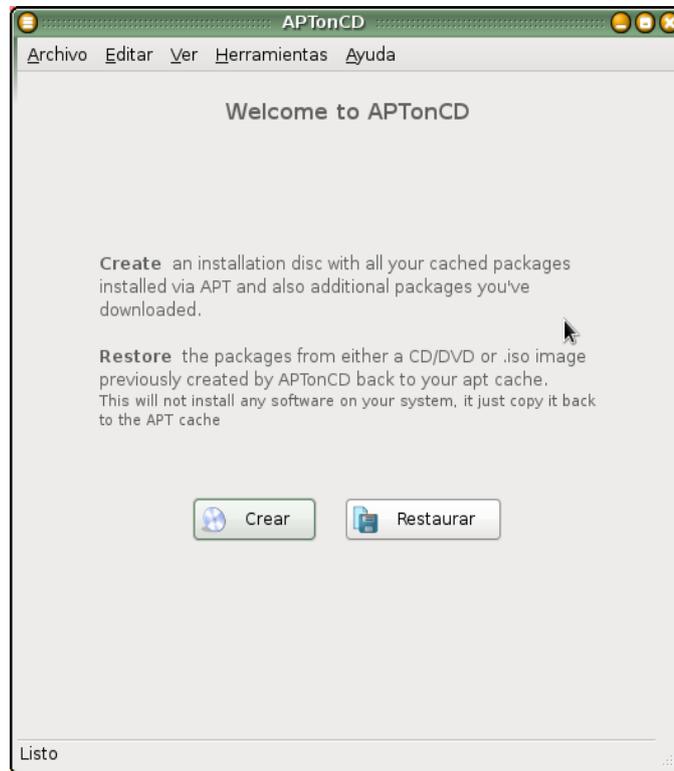
```
# aptitude search aptoncd
```

Y luego se la instala con el comando:

```
# apt-get install aptoncd
```

Y listo se tiene la herramienta disponible, para trabajar con la misma hay que ir a la opción Sistema, luego Administración y APTonCD

Luego se levanta la interfaz de APTonCD solamente dando clic en ella y se escoge de la pantalla inicial la opción de crear como muestra la figura 8.29.



**Figura 8.29 Inicio AptonCd**

Posterior a esto se espera que se cargue los paquetes que se encuentra en el directorio `/var/cache/apt/archives` que son `.deb` y que han instalados recientemente en el sistema y que van a ser objeto de unión para formar la iso del CD/DVD, si se quiere se puede añadir más paquetes con la opción agregar pero si no solo se procede a pulsar en el botón grabar y se continua con el proceso. (Ver figura 8.30).

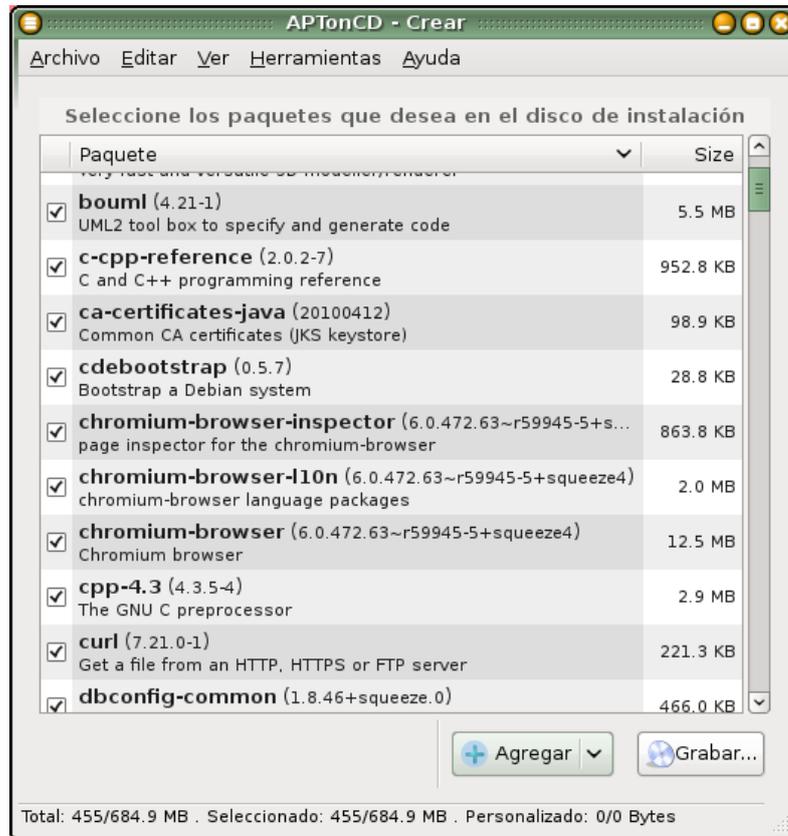


Figura 8.30 Paquetes AptonCd

Cuando se hace el paso que es grabar el CD/DVD se tiene el despliegue de la siguiente pantalla tal como indica la figura 8.30, aquí se da la opción de “tipo de medio” donde se va a efectuar la grabación, como se quiere hacer un CD/DVD se escoge ya sea CD o DVD (según la capacidad del medio), se elige la carpeta destino, luego se ve que por defecto se le pone un nombre en este caso “dvdPaquetesUnix” a la iso. La siguiente opción es para saber si se desea hacer un metapaquete, esto permite que los paquetes se instalen de una sola porque son agrupados en un solo paquete contenedor.



**Figura 8.31 Creación del CD/DVD con AptonCd**

Al dar clic en el botón aplicar de la figura 8.31, se genera la iso que se va a alojar en **/home/unlix**, luego puede ser grabada por cualquier software de grabación de CD/DVDs, obteniendo de esta manera un repositorio primario alojado en un CD/DVD para poder hacer la actualización en algún sistema derivado de Debian (UNLIX 2.0) sin la necesidad de estar conectado a internet. (Ver figura 8.32)



**Figura 8.32 Grabación del CD/DVD con AptonCd**



En si APTonCD esta es una herramienta que permite hacer repositorios de CD/DVDs, que son muy útiles para hacer actualizaciones al sistema sin la necesidad de estar conectado al internet, esta herramienta en UNLIX 2.0, ya viene incluida por defecto de igual forma que otras herramientas útiles para otras necesidades, por eso UNLIX 2.0 es un sistema confiable y útil.

## 8.6 PLAN DE VALIDACIÓN.

En esta fase de pruebas del proyecto UNLIX 2.0 GNU/LINUX, se la realizó en algunas computadoras, con personal de la Universidad Nacional de Loja (docentes y estudiantes de la carrera de Ingeniería en Sistemas, de la carrera de informática educativa, Unidad de Telecomunicaciones e Información, departamento de mantenimiento).

Se quedó demostrado que UNLIX 2.0 posee una gran compatibilidad con diferentes arquitecturas, ya en todos los equipos donde fue probado el producto funcionó de forma correcta, sin ninguna novedad, claro que se debe tomar muy en cuenta que en equipos modernos (min 2 GB de memoria Ram), la velocidad de arranque y ejecución de los programas es muy buena, pero no se descarta equipos con 512 de memoria Ram, donde UNLIX 2.0 funciona de forma correcta.

### 8.6.1 Pruebas con Usuarios

Para realizar el plan de pruebas, se tomó la muestra con estudiantes, profesores y administrativos de la Universidad Nacional de Loja y para obtener la muestra se utiliza **el método muestreo aleatorio simple**, que enuncia que cada elemento de la población tiene la misma probabilidad de ser tomado en cuenta para el estudio de la misma como muestra. Este método es utilizado con la variante de muestreo sin reemplazo, que es coger un elemento de la población y no volverlo a poner dentro de la misma para que sea tomado de nuevo en cuenta dentro de la muestra, es decir elemento tomado como muestra se excluye de la población por eso es sin reemplazo.



Para determinar la muestra en el caso de los posibles usuarios de UNLIX 2.0 se toma como fórmula estadística la siguiente:

$$n = \frac{n'}{1 + \frac{n'}{N}}$$

Teniendo en cuenta que  $n' = \frac{s^2}{\sigma^2}$

Dónde:

n es el número de muestra

N es la población

$\sigma^2$  es la varianza de población respecto de determinadas variables

$s^2$  es la varianza de la muestra la cual puede determinarse en términos de probabilidad

$s^2 = p(1-p)$

(se) es error estándar

$(se)^2 =$  es el error estándar al cuadrado que sirve para determinar  $\sigma^2$ , por lo que

$\sigma^2 = (se)^2$  es la varianza poblacional.

Para el caso de UNLIX 2.0 se tiene una población de N=14000 usuarios entre alumnos docentes y empleados, el estudio que se va a hacer en esta población es con un 98% de confiabilidad puesto que la Universidad Nacional de Loja, posee alumnos con conocimientos básicos en el manejo y uso de computadoras, así como también sus docentes y empleados. La confiabilidad es la probabilidad que se tiene para determinar si el sistema operativo UNLIX 2.0 va a tener un nivel de aceptación bueno dentro de la Universidad, por lo que  $p = 0,98$  y por ende  $s^2 = p(1-p)=0,0196$ . Esta muestra se la toma con un error estimado de 0.032 quedando el valor como sigue  $(se)=0,032$ , por lo que se obtiene a  $\sigma^2 = (se)^2 = 0,001024$ . Entonces aplicando la fórmula de n para obtener la muestra se consigue como resultado lo siguiente:



$$n = \frac{\frac{0,0196}{0,001024}}{1 + \frac{19,141}{14000}}$$

$$n = 19,11$$

Es decir se necesita hacer 19 encuestas a los usuarios de UNLIX 2.0, con esta muestra se procedió a realizar la respectiva encuesta obteniendo el siguiente análisis.

### **8.6.2 Análisis de los resultados de la pruebas**

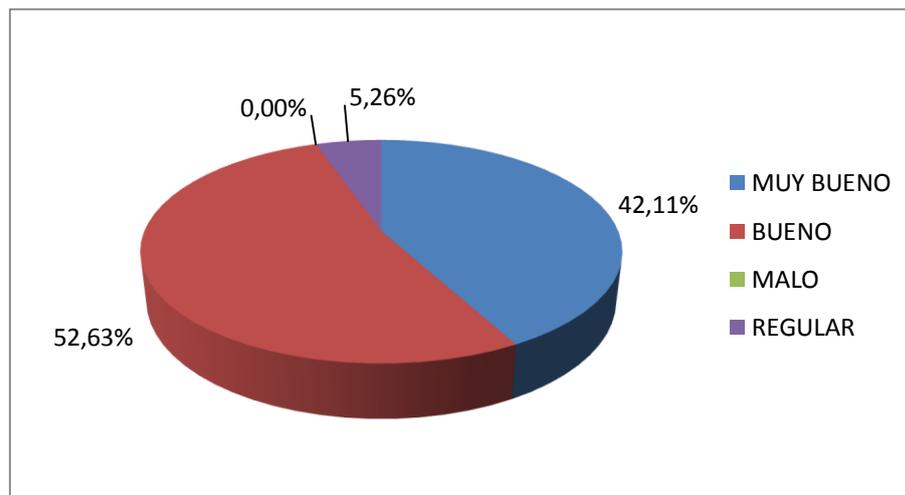
Este análisis de los resultados de la tabulación de las encuestas del sistema operativo UNLIX 2.0 GNU/LINUX se la hizo en base a las siguientes preguntas:



1. ¿El sistema operativo Unix 2.0 a su parecer tiene buen tiempo de respuesta al momento de iniciar en el computador en relación al rendimiento de su máquina?

VARIABLE	FRECUENCIA	PORCENTAJE
MUY BUENO	8	42,11%
BUENO	10	52,63%
MALO	0	0,00%
REGULAR	1	5,26%
TOTAL	19	100,00%

**Tabla 8.25 Tiempo de respuesta del sistema**



**Figura 8.33 Tiempo de respuesta del sistema**

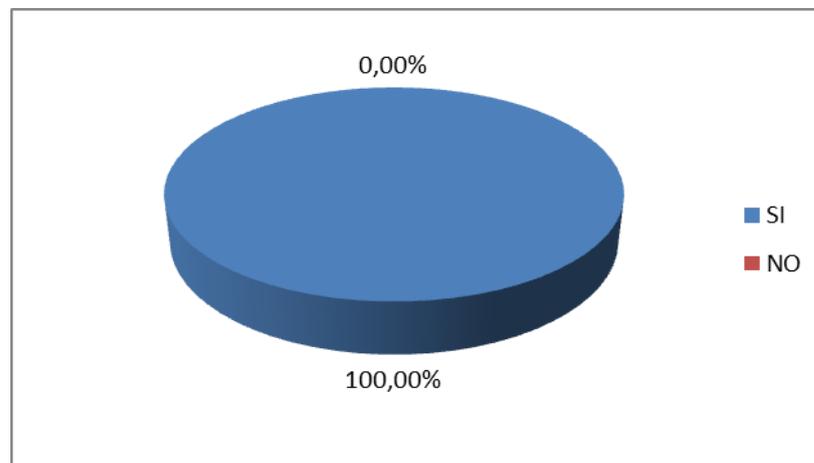
De los 19 encuestados un 42,11 % expresan que es muy bueno el tiempo de respuesta que tiene el sistema al momento de iniciar en el computador, por consiguiente se puede decir que este sistema es rápido a la hora de responder con las tareas que el usuario requiera, de la misma forma un 52,63 % expresan que es bueno el tiempo de respuesta, corroborando así lo antes mencionado, califican como regular el tiempo de respuesta un 5,26 % y como malo el 0 %. (Ver tabla 8.25 y figura 8.33)



2. ¿Unix 2.0 se levantó sin problemas en su computador?

VARIABLE	FRECUENCIA	PORCENTAJE
SI	19	100,00%
NO	0	0,00%
TOTAL	19	100,00%

**Tabla 8.26 Levantamiento de Unix 2.0**



**Figura 8.34 Levantamiento de Unix 2.0**

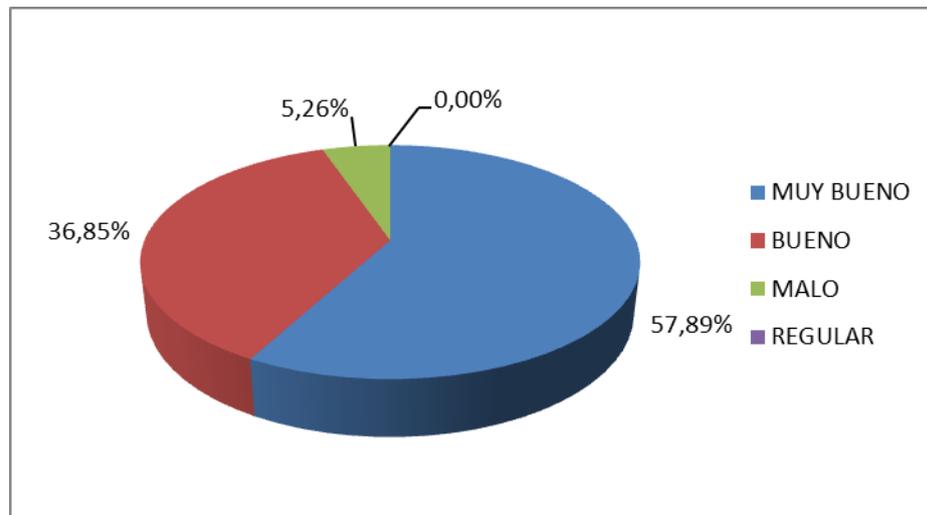
Un 100 % se pronuncia a favor de que el sistema se levanta sin problemas en una computadora, teniendo en cuenta que puede ser de cualquier arquitectura, el sistema funciona bien en computadoras nuevas y viejas. (Ver tabla 8.26 y figura 8.34)



3. ¿Piensa usted que Unix 2.0 tiene buena accesibilidad en cuanto al manejo de su interfaz de escritorio?

VARIABLE	FRECUENCIA	PORCENTAJE
MUY BUENO	11	57,89%
BUENO	7	36,85%
MALO	1	5,26%
REGULAR	0	0,00%
TOTAL	19	100,00%

**Tabla 8.27 Accesibilidad de Unix 2.0**



**Figura 8.35 Accesibilidad de Unix 2.0**

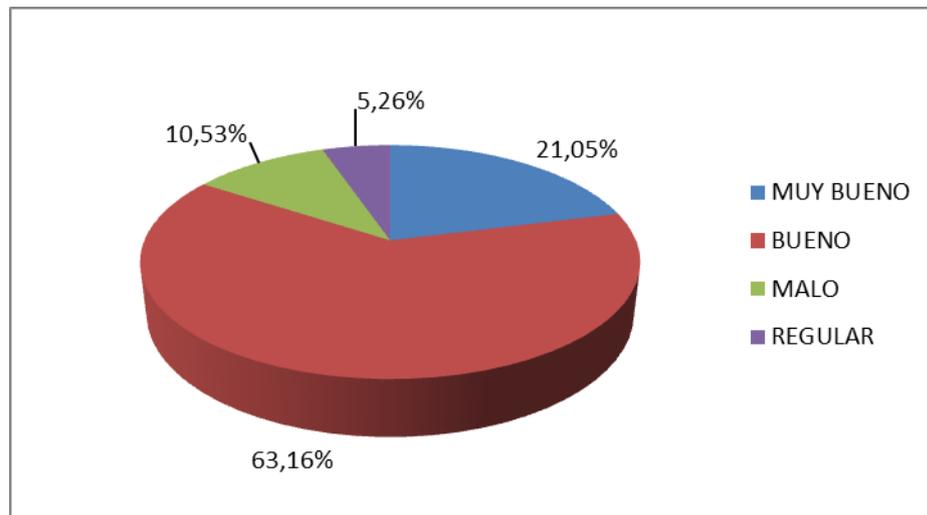
Es muy buena la accesibilidad al sistema y a sus interfaces, por lo que se tiene 57,89 % a favor de la opción muy bueno, un 36,85 % está a favor de la opción, bueno entonces se puede decir que el sistema Unix 2.0 tiene buenas interfaces, en la opción malo se tiene 0 % y como regular hay un 5,26 que menciona deficiencias en el sistema. (Ver tabla 8.27 y figura 8.35)



4. ¿Cómo califica la personalización de Unlix 2.0 le parece acorde a los colores que identifican a la Universidad Nacional de Loja?

