



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

HERRAMIENTA PARA AUTOMATIZAR LA CREACIÓN DE LIVE CDS PERSONALIZADOS

Sergio Arnaldo Méndez Aguilar

Asesorado por el Ing. Edgar René Ornelis Hoíl

Guatemala, octubre de 2009

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**HERRAMIENTA PARA AUTOMATIZAR LA CREACIÓN
DE LIVE CDS PERSONALIZADOS**

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR:

SERGIO ARNALDO MÉNDEZ AGUILAR
ASESORADO POR EL ING. EDGAR RENÉ ORNELIS HOÍL

AL CONFERÍRSELE EL TÍTULO DE
INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, OCTUBRE DE 2009

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Inga. Glenda Patricia García Soria
VOCAL II	Inga. Alba Maritza Guerrero de López
VOCAL III	Ing. Miguel Ángel Dávila Calderón
VOCAL IV	Br. José Milton De León Bran
VOCAL V	Br. Isaac Sultán Mejía
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Victor Hugo de León Barrios
EXAMINADOR	Ing. Juan Alvaro Díaz Ardavín
EXAMINADOR	Ing. Pedro Pablo Hernández Ramírez
SECRETARIA	Inga. Marcia Ivónne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**HERRAMIENTA PARA AUTOMATIZAR LA CREACIÓN
DE LIVE CDS PERSONALIZADOS,**

tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha julio de 2008.



Sergio Arnaldo Méndez Aguilar



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 21 de agosto de 2009.

Señor
Ing. Carlos Azurdia
Carrera de Ciencias y Sistemas
Facultad de Ingeniería
Universidad de San Carlos de Guatemala
Guatemala, Ciudad

Respetable Ing. Azurdia:

El motivo de la presente es para informarle que como asesor del estudiante Sergio Arnaldo Méndez Aguilar he procedido a revisar el trabajo de graduación titulado "Herramienta para automatizar la creación de Live CDs Personalizados" y que de acuerdo a mi criterio el mismo se encuentra concluido.

He tenido reuniones periódicas con el estudiante y luego de haber revisado cuidadosamente el trabajo, considero que cumple con los requisitos de calidad y profesionalismo que deben caracterizar a un futuro profesional de la informática.

Sin otro particular me suscribo de usted,

Atentamente,


Ing. Edgar René Ornelis Hoil
Ingeniero en Ciencias y Sistemas
Colegiado No 4830



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 26 de Agosto de 2009

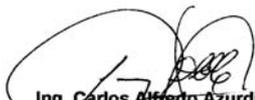
Ingeniero
Marlon Antonio Pérez Turk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **SERGIO ARNALDO MENDEZ AGUILAR**, titulado: "HERRAMIENTA PARA AUTOMATIZAR LA CREACION DE LIVE CDS PERSONALIZADOS", y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación



E
S
C
U
L
A

D
E

C
I
E
N
C
I
A
S

S
I
S
T
E
M
A
S

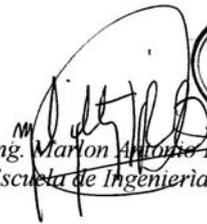
UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL.: 24767644

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, de trabajo de graduación titulado **“HERRAMIENTA PARA AUTOMATIZAR LA CREACIÓN DE LIVE CDS PERSONALIZADOS”**, presentado por el estudiante **SERGIO ARNALDO MÉNDEZ AGUILAR**, aprueba el presente trabajo y solicita la autorización del mismo.*

“ID Y ENSEÑAD A TODOS”



Ing. Marlon Antonio Perez Turk
Director, Escuela de Ingeniería Ciencias y Sistemas

Guatemala, 01 de octubre 2009

Universidad de San Carlos
de Guatemala



Facultad de Ingeniería
Decanato

Ref. DTG.367.2009

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **HERRAMIENTA PARA AUTOMATIZAR LA CREACIÓN DE LIVE CDS PERSONALIZADOS**, presentado por el estudiante universitario **Sergio Arnaldo Méndez Aguilar**, procede a la autorización para la impresión del mismo.

IMPRÍMASE.


Ing. Murphy Olimpo Paiz Recinos
DECANO

Guatemala, octubre de 2009



/gdech

ACTO QUE DEDICO A:

Dios en primer lugar, por haberme permitido conocerlo en esta universidad y darme una nueva oportunidad, para empezar a cambiar mi vida en mi actuar y en mi forma de pensar.

Mis padres y a mi hermana, que me brindaron todo el apoyo posible de acuerdo a sus capacidades, ya que ellos trabajaron muy duro para poder brindarme la oportunidad de lograr finalizar mis estudios universitarios.

Familiares, que me brindaron también su apoyo para poder avanzar en mis estudios y así finalizar mi carrera universitaria.

Amigos, que me apoyaron y con los cuales compartí muchas noches de trabajo duro, sacrificando muchas veces las salidas, por noches de desvelo y duro trabajo.

Catedráticos, que me impartieron clases, ya que a través de ellos fui formado académicamente y aprendí muchas cosas que me han servido para poder desenvolverme en el área laboral.

Mi asesor de trabajo de graduación, que me brindó su experiencia para poder llevar a cabo este trabajo.

La Universidad San Carlos de Guatemala, porque fue mi casa de estudios en la cual viví mucho tiempo y me entregué por completo para poder terminar mis estudios y así poder ayudar a mi país.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
LISTA DE SÍMBOLOS	IX
GLOSARIO.....	XI
RESUMEN.....	XVII
OBJETIVOS	XIX
INTRODUCCIÓN.....	XXI
1. ASPECTOS GENERALES.....	1
1.1. Historia del sistema operativo GNU/Linux	1
1.2. Kernel Linux.....	2
1.3. Gestor de arranque.....	3
1.4. Live CD.....	4
1.5. Formas de crear un Live CD.....	6
1.5.1. Knoppix	7
1.5.2. Catalyst	7
1.5.3. Slax	7
1.5.4. Remastering y remaster	8
1.6. Herramientas existentes para crear Live CDs	9
1.6.1. Intellibuild	9
1.6.2. MySlax Creator.....	10
1.6.3. LiveCD Fruid.....	11
1.6.4. Reconstructor	12
1.7. Elementos de Compresión.....	12
1.7.1. Unionfs	13
1.7.2. Squashfs LZMA	14

1.7.3. Aufs.....	14
2. ESTRUCTURA DE UN LIVE CD.....	17
2.1. Sistema de archivos (estructura de carpetas)	17
2.2. SysLinux.....	19
2.3. Initrd	20
2.4. Auto detección de dispositivos	20
2.4.1. Hotplug.....	21
2.4.2. Kudzu.....	22
2.4.3. Anaconda	22
2.5. Secuencia general de arranque	24
3. CARACTERÍSTICAS DE DISTRIBUCIONES BASE.....	27
3.1. Debian.....	27
3.2. Red Hat	29
3.3. Fedora.....	30
3.4. Slackware.....	31
3.5. Gentoo.....	32
3.6. LSB(Linux Standard Base).....	33
3.7. Análisis de distribuciones	35
4. MANEJO DE PAQUETES.....	39
4.1. Paquetes .deb	39
4.2. Paquetes .rpm	40
4.3. Paquetes .tgz	42
4.4. Portage.....	45

5. HERRAMIENTA PARA CREAR LIVE CDS	49
5.1. Elección de distribución GNU/Linux.....	49
5.1.1. Pruebas paquetería y estructura del sistema	49
5.1.2. Resultados.....	51
5.2. Definición de componentes.....	53
5.2.1. Kernel 2.6	53
5.2.2. Módulos LZM.....	53
5.2.3. Scripts en BASH.....	54
5.2.4. Scripts de Slackware	55
5.2.5. Live scripts.....	55
5.3. Arquitectura	56
5.3.1. Flujo de scripts de la herramienta.....	56
5.3.2. Procesos para crear un Live CD.....	57
5.4. Comparación con otras herramientas.....	60
5.5. Pruebas de la aplicación.....	62
5.6. Pruebas unitarias y de integración.....	62
5.7. Pruebas del sistema y aceptación	63
5.7.1. Resultados.....	64
CONCLUSIONES	67
RECOMENDACIONES.....	69
BIBLIOGRAFÍA	71
APÉNDICES.....	77

ÍNDICE DE ILUSTRACIONES

FIGURAS

1. Contenido de un Live CD basado en Knoppix	25
2. Diagrama de procesos para crear un Live CD.....	57
3. Diagrama de componentes de la herramienta.....	56
4. Pantalla inicial de la aplicación	88
5. Pantalla de ayuda.....	89
6. Pantalla de selección de idioma de teclado.....	89
7. Diálogo para prueba de idioma seleccionado.....	90
8. Diálogo para seleccionar partición de trabajo.....	90
9. Selección de tipo de formato a aplicar a la partición	91
10. Diálogo para prueba de idioma seleccionado.....	91
11. Pantalla de formateo de partición	92
12. Pantalla de selección de otras particiones	92
13. Carga de particiones al archivo /etc/fstab.....	93
14. Selección de sistemas mínimos pre configurados.....	93
15. Selección de paquetes disponibles	94
16. Opción para generar el Live CD basado en el sistema creado.....	97
17. Pantalla previa a la configuración del sistema.....	97
18. Diálogo de soporte de mouse en consola	98

19. Diálogo para poder configurar la red.....	98
20. Diálogo para ingresar el nombre de la máquina.....	99
21. Diálogo para ingresar dominio de la máquina	99
22. Diálogo para ingresar el proveedor de conexión.....	99
23. Diálogo de resumen de configuración de red.....	100
24. Diálogo de selección de servicios	100
25. Diálogo para poder seleccionar fuentes para consola	101
26. Diálogo para establecer contraseña de super usuario	101
27. Diálogo para poder seleccionar fuentes para consola	102
28. Ingreso de nombre del Live CD.....	102
29. Ingreso de uso de memoria para módulos	102
30. Mensaje para mostrar ubicación de archivos generados	103
31. Mensaje de ubicación de imagen ISO del Live CD creado	103

TABLAS

I. Organización de sistema de archivos de Debian.	17
II. Directorios comunes en / (La raíz)	18
III. Carpeta de contenido estático y variable	18
IV. Comparación entre distribuciones base	35
v. Características de paqueterías	46
VI. Características de herramientas para crear Live CDs.....	60

LISTA DE SÍMBOLOS

Símbolo	Significado
#	Indicador de línea de comandos.
.deb	Extensión de paquete de software basado en Debian.
.ISO	Extensión de una imagen de CD o DVD.
.rpm	Extensión de paquete de software basado en Red Hat.
.tgz	Extensión de un paquete de software basado en Slackware.
GB	Gigabyte.
MB	Megabyte.
MB/s	Megabytes por segundo.

GLOSARIO

BASH	BASH (<i>Bourne-again shell</i> , otro shell bourne) es un intérprete de comandos de Unix escrito para el proyecto GNU.
Cloop	Es un módulo del kernel Linux que da soporte para la compresión de dispositivos virtuales, comúnmente usado para comprimir un sistema de archivos para un Live CD.
CRAMFS	Es un sistema de archivos <i>ROM</i> de lectura, diseñado para ser simple y eficiente en el uso de espacio.
Ext2	ext2 (<i>second extended filesystem</i>) es un sistema de archivos para el kernel Linux que no implementa registro por diario(<i>Journaling</i>).
Ext3	ext3 (<i>third extended filesystem</i>) es un sistema de archivos con registro por diario (<i>journaling</i>).
Fluxbox	Es un administrador de ventanas basado en el código de Blackbox 0.61.1.
FHS	FHS (<i>Filesystem Hierarchy Standard</i>), es un estándar que define la estructura de carpetas del sistema de archivos para sistemas operativos GNU/Linux y algunos sistemas Unix.
FTP	FTP (<i>File Transfer Protocol</i>) Es un protocolo para transferencia de archivos.

GNOME	Gnome (<i>GNU Network Object Model Environment</i>) es un entorno de escritorio e infraestructura de desarrollo para sistemas operativos Unix/GNU/Linux, compuesto enteramente de software libre.
GNU	GNU es un acrónimo recursivo que significa <i>GNU is Not Unix</i> .
GRP	Es un paquete binario para instalar software en Gentoo.
GTK	Es una librería para construir interfaces gráficas.
HTTP	HTTP (<i>HyperText Transfer Protocol</i>) es el protocolo usado en cada transacción de la Web.
IEEE 1394	Es un estándar multiplataforma para entrada/salida de datos en serie a gran velocidad. Es usado comúnmente para cámaras digitales y videocámaras a computadoras.
Imagen ISO	Es un archivo donde se almacena una copia o imagen exacta de un sistema de ficheros, normalmente un disco compacto (como un CD o un DVD) .
iSCSI	Internet SCSI (iSCSI) es un estándar que permite el uso del protocolo SCSI sobre redes TCP/IP.
Jboss	Jboss es un servidor de aplicaciones J2EE de código abierto implementado en Java.
K3B	K3b (<i>KDE, burn, baby, burn</i>) es una interfaz gráfica para grabación de discos compactos y DVD bajo los sistemas operativos GNU/Linux y FreeBSD.
Loopback	El dispositivo de red <i>loopback</i> es un interfaz de red virtual, que representa al propio dispositivo independientemente de la dirección IP que se le haya asignado.

LVM	LVM es una implementación de un administrador de volúmenes lógicos para el kernel Linux.
MBR	MBR (<i>Master Boot Record</i>), es el sector de arranque de 512 bytes, que contiene la información de las particiones del disco duro.
Ncurses	Ncurses (<i>new curses</i>) es una biblioteca de programación que permite crear interfaces casadas en texto.
NFS	NFS (<i>Network File System</i>) es un protocolo de nivel de aplicación, según el Modelo OSI que es utilizado para sistemas de archivos distribuido en un entorno de red de computadoras de área local.
PCI	PCI (<i>Peripheral Component Interconnect</i>) es un bus de ordenador estándar para conectar dispositivos periféricos directamente a su placa base.
Qemu	QEMU es un emulador de procesadores basado en la traducción dinámica de binarios .
RAID	RAID (<i>Redundant Array of Inexpensive Disks</i>) se refiere a un sistema de almacenamiento, que usa múltiples discos duros entre los que distribuyen o replican los datos.
RAM	La memoria de acceso aleatorio RAM (<i>Random Access Memory</i>) es la memoria donde el procesador recibe las instrucciones y guarda los resultados, es una memoria temporal o volátil, ya que si se apaga la computadora se borran los datos que estaban almacenados en esta memoria.
Rsync	Es una aplicación de software para sistemas tipo Unix que ofrece transmisión eficiente de datos incrementales, comprimidos y cifrados.

S/390	Se refiere a la plataforma más rentable de hardware (servidores) de IBM.
SELinux	SELinux (<i>Security-Enhanced Linux</i>) es una característica de seguridad de Linux que provee una variedad de políticas de seguridad, a través del uso de módulos en el kernel Linux.
Splash	Imagen utilizada en la carga de un administrador de ventanas, o la imagen que se usa al cargar el sistema GNU/Linux.
Stage	Es la distribución Gentoo, con una cantidad de programas previamente compilados listos para ser usados.
Swap	Es el espacio de intercambio de un disco.
Tortilla Burner	Herramienta de origen guatemalteco para crear Live CDs basados en el Live CD de Slax con los paquetes de Slackware.
USB	El USB (<i>Universal Serial Bus</i> , bus universal en serie) es un puerto que sirve para conectar periféricos a una computadora.
Vanilla	Es el <i>software</i> de computadora que no ha sido personalizado y es distribuido en su forma original.
VMWare	VMware (VM de <i>Virtual Machine</i>) filial de EMC Corporation es la empresa que proporciona la mayor parte del <i>software</i> de virtualización disponible para ordenadores compatibles X86.
VNC	VNC (<i>Virtual Network Computing</i>) es un programa de software libre que permite tomar el control del ordenador servidor remotamente a través de un ordenador cliente.
XFCE	Es un entorno de escritorio ligero para sistemas tipo Unix que se configura íntegramente con el ratón o mouse.

Xorg X.Org es una implementación de código abierto del sistema *X Window System*, que surge como un proyecto diferente basado en XFree86.

Zisofs Es una extensión del sistema de archivos ISO 9660 que agrega soporte transparente en la compresión del sistema de archivos.

RESUMEN

Este trabajo consiste en la investigación y práctica de los conceptos necesarios para crear un Live CD a través del desarrollo de una herramienta que permita crearlos.

El primer capítulo abarca los conceptos básicos para poder crear un Live CD, entre estos podemos mencionar los conceptos de kernel, los sistemas de compresión de un Live CD, gestor de arranque, herramientas utilizadas para crear un Live CDs, etc.

El segundo capítulo tiene conceptos relacionados con la estructura de un Live CD, entre estos podemos mencionar la estructura del sistema de archivos, el concepto de initrd, syslinux y la auto detección de dispositivos.

El tercer capítulo tiene un análisis las características de las distribuciones consideradas base (Debian, Red Hat, Slackware y Gentoo) y el proyecto para estandarizar una distribución.

En el cuarto capítulo se analizan las características de los sistemas de administración de paquetes de las distribuciones base, mostrando las ventajas y desventajas de cada una.

Finalmente en el quinto capítulo se describe la herramienta desarrollada: Tortilla Burner para crear Live CDs, describiendo la arquitectura de la aplicación, los componentes que la integran y el proceso para crear Live CDs.

OBJETIVOS

General:

Crear una herramienta de software para el sistema operativo GNU/Linux, la cual permite crear Live CDs de una forma dirigida y automatizada en los diversos pasos en la construcción de un Live CD, de tal manera que la persona que utiliza la herramienta aprenda y practique los conceptos involucrados en la creación de un Live CD.

Específicos:

1. Crear un Live CD con la herramienta a desarrollarse.
2. Explicar los conceptos básicos que involucran la creación de un Live CD.
3. Crear un manual de usuario que explique el uso de la herramienta y facilite el uso de esta a usuarios de la comunidad GNU/Linux.
4. Introducir a los usuarios de sistemas GNU/Linux en Guatemala al desarrollo de estos sistemas a través de publicar o dar a conocer el trabajo, en algún congreso o sitio de software libre tales como sourceforge.net, freshmeat.net, slax.org, etc.

INTRODUCCIÓN

El uso del software libre o código abierto genera soluciones de alta calidad y de bajo costo, de tal forma que permite ser competitivo con respecto al software propietario. En Guatemala, el conocimiento del manejo de aplicaciones o soluciones a través de tecnologías de estas filosofías aun no ha sido explotado totalmente. Actualmente estas opciones están disponibles y no tiene costo alguno obtener una copia de estas tecnologías para desarrollo.

El desarrollo de proyectos basados en Software Libre permite dar a conocer el uso del software libre o código abierto en Guatemala. El desarrollo de la herramienta del presente trabajo (Tortilla Burner), permite que el usuario aprenda a crear sistemas GNU/Linux, conocer y manejar el software que utilizan estos sistemas para poder configurarlo y optimizarlo.

Tortilla Burner permite a los usuarios conocer los sistemas GNU/Linux para poder crear soluciones para fin comercial o no comercial, de alta calidad y bajo costo en el país de Guatemala, crear fuentes de trabajo y motivar a crear comunidades de usuarios para desarrollar proyectos que solucionen problemas y ayuden al país en el desarrollo informático.

En los siguientes capítulos se describen los conceptos básicos que sirvieron de base para crear la herramienta Tortilla Burner, para la automatización de la creación de Live CDs personalizados.

1. ASPECTOS GENERALES

Este capítulo trata sobre las bases y conceptos básicos para poder desarrollar un Live CD y establecer los criterios adecuados para seleccionar una distribución de GNU/Linux para el desarrollo de la herramienta para crear Live CDs.

1.1. Historia del sistema operativo GNU/Linux

El movimiento GNU/Linux inicia en el año 1971, cuando Richard Stallman comienza su carrera en el MIT, en un grupo de software libre. En la década de los ochenta, Richard Stallman concibió la *Free Software Foundation* (FSF) y el proyecto GNU. El proyecto GNU tenía como fin desarrollar un sistema compatible con Unix llamado GNU, desarrollado totalmente como software libre. En 1990, se habían encontrado o escrito la mayoría de los componentes mayores del sistema operativo excepto el kernel o núcleo¹.

Para el año 1991, un estudiante de la Universidad de Helsinki en Finlandia, llamado Linus Benedict Torvalds, comenzó a desarrollar su propio kernel como proyecto personal en su tesis sobre el sistema operativo minix. A principios del siguiente año, llegó a la versión estable 0.12 de su proyecto y dió el paso a la versión que fue llamada 0.95 que fue liberada bajo la licencia GPL.

¹ (Foundation(FSF), 2008)

Posteriormente se une Linux con el sistema GNU, para así formar el un sistema operativo totalmente libre llamado GNU/Linux².

1.2. Kernel Linux³

Muchas personas desconocen la relación que existe entre los términos GNU y Linux. El término Linux no es el sistema operativo, según Richard Stallman, Linux es el núcleo, es decir el programa que asigna los recursos de la máquina a los otros programas que se ejecutan. El kernel es esencial ya que solo no tiene funcionalidad. Para funcionar debe de estar en el contexto de un sistema operativo GNU, es decir un sistema GNU, con Linux como núcleo.

Según Richard Stallman, un sistema GNU/Linux es un sistema operativo libre que fue resultado del proyecto GNU, el cual creó un sistema libre basado en Unix, con un núcleo desarrollado inicialmente por Linux Torvalds. Dicho núcleo fue llamado Linux y la unión de estas dos partes formó el sistema operativo GNU/Linux.

² (Stewart, 2007)

³ (Stallman, 2003)

1.3. Gestor de arranque

Según el manual de administración de Suse⁴, un gestor de arranque (*bootmanager*) es un programa que se instala en el MBR (principalmente o en otro sector de un dispositivo de almacenamiento) de un disco duro y permite mantener varios sistemas operativos en un ordenador y alternar entre ellos. También permite al usuario seleccionar el sistema operativo durante el arranque. Para poder reiniciar y arrancar en otro sistema operativo es necesario reiniciar el ordenador y seleccionar el sistema operativo deseado.

Según el proyecto de GNU GRUB⁵, técnicamente un gestor de arranque es el primer programa de software que corre cuando la computadora inicia o arranque. Tiene la responsabilidad de cargar y transferir el control al kernel del sistema que este operando, luego se inicializa el resto del sistema operativo(ej.GNU).

Entre los gestores de arranque comúnmente usados tenemos al GRUB (*Grand Unified Bootloader*) y LILO (*Linux Loader*), que son gestores de arranque múltiple, es decir tienen la capacidad de iniciar desde uno de los sistemas operativos instalados en la computadora.

⁴ (Frank Bodammer, 2009)

⁵ (Okuji, 2006)

Según Laurence Bonney⁶, las diferencias entre estos dos gestores de arranque son las siguientes:

- LILO no tiene una interface interactiva, grub si lo soporta.
- LILO no soporta arranque del sistema a través de red, grub si lo soporta.
- LILO almacena la información con respecto a los sistema operativos (que puede cargar) físicamente en el MBR si cambiamos la información debemos de volverla a almacenar al MBR, en cambio grub almacena esta información en un archivo del disco duro, de tal forma que podemos modificar este archivo para mandar los parámetros respectivos para cargar un sistema GNU/Linux sin necesidad de escribir en el MBR cada vez que modifiquemos el archivo.

1.4. Live CD

Según Diego Fernando Nieto, “El origen de los Live CD, proviene de la necesidad de que muchas veces los usuarios de los Sistemas Operativos, buscan evaluar las características de un Sistema Operativo sin comprometer los datos y el Hardware de sus PC. Ante esto la comunidad de desarrollo del Software Libre, creo una distribución de Sistemas Operativos populares como Linux y FreeBSD en CD, así nace el concepto de Live CD”⁷.

⁶ (Laurence Bonney, 2005)

⁷ (Nieto, 2005)

Entre los conceptos más completos de Live CD podemos mencionar, según Juan Croissier define un Live CD como "un sistema operativo funcional que se puede ejecutar directamente desde un CD, en vez de tener que hacerlo desde el disco duro"⁸.

Según Diego Nieto "un Live CD es un Sistema Operativo totalmente configurado y adaptable a distinto Hardware del Usuario"⁹, usualmente este sistema operativo esta contenido en un CD o una memoria flash.

Complementando estos dos conceptos anteriores, un Live CD, es un sistema operativo funcional que se ejecuta directamente desde un CD, que contienen gran variedad de aplicaciones cargándolas a la memoria RAM, sin necesidad de tener que hacerlo desde el disco duro, teniendo o no la posibilidad de guardar cambios, o bien de poder instalar este sistema operativo en el disco duro.

Según los desarrolladores de Knoppix¹⁰, un Live CD puede ser usado como:

- Un sistema de demostración.
- CD educativo.
- Sistema de rescate.
- Adaptado para ser un sistema para presentar *software*.

⁸ (Croissier, 2002)

⁹ (Nieto, 2005)

¹⁰ (Knopper, 2009)

Un Live CD puede ser utilizado para tener maquinas con el sistema operativo GNU/Linux, sin necesidad de instalarlo, y de esta forma poder realizar pruebas de seguridad, rescatar datos en caso de que no arranca algún sistema operativo que tenemos instalado en el disco duro, detectar el hardware que utiliza nuestro computador, modificar archivos de nuestro sistema operativo instalado, para tratar de recuperar datos, etc.

Para usar un Live CD, el usuario descarga la imagen ISO desde Internet y luego la graba con su programa preferido para grabar CDs, posteriormente inicia la computadora desde la unidad de CD.

1.5. Formas de crear un Live CD

Las formas en las cuales podemos crear un Live CD son variadas, dependen de la tecnología de compresión de archivos que utilicemos y también de la distribución de GNU/Linux que vamos a escoger, teniendo también opción no solo de escoger formas ya existentes para crearlos, sino poder crear nuevas formas de hacerlos. En esta sección se describen las formas y técnicas para poder crear un Live CD.

1.5.1. Knoppix¹¹

Es un Live CD desarrollado por Klaus Knopper, basado en Debian GNU/Linux. Knoppix tiene una forma personalizada para iniciar la secuencia de arranque, también permite la detección de hardware y configuración de las propiedades del monitor, para usar el ambiente gráfico, la forma en que lo hace son originales de Knoppix. Incluye una gran variedad software para escritorio, análisis de redes, etc. Puede ser modificado para generar versiones personalizadas de Knoppix.

1.5.2. Catalyst¹²

Catalyst es una herramienta multifacética, la cual permite construir todos los aspectos de una versión de Gentoo: *stage tarballs*, *GRP package sets* y CDs de instalación. Esta dirigida a los usuarios que desean crear sus propios *stages* de Gentoo, Live CDs, etc. Catalyst es personalizable, expandible y mantenible.

1.5.3. Slax

Es un Live CD creado por Tomas Matejicek basado en Slackware GNU/Linux. Contiene un sistema modular de paquetes, es portable y pequeño lo cual permite que sea un Live CD rápido. Puede ser instalado en una memoria USB¹³.

¹¹ (Knopper, 2009)

¹² (Foundation, 2001)

¹³ (Matejicek, 2002)

Slax contiene una colección de software distribuida en módulos, estos módulos contienen software para soportar un ambiente gráfico, software para escrito, etc. También permiten al usuario poder crear su propia versión de Slax de forma sencilla¹⁴.

Para poder crear un Live CD basado en Slax, pueden descargarse los Live Scripts que se encuentran en el sitio www.linux-live.org, estos *scripts* son originalmente creados por Tomas Matejcek.

SLAX utiliza aufs y el squashfs, estos permiten tener un sistema de compresión de lectura y escritura en memoria, esto permite crear sistemas con más software dentro de un CD de 700 MB¹⁵.

1.5.4. Remastering y remaster

El *remastering*, se refiere a agregar paquetes y otras modificaciones al sistema de archivos de un Live CD el cual deseamos modificar. El realizar un *remastering*, lleva un tiempo considerable para generar la imagen ISO del nuevo sistema de archivos, pero realizarlo permite la optimización total del sistema de archivos del Live CD¹⁶.

¹⁴ (Matejcek, 2002)

¹⁵ (Matejcek, 2009)

¹⁶ (Rankin, 2004)

Por otro lado, el *remaster*, consiste en guardar configuraciones e instalar programas en un Live CD, para posteriormente cargar dichas configuraciones y programas al iniciar en otro momento el Live CD. El *remaster* no permite la optimización total del sistema de archivos del Live CD, ya que es solo de lectura. Esto no permite eliminar programas, solo agregar o modificar archivos de configuración.

1.6. Herramientas existentes para crear Live CDs

En esta sección se describen las características de algunas de las herramientas que existen para crear Live CDs.

1.6.1. Intellibuild¹⁷

Intellibuild (iBuild) es un programa que permite crear Live CDs personalizado como Knoppix y Morphix.

El funcionamiento de iBuild consiste en abrir una plantilla, seleccionar los programas a incluir, presionar el botón “Build” y esperar a que genere la imagen ISO que posteriormente se deberá de grabar en un CD.

iBuild es que una herramienta sencilla y rápida con plantillas que permiten personalizar muchas opciones del Live CD, como la imagen de *bootsplash*, configuraciones de archivos de arranque, etc.

iBuild permite crear Live CDs basados en la distribución Debian.

¹⁷ (Jure Cuhalev, 2009)

1.6.2. MySlax Creator¹⁸

MySLAX Creator es una herramienta que permite crear un Live CD basado en Slax utilizando los Live Scripts, esta herramienta está en su versión de Windows, permitiendo ejecutar en Windows XP/2000/NT. Una opción disponible para Windows es posible emulando su ejecución a través del programa Wine.

Entre las características de este programa podemos mencionar:

- Agrega módulos (programas en Slax comprimidos), a los directorios /modules y /optional (extensiones .mo y .img).
- Agrega archivos generales a la carpeta /extra.
- Remueve módulos de una imagen ISO.
- Cambia las opciones de arranque del Live CD.
- Crea una imagen ISO personalizada de MySLAX.
- Graba en un CD la imagen ISO generada.
- Prepara una memoria USB *bootable* copiando todos los archivos de SLAX a ella.

Esta herramienta puede ser utilizada con otros derivados de SLAX como Klax, gNOX, GobliX y Mutagenix pero solo existe retroalimentación para estos casos en foros para los usuarios de SLAX.

¹⁸ (About MySLAX Creator, 2005)

1.6.3. LiveCD Fruid¹⁹

LiveCD Fruid es una herramienta que ayuda a crear un Live CD personalizado, basándose en la distribución Mandriva.

Esta herramienta contiene un conjunto de diálogos para el entorno gráfico X, los cuales permiten realizar una gran cantidad de configuraciones personalizadas con respecto a la configuración del sistema a ser creado como un Live CD.

Las características que incluye esta herramienta son:

- Opciones simples y avanzadas de personalización.
- Un indicador de progreso de desarrollo del Live CD.
- Regresar a pasos anteriores de personalización.
- Personalizar la imagen de *splash* de el Live CD y el ambiente del usuario.
- Configurar el tamaño de los archivos ISO del Live CD generado.
- Seleccionar los datos que se van a incluir o excluir en el Live CD.

¹⁹ (Walt, 2005)

1.6.4. Reconstructor²⁰

Reconstructor es un herramienta para crear Live CDs de Ubuntu. Permite personalizar el disco de Desktop (Live), Alternate (Install) o el disco Server de Ubuntu.

Para la personalización del Desktop (Live), permite modificar todo el ambiente, agregar o quitar software, el tema default (*splash*, temas, fuentes, fondo de escritorio, etc), agregar *links* en el escritorio, etc.

Para el disco de Alternate y Server, permite agregar software adicional al disco que se desea instalar.

Reconstructor esta escrito en lenguaje Python y esta liberado bajo la licencia GNU General Public License (GPL).

1.7. Elementos de Compresión

Una parte importante en un Live CD, es el sistema de compresión que tiene el sistema de archivos, hay distintas formas de compresión para el sistema de archivos, entre ellas podemos mencionar Unionfs, Squashfs LZMA y Aufs. En las siguientes secciones se detalla cada uno de estos sistemas de archivos.

²⁰ (Reconstructor, 2009)

1.7.1. Unionfs

Unionfs fue inicialmente desarrollado por el profesor Erez Zadok junto con su equipo, en la Universidad Stony Brook²¹.

El sistema de archivos Unionfs, nos permite unir varias estructuras de archivos (ramas) en un solo directorio, manteniendo el contenido de cada carpeta física por separado²².

Actualmente el sistema de archivos Unionfs, es usado en sistemas Live CD o sistemas sin disco (arranque desde red), donde el directorio de solo lectura root y es combinado con el sistema de archivos tmpfs (que reside en la memoria y es modificable) para que los archivos de lectura de root pueden ser modificados, y la modificación puede guardarse en la memoria (en tmpfs). Unionfs fue reemplazado por muchas de las nuevas características que tiene el sistema aufs. Entre algunos proyectos que utilizan o utilizaron Unionfs podemos mencionar los siguientes Live CDs: Knoppix, Kanotix, Mandriva One, Morphix, entre otros²³.

²¹ (Okajima, 2007)

²² (Unionfs: A Stackable Unification File System, 2007)

²³ (Matejicek, 2007)

1.7.2. Squashfs LZMA²⁴

El código inicial de este proyecto fue realizado por Junjiro Okajima. El proyecto Squashfs-LZMA contiene el algoritmo LZMA que puede comprimir mejor que gzip, el sistema de archivos squashfs actualmente utiliza el algoritmo LZMA.

La compresión a través del formato LZMA es lento, pero el sistema squashfs solo necesita que sea compresionado una vez. Al ser leído millón de veces, los desarrolladores tuvieron especial cuidado en la velocidad de descompresión. LZMA es muy rápido en la descompresión alrededor de 10-20 MB/s en un CPU de 2GHz.

Actualmente Squashfs no soporta la compresión LZMA directamente en el código fuente y no está incluido en el kernel Linux, el agregar el soporte de LZMA a squashfs disminuye la posibilidad de la inclusión de squashfs en Linux. Debido a esta razón, el soporte de LZMA solo se puede conseguir a través de parches que se pueden obtener en el sitio de <http://www.squashfs-lzma.org>.

1.7.3. Aufs²⁵

Aufs fue creado por Junjiro Okajima y nace del rediseño y re implementación del Unionfs de las versiones 1.x. Aufs se origina cuando su creador tuvo varias ideas e implementaciones y es así que crea un Unionfs muy distinto de las características básicas del original Unionfs.

²⁴ (Junjiro Okajima, 2006)

²⁵ (Okajima, 2007)

Aufs es importante y utilizado comúnmente en los Live CD, permitiendo la unión del contenido de archivos y modificación del contenido de los archivos en memoria.

Entre las características de Aufs se encuentran:

- Une varios directorios en un simple sistema de archivos virtual, el cual es llamado *branch* o rama.
- Tiene la opción de poder especificar los permisos la rama, los cuales pueden ser 'readonly', 'readwrite' y 'whiteout-table'.
- La rama superior, archivos/directorios en la rama de solo lectura son modificables lógicamente.
- Hay manipulación dinámica de la rama con operaciones como agregar y borrar.
- Muchas de las características de su predecesor Unionfs.
- Seguro y rápido.
- Un *loopback* es montado como una rama.
- Mantiene automáticamente /etc/mstab con un script externo localizado en /sbin/mount.aufs.
- Revierte una llamada al sistema cuando un error aparece con el aufs.
- Permite exportar vía NFS.
- Soporta directorios de NFS de lectura.
- Ofrece acceso directo a un archivo en la rama.

En general, Aufs es un sistema de archivos derivado y mejorado de la serie de Unionfs versiones 1.x, el cual permite unir varios directorios en un simple sistema de archivos virtual, el cual es llamado *branch* o rama y por sus características ha sido incluido en proyectos de desarrollo de Live CDs, tales como: Slax, Knoppix y Gentoo (Catalyst).

2. ESTRUCTURA DE UN LIVE CD

El estudio de la estructura del sistema de archivos de un Live CD es importante para lograr el desarrollo de una forma más adecuada de la herramienta propuesta, en este capítulo se abordan los temas sobre la estructura del Live CD a generar por la herramienta.

2.1. Sistema de archivos (estructura de carpetas)²⁶

Un sistema de archivos es el método y la estructura de datos que un sistema operativo usa para almacenar archivos en un disco o partición. Comúnmente Linux organiza su sistema de archivos iniciando de un directorio raíz llamado root (/).

Tabla I. Organización de sistema de archivos de Debian.

/bin	Comandos esenciales.
/boot	Archivos estáticos del gestor de arranque.
/dev	Archivos de dispositivos.
/etc	Archivos de configuración del sistema.
/lib	Librerías esenciales compartidas y módulos del kernel.
/media	Punto de montaje para medios extraíbles.
/mnt	Punto de montaje para montar un sistema de archivos temporal.
/opt	Software agregado al instalar paquetes.
/sbin	Binarios esenciales del sistema.
/srv	Datos para los servicios provistos por el sistema actual.
/tmp	Archivo temporales.
/usr	Jerarquía secundaria
/var	Datos variables

²⁶ (Nguyen, 2004)

En Tabla I se detalla estructura jerárquica de carpetas que utiliza el sistema GNU/Linux basado en un Debian 3.0r0 (Woody) .

En un sistema de archivos se pueden encontrar los siguientes directorios o enlaces simbólicos en el / (La raíz), si el subsistema correspondiente esta instalado:

Tabla II. Directorios comunes en / (La raíz)

/	Directorio raíz
/home	Carpeta para almacenar la configuración y archivos de usuarios (Opcional).
/lib	Carpeta para compartir librerías(Opcional).
/root	Carpeta principal del administrador del sistema.

En esta estructura general se encuentran archivos que se pueden “compartir”, es decir archivos que son definidos en un *host* y utilizados en otros *host*, y archivos “no compartibles”, es decir archivos que no son compartibles, por ejemplo: los archivos en los directorios de un usuario se pueden “compartir”, mientras que los archivos de bloqueo de dispositivos son “no compartibles”.

Los archivos “estáticos” incluyen archivos binarios, librerías, documentación de archivos y otros archivos que no se cambian sin la intervención de administrador de sistema. Los archivos “variables” son definidos como archivos que no son estáticos

Tabla III. Carpeta de contenido estático y variable

	Compartibles	No compartibles
Estáticos	/usr /opt	/etc /boot
Variables	/var/mail /var/spool/news	/var/run /var/lock

Esta parte constituye de una forma general de la estructura de carpetas dentro de un sistema operativo GNU/Linux, y constituye parte del conocimiento necesario para tener claridad en la forma de utilizar los archivos en un sistema GNU/Linux.

2.2. SysLinux²⁷

Para desarrollar un Live CD es importante utilizar una herramienta que permita brindar el soporte para poder iniciar un sistema operativo contenido en un CD. El proyecto SYSLINUX comprende una serie de herramientas (gestores de arranque) que permiten cargar un sistema operativo dependiendo el medio. Entre estos: SYSLINUX permite iniciar un sistema operativo contenido en un sistema FAT (MS-DOS), PXELINUX permite cargar el sistema a través de red, ISOLINUX permite cargar un sistema operativo desde un CDROM y EXTLINUX permite cargar el sistema en una partición ext2 o ext3. Adicionalmente se proporciona la herramienta MEMDISK que permite arrancar sistemas operativos propietarios (como DOS), este es usado en conjunto con PXELINUX y ISOLINUX.

En general el proyecto SysLinux contiene gestores de arranque para diversos escenarios para poder arrancar el sistema operativo GNU/Linux.

²⁷ (The Syslinux Project, 2009)

2.3. Initrd²⁸

Un initrd (Initial RAM disk) es un sistema de archivos inicial, el cual es cargado antes que el sistema principal de archivos este disponible. Tiene un conjunto mínimo de directorios, módulos y ejecutables (como insmod).

Para que el sistema utilice el initrd, primero se debe de cargar el kernel posteriormente el initrd. Este da el soporte necesario para cargar el sistema de archivos principal.

El initrd es que es un sistema de archivos transitorio y su tiempo de vida es corto.

2.4. Auto detección de dispositivos

La detección y configuración de los dispositivos del sistema operativo es muy importante para dar funcionamiento y vida a un sistema operativo, sin esta es imposible realizar muchas tareas como transferir datos, crear datos, etc.

En esta sección se describen algunas de las formas de configurar dispositivos en el sistema operativo GNU/Linux.

²⁸ (Jones, 2003)

2.4.1. Hotplug²⁹

El Hotplug en el sistema Linux comenzó en el kernel 2.4 (en enero 2001), el *hotplugging* es una característica estándar de un sistema operativo GNU/Linux.

Hotplug permite conectar nuevos dispositivos y utilizarlos inmediatamente. Esto facilita a los usuarios usar sus dispositivos sin la necesidad de conocer mucho de la administración del sistema operativo. Una característica básica de los sistemas operativos debe de ser que los sistemas deben de ser parcialmente auto configurables por si mismos.

El Hotplug incluye soporte para USB y dispositivos PCI (Cardbus) y también se pueden configurar automáticamente algunas interfaces comunes de red. Otras versiones más nuevas de Hotplug incluyen soporte para IEEE 1394 (Firewire/i.Link) y se pueden descargar otros controladores para dispositivos USB licenciados (*firmware*). También hay soporte para equipos portables y mainframes S/390.

Para el kernel Linux 2.6, el Hotplug fue integrado con el modelo de controladores, de tal manera que cualquier bus o clase pueda reportar eventos de *hotplug* cuando los dispositivos son agregados y removidos.

Hotplug es soportado en muchas distribuciones de GNU/Linux tales como: RedHat, Debian, etc.

²⁹ (Linux Hotplugging, 2009)

2.4.2. Kudzu³⁰

Kudzu es una herramienta diseñada por Red Hat para detectar y configurar dispositivos, en el momento de carga del sistema detecta si algún dispositivo de hardware ha sido agregado o removido. Es conveniente utilizar Kudzu si se piensa agregar algún nuevo dispositivo.

El proceso básico de configuración de dispositivos consiste que al iniciar Kudzu se detecta el hardware actual y chequea si está en su base de datos (*/etc/sysconfig/hwconf*). Entonces se determina si fue agregado o removido hardware del sistema. Esto permite a los usuarios configurar hardware agregado o quitar la configuración cuando el hardware es removido. Luego actualiza la base de datos.

De esta forma Kudzu administra los dispositivos en el sistema. Es importante mencionar que kudzu se ejecuta al iniciar el sistema, también se puede configurar para iniciar siempre al arranque y que no afecte en el rendimiento del sistema³¹.

2.4.3. Anaconda³²

Anaconda es el programa para instalar un sistema operativo GNU/Linux utilizado por Fedora, Red Hat Enterprise Linux y algunas otras distribuciones.

³⁰ (Linux/Unix Command kudzu, 2009)

³¹ (DiMaggio, 2009)

³² (Fedora, 2009)

El modo de operar de Anaconda es el siguiente: durante la instalación, el hardware del computador es identificado y configurado y los sistemas de archivos apropiados son creados dependiendo de la arquitectura del sistema. Luego Anaconda permite al usuario instalar el software del sistema operativo en el computador. Anaconda también puede actualizar una instalación existente de versiones anteriores de la misma distribución. Luego de que la instalación es finalizada, se puede reiniciar al sistema instalado y continuar realizando la personalización del sistema luego de entrar por primera vez.

Por otra parte, Anaconda es un instalador bastante sofisticado de tal forma que permite instalaciones de forma remota o local con CDs, DVDs, imágenes almacenadas en un disco duro, NFS, HTTP y FTP.

Entre otras de sus características tenemos que el proceso de instalación puede realizarse de forma rápida para proveer una instalación que puede ser duplicada en otras máquinas, esto se puede realizar debido a que la instalación permite soporte de VNC en máquinas sin monitor, esto en caso de un ambiente de red. Anaconda reconoce una variedad de dispositivos de almacenamientos avanzados que incluyen LVM, RAID, iSCSI y multipath los cuales son soportados por un programa de particionamiento que incluye Anaconda.

2.5. Secuencia general de arranque³³

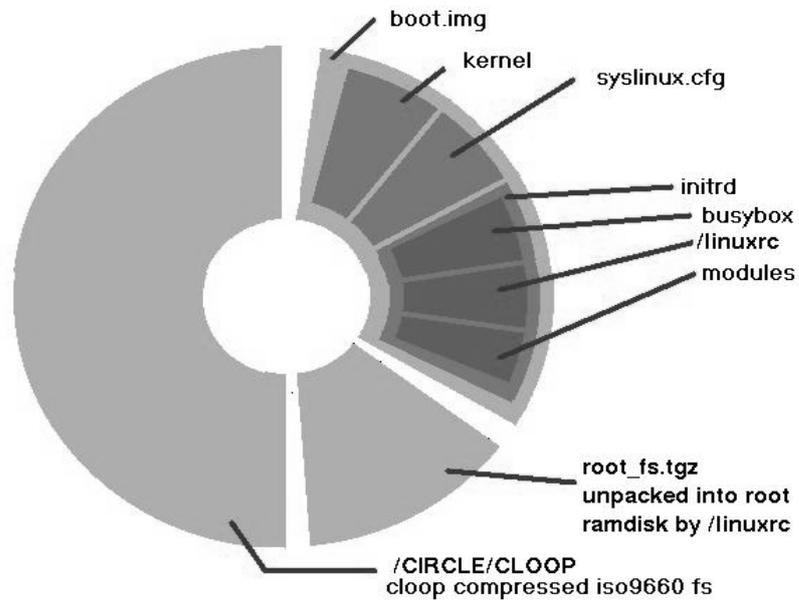
La secuencia general de arranque de un Live CD consta de los siguientes pasos:

1. Se carga el Torito y realiza la creación de una o más imágenes de disco en el CD-ROM.
2. El BIOS localiza estas imágenes y crea una imagen (imágenes de diskettes o discos duros) de disco emulado desde donde carga. También está el modo no-emulation, en donde el BIOS carga los sectores desde un archivo específico y lo ejecuta sin necesidad de emular discos.
3. El gestor de arranque (Ej. SYSLINUX) llama la imagen del kernel del sistema. Antes de pasar el control al kernel, llama al initrd, cargando los módulos necesarios para cargar el sistema raíz y ejecutando las instrucciones contenidas en el script linuxrc dentro del initrd, cediendo luego el control al kernel.
4. El kernel da soporte al hardware del computador, ejecutando finalmente el script INIT del sistema, el cual contiene el conjunto de scripts a ejecutarse en los niveles de ejecución del sistema.

Estos pasos permiten la inicialización de servicios y programas del sistema. Dependiendo del software instalado, podremos ejecutar un entorno gráfico y otros programas. El contenido general del sistema de archivos de un Live CD se muestra en la Figura 1.

³³ (Barlow, 2005)

Figura 1. Contenido de un Live CD basado en Knoppix



Fuente: Daniel Barlow³⁴

La gráfica muestra las partes básicas de un Live CD, sin estos archivos no es posible cargar un sistema vivo en este caso un Live CD, desde el momento de arranque del computador. Entre todos estos archivos contienen un sistema de carga para poder dar soporte al hardware del computador al cargar el sistema GNU/Linux disponible desde el USB.

³⁴ (Barlow, 2005)

3. CARACTERÍSTICAS DE DISTRIBUCIONES BASE

Este capítulo describe las diferentes formas que tienen las distribuciones GNU/Linux para el sistema de administración de paquetes, niveles de ejecución, configuración de programas y servicios, desarrollo de software, comunidades de usuarios, etc.

3.1. Debian

El desarrollo de Debian lo comenzó Ian Murdock en agosto de 1993, con la finalidad de crear una distribución abierta. Esta distribución se maneja a través del proyecto de desarrollo llamado “El Proyecto Debian”, y tiene como finalidad desarrollar un sistema operativo libre. Contiene más de 18,733 paquetes los cuales se pueden obtener gratuitamente.

La estabilidad de las versiones que ha publicado han sido de bastante calidad en los últimos años, esto ha hecho que muchas distribuciones se basen en este sistema. Tal es el caso de la distribución Ubuntu, que permite utilizar los repositorios de Debian, para instalar paquetes no incluidos en sus paquetes oficiales. Knoppix que es un Live CD muy popular, basado en Debian, aportó nuevas tecnologías a la comunidad GNU/Linux; en el caso de Knoppix, fue uno de los sistemas pioneros como Live CD, que introdujeron su propia forma de configurar dispositivos, configurar sistema X, etc.

El sistema de administración de paquetes es manejado a través de *apt-get*, utiliza una de las dos paqueterías más utilizadas por las distribuciones GNU/Linux y contiene el repositorio más grande de paquetes para una distribución.

Debian es un sistema avanzado que necesita tener cierto conocimiento de sistemas GNU/Linux. Soporta varios lenguajes y tiene muchas comunidades que dan soporte a usuarios de Debian³⁵.

Para liberar una versión de Debian, se comienza a través del desarrollo de una versión *unstable* (inestable), los nombres de las versiones de Debian hacen referencia a los personajes de Toy Story. La versión *unstable* o de desarrollo tiene una rama de desarrollo llamada *sid*. La versión *testing* es generada al trasladar paquetes de la versión *unstable* con ciertos criterios a la versión *testing*. Luego de ciertas revisiones de satisfacer ciertos criterios, los desarrolladores de Debian generan una versión a partir de la versión *testing* a la que llaman *stable*³⁶.

Debian cuenta con un proceso arduo para poder ser un desarrollador y mantenedor oficial de paquetes, es por eso que sus paquetes tienen una gran estabilidad.

³⁵ (polishlinux.org - Debian, 2009)

³⁶ (Debian, 2009)

3.2. Red Hat³⁷

Red Hat es la compañía responsable de la creación y mantenimiento de una distribución del sistema operativo GNU/Linux principalmente Red Hat Enterprise Linux y Fedora. Red Hat Enterprise Linux es la versión empresarial de Red Hat y se desarrolla a través de una versión experimental, la actual distribución Fedora. Luego que los desarrolladores prueban que se cumplan con ciertos estándares de calidad, la versión de Fedora pasa a ser Red Hat Enterprise Linux. En los *middleware* patrocina a jboss.org y distribuye la versión profesional bajo la marca JBoss Enterprise.

Red Hat ha logrado la comercialización de diferentes productos y servicios basados en software de código abierto. Esta empresa cuenta con una amplia infraestructura en la que se cuentan más de 2.000 empleados en 28 lugares del mundo.

Los Programadores empleados de Red Hat han desarrollado múltiples paquetes de software libre. Algunos de los aportes a la comunidad de software libre han sido el sistema de administración de paquetes (RPM) y varias utilidades para la administración y configuración de equipos, como *sndconfig* o *mouseconfig*.

Debido a su popularidad y estabilidad, varias distribuciones han sido basadas en el sistema de administración de paquetes RPM que maneja Red Hat, entre ellas podemos mencionar: Mandriva Linux, Yellow Dog Linux (sólo para PowerPC), y CentOS (compilada a partir de las fuentes de Red Hat).

³⁷ (kamiyeye, 2009)

3.3. Fedora³⁸

Fedora es un proyecto abierto independiente de Red Hat y es un sistema para propósito general que no está diseñado para un grupo de usuario en específico, lo cual permite que sea una distribución apta para usuarios de casa, programadores y servidores corporativos, etc. Incluye software reciente y fue la primera distribución con SELinux en la instalación por defecto.

Fedora tiene varias herramientas para desarrolladores entre estas están el IDE Eclipse, librerías para desarrollo web y Mono (soporte para lenguajes de Microsoft) y aplicaciones para desarrollo disponibles sin ningún costo.

Incluye entornos gráficos de escritorios más populares como GNOME y KDE, XFCE y Fluxbox. El DVD de instalación contiene todos los paquetes necesarios para instalar cualquiera de los entornos gráficos anteriormente mencionados sin necesitar versiones específicas de la distribución que incluyan un escritorio en específico.

Es importante mencionar que Fedora constituye la rama experimental de Red Hat Enterprise Linux, esto proyecto nace de la versión de Red Hat 9.0. A partir de esto nace el proyecto Fedora, y la línea empresarial Enterprise Linux³⁹.

³⁸ (polishlinux.org - Fedora, 2009)

³⁹ (Johnson, 2003)

3.4. Slackware⁴⁰

La distribución Slackware fue iniciada por Patrick Volkerding a finales de 1992, y liberada el 17 de Julio de 1993. Volkerding utilizó una de las distribuciones de esa época SLS Linux (creada por Soft Landing Systems). Volkerding compuso errores cuando los encontraba y los combinó en su propia distribución privada para que él y sus amigos pudieran usarla. Esta distribución privada rápidamente ganó popularidad, así que Volkerding decidió llamarlo Slackware y hacerlo públicamente accesible. La evolución inicial de Slackware incluyó una instalación amigable al usuario basada en menús de modo texto, como también un sistema para manejo de paquetes, que permitía a los usuarios; agregar, remover y actualizar paquetes de software en sus sistemas.

Slackware es un avanzado sistema operativo GNU/Linux, diseñado con las características esenciales de ser fácil de usar y estable. Incluye el software más popular, siempre reteniendo un sentido de tradición en su estructura, simplicidad, facilidad, flexibilidad y poder. Cumple con los estándares Linux que se han publicado, como el sistema de archivos Linux estándar (FHS) y está entre una de las distribuciones más populares, estables y amigables entre todas las distribuciones GNU/Linux que existen. Principalmente está enfocado a usuarios avanzados que demandan el control en el sistema.

Slackware utiliza un kernel vanilla, es decir la funcionalidad del sistema no disminuye si se reemplaza el kernel por defecto y se recompilara. La distribución no tiene un instalador gráfico pero si uno basado en una interfaz de modo texto con Ncurses.

⁴⁰ (Slackware, 2009)

No hay asistentes gráficos como Yast de Novell, solo asistentes básicos del sistema como: Instalación de paquetes, selección del entorno gráfico.

Slackware es una de las distribuciones más antiguas que aun sobreviven. Trata de ser lo más parecido a Unix. No trata de sobresalir con interfaces gráficas para manejar procesos, sino que coloca a los usuarios en el mando de control y les permite aprender como funciona el sistema. El desarrollo de esta distribución no es apresurado, es decir los desarrolladores prefieren tener el sistema estable antes de liberarlo para el uso de los usuarios.

3.5. Gentoo

Gentoo es un sistema operativo libre que puede ser basado tanto en GNU/Linux como en FreeBSD. Es uno de los sistemas más configurables, extensibles y adaptables, es por eso que Gentoo es llamado una meta distribución, esto permite al usuario tener la parte de sistema que quiere. Para la administración de paquetes se utiliza el Portage (una herramienta basada en BSD para administración de software), este permite que el sistema entero o algún paquete en específico puede ser optimizado para el hardware, de tal forma que permite aumentar la robustez del sistema⁴¹.

El proceso de instalación de Gentoo permite configurar y optimizar todo el sistema entero; kernel, software a instalar, administración de usuarios, etc. Con respecto a la configurabilidad, cada aspecto del sistema puede ser configurado por el usuario. Es importante mencionar que el usuario necesita conocer el sistema muy bien para luego poder configurar el sistema operativo.

⁴¹ (Gentoo, 2009)

Gentoo cuenta con una gran cantidad de desarrolladores, los cuales proveen documentación, infraestructura (lista de correos, sitios, foros, etc), calidad de desarrollo, seguridad, etc., es decir todo de una forma muy profesional⁴².

3.6. LSB(Linux Standard Base)

El proyecto LSB (*Linux Standard Base*) propone un conjunto de estándares para garantizar que el software se pueda ejecutar en cualquier distribución, reducir las diferencias entre distribuciones para lograr que el software sea portable y disminuir los costos de desarrollo para los distribuidores de software independientes para plataformas GNU/Linux⁴³. Para lograr la portabilidad es importante manejar un estándar para crear aplicaciones (LSB) y un estándar de sistemas de archivos (FHS, *Filesystem Hierarchy Standard*)⁴⁴.

LSB ayuda a los proveedores de software que han tenido problemas para que su software pueda ejecutarse independientemente del sistema de administración de paquetes.

El estándar LSB en su siguiente versión propone resolver el problema con una interface que actúe como un puente entre los sistemas de paquetes y los instaladores de software. La definición de proponer un estándar en la paquetería en LSB a tenido problemas y ha sido poco aceptado, por eso se propone tener un sistema de paquetes común, como: autopackage.

⁴² (polishlinux.org - Gentoo, 2009)

⁴³ (Foundation L. , 2009)

⁴⁴ (Porting to the LSB, 2009)

Murdock miembro de LSB y líder del proyecto Debian, rechaza la idea de crear un nuevo sistema como estándar, ya que convencer al mundo que rechace la inversión que ha hecho estando con una distribución no es razonable. El estandarizar el manejo de la administración de paquetes ha provocado conflictos ya que cada distribución defiende sus ventajas en su sistema de administración de paquetes, este es el caso del conflicto de los miembros de las distribuciones Debian y Red Hat que lideran el desarrollo LSB. Una crítica de LSB es que aparenta favorecer a los vendedores de software esto es a causa de que el proyecto está liderado por empresas grandes de software como IBM⁴⁵.

Para usar un estándar, el diseño de un programa debe basarse y acoplarse a su estructura. Este fue el caso de HTML, cuando antes las páginas estaban diseñadas para los navegadores de Internet Explorer y Netscape. LSB es bueno para las compañías cuando hablan de estándares, ya que es algo que les permite brindar o certificar confianza de sus aplicaciones⁴⁶.

El creador Slackware menciona que adoptaría LSB si tiene éxito, de lo contrario sería una pérdida de tiempo. Si LSB funciona traería beneficios a la distribución, pero a su vez menciona que los estándares como LSB y FHS son de beneficio. Slackware utiliza el FHS, y es una de las distribuciones más estándar que hay actualmente. La implementación del estándar LSB es una incertidumbre que tienen muchas distribuciones⁴⁷.

⁴⁵ (Byfield, 2007)

⁴⁶ (Rankin, 2004)

⁴⁷ (Roblino, 2000)

3.7. Análisis de distribuciones

A continuación se muestra una tabla con las ventajas y desventajas de las distribuciones que han sido pioneras y de las cuales se han derivado muchas otras distribuciones.

Tabla IV. Comparación entre distribuciones base

Características	Debian	Red Hat	Fedora	Slackware	Gentoo
Kernel Linux	X	X	X	X	X
Kernel Hurd	X				
Otros kernel	X				
Varias arquitecturas	X	X	X	X	X
Comunidad de usuarios	X	X	X	X	X
Amplia cantidad de paquetes oficiales	X	X	X		X
Paquetes extra	X	X	X	X	X
Sistema amigable		X	X		
Interfaz gráfica de configuración		X			
Interfaz de texto para configuración	X	X	X	X	
FHS				X	
LSB	X	X	X		
Multipropósito			X		
Versión empresarial		X			
Certificaciones	X	X			
Gratuita	X		X	X	X
Paquetes optimizados para máquina					X
Liviana para soportar hardware viejo				X	

Debian es una distribución muy popular y estable que soporta diversos kernel como el Hurd, el kernel de FreeBSD, etc. Debian se caracteriza por tener el repositorio de paquetes oficiales y no oficiales más grandes de todas las distribuciones.

Debian cuenta con grandes comunidades de usuarios, y debido a su estabilidad y popularidad muchas distribuciones se han basado en esta, tal es el caso de Ubuntu. Ubuntu ha madurado al nivel de generar certificaciones de su uso.

Red Hat es una empresa que tiene proyectos tanto para la comunidad de software libre con Fedora, como soluciones empresariales con Red Hat Enterprise Linux. Red Hat se caracteriza por la seriedad de la distribución, la calidad y estabilidad de los paquetes en sus distribuciones. Incluye un grupo de diálogos gráficos que permiten configurar servicios del sistema.

Fedora se caracteriza por ser una distribución multipropósito, siendo el producto experimental para generar versiones estables de Red Hat Enterprise Linux. Fedora se puede descargar gratuitamente. Mientras que Red Hat Enterprise Linux, tiene un costo para poder adquirirlo, recibir soporte y actualizarlo.

Slackware se caracteriza por la sencillez para poder configurar el sistema a partir de la línea de comandos, a pesar que no tiene una gran cantidad de paquetes oficiales como las otras distribuciones, permite incluir software a través de repositorios no oficiales y permite integrar otras paqueterías. Slackware es un sistema liviano apto para poder reutilizar hardware antiguo. Guarda un estándar que le permite ser muy parecido a Unix, es por eso que en los manuales de aprendizaje de GNU/Linux mencionan mucho esta distribución como la más estándar.

Gentoo se caracteriza por ser una distribución avanzada, que permite optimizar desde los paquetes hasta el sistema entero. Esto permite que sea un sistema óptimo para la computadora y rápido para ejecutar aplicaciones. Una gran comunidad de usuarios mantiene sus paquetes.

Como podemos observar cada una tiene distintas características mostradas, es decir fortalezas y debilidades, no hay una distribución mejor que otra, sino que cada una es especialista en su forma de administrar el sistema (software, controladores, etc), todas parten del sistema GNU/Linux administrado y configurado de forma distinta, logrando satisfacer a distintos grupos de desarrolladores, empresas y personas individuales en el mundo.

4. MANEJO DE PAQUETES

Este capítulo explica la forma que utilizan las distribuciones Debian, RedHat, Slackware y Gentoo, para administrar los paquetes de software.

4.1. Paquetes .deb

El programa predeterminado para manejar estos paquetes en Debian es dpkg.

La forma de nombrar un paquete en Debian es:

- NombrePaquete_Version-VersionDesarrollo.deb
Ejemplo: lynx-ssl_2.8.3.1-1.deb

Según Joaquín Pascual⁴⁸, las características de los paquetes .deb tenemos que:

- **Pueden ser esenciales** (para crear el sistema base).
- **Tienen niveles de prioridad:** Requeridos, importantes, estándar, opcionales y extras.
- **Básicos, de la instalación inicial (base):** Bibliotecas, administración, desarrollo, comunicaciones, otros.

⁴⁸ (Pascual, 2001)

Para ser desarrollador oficial Debian, se deben de pasar varias etapas para comprobar que el desarrollador cumpla con los requerimientos mínimos definidos, para garantizar la calidad de sus paquetes. Debido a esto Debian ofrece estabilidad en los paquetes de software que distribuye.

Algunos de los programas utilizados para la administración de paquetes en Debian son:

- **Bajo nivel:**
 - dpkg-deb
 - Wajig
- **Individual:**
 - dpkg
 - apt-get
- **Interfaces de alto nivel:**
 - dselect
 - gnome-apt
 - Synaptic (administrador basado en la librería GTK)
 - Aptitude (puede funcionar en modo texto, Ncurses)

Otras herramientas para administración de paquetes en Debian son deborphan y GtkOrphan tools⁴⁹.

4.2. Paquetes .rpm

El programa predeterminado para administrar los paquetes RPM (*RedHat Package Manager*) es el comando *rpm*.

⁴⁹ (polishlinux.org - Debian, 2009)

La forma de nombrar un paquete en Red Hat es:

- NombrePaquete-Versión-VersiónDesarrollo.Arquitectura.rpm
Ejemplo: xsnow-1.40-5.i386.rpm

Dentro de los programas utilizados para administrar paquetes .rpm tenemos:

- **Individual**
 - rpm (programa oficial para administrar paquetes RPM)
 - yum (administrador de paquetes para distribuciones basadas en RPM)
 - urpmi (administración de paquetes de Mandriva)
- **Alto Nivel**
 - Gnorpm (interfaz de Gnome para administrar rpm)
 - Kpackage (interfaz de KDE para administrar rpm)

Según Dan Poirier⁵⁰, el RPM (*RedHat Package Manager*) es una de las herramientas de paquetes más usada en distribuciones de GNU/Linux, esta permite distribuir software ya compilado, y luego instalarlo a través de un comando, este sistema fue diseñado en base al estándar de Linux en su versión 1.0.0.

⁵⁰ (Poirier, 2001)

Es importante mencionar Yum es una herramienta que ha ganado popularidad para administrar paquetes .rpm. Su funcionalidad es similar al apt-get, permitiendo instalar y resolver las dependencias, remover o actualizar paquetes⁵¹.

Por otra parte, urpmi es programa encargado de administrar los paquetes en Mandriva, a través de la paquetería RPM.

4.3. Paquetes .tgz

El programa predeterminado para administrar paquetes en Slackware en installpkg.

La forma de nombrar un paquete en Slackware es:

- NPaquete-Ver-Arquitectura-VerDesaTresInicialesDesa.tgz

Ejemplo: apache-1.2.1-i386-2jim.tgz

En la arquitectura se puede incluir *noarch* para indicar que no esta compilado en ninguna arquitectura, esto se puede aplicar a programas que son hechos como scripts en BASH. También el paquete termina en tres iniciales que desee poner el desarrollador para identificar sus paquetes.

⁵¹ (Project, 2009)

Los paquetes de Slackware están clasificados en discos que se nombran según la función del programa: A, AP, D, E, F, K, KDE, KDEI, L, N, T, TCL, X, XAP y Y. Por ejemplo el disco que contiene todos los programas necesarios para utilizar el sistema X Window o entorno gráfico, es el disco X.

La falta de manejo de dependencias a menudo es mencionada como un debilidad de Slackware pero resuelve otro problema, el “infierno de dependencias” (*Dependency hell*). Esta característica es valorada por usuarios experimentados, ya que un sistema de manejo de paquetes (como rpm o dpkg), algunas veces pueden ocasionar problemas como dependencias redundantes, que falten dependencias, dependencias circulares y conflictos entre paquetes.

El administrador de paquetes de Slackware *pkgtool* existió antes de la paquetería rpm. Muchos piensan que un sistema de administración de paquetes por definición debe de verificar las dependencias de los paquetes. Esto no siempre es cierto. Esto no quiere decir que los paquetes de Slackware no tengan las dependencias, sino que la tarea de resolverlas se deja al administrador del sistema⁵².

Los paquetes de Slackware fueron basados en un simple archivo tgz que no contiene información sobre las dependencias, aunque en la actualidad existen paquetes no oficiales que contienen esta información.

⁵² (Alan Hicks, 2005)

Los programas utilizados por Slackware para administrar paquetes son:

- **pkgtool:** pkgtool es una herramienta que permite remover e instalar paquetes a través de una herramienta dirigida por menús. No resuelve dependencias.
- **installpkg, removepkg, upgradepkg, explodepkg, makepkg:** Estos comandos son incluidos en la distribución de Slackware y permiten administrar y crear paquetes de Slackware.
- **SlackBuild Scripts:** Son scripts para crear paquetes extras para Slackware.
- **Swaret (*SlackWARE Tool*):** Es un programa que instala paquetes de los repositorios oficiales de Slackware y resuelve sus dependencias. También permite darle mantenimiento a los paquetes.
- **Rpm2tgz:** Convierte rpm a formato tgz.
- **Slapt-get:** Reconoce las librerías así como aplicaciones cuando busca dependencias. Esta opción esta disponible solo para paquetes extra no oficiales de Slackware, ya que los paquetes oficiales no contienen esa información.
- **Sistema de ports (Emerde, pkgsrc, portpkg).**
- **Gslapt:** Interface para swaret y slapt-get.

Existen scripts, sistemas de ports y programas para línea de comandos para instalar paquetes. Slackware tiene una pequeña cantidad de paquetes oficiales a comparación de otras distribuciones. Existen paquetes no oficiales que pueden descargarse en LinuxPackages y Slacky.it en formato tgz, los cuales pueden contener meta-información de las dependencias.

Slackware tiene un sistema de archivos bastante estándar, esto permite compilar a partir del código fuente paquetes extra de los oficiales sin mayor dificultad, también se pueden conseguir paquetes pre compilados⁵³.

4.4. Portage

Según el proyecto Gentoo, el sistema Portage que es el corazón de Gentoo Linux y desempeña varias funciones clave. En si el Portage es el sistema de distribución de software de Gentoo. Para obtener el software más reciente en Gentoo se puede ejecutar un comando para realizar su actualización(`emerge -sync`). Cuando se ejecuta este comando se indica al Portage que debe actualizar el "árbol Portage" local, a través de Internet. Este árbol contiene una colección completa de guiones utilizados para crear e instalar los últimos paquetes Gentoo. Actualmente tienen más de 10,000 paquetes el árbol Portage.

El Portage permite construir e instalar paquetes. Si se desea instalar un paquete, se escribe `emerge paquete`, de esta forma el Portage automáticamente construye una versión del paquete adaptada a sus especificaciones exactas, siendo optimizada para el hardware del sistema, permitiendo también activar o desactivar opciones para compilar el software que anteriormente se solicitó al Portage.

El Portage también permite tener actualizado el software del sistema cada vez que el usuario invoque los comandos respectivos para realizar una actualización adecuada de nuevos paquetes agregados al árbol o también se permite instalar nuevas versiones del Portage actualizadas⁵⁴.

⁵³ (polishlinux.org - Slackware, 2009)

⁵⁴ (Gentoo, 2009)

4.5 Análisis entre paqueterías

La Tabla V resume las características que diferencian a cada paquetería.

Tabla V. Características de paqueterías

Característica	.deb	.rpm	.tgz	Portage
Resolución de dependencias	X	X		X
Desarrollo con control de calidad	X	X	X	X
Conexión a Internet	X			X
Optimizados	X	X	X	
Conflicto de dependencias	X	X		
Complejos de crear	X	X		
Interfaz gráfica	X	X	X	
Modo texto	X	X	X	
Compilación personalizada			X	X
Compatibilidad con Portage			X	X
Uso de scripts para generación de paquetes			X	X
Puede integrar otras paqueterías			X	
Repositorios oficiales y no oficiales	X	X	X	X
Comunidad de usuarios para soporte de paquetes	X	X	X	X
Dependencia de lenguaje de programación				X

Para la administración de paquetes de Debian, apt-get y dpkg tienen opciones avanzadas de instalación que pueden dificultar a veces la instalación normal de paquetes.

Los paquetes RPM ofrecen una versión estándar de instalación sin tanta complejidad, pero no permite realizar ciertos tipos de parametrización para realizar instalaciones. RPM necesita incluir más opciones de instalación para usuarios avanzados.

Los paquetes de Slackware deben de incluir meta información para poder competir con las demás paqueterías, pero debe de mantener la opción de instalar sin resolución de dependencias, ya que tiene la ventaja que al no resolverlas impide que exista un conflicto de versiones de paquetes, que pueden ocurrir con paquetes .deb, rpm y el Portage.

El Portage permite una optimización y configuración total a través de la compilación de paquetes. Pero necesita de cierto conocimiento para aprovechar todas las opciones que tiene el Portage para instalar programas. Esto hace que muchas veces el Portage sea complejo de utilizar. El Portage depende de la instalación de Python para funcionar.

Los paqueterías de Debian, Red Hat y Gentoo, necesitan la mayoría de veces conexión a internet, siendo muchas veces imposible para usuarios sin conexión satisfacer las dependencias que necesitan los programas para ser ejecutados.

La forma estándar del diseño de la paquetería de Slackware, permite la administración de paquetes a través de la adaptación del Portage de Gentoo, teniendo la disponibilidad de poder adoptar otras paqueterías para poder convivir en el mismo sistema. Slackware ofrece estas opciones que no tienen las otras paqueterías.

Cada distribución GNU/Linux con su sistema de paqueterías tiene tanto ventajas como desventajas, cada una ofreciendo características únicas y aptas para ciertas situaciones. Es por eso que existen muchas variantes de distribuciones GNU/Linux en el mundo del software libre.

5. HERRAMIENTA PARA CREAR LIVE CDS

En este capítulo se explica la arquitectura de la aplicación, las tecnologías involucradas para su desarrollo, un análisis de la herramienta Tortilla Burner y otras herramientas disponibles en otras distribuciones.

5.1. Elección de distribución GNU/Linux

Para poder seleccionar una distribución GNU/Linux adecuada a los objetivos de este trabajo, fue necesario realizar pruebas sobre el funcionamiento, estructura de la distribución y la administración de paquetes de software del sistema. En las siguientes secciones contienen las distintas pruebas realizadas para seleccionar la distribución GNU/Linux adecuada.

5.1.1. Pruebas paquetería y estructura del sistema

Para probar las distintas paqueterías o sistemas de administración de paquetes, se descargaron Live CDs basados en paquetes .rpm, .tgz, .deb y el Portage. Las pruebas consistieron en evaluar la facilidad de instalación de nuevos programas y la facilidad de poder crear un sistema desde cero en un directorio.

A continuación se listan los programas y Live CDs utilizados por cada paquetería:

- Pruebas de comandos apt-get, dpkg con el Live CD de Ubuntu.
- Pruebas de comandos rpm y yum con el Live CD Berry.
- Pruebas de comandos slapt-get,swaret, installpkg con el CD de instalación Slackware 12.1.
- Pruebas del Portage de Gentoo con el comando emerge con el CD de instalación de Gentoo.
- Pruebas de Portage de Gentoo con el comando emerde que es una adaptación del Portage para Slackware.

Para poder determinar la estructura de la distribución más adecuada a utilizar, se tomó en cuenta el aprendizaje de comandos y estructuras básicas relacionas con sistemas GNU/Linux. A partir de esto se realizaron distintas pruebas que consistieron en evaluar la facilidad de configuración del sistema para las distribuciones Gentoo, Slackware, Red Hat y Debian.

Entre estas pruebas podemos mencionar:

- Configuración de red del sistema.
- Configuración de entorno gráfico.
- Configuración para cargar particiones.
- Configuración de variables del sistema.
- Configuraciones de escritorio.

5.1.2. Resultados

Con respecto al manejo de las paqueterías .deb, .rpm, .tgz y el Portage, se obtuvieron los siguientes resultados:

- La paquetería .rpm no permite crear un sistema desde cero en una carpeta.
- La paquetería .deb permitió crear un sistema desde cero en una carpeta, aunque es más difícil como se realiza con un paquete .tgz.
- La paquetería .deb y el Portage, necesita conexión de internet para poder satisfacer las dependencias necesarias para instalar software, en la mayoría de casos.
- La complejidad de la clasificación de paquetes de la paquetería .deb, dificulta crear un sistema desde cero.
- La paquetería .tgz permite crear de una forma sencilla un sistema desde cero en una carpeta.
- La paquetería .tgz permite instalar y configurar el software de una forma más flexible que las otras paqueterías.
- El funcionamiento del Portage es complejo, con la ventaja que permite optimizar el sistema, permitiendo la ejecución de programas de forma más rápida.

Los resultados obtenidos para las pruebas de estructura de las distribuciones Gentoo, Slackware, Red Hat y Debian, fueron los siguientes:

- Slackware fue más flexible con respecto a la configuración y creación de un sistema desde cero, esto permite que Slackware pueda adoptar el sistema de administración de paquetes de Debian, Red Hat y Gentoo.
- Gentoo mostró ser un sistema complejo que necesita de conocimiento previo del sistema para poder configurarlo y optimizarlo.
- Debian mostró ser menos complejo que Gentoo para la configuración y administración de servicios y paquetes del sistema.
- Red Hat no fue tan flexible para la configuración manual del sistema como Slackware, Gentoo y Debian.

Estos resultados nos muestran que Slackware, brinda una mayor flexibilidad para configuración y administración de paquetes del sistema, lo cual permite crear una distribución GNU/Linux desde cero de forma sencilla. Es un sistema estándar que permite a los usuarios aprender a utilizar un sistema GNU/Linux.

5.2. Definición de componentes

En esta sección se explican los distintos componentes de software que integran la aplicación y que permiten que se puedan generar Live CD personalizados.

5.2.1. Kernel 2.6

El kernel utilizado por la herramienta es la versión 2.6.27.7. El kernel 2.6 esta asociado a proyectos de reconociendo y configuración de hardware como: Hotplug, Automount, etc.

El kernel de la herramienta fue configurado basándose en la configuración del kernel 2.6.24.5 de Slackware 12.1, sin soporte SMP (*Symmetric Multi-Processing*). Soporta aufs, squashfs y es modular para poder utilizar Hotplug y ser lo más genérico posible, para soportar la mayor cantidad de hardware.

5.2.2. Módulos LZM

En un estudio sobre el rendimiento de sistemas de archivos comprimidos, se utilizaron dos Live CD, un Live CD de Ubuntu para probar los rendimientos de Zisofs, Cloop, Squashfs 2.0 y Squashfs 2.1, y para probar el rendimiento de CRAMFS en comparación de Squashfs se utilizó el Live CD Damn Small Linux.

Los resultados sobre el rendimiento de estos sistemas de archivos de compresión, reveló que Squashfs en las 2 versiones anteriormente citadas, tenían un mejor rendimiento en ver los archivos de un directorio, entradas y salidas secuenciales, aleatoriedad de entrada y salida, también mostró una mejor compresión de archivos tanto en un CD como en un disco duro⁵⁵.

Esto muestra que squashfs es una buena opción para ser utilizada como un sistema de archivos compresado.

Tortilla Burner comprime el sistema de archivos en módulos con extensión .lzm, a través de la compresión de carpetas utilizando Squashfs. Junto con Aufs, estos módulos que son cargados a memoria pueden ser modificados, creando la funcionalidad completa de un Live CD basado en otras tecnologías.

Los Live CDs basados en Slax utilizan esta compresión. Actualmente la mayoría de Live CDs utilizan las tecnologías Squashfs y Aufs. El utilizar estas tecnologías permite tener funciones similares a los Live CDs generados en la actualidad.

5.2.3. Scripts en BASH

La programación de scripts con BASH permite tener una interacción sencilla con la línea de comandos y combinando los scripts con Ncurses, proporciona una interfaz gráfica amigable al usuario basándonos en pantallas de modo texto.

⁵⁵ (filesystem performance test , 2009)

El uso de BASH en el desarrollo de este trabajo fue fundamental, ya que los scripts que componen la herramienta desarrollada, utilizan varios programas que son llamados desde la línea de comandos.

5.2.4. Scripts de Slackware

Los scripts modificados de la instalación de Slackware, proporcionan una gran cantidad de procesos necesarios para poder crear un sistema temporal para crear un Live CD.

Para poder obtener estos scripts, se descomprimió el archivo initrd.gz (ubicado en el disco 1 de Slackware) que se carga en la instalación Slackware. Este archivo, contiene un sistema mínimo de arranque que permite realizar la instalación de la distribución de Slackware en el disco duro, para el caso de la herramienta del trabajo el sistema se instala en una carpeta.

5.2.5. Live scripts

Los Live Scripts son scripts utilizados para crear Live CDs basados en Slax. Estos scripts fueron una parte fundamental para la creación de la herramienta, ya que integran el kernel y el sistema de archivos creado, los cuales sirven para generar el sistema vivo o Live CD. Estos scripts fueron modificados para configurar el nombre y memoria utilizados por el Live CD a crearse. Los Live scripts pueden descargarse en <http://www.linux-live.org>. La versión utilizada de estos scripts en la herramienta fue la versión 6.2.9.

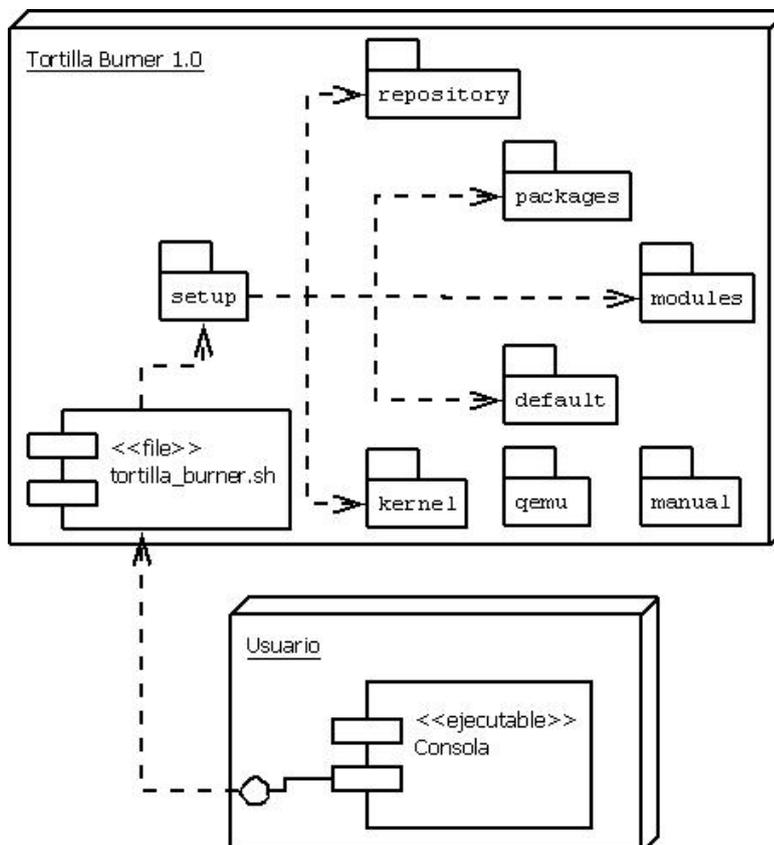
5.3. Arquitectura

A continuación se describe la forma en que fue diseñada y creada la aplicación, el proceso y la interacción entre los componentes del programa que intervienen en el proceso de la creación de un Live CD.

5.3.1. Flujo de scripts de la herramienta

En la Figura 2 se muestran las dependencias entre scripts y archivos según su uso organizados en carpetas.

Figura 2. Diagrama de componentes de la herramienta

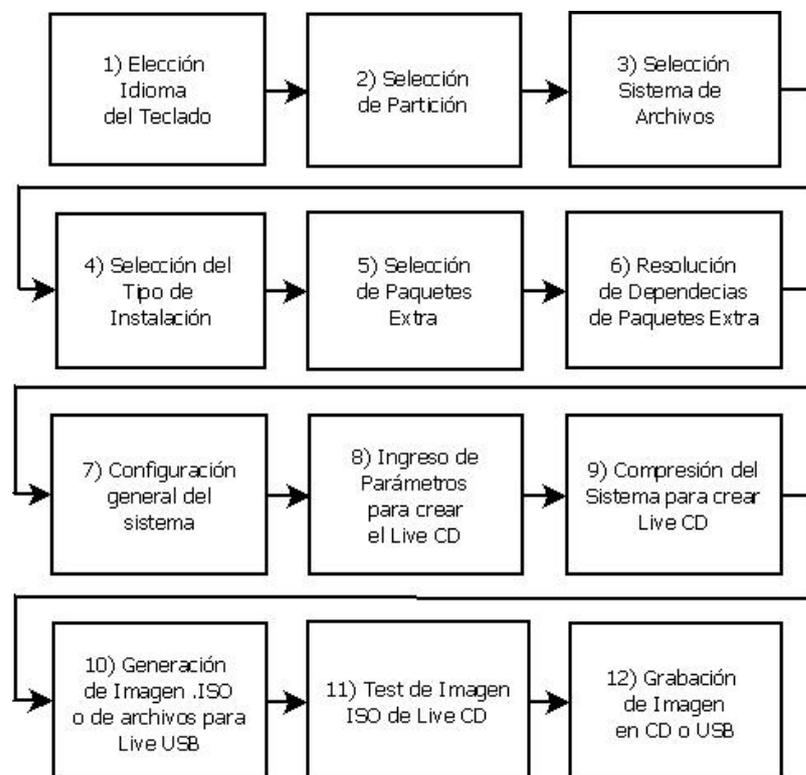


Como podemos observar el usuario a partir de una consola ejecuta el script de arranque de la aplicación *tortilla_burner.sh*. Cada archivo o script esta ubicado en una carpeta según su naturaleza. El script *tortilla_burner.sh* utiliza los diversos archivos ubicados en estas carpetas para ejecutar los procesos descritos en la sección anterior y así poder generar la imagen de un Live CD. Las carpeta *qemu* y *manual* no son utilizadas por el script *tortilla_burner.sh* pero contiene el programa *qemu* que sirve para emular la ejecución de la imagen del Live CD, y *manual* contiene la documentación para poder utilizar Tortilla Burner.

5.3.2. Procesos para crear un Live CD

En la Figura 3 se describen los pasos necesarios para poder crear un Live CD.

Figura 3. Diagrama de los procesos para crear un Live CD



Se inicia la herramienta ejecutando el script *live_creator.sh*, que verifica el usuario que actualmente esta tratando de ejecutar la aplicación sea root. Este script se encarga de establecer todas las variables de entorno necesarias para poder invocar otros scripts especializados según su función.

Luego el script *setup* es ejecutado lanzando el menú de la aplicación. El proceso comienza al seleccionar "KEYMAP". Internamente el script *setup* ejecuta el script *SeTkeymap*, el cual se encarga de la configuración del idioma del teclado (proceso no.1) del Live CD. Posteriormente para la selección de una partición (proceso no.2) de trabajo, *setup* ejecuta el script *SeTpartitions* que formatea la partición (proceso no.3) escogida con el tipo (ext2, ext3, reiserfs, etc) seleccionado, luego ejecuta *SetDOS* para agregar cualquier partición de Windows al archivo */etc/fstab*.

Para la selección del tipo de instalación (proceso no.4), *setup* ejecuta el script *SeTLiveDisk*, el cual selecciona el tipo de disco a descomprimir para poder crear el sistema GNU/Linux que se convertirá en Live CD. Este script ejecuta *INSSystem*, el cual es encargado de descomprimir los discos pre configurados (módulos .lzm en la carpeta *modules*), luego ejecuta *INSWaret* encargado de generar el menú con los paquetes restantes, sin incluir los previamente instalados en los discos pre configurados con *INSSystem*. Estos scripts permiten la selección de tipo de sistema a instalar y la selección de paquetes extra (proceso no.5) a instalar.

Para poder crear el menú dinámico se monta el repositorio creado con los *swaret-tools* (incluido en Tortilla Burner), posteriormente el archivo de configuración de *swaret.conf* ubicado en */etc* tiene los parámetros necesarios para poder acceder a este repositorio desde la carpeta */mnt/repository*.

El menú se genera obteniendo la lista de paquetes no instalados con el comando *swaret --list*, clasificándolos de acuerdo a su tipo de disco (A, AP, X, KDE, N, etc) generando un archivo por cada disco para generar el menú de cada disco, estos archivos son generados en la carpeta */tmp/scripts_live*, posteriormente se instalan los paquetes seleccionados con el comando:

```
#chroot /mnt/MiLinux /usr/sbin/swaret --install pkg_seleccionado.tgz
```

Swaret permite resolver las dependencias (proceso no.6) de cada paquete extra seleccionado. Después se detiene el proceso para que el usuario pueda realizar configuraciones avanzadas y manuales que necesite o bien pueda seleccionar la opción “LIVE” del Menú de la aplicación.

Al seleccionar el menú “LIVE” se ejecuta en *setup* el script *SeTconfig* que ejecuta todos los scripts de configuración de Slackware (proceso no.7) instalados dependiendo de los paquetes seleccionados, posteriormente se ejecuta el script *SeTLiveCD*, el cual obtiene todos los parámetros necesarios para configurar el archivo *.config* (proceso no.8) que contiene la ubicación de sistema para crear como Live CD, el nombre del Live CD y la cantidad de memoria RAM utilizada para la carga de módulos del Live CD.

Después de la configuración de los Live Scripts para generar el Live CD, se empiezan a comprimir las carpetas (proceso no.9) del sistema creado para generar los módulos necesarios para cargar el sistema en el Live CD a crearse, al terminar este proceso, se ejecuta el archivo *make_iso.sh* de los Live Scripts para generar la imagen *.iso* (proceso no.10) del nuevo Live CD, o bien se permite copiar el contenido creado en una memoria USB para una Live USB. Antes de grabar o copiar el contenido del nuevo sistema vivo en un Disco o en una memoria USB, se puede probar la imagen (proceso no.11) con el programa *qemu* (proceso no.12) del CD con el programa que seleccione (ej.: K3B).

5.4. Comparación con otras herramientas

En esta sección se compara Tortilla Burner con las siguientes herramientas: Reconstructor y MySlax Creator. La Tabla VI, resume las características de cada una de las herramientas.

Tabla VI. Características de herramientas para crear Live CDs

Característica	Herramienta		
	Reconstructor	MySlax Creator	Tortilla Burner
Remastering de CD	X	X	X
Manejo de módulos(LZM)		X	X
Soporte de varias arquitecturas Kernel	X		
Manejo carpeta de trabajo	X	X	X
Manejo de repositorios	X		
Manejo de repositorios offline			X
Manejo de temas de carga	X		
Manejo de temas de escritorio	X	X	
Graba imagen .iso en CD	X	X	X
Navegación opciones	X	X	X
Llamada a terminal	X	X	X
Interfaz X11	X	X	
Interfaz consola			X
Ejecución en Windows		X	
Envío de parámetros al cargar Live CD		X	
Test del CD		X	X
Instalación portable			X
Instalación en memoria USB		X	X
Ayuda de la aplicación en menú principal			X

Reconstructor crea Live CDs basados en Ubuntu (distribución popular basada en Debian). Fue desarrollada su interfaz gráfica en lenguaje Python, soporta varias arquitecturas de procesador y permite personalizar los temas de los escritorios de Gnome y KDE.

MySlax Creator crea Live CDs basados en Slax. Diseñada para el sistema operativo Windows. Los Live CDs que permite crear, se basan en la imagen de un Live CD basado en Slax, permitiendo agregar nuevos paquetes. Configurar los parámetros iniciales de carga del Live CD, las imágenes de fondo de KDE y XFCE y el Xorg. Los sistemas generados pueden copiarse en una memoria USB o CD, no utiliza repositorios para crear un Live CD y permite probarlos con VMWare.

Tortilla Burner crea Live CDs basados en Slackware con el sistema modular de Slax para arquitecturas de procesadores x86. Desarrollado con interfaz en modo texto con BASH y Ncurses, permite a usuarios avanzados optimizar el sistema a crear, no necesita conexión a Internet ya que incluye un repositorio en la herramienta, que puede ser modificado para agregar nuevos programas, maneja particiones para guardar el sistema creado y se puede probar la imagen del sistema generado con qemu.

En comparación, Tortilla Burner ofrece una interfaz más liviana y manejo de particiones para desarrollar Live CDs. Su diseño permite que la herramienta sea más sencilla de modificar. Esta herramienta incluye muchas de las funciones comunes en una herramienta para crear Live CDs, contiene características de las herramientas Reconstructor y MySlax Creator. Tortilla Burner puede ser modificado para crear Live CDs basados en otras distribuciones.

5.5. Pruebas de la aplicación

Para evaluar la aplicación se utilizaron los siguientes atributos de calidad:

- **Funcionalidad:** habilidad del software para realizar el trabajo deseado
- **Fiabilidad:** habilidad del software para mantenerse funcionando
- **Eficiencia:** habilidad del software para responder a una petición de usuario con la velocidad apropiada
- **Usabilidad:** habilidad del software para satisfacer al usuario.
- **Mantenibilidad:** habilidad del software para poder realizar cambios en él rápidamente y con una adecuada proporción cambio/costo.
- **Portabilidad:** habilidad del software para correr en diferentes entornos informáticos.

A través de distintas estrategias de pruebas se evaluaron estos aspectos a través de pruebas dentro del desarrollo, pruebas de usuarios y a través de una encuesta con preguntas acerca de la funcionalidad de la aplicación.

5.6. Pruebas unitarias y de integración

Las pruebas unitarias realizadas para evaluar el funcionamiento de cada script por separado, consistieron en probar las distintas funciones incluidas en cada uno de los scripts, depurando posibles errores. Una de las primeras pruebas de integración, que nos permiten ver el funcionamiento en conjunto con los distintos scripts, consistió en la llamada de las distintas opciones dentro del menú de la aplicación.

Estas opciones llaman a los scripts necesarios para iniciar un proceso dentro de la creación de un Live CD. Debido a que el programa esta desarrollado como software libre y desarrollado en lenguaje BASH y Ncurses, resulta muy sencillo poder realizar modificaciones a la herramienta, esto permite que su mantenibilidad sea fácil. Estas características permiten que sea una aplicación con buena portabilidad, ya que la mayoría de distribuciones GNU/Linux tienen BASH y Ncurses instalado por defecto en el sistema de archivos.

5.7. Pruebas del sistema y aceptación

Con respecto a las pruebas del sistema, las cuales nos permiten evaluar el funcionamiento de la aplicación en un ambiente de desarrollo, se evaluó la portabilidad. Instalando la aplicación en una Laptop con procesador Celeron M, y 512 MB de RAM, no mostró dificultad para instalar el programa. Dentro de algunas pruebas de creación de Live CD tenemos que se creó un Live CD, en 30 minutos, seleccionando las opciones mínimas para creación de un Live CD.

También se realizaron pruebas con usuarios, permitiendo realizar las pruebas *alpha* o pruebas en ambiente de desarrollo, encontrando problemas con el hardware como la necesidad de la interfaz de red activa, problemas en la generación del initrd del Live CD, estos fueron corregidos, cumpliendo con un ciclo de pruebas y mejoras dentro de las pruebas *alpha*.

Estos usuarios evaluaron la funcionalidad de la aplicación a través de la experimentación y la demostración de la herramienta para crear Live CDs en una actividad de software libre. Luego de hacer las correcciones necesarias al probar la aplicación en un ambiente de desarrollo, se realizaron nuevas pruebas con usuarios. Estas actividades permitieron recolectar datos a través de una encuesta que permitió evaluar la funcionalidad, eficiencia y usabilidad de la aplicación.

5.7.1. Resultados

Para poder evaluar los resultados de los atributos de funcionalidad, eficiencia y usabilidad, se realizó una ponderación de las distintas preguntas de la encuesta como: inaceptable, mínimo aceptable, satisfactorio y como máximo puntaje muy satisfactorio. Los resultados se muestran a continuación:

- **Funcionalidad y la eficiencia:** Dentro de las distintas preguntas realizadas sobre el uso de la aplicación, facilidad de la interfaz de modo texto, personalizaciones permitidas, etc. En promedio se mostró que la funcionalidad fue satisfactoria, abarcando un 80% de resultados positivos.
- **Usabilidad:** Un 66% mostró interés en que se realizaran mejoras de interfaz, es decir incluir una interfaz gráfica, como también incluir más personalizaciones para crear el Live CD, extra de las incluidas en la aplicación. Un 88% opinó que es muy satisfactoria la interfaz de modo texto y les pareció sencillo de utilizar.

Dentro de las preguntas de mejoras en la encuesta se pudo observar que el 70% de los usuarios les interesa la opción de poder crear Live USB es decir sistemas Live que se ejecutan y arrancan desde una USB. Por otra parte el 50% les interesa que la herramienta incluya otras personalizaciones de las ya incluidas; estas fueron las dos mejoras con mayor porcentaje.

También se observó que las distribuciones más utilizadas son Ubuntu con un 90% de preferencia y el 50% Debian.

Dentro de los usos que le darían los usuarios a la herramienta se encontró que un 70% está interesado en crear Live CDs de rescate y un 60% solo en crear un Live CD con fines de demostración.

CONCLUSIONES

1. Es posible realizar una herramienta para crear Live CDs personalizados.
2. La tendencia de los usuarios es el uso de distribuciones basadas en Debian. Se mostró que un 90% usa Ubuntu, esto influye para que versiones siguientes de la herramienta Tortilla Burner de soporte para crear Live CDs se basen en esta distribución.
3. La herramienta debe de incluir en nuevas versiones una interfaz gráfica basada en gráficos X, ya que actualmente la mayoría de herramientas soportan esta interfaz y las encuestas realizadas muestran que estos tipos de interfaces facilitan al usuario el uso y aprendizaje de herramientas.
4. El repositorio de la herramienta debe de incluir paquetes y personalizaciones que permitan al usuario crear demostraciones y discos de rescate, ya que a través de la evaluación de la aplicación se mostró que la preferencia de los usuarios abarca estas dos áreas.
5. Una de las debilidades encontradas fueron el manejo de proyectos de creación de Live CDs en todas las herramientas, esta es una opción que carecen las herramientas actuales para crear Live CDs y a la vez representa una mejora futura para la herramienta Tortilla Burner.

RECOMENDACIONES

1. Modificar la aplicación para que pueda ser utilizada como un estándar para crear Live CDs basada en cualquier distribución, esto incluye las distribuciones más populares basadas en los paquetes .deb y .rpm.
2. Publicar el proyecto en sitios de desarrollo de software para GNU/Linux como sourceforge.net, freashmeat.net, para promover su difusión, uso y poder encontrar una fuente de bajo costo para detección de errores en la aplicación.
3. Dar a conocer el proyecto en congresos de software y actividades relacionadas a comunidades de usuarios de GNU/Linux para poder encontrar personas que quieran integrarse al equipo de desarrollo de mejoras de la aplicación desarrollada.
4. Registrar la propiedad intelectual de la aplicación para que nadie robe la idea, ya que es la primera herramienta guatemalteca creada con respecto a la creación de Live CDs.
5. Dar a conocer la solución creada a organizaciones como CONCYT para obtener apoyo financiero, continuar el desarrollo y mejora del proyecto.

BIBLIOGRAFÍA

1. *About MySLAX Creator.* (abril de 2005). <http://myslax.bonsonno.org/about.php>, (abril 2009).
2. Alan Hicks, C. L. (2005). *Slackware Package Management.* <http://www.slackbook.org/html/book.html#PACKAGE-MANAGEMENT>, (junio 2009).
3. Barlow, D. (01 de marzo de 2005). *Building Your Own Live CD.* <http://www.linuxjournal.com/article/7246>, (abril 2009).
4. Byfield, B. (17 de enero de 2007). *Linux Standard Base plans cross-format package API.* <http://www.linux.com/archive/feature/59502>, (julio 2009).
5. Croissier, J. J. (28 de octubre de 2002). *Cómo hacer un Live-CD.* Explicación de cómo hacer un CD a partir de Knoppix. http://linux.pricoinsa.es/Como_hacer_un_Live-CD.html, (abril 2009).
6. *Debian.* (2009). <http://www.debian.org>, (agosto 2009).
7. DiMaggio, L. (2009). *Understanding your (Red Hat Enterprise Linux) daemons.* <http://magazine.redhat.com/2007/03/09/understanding-your-red-hat-enterprise-linux-daemons>, (abril 2009).
8. Fedora, R. H. (13 de enero de 2009). *Anaconda.* <http://fedoraproject.org/wiki/Anaconda>, (enero 2009).
9. *filesystem performance test .* (2009), GENERAL INFORMATION ON PERFORMANCE TESTS. <http://kerneltrap.org/files/PERFORMANCE.README.txt>, (abril 2009).

10. Foundation(FSF), F. S. (26 de enero de 2008). *Historia del Proyecto GNU*. <http://www.gnu.org/gnu/gnu-history.es.html>, (abril 2009).
11. Foundation, G. (2001). *Catalyst*. <http://www.gentoo.org/proj/en/releng/catalyst>, (abril 2009).
12. Foundation, L. (2009). *Linux Standard Base (LSB)*. <http://www.linuxfoundation.org/collaborate/workgroups/lsb>, (julio 2009).
13. Frank Bodammer, S. D. (2009). *SUSE LINUX 9.1 – Administration Guide*. <http://www.novell.com/documentation/suse91/suselinux-adminguide/html/ch07s02.html>, (abril 2009).
14. Gentoo. (2009). *Gentoo Linux - Acerca de Gentoo*. <http://www.gentoo.org/main/es/about.xml>, (junio 2009).
15. Johnson, M. K. (22 de septiembre de 2003). *Fedora Project: Announcing New Direction*. Obtenido de <http://www.redhat.com/archives/rhl-list/2003-September/msg00064.html>, (julio 2009).
16. Jones, M. T. (31 de julio de 2003). *Linux initial RAM disk (initrd) overview*. <http://www.ibm.com/developerworks/linux/library/l-initrd.html>, (abril 2009).
17. Junjiro Okajima, P. L. (2006). *Official Squash LZMA*. <http://www.squashfs-lzma.org>, (abril 2009).
18. Jure Cuhalev, S. B. (2009). *Intellibuild*. <http://ibuild.livecd.net>, (abril 2009).
19. kamiyeye. (2009). *RedHatEnt*. <http://fedoraproject.es/redhatent>, (junio 2009).
20. Knopper, K. (2009). *Knoppix Linux Live CD*. <http://www.knoppix.org>, (abril 2009).

21. Laurence Bonney, W. M. (24 de agosto de 2005). *Boot loader showdown: Getting to know LILO and GRUB*. <http://www.ibm.com/developerworks/library/l-bootload.html>, (abril 2009).
22. *Linux Hotplugging*. (2009). <http://linux-hotplug.sourceforge.net>, (abril 2009).
23. *Linux/Unix Command kudzu*. (2009). http://linux.about.com/od/commands/l/blcmdl8_kudzu.htm, (abril 2009).
24. Matejcek, T. (2009). *Linux Live scripts*. <http://www.linux-live.org>, (abril 2009).
25. Matejcek, T. (2002). *Slax - su sistema operativo de bolsillo*. <http://www.slax.org/?lang=es>, (abril 2009).
26. Matejcek, T. (2007). *UnionFS for Linux*. Obtenido de <http://www.unionfs.org>, (abril 2009).
27. Nguyen, B. (30 de julio de 2004). *The Root Directory*. <http://tldp.org/LDP/Linux-Filesystem-Hierarchy/html/the-root-directory.html>, (abril 2009).
28. Nieto, D. F. (2005). *Manual de Creación de un LiveCD de FreeBSD*. http://www.opensob.org/docs/manual_creacion_livecd/html/manual-creacion-livecd.html, (abril 2009).
29. Okajima, J. (mayo de 2007). *Sitio Oficial de Aufs*. <http://aufs.sourceforge.net>, (abril 2009).
30. Okuji, Y. K. (2006). *GNU GRUB*. <http://www.gnu.org/software/grub>, (abril 2009).

31. Pascual, J. S. (16 de octubre de 2001). <http://jungla.dit.upm.es/~joaquin/las/programa/admin/paqueteria.pdf>, (abril 2009).
32. Poirier, D. (01 de noviembre de 2001). *Usando RPM en Red Hat Linux*. <http://www.plazalinux.com/modules.php?name=News&file=article&sid=546>, (junio 2009).
33. *polishlinux.org - Debian*. (2009). <http://polishlinux.org/linux/debian>, (junio 2009).
34. *polishlinux.org - Fedora*. (2009). <http://polishlinux.org/linux/fedora>, (junio 2009).
35. *polishlinux.org - Gentoo*. (2009). <http://polishlinux.org/linux/gentoo>, (junio 2009).
36. *polishlinux.org - Slackware*. (2009). <http://polishlinux.org/linux/slackware>, (junio 2009).
37. *Porting to the LSB*. (2009). <http://ldn.linuxfoundation.org/article/porting-lsb>, (junio 2009).
38. Project, F. (2009). *Tools/yum - Fedora Project*. <http://fedoraproject.org/wiki/Tools/yum>, (junio 2009).
39. Rankin, K. (13 de septiembre de 2004). *Why Linux Standard Base 2.0 won't matter*. http://www.oreillynet.com/sysadmin/blog/2004/09/why_linux_standard_base_20_won.html, (julio 2009).
40. Rankin, Kyle. (22 de noviembre de 2004). *Remaster Knoppix without Remastering*. http://www.oreillynet.com/sysadmin/blog/2004/11/remaster_knoppix_without_remas.html, (abril 2009).

41. *Reconstructor*. (2009). <http://reconstructor.aperantis.com>, (junio 2009).
42. Roblimo. (17 de marzo de 2000). *Replies from Slackware Founder Patrick Volkerding*. <http://slashdot.org/interviews/00/03/17/1120205.shtml>, (julio 2009).
43. Slackware. (2009). www.slackware.com, (junio 2009).
44. Stallman, R. (03 de julio de 2003). *Linux y el Proyecto GNU*. <http://www.gnu.org/gnu/linux-and-gnu.es.html>, (abril 2009).
45. Stewart, W. (2007). *Linux History*. LivingInternet.com
http://www.livinginternet.com/i/iw_unix_gnulinux.htm, (abril 2009).
46. *The Syslinux Project*. (09 de abril de 2009).
http://syslinux.zytor.com/wiki/index.php/The_Syslinux_Project, (abril 2009).
47. *Unionfs: A Stackable Unification File System*. (09 de enero de 2007).
<http://www.filesystems.org/project-unionfs.html>, (abril 2009).
48. Walt, D. v. (Noviembre de 2005). *LiveCD_Druid Manual V02-00*.
<http://forgeftp.novell.com/livecd-druid/help/index.html>, (abril 2009).

APÉNDICES

MANUAL TÉCNICO

REQUERIMIENTOS

La aplicación puede ser ejecutada en sistemas Slackware GNU/Linux con procesadores mayores o iguales a Pentium 4 con más de 256 MB de memoria RAM, con un mínimo de 6 a 8 GB libres de disco duro para poder crear sistemas en los cuales se basarán los Live CD e imágenes .ISO de los Live CD Creados.

VARIABLES DE ENTORNO

Estas variables son creadas en el script *live_creator.sh*.

A continuación se listan las variables utilizadas y su uso en la aplicación:

TMP(/tmp/scripts_live): Contiene el directorio donde se guardan archivos temporales para manejar el proceso de creación de un Live CD.

SETUP_DIR(directorio_actual/setup): Directorio que contiene todos los scripts para crear un Live CD.

DEFAULT_DIR(directorio_actual/default): Directorio que guarda configuración o modificaciones de programas default.

KERNEL_DIR(directorio_actual/kernel): Directorio del kernel.

VERSION_LS(6.2.9): Versión de Live Scripts Utilizada.

SCRIPT_LIVE(linux-live-\$VERSION): Directorio donde se encuentra la versión de Live SCRIPTS.

MODULES_DIR(directorio_actual/modules): Directorio de módulos comprimidos.

MANUAL_DIR(directorio_actual/manual): Directorio de Manuales.

PACKAGES_DIR(directorio_actual/packages): Directorio de los paquetes.

DESCRIPCIÓN DE ARCHIVOS UTILIZADOS

CARPETA default: En la carpeta default se guardan todos los archivos relacionados con configuraciones por defecto que utilizan ciertos comandos en la herramienta.

A continuación se describe el uso de cada archivo utilizado en la aplicación:

build: Es una versión personalizada del archivo del mismo nombre, distribuido en los Live Scripts de la versión 6.2.9, esta automatiza la compresión de los módulos a crearse sin necesidad de preguntar o confirmar datos sobre el kernel, nombre, etc., necesarios para generar la imagen ISO del Live CD.

config_original: Es una versión personalizada del archivo .config incluido en los Live Scripts de la versión 6.2.9, que incluye la información por defecto para poder crear un Live CD.

make_iso.sh: Es una versión personalizada del script con el mismo nombre incluido en los Live Scripts de la versión 6.2.9 ya que permite generar la imagen ISO sin preguntar ninguna información al usuario, es decir crear la imagen de forma automática.

swaret: Este archivo es el script modificado de la herramienta swaret que instala paquetes, ya que esta versión personal de swaret permite instalar paquetes sin preguntar si desea o no instalar el paquete dependiente para satisfacer las dependencias de un paquete que se este instalando.

swaret.conf: Este archivo contiene la configuración necesaria para que el comando swaret de Linux permita utilizar el repositorio montado en la ubicación */mnt/repository* y de esta forma buscar los paquetes a instalarse en esta carpeta.

CARPETA kernel: Esta carpeta contiene las fuentes para crear el kernel dentro de la carpeta source y dentro de binary el kernel ya compilado.

CARPETA manual: Esta carpeta contiene la documentación técnica y de usuario, como el marco teórico que respalda el desarrollo de esta herramienta.

CARPETA modules: Dentro de esta carpeta se encuentran los módulos necesarios para armar un sistema mínimo con o sin entorno gráfico. Esta contiene 3 carpetas y cada una contiene un archivo .lzm y un script.

El archivo .lzm permite crear una parte de un sistema mínimo mientras que el archivo .sh permite crear el módulo .lzm, de esta forma al modificar estos archivos se pueden agregar nuevos paquetes para incluir en cada módulo.

Existen 3 carpetas *core_base*, que contiene el módulo para crear una base mínima de archivos para crear un sistema GNU/Linux, *xorg* contiene el módulo para integrar el sistema de gráficos X al sistema base y *desktop_kde* contiene los archivos mínimos para poder incluir kde al sistema de gráficos X. El sufijo para el nombre de cada archivo “pe” significa *personal edition*, para diferenciar este archivo del cual se basó originalmente del módulo del sistema base de Slax. Están numerados del 1 al 3 en su orden de instalación y dependencia, es decir el 3 necesita instalado el módulo 1 y 2.

CARPETA packages: Esta carpeta contiene software utilizado internamente por la herramienta.

A continuación se listan los paquetes utilizados internamente por la herramienta:

keymaps.tar.gz: Este archivo está ubicado en la carpeta kb dentro de packages y contiene los tipos de letra para distintos tipos de teclado.

linux-2.6.27.7-i486-1.tgz: Este paquete contiene el kernel 2.6.27.7 pre compilado y adaptado para poder ser usado en los Live CDs generados.

linux-live-6.2.9.tar.gz: Este archivo contiene los Live Scripts que son utilizados para poder crear todo el proceso de compresión para poder generar los módulos que son cargados por el Live CD a crearse y de generar la imagen ISO conteniendo estos módulos y otros archivos necesarios para poder crear un Live CD.

swaret-1.6.3-noarch-2.tgz: Este archivo contiene la herramienta para instalar software del repositorio incluido en esta herramienta.

swaret-tools-0.96.tar.gz: Este es un paquete extra para ser usado por desarrolladores, ya que este paquete permite crear un repositorio de paquetes.

CARPETA qemu: Esta carpeta contiene el paquete qemu para Slackware compilado con un SlackBuild, este programa se incluye para poder probar una imagen .iso sin necesidad de tenerlo que grabar en un CD.

CARPETA repository: Esta carpeta contiene el repositorio de paquetes que puede ser reemplazado por los desarrolladores, para crear este repositorio utilizó el script para crear repositorios incluido en los swaret-tools y luego fue empaquetado en una imagen ISO con el comando mkisofs y con el parámetro --J para mantener sin alterar los nombres de los archivos.

CARPETA setup: Esta carpeta es una de las más importantes para el proyecto de Tortilla Burner, contiene todos los scripts que preparan el sistema para instalar paquetes y crear el Sistema GNU/Linux en el cual se basará el Live CD, permite configurar el sistema entre otras opciones. A continuación se lista la explicación de los scripts más importante y aportados ya que este conjunto de scripts esta basado en el instalador oficial de Slackware 12.1.

DEADJOE: Sirve para registrar la bitácora de los pasos realizados en la herramienta.

INSNFS: Este archivo sirve para usar un recurso compartido NFS el cual contiene los discos que contienen los paquetes de Slackware, este se usa como fuente para instalar paquetes.

INSdir: Este archivo indica una carpeta para instalar el sistema dentro del sistema de archivos actual.

SeTDOS: Este archivo permite detectar y agregar particiones de Windows al archivo `/etc/fstab` del sistema a crearse.

SeTPKG: Este archivo crea un menú dinámico de los discos(A, AP, X, etc) que pueden instalarse en Slackware, si existe el disco se incluye en el menú.

SeTfull: Este archivo permite revisar si el disco esta lleno.

SeTmedia: Este archivo permite seleccionar el origen de los discos de Slackware para una instalación normal con el instalador de Slackware.

SeTpasswd: Permite configurar la contraseña que utilizará el super usuario (root) del sistema.

migrate.sh: Este archivo realiza configuración de la carpeta temporal a usar en la partición que se montará para crear el nuevo sistema.

Setpkg: Este script es utilizado si no se encuentra el archivo por defecto para selección de discos a instalar.

unmigrate.sh: Este desmonta y elimina la carpeta temporal para configuración que utiliza el script `migrate.sh`.

FDhelp: Es el archivo utilizado para mostrar el contenido de la ayuda de la herramienta.

INSSystem: Este script instala los distintos discos pre configurados en la partición seleccionada por la herramienta.

INShd: Permite instalar teniendo como fuente de los discos de Slackware una carpeta en el disco duro.

SeTLiveCD: Este script permite ingresar los parámetros necesarios para poder comenzar a generar la imagen `.iso` del sistema creado.

SeTconfig: Este script permite ejecutar todos los scripts necesarios para poder realizar la configuración del sistema temporal en el cual se basará la herramienta para crear un Live CD.

SeTkernel: Este archivo se encarga de poder instalar un kernel opcional de los distribuidos originalmente por Slackware.

SeTnet: Permite configurar la red para poder realizar operaciones, como la conexión a un servidor NFS para instalar paquetes en la partición seleccionada para crear el sistema en el que se basará el Live CD a crear.

SeTswap: Este archivo permite seleccionar la partición que utilizará el nuevo sistema a crearse.

nopartHELP: Es un archivo que contiene otra información de ayuda.

setup: Este archivo es el más importante del proyecto ya que por medio de él se seleccionan las opciones a ejecutarse, contiene el menú principal de la aplicación y permite ejecutar todo el proceso de la creación de un Live CD.

INSURL: Permite configurar una instalación de Slackware con fuente de paquetes de un servidor FTP o HTTP.

PROMPThelp: Es un archivo que contiene información de ayuda acerca de los tipos de instalación completos.

SeTfdHELP: Permite mostrar los archivos de ayuda de la aplicación seleccionando HELP en el menú principal de la aplicación.

SeTkeymap: Permite configurar el idioma del teclado.

SeTnopart: Permite mostrar ayuda para realizar el particionamiento y el sistema de archivos correcto para instalar un sistema GNU/Linux.

fixdate: Este script permite capturar la fecha actual de la computadora y usarla para configurar el sistema GNU/Linux que crea la herramienta.

pkgtool: Este script es una copia del configurador del sistema de Slackware.

INSCD: Este script permite detectar e instalar los paquetes de Slackware a partir de un CD como fuente de paquetes de instalación.

INSWSwaret : Este archivo permite crear el menú dinámico de paquetes extras a incluirse, resolviendo las dependencias que necesite cada paquete seleccionado para posteriormente convertir el sistema creado en un Live CD.

SeTLiveDisk: Este script permite seleccionar el tipo de disco pre configurado, para que a partir de este se seleccionen paquetes extra y de esta forma se cree un sistema GNU/Linux.

SeTmaketag: Archivo que permite usar TAGFiles dentro de la instalación de los paquetes de Slackware.

SeTpartitions: Este script permite formatear una partición con el tipo seleccionado de sistema de archivos.

installpkg: Una copia del comando para instalar paquetes .tgz de slackware.

removepkg: Una copia del comando para desinstalar paquetes .tgz de slackware.

slackinstall: Este script permite instalar cada uno de los discos de una instalación completa de slackware.

MANUAL DE USUARIO

Para ejecutar Tortilla Burner se debe de preparar el sistema para poder crear Live CDs.

En el primer paso se debe de hacer una instalación de la distribución Slackware 12.1 GNU/Linux en una partición del disco duro, con el software necesario para poder particionar y ejecutar Ncurses, si el usuario lo desea puede crear una partición extra para poder crear en dicha partición el sistema en el que cual se basará el Live CD a crearse. Para poder particionar el disco duro se puede utilizar el comando fdisk, cfdisk o la herramienta gráfica de QTParted.

El tercer paso es montar el DVD o bien la imagen que contiene la herramienta; para esto debemos ejecutar los siguientes comandos:

```
#mount /dev/cdrom /mnt/cdrom  
ó  
#mount tortilla_burner.iso /mnt/[Carpeta] -o loop
```

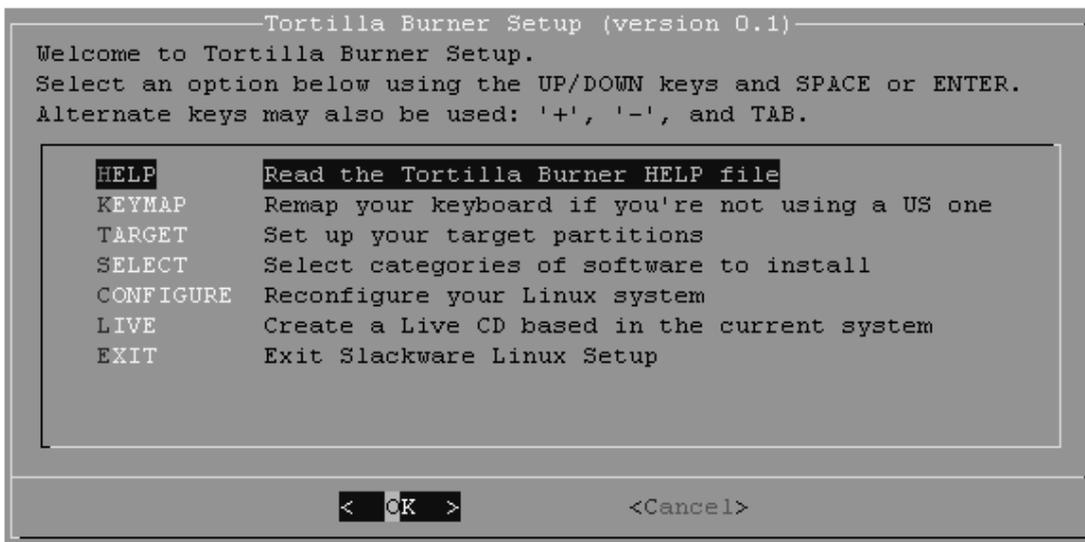
Para ejecutar la aplicación se ejecuta el script *tortilla_burner.sh* que se encuentra en la carpeta *tortilla_burner* incluida dentro del la imagen ISO o dentro del DVD que contiene la imagen .iso grabada.

Este script debe ser ejecutado con el usuario root, de no ser así el programa mostrará el mensaje "You have no permission to run `./tortilla_burner.sh` as non-root user.". Para ejecutar la aplicación escribimos:

```
#!/live_creator.sh
```

Posteriormente aparece el menú principal de la aplicación.

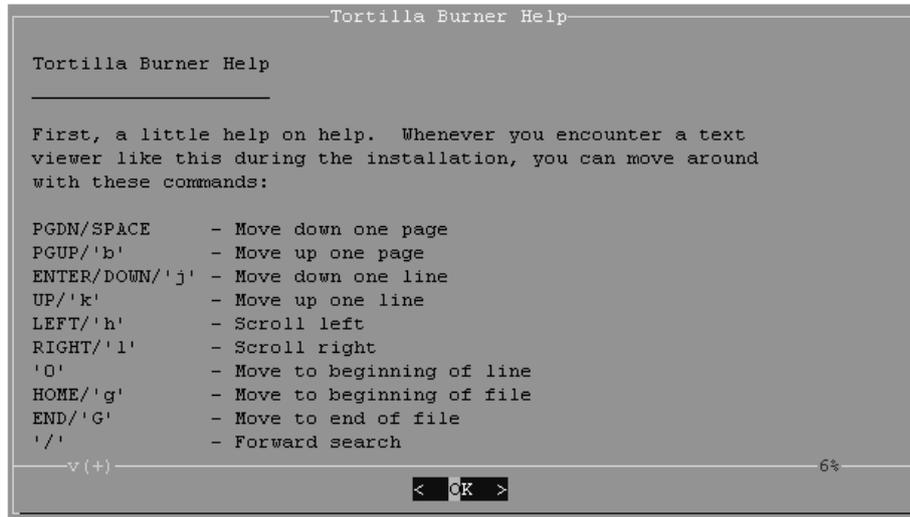
Figura 4. Pantalla inicial de la aplicación



El menú de la aplicación está integrado de las opciones HELP, KEYMAP, TARGET, SELECT, CONFIGURE, LIVE y EXIT.

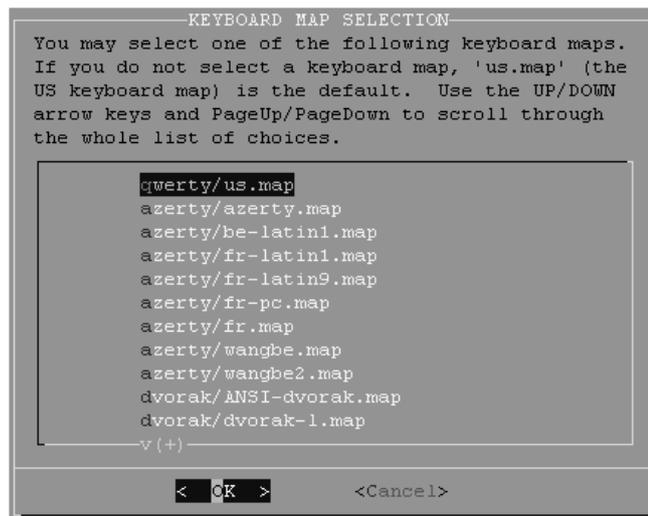
Al seleccionar la opción HELP, la aplicación muestra una guía para poder utilizar la herramienta, como también consejos para realizar personalizaciones del sistema a crear como un Live CD.

Figura 5. Pantalla de ayuda



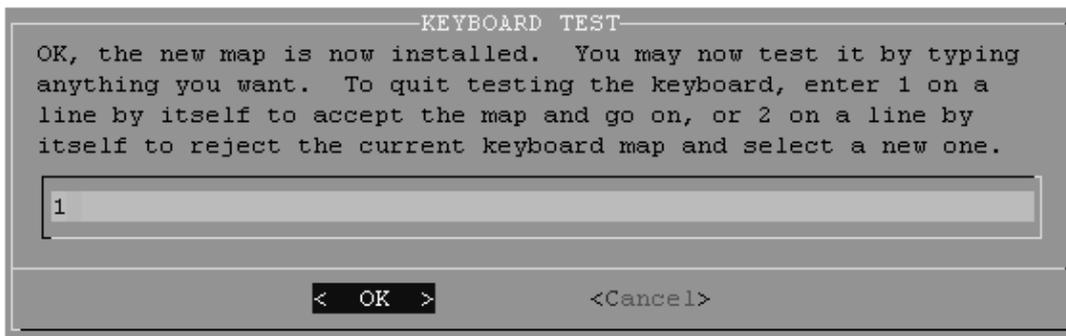
Al seleccionar la opción de **KEYMAP** muestra una pantalla para poder seleccionar el idioma del teclado para el Live CD.

Figura 6. Pantalla de selección de idioma de teclado



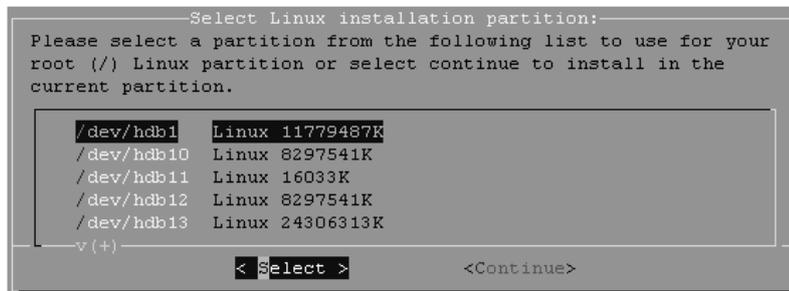
Luego de seleccionar el idioma deseado se muestra una pantalla en la cual se permite probar el idioma seleccionado y si se ingresa la tecla "1" se confirma el idioma de teclado por defecto del Live CD a crearse.

Figura 7. Diálogo para prueba de idioma seleccionado



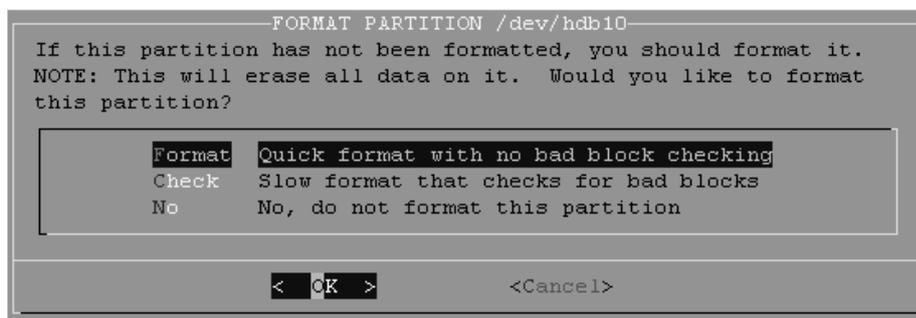
Posteriormente de seleccionar el idioma del teclado, se elige si se desea crear un Live CD dentro de una partición o bien dentro de una carpeta en la partición del sistema donde se esta ejecutando la aplicación, para esto la aplicación muestra las particiones disponibles o si se selecciona <Continue> crea el sistema en la misma partición del sistema que ejecuta la aplicación para crear Live CDs.

Figura 8. Diálogo para seleccionar partición de trabajo



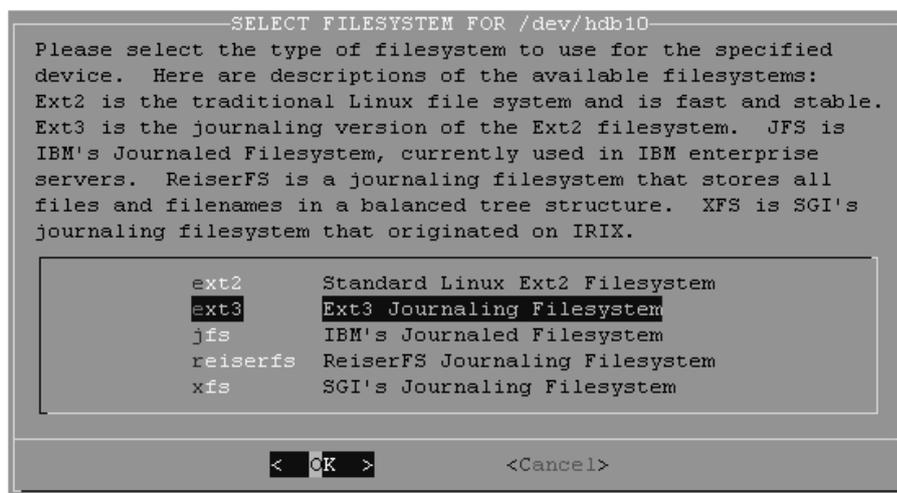
Si se selecciona una partición, posteriormente se monta y formatea si así se desea, sino se direcciona todo el trabajo a la carpeta /mnt/MiLinux ubicada en la partición del sistema en el que se ejecuta Tortilla Burner. En la Figura 9 se selecciona la opción de Formatear la partición.

Figura 9. Selección de tipo de formato a aplicar a la partición



Luego se elige el tipo de sistemas de archivos para darle formato a la partición seleccionada anteriormente.

Figura 10. Diálogo para prueba de idioma seleccionado



Después de seleccionar los parámetros para formatear la partición, se muestra un dialogo para indicar que el proceso de formateo se esta realizando.

Figura 11. Pantalla de formateo de partición

```
—FORMATTING—
Formatting /dev/hdb10
Size in 1K blocks: 8297541
Filesystem type: reiserfs
```

Si fuera necesario seleccionamos otras particiones para cargar las carpetas del nuevo sistema a crear(ej.: /opt,/boot), esta opción también permite crear un sistema GNU/Linux el cual puede ser utilizado, ya que la herramienta esta basada en los scripts de Slackware. Para poder arrancar este sistema es necesario realizar las configuraciones de Swap.

Figura 12. Pantalla de selección de otras particiones

```
—Select other Linux partitions for /etc/fstab—
You seem to have more than one partition tagged as type Linux.
You may use these to distribute your Linux system across more than
one partition.  Currently, you have /dev/hdb10 mounted as your /
partition.  You might want to mount directories such as /home or
/usr/local on separate partitions.  You should not try to mount
/etc, /sbin, or /bin on their own partitions since they contain
utilities needed to bring the system up and mount partitions.
Also, do not reuse a partition that you've already entered before.
Please select one of the Linux partitions listed below, or if
you're done, hit <Continue>.

/dev/hdb1  Linux 1179487K
{IN USE}  /dev/hdb10 on / Linux 8297541K
/dev/hdb11 Linux 16033K
/dev/hdb12 Linux 8297541K
v(+)

< Select >      <Continue>
```

Posteriormente se agregan las particiones seleccionadas al archivo */etc/fstab* en caso de que el sistema no se vaya a crear como Live CD.

Figura 13. Carga de particiones al archivo */etc/fstab*

```
-----DONE ADDING LINUX PARTITIONS TO /etc/fstab-----
Adding this information to your /etc/fstab:
/dev/hdb10      /                reiserfs  defaults    1    1
-----100%-----
< OK >
```

Después de ser creada la partición que será utilizada para crear el sistema GNU/Linux que contendrá el nuevo Live CD, se mostrará una pantalla de selección del tipo de Disco pre configurado para crear el sistema.

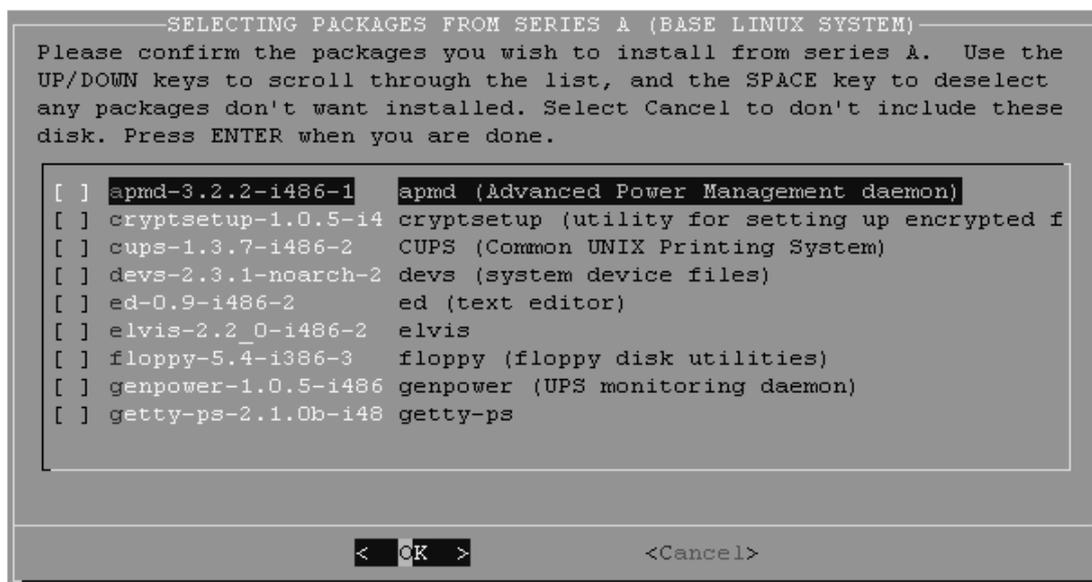
Figura 14. Selección de sistemas mínimos pre configurados

```
-----Tortilla Burner (version 0.1)-----
Select the type of installation below using the UP/DOWN keys and
SPACE or ENTER. Alternate keys may also be used: '+', '-', and TAB.
-----
MINIMAL  Only install a minimal core system
X        Install the core an X11 system
KDE      Full minimal system with kde and xorg
EXIT     Exit Live Burner setup
-----
< OK >      <Cancel>
```

En la pantalla de la Figura 14, se muestran 3 opciones básicas, “**MINIMAL**” la cual desempaqueta un sistema base mínimo para crear un sistema basado en consola, “**X**” que instala el sistema mínimo más un conjunto de paquetes necesarios para poder ejecutar aplicaciones gráficas como Firefox, GIMP, etc., y “**KDE**” que permite instalar un sistema mínimo con soporte de gráficos y el administrador de ventanas KDE. La opción “**EXIT**”, permite abortar la selección de paquetes.

Luego de que se desempaqueta el sistema seleccionado, aparece una sucesión de diálogos en los cuales se puede seleccionar software extra a incluir en el sistema a convertir en Live CD.

Figura 15. Selección de paquetes disponibles



El software extra incluido en la aplicación se encuentra en un repositorio interno distribuido en una imagen .iso llamada *repository.iso*, en el cual se encuentra el software (paquetes .tgz) clasificado según su uso.

Entre estas pantallas se mostrarán las siguientes clasificaciones:

- A: Programas para crear el sistema Base.
- AP: Aplicaciones basadas en consola.
- D: Programas para desarrollo.
- E: Emacs.
- F: FAQs y Documentación.
- KDE: Administrado de ventanas gráficas KDE.
- L: Librerías necesarias para programas gráficos.
- N: Programas relacionados a servicios y monitoreo de redes.
- T: teTex.
- TCL: Tcl, Tk, TclX.
- X: Soporte para sistema gráfico X .
- XAP: Aplicaciones para sistema gráfico X.
- Y: Juegos.

Para seleccionar el software, nos ubicamos en el botón <OK> y luego presionamos <Barra Espaciadora> para hacer efectiva la selección del software. Sino se desea incluir ningún paquete de esa clasificación seleccionamos <CANCEL>.

Después de seleccionar el software extra a incluir, la aplicación inicia un proceso de resolución de dependencias para incluir todo el software necesario para poder ejecutar los programas seleccionados en el paso anterior. Dependiendo del software seleccionado puede variar el tiempo de compresión para generar los módulos con el software seleccionado. Estos módulos son cargados en el momento de arranque del nuevo Live CD, y habilitan el uso del software seleccionado en los pasos anteriores.

Posteriormente de la instalación de software se regresa a la pantalla inicial para poder iniciar la compresión y generación de módulos para crear el Live CD. Al no permitir que se ejecute automáticamente este paso, permite al usuario realizar otras personalizaciones al sistema, esta opción es para usuarios intermedios o avanzados, esto se debe hacer en una línea de comandos.

Para realizar esta personalización, el usuario debe de montar /dev, /proc y /sys al sistema creado en /mnt/MiLinux y posteriormente ejecutar el comando chroot para realizar las personalizaciones necesarias.

Para realizar una personalización avanzada el usuario debe acceder al sistema con los siguientes comandos:

```
#mount repository.iso /mnt/MiLinux/mnt/repository -o loop
#mount -bind /proc /mnt/MiLinux/proc
#mount -bind /dev /mnt/MiLinux/dev
#mount -bind /sys /mnt/MiLinux/sys
#chroot /mnt/MiLinux /bin/bash
```

Es importante mencionar que el sistema creado incluye una versión modificada del programa Swaret(Programa que instala paquetes para slackware) para resolver dependencias de los paquetes a instalar sin necesidad de que confirme la instalación de los paquetes necesarios para resolver sus dependencias.

Luego de haber realizado las personalizaciones necesarias el usuario deberá ejecutar exit en la línea de comandos y luego deberá de desmontar las ubicaciones /mnt/MiLinux/proc, /mnt/MiLinux/dev, /mnt/MiLinux/sys, esto como una buena práctica.

El paso siguiente a esta personalización para usuarios avanzados será ejecutar el script para generar la imagen .iso del Live CD a crearse. O bien seleccionar la opción “LIVE” en el menú principal. Para generar el Live CD se selecciona la opción “LIVE”, posteriormente se muestran una serie de configuraciones básicas del sistema para crear el Live CD.

Figura 16. Opción para generar el Live CD basado en el sistema creado

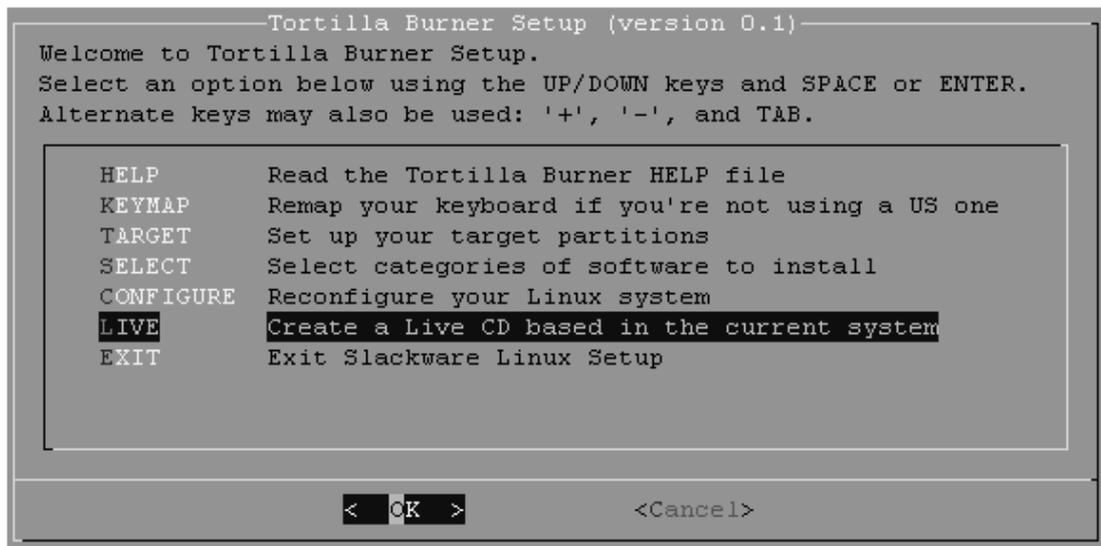
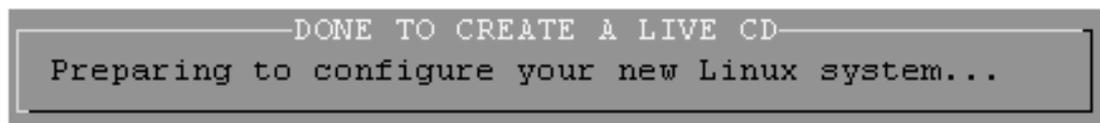


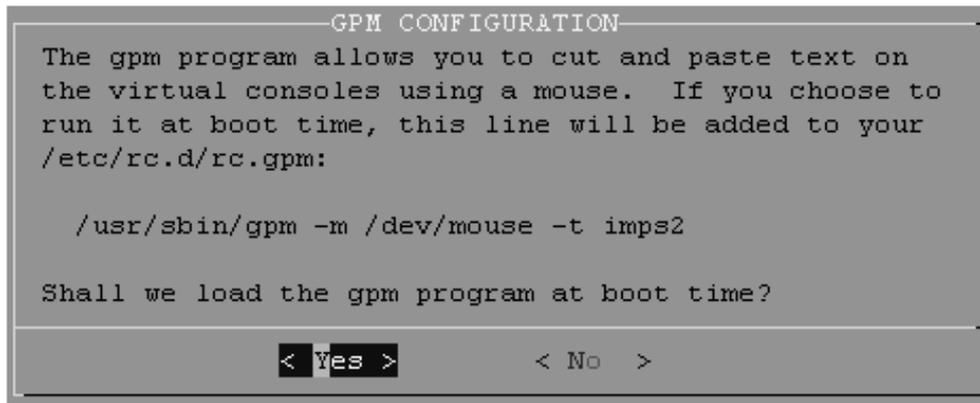
Figura 17. Pantalla previa a la configuración del sistema



Dentro de las configuraciones incluidas por Tortilla Burner esta la configuración de red, de fuentes para consola, de los servicios a iniciar en el momento de arranque de la computadora para pasar finalmente a la generación de la imagen del Live CD.

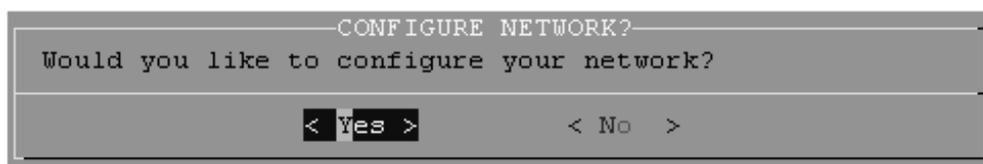
En la Figura 18 se muestra la configuración de soporte de uso de mouse en consola al momento de carga.

Figura 18. Diálogo de soporte de mouse en consola



Para la configuración de red se muestra la pantalla si se desea o no realizar la configuración de red o detección de conexiones de internet o redes al momento de la carga del Live CD.

Figura 19. Diálogo para poder configurar la red



Posteriormente se realiza la configuración del nombre la máquina y del dominio al que pertenece.

En las siguientes pantallas se realiza la configuración de red del sistema.

Figura 20. Diálogo para ingresar el nombre de la máquina

```
ENTER HOSTNAME
First, we'll need the name you'd like to give your host.
Only the base hostname is needed right now. (not the domain)

Enter hostname:
blackbird

< OK >      <Cancel>
```

Figura 21. Diálogo para ingresar dominio de la máquina

```
ENTER DOMAINNAME FOR 'blackbird'
Now, we need the domain name for this machine, such as:
example.org

Do not supply a leading '.'

Enter domain name for blackbird:
net4us.org

< OK >      <Cancel>
```

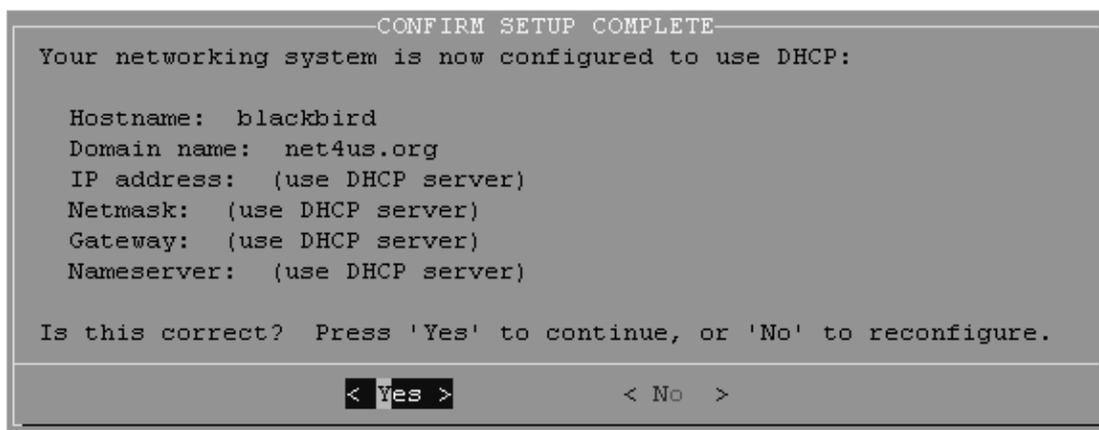
Figura 22. Diálogo para ingresar el proveedor de conexión

```
SET DHCP HOSTNAME
Some network providers require that the DHCP hostname be
set in order to connect. If so, they'll have assigned a
hostname to your machine, which may look something like
CC-NUMBER-A (this depends on your ISP). If you were
assigned a DHCP hostname, please enter it below. If you
do not have a DHCP hostname, just hit ENTER or Cancel.

< OK >      <Cancel>
```

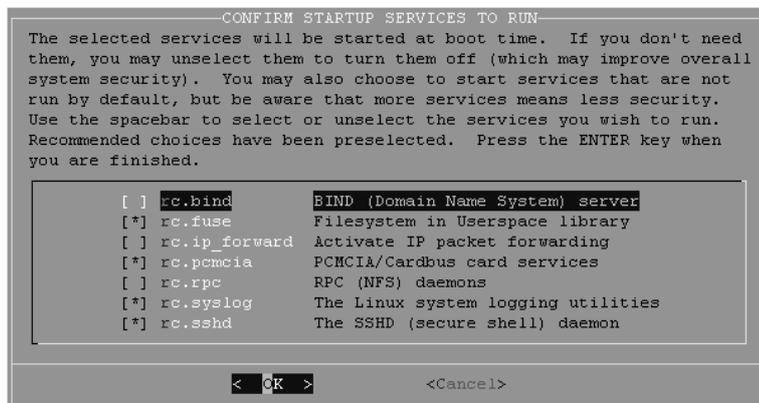
Finalmente se muestra un resumen de la configuración de red realizada.

Figura 23. Diálogo de resumen de configuración de red



La siguiente pantalla que se muestra permite seleccionar los servicios que se van a cargar al momento de arranque del Live CD a generar, para seleccionar un servicio debemos de presionar <Barra Espaciadora> para marcarlo, o volver a presionar <Barra Espaciadora> para desmarcar un servicio previamente marcado.

Figura 24. Diálogo de selección de servicios



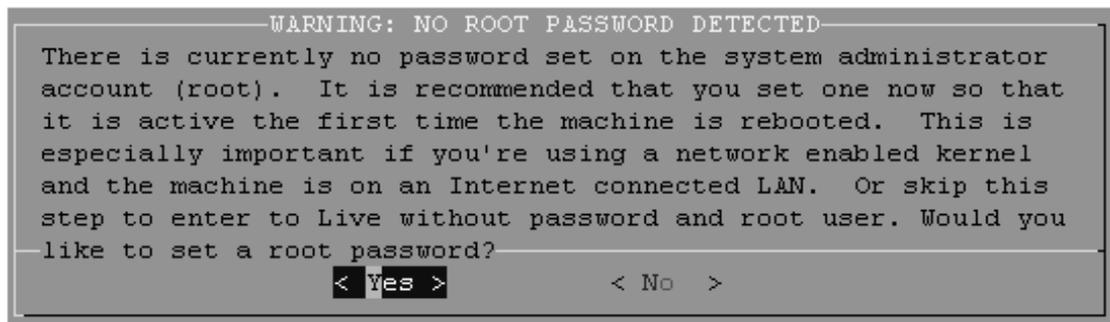
Luego se seleccionan otras fuentes para el tipo de letra de la consola.

Figura 25. Diálogo para poder seleccionar fuentes para consola



La última configuración importante es establecer la contraseña de administrador del sistema, para esto se ofrecen dos opciones si se ingresa una contraseña se deberá de apuntar para poder acceder al sistema, sino se ingresa si el usuario al tratar de ingresar al Live CD como super usuario(root), el usuario no pide contraseña, de tal forma que se permite manejar de una forma más libre respecto a permisos y ejecución de comandos del Live CD.

Figura 26. Diálogo para establecer contraseña de super usuario



Al terminar estas configuraciones muestra un mensaje para indicar que el programa esta listo para empezar a crear la imagen que contendrá todo el sistema configurado con los parámetros ingresados, el cual se podrá ejecutar desde el Live CD a crearse.

Figura 27. Diálogo para poder seleccionar fuentes para consola



Figura 28. Ingreso de nombre del Live CD

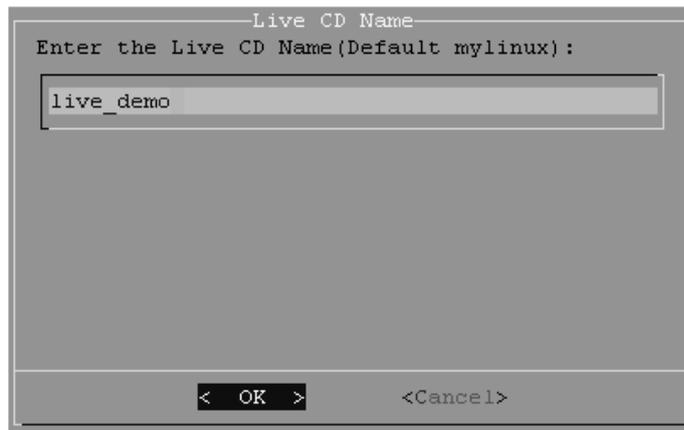
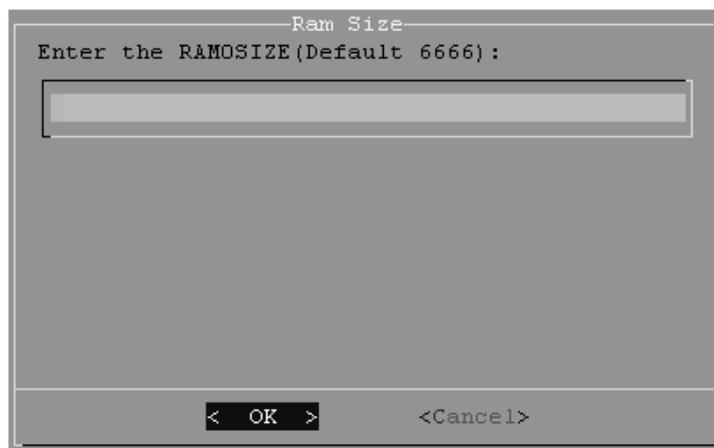


Figura 29. Ingreso de uso de memoria para módulos



Al terminar la generación de los archivos necesarios se muestra un mensaje de la ubicación donde se encuentra los archivos generados para poder grabar la imagen del Live CD en un disco o bien instalar el sistema en una memoria USB.

Figura 30. Mensaje para mostrar ubicación de archivos generados



Figura 31. Mensaje de ubicación de imagen ISO del Live CD creado



También podemos probar la imagen .ISO sin necesidad de grabarla en un disco con qemu, este programa esta incluido dentro de la imagen de Tortilla Burner(Carpeta qemu), para poder probar la imagen ejecutamos:

```
#qemu-system-x86_64 -cdrom IMAGEN.iso -boot d
```

Las pantallas anteriores muestran la configuración básica para empezar a comprimir y generar los archivos necesarios para poder crear el Live CD a partir del sistema creado anteriormente.