

# Respuestas a preguntas frecuentes relacionadas con el Software Libre

Colibrí <colibri@bachue.com>

Version 0.2, Mayo 8 2004

## Resumen

Este documento pretende recopilar las respuestas a las principales preguntas frecuentes relacionadas con el software libre. Aunque la mayoría de preguntas son técnicas, también se incluyen preguntas de otra índole. El público objetivo de cada respuesta varía: algunas son respondidas para personas con muy pocos conocimientos sobre software libre mientras que otras requieren conocimientos avanzados para su comprensión. Este documento es elaborado de manera colaborativa por miembros de Colibrí y aún está bastante incompleto.

## **Nota de Copyright**

Copyright © 2003, 2004 Colibrí  
Colibrí <colibri@bachue.com>

Este documento puede ser usado en los términos descritos en la Licencia Pública GNU versión 2 o posterior. (<http://www.gnu.org/copyleft/gpl.html>)

# Índice general

<b>1. Información general sobre ésta FAQ.</b>	<b>1</b>
1.1. ¿Cuáles son los derechos de autor de este documento? . . . . .	1
1.2. ¿Quiénes son los autores de este documento? . . . . .	1
1.3. ¿Cómo puedo contribuir a mejorar este documento? . . . . .	2
1.4. ¿Cuál es la versión actual de este documento? . . . . .	3
1.5. ¿Qué herramientas utilizan para construir este documento? . . . . .	3
<b>2. Preguntas básicas sobre GNU/Linux</b>	<b>5</b>
2.1. ¿Qué es Linux? <sup>1</sup> . . . . .	5
2.2. ¿Cuál es la pronunciación correcta de Linux? <sup>2</sup> . . . . .	5
2.3. ¿Cuáles son las principales ventajas de Linux? <sup>3</sup> . . . . .	6
2.4. ¿Es cierto que GNU/Linux es difícil de usar? <sup>4</sup> . . . . .	6
2.5. ¿Qué se puede hacer con GNU/Linux? <sup>5</sup> . . . . .	6
2.6. ¿Cuál es la historia de Linux? <sup>6</sup> . . . . .	7
2.7. ¿Que hardware se debe usar para trabajar con video y audio en GNU/Linux? <sup>7</sup> .	7
2.8. Distribuciones de GNU/Linux . . . . .	8
2.8.1. ¿Qué es una distribución? <sup>8</sup> . . . . .	8

<sup>1</sup>Esta respuesta fue escrita por Alejandro Forero Cuervo incorporando varias modificaciones al texto del MerFAQ que sirvió como base.

<sup>2</sup>Esta respuesta provenía del MerFAQ (como parte de otra respuesta) y fue editada ligeramente por Alejandro Forero Cuervo.

<sup>3</sup>Esta respuesta fue escrita por Alejandro Forero Cuervo.

<sup>4</sup>Esta pregunta y su respuesta fueron añadidas por Alejandro Forero Cuervo.

<sup>5</sup>Esta respuesta fue escrita por Alejandro Forero Cuervo con base en ideas provenientes del MerFAQ.

<sup>6</sup>Esta respuesta fue escrita por Alejandro Forero Cuervo incorporando varias modificaciones al texto del MerFAQ que sirvió como base.

<sup>7</sup>Camilo Martínez envió esta pregunta a la lista de correo de Colibrí. La respuesta se basa en la enviada por Mauricio Alvarez, con pequeñas ediciones de forma por Alejandro Forero Cuervo.

<sup>8</sup>Esta pregunta fue respondida por Alejandro Forero Cuervo, con base en información del MerFAQ.

2.8.2. ¿Cuáles son las principales distribuciones? <sup>9</sup> . . . . .	8
2.8.3. ¿Qué distribución me recomiendan utilizar? <sup>10</sup> . . . . .	9
2.8.4. ¿Cómo puedo obtener una copia de una distribución? <sup>11</sup> . . . . .	10
2.9. ¿Qué tipo de hardware necesito para correr GNU/Linux? <sup>12</sup> . . . . .	10
2.10. Virus en GNU/Linux . . . . .	11
2.10.1. ¿Es cierto que GNU/Linux es inmune a virus? . . . . .	11
2.10.2. ¿Existen antivirus para GNU/Linux? <sup>13</sup> . . . . .	12
<b>3. Administración del sistema</b> . . . . .	<b>13</b>
3.1. Bases de datos . . . . .	13
3.1.1. ¿Cómo configurar PostgreSQL? <sup>14</sup> . . . . .	13
3.2. Servicios de Internet . . . . .	14
3.2.1. ¿Cómo administrar Apache 1.3.x? <sup>15</sup> . . . . .	14
3.2.2. ¿Cómo configurar Sendmail? <sup>16</sup> . . . . .	21
3.3. Periféricos . . . . .	25
3.3.1. ¿Cómo montar una unidad de diskette o cdrom? . . . . .	25
3.3.2. ¿Cómo configurar una impresora PPA (aka. winprinter)? <sup>17,18</sup> . . . . .	27
3.3.3. ¿Cómo configuro un Winmodem? . . . . .	30
3.3.4. ¿Cómo configuro el mapa de caracteres de mi teclado? . . . . .	34
3.4. Otras preguntas sin clasificar aún. . . . .	42
3.4.1. ¿Cómo configuro el X Window System? . . . . .	42
3.4.2. ¿Cómo restringir los directorios públicos en Apache? <sup>19</sup> . . . . .	45

<sup>9</sup>Esta pregunta fue respondida por Alejandro Forero Cuervo.

<sup>10</sup>Esta pregunta fue respondida por Alejandro Forero Cuervo, con especial cuidado en ser objetivo.

<sup>11</sup>Esta pregunta fue respondida por Alejandro Forero Cuervo con base en la información proveniente del MerFAQ.

<sup>12</sup>Respuesta escrita por Alejandro Forero Cuervo con base en el texto original de la MerFAQ y sugerencias de Jhair Tocancipá.

<sup>13</sup>Respuesta escrita por Alejandro Forero Cuervo con base en información enviada a la lista por Polkan García.

<sup>14</sup>Respuesta del Grupo Merlinux. 16 de Octubre del 2001.

<sup>15</sup>Por Juan Luis Baptiste. jlbaptiste@merlinux.org. Julio 28 de 1999.

<sup>16</sup>Por Juan Luis Baptiste. jlbaptiste@merlinux.org. Septiembre 17 de 1999.

<sup>17</sup>Respuesta por Herman Fabián Sandoval. Revisada por Offray Vladimir Luna Cárdenas.

<sup>18</sup>Específico para las siguientes impresoras: HP710, HP720, HP820 y HP1000.

<sup>19</sup>Repuesta por Jaime Torres, MerLinux

<b>4. Desarrollo</b>	<b>49</b>
4.1. ¿Qué ambientes de programación basados en software libre se encuentran disponibles? <sup>20</sup>	49
4.2. ¿Qué significa un paradigma de programación? <sup>21</sup>	50
4.3. ¿Qué quiere decir que un lenguaje sea funcional? <sup>22</sup>	51
4.4. Programación orientada a objetos <sup>23</sup>	52
4.4.1. ¿Qué es la programación orientada a objetos?	52
4.4.2. ¿Qué lenguajes orientados a objetos existen?	53
4.4.3. ¿Qué quiere decir que un objeto hereda de otro?	53
4.4.4. ¿Qué quiere decir que un lenguaje permite la herencia múltiple?	54
4.4.5. ¿Qué es el encapsulamiento en un lenguaje orientado a objetos?	54
4.4.6. ¿Qué es el paso de mensajes en un lenguaje orientado a objetos?	54
4.4.7. ¿Qué programas de software libre han sido desarrollando utilizando programación orientada a objetos?	54
4.5. ¿Qué quiere decir que un lenguaje provea recolección de basura automática? <sup>24</sup>	55
4.6. ¿Qué son las continuaciones?	55
4.7. Desarrollo en C	55
4.7.1. ¿Qué ventajas y desventajas tiene C en comparación con otros lenguajes de programación?	55
4.8. Desarrollo en Haskell	55
4.9. Desarrollo en Java	55
4.9.1. ¿Qué herramientas libres existen para desarrollo en Java? <sup>25</sup>	55
4.9.2. ¿Qué ventajas y desventajas tiene Java en comparación con otros lenguajes de programación?	57
4.10. Desarrollo en Lisp	57
4.10.1. ¿Qué es Lisp? <sup>26</sup>	57

<sup>20</sup>Esta pregunta fue propuesta por Jhair Tocancipá y editada, con base en sugerencias de Juan Luis Baptiste, Gustavo Gonzalez, y Sergio Roa, por Alejandro Forero Cuervo.

<sup>21</sup>Esta respuesta fue redactada por Alejandro Forero Cuervo.

<sup>22</sup>Esta respuesta fue escrita por Alejandro Forero Cuervo, con base en información extraída del documento Erbis Scheme de su autoría.

<sup>23</sup>Las respuestas de esta sección fueron escritas por Sergio Roa [s.roa@computer.org](mailto:s.roa@computer.org) y Alejandro Forero Cuervo [bachue@bachue.com](mailto:bachue@bachue.com).

<sup>24</sup>Pregunta planteada y contestada por Alejandro Forero Cuervo.

<sup>25</sup>Repuesta inicialmente redactada por Jhair Tocancipá

<sup>26</sup>Esta respuesta fue redactada por Alejandro Forero Cuervo.

4.10.2. ¿Cuál es la historia de Lisp? <sup>27</sup>	57
4.10.3. ¿Qué características particulares diferencian a Lisp de otros lenguajes de programación? <sup>28</sup>	57
4.10.4. ¿Cuáles son las principales diferencias entre Common Lisp y Scheme?	58
4.10.5. ¿Qué es Emacs-Lisp? <sup>29</sup>	58
4.10.6. ¿Cómo puedo aprender a programar en Lisp? <sup>30</sup>	58
4.10.7. ¿Qué implementaciones de Lisp me recomiendan utilizar? <sup>31</sup>	59
4.10.8. ¿Qué programas de software libre hay escritos en Lisp? <sup>32</sup>	59
4.11. Desarrollo en Perl	60
4.11.1. ¿Qué ventajas y desventajas tiene Perl en comparación con otros lenguajes de programación?	60
4.11.2. ¿Cómo puedo aprender a programar en Perl?	60
4.12. Desarrollo en PHP	60
4.13. Desarrollo en Python	60
4.13.1. ¿Qué ventajas y desventajas tiene Python en comparación con otros lenguajes de programación?	60
<b>5. Aplicaciones</b>	<b>61</b>
5.1. Internet	61
5.1.1. ¿Cuáles son los comandos básicos de <i>Internet Relay Chat (IRC)</i> ? <sup>33</sup>	61
5.1.2. ¿Cómo conectarse a Internet?	64
5.1.3. ¿Que programas de noticias/correo electrónico libres pueden recomendar? <sup>34</sup>	64
<b>6. Software Libre y Educación</b>	<b>67</b>
6.1. Organizaciones en Colombia	67
6.1.1. ¿Existen organizaciones en Colombia dedicadas a tratar el tema de software libre y la educación?	67

<sup>27</sup>Esta respuesta fue redactada por Alejandro Forero Cuervo.

<sup>28</sup>Esta respuesta fue escrita por Alejandro Forero Cuervo. Partes de la información fueron extraídas del R5RS.

<sup>29</sup>Respuesta por Alejandro Forero Cuervo.

<sup>30</sup>Esta inquietud fue planteada por Camilo Uribe y respondida por Alejandro Forero Cuervo.

<sup>31</sup>Esta respuesta fue escrita por Alejandro Forero Cuervo.

<sup>32</sup>Esta respuesta fue añadida por Alejandro Forero Cuervo.

<sup>33</sup>Respuesta original por Juan F. A. Saldarriaga [nebiros@gigax.org](mailto:nebiros@gigax.org), editada y ampliada por Jhair Tocancipá.

<sup>34</sup>Propuesta por Jhair Tocancipá.

<b>7. Instalación</b>	<b>69</b>
7.1. No hay preguntas aún en esta sección. . . . .	69



## Capítulo 1

# Información general sobre ésta FAQ.

### 1.1. ¿Cuáles son los derechos de autor de este documento?

Inicialmente éste documento fue protegido bajo los términos de la GNU FDL (GFDL, Free Documentation License). Desafortunadamente ésta licencia contiene términos que pueden hacer de una obra una no 100 % libre. Por ejemplo, la GFDL permite incluir secciones invariantes en una obra, las cuales no pueden ser modificadas (ni removidas), haciendo la GFDL incompatible con licencias libres como la GPL.

Puede encontrar un resumen de las desventajas de la GFDL aquí ([http://people.debian.org/~srivasta/Position\\_Statement.html](http://people.debian.org/~srivasta/Position_Statement.html)).

Una discusión en la lista debian-legal, provee puntos de vista a favor y en contra de la GFDL. La discusión se encuentra archivada aquí (<http://thread.gmane.org/gmane.linux.debian.devel.legal/7266>)

Este documento está protegido bajos los términos de la Licencia Pública GNU versión 2 o posterior.

### 1.2. ¿Quiénes son los autores de este documento?

La versión inicial de este documento se basó en los contenidos de las FAQs del grupo de usuarios de Linux Merlinux (<http://www.merlinux.org/>), quienes amablemente los cedieron a Colibrí para este propósito.

En el contenido de éste documento han trabajado, entre otras, las siguientes personas (en orden alfabético):

- Hernan Pablo Alvarez <[hernanalvarez@arnet.com.ar](mailto:hernanalvarez@arnet.com.ar)> (‘¿Cómo configuro un Win-modem?’ en la página 30)

- Mauricio Alvarez <alvarez@ac.upc.es> (respuesta '¿Que hardware se debe usar para trabajar con video y audio en GNU/Linux?'<sup>1</sup> en la página 7)
- Juan Luis Baptiste <juancho@linuxmail.org>
- Alejandro Forero Cuervo <bachue@bachue.com>
- Offray Vladimir Luna <offray.luna@javeriana.edu.co> (respuesta '¿Cómo configurar una impresora PPA (aka. winprinter)?'<sup>2,3</sup> en la página 27)
- Sergio Roa <s.roa@computer.org>
- Juan F. A. Saldarriaga <nebiros@gigax.org> ('PCTel mini-COMO Version 1.1'<sup>4</sup> en la página 31)
- Herman Fabián Sandoval (respuesta '¿Cómo configurar una impresora PPA (aka. winprinter)?'<sup>5,6</sup> en la página 27)
- Igor Támara <igor@tamarapatino.org>
- Jhair Tocancipá Triana <jhair\_tocancipa@gmx.net>
- Jaime Torres (respuesta a pregunta '¿Cómo restringir los directorios públicos en Apache?'<sup>7</sup> en la página 45)
- Camilo Uribe <kmilo@softhome.net> (propuesta de incluir archivo con la pronunciación de Linux en '¿Cuál es la pronunciación correcta de Linux?'<sup>8</sup> en la página 5)

### 1.3. ¿Cómo puedo contribuir a mejorar este documento?

Este documento se desarrolla de forma abierta y colaborativa por el equipo de webmasters de Colibrí. Para coordinar el trabajo utilizamos la lista de correo <colibri-www@bachue.com> o el canal de IRC (chat) #colibri en el servidor irc.freenode.net.

Lo invitamos contribuir con la elaboración de éste documento mejorando la calidad de las respuestas o adicionando nuevas preguntas. Para hacerlo basta con que nos escriba a <colibri-www@bachue.com> con sus sugerencias. Las críticas constructivas siempre serán bienvenidas.

---

<sup>1</sup>Camilo Martínez envió esta pregunta a la lista de correo de Colibrí. La respuesta se basa en la enviada por Mauricio Alvarez, con pequeñas ediciones de forma por Alejandro Forero Cuervo.

<sup>2</sup>Respuesta por Herman Fabián Sandoval. Revisada por Offray Vladimir Luna Cárdenas.

<sup>3</sup>Específico para las siguientes impresoras: HP710, HP720, HP820 y HP1000.

<sup>4</sup>Respuesta de Juan F. A. Saldarriaga. nebiros@gigax.org. Grupo de Implementación GNU y Arquitecturas \*X (GiGAX) - <http://www.gigax.org>. 15 de Abril del 2003.

<sup>5</sup>Respuesta por Herman Fabián Sandoval. Revisada por Offray Vladimir Luna Cárdenas.

<sup>6</sup>Específico para las siguientes impresoras: HP710, HP720, HP820 y HP1000.

<sup>7</sup>Respuesta por Jaime Torres, MerLinux

<sup>8</sup>Esta respuesta provenía del MerFAQ (como parte de otra respuesta) y fue editada ligeramente por Alejandro Forero Cuervo.

La última versión de este documento se encuentra almacenada en el repositorio de contenidos del sitio web de Colibrí. Si desea, puede descargar la última versión, trabajar sobre ella y enviarnos un archivo con sus cambios para que los integremos a la versión oficial. Hacer ésto puede agilizar bastante el proceso de inclusión de sus modificaciones en la versión oficial.

Puede encontrar instrucciones detalladas (<http://bachue.com/colibri/acerca-de-web.html#subversion>) sobre cómo utilizar Subversion para enviarnos archivos de cambios a los contenidos de nuestro repositorio. Si planea realizar varios cambios, no dude en pedirnos que le creémos una cuenta (login) para que pueda aplicarlos directamente.

Los archivos desde los que se genera éste documento están almacenados en el directorio `/info/faq` del repositorio.

## 1.4. ¿Cuál es la versión actual de este documento?

Este documento se encuentra en constante desarrollo y aún no existe una versión oficial del mismo. Periódicamente puede encontrar nuevas versiones en distintos formatos en los siguientes enlaces:

- HTML (<http://bachue.com/colibri/faq/html/>)
- Texto plano ([http://people.freenet.de/colibri\\_faq/faq.txt](http://people.freenet.de/colibri_faq/faq.txt))
- PDF ([http://people.freenet.de/colibri\\_faq/faq.pdf](http://people.freenet.de/colibri_faq/faq.pdf))
- PS ([http://people.freenet.de/colibri\\_faq/faq.ps.gz](http://people.freenet.de/colibri_faq/faq.ps.gz))

## 1.5. ¿Qué herramientas utilizan para construir este documento?

Este documento fue escrito usando la DebianDoc SGML DTD y herramientas pertinentes.

Puede encontrar más información sobre el uso de SGML en Debian GNU/Linux en <http://www.debian.org/doc/manuals/sgml-howto/>.



## Capítulo 2

# Preguntas básicas sobre GNU/Linux

En este capítulo incluimos la respuesta a preguntas básicas relacionadas con GNU/Linux.

El encargado de éste capítulo es Alejandro Forero Cuervo <bachue@bachue.com>.

### 2.1. ¿Qué es Linux? <sup>1</sup>

Cuando la mayoría de la gente utiliza el término Linux, se refieren a un sistema operativo particular (programa que administra los recursos de un computador, tal y como DOS, Windows, MacOS, etc.).

Linux, técnicamente, es únicamente un componente de todo el sistema operativo (el núcleo), y viene acompañado por una gran cantidad de herramientas de otros proyectos sin las cuales sería completamente inútil. Entre ellas se encuentran las herramientas desarrolladas por el proyecto GNU (tan importantes que algunas personas se refieren al sistema como “GNU/Linux”), la implementación del sistema de ventanas X conocida como XFree86, el servidor web Apache, el lenguaje de programación Perl, el entorno gráfico KDE y muchísimas más.

### 2.2. ¿Cuál es la pronunciación correcta de Linux? <sup>2</sup>

Linux se pronuncia tal y como se leería en español (y no “lainucs”, como sonaría en inglés). Esto obedece a que el término viene del finlandés, cuya pronunciación se asemeja bastante a la del español. Puede encontrar un archivo de audio donde Linus Torvalds pronuncia “Linux” en <http://www.paul.sladen.org/pronunciation/>.

---

<sup>1</sup>Esta respuesta fue escrita por Alejandro Forero Cuervo incorporando varias modificaciones al texto del MerFAQ que sirvió como base.

<sup>2</sup>Esta respuesta provenía del MerFAQ (como parte de otra respuesta) y fue editada ligeramente por Alejandro Forero Cuervo.

### 2.3. ¿Cuáles son las principales ventajas de Linux?<sup>3</sup>

Quizá la principal ventaja de GNU/Linux es que los principales componentes necesarios para su funcionamiento son desarrollados y distribuidos en el modelo de software libre. Por ejemplo, el XFree86 utiliza su propia licencia de software libre mientras que Linux (el núcleo) y la librería de funciones de C son distribuidas bajo la licencia GNU GPL. Aunque no hay nada que impida el desarrollo y distribución de programas no libres que operen en GNU/Linux, su comunidad de usuarios suele prestar especial atención a la libertad del software y por eso es que han tenido especial cuidado en que no se requieran componentes no libres (propietarios) para utilizarlo.

Pendiente: añadir información técnica.

### 2.4. ¿Es cierto que GNU/Linux es difícil de usar?<sup>4</sup>

GNU/Linux tiene una gran cantidad de interfaces diferentes, diseñadas con objetivos diferentes, entre las cuales elegir. Aunque algunas son bastante sofisticadas y tienden a ser preferidas por usuarios expertos, interfaces como las de los proyectos GNOME y KDE son diseñadas con especial cuidado en ser muy intuitivas lo que las hace muy fáciles de usar. Ambos proyectos distribuyen una cantidad considerable de programas muy fáciles de utilizar para las principales labores que los usuarios comunes de computadores requieren. Gracias a estos proyectos muchas personas con escasos conocimientos técnicos sobre computadores han encontrado en GNU/Linux una excelente alternativa a los sistemas de software no libres.

### 2.5. ¿Qué se puede hacer con GNU/Linux?<sup>5</sup>

GNU/Linux se puede utilizar para proveer diferentes servicios relacionados con internet (servidor web, servidor FTP, servidor de bases de datos, proveedor de acceso, etc.) y en esto su alta estabilidad y confiabilidad pueden ser una gran ventaja.

GNU/Linux también se puede usar para manejar una estación de trabajo personal: tiene una gran variedad de ambientes visuales, algunos de los cuales lo hacen muy fácil de usar. Existe una buena cantidad de programas para GNU/Linux (la mayoría de software libre) que permiten utilizarlo para las principales tareas para las que se requiere un computador de hogar: navegar en Internet, hacer documentos de texto, trabajar en hojas de cálculo, elaborar presentaciones con gráficas, llevar un registro de actividades, manejar contabilidad e información de finanzas, escuchar música, ver video, etc..

---

<sup>3</sup>Esta respuesta fue escrita por Alejandro Forero Cuervo.

<sup>4</sup>Esta pregunta y su respuesta fueron añadidas por Alejandro Forero Cuervo.

<sup>5</sup>Esta respuesta fue escrita por Alejandro Forero Cuervo con base en ideas provenientes del MerFAQ.

## 2.6. ¿Cuál es la historia de Linux? <sup>6</sup>

Linux, el kernel, nació en 1991 como una actividad recreativa de un estudiante en la Universidad de Helsinki, Finlandia, llamado Linus Torvalds. Originalmente fue llamado Freax y se basó en el kernel Minix, desarrollado por Andrew Tanenbaum. El propósito original de Minix era contar con un kernel bastante sencillo que pudiera ser comprendido en su totalidad por estudiantes universitarios en poco tiempo. Muchas personas sugerían cambios a Minix que aumentarían su desempeño pero Tanenbaum incluía muy pocas pues tendían a aumentar la complejidad del código y, por ende, lo alejaban de su objetivo. Linux nació cuando Torvalds decidió apartarse del objetivo de Minix y sacar su propio kernel.

Además de Linux hay una gran cantidad de componentes necesarios para el funcionamiento del sistema operativo. El proyecto GNU, que ha construido muchos de ellos, fue iniciado en 1984 por Richard Stallman con el propósito de crear una alternativa al sistema operativo Unix en esa época. Linus Torvalds realizó las modificaciones necesarias a las herramientas de GNU para que funcionaran utilizando su kernel.

## 2.7. ¿Que hardware se debe usar para trabajar con video y audio en GNU/Linux? <sup>7</sup>

Para trabajar con video se recomienda tener mucha capacidad de almacenamiento en discos duros de buena velocidad y utilizar en ellos algún un sistema de archivos adecuado como XFS, ext3 o ReiserFS. Si se piensa hacer cambios de formatos (transcoding) es importante tener un procesador de alta velocidad.

Otro aspecto importante de hardware es la tarjeta de video. La mayoría de los fabricantes grandes tiene soporte para Linux. ATI (<http://www.ati.com/support/faq/linux.html>), NVIDIA (<http://www.nvidia.com/object/linux.html>) y MATROX (<http://www.matrox.com/mga/support/drivers/latest/home.cfm>) son algunos que soportan activamente el uso de sus tarjetas con el kernel Linux.

En el número de diciembre de 2003 del Linux Journal (<http://www.linuxjournal.com/modules.php?op=modloadname=NS-lj-issues/issue116file=index>) hay un especial sobre multimedia. Incluye varios artículos sobre DVD y uno sobre como armar una estación de trabajo para audio y video con procesadores AMD de 64 bits.

El número de febrero de 2001 (<http://www.linuxjournal.com/modules.php?op=modloadname=NS-lj-issues/issue81file=index>) trae un especial sobre multimedia en Linux. Tiene una seria interesante de artículos sobre video comenzando desde la armada del PC, pasando por la instalación de Debian GNU/Linux y revisando las aplicaciones más comunes para video y audio (ver <http://www.linuxjournal.com/article.php?sid=4483>).

---

<sup>6</sup>Esta respuesta fue escrita por Alejandro Forero Cuervo incorporando varias modificaciones al texto del MerFAQ que sirvió como base.

<sup>7</sup>Camilo Martínez envió esta pregunta a la lista de correo de Colibrí. La respuesta se basa en la enviada por Mauricio Alvarez, con pequeñas ediciones de forma por Alejandro Forero Cuervo.

Para hacer streaming con base en MPEG4 en GNU/Linux puedes consultar las páginas del MPEG4IP Project (<http://mpeg4ip.sourceforge.net/>), el FFmpeg Multimedia System (<http://ffmpeg.sourceforge.net/>) y el Apple Darwin Streaming Server (<http://developer.apple.com/darwin/projects/streaming/>).

## 2.8. Distribuciones de GNU/Linux

### 2.8.1. ¿Qué es una distribución? <sup>8</sup>

Como las respuestas anteriores explican, hay una gran cantidad de componentes necesarios para el funcionamiento del sistema operativo. En algunos casos hay múltiples alternativas y diversas maneras de configurar el sistema, todas igualmente válidas.

Una distribución de GNU/Linux es un conjunto es una selección particular de los componentes que se utilizarán, la manera en que se configurarán y los programas complementarios que se distribuirán junto con el sistema operativo para brindar funcionalidad adicional.

Algunas distribuciones suelen tener propósitos muy particulares como brindar funcionalidad específica para artistas, llevar la mayor cantidad de funcionalidad que quepa únicamente en un diskette o dar prioridad a la seguridad del sistema, pero la mayoría suelen ser diseñadas para propósitos generales.

### 2.8.2. ¿Cuáles son las principales distribuciones? <sup>9</sup>

El siguiente es un listado incompleto de distribuciones con una breve descripción de su funcionalidad y características distintivas:

- Debian GNU/Linux (<http://www.debian.org>), distribución que presta especial atención a la libertad del software. Es desarrollado por un gran número de personas en muchos países y no por una compañía particular.
- Gentoo (<http://www.gentoo.org>), distribución de propósito general que permite ser optimizada para el procesador particular de la máquina en que será ejecutada de manera relativamente sencilla.
- Knoppix (<http://www.knoppix.org>), distribución basada en Debian con el propósito de permitir utilizar GNU/Linux sin necesidad de instalarlo en el disco duro (cargando todos los programas desde el CD en la medida en que son utilizados).
- GNU/LinEx (<http://www.linex.org>), distribución basada en Debian desarrollada en España por la Junta de Extremadura con el propósito de contribuir al desarrollo de la Red Tecnológica Educativa y difundir el uso del software libre en la región para eliminar barreras económicas al acceso a la tecnología.

---

<sup>8</sup>Esta pregunta fue respondida por Alejandro Forero Cuervo, con base en información del MerFAQ.

<sup>9</sup>Esta pregunta fue respondida por Alejandro Forero Cuervo.

- Mandrake Linux (<http://www.mandrakelinux.com>), distribución de propósito general, basada en RedHat.
- Mklinux (<http://www.mklinux.org/>), distribución creada específicamente para correr en computadores Apple (con procesadores PowerPC).
- RedHat (<http://www.redhat.com>), distribución de propósito general.
- Slackware (<http://www.slackware.com>), distribución de propósito general.
- SuSE (<http://www.suse.com>), distribución de propósito general desarrollada en Alemania.
- Yellow Dog Linux (<http://www.yellowdoglinux.com/>), distribución creada específicamente para correr en computadores Apple (con procesadores PowerPC).

Existen muchas distribuciones de Linux, debido a su carácter de software libre. Si está interesado en conocer más distribuciones puede dirigirse al enlace The LWN.net Linux Distribution List (<http://lwn.net/Distributions/>), el cual ofrece un listado bastante completo, incluyendo información detallada sobre cada una de ellas y una clasificación especial para cada tipo de distribución.

### 2.8.3. ¿Qué distribución me recomiendan utilizar? <sup>10</sup>

Seleccionar la mejor distribución es un asunto enteramente subjetivo. Cada una de las principales distribuciones tienen tanto fortalezas como debilidades, admiradores como detractores.

Si nunca ha configurado GNU/Linux y no tiene mucha experiencia con computadores puede que le interese probar suerte con Knoppix, RedHat o Mandrake, que tienden a detectar el hardware y ofrecer un entorno bastante amigable a los novatos.

Si desea utilizar GNU/Linux si instalarlo en su disco duro (con las limitaciones que ésto implica) le recomendamos probar Knoppix o una de sus derivaciones, diseñadas específicamente con éste propósito.

Si, por otro lado, conoce bastante acerca del funcionamiento de los computadores y tiene interés en utilizar una distribución bastante flexible y poderosa, probablemente se incline por probar suerte con Debian, Slackware o Gentoo.

Si tiene un computador Apple, probablemente querrá utilizar Yellow Dog Linux, Mklinux, o Debian GNU/Linux.

Si está especialmente preocupado por la libertad del software en su instalación, seguramente querrá probar suerte con Debian.

---

<sup>10</sup>Esta pregunta fue respondida por Alejandro Forero Cuervo, con especial cuidado en ser objetivo.

#### 2.8.4. ¿Cómo puedo obtener una copia de una distribución? <sup>11</sup>

Hay varias maneras de conseguir un CD con una distribución de GNU/Linux.

Una forma, quizá la más sencilla, es bajar la imagen (ISO) de los CDs de la distribución de su elección y quemarla a un CD. Para hacerlo necesitará una conexión a internet con buen ancho de banda pues cada CD ocupa alrededor de 700 MB.

Un buen lugar para buscar imágenes ISO es LinuxISO.org (<http://www.linuxiso.org>), donde encontrará un listado de distribuciones.

Tenga en cuenta que, dado que la mayoría de distribuciones de GNU/Linux son basadas en software libre (o, cuando menos, software de libre redistribución), el proceso de descarga de una distribución de GNU/Linux a través de internet es completamente legal.

Otra opción para conseguir una distribución de GNU/Linux particular es encargarla a través de su sitio web oficial. Las ventajas de ésta opción son que probablemente recibirá documentación impresa acompañando sus CDs y, más importante, que estará contribuyendo económicamente al desarrollo de la distribución de su elección. Las desventajas son que, dependiendo de la distribución, el costo puede incrementarse un poco y que tendrá que esperar un tiempo antes de recibir sus CDs.

Una tercera opción similar a la anterior es encargar los CDs de alguna compañía especializada en distribuirlos. Puede consultar CheapBytes (<http://www.cheapbytes.com/>) o, en Colombia, CosasLibres.com (<http://www.cosaslibres.com>).

Una última opción es asistir a uno de los festivales de instalación de software libre organizados por Colibrí y por otras comunidades de usuarios en el país y comprarnos un CD con la distribución que le interese.

#### 2.9. ¿Qué tipo de hardware necesito para correr GNU/Linux? <sup>12</sup>

Linux corre sobre una gran cantidad de equipos. Dependiendo de sus necesidades necesitará más o menos memoria y espacio en disco.

Si tiene un computador con un 386 o un 486, 4 MB de memoria (RAM) y 200 MB de disco duro podrá instalarle GNU/Linux pero no podrá ejecutar ambientes gráficos muy sofisticados (como GNOME o KDE) ni aplicaciones con grandes requerimientos de recursos (como Mozilla o OpenOffice). Sin embargo, esta clase de equipos suelen portarse bastante bien si son configurados para prestar por red cierto tipo de servicios que requieren muy pocos recursos de procesador y memoria (como, por ejemplo, correr en ellos un servidor web que despliegue archivos estáticos).

Si no tiene espacio libre en disco duro podrá ejecutar GNU/Linux sin instalarlo utilizando un Live-CD. En este caso, en vista de que toda la información se almacena en la memoria, es

---

<sup>11</sup>Esta pregunta fue respondida por Alejandro Forero Cuervo con base en la información proveniente del MerFAQ.

<sup>12</sup>Respuesta escrita por Alejandro Forero Cuervo con base en el texto original de la MerFAQ y sugerencias de Jhair Tocancipá.

recomendable que su equipo tenga al menos 64 MB de memoria (aunque podrá correrlos en equipos con menos memoria).

Para una instalación normal de una distribución de propósito general un procesador Pentium o AMD K5 de 100 Mhz con 32 MB de RAM y 500 MB disponibles en disco duro serán suficientes (aunque, en un computador con estas características, el sistema responderá de manera relativamente lenta).

Otra opción para correr GNU/Linux es instalarla en un asistente electrónico personal (un computador de mano) como las Palm Pilot o los iPaq de Compaq.

GNU/Linux corre en una gran cantidad de procesadores. Por ejemplo, la versión 3.0 de Debian GNU/Linux soporta, en adición a los de Intel (386, 486, toda la familia de Pentium y los procesadores similares fabricados por Cyrix y AMD) procesadores Motorola 68k (incluidos en máquinas como los Atari, Amiga, NeXT y Apollo), Sun SPARC, Digital Alpha, Motorola/I. B. M. PowerPC, ARM, MIPS, HP-PA-RISC y S/390 de I. B. M., entre otros.

Linux soporta además máquinas con múltiples procesadores y suele manejar los dispositivos característicos de servidores (como tape backups o sistemas RAID) sin problemas.

Para encontrar información más detallada sobre el hardware soportado por GNU/Linux le recomendamos que consulte el Linux Hardware Compatibility HOWTO (<http://www.tldp.org/HOWTO/Hardware-HOWTO/>).

## 2.10. Virus en GNU/Linux

### 2.10.1. ¿Es cierto que GNU/Linux es inmune a virus?

Algunas personas afirman que GNU/Linux es inmune a virus y otros males similares muy comunes en otros sistemas por razones de diseño. GNU/Linux utiliza el esquema de permisos de Unix, lo que dificulta el daño que puede hacer al sistema la ejecución de código con propósitos negativos. En caso de correr código maligno, éste únicamente tendrá acceso al sistema con los permisos del usuario original, lo que en general le impedirá modificar los archivos correspondientes a las aplicaciones para instalarse.

Sin embargo, esto no garantiza la imposibilidad de construir virus que afecten máquinas Unix. En teoría un virus relativamente sofisticado no necesita modificar archivos del sistema: aún cuando corra con las restricciones de un usuario normal, esto debe ser suficiente para afectar su entorno (de manera que, por ejemplo, sea ejecutado cada cierto tiempo) y propagarse.

Otras personas opinan que la única razón por la que no han habido mayores casos de virus que afecten máquinas GNU/Linux es porque es mucho menos popular que las máquinas Windows. Al estar instalado en alrededor del 90% de los computadores de escritorio, Windows suele ser un sistema operativo mucho más atractivo como objetivo para las personas que escriben y diseminan virus.

Sea por razones teóricas (relacionadas con el sistema de permisos de Unix) o por razones prácticas (relacionadas con su baja penetración en computadores de escritorio), en la práctica los riesgos de contagio de virus en máquinas GNU/Linux son inexistentes.

### 2.10.2. ¿Existen antivirus para GNU/Linux? <sup>13</sup>

Sí.

Dado que GNU/Linux es bastante utilizado en servidores para intercambio de archivos (por ejemplo en servidores de correo electrónico o de archivos compartidos), es bastante útil contar con programas para remover virus que afecten a otros sistemas.

ClamAV, disponible en <http://www.clamav.net>, es un excelente antivirus de software libre, muy bien integrado con servidores de correo y otras herramientas. La lista de virus reconocidos es actualizada constantemente y está siempre disponible.

---

<sup>13</sup>Respuesta escrita por Alejandro Forero Cuervo con base en información enviada a la lista por Polkan García.

---

## Capítulo 3

# Administración del sistema

### 3.1. Bases de datos

#### 3.1.1. ¿Cómo configurar PostgreSQL?<sup>1</sup>

Para configurar PostgreSQL son necesarios los siguientes pasos:

1. *Inicializar la base de datos:* Ésto puede hacerse, ejecutando el comando `initdb` con los siguientes parámetros:
  - Path donde se encuentran los fuentes del `template1` para la creación de la base de datos. Para ubicar estos fuentes se hace una búsqueda del archivo `local1_template1.bki.source` que usualmente lo deja en el directorio `/usr/lib/pgsql`
  - Path donde va a residir la base de datos Postgres y todos sus archivos de configuración. Este usualmente se deja en el home directory del usuario `postgres` en el directorio `/var/lib/pgsql`.

Entonces se debe ejecutar el comando de la siguiente forma:

```
initdb --pglib=/usr/lib/pgsql --pgdata=/var/lib/pgsql
```

Éste se debe ejecutar como usuario `postgres`. Para ingresar al sistema como este usuario se debe ejecutar por línea de comandos la siguiente instrucción:

```
su -l postgres
```

siendo `su` (super user) en un principio.

---

<sup>1</sup>Respuesta del Grupo Merlinux. 16 de Octubre del 2001.

2. *Activar el servicio postmaster*: Una vez se inicializa la base de datos, hay que activar el servicio `postmaster` para aceptar conexiones. Si se desea aceptar las conexiones utilizando el protocolo TCP/IP, se debe ejecutar la siguiente instrucción:

```
postmaster -D/var/lib/pgsql -i &
```

En este path se encuentra la jerarquía de directorios `base/template1`, de donde toma archivos de configuración para inicio como `pg_class`. Si no se desea aceptar conexiones remotas, sencillamente se omite la bandera `-i` de la instrucción.

3. *Activar el demonio postgres*: Finalmente, para activar el demonio de PostgreSQL al iniciar la máquina, se debe ejecutar como `root` el comando `setup` y en la configuración de servicios se el comando `postgres` para que arranque al iniciar la máquina.

Para modificar la forma como ejecuta el `postmaster`, se puede editar el archivo `/etc/rc.0/init.d/postgres` para habilitar la opción de conexiones remotas TCP/IP se añade la bandera `-i` a la línea de la opción `start` que aparece así:

```
su -l postgres -c'/usr/bin/postmaster -S -D/var/lib/pgsql &
```

Para cambiar el puerto TCP en el que se recibirán conexiones (que por defecto es el 5432) se puede agregar la opción `-p<#puerto>`.

## 3.2. Servicios de Internet

### 3.2.1. ¿Cómo administrar Apache 1.3.x?<sup>2</sup>

#### Introducción

Éste documento es una guía para la rápida configuración del servidor de Web Apache para la versión 1.3.6 o superiores, ya que a partir de ésta versión hay ciertos cambios en los archivos de configuración de Apache; ésto no quiere decir que de pronto no sirva con versiones anteriores. Esta guía la estoy desarrollando porque la primera vez que me puse en el trabajo de configurar el Apache no me fué muy bien ya que los archivos de configuración son un poco complejos de entender al principio, por eso voy a tratar de explicar cada parte de éstos lo mejor que pueda para que no tengan que pasar por las que yo pasé alguna vez. Aparte de la configuración básica voy a cubrir temas como la configuración del Apache como Proxy y Servidores Virtuales (Virtual Hosting) a nivel de nombres y de IP's.

---

<sup>2</sup>Por Juan Luis Baptiste. [jbaptiste@merlinux.org](mailto:jbaptiste@merlinux.org). Julio 28 de 1999.

## Resumen

Un servidor de Web siempre consta de dos partes, los archivos de configuración, logs y módulos especiales, y los archivos y scripts de la página web. Cada uno de estos grupos de archivos está ubicado en directorios diferentes, para el caso de Apache por lo general los de configuración se encuentran en `/etc/httpd`, el cual se divide en tres subdirectorios, `conf` (archivos de configuración), `log` (archivos de auditoría) y `modules` (módulos extra para el Apache). Los archivos importantes son los que están en el subdirectorio `conf`, los cuales son tres, `httpd.conf`, `srn.conf` y `access.conf`. Más adelante voy a explicar para qué sirve cada uno y cómo se configuran. Los archivos de la página web (los que se van a publicar en Internet) se encuentran por lo general bajo el directorio `/home/httpd`, el cual se divide en tres subdirectorios, `html`, `cgi-bin` y `icons`. En el directorio `html` se encuentran, como su nombre lo dice, los documentos para publicar, que pueden tener extensión `.html`, `.shtml`, `.cgi` o `.php3` normalmente, aunque pueden ser más. En el directorio `cgi-bin` se colocan los scripts o CGI'S (Common Gateway Interface) que van a ser invocados a través de un cliente (browser) a nuestro servidor Apache. Estos scripts o cgi's pueden estar hechos en C/C++, Perl o Php3. Por último en el directorio `icons` se encuentran algunos íconos usados por Apache.

## Configuración del Apache

### Configuración Básica

**http.conf** Éste es el archivo de configuración principal del Apache. Aquí se encuentran los DSO'S (Dynamic Shared Objects) que está usando el Apache (los módulos) que pueden estar encadenados estáticamente o que pueden ser cargados al momento de arrancar el demonio de `http` (`httpd`). Para éste último caso hay que agregar una línea con la directiva `LoadModule` por cada módulo extra que se va a cargar, por ejemplo: `LoadModule foo_module libexec/mod_foo.so`. Apache ya trae todas estas líneas listas, esto sólo se usa en caso de que se agregue algún módulo extra. Para que funcione también hay que agregar una línea con la directiva `AddModule`, así: `AddModule foo_module.c`.

- `ServerType`: Puede ser `inetd` o `standalone`.
- `inetd`: Esta opción quiere decir que el Apache va a correr como un servicio de `inetd`; en éste caso hay que agregar una línea en el `/etc/inetd.conf` para que arranque.
- `standalone`: Esta opción quiere decir que va a correr como un demonio independiente y en este caso se agrega a los scripts de arranque del sistema (`/etc/rc.d/rc.3/`). Este es el que viene por default.
- `Port`: Puerto en el cual el `httpd` va a estar esperando por conexiones. El default es 80.
- `HostnameLookups`: Indica si se van a guardar en los registros de auditoría los nombres de los servidores y sus IP's o solo sus IP's. El default es `off` (solo IP's).

- `User` y `Group`: Indica el usuario y el grupo bajo los cuales va a correr el demonio `httpd`.
- `ServerAdmin`: Indica una dirección de correo electrónico a la cual se pueden reportar problemas que se presenten con el servidor de web. Esta es la dirección que va a aparecer con las páginas de error que muestra el Apache.
- `ServerRoot`: Indica el directorio en donde se guardan los archivos de configuración y registros de errores (logs). El default es `/etc/httpd`.
- `BindAddress`: Esta directiva se usa en conjunto con la de `VirtualHost` que va a ser explicada mas adelante. `BindAddress` se usa cuando se necesita que el `httpd` escuche por conexiones a múltiples direcciones IP's o a múltiples dominios. Sus posibles valores son:
  - Una dirección IP (cuando se quieren escuchar las conexiones a todas las IP's del servidor en el que está corriendo el `httpd`).
  - Un dominio completamente calificado en InterNIC.

Por default esta directiva viene comentada (#).

- `ErrorLog` indica el lugar y el nombre del archivo de registry de errores. Debajo de éste hay varias directivas respecto el nivel de auditoría, y formato de este archivo de registro, el registro de acceso y otros archivos.
- `PidFile`: Guarda el número del proceso del `httpd` que está corriendo. Se encuentra en `/var/run/httpd.pid`.
- `ServerName`: Sirve para indicar el nombre del servidor en caso de que el Apache no sea capaz por alguna razón de averiguarlo por su cuenta.
- `UseCanonicalName` (nuevo con versiones  $\geq 1.3$ ): Se usa cuando hay URL's que referencian al mismo servidor. Cuando está en `on`, Apache va usar las directivas `ServerName` y `Port` para construir un nombre canónico para el servidor. Si está en `off`, Apache va a construir la URL a partir del `hostname` y puerto que indique el cliente. Esta directiva agiliza el proceso de carga de una página por lo que no se tiene que esperar a consultar al cliente sino con los datos de la propia página se puede construir el enlace.
- `CacheNegotiatedDocs`: se usa para indicarle a los servidores proxy cuando usar el caché o no, Por default viene comentado, para que los servidores proxy no guarden en el caché el documento negociado, si se descomenta se producirá lo contrario y los servidores proxy podrán guardar el documento en el caché.
- `Timeout`: Es el número de segundos antes de que se envíe una señal de time out.
- `KeepAlive`: Para permitir conexiones persistentes (más de un pedido a la vez por conexión). Por default viene en `on`.
- `MaximunKeepAliveRequests`: Indica el número de pedidos (requests) máximos que se pueden hacer al servidor por cada conexión persistente. Coloque 0 para un número ilimitado de pedidos, lo recomendable es colocar un número alto. El default es 100.

- `KeepAliveTimeout`: Número de segundos que hay que esperar entre cada pedido.
- `StartServers`: Número de servidores que se deben lanzar en el momento que arranca el `httpd`. Estos servidores son los que se van a encargar de manejar las conexiones de los clientes. Un servidor por cliente, el default es 10.
- `MaxClients`: Indica el número máximo de clientes que pueden estar conectados simultáneamente al servidor. El default es 150.
- `MaxRequestsPerChild`: Es el número de pedidos que puede aceptar un servidor hijo antes de morir. Esto quiere decir que en el momento que un servidor complete el número de pedidos especificado en esta directiva se termina de correr y es lanzado otro nuevo en su lugar.
- *Directivas del Proxy*: Estas van a ser explicadas con detalle mas adelante.
- *Directivas de los servidores virtuales*: También se explicarán con detalle en otra sección.

**srm.conf** Este archivo se refiere a la configuración del directorio en el cual se encuentran los documentos y scripts para publicar.

- `DocumentRoot`: Indica el directorio en donde se encuentran las páginas. Por default viene en `/home/httpd/html`.
- `UserDir`: Indica el nombre del directorio en el cual los usuarios comunes pueden colocar sus archivos de html para que sean publicados en sus respectivos directorios bajo la dirección (e.g. <http://servidor.dominio.com/~usuario>).
- `DirectoryIndex`: Indica el orden en el que se van a buscar los archivos índice de una página en un directorio. Se pueden colocar varios documentos, por ejemplo: `DirectoryIndex index.html, index.shtml, index.cgi, index.php3, etc`. En otras palabras, cuando un navegador quiere entrar por ejemplo a `www.holas.com` el servidor de web tiene que mostrarle al navegador una página, entonces éste según esta directiva mira cual es el nombre que esté de primeras allá y busca ese documento en el directorio que indica `DocumentRoot` y lo envía, si no está busca el siguiente que esté ahí y así sucesivamente, si no encuentra ninguno muestra un error de `File not found`.
- `AddIcon` (puede ser también `AddIconByType` o `AddIconByEncoding`): Lo único que hace esto es relacionar un ícono con un tipo de archivo que es determinado por su extensión.
- `DefaultIcon`: Ícono que se va a mostrar con los archivos de tipo desconocido.
- `IndexIgnore`: Aquí se colocan los nombres de los archivos que no se quieren que sean listados.
- `AccessFileName`: Cuando se devuelve un documento al cliente, el servidor primero revisa una lista de acceso a ese directorio para ver si el cliente está autorizado o no a ver cualquier documento en ese directorio. El default es `.htaccess`.

- `DefaultType`: Tipo MIME que se muestra por defecto cuando el servidor se encuentra con un tipo MIME que no sabe manejar.
- `AddLanguage`: Se refiere a los posibles lenguajes en los que se podría mostrar una página, éstas sirven para negociar con el cliente en que idioma quiere que se le mande una página.
- `Alias`: Son abreviaciones a directorios para acortar las rutas, por ejemplo está `/icons/` que es un alias de `/home/httpd/icons/`. Sirve para no tener que poner por ejemplo `http://www.pagina.com/home/httpd/icons/unknown.gif`, se coloca `http://www.pagina.com/icons/unknown.gif`.
- `ScriptAlias`: Lo mismo que el anterior pero con el directorio de los CGI's.
- `AddType`: Sirve para agregar tipos MIME sin tener que editar el archivo `/etc/mime.types`. Por ejemplo: `AddType application/x-httpd-php3 .php3`.
- `AddHandler`: Sirve para identificar con otro nombre a cierto tipo de archivos por sus extensiones, por ejemplo: `AddHandler cgi-script .cgi`.
- `MetaDir`: Indica el directorio en donde se van a encontrar archivos de tipo meta. Este directorio tiene que estar oculto (el nombre tiene que empezar con un `."`). Estos archivos incluyen encabezados de HTTP adicionales para ser incluidos cuando se envía un documento.
- `MetaSuffix`: Extensión de los archivos tipo meta.
- `BrowserMatch`: Se usa para cuestiones de compatibilidad con los navegadores y las versiones de HTTP que usan (1.1/1.0).

**access.conf** Este es el archivo de configuración de acceso global de Apache. Aquí se definen todas las políticas de acceso, las cuales indican qué servicios se permiten, a quién se le permiten y bajo qué circunstancias. La configuración se hace por cada directorio al que se desea tener acceso.

Cada directorio se especifica con la directiva .

- `Options`: Indica las opciones que puede ejecutar un directorio. Estas pueden ser: `None`, `All`, `Indexes`, `Includes`, `FollowSymLinks`, `ExecCGI` o `MultiViews`.
- `AllowOverride`: Cuando el servidor se encuentra con un archivo `.htaccess` él necesita saber que directivas declaradas en ese archivo pueden saltarse información de acceso anterior. Sus posibles valores son:
  - `None`: En este caso el servidor no leerá el archivo.
  - `All`: Permitirá todas las directivas, que pueden ser una o todas las siguientes:

- **AuthConfig:** Permite el