VARIABLE	FRECUENCIA	PORCENTAJE
MUY BUENO	4	21,05%
BUENO	12	63,16%
MALO	2	10,53%
REGULAR	1	5,26%
TOTAL	19	100,00%

**Tabla 8.28 Personalización de Unlix 2.0**



**Figura 8.36 Personalización de Unlix 2.0**

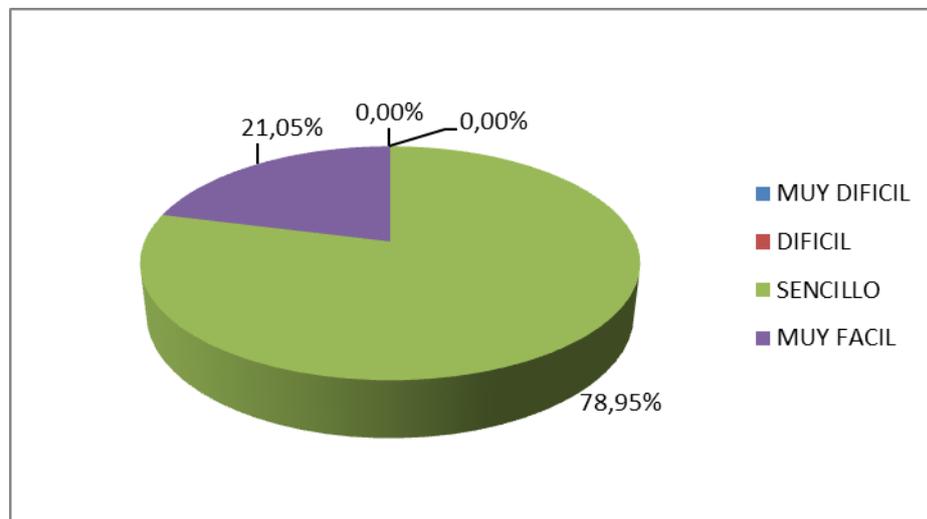
Como sistema que pertenece a la Universidad Nacional de Loja tiene todo su diseño de identificación de la institución dando un 63,16 % como buena la estética del sistema y un 21,05 % como muy bueno, se tiene una apreciación regular del 5,26 % y como mala el 10,53 %. (Ver tabla 8.28 y figura 8.36)



5. ¿Trabajar con el sistema Unix 2.0 le parece?

VARIABLE	FRECUENCIA	PORCENTAJE
MUY DIFÍCIL	0	0,00%
DIFÍCIL	0	0,00%
SENCILLO	15	78,95%
MUY FÁCIL	4	21,05%
TOTAL	19	100,00%

**Tabla 8.29 Trabajar con Unix 2.0**



**Figura 8.37 Trabajar con Unix 2.0**

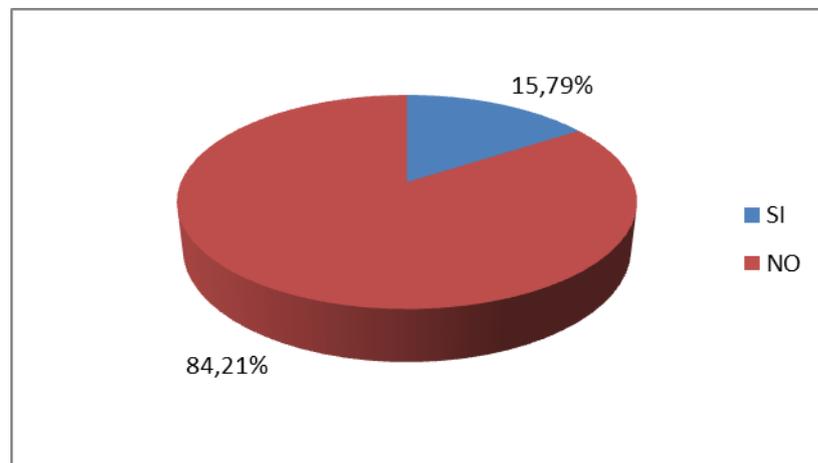
El 78,95 % siente como sencillo el manejo y trabajo con Unix 2.0 y un 21,05 % como muy fácil, obteniendo así un 0 % para la opción muy difícil y 0 % para la opción difícil, por eso se puede decir que trabajar con unix es sencillo. (Ver tabla 8.29 y figura 8.37).



6. ¿Piensa usted que se debe tener conocimientos avanzados de computación para utilizar Unix 2.0?

VARIABLE	FRECUENCIA	PORCENTAJE
SI	3	15,79%
NO	16	84,21%
TOTAL	19	100,00%

**Tabla 8.30 Conocimientos de Unix 2.0**



**Figura 8.38 Conocimientos de Unix 2.0**

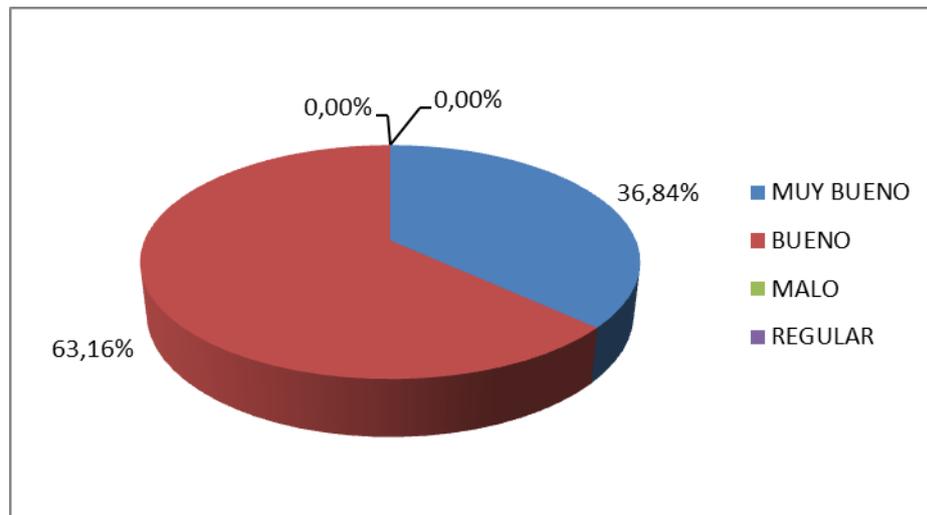
Se puede decir que no se necesita tener conocimientos avanzados de computación para manejar Unix 2.0 debido a que un 84,21 % afirma esto, mientras que la negativa solo recibe un 15,79 % (Ver tabla 8.30 y figura 8.38)



7. ¿Cómo califica el rendimiento de Unix 2.0 a la hora de levantar algún programa dentro del sistema?

VARIABLE	FRECUENCIA	PORCENTAJE
MUY BUENO	7	36,84%
BUENO	12	63,16%
MALO	0	0,00%
REGULAR	0	0,00%
TOTAL	19	100,00%

**Tabla 8.31 Calificación de Unix 2.0**



**Figura 8.39 Calificación de Unix 2.0**

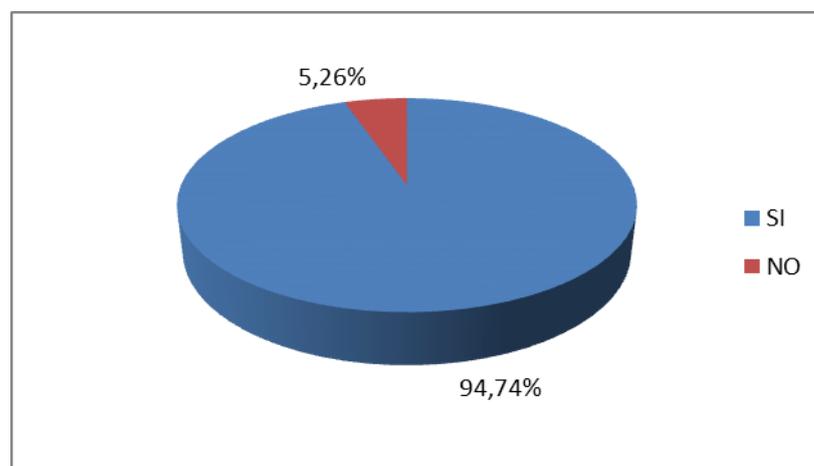
Para levantar cualquier programa Unix 2.0 es un sistema bueno porque tiene un porcentaje de apoyo de 63,16 % y como opción de muy bueno se tiene un 36,84 % entonces el sistema se comporta bien para resolver rápidamente el entorno de cualquier programa porque no es ni regular ni malo debido a que estas dos opciones tienen 0 %. (Ver tabla 8.31 y figura 8.39)



8. ¿Cree usted que Unix 2.0 es confiable, seguro y estable?

VARIABLE	FRECUENCIA	PORCENTAJE
SI	18	94,74%
NO	1	5,26%
TOTAL	19	100,00%

**Tabla 8.32 Confiabilidad de Unix 2.0**



**Figura 8.40 Confiabilidad de Unix 2.0**

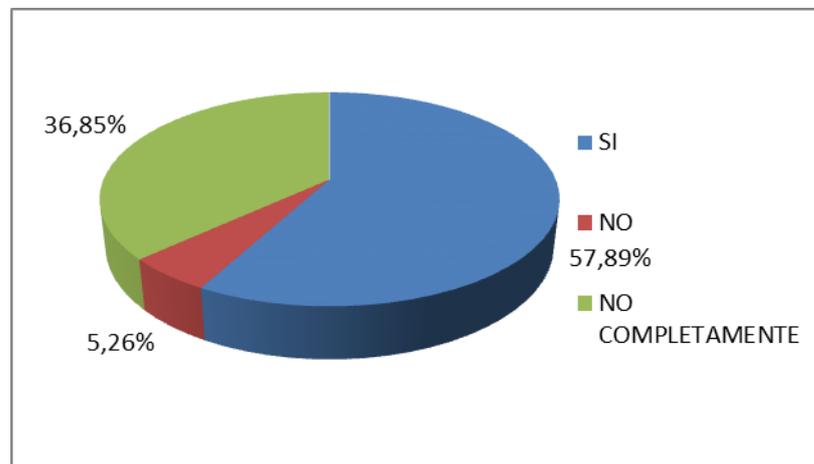
El sistema es confiable seguro y estable ya que un 94,74 % de los encuestados mencionan que sí, de acuerdo al comportamiento de Unix 2.0 en sus computadoras y hay una respuesta negativa solo del 5,26 %. (Ver tabla 8.32 y figura 8.40)



9. ¿Considera que Unix 2.0 le ayuda a resolver sus actividades dentro de su ambiente de trabajo con la utilización de algún programa o aplicación que trae este sistema?

VARIABLE	FRECUENCIA	PORCENTAJE
SI	11	57,89%
NO	1	5,26%
NO COMPLETAMENTE	7	36,85%
TOTAL	19	100.00%

**Tabla 8.33 Utilización de Unix 2.0**



**Figura 8.41 Utilización de Unix 2.0**

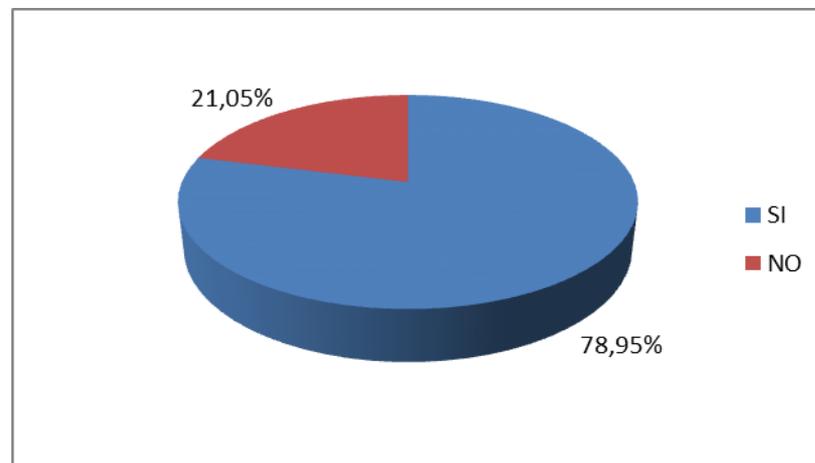
Con un 57,89 % se afirma que Unix 2.0 si resuelve problemas cotidianos de un usuario tecnológico, mientras que la opción de negativa ante esto es de un 5,26 % y hay quienes expresan que Unix 2.0 no resuelve completamente las necesidades de los usuarios con un porcentaje del 36,85 %. (Ver tabla 8.33 y figura 8.41)



10. ¿Unix 2.0 si cumple con los requerimientos y expectativas como usuario, es decir es para usted lo que esperaba de un sistema operativo?

VARIABLE	FRECUENCIA	PORCENTAJE
SI	15	78,95%
NO	4	21,05%
TOTAL	19	100,00%

**Tabla 8.34 Expectativa de Unix 2.0**



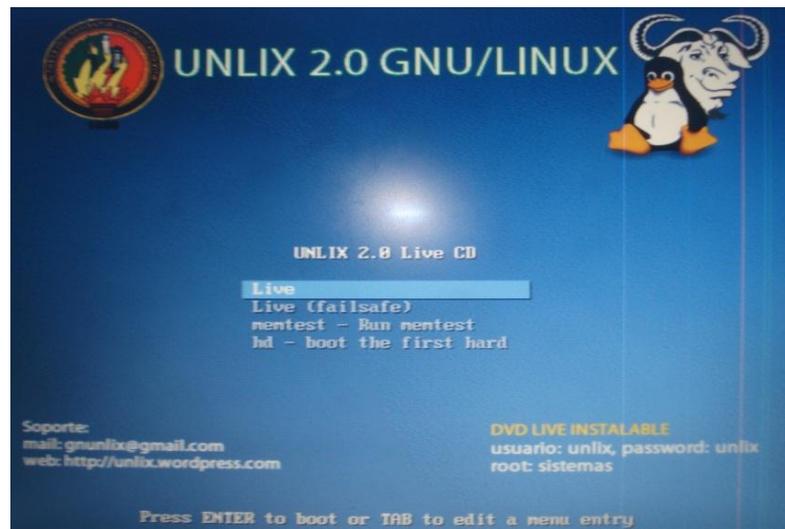
**Figura 8.42 Expectativa de Unix 2.0**

Para el 78,95 % que dice que si cumple el sistema operativo Unix 2.0 las expectativas del usuario, se tiene en buen concepto el sistema ya que es un sistema nuevo y útil, pero así mismo hay una negativa del 21,05 % que resulta de la no aceptación del sistema como herramienta útil y solvente, En definitiva Unix 2.0 es un buen sistema que resuelve las tareas en su gran mayoría de los usuarios tecnológicos. (Ver tabla 8.34 y figura 8.42).



### 8.6.3 Pruebas de Inicio

En el 100 % de equipos donde se probó la aplicación fusionó sin ningún problema, tanto dentro del campus de la Universidad Nacional de Loja, como fuera de la misma. Como se sabe para trabajar con UNLIX 2.0, primero se ingresa al modo Live DVD o USB Live y después el usuario decide si desea instalar el producto o no. (Ver figura 8.43).



**Figura 8.43 Pantalla inicial para instalar UNLIX 2.0**

Una vez que se inicializo con UNLIX 2.0 se observa que existen una gran variedad de paquetes de software que han sido seleccionados en base a un estudio dentro de la Universidad Nacional de Loja y en especial del Área de la Energía específicamente las carreras de sistemas, Geología, Electromecánica.

### 8.6.4 Pruebas de hardware y Software

La característica principal de UNLIX 2.0 es su compatibilidad, así como no se tuvo problemas con el inicio también no presentaron inconvenientes con el video, sonido y red en la diferentes computadoras con arquitecturas i386 y amd64 donde se hicieron las pruebas, solo en el caso de una laptop modelo antiguo en marcha Toshiba, se presentó el problema con la tarjeta inalámbrica, pero se lo soluciono al mismo haciendo la instalación desde el repositorio de “unixServer” del driver de la tarjeta respectiva.



La presente versión de UNLIX 2.0 viene cargado por defecto los drivers de las tarjetas inalámbricas de laptops como las Hp Pavilion dm4 y las compac Presario M2000 y sus derivados, es decir cuando se pruebe UNLIX 2.0 en estos modelos no habrá problema de drivers de las tarjetas inalámbricas.

Se recomienda trabajar con equipos que tenga como mínimo 512 de memoria RAM y el procesador puede ser un Pentium IV en adelante, para que el sistema trabaje de forma correcta.



**Figura 8.44 Trabajando con UNLIX 2.0**

Una vez levantado el sistema, todos los programas que trae UNLIX 2.0 se levantan de forma correcta, tanto para el modo Live o si UNLIX 2.0 está instalado en la computadora. (Ver figura 8.44)

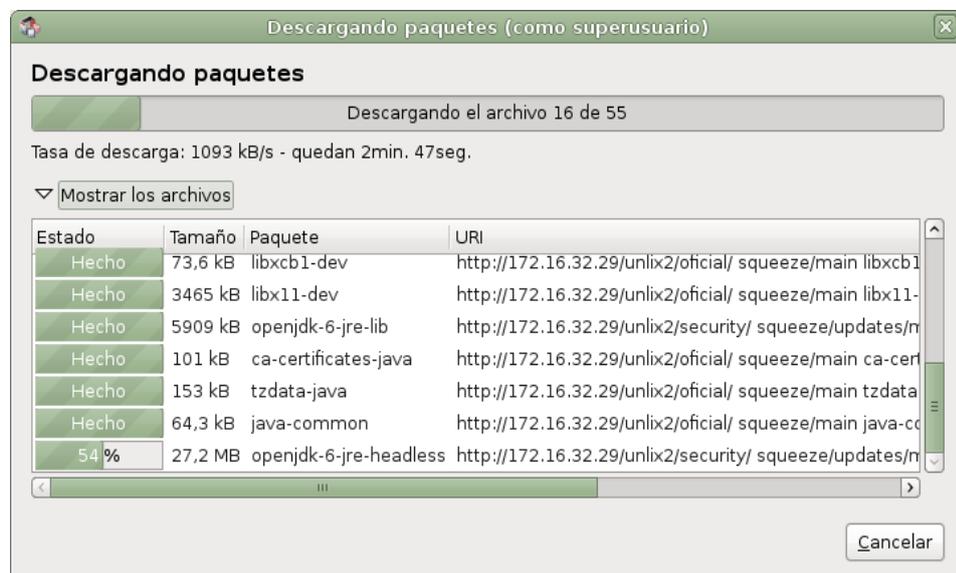
### 8.6.5 Pruebas con el Repositorio

UNLIX 2.0 GNU/LINUX viene configurado por defecto para que sea actualizado dentro de la red del campus la Universidad Nacional de Loja, salvo que alguien que trabaje con UNLIX 2.0 se encuentre fuera la de Universidad Nacional de Loja y decida



actualizar el producto debería ir a los orígenes del software y habilitar las opciones para que se actualice desde los repositorios oficiales de Debian. (Ver capítulo 6.2 Administración del Software fuera de la UNL, del manual de Usuario del presente proyecto).

Para las actualizaciones de algún paquete (actualizar, quitar o agregar), se puede usar el gestor de paquetes synaptic o desde la consola se puede usar el apt-get. (Ver capítulo 6.3 Trabajando con los gestores de Paquetes, del manual de Usuario del presente proyecto).



**Figura 8.45 Actualización con UNLIX 2.0**

La figura 8.45 muestra cómo se instala un programa dentro de la red del campus de la Universidad Nacional de Loja, así mismo una vez que han agregado o quitado algún paquete, el sistema se encarga de hacer la configuración respectiva del mismo tal como lo muestra la figura 8.46.

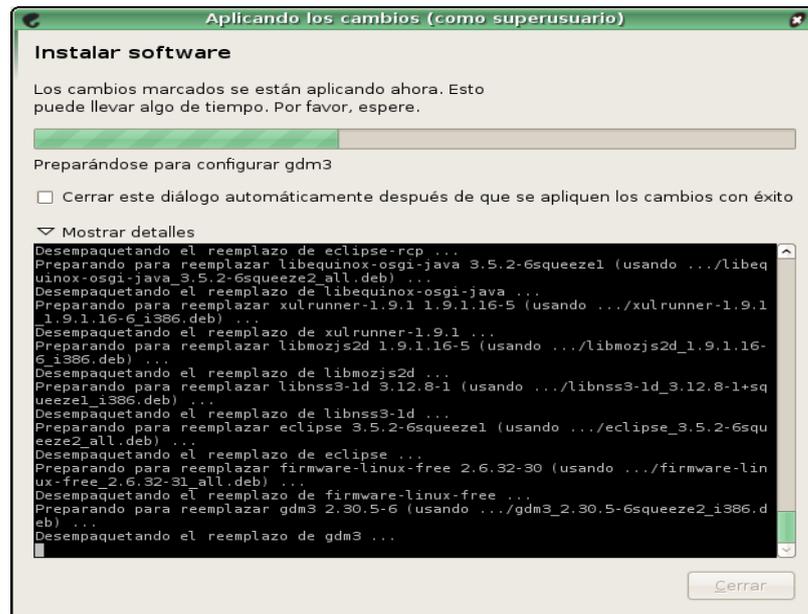


Figura 8.46 Configurando Programas en UNLIX 2.0

Como se puede observar se pueden agregar o quitar todos los paquetes de software que se encuentran en el repositorio del servidor de UNLIX 2.0, el mismo que es completo ya que es basado en el repositorio del servidor oficial de Debian, por lo tanto se puede decir que el repositorio es completo y que funciona sin ningún tipo de problemas.



## 9. VALORACIÓN TÉCNICA ECONÓMICA

Para el análisis de la viabilidad económica y social del proyecto UNLIX 2.0 GNU/LINUX, se deduce que técnicamente se puede cuantificar y se va a comprobar que la inversión es bien baja debido a que se utiliza software libre, esta es la principal ventaja en relación a costos del uso de software propietario. UNLIX 2.0 GNU/LINUX ha tenido la intervención de recursos humanos, técnicos, materiales y tecnológicos, así se tiene: (Ver tablas 9.1, 9.2, 9.3, 9.4, 9.5, 9.6 y 9.7)

Recursos Humanos				
Recursos	Cantidad	# de Horas	Hora/hora	Valor Total
Investigadores	2	-	-	\$ 000,00
Director de Tesis	1	-	-	\$ 000,00
Asesor en Servidor Linux	1	5	\$ 100,00	\$ 100,00
<b>SUBTOTAL</b>				<b>\$ 100,00</b>

Tabla 9.1 Valoración Económica Recursos Humanos

Recursos Materiales	
Recursos	Valor Total
Papel	\$ 20,00
DVDs	\$ 50,00
Cartuchos de tinta para impresora	\$ 50,00
Impresiones	\$ 100,00
<b>SUBTOTAL</b>	<b>\$ 220,00</b>

Tabla 9.2 Valoración Económica Recursos Materiales



Hardware	
Recursos	Valor Total
Computador portátil HP core I5 arquitectura 64 bits	\$ 1200,00
Computador portátil COMPAC PRESARIO M2000	\$ 600,00
CPU (Server) Pentium IV	\$ 200,00
3 Memorias USB 4 GB	\$ 30,00
Impresora HP 2050	\$ 90,00
<b>SUBTOTAL</b>	<b>\$ 2120,00</b>

**Tabla 9.3 Valoración Económica Recursos Hardware**

Software	
Recursos	Valor Total
Debian 6.0.1	\$ 000,00
Servidor Apache	\$ 000,00
OpenOffice	\$ 000,00
<b>SUBTOTAL</b>	<b>\$ 000,00</b>

**Tabla 9.4 Valoración Económica Recursos Software**

Comunicaciones			
Recursos	Cantidad	Valor /Mes	Valor Total
Internet	24	\$ 18	\$ 432,00
<b>SUBTOTAL</b>			<b>\$ 432,00</b>

**Tabla 9.5 Valoración Económica Recursos Comunicaciones**



<b>Recursos Técnicos Tecnológicos</b>	
<b>Descripción</b>	<b>Valor Total</b>
Hardware	\$ 2120,00
Software	\$ 000,00
Comunicaciones	\$ 432,00
<b>SUBTOTAL</b>	<b>\$ 2552,00</b>

**Tabla 9.6 Valoración Económica Recursos Técnicos y Tecnológicos**

<b>Resumen Costos</b>	
<b>Descripción</b>	<b>Valor Total</b>
Recursos Humanos	\$ 100,00
Recursos Materiales	\$ 220,00
Resumen Técnicos y Tecnológicos	\$ 2552,00
<b>SUBTOTAL</b>	<b>\$ 2872,00</b>

**Tabla 9.7 Aproximación del Costo Real del Proyecto**



## 10. CONCLUSIONES

Finalizando el presente trabajo de investigación “CREACIÓN DEL REPOSITORIO DE PAQUETES Y ACTUALIZACIÓN DEL SISTEMA OPERATIVO UNLIX” se pudo cumplir de manera exitosa los objetivos propuestos inicialmente y en este proceso de aprendizaje investigativo y práctico se puede concluir con lo siguiente:

- El repositorio de UNLIX 2.0 GNU/LINUX, permitió cubrir todas las actualizaciones de paquetes de las computadoras con este sistema conectadas a la red dentro del campus de la Universidad Nacional de Loja.
- Con el estudio realizado al sistema operativo UNLIX (en su primera versión) y al sistema operativo Debian Squeeze 6.0.1, se pudo determinar los paquetes de software más adecuados y actuales para UNLIX 2.0 GNU/LINUX.
- Tomando como base el repositorio oficial de Debian Squeeze, se consiguió tener un repositorio completo, actualizado y estable, para que sea manejado por los usuarios de UNLIX 2.0 GNU/LINUX.
- Gracias a la colaboración de la Unidad de Telecomunicaciones e Información, se pudo alojar el repositorio de paquetes en un servidor para los usuarios de UNLIX 2.0 GNU/LINUX dentro de la Universidad Nacional de Loja.
- El repositorio primario de paquetes en un CD/DVD, permitió tener una amplia disponibilidad de paquetes de software compatibles con UNLIX 2.0 y Debian para que sean usados en la actualización de estos sistemas en sitios sin conexión.
- UNLIX 2.0 GNU/LINUX como distribución que trabaja en modo Live Instalable, permitió ser ejecutado de forma segura y confiable en algunas arquitecturas (i386, amd64) de computadoras de algunos usuarios.



- Usando la herramienta Remastersys, se pudo comprobar que es potente a la hora de crear distribuciones y confiable para crear Lives CD/DVDs o Lives USBs con sistemas GNU/LINUX basados en Debian y sus derivados.
- El presente proyecto de tesis permitió cubrir una recomendación del proyecto de tesis UNLIX (primera versión), la misma que consistió en la construcción de un repositorio de paquetes para el sistema operativo UNLIX en una nueva versión.
- La culminación del presente proyecto de tesis, ayudó a tener un expertiz en el uso y administración de sistemas GNU/LINUX y de conocer la importancia de la filosofía del software libre.



## 11. RECOMENDACIONES

En base a los resultados obtenidos con el desarrollo del presente proyecto de investigación, se recomienda lo siguiente:

- Usar el repositorio de paquetes del servidor de UNLIX 2.0 para que puedan actualizar aplicaciones de forma rápida y segura.
- Estar periódicamente pendientes de informarse de las recientes versiones que lanzan los sistemas GNU/LINUX, ya que trae consigo por lo general nuevas mejoras y funcionalidades de sus paquetes de software.
- Para crear repositorios de sistemas basados en Debian, seguir la estructura del repositorio oficial, debido a que este mantiene soporte por una gran cantidad de desarrolladores a nivel mundial y por esta razón es actualizado periódicamente.
- Importante asignar una dirección de IP pública para que los usuarios fuera del campus de la Universidad Nacional de Loja accedan al repositorio de UNLIX 2.0 GNU/LINUX mediante el internet y así actualicen estos sistemas.
- Cuando no se tiene conexión a la red, usar el CD/DVD primario de paquetes para instalar en UNLIX 2.0 o Debian algún paquete que esté disponible en el CD/DVD.
- Considerar UNLIX 2.0 GNU/LINUX como sistema operativo oficial de la Universidad, ya que cubre los requerimientos de software de algunas carreras y como sistema libre está respaldado por el artículo 32 de la ley de educación Superior. (Ver anexo F).



- Utilizar la herramienta Remastersys, que es muy potente y confiable a la hora de crear distribuciones.
- Considerar la liberación del proyecto SGA (Sistema de Gestión Académica de la UNL) y todas las implicaciones que esto amerita, para que pueda ser considerado dentro del proyecto UNLIX en una futura versión.
- Estudiar y trabajar con UNLIX 2.0 GNU/LINUX y así hacer que el proyecto UNLIX continúe siendo mejorado.



## 12. BIBLIOGRAFÍA Y REFERENCIAS

### LIBROS:

- PÉREZ David, GINESTA Marc y Otros. 2007. Software Libre. Segunda Edición. Barcelona. 314 pp.

### TESIS:

- MORA Roberto, QUEZADA Maritza. Creación del Sistema Operativo UNLIX. Tesis de Grado. Loja – Ecuador, año 2009.

### SITIOS WEB:

- BOGDRAKE. 2010. Información sobre repositorios. [en línea]. [<http://blogdrake.net/que-es-un-repositorio>].
- DEBIAN. Página oficial de debían. Información general de Debian. [en línea], [<http://www.debian.org/index.es.html>].
- WIKIPEDIA, Enciclopedia Libre. Filosofía GNU/LINUX. [en línea], [<http://es.wikipedia.org/wiki/GNU>].
- WIKIPEDIA, Enciclopedia Libre. Información del proyecto LINUX. [en línea], [<http://es.wikipedia.org/wiki/GNU/Linux>].
- WIKIPEDIA, Enciclopedia Libre. 2010. Información sobre herramienta reconstructor. [en línea]. [<http://es.wikipedia.org/wiki/Reconstructor>].
- Información del proceso de remasterizar usando Ubuntu. En línea. [[http://files.nireblog.com/blogs/meth/files/remasterizar\\_ubuntu\\_.pdf](http://files.nireblog.com/blogs/meth/files/remasterizar_ubuntu_.pdf)].
- GARFIO, Página oficial de garfio. Información de la herramienta garfio. [en línea]. [<http://www.garfio.org.ar/>].
- Proyecto GNU/LINUX. [en línea], [<http://www.gnu.org/philosophy/free-sw.es.html>].



- Licencias de software libre. Softwarelicenses.2011. En línea. [<http://www.gnu.org/licenses/license-list.html#GPLCompatibleLicense>].
- Linux y el Proyecto GNU/LINUX. [en línea], [<http://www.gnu.org/gnu/linux-and-gnu.es.html>].
- Licencias Software libre 2011. En línea. [<http://www.informatica-hoy.com.ar/software-libre-gnu/Tiposde-licencia-de-Software.php>].
- Isotton Aaron. Información sobre repositorios en debían. En línea. [<http://myhomemd.com.ar/doc/howto-repositorios/howto-repositorios.pdf>].
- LINUX, Información de Soporte Linux Básico. [en línea], [<http://www.obtengalinux.org/linux/>].
- REMASTERSYS, Página oficial de remastersys. Información de la herramienta remastersys. [en línea]. [<http://remastersys.sourceforge.net/>].



# 13. ANEXOS



## ANEXO A: GLOSARIO DE TÉRMINOS

**Alien.-** Es un paquete de software que sirve para transformar paquetes con extensión rpm o paquetes comprimidos a la extensión .deb.

**Archivo .deb.-** Es un archivo que tiene el formato con la extensión deb de los paquetes de software de Debian, llevan ese nombre porque es en honor al nombre Deborah Murdock que es la ex esposa del fundador de la distribución de debian Ian Murdock.” Los paquetes deb son archivos ar estándar de Unix que incluyen dos archivos tar en formato gzip, bzip2 o lzma: uno de los cuales alberga la información de control y el otro los datos. Un paquete deb también puede ser convertido a otros formato de paquete y viceversa usando la aplicación alien.”<sup>23</sup>

**Archivo .iso.-** Esta clase de archivos son denominados así por la norma ISO 9660 de la Organización Internacional para la Estandarización, ya que son unos archivos que agrupa un sistema de archivos cualquiera y construye una copia del mismo y esta copia es la que se puede almacenar en cualquier medio sin sufrir variaciones en su estructura, por eso es que se pueden hacer copias de discos duros, cds, memorias USB, diskettes. Por eso son ideales para distribuir sistemas operativos, es así que en la internet los archivos .iso son la clase de archivos que se encuentran para los diferentes sistemas operativos ahí cargados listos para su descarga vía online.

**Archivo .sh.-** Estos archivos son un conjunto de sentencias de código que forman un bloque y hacen una tarea determinada y se lo puede ejecutar, se los guarda con extensión .sh porque es la programación del shell, también se los denomina scripts.

**Backups.-** Es un conjunto de datos en un archivo que es una copia de seguridad de otro, es muy usado para respaldar base de datos o repositorios. Por la necesidad misma de conservar los datos para su respectivo uso.

---

<sup>23</sup> Wikipedia Archivo .deb.2011.[en línea],[ <http://es.wikipedia.org/wiki/Deb>], [Consulta: 28 de mayo del 2011]



**Bash.-** Es un programa informático que interpreta ordenes está basado en la Shell de Unix y este programa fue creado para interpretar los comandos en los sistemas GNU/LINUX.

**Boot.-** Es la secuencia de arranque en ingles Boot o Booting que se ve su presencia cuando inicia un sistema operativo en una computadora y esta se encarga de la inicialización del sistema y de sus dispositivos.

**Bootear.-** Es inicializar el sistema con los parámetros definidos previa configuración establecida.

**Booteo.-** Acción de bootear.

**Bugs.-** Es un error en la codificación, estos errores son llevados a control por los desarrolladores para corregirlos y por eso se pide él envío de estos para su posterior análisis y para que versiones nuevas del programa ya no tengan los mismo bug en su estructura.

**Busybox.-** *“En computación, BusyBox es una programa que combina muchas utilidades estándares de Unix en un solo ejecutable pequeño. Es capaz de proveer la mayoría de las utilidades que están especificadas para los sistemas Unix además de muchas de las utilidades que esperas ver en los sistemas GNU/Linux. Busybox es utilizada normalmente en sistemas que funcionen desde un disco flexible o en sistemas con Linux embebido”*<sup>24</sup>

**Cliente terminal.-** Es una computadora que está conectada a una red y tiene acceso a un servicio y es donde alguna aplicación tiene la interfaz de acceso como cliente o usuario.

---

<sup>24</sup> Wikipedia Busybox.2011.[en línea],[ <http://es.wikipedia.org/wiki/Busybox>], [Consulta: 27 de mayo del 2011]



**Consola o terminal.-** Es el entorno de escritura de comandos en texto de los sistemas GNU/LINUX, se la denomina terminal porque es la última interfaz donde se puede acceder al sistema con modo texto desde el entorno gráfico.

**Cron.-** Es el proceso donde se ejecutan las tareas periódicamente, la configuración de la periodicidad de ejecución de estas tareas se hace en el archivo crontab en los sistemas GNU/LINUX.

**Customizacion.-** Es la adaptación de una herramienta u objeto a las preferencias del usuario o propietario.

**Descargar.-** En el campo de la informática descargar es la acción de bajar un archivo que no reside en un determinado computador pero que si lo alberga otro que está en red con el primero y este permite el acceso para transferir el archivo para su posterior uso.

**Directorio.-** Un directorio es una agrupación de archivo de datos que tienen una categoría u propósito, pueden ser de cualquier carácter agrupados para diferente fin que decida el usuario.

**Distro.-** Es un término que se usa para denominarlas a las distribuciones de los sistemas GNU/LINUX.

**DNS.-** *“Domain Name System o DNS (en español: sistema de nombres de dominio) es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a Internet o a una red privada. Este sistema asocia información variada con nombres de dominios asignado a cada uno de los participantes. Su función más importante, es traducir (resolver) nombres inteligibles para los humanos en identificadores binarios asociados con los equipos conectados a la red, esto con el propósito de poder localizar y direccionar estos equipos mundialmente.”*<sup>25</sup>

---

<sup>25</sup> Wikipedia DNS.2011.[en línea],[ <http://es.wikipedia.org/wiki/DNS>], [Consulta: 28 de mayo del 2011]



**Dpkg.-** Es un programa base para la gestión de los paquetes en los sistemas GNU/LINUX,” Fue creado por Ian Jackson en 1993. Se utiliza para instalar, quitar, y proporcionar información sobre los paquetes .deb.”<sup>26</sup>

**Ejecutar.-** En el campo de la informática es la acción de iniciar la carga de un programa informático o de cualquier archivo ejecutable, en inglés es run, execute en español correr.

**Expertiz.-** *“La palabra expertiz se usa con un significado muy similar al de ‘pericia’, aunque a veces con el matiz añadido de estar asociado a un entorno de trabajo concreto en él se aplican técnicas de saber hacer y se emplean herramientas informáticas concretas. Probablemente sea un calco del inglés expertise. En frases como esa es mi expertiz equivale a ‘especialidad’. En casos como tiene un amplio expertiz en el sistema operativo equivale a ‘experiencia’.*

*Se usa tanto en femenino como en masculino, aunque parece que la expertiz tiene algo más de presencia que el expertiz.”*<sup>27</sup>

**Frenode.-** *“Freenode (nodo libre) es una red de servidores IRC orientado al software libre. En él se encuentran canales de proyectos de código abierto en diferentes idiomas”*<sup>28</sup>

**Gnewsense.-** “gNewSense es un sistema operativo GNU/Linux basado en Debian y Ubuntu. Fue diseñado con la meta de proporcionar una distribución 100% libre a los usuarios que desean utilizar solamente software libre. Recibe apoyo oficial de la Free

---

<sup>26</sup> Wikipedia Dpkg.2011.[en línea],[ <http://es.wikipedia.org/wiki/Dpkg>], [Consulta: 2 de junio del 2011]

<sup>27</sup> Wikilengua definiciones.2011.[en línea],[ <http://www.wikilengua.org/index.php/expertiz>], [Consulta: 3 de junio del 2011]

<sup>28</sup> Wikipedia Freenode.2011.[en línea],[ <http://es.wikipedia.org/wiki/Freenode>], [Consulta: 3 de junio del 2011]



Software Foundation y es el sistema operativo que usa Richard Stallman.”<sup>29</sup>

**Gnome.-** *“Es un entorno de escritorio e infraestructura de desarrollo para sistemas operativos Unix y derivados Unix como GNU/Linux, BSD o Solaris; compuesto enteramente de software libre.”*<sup>30</sup>

**Ip.-** *“Una dirección IP es una etiqueta numérica que identifica, de manera lógica y jerárquica, a un interfaz (elemento de comunicación/conexión) de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP (Internet Protocol), que corresponde al nivel de red del protocolo TCP/IP. Dicho número no se ha de confundir con la dirección MAC que es un identificador de 48bits para identificar de forma única a la tarjeta de red y no depende del protocolo de conexión utilizado ni de la red. La dirección IP puede cambiar muy a menudo por cambios en la red o porque el dispositivo encargado dentro de la red de asignar las direcciones IP, decida asignar otra IP (por ejemplo, con el protocolo DHCP), a esta forma de asignación de dirección IP se denomina dirección IP dinámica (normalmente abreviado como IP dinámica).”*<sup>31</sup>

**IRC.-** Internet Relay Chat es un protocolo de comunicación en tiempo real basado en texto que permite la comunicación entre dos o más personas.

**KDE.-** Es un proyecto de Software libre para tener un entorno de escritorio propio para los sistemas GNU/LINUX, Mac OS, X Windows, etc.

**Laptop.-** Es un computador personal portátil.

---

<sup>29</sup> Wikipedia GnewSense.2011.[en línea],[ <http://es.wikipedia.org/wiki/GNewSense>], [Consulta: 3 de junio del 2011]

<sup>30</sup> Wikipedia Gnome.2011.[en línea],[ <http://es.wikipedia.org/wiki/GNOME>], [Consulta: 3 de junio del 2011]

<sup>31</sup> Wikipedia Dirección IP.2011.[en línea],[ [http://es.wikipedia.org/wiki/Dirección\\_IP](http://es.wikipedia.org/wiki/Dirección_IP)], [Consulta: 3 de junio del 2011]



**Levantar.-** En informática levantar un programa es la acción de desplegar la interfaz de usuario para poder utilizar dicho programa, levantar también es como ejecutar un programa.

**Licencia.-** Es un archivo de texto que explica el origen o dueño de tal o cual objeto o programa, permite la legalización del mismo con todas sus restricciones de uso, es un contrato entre el usuario y el creador del programa u objeto para definir su uso.

**Live CD/DVD.-** Es un sistema operativo cargado en un CD o DVD y desde el cual puede ejecutarse sin problemas en una computadora cargándose en la RAM de la misma por lo que se los denomina CD o DVD vivos, porque se ejecutan sin la necesidad de instalarlos.

**Mainframes.-** *“Una computadora central o mainframe es una computadora grande, potente y costosa usada principalmente por una gran compañía para el procesamiento de una gran cantidad de datos; por ejemplo, para el procesamiento de transacciones bancarias”*<sup>32</sup>

**Medio.-** En informática es un modo por el cual se puede transmitir datos almacenados en este caso existen CDs, Discos, etc.

**Metapaquete.-** Es la agrupación de varios paquetes pero que tienen un solo lanzador de instalación, es decir se pueden instalar los paquetes en un solo grupo.

**Mirror.-** En español es espejo, pero en informática se dice este término a un servidor creado a imagen y semejanza de otro ubicado en otra dirección de la red, y que sirve para lo mismo que el original.

---

<sup>32</sup> Wikipedia Mainframes.2011.[en línea],[ <http://es.wikipedia.org/wiki/Mainframes>], [Consulta: 4 de junio del 2011]



**Montar.-** *“Se denomina montar, en informática, a la acción de integrar un sistema de archivos alojado en un determinado dispositivo dentro del árbol de directorios de un sistema operativo.”*<sup>33</sup>

**Parche.-** *“En informática, un parche consta de cambios que se aplican a un programa, para corregir errores, agregarle funcionalidad, actualizarlo, etc.*

*Si bien los parches suelen ser desarrollados por programadores ajenos a los autores iniciales del proyecto, esto no siempre es así. Un parche puede ser aplicado tanto a un binario ejecutable como al código fuente de cualquier tipo de programa, incluso, un sistema operativo.*

*El origen del nombre probablemente se deba a la utilidad de Unix llamada patch creada por Larry Wall.”*<sup>34</sup>

**Pc.-** Es un computador por la denominación en inglés de Personal Computer o en español computadora personal, y es así como se ha generalizado el nombre para denominar a una computadora.

**Plugins.-** *“Un complemento es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API. También se lo conoce como plug-in (del inglés "enchufable"), add-on (agregado), complemento, conector o extensión.”*<sup>35</sup>

**Putty.-** Es un programa cliente SSH disponible para Windows y Linux, se puede hacer Telnet, rlogin y TCP raw, es de licencia libre.

---

<sup>33</sup> Wikipedia Montar en informática.2011.[en línea],[ [http://es.wikipedia.org/wiki/Montar\\_\(informática\)](http://es.wikipedia.org/wiki/Montar_(informática))], [Consulta: 4 de junio del 2011]

<sup>34</sup> Wikipedia Parche en informática.2011.[en línea],[ [http://es.wikipedia.org/wiki/Parche\\_\(informática\)](http://es.wikipedia.org/wiki/Parche_(informática))], [Consulta: 4 de junio del 2011]

<sup>35</sup> Wikipedia Plugins.2011.[en línea],[ <http://es.wikipedia.org/wiki/Plugins>], [Consulta: 4 de junio del 2011]



**Script.-** Es un conjunto de sentencias de código que realizan una tarea determinada dentro del sistema y que pueden ser almacenada para su posterior ejecución, la extensión para guardar un script es la .sh , en realidad un script es un archivo de comandos.

**Scripting.-** Es programar scripts es decir hacer programas con el código que viene en forma original sin la necesidad de compilación, tal como se lo escribe se lo ejecuta, es por esto que su código fuente es en forma de texto y son interpretados comando por comando aunque también a un script se lo puede compilar pero no es necesario ni es muy usada esta opción.

**Servicio SSH.-** *“(Secure SHell). Protocolo seguro y un conjunto de herramientas para reemplazar otras más comunes (inseguras). Fue diseñado desde el principio para ofrecer un máximo de seguridad y permitir el acceso remoto a servidores de forma segura.”*<sup>36</sup>

**Sources.list.-** Es el archivo del sistema de Debian y derivados en donde se aloja las direcciones de acceso a los repositorios, este se encuentra en /etc/apt/sources.list.

**Squashfs.-** *“Es un sistema de archivos comprimido de sólo lectura para Linux. SquashFS comprime archivos, inodos y directorios, y soporta tamaños de bloque de hasta 1024 KB para mayor compresión. SquashFS es también software libre (licenciado como GPL) para acceder a sistemas de archivos SquashFS”*<sup>37</sup>

**Urpmi.-** *“Es una herramienta de administración de paquetes que se encarga de instalar, eliminar, actualizar y hacer preguntas a los paquetes de software de aplicaciones locales o remotas (vía red). Envuelve al administrador de paquetes RPM (RedHat Package Manager) extendiendo las funcionalidades que otorga. Uno de sus principales*

---

<sup>36</sup> Mastermagazine SSH.2011.[en línea],[ <http://www.mastermagazine.info/termino/6775.php>], [Consulta: 5 de junio del 2011]

<sup>37</sup> Wikipedia Squashfs.2011.[en línea],[ <http://es.wikipedia.org/wiki/SquashFS>], [Consulta: 5 de junio del 2011]



*fin es de facilitar la instalación de programas con herramientas que desprenden los problemas comunes sobre ausencia de dependencias. URPMI Trabaja con las fuentes oficiales de Mandriva Linux o con fuentes externas tales como las que otorga Penguin Liberation Front y otros repositorios no oficiales.”<sup>38</sup>*

**Versión.-** Es la mejora que se hace a un programa informático y se hace su lanzamiento en determinada fecha para indicar la estabilidad de dicho software, por ejemplo la versión beta, que se dice cuando está el software en modo de prueba.

**Webmin.-** *“Es una herramienta de configuración de sistemas accesible vía web para OpenSolaris, GNU/Linux y otros sistemas Unix. Con él se pueden configurar aspectos internos de muchos sistemas operativos, como usuarios, cuotas de espacio, servicios, archivos de configuración, apagado del equipo, etcétera, así como modificar y controlar muchas aplicaciones libres, como el servidor web Apache, PHP, MySQL, DNS, Samba, DHCP, entre otros.”<sup>39</sup>*

---

<sup>38</sup> Wikipedia Urpmi.2011.[en línea],[ <http://es.wikipedia.org/wiki/Urpmi>], [Consulta: 5 de junio del 2011]

<sup>39</sup> Buenas tareas Webmin.2011.[en línea],[ <http://www.buenastareas.com/ensayos/Webmin/364737.html>], [Consulta: 6 de junio del 2011]



**ANEXO B: MODELO DE ENCUESTA REALIZADAS EN LAS PRUEBAS**

**UNIVERSIDAD NACIONAL DE LOJA  
ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO  
RENOVABLES  
CARRERA DE INGENIERÍA DE SISTEMAS  
ENCUESTA SOBRE UNLIX 2.0**

Estimado Señor (a), de la manera más comedida le solicitamos responder la siguiente encuesta, los datos proporcionados aquí serán utilizados únicamente con fines de investigación académica, por lo cual le solicitamos su sinceridad en cada respuesta.

**Datos del encuestado:**

**Nombre:**.....

**Cargo que desempeña:**.....

**Fecha:**.....

1. ¿El sistema operativo Unix 2.0 a su parecer tiene buen tiempo de respuesta al momento de iniciar en el computador en relación al rendimiento de su máquina?

MUY BUENO ( )      BUENO ( )      MALO ( )      REGULAR ( )

2. ¿Unix 2.0 se levantó sin problemas en su computador?

SI ( )      NO ( )

Por qué:.....  
.....

3. ¿Piensa usted que Unix 2.0 tiene buena accesibilidad en cuanto al manejo de su interfaz de escritorio?

MUY BUENO ( )      BUENO ( )      MALO ( )      REGULAR ( )

4. ¿Cómo califica la personalización de Unix 2.0 le parece acorde a los colores que identifican a la Universidad Nacional de Loja?

MUY BUENO ( )      BUENO ( )      MALO ( )      REGULAR ( )

5. ¿Trabajar con el sistema Unix 2.0 le parece?

MUY DIFÍCIL ( )      DIFÍCIL ( )      SENCILLO ( )      MUY FÁCIL ( )



6. ¿Piensa usted que se debe tener conocimientos avanzados de computación para utilizar Unix 2.0?

SI ( ) NO ( )

PORQUE.....  
.....

7. ¿Cómo califica el rendimiento de Unix 2.0 a la hora de levantar algún programa dentro del sistema?

MUY BUENO ( ) BUENO ( ) MALO ( ) REGULAR ( )

8. ¿Cree usted que Unix 2,0 es confiable, seguro y estable?

SI ( ) NO ( )

PORQUE.....  
.....

9. ¿Considera que Unix 2.0 le ayuda a resolver sus actividades dentro de su ambiente de trabajo con la utilización de algún programa o aplicación que trae este sistema?

SI ( ) NO ( ) NO COMPLETAMENTE ( )

10. ¿Unix 2.0 si cumple con los requerimientos y expectativas como usuario, es decir es para usted lo que esperaba de un sistema operativo?

Si ( ) NO ( )

OBSERVACIONES.....  
.....  
.....  
.....  
.....

**Gracias por su colaboración**



### ANEXO C: ENCUESTAS REALIZADAS



UNIVERSIDAD NACIONAL DE LOJA  
ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO  
RENOVABLES  
CARRERA DE INGENIERÍA DE SISTEMAS  
ENCUESTA SOBRE UNLIX 2.0

Estimado Señor (a), de la manera más comedida le solicitamos responder la siguiente encuesta, los datos proporcionados aquí serán utilizados únicamente con fines de investigación académica, por lo cual le solicitamos su sinceridad en cada respuesta.

**Datos del encuestado:**

Nombre: Shon Alexander Calderin Samorin

Cargo que desempeña: Tecnico Sección Redes

Fecha: 29/04/2011

1. ¿El sistema operativo Unix 2.0 a su parecer tiene buen tiempo de respuesta al momento de iniciar en el computador en relación al rendimiento de su máquina?

MUY BUENO (X) BUENO ( ) MALO ( ) REGULAR ( )

2. ¿Unix 2.0 se levantó sin problemas en su computador?

SI (X) NO ( )

Por qué:.....  
.....  
.....

3. ¿Piensa usted que Unix 2.0 tiene buena accesibilidad en cuanto al manejo de su interfaz de escritorio?.

MUY BUENO (X) BUENO ( ) MALO ( ) REGULAR ( )

4. ¿Cómo califica la personalización de Unix 2.0 le parece acorde a los colores que identifican a la Universidad Nacional de Loja?

MUY BUENO ( ) BUENO (X) MALO ( ) REGULAR ( )

5. ¿Trabajar con el sistema Unix 2.0 le parece?

MUY DIFÍCIL ( ) DIFÍCIL ( ) SENCILLO ( ) MUY FÁCIL (X)

a



6. ¿Piensa usted que se debe tener conocimientos avanzados de computación para utilizar Unix 2.0?

SI ( ) NO (X)

PORQUE *Fácil y amigable de utilizar para usuarios novatos.*

7. ¿Cómo califica el rendimiento de Unix 2.0 a la hora de levantar algún programa dentro del sistema?

MUY BUENO (X) BUENO ( ) MALO ( ) REGULAR ( )

8. ¿Cree usted que Unix 2,0 es confiable, seguro y estable?

SI (X) NO ( )

PORQUE.....

9. ¿Considera que Unix 2.0 le ayuda a resolver sus actividades dentro de su ambiente de trabajo con la utilización de algún programa o aplicación que trae este sistema?

SI (X) NO ( ) NO COMPLETAMENTE ( )

10. ¿Unix 2.0 si cumple con los requerimientos y expectativas como usuario, es decir es para usted lo que esperaba de un sistema operativo?

SI (X) NO ( )

OBSERVACIONES.....  
.....  
.....  
.....  
.....  
.....  
.....

Gracias por su colaboración

b



UNIVERSIDAD NACIONAL DE LOJA  
ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO  
RENOVABLES  
CARRERA DE INGENIERÍA DE SISTEMAS  
ENCUESTA SOBRE UNLIX 2.0

Estimado Señor (a), de la manera más comedida le solicitamos responder la siguiente encuesta, los datos proporcionados aquí serán utilizados únicamente con fines de investigación académica, por lo cual le solicitamos su sinceridad en cada respuesta.

Datos del encuestado:

Nombre: Julio César Idrobo Contento

Cargo que desempeña: Ing. en Mantenimiento

Fecha: 19 de Abril del 2011

1. ¿El sistema operativo Unix 2.0 a su parecer tiene buen tiempo de respuesta al momento de iniciar en el computador en relación al rendimiento de su máquina?

MUY BUENO ( ) BUENO (x) MALO ( ) REGULAR ( )

2. ¿Unix 2.0 se levantó sin problemas en su computador?

SI (x) NO ( )

Por qué: tengo una maquina de buenas características

3. ¿Piensa usted que Unix 2.0 tiene buena accesibilidad en cuanto al manejo de su interfaz de escritorio?.

MUY BUENO (x) BUENO ( ) MALO ( ) REGULAR ( )

4. ¿Cómo califica la personalización de Unix 2.0 le parece acorde a los colores que identifican a la Universidad Nacional de Loja?

MUY BUENO (x) BUENO ( ) MALO ( ) REGULAR ( )

5. ¿Trabajar con el sistema Unix 2.0 le parece?

MUY DIFÍCIL ( ) DIFÍCIL ( ) SENCILLO ( ) MUY FÁCIL (x)



6. ¿Piensa usted que se debe tener conocimientos avanzados de computación para utilizar Unix 2.0?

SI ( ) NO (x)

PORQUE... *es muy sencillo* .....

7. ¿Cómo califica el rendimiento de Unix 2.0 a la hora de levantar algún programa dentro del sistema?

MUY BUENO (x) BUENO ( ) MALO ( ) REGULAR ( )

8. ¿Cree usted que Unix 2,0 es confiable, seguro y estable?

SI (x) NO ( )

PORQUE... *trabaja un sistema de seguridad* .....

9. ¿Considera que Unix 2.0 le ayuda a resolver sus actividades dentro de su ambiente de trabajo con la utilización de algún programa o aplicación que trae este sistema?

SI (x) NO ( ) NO COMPLETAMENTE ( )

10. ¿Unix 2.0 si cumple con los requerimientos y expectativas como usuario, es decir es para usted lo que esperaba de un sistema operativo?

Si (x) NO ( )

OBSERVACIONES... *Felicitaciones es una herramienta muy confiable* .....

*Felicidades*  
Gracias por su colaboración





UNIVERSIDAD NACIONAL DE LOJA  
ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO  
RENOVABLES  
CARRERA DE INGENIERÍA DE SISTEMAS  
ENCUESTA SOBRE UNLIX 2.0

Estimado Señor (a), de la manera más comedida le solicitamos responder la siguiente encuesta, los datos proporcionados aquí serán utilizados únicamente con fines de investigación académica, por lo cual le solicitamos su sinceridad en cada respuesta.

Datos del encuestado:

Nombre: Ing. Milton Labanda

Cargo que desempeña: Docente Sistemas UNL

Fecha: 29-04-2011

1. ¿El sistema operativo Unix 2.0 a su parecer tiene buen tiempo de respuesta al momento de iniciar en el computador en relación al rendimiento de su máquina?

MUY BUENO ( ) BUENO (X) MALO ( ) REGULAR ( )

2. ¿Unix 2.0 se levantó sin problemas en su computador?

SI (X) NO ( )

Por qué: lo ~~se~~ ejecuta en una máquina virtual.

3. ¿Piensa usted que Unix 2.0 tiene buena accesibilidad en cuanto al manejo de su interfaz de escritorio?

MUY BUENO ( ) BUENO (X) MALO ( ) REGULAR ( )

4. ¿Cómo califica la personalización de Unix 2.0 le parece acorde a los colores que identifican a la Universidad Nacional de Loja?

MUY BUENO (X) BUENO ( ) MALO ( ) REGULAR ( )

5. ¿Trabajar con el sistema Unix 2.0 le parece?

MUY DIFÍCIL ( ) DIFÍCIL ( ) SENCILLO (X) MUY FÁCIL ( )

P



6. ¿Piensa usted que se debe tener conocimientos avanzados de computación para utilizar Unix 2.0?

SI ( ) NO (X)

PORQUE..... *Es relativamente sencillo* .....

7. ¿Cómo califica el rendimiento de Unix 2.0 a la hora de levantar algún programa dentro del sistema?

MUY BUENO ( ) BUENO (X) MALO ( ) REGULAR ( )

8. ¿Cree usted que Unix 2,0 es confiable, seguro y estable?

SI (X) NO ( )

PORQUE..... *está basado en Debian.* .....

9. ¿Considera que Unix 2.0 le ayuda a resolver sus actividades dentro de su ambiente de trabajo con la utilización de algún programa o aplicación que trae este sistema?

SI ( ) NO ( ) NO COMPLETAMENTE (X)

10. ¿Unix 2.0 si cumple con los requerimientos y expectativas como usuario, es decir es para usted lo que esperaba de un sistema operativo?

Si (X) NO ( )

OBSERVACIONES..... *Requeria jython y + programas relacionados con algunas otras carreras de la universidad ya que pensaba a priori que esta orientada a la Universidad Nacional de Loja.* .....

Gracias por su colaboración

9



UNIVERSIDAD NACIONAL DE LOJA  
ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO  
RENOVABLES  
CARRERA DE INGENIERÍA DE SISTEMAS  
ENCUESTA SOBRE UNLIX 2.0

Estimado Señor (a), de la manera más comedida le solicitamos responder la siguiente encuesta, los datos proporcionados aquí serán utilizados únicamente con fines de investigación académica, por lo cual le solicitamos su sinceridad en cada respuesta.

Datos del encuestado:

Nombre: Karina Vásquez

Cargo que desempeña: Estudiante

Fecha: 23-04-2011

1. ¿El sistema operativo Unix 2.0 a su parecer tiene buen tiempo de respuesta al momento de iniciar en el computador en relación al rendimiento de su máquina?

MUY BUENO ( ) BUENO (X) MALO ( ) REGULAR ( )

2. ¿Unix 2.0 se levantó sin problemas en su computador?

SI (X) NO ( )

Por qué: .....

3. ¿Piensa usted que Unix 2.0 tiene buena accesibilidad en cuanto al manejo de su interfaz de escritorio?

MUY BUENO ( ) BUENO (X) MALO ( ) REGULAR ( )

4. ¿Cómo califica la personalización de Unix 2.0 le parece acorde a los colores que identifican a la Universidad Nacional de Loja?

MUY BUENO ( ) BUENO (X) MALO ( ) REGULAR ( )

5. ¿Trabajar con el sistema Unix 2.0 le parece?

MUY DIFÍCIL ( ) DIFÍCIL ( ) SENCILLO (X) MUY FÁCIL ( ) b1



6. ¿Piensa usted que se debe tener conocimientos avanzados de computación para utilizar Unix 2.0?

SI (✓) NO ( )

PORQUE... *Un poco para tener conocimiento básico* .....

7. ¿Cómo califica el rendimiento de Unix 2.0 a la hora de levantar algún programa dentro del sistema?

MUY BUENO ( ) BUENO (✓) MALO ( ) REGULAR ( )

8. ¿Cree usted que Unix 2.0 es confiable, seguro y estable?

SI (✓) NO ( )

PORQUE.....

9. ¿Considera que Unix 2.0 le ayuda a resolver sus actividades dentro de su ambiente de trabajo con la utilización de algún programa o aplicación que trae este sistema?

SI (✓) NO ( ) NO COMPLETAMENTE ( )

10. ¿Unix 2.0 si cumple con los requerimientos y expectativas como usuario, es decir es para usted lo que esperaba de un sistema operativo?

SI (✓) NO ( )

OBSERVACIONES.....  
.....  
.....  
.....  
.....  
.....  
.....

Gracias por su colaboración

CS



## **ANEXO D: CERTIFICACIÓN DE LOS DESARROLLADORES DE UNIX (PRIMERA VERSIÓN)**

Celica, 12 de noviembre de 2010

### **CERTIFICACION**

Nosotros, **Maritza del Carmen Quezada Sarango** y **Roberto Carlos Mora Quezada**, desarrolladores del proyecto de tesis cuyo tema versa “**DESARROLLO DE UNA DISTRIBUCIÓN LINUX BASADA EN DEBIAN PARA EL ÁREA DE ENERGÍA, LAS INDUSTRIAS Y RECURSOS NATURALES NO RENOVABLES DE LA UNL**”, certificamos que los egresados **Cesar A. Silverio Mendieta** y **Víctor Ignacio Cueva Cango**, realizaron una entrevista sobre el analisis, diseño y desarrollo de nuestro proyecto, debido a que estan desarrollando su proyecto de tesis cuyo tema versa "CREACIÓN DEL REPOSITORIO DE PAQUETES Y ACTUALIZACIÓN DEL SISTEMA OPERATIVO UNIX "

Es cuanto podemos certificar en honor de la verdad, autorizo a los interesados hacer uso del presente cuando lo crean conveniente.

Atentamente



Ingeniera. **Maritza del Carmen Quezada**



Ingeniero. **Roberto Carlos Mora**



## ANEXO E: LICENCIA DE UNLIX 2.0 GNU/LINUX

Open License Publicación

v1.0, 08 de junio 1999

### I. REQUISITOS DE LAS DOS VERSIONES sin modificar y MODIFICADOS

Las obras de Publicación Abierta puede ser reproducido y distribuido en su totalidad o en parte, en cualquier medio físico o electrónico, siempre que los términos de esta licencia se cumplen, y que esta licencia o una incorporación de la misma por la referencia (con las opciones elegidas por el autor (s) y / o el editor) se muestra en la reproducción.

Copyright (c) <2011> por <Universidad Nacional de Loja>. Este material puede ser distribuido únicamente bajo los términos y condiciones establecidos en la Licencia de Publicación Abierta, v1.0 o posterior (la última versión está disponible en <http://www.opencontent.org/openpub/>). "La distribución de versiones modificadas de este documento queda prohibida sin el permiso explícito del titular del copyright."

La redistribución comercial de material de Publicación Abierta con licencia está permitida.

Cualquier publicación en forma de libro estándar (papel), exigirá la cita de la editorial original y el autor. Los nombres de los autores y los editores deberán figurar en todas las superficies externas del libro. En todas las superficies externas del libro el nombre del editor original será tan grande como el título de la obra y ha sido citado como posesivo con respecto al título.



## II. DERECHOS DE AUTOR

Los derechos de autor a cada publicación abierta es propiedad de su autor (s) o su designado.

## III. ALCANCE DE LA LICENCIA

Los términos de licencia se aplican a todos los trabajos de publicación abierta, a menos que expresamente se indica en el documento.

El simple hecho de abrir las obras de publicación o de una parte de una obra de Publicación Abierta con otras obras o programas en los medios de comunicación, no serán causa de esta licencia se aplique a las otras obras. El trabajo conjunto deberá contener una nota que indique la inclusión del material de Publicación Abierta y el aviso de copyright.

**DIVISIBILIDAD.** Si alguna parte de esta licencia se encuentra para ser inaplicable en cualquier jurisdicción, las partes restantes de la licencia permanecerán en vigor.

**SIN GARANTÍA.** Abra la publicación de obras con licencia y proporciona "tal cual" sin garantía de ningún tipo, expresa o implícita, incluyendo pero no limitado a, las garantías implícitas de comerciabilidad o idoneidad para un propósito particular o de una garantía de no infracción.

## IV. REQUISITOS DE LAS OBRAS MODIFICADOS

Todas las versiones modificadas de los documentos cubiertos por esta licencia, incluyendo traducciones, antologías, compilaciones y documentos parciales, deben cumplir los siguientes requisitos:

La versión modificada debe ser etiquetada como tal.



La persona que efectúe las modificaciones deben ser identificados y las modificaciones de fecha.

Acuse de recibo del original autor y el editor en su caso debe conservarse de acuerdo con las prácticas normales de la cita académica.

La ubicación del documento original sin modificaciones debe ser identificado.

El autor original (o autores) nombre (s) no puede ser utilizado para afirmar o implicar aprobación del documento resultante sin que el autor original (o autores) permiso.

## V. BUENAS PRÁCTICAS RECOMENDACIONES

Además de los requisitos de la presente licencia, se solicita y recomienda encarecidamente que los redistribuidores de:

Si va a distribuir de Publicación Abierta trabaja en papel o CD-ROM, se proporciona notificación por correo electrónico a los autores de su intención de redistribuir por lo menos treinta días antes de su manuscrito o congelación de los medios de comunicación, para dar tiempo a los autores para proporcionar documentos actualizados. Esta notificación debe describir las modificaciones, en su caso, realizados en el documento.

Todas las modificaciones de fondo (incluso pérdidas) ser claramente marcados en el documento o bien describen en un anexo al documento.

Por último, si bien no es obligatoria en virtud de esta licencia, se considera una buena forma para ofrecer una copia gratuita de cualquier papel y la expresión de CD-ROM de una obra de Publicación Abierta con licencia a su autor (s).



## **ANEXO F: LEY ORGÁNICA DE EDUCACIÓN SUPERIOR**

**Registro Oficial N° 298 -- martes 12 de octubre del 2010**

**.. Art. 31.- De los legados o donaciones.-** Los legados que realicen las personas naturales y las donaciones que efectúen las personas jurídicas o naturales a las instituciones de educación superior, al Consejo de Educación Superior, o al Consejo de Evaluación, Acreditación y Aseguramiento de la Calidad de la Educación Superior, estarán exonerados de los impuestos correspondientes.

Los bienes que hayan sido transferidos por donación o legados se incorporarán al patrimonio de las instituciones de educación superior, y podrán ser enajenados exclusivamente para mantener o incrementar el patrimonio de la institución beneficiaria de la donación, o podrán ser donados a otras instituciones de educación superior públicas o particulares, según lo previsto en esta Ley y la reglamentación que para el efecto expida el Consejo de Educación Superior.

Cuando no se haya establecido por parte del donante o legatario el destino de la donación, los recursos obtenidos por este concepto deberán destinarse únicamente a inversiones en infraestructura, recursos bibliográficos, equipos, laboratorios, cursos de pregrado y posgrado, formación y capacitación de profesores o profesoras y para financiar proyectos de investigación. La Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación velará por el cumplimiento de esta disposición.

**Art. 32.- Programas informáticos.-** Las empresas que distribuyan programas informáticos tienen la obligación de conceder tarifas preferenciales para el uso de las licencias obligatorias de los respectivos programas, a favor de las instituciones de educación superior, para fines académicos.

Las instituciones de educación superior obligatoriamente incorporarán el uso de programas informáticos con software libre.

**Art. 33.- Acreditación de rentas.-** El Ministerio de Finanzas dispondrá la acreditación automática de las rentas establecidas a favor de las instituciones de régimen público y particular que reciben asignaciones y rentas del Estado, de conformidad con la Ley.

**Art. 34.- Endeudamiento público de las instituciones de educación superior públicas.-** Las instituciones de educación superior públicas pueden contraer endeudamiento público cumpliendo las disposiciones de la Constitución y la Ley correspondiente. El endeudamiento únicamente puede ser usado para programas y proyectos de inversión, para infraestructura y equipamiento, con criterios de mejoramiento de la calidad.

.....



**ANEXO G: CERTIFICACIÓN DE LA UNIDAD DE TELECOMUNICACIONES E  
INFORMACIÓN**

**UNIVERSIDAD NACIONAL DE LOJA**  
UNIDAD TELECOMUNICACIONES E INFORMACIÓN

LIC. JAMIL RAMÓN CARRIÓN

**CERTIFICA:**

Que los señores egresados de la Carrera de Ingeniería en Sistemas: César Augusto Silverio Mendieta y Víctor Ignacio Cueva Cango, realizaron las pruebas del Proyecto denominado “CREACIÓN DEL REPOSITORIO DE PAQUETES Y ACTUALIZACIÓN DEL SISTEMA OPERATIVO UNIX.”, indicando que el Servidor se encuentra en la Unidad de Telecomunicaciones e Información, el mismo que se encuentra funcionando y listo para su explotación.

Es cuanto puedo certificar en honor a la verdad.

Loja, 13 de junio de 2011

Lo certifica,



Lic. Jamil Ramón C.  
**DIRECTOR TELECOMUNICACIONES E INFORMACIÓN**





**ANEXO H: CERTIFICACIONES DE CONFERENCIA DE UNLIX 2.0 GNU/LINUX**



**UNIVERSIDAD NACIONAL DE LOJA**  
**CARRERA INFORMÁTICA EDUCATIVA**  
**ASOCIACIÓN CARRERA INFORMÁTICA EDUCATIVA**

Confiere el presente:

**CERTIFICADO**

**R: Sr. Cesar Silverio**

Por haber participado con la temática "UNLIX2.0 Versión Estudiantil en el Foro - Conferencia " El uso del Software Libre en la Carrera de Informática Educativa y Casa Abierta de los los Proyectos Investigativos Modulares desarrollados por los alumnos de la carrera" Con una duración de 20 horas.

Loja, enero 2011

Matias Ramirez Bravo, Mgs. SECRETARIO ABOGADO DEL AREA EDUCATIVA  
 Ing. Galin Guevarra Espinoza COORDINADOR DE LA CARRERA  
 Diego Sasavedra PRESIDENTE DE LA ASOCIACION



**UNIVERSIDAD NACIONAL DE LOJA**  
**CARRERA INFORMÁTICA EDUCATIVA**  
**ASOCIACIÓN CARRERA INFORMÁTICA EDUCATIVA**

Confiere el presente:

**CERTIFICADO**

**R: Sr. Victor Cueva**

Por haber participado con la temática "UNLIX2.0 Versión Estudiantil en el Foro - Conferencia " El uso del Software Libre en la Carrera de Informática Educativa y Casa Abierta de los los Proyectos Investigativos Modulares desarrollados por los alumnos de la carrera" Con una duración de 20 horas.

Loja, enero 2011

Matias Ramirez Bravo, Mgs. SECRETARIO ABOGADO DEL AREA EDUCATIVA  
 Ing. Galin Guevarra Espinoza COORDINADOR DE LA CARRERA  
 Diego Sasavedra PRESIDENTE DE LA ASOCIACION



## ANEXO I. ANTEPROYECTO DE TESIS



# UNIVERSIDAD NACIONAL DE LOJA

ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS  
RECURSOS NATURALES NO RENOVABLES

## CARRERA DE INGENIERÍA EN SISTEMAS

CREACIÓN DEL REPOSITORIO DE PAQUETES Y  
ACTUALIZACIÓN DEL SISTEMA OPERATIVO UNIX

TESIS PREVIA A LA OBTENCIÓN  
DEL TÍTULO DE INGENIERO EN  
SISTEMAS.

**AUTORES:** César Augusto Silverio Mendieta.  
Víctor Ignacio Cueva Cango.

**DIRECTOR:** Ing. Edison Leonardo Coronel Romero

**LOJA-ECUADOR**

**2010**



## **1. TITULO**

### **“CREACIÓN DEL REPOSITORIO DE PAQUETES Y ACTUALIZACIÓN DEL SISTEMA OPERATIVO UNIX”**



## 2. PROBLEMÁTICA

### 2.1 Situación Problemática

Los sistemas operativos hoy en día son herramientas de software que permiten que las computadoras puedan funcionar de una manera perfecta y suplir las necesidades de empresas y usuarios que ven en la tecnología una solución para sus problemas.

En la era actual la tecnología es la herramienta que nos permite resolver problemas que en épocas pasadas eran complicadas y difíciles de solucionar, ahora en la era de las computadoras, los teléfonos celulares, las telecomunicaciones y la era digital son resueltos fácilmente, porque hay un acoplamiento gradual de la tecnología con el hombre y la sociedad. Gracias a este acoplamiento la sociedad está marcando etapas de desarrollo de acuerdo a como se van descubriendo nuevos métodos de solución eficientes y rápidos que permitan que la sociedad satisfaga sus necesidades de cambio y progreso. Por esta razón existe una total avalancha de nuevos conocimiento y verdades surgidas a partir de experimentación y verificaciones que contemplan a los datos y a la información como engrane principal que mueve a toda esta rueda de cambios tecnológicos, como los datos y la información no solo se presentan así como un conjunto de frases con sentido y un objetivo trazado sino también se presentan como un elaborado proyecto de control y manejo de maquinaria mediante órdenes escritas como objetivo general o también como un conjunto de palabras o instrucciones que tienen el mismo fin de controlar instrumentos, aquí es donde se evidencia la necesidad y la importancia de que este conjunto de instrucciones estén acorde con los avances tecnológicos de la época es decir que estén perfectamente actualizados y de esta forma no queden obsoletos los instrumentos que poseen estas instrucciones, dicho de otra manera para que las computadoras no queden desactualizadas en su software o sistema operativo deben ser actualizables y de esta manera tener el sistema siempre a la vanguardia de la tecnología .

Por eso es que la Universidad Nacional de Loja ha optado por tener su propia distribución de un sistema operativo denominación UNLIX cuya creación fue hecha por dos egresados de la carrera de ingeniería de sistemas Roberto Mora y Maritza



Quezada, fue concebido el sistema como un trabajo de tesis previo a la obtención del título de ingeniería, esta tesis al culminarse dio como resultado la distribución del sistema operativo UNLIX denominado así por las siglas de UNL que significa Universidad Nacional de Loja con la terminación IX que significa la unión con la palabra LINUX para formar el acrónimo UNLIX y obtener así el nombre del sistema para que la universidad tuviera su propia distribución libre de un sistema operativo el mismo que con fines educativos presta servicio a los docente y estudiantes de la universidad teniendo como ventajas el uso sencillo y fácil, el costo de licencia que es prácticamente cero y la confiabilidad de un buen sistema libre como base que es debían, por esta razón es necesario tener una nueva versión del sistema construyendo un repositorio general de paquetes que permita la actualización del mismo para tener un ámbito más amplio de uso y sobretodo más eficiente, ya que los paquetes del sistema se desactualizan conforme avanza la versión de la distribución base , aquí es donde el repositorio de paquetes cumple un papel fundamental para que esta actividad de actualización sea correcta.

En consecuencia la universidad es la que se beneficiaría más con la creación del repositorio de distribución de UNLIX, porque es donde se va a realizar la construcción del mismo y le quedaría como única opción de uso para el sistema UNLIX que ayuda a estudiantes, maestros y público en general. Como es de conocimiento la desactualización en todo ámbito deja en obsolencia todas las cosas por lo que es necesario actualizarlas.

Como es necesario también que los usuarios tengan opciones de actualización del sistema porque los anchos de banda de conexión a internet son limitados para la descarga de paquetes, se piensan en alternativas que lleven a la actualización del sistema de otra forma como por ejemplo, usando un DVD, donde se albergue un repositorio primario de actualización del sistema operativo UNLIX y la tercera alternativa de actualización, es usando la tecnología cliente servidor, ya que el repositorio de paquetes va estar dentro de un servidor remoto de la Universidad.



## Problema General de investigación

La constante desactualización del sistema operativo UNLIX en todos sus paquetes de software principales y de aplicación presentan el inconveniente de que el sistema no sería funcional en toda su magnitud con el pasar del tiempo y por consiguiente se dejaría sin importancia la utilización eficiente y rápida de la distribución.

## 2.2 Delimitación

La presente investigación se limita a construir un repositorio en donde el usuario de forma automática pueda hacer que su sistema operativo UNLIX realice las actualizaciones sin inconvenientes.

## Problemas específicos de investigación

En la época actual existe la necesidad de actualización del software y en específico de los sistemas operativos de distribución libre con lo cual se presentan los siguientes problemas específicos.

- La falta de conocimiento de que paquetes son de fundamental importancia para que estén presentes en el sistema.
- Falta de un repositorio que actualice los paquetes fuente y de aplicación del sistema.
- La carencia latente de la actualización del sistema desde un servidor donde repose el repositorio.
- La no presencia de un medio extraíble como fuente de alojamiento de un repositorio.
- La insuficiencia de compatibilidad de algunos sistemas operativos en diferentes arquitecturas.

Con la solución a estos problemas específicos del sistema la creación de un repositorio es perfectamente viable y requerido.



### 2.2.1 Espacio

Una de las debilidades de UNLIX son las actualizaciones, el objetivo es el desarrollar el repositorio correspondiente para la administración de las actualizaciones de los paquetes dentro del mencionado sistema operativo, a través de un servidor remoto y de un DVD.

### 2.2.2 Tiempo

El tiempo que se dedicará para estudiar, analizar, comprender, desarrollar y probar el objeto de estudio es de aproximadamente siete meses.

### 2.2.3 Unidades de Observación

Para realizar el repositorio y actualización de UNLIX, se tomará en cuenta los siguientes elementos de observación:

- ✓ Estudio completo del Sistema operativo UNLIX.
- ✓ Uso del Servidor del AEIRNN de la Universidad Nacional de Loja.
- ✓ Análisis, Estudio y Manejo de los paquetes y actualizaciones del Sistema Operativo Debían.
- ✓ Diseño de las interfaces Usuario-Sistema para la administración de las actualizaciones del sistema operativo UNLIX con la ayuda y uso de un manejador de paquetes
- ✓ Uso de Lenguaje de programación libres (Phyton, Mono, Java)



### 3 JUSTIFICACION

#### 3.1 Justificación

Las computadoras hoy en día están presentes casi en todas las actividades del ser humano y por esta razón se han vuelto indispensables a la hora de realizar las tareas de la sociedad. Es por esto que casi para todos los usuarios directos e indirectos de las computadoras, es conocido el lenguaje cotidiano de estas máquinas fascinantes, por esta razón los sistemas operativos están presentes hoy por hoy en todas las áreas que una computadora haya incursionado, de esta forma es que hay tantos sistemas operativos como computadores existen en la sociedad, claro que en una menor cantidad.

En esencia el sistema operativo hace funcionar la máquina (hardware), este es indispensable para las computadoras, lo que nos induce a las personas a conocer cómo funcionan, aprender y entender la configuración del mismo para tener dominio y optimizar el funcionamiento de un equipo informático.

Cuando existen necesidades que tienen que ser solventadas de las cuales uno como usuario o ingeniero de sistemas busca soluciones adecuadas a situaciones extraordinarias en cualquier ámbito ya sea de negocios, cultura, industria, educación, etc. Ahí es cuando se necesita un sistema operativo acorde a la necesidad y acorde a la situación que se presenta, es decir, un sistema operativo que satisfaga las expectativas deseadas. En la época de los 80's era casi imposible tener un sistema operativo para suplir ciertas necesidades, porque a estos solo se los construía en las grandes empresas de desarrollo de software, pero en la actualidad con la filosofía libre de GNU/LINUX esto se torna mucho más sencillo y factible de realización, debido principalmente a la flexibilidad de manipulación de código fuente presente en esta filosofía dando, así la posibilidad de crear nuestras propias distribuciones de sistemas operativos y utilización de otros sistemas libres para uso diario a nivel de hogar o empresarial.

Para que un sistema operativo se desempeñe de una buena forma se necesita que los paquetes que este usa estén constantemente actualizados, ya que estos por el vertiginoso avance tecnológico que tienen las computadoras se desactualizan rápidamente dando origen así a un problema que es resuelto



agrupando estos paquetes en un solo lugar, obteniendo de esta manera a los repositorios de distribuciones cuyo uso masificado permite actualizar un sistema operativo con eficiencia.

Por tal motivo el tener un repositorio de actualizaciones para el sistema operativo UNLIX es de primordial importancia, ya que sin este repositorio sería imposible darle un adecuado uso al nuevo sistema operativo denominado UNLIX, es por esto que se tornaría necesario albergar el repositorio de distribución UNLIX en el servidor del área para que sea fácilmente actualizable.

Con la presencia del repositorio el sistema operativo UNLIX, será más flexible y consolidado como sistema global para que pueda ser usado con más aplicaciones obteniendo de esta manera un ámbito más amplio de uso y manejo y sobre todo para que se mantenga actualizado y el sistema este a la vanguardia de la tecnología.

### **3.2 Viabilidad**

Consideramos viable la realización de este proyecto, porque se cuenta con todos los elementos necesarios para su desarrollo, como elementos tendríamos la distribución UNLIX, la cual se la actualizará con todo el proceso que este proyecto conlleva, el otro elemento que se considera para la viabilidad, es el marco legal que provee el software libre, el mismo que permite tener una opción de manejo para modificar o adaptar a la distribución UNLIX y obtener así su actualización, la misma se realizará con la creación del repositorio de paquetes y actualización del sistema operativo UNLIX.



## OBJETIVOS

### 4.1 Objetivo general

- Disponer de un repositorio general de actualización de paquetes para el sistema operativo UNLIX.

### 4.2 Objetivos específicos

1. Realizar un análisis de los paquetes de UNLIX para determinar posibles actualizaciones e inclusiones.
2. Crear un repositorio general de paquetes para el sistema operativo UNLIX utilizando la versión estable de debían Lenny.
3. Montar el repositorio estándar de paquetes para UNLIX en un servidor del Área de Energía de Industrias y Recursos Naturales no Renovables.
4. Almacenar un repositorio primario en DVD para actualizar el sistema sin necesidad de conexión
5. Realización del plan de validación del repositorio de paquetes utilizando diferentes arquitecturas.



## 5. MARCO TEÓRICO

Dentro del mundo de las computadoras conocemos que para todo hardware se requiere un software, dentro de este campo tenemos los sistemas operativos, los mismos que nos permiten interactuar con nuestro hardware. Hoy en día se conoce la el desarrollo de software está creciendo a pasos agigantados, lo que os compromete a que actualicemos nuestros sistemas operativos y software en general. Las empresas encargadas de elaborar sistemas operativos, cuenta con los sus respectivos repositorios de paquetes (para ejercer las actualizaciones).

Se sabe que los sistemas operativos cuentan con un núcleo es una parte fundamental de un sistema operativo, que en esencia es un núcleo y una serie de programas que permiten a un usuario interactuar con el sistema. En nuestro caso, es lo que se denomina GNU/Linux: núcleo Linux con las aplicaciones base del sistema del proyecto GNU.

Ahora bien, por lo general, cuando se habla de Linux mucha gente se refiere indistintamente a uno u otro concepto.

Las distribuciones están formadas por paquetes. “**Un paquete** viene a ser una pieza de software que cumple una determinada funcionalidad. Así pues, un paquete puede ser un programa, una biblioteca, un conjunto de plugins... Pueden ser incluso, por ejemplo, iconos.”<sup>40</sup>.

Básicamente, a que una distribución puede verse como un gran LEGO. Cada paquete es una pieza del LEGO. A la hora de construir el LEGO, para colocar algunas piezas, debes haber puesto otras antes (las llamadas *dependencias*). Algunas piezas son claves, y forman la base para que se puedan colocar todas las demás (como por ejemplo, el kernel). Otras, en cambio, sólo son necesarias para piezas muy concretas. Y muchas piezas no tienen nada que ver unas con otras.

---

<sup>40</sup> <http://es.wikipedia.org/wiki/>



### ¿Qué es un repositorio?

Y, al fin, llegamos al concepto de repositorio: “El repositorio es un almacén de paquetes (como la caja con las piezas). Los repositorios pueden estar en internet, en un medio extraíble como un CD, en el disco duro.” 41.

Hay tanto repositorios oficiales (con todos los paquetes de la distribución, así como sus actualizaciones de seguridad y bugs), como repositorios con software que Mandriva no puede incluir en los oficiales (el llamado PLF, que contiene códecs multimedia propietarios, programas libres pero con restricciones legales y cosas así), como repositorios mantenidos por voluntarios con las últimas versiones del software (con los que hay que tener mucho cuidado. Personalmente, no te recomendaría usarlos).

Históricamente, los repositorios oficiales públicos de Mandriva en Internet contenían únicamente software libre. A partir de la versión 2007.1 fue añadido otro repositorio oficial público más (llamado *non-free*) que contiene drivers privativos para tarjetas gráficas y otros dispositivos. Los socios de Mandriva disponen, además, de repositorios adicionales con software privativo ya empaquetado, por ejemplo. El PLF es de acceso público para todo el mundo y tiene una rama con software libre únicamente, y otra con software privativo únicamente.

Y, finalmente, nos encontramos con el sistema de gestión de paquetes. En el caso de Mandriva, éste es **urpmi**. Es el encargado de gestionar la instalación, actualización y desinstalación de los paquetes de la distribución.

Los repositorios incluyen un fichero con las dependencias de todos los paquetes que contienen (los llamados `hdlist.cz`, o `synthesis.hdlist.cz`), es decir, las instrucciones del LEGO. La diferencia está en que el primero contiene, además de las dependencias, información sobre cada paquete (para qué sirve, qué ficheros contiene, etc), mientras que el segundo únicamente tiene las dependencias. El primero, por tanto, es mucho más grande que el segundo, pero contiene información muy útil)

---

41 <http://www.wikipedia.org/wikik>



El gestor de paquetes actúa como el constructor del LEGO y utiliza dichas instrucciones para saber qué piezas debe poner antes de una concreta (es decir, resolver las dependencias de los paquetes).

## ¿Qué es Linux?

Más que un sistema operativo es una filosofía, pero se lo puede definir como “un Sistema Operativo. Es una implementación de libre distribución UNIX para computadoras personales (PC), servidores, y estaciones de trabajo. Fue desarrollado para el i386 y ahora soporta los procesadores i486, Pentium, Pentium Pro y Pentium II, así como los clones AMD y Cyrix. También soporta máquinas basadas en SPARC, DEC Alpha, PowerPC/PowerMac, y Mac/Amiga Motorola 680x0.

Como sistema operativo, Linux es muy eficiente y tiene un excelente diseño. Es multitarea, multiusuario, multiplataforma y multiprocesador; en las plataformas Intel corre en modo protegido; protege la memoria para que un programa no pueda hacer caer al resto del sistema; carga sólo las partes de un programa que se usan; comparte la memoria entre programas aumentando la velocidad y disminuyendo el uso de memoria; usa un sistema de memoria virtual por páginas; utiliza toda la memoria libre para cache; permite usar bibliotecas enlazadas tanto estática como dinámicamente; se distribuye con código fuente; usa hasta 64 consolas virtuales; tiene un sistema de archivos avanzado pero puede usar los de los otros sistemas; y soporta redes tanto en TCP/IP como en otros protocolos.”<sup>42</sup>.

La Historia de LINUX hace su aparición a principios de la década de los noventa, era el año 1991 y por aquel entonces un estudiante de informática de la Universidad de Helsinki, llamado Linus Torvalds empezó, como una afición y sin poderse imaginar a lo que llegaría este proyecto, a programar las primeras líneas de código de este sistema operativo llamado LINUX.

---

<sup>42</sup> <http://es.wikipedia.org/wiki/GNU/Linux>



Este comienzo estuvo inspirado en MINIX, un pequeño sistema Unix desarrollado por Andy Tanenbaum. Las primeras discusiones sobre Linux fueron en el grupo de noticias como minix, en estas discusiones se hablaba sobre todo del desarrollo de un pequeño sistema Unix para usuarios de Minix que querían más.

Linus nunca anuncio la versión 0.01 de Linux (agosto 1991), esta versión no era ni siquiera ejecutable, solamente incluía los principios del núcleo del sistema, estaba escrita en lenguaje ensamblador y asumía que uno tenía acceso a un sistema Minix para su compilación.

El 5 de octubre de 1991, Linus anuncio la primera versión "Oficial" de Linux, - versión 0.02. Con esta versión Linus pudo ejecutar Bash (GNU Bourne Again Shell) y gcc (El compilador GNU de C) pero no mucho más funcionaba. En este estado de desarrollo ni se pensaba en los términos soporte, documentación, distribución. Después de la versión 0.03, Linus salto en la numeración hasta la 0.10, más y más programadores a lo largo y ancho de internet empezaron a trabajar en el proyecto y después de sucesivas revisiones, Linus incremento el número de versión hasta la 0.95 (Marzo 1992). Más de un año después (diciembre 1993) el núcleo del sistema estaba en la versión 0.99 y la versión 1.0 no llego hasta el 14 de marzo de 1994. Desde entonces no se ha parado de desarrollar, la versión actual del núcleo es la 2.2 y sigue avanzando día a día con la meta de perfeccionar y mejorar el sistema.

Linus Benedict Torvalds nació en Helsinki, Finlandia, el año 1969. Empezó a "trabajar" con ordenadores a los 10 años, cuando su abuelo le compró un Comodore el año 1980. Éste buen señor era un matemático y estadista. Trabajaba a la Universidad y fue quién "enganchó" al mundo de los computadores a nuestro buen amigo Linus.

Con el paso del tiempo, Linus pasó a tener un Sinclair QL, un gran ordenador de Clive Sinclair (creador del conocido Spectrum), que tenía algún pequeño error de diseño. Linus se sintió especialmente atraído por esta máquina, después de crear aplicaciones para ésta computadora y de haber retocado su hardware con la finalidad de adaptarlo a sus necesidades. El problema que tenía dicha máquina



era que los recursos eran insuficientes para poder llevar a la práctica los planes de Linus. Además, no era un equipo compatible. Así pues, el mes de enero de 1991 compró su primer PC, un 386.

Retrocedamos otra vez en el tiempo, hasta 1988. Éste año fue cuando Linus entró a la Universidad. Este mismo año fue cuando un sistema operativo didáctico, basado en Unix y creado por Andy Tannenbaum, empezó a cobrar importancia. Dicho sistema operativo era el famoso Minix.

Linus entró a formar parte de la comunidad de usuarios de Minix. Tannenbaum cometió un error en su sistema operativo. Era demasiado limitado, tanto técnicamente como políticamente, es decir, en ningún momento tuvo en cuenta la posibilidad de incluir Minix al proyecto GNU (creado el año 1983 por Richard Salmen). En realidad, la creación de Andy Tannenbaum estaba pensada para ser distribuida comercialmente. Su principal error fue ceder todos los derechos a Prentice Hall, que empezó a cobrar 150 dólares por licencia (unas 22.500 ptas. / 135.23 euros).

Así pues, Linus tomó la decisión de cambiar esta política debido a que el sistema Minix era ideal para los estudiantes de sistemas operativos, y su precio era considerablemente alto. Llegamos de nuevo al año 1991, cuando Linus se acabó de comprar su primer 386. En aquellos momentos, la intención de nuestro amigo era clara: crear un nuevo Kernel de UNIX basado en el Kernel de Minix y modificarlo periódicamente de manera que fuera capaz de ejecutar aplicaciones GNU.

A pesar de todo, no toda la vida del Sr. Linus ha estado dedicada exclusivamente al sistema operativo Linux.

Actualmente Linus Torvalds vive desde el año 1997 en Santa Clara, California, junto con su esposa Tove y sus dos hijas, Patricia Miranda y Daniela. Le encanta jugar con sus dos pequeñas. Está trabajando para Transmeta, aunque nadie no sabe demasiado bien en qué trabaja. Parece ser que existe una cláusula en su contrato que le impide revelar dicha información. De todas formas, se sabe que Transmeta está desarrollando un nuevo procesador con un juego de instrucciones



RISC capaz de ejecutar código x86.

Tal vez una de las aficiones más conocidas de Linus es que le gusta la cerveza, más aún si es Guinness. Así lo demostraron los miembros del SVLUG (Silicon Valley Linux User Group) cuando Linus fué a visitarlos para llevar a cabo una conferencia y le regalaron una caja de cerveza.

Poca cosa más de sabe de este personaje que ha cambiado la forma de ver los ordenadores durante estos últimos años y que ha conseguido dar un buen empujón al proyecto GNU.

Entre las características “fuente *Infosheet-Como. Autor: Ivan Casado*” :

- **Multitarea:** La palabra multitarea describe la habilidad de ejecutar varios programas al mismo tiempo. LINUX utiliza la llamada multitarea preventiva, la cual asegura que todos los programas que se están utilizando en un momento dado serán ejecutados, siendo el sistema operativo el encargado de ceder tiempo de microprocesador a cada programa.
- **Multiusuario:** Muchos usuarios usando la misma maquina al mismo tiempo.
- **Multiplataforma:** Las plataformas en las que en un principio se puede utilizar Linux son 386-, 486-. Pentium, Pentium Pro, Pentium II, Amiga y Atari, también existen versiones para su utilización en otras plataformas, como Alpha, ARM, MIPS, PowerPC y SPARC.
- **Multiprocesador:** Soporte para sistemas con mas de un procesador esta disponible para Intel y SPARC.
- Funciona en modo protegido 386.
- Protección de la memoria entre procesos, de manera que uno de ellos no pueda colgar el sistema.
- Carga de ejecutables por demanda: Linux sólo lee del disco aquellas partes de un programa que están siendo usadas actualmente.
- Política de copia en escritura para la compartición de páginas entre ejecutables: esto significa que varios procesos pueden usar la misma zona de memoria para ejecutarse. Cuando alguno intenta escribir en esa memoria, la página (4Kb de memoria) se copia a otro lugar. Esta política de copia en



escritura tiene dos beneficios: aumenta la velocidad y reduce el uso de memoria.

- Memoria virtual usando paginación (sin intercambio de procesos completos) a disco: A una partición o un archivo en el sistema de archivos, o ambos, con la posibilidad de añadir más áreas de intercambio sobre la marcha. Un total de 16 zonas de intercambio de 128Mb de tamaño máximo pueden ser usadas en un momento dado con un límite teórico de 2Gb para intercambio. Este límite se puede aumentar fácilmente con el cambio de unas cuantas líneas en el código fuente.
- La memoria se gestiona como un recurso unificado para los programas de usuario y para el caché de disco, de tal forma que toda la memoria libre puede ser usada para caché y ésta puede a su vez ser reducida cuando se ejecuten grandes programas.
- Librerías compartidas de carga dinámica (DLL's) y librerías estáticas.
- Se realizan volcados de estado (core dumps) para posibilitar los análisis post-mortem, permitiendo el uso de depuradores sobre los programas no sólo en ejecución sino también tras abortar éstos por cualquier motivo.
- Compatible con POSIX, System V y BSD a nivel fuente.
- Emulación de iBCS2, casi completamente compatible con SCO, SVR3 y SVR4 a nivel binario.
- Todo el código fuente está disponible, incluyendo el núcleo completo y todos los drivers, las herramientas de desarrollo y todos los programas de usuario; además todo ello se puede distribuir libremente. Hay algunos programas comerciales que están siendo ofrecidos para Linux actualmente sin código fuente, pero todo lo que ha sido gratuito sigue siendo gratuito.
- Control de tareas POSIX.
- Pseudo-terminales (pty's).
- Emulación de 387 en el núcleo, de tal forma que los programas no tengan que hacer su propia emulación matemática. Cualquier máquina que ejecute Linux parecerá dotada de coprocesador matemático. Por supuesto, si el ordenador ya tiene una FPU (unidad de coma flotante), esta será usada en lugar de la emulación, pudiendo incluso compilar tu propio kernel sin la emulación matemática y conseguir un pequeño ahorro de memoria.



- Soporte para muchos teclados nacionales o adaptados y es bastante fácil añadir nuevos dinámicamente.
- Consolas virtuales múltiples: varias sesiones de login a través de la consola entre las que se puede cambiar con las combinaciones adecuadas de teclas (totalmente independiente del hardware de video). Se crean dinámicamente y puedes tener hasta 64.
- Soporte para varios sistemas de archivo comunes, incluyendo minix-1, Xenix y todos los sistemas de archivo típicos de System V, y tiene un avanzado sistema de archivos propio con una capacidad de hasta 4 Tb y nombres de archivos de hasta 255 caracteres de longitud.
- Acceso transparente a particiones MS-DOS (o a particiones OS/2 FAT) mediante un sistema de archivos especial: no es necesario ningún comando especial para usar la partición MS-DOS, esta parece un sistema de archivos normal de Unix (excepto por algunas restricciones en los nombres de archivo, permisos, y esas cosas). Las particiones comprimidas de MS-DOS 6 no son accesibles en este momento, y no se espera que lo sean en el futuro. El soporte para VFAT (WNT, Windows 95) ha sido añadido al núcleo de desarrollo y estará en la próxima versión estable.
- Un sistema de archivos especial llamado UMSDOS que permite que Linux sea instalado en un sistema de archivos DOS.
- Soporte en sólo lectura de HPFS-2 del OS/2 2.1
- Sistema de archivos de CD-ROM que lee todos los formatos estándar de CD-ROM.
- TCP/IP, incluyendo ftp, telnet, NFS, etc.
- Appletalk.
- Software cliente y servidor Netware.
- Lan Manager / Windows Native (SMB), software cliente y servidor.
- Diversos protocolos de red incluidos en el kernel: TCP, IPv4, IPv6, AX.25, X.25, IPX, DDP, Netrom, etc.



## ¿Qué es GNU/Linux?

“Es uno de los términos empleados para referirse al sistema operativo libre similar a Unix que usualmente utiliza herramientas de sistema GNU. Su desarrollo es uno de los ejemplos más prominentes de software libre; todo el código fuente puede ser utilizado, modificado y redistribuido libremente por cualquiera bajo los términos de la GPL (Licencia Pública General de GNU) y otras licencias libres.

A pesar de que Linux es el sistema operativo, cabe notar que debido a que usualmente se maneja con las herramientas GNU, una parte significativa de la comunidad, así como muchos medios generales y especializados, prefieren utilizar el término *Linux* para referirse a la unión de ambos proyectos. Para más información consulte la sección "Denominación GNU/Linux" o el artículo "Controversia por la denominación GNU/Linux". Una vez acabado el proyecto GNU, reclamará a su núcleo como motor principal, el conocido como Hurd.

A las variantes de esta unión de programas y tecnologías, se las denomina distribuciones. Su objetivo consiste en ofrecer ediciones que cumplan con las necesidades de determinado grupo de usuarios. Algunas de ellas son especialmente conocidas por su uso en servidores y supercomputadoras. No obstante, es posible instalarlo en una amplia variedad de hardware como computadoras de escritorio y portátiles.

En computadoras de bolsillo, teléfonos móviles, dispositivos empotrados, videoconsolas y otros, puede darse el caso en que las partes de GNU se reemplacen por alternativas más adecuadas. Para saber más sobre las arquitecturas soportadas<sup>43</sup>

---

43 <http://es.wikipedia.org/wiki/GNU/Linux>



## Linux frente a los otros sistemas operativos

Linux es una muy buena alternativa frente a los demás sistemas operativos. Más allá de las ventajas evidentes de costo, ofrece algunas características muy notables.

En comparación con las otras versiones de Unix para PC, la velocidad y confiabilidad de Linux son muy superiores. También está en ventaja sobre la disponibilidad de aplicaciones, ya que no hay mucha difusión de estos otros Unixes (como Solaris, XENIX o SCO) entre los usuarios de PC por sus altos costos.

Comparado con sistemas operativos como los diferentes Microsoft Windows, Linux también sale ganando. Los bajos requisitos de hardware permiten hacer un sistema potente y útil de aquel 486 que algunos guardan en un armario. Esta misma característica permite aprovechar al máximo las capacidades de las computadoras más modernas. Es poco práctico tener una PC con 16 Mb de RAM y ponerle un sistema operativo que ocupa 13 (que es lo que reporta sobre Windows 95 el System Information de Symantec). No solo es superior respecto a el sistema de multitarea y de administración de memoria, sino también en la capacidades de networking (conectividad a redes) y de multiusuario (aún comparando con sistemas multiusuario como NT). La única desventaja de Linux frente a estos sistemas, es la menor disponibilidad de software, pero este problema disminuye con cada nuevo programa que se escribe para el proyecto GNU, y con algunas empresas que están desarrollando software comercial para Linux (por ej., Corel).

Cuando hablamos de GNU/LINUX, nos referimos a la filosofía libre, se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

## HOWTO Repositorio Debían

“Un repositorio Debían es un conjunto de paquetes Debían organizados en un árbol de directorio especial, el cual también contiene unos cuantos archivos adicionales los que a su vez contienen índices y checksums (sumas de veri\_caci\_on) de los paquetes. Si un usuario agrega un repositorio a su archivo/etc/apt/sources.list,



puede fácilmente ver e instalar todos los paquetes disponibles dentro de el, de la misma forma que a los paquetes contenidos en Debían.

Un repositorio puede estar tanto online como offline (por ejemplo en un CDROM), aunque es mas común el primer caso.

Este documento explica cómo trabajan los repositorios Debían, como crearlos, y como agregarlos al sources.list correctamente<sup>44</sup>

### **Cómo funcionan los repositorios?**

Un repositorio consiste en al menos un directorio con algunos paquetes DEB en el, y dos archivos especiales: Packages.gz para los paquetes binarios, y Sources.gz para los paquetes fuente.

Si su repositorio esta listado correctamente en sources.list (mas sobre esto adelante), apt-get obtendrá el índice de Packages.gz si los paquetes binario están listados (con la palabra clave deb) y Sources.gz si los fuentes están listados (con la palabra clave deb-src).

Packages.gz contiene el nombre, versión, tamaño, descripción corta y larga, y las dependencias de cada paquete, mas algo de información adicional la cual no es de interés para nosotros. Toda esta información es listada (y usada por) los administradores de paquetes de Debían tales como dselect o aptitude.

Sources.gz contiene el nombre, versión y dependencias de construcción (los paquetes necesitados para construir) de cada paquete (mas algo de información la cual tampoco es de interés para nosotros); esta información es usada por apt-get source y herramientas similares.

Hay un archivo opcional Release que contiene alguna información sobre su repositorio; esta es usada para hacer Pinning, un truco interesante que no explicare en este documento.

---

<sup>44</sup> <http://myhomemdz.com.ar/doc/howto-repositorios/howto-repositorios.pdf>



Así, una vez que haya creado su repositorio, puede listar e instalar todos sus paquetes junto a los de Debían; si actualiza un paquete, será actualizado cuando el usuario ejecute `apt-get upgrade`; y cada usuario podrá ver fácilmente una descripción corta y otra información importante sobre sus paquetes.

Pero hay más. Si son creados apropiadamente, los repositorios pueden ofrecer distintos paquetes por cada distribución soportada y para cada una de las (actualmente once) arquitecturas soportadas; `apt` automáticamente buscará la correcta de acuerdo a la máquina del usuario, sin que \_el tenga que conocer sobre todas las diferentes arquitecturas. También le permite agrupar sus paquetes en componentes, justo como los paquetes Debían son divididos en `main`, `non-free` y `contrib`. Así, sobre todo si su software es multiplataforma, amara los repositorios de paquetes.

### Como preparar un repositorio?

“Hay dos tipos de repositorios: unos muy complejos donde el usuario solo tiene que especificar la ruta base al repositorio, la distribución y los componentes que quiere (`apt` buscará automáticamente para la arquitectura correcta, si está disponible), y unos más simples donde el usuario tiene que especificar una ruta exacta (y `apt` no hará magia para descubrir cuales paquetes son los correctos). Los primeros son un poco más complejos de configurar, pero más fáciles de usar, y siempre deben ser usado para repositorios complejos y/o multiplataforma; los últimos son más simples de configurar, pero deben ser usados solo para repositorios pequeños o de única arquitectura.”<sup>45</sup>

“Aunque no es realmente correcto, llamare a los primeros Repositorios Automáticos y a los últimos Repositorios Triviales.

Los primeros (Repositorios Automáticos) nos dicen que los paquetes libres van en `main`; los no libres en `non-free`, y los libres que dependen de otros no libres en `contrib`. Debían actualmente soporta 11 arquitecturas; He omitido la mayoría de

---

<sup>45</sup> <http://myhomemdz.com.ar/doc/howto-repositorios/howto-repositorios.pdf>



ellas en pos de la brevedad. Cada directorio binary-\* contiene un Packages.gz y un archivo opcional Release; cada directorio source contiene un Source.gz y un archivo opcional Release.

Observe que los paquetes no tienen que estar en el mismo directorio que los archivos índices, porque los archivos índices contienen rutas a los paquetes individuales; de hecho, estos podrían estar en cualquier parte dentro del repositorio. Esto hace posible crear pools.

Usted es libre de crear cualquier cantidad de distribuciones y componentes, y llamarlos como desee; los que use en el ejemplo son solo los usados en Debían.

Podría, por ejemplo, crear las distribuciones current y beta (en vez de stable, testing y unstable), y los componentes foo, bar, baz y qux (en vez de main, contrib y no-free).

Si bien puede llamar a los componentes como desee, es generalmente una buena idea usar las distribuciones Debían estándar, porque eso es lo que esperan los usuarios de Debían.

Por otro lado los repositorios triviales consisten en un directorio raíz y de tantos subdirectorios como desee. Como los usuarios tienen que especificar la ruta hacia la raíz del repositorio y la ruta relativa entre la raíz y el directorio con el archivo de índice en él, usted es libre de hacer lo que quiera (incluso poner todo dentro de la raíz del repositorio; entonces, la ruta relativa será simplemente /)".<sup>46</sup> Por ejemplo:  
Repositorio Trivial con 2 subdirectorios

(La raíz de su repositorio)

```
|  
|-binary  
+-source
```

Ahora lo que se va a ver más adelante es sobre los archivos índice, release los que nos permitirán tener una visión más profunda del funcionamiento general de un repositorio en su totalidad.

---

<sup>46</sup> <http://myhomemd.com.ar/doc/howto-repositorios/howto-repositorios.pdf>



Los **archivos índice** “dpkg-scanpackage genera el archivo Packages y dpkg-scansources el archivo Sources.

Ambos envían su salida a stdout; por lo tanto, para generar los archivos comprimidos, puede usar un comando encadenado como el que sigue: dpkg-scanpackages argumentos | gzip -9c > Packages.gz.

Las dos herramientas trabajan de la misma forma; Ambas toman dos argumentos (en realidad hay más, pero no voy a entrar en detalle de eso aquí; puede leer las páginas man si quiere conocer más): el primero es el directorio bajo el cual los paquetes están, y el segundo es el archivo de override. Nosotros no necesitamos archivos override para repositorios simples, pero como es un argumento requerido, pasaremos simplemente /dev/null.

dpkg-scanpackages escanea los paquetes .deb; dpkg-scansources escanea los archivos .dsc. Es entonces necesario poner los archivos .orig.gz, .diff.gz y .dsc juntos. Los archivos .changes no son necesarios.

Podría también usar el argumento pathprefix de las dos herramientas para simplificar la sintaxis un poco; Deje esto como un ejercicio para el lector (está documentado en la paginas man).

Por su parte **en los relase**, si quiere habilitar a los usuarios de su repositorio a usar Pinning con \_el, debe incluir un archivo Release en todo directorio que contenga un archivo /indice (puede leer más sobre pinning en el APT HOWTO (<http://www.debian.org/doc/manuals/apt-howto/>)).

Los archivos Release son archivos de texto corto y simple de la siguiente forma:

Archive: archivo

Component: componente

Origin: tu Compañía

Label: repositorio Debian de tuCompañía

Architecture: arquitectura



**Archive** El nombre de la distribución de Debían a la que pertenecen (o son diseñados para) los paquetes en este directorio, p.e. stable, testing or unstable.

**Component** El componente de los paquetes en el directorio, por ejemplo main, non-free, o contrib.

**Origin** El nombre de quien hizo los paquetes.

**Label** Alguna etiqueta adecuada para los paquetes o para su repositorio. Use su imaginación.

**Architecture** La arquitectura de los paquetes en este directorio, tales como i386, sparc o source.

Es importante que Archive y Architecture sean correctos, ya que se utilizan principalmente para hacer pinning. Los otros son menos importantes.

**Los pools**, con los repositorios automáticos, distribuir los paquetes en los diferentes directorios rápidamente conduciría a una bestia inmanejable. Esto también es un desperdicio de espacio y ancho de banda, ya que hay muchos paquetes (por ejemplo paquetes de documentación) que son los mismos para todas las arquitecturas.

En estos casos, una posible solución es un pool. Un pool es un directorio adicional bajo la raíz del repositorio que contiene todos los paquetes (los binarios para todas las arquitecturas, distribuciones, y componentes, y todas las fuentes). A través de una inteligente combinación de archivos override (los cuales no son cubiertos en este documento) y de scripts muchos problemas pueden ser evitados.

Un buen ejemplo de un repositorio 'pooleado' es el mismo repositorio de Debían. Los pools son solo útiles para grandes repositorios; Nunca he hecho alguno y no creo necesitarlo en un futuro próximo, y esa es la razón por la que no explico como hacer uno aquí. Si piensa que esa sección debe ser adherida, siéntase libre para escribir una, y contactarme.

Hay varias herramientas para automatizar y facilitar la creación de archivos Debían; He listado las más notables de ellas aquí.



apt-ftpparchive es usada para mover una colección de archivos de paquetes Debían dentro de una jerarquía de archivos apropiada como es usada en los archivos Debian oficiales. Es parte del paquete apt-utils.

apt-move es usado para mover una colección de archivos de paquetes Debían dentro de una jerarquía de archivos apropiada como es usada en los archivos Debian oficiales.

El uso de un repositorio es muy simple, pero depende de qué tipo de repositorio haya hecho: binario o fuente, y automático o trivial. Cada repositorio recibe una línea en sources.list; para uno binario, use el comando deb, y para uno fuente el comando deb-src. Cada línea tiene la siguiente sintaxis: deb|deb-src uri distribución [componente1 ] [componente2 ] [...]

La url es la URI de la raíz del repositorio, como ftp://ftp.yoursite.com/debian, http://yoursite.com/debian, o, para archivos locales, file:///home/ joe/my-debian-repository. La barra invertida es opcional.

Para repositorios automáticos, tiene que especificar una distribución y uno o más componentes; la distribución no debe terminar en una barra invertida.<sup>47</sup>

Algo importante que se debe señalar sobre el **gestor de paquetes synaptic en Debian** está organizado en paquetes. Synaptic es un frontend gráfico del sistema gestor de paquetes apt-get. Desde Synaptic se permite la ejecución de las acciones en línea de orden desde un entorno gráfico, más sencillo y amigable.

Synaptic proporciona las siguientes características:

- Instalar, eliminar, configurar, actualizar y descargar paquetes.
- Actualizar el sistema completo.
- Gestión del repositorio de paquetes.
- Búsqueda de la lista de paquetes por nombre, descripción y otras propiedades de los paquetes.
- Filtrado, de la lista de paquetes conocidos, por estatus, sección o propiedades.

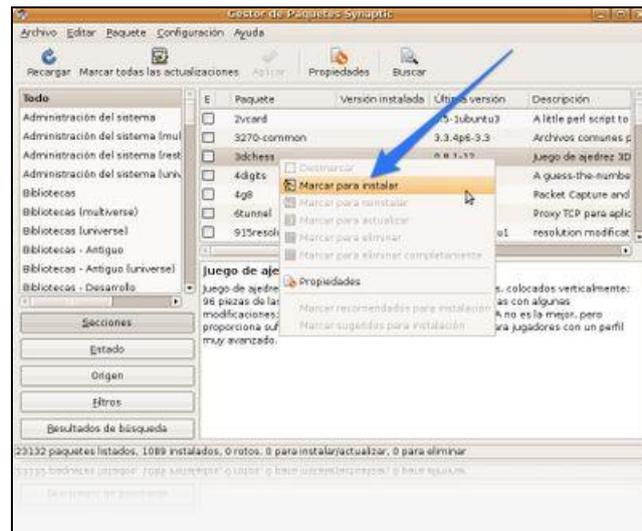
---

<sup>47</sup> <http://myhomemd.com.ar/doc/howto-repositorios/howto-repositorios.pdf>



- Ordenar listados de paquetes por letra inicial, estatus...
- Visualizar toda la documentación en línea referida al paquete.

Es necesario ser root para aplicar cambios sobre el sistema de paquetes. Y podemos acceder en forma gráfica desde el menú de debían.



Interfaz del Gestor de paquetes Synaptic

“**Synaptic** es un avanzado sistema para instalar o eliminar aplicaciones de tu sistema. El entorno es gráfico, como en El centro de software de Ubuntu, pero mucho más potente. Con Synaptic tienes el control completo de los paquetes (aplicaciones) instalados en tu sistema.”<sup>48</sup>

## Remasterizar

Para explicar la remasterización vamos a ver que nos dice la página de Ubuntu, “el proceso de remasterizar el CD de instalación de Ubuntu no es especialmente complejo, pero es un poco tedioso y meticuloso.

El CD de instalación de Ubuntu (desde Ubuntu 6.06, se llama 'alternative install' o 'server' CD) tiene tres partes importantes: un bootloader o cargador de arranque (ISOLINUX en sistemas amd64/i386, yaboot para powerpc) y su configuración, el

<sup>48</sup> [http://www.guia-ubuntu.org/index.php?title=A%C3%B1adir\\_aplicaciones](http://www.guia-ubuntu.org/index.php?title=A%C3%B1adir_aplicaciones)



cual arranca todo; debian-installer, instalador de debían (también conocido como d-i), que es un auténtico sistema Ubuntu en miniatura; y una estructura de repositorio al estilo de Debian, que toma casi todo el espacio en el disco dentro de los directorios "pool" y "dists". Construir un nuevo CD implica modificar sus partes.”<sup>49</sup>

---

<sup>49</sup> [http://files.nireblog.com/blogs/meth/files/remasterizar\\_ubuntu\\_.pdf](http://files.nireblog.com/blogs/meth/files/remasterizar_ubuntu_.pdf)



## 6. METODOLOGÍA

### 6.1 Matriz de Consistencia general

<b>PROBLEMA DE INVESTIGACIÓN:</b>			
La constante falta de actualización del sistema operativo UNLIX en todos sus paquetes de software principales y de aplicación presentando el inconveniente de que el sistema no sería funcional en toda su magnitud y por consiguiente se caería en el desuso o subutilización del mismo.			
TEMA	OBJETO DE INVESTIGACIÓN	OBJETIVO DE LA INVESTIGACIÓN	HIPÓTESIS DE INVESTIGACIÓN
“Creación del repositorio y actualización del sistema operativo UNLIX”.	El sistema operativo UNLIX como base primordial de cambio a la hora de actualizarlo que permitirá desarrollar el componente principal de la investigación la creación del repositorio en si como resultado de la investigación	Disponer de un repositorio general de paquetes para el sistema operativo UNLIX con el fin de que este se mantenga siempre actualizado	El repositorio general de paquetes servirá para actualizar el sistema operativo UNLIX

*Matriz de Consistencia General*

### 6.2 Materiales, métodos y técnicas de trabajo

#### 6.2.1 Materiales

Los materiales que se utilizaran en el proyecto de investigación son:

- Material Bibliográfico
- Recargar de tinta negra
- Recargar de tinta color
- Resma de papel
- Copias
- Internet



- Transporte
- Varios

### 6.2.2 Métodos

**Tipo de investigación.-** Este proyecto se lo investigara de una manera experimental basado en la observación el análisis y la investigación de elementos que conlleven a solucionar los problemas que surgen al tratar de cumplir con los objetivos trazados

**Metodología.-** En este proyecto se utilizara el método científico enfoque de sistemas que determina el estado actual y el posible estado futuro del sistema se está analizando en este caso del proyecto de investigación, este método consta de dos métodos inductivo deductivo, el método inductivo que se vale de la inducción completa que acerca una conclusión general a partir del estudio de todos los casos particulares e inducción incompleta que parte de uno o varios casos particulares para concluir algo general todo con la técnica de la observación para ir de lo particular a lo general, el método deductivo va en un sentido contrario complementario, es decir de lo general a lo particular, buscando siempre una manera de englobar hechos particulares de acuerdo a un hecho conocido general permitiendo aplicar conceptos generales ha hecho particulares.

**Estudio preliminar.-** en esta investigación se hará un estudio preliminar que nos permita tener un idea general de todas las variables del proyecto, en este estudio de hace una revisión de todo lo que contempla el proyecto para ver posibles concatenaciones y orden identificando todo lo que sea de uso para el mejor desarrollo de la investigación.

**Análisis.-** en el análisis se hace un filtro de información obteniendo todos los datos que son necesarios para el desarrollo de la investigación y de esta manera tener el resultado esperado.



**Recolección de datos primarios y secundarios.-** En esta parte de metodología que se usara es la de identificar de toda la información importante de la investigación, conocer cuáles son los datos más importantes de desarrollo y datos complementarios que nos permitan tener la idea global y la solución en de estos procesos que formar parte del proyecto.

**Diseño.-** en esta parte el bosquejo que sale del análisis no da la pauta para que el repositorio sea creado en un lenguaje de alto nivel con la inclusión de la programación del repositorio con el apt del mismo, en este diseño se crea el repositorio general de paquetes para que sea utilizado por el sistema como objeto de actualización del mismo.

**Implementación.-** Es el proceso de verificar instalar equipos en este caso el servidor, instalar aplicaciones y crear archivos para el mantenimiento del sistema.

**Pruebas.-** En esta línea de metodología se hace las rigurosas pruebas que nos permitirá garantizar la estabilidad del repositorio como base principal de actualización del sistema operativo UNLIX.

### 6.2.3 Técnicas de Investigación

Dentro de las técnicas a utilizar tenemos:

- **La Entrevista,** nos ayudará a realizar conocer más sobre la estructura del proyecto UNLIX.
- **La Observación,** Nos beneficiará ya que mediante esta técnica se podrá comparar algunas distribuciones de sistemas operativos basados en GNU/LINUX.
- **Internet,** Es un mecanismo indispensable para poder obtener información.









## 8. PRESUPUESTO Y FINANCIAMIENTO

### a) Recursos Humanos

Los recursos humanos para la elaboración de este proyecto son:  
Los autores y asesores, y los que apoyan en segunda instancia a la realización del mismo, ver (*Tabla 1 Presupuesto y Financiamiento.*)

### b) Recursos Técnicos, Tecnológicos y Materiales

Nos permite identificar algunas variables, como por ejemplo:

- Material Bibliográfico.
- Recargar de tinta negra
- Recargar de tinta color
- Resma de papel
- Internet
- Transporte
- Hardware
- Software, S.O y Lenguajes de programación

Para un mejor detalle, Ver (*Tabla 1 Presupuesto y Financiamiento.*)



## 8.1 Presupuesto y Financiamiento

Descripción	Cantidad	# Horas	Costo Unitario	Costo Total
<b>RECURSOS HUMANOS</b>				<b>3360,00</b>
Desarrolladores	2	1680	2,00	3360,00
Director de Tesis	1	80	0,00	0,00
<b>RECURSOS TÉCNICOS</b>				<b>1937,00</b>
<b>Hardware</b>				<b>1304,00</b>
Laptop Core 2 Duo, 3GB RAM, DVD-ROM, 320 GB	1		800,00	800,00
Laptop P. IV, 1 GB RAM, DVD-ROM, 120 GB	1		400,00	400,00
Memory Flash HP 2 GB	2		24,00	24,00
Impresora HP F380 Multifunción	1		80,00	80,00
<b>Software</b>				<b>3,00</b>
Sistema operativo UNLIX 1.0 (costo del medio)	1		1,00	1,00
Sistema operativo Debian Lenny 5.0 (costo de medio)	1		1,00	1,00
Sistema Operativo Ubuntu 9.10 (costo del medio)	1		1,00	1,00
<b>Comunicaciones</b>				<b>180,00</b>
Servicio de Internet		300	0,60	180,00
<b>Capacitación</b>				<b>450,00</b>
Cursos	3	120	150,00	450,00
<b>RECURSOS MATERIALES</b>				<b>250,00</b>
Suministros y materiales de oficina			150,00	150,00
Transporte y movilización			100,00	100,00
<b>5% Imprevistos</b>				<b>277,35</b>
<b>Sub Total</b>				<b>5547,00</b>
<b>TOTAL</b>				<b>5824,35</b>

Tabla 1 Presupuesto y Financiamiento

## 8.2 Financiamiento

El proyecto será financiado en su totalidad por los desarrolladores.



## 9. BIBLIOGRAFÍA

### Libros:

- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. 1999. El Lenguaje Unificado de Modelado. España, Addison Wesley, 432 pp.

### Tesis:

- MORA Roberto y QUEZADA Maritza; Creación del Sistema Operativo UNLIX. Tesis de Grado. Loja – Ecuador, año 2009.

### Sitios Web:

1. Wikipedia. Enciclopedia libre. [<http://es.wikipedia.org/wiki/GNU/Linux>], [consulta realizada el 04 de diciembre de 2009].
2. Información del proceso de remasterizar usando ubuntu. [[http://files.nireblog.com/blogs/meth/files/remasterizar\\_ubuntu\\_.pdf](http://files.nireblog.com/blogs/meth/files/remasterizar_ubuntu_.pdf)], [consulta realizada el 03 de enero de 2010].
3. GNU ORG. Sitio del proyecto GNU Operating System [<http://www.gnu.org/philosophy/free-sw.es.html>]. [consulta realizada el 20 de noviembre de 2009].
4. Portal Oficial de Ubuntu. [[http://www.guia-ubuntu.org/index.php?title=A%C3%B1adir\\_aplicaciones](http://www.guia-ubuntu.org/index.php?title=A%C3%B1adir_aplicaciones)], [consulta realizada el 20 de diciembre de 2009].
5. Isotton Aaron. Información sobre repositorios en debían. [<http://myhomemdz.com.ar/doc/howto-repositorios/howto-repositorios.pdf>], [consulta realizada el 04 de diciembre de 2009].
6. GNU/LINUX MATTRES. Sitio que se obtiene información del proyecto Linux. [<http://www.obtengalinux.org/linux>]. [consulta realizada el 15 de noviembre de 2009].
7. UBUNTU AND CANONICAL. Sitio oficial de Ubuntu. [<http://www.ubuntu.com/getubuntu/download>], [consulta realizada el 10 de noviembre de 2009].



# 10. ANEXOS



## 10 ANEXOS

### 10.1 Matriz de Consistencia Específica

<b>PROBLEMA ESPECÍFICO:</b> La falta de conocimiento de que paquetes son de fundamental importancia para que estén presentes en el sistema			
<b>OBJETIVO ESPECÍFICO</b>	<b>HIPÓTESIS ESPECÍFICA</b>	<b>UNIDAD DE OBSERVACIÓN</b>	<b>SISTEMA CATEGORIAL</b>
Realizar un análisis de los paquetes de UNLIX para determinar posibles actualizaciones e inclusiones.	La presencia de los paquetes harían funcionar de mejor manera el sistema operativo UNLIX	Paquetes y el sistema en si para verificar si faltan o sobran paquetes	Estudio preliminar Análisis Investigación Observación

*Matriz de Consistencia Específica 1*



<b>PROBLEMA ESPECÍFICO:</b> Falta de un repositorio que actualice los paquetes del sistema.			
<b>OBJETIVO ESPECÍFICO</b>	<b>HIPÓTESIS ESPECÍFICA</b>	<b>UNIDAD DE OBSERVACIÓN</b>	<b>SISTEMA CATEGORIAL</b>
Crear un repositorio general de paquetes para el sistema operativo UNLIX utilizando la versión estable de debían Lenny	El repositorio sería la solución a los problemas de actualización del sistema	Sistema operativo UNLIX Apt get Paquetes	Análisis Diseño Base de datos Ingeniería de software Implementación

*Matriz de Consistencia Específica 2*



<b>PROBLEMA ESPECÍFICO:</b> La carencia latente de la actualización del sistema desde un servidor donde repose el repositorio			
<b>OBJETIVO ESPECÍFICO</b>	<b>HIPÓTESIS ESPECÍFICA</b>	<b>UNIDAD DE OBSERVACIÓN</b>	<b>SISTEMA CATEGORIAL</b>
Montar el repositorio estándar de paquetes para UNLIX en un servidor del Área de Energía de Industrias y Recursos Naturales no Renovables	El repositorio como servicio sería la forma más simple de actualización del sistema	Sistema operativo Paquetes Apt get Servidor Terminal	Análisis Diseño Servidores Redes

*Matriz de Consistencia Específica 3*



<b>PROBLEMA ESPECÍFICO:</b> La no presencia de un medio extraíble como fuente de alojamiento de un repositorio.			
OBJETIVO ESPECÍFICO	HIPÓTESIS ESPECÍFICA	UNIDAD DE OBSERVACIÓN	SISTEMA CATEGORIAL
Almacenar un repositorio primario en dvd para actualizar el sistema sin necesidad de conexión	El repositorio almacenado en dvd seria fuente de actualización del sistema.	Dvd Sistema operativo Paquetes Apt on dvd	Análisis Diseño Base de datos servidores

*Matriz de Consistencia Especifica 4*



<b>PROBLEMA ESPECÍFICO:</b> La insuficiencia de compatibilidad de algunos sistemas operativos en diferentes arquitecturas.			
<b>OBJETIVO ESPECÍFICO</b>	<b>HIPÓTESIS ESPECÍFICA</b>	<b>UNIDAD DE OBSERVACIÓN</b>	<b>SISTEMA CATEGORIAL</b>
Realización del plan de validación del repositorio de paquetes utilizando diferentes arquitecturas.	Las pruebas de validación verifican la amplia compatibilidad de las diferentes arquitecturas de los sistemas.	Pcs (algunas arquitecturas).  Sistema operativo  Paquetes	Experimentación  Pruebas

*Matriz de Consistencia Específica 5*



### 10.2 Matriz de Operatividad de Objetivos

<b>OBJETIVO:</b> Realizar un análisis a los paquetes de UNLIX para determinar posibles actualizaciones e inclusiones.						
ACTIVIDAD O TAREA	METODOLOGÍA	FECHA		RESPONSABLES	PRESUPUESTO	RESULTADOS ESPERADOS
		INICIO	FINAL			
Conocer el manejo y administración del sistema operativo Debian Lenny.	Instalar Debian Lenny, comenzar a manejarlo y estudiar el manejo del sistema en general	15/02/10	31/02/10	Víctor Cueva, Cesar Silverio.	\$ 50	Administración de los Sistemas Operativos Basados en GNU/LINUX
Conocer el manejo y administración del sistema operativo UNLIX.	Instalar Debian Lenny, comenzar a manejarlo y estudiar el manejo del sistema en general	01/03/10	07/03/10	Víctor Cueva, Cesar Silverio.	\$ 20	Conocer el funcionamiento y Administración del Sistema Operativo UNLIX.
Analizar los paquetes y su estructura de los Sistemas Operativos Basados en GNU/LINUX.	Leer y probar con algunos Sistemas Operativos GNU/LINUX como llevan estructurados los paquetes.	08/03/10	15/03/10	Víctor Cueva, Cesar Silverio.	\$ 20	Tener conocimiento de cómo están estructurados los paquetes de algunos sistemas operativos basados en GNU/LINUX



Matriz de Operatividad de Objetivos Específicos 1

<b>OBJETIVO:</b> Crear un repositorio general de paquetes para el sistema operativo UNLIX utilizando la versión estable de debían lenny.						
ACTIVIDAD O TAREA	METODOLOGÍA	FECHA		RESPONSABLES	PRESUPUESTO	RESULTADOS ESPERADOS
		INICIO	FINAL			
Investigar y Recopilar información de los paquetes necesarios y principales que manejan los sistemas operativos basados en GNU/LINUX.	Visitar Internet y probar los sistemas operativos como Debían Lenny y para comparación Ubuntu.	16/03/10	20/03/10	Víctor Cueva, Cesar Silverio.	\$ 50	Tener una base de datos de los paquetes más importantes que para que UNLIX pueda actualizar.
Observar el catálogo de paquetes disponibles en el sistema operativo UNLIX	Administrar el sistema operativo UNLIX.	21/04/10	01/04/10	Víctor Cueva, Cesar Silverio.	\$ 50	Tener la base de datos de los paquetes que posee UNLIX.
Comparar y verificar los paquetes que requieren actualizaciones entre los sistemas operativos Debían Lenny y UNLIX.	Comparación de los Sistemas Operativos Debían Lenny y UNLIX.	02/04/10	21/04/10	Víctor Cueva, Cesar Silverio.	\$ 50	Tener la base de datos de los paquetes que serán actualizados en UNLIX.



Matriz de Operatividad de Objetivos Específicos 2

<b>OBJETIVO:</b> Montar el repositorio estándar de paquetes para UNLIX en un servidor del Área de Energía de Industrias y Recursos Naturales no Renovables.						
ACTIVIDAD O TAREA	METODOLOGÍA	FECHA		RESPONSABLES	PRESUPUESTO	RESULTADOS ESPERADOS
		INICIO	FINAL			
Subir los paquetes en el servidor del área actualizados que serán usados por UNLIX	Montar los paquetes desde un cliente hacia un servidor remoto.	22/04/10	22/05/10	Víctor Cueva, Cesar Silverio.	\$ 150	Tener una variedad de paquetes actualizados en el servidor del área de Energía de la UNL.
Programar la interfaz para que el usuario de UNLIX administre las actualizaciones.	Diseño y programación de la interfaz para la comunicación	23/05/10	22/06/10	Víctor Cueva, Cesar Silverio.	\$ 150	Tener la una interfaz amigable para realizar la actualizaciones.
Realizar pruebas de comunicación para realizar las actualizaciones entre el servidor del área de Energía de la UNL y un cliente de UNLIX.	Pruebas de red usando los protocolos y puertos adecuados para la comunicación	23/06/10	01/08/10	Víctor Cueva, Cesar Silverio.	\$ 150	Poseer la comunicación entre el servidor del área de energía de la UNL y un cliente que use UNLIX.



Matriz de Operatividad de Objetivos Específicos 3

<b>OBJETIVO:</b> Almacenar un repositorio primario en DVD para actualizar el sistema sin necesidad de conexión						
ACTIVIDAD O TAREA	METODOLOGÍA	FECHA		RESPONSABLES	PRESUPUESTO	RESULTADOS ESPERADOS
		INICIO	FINAL			
Diseñar una presentación amigable para el DVD que llevara los paquetes para la actualización de UNLIX	Usar una aplicación web para que el DVD bootee y el usuario pueda hacer sus actualizaciones desde el DVD	02/08/10	18/08/10	Víctor Cueva, Cesar Silverio.	\$ 50	Una presentación amigable para el usuario de UNLIX
Elaborar el DVD con sus respectivas actualizaciones destinados especialmente para el sistema operativo UNLIX	Quemar el DVD, con su respectiva presentación y junto a las actualizaciones que se requiere.	19/08/10	01/09/10	Víctor Cueva, Cesar Silverio.	\$ 50	Tener listo el DVD de actualizaciones de UNLIX.

Matriz de Operatividad de Objetivos Específicos 4



<b>OBJETIVO:</b> Realización del plan de validación del repositorio de paquetes utilizando diferentes arquitecturas.						
ACTIVIDAD O TAREA	METODOLOGÍA	FECHA		RESPONSABLES	PRESUPUESTO	RESULTADOS ESPERADOS
		INICIO	FINAL			
Depurar y Corregir errores del proyecto.	Identificando los errores del proyecto	02/09/10	21/09/10	Víctor Cueva, Cesar Silverio.	\$ 150	Depurar el repositorio de paquetes de actualización de UNLIX.

Matriz de Operatividad de Objetivos Específicos 5



### 10.3 Matriz de control de resultados

	RESULTADOS	FECHA		FIRMA DIRECTOR TESIS
1	Conocer el manejo y administración del sistema operativo Debían Lenny.	15/02/10	31/02/10	
2	Conocer el manejo y administración del sistema operativo UNLIX.	01/03/10	07/03/10	
3	Analizar los paquetes y su estructura de los Sistemas Operativos Basados en GNU/LINUX.	08/03/10	15/03/10	
4	Investigar y Recopilar información de los paquetes necesarios y principales que manejan los sistemas operativos basados en GNU/LINUX.	16/03/10	20/03/10	
5	Observar el catálogo de paquetes disponibles en el sistema operativo UNLIX	21/04/10	01/04/10	
6	Comparar y verificar los paquetes que requieren actualizaciones entre los sistemas operativos Debían Lenny y UNLIX.	02/04/10	21/04/10	



<b>7</b>	Subir los paquetes en el servidor del área actualizados que serán usados por UNLIX	22/04/10	22/05/10	
<b>8</b>	Programar la interfaz para que el usuario de UNLIX administre las actualizaciones.	23/05/10	22/06/10	
<b>9</b>	Realizar pruebas de comunicación para realizar las actualizaciones entre el servidor del área de Energía de la UNL y un cliente de UNLIX.	23/06/10	01/08/10	
<b>10</b>	Diseñar una presentación amigable para el DVD que llevara los paquetes para la actualización de UNLIX	02/08/10	18/08/10	
<b>11</b>	Elaborar el DVD con sus respectivas actualizaciones destinados especialmente para el sistema operativo UNLIX	19/08/10	01/09/10	
<b>12</b>	Depurar y Corregir errores del proyecto.	02/09/10	21/09/10	

*Matriz de Control de resultados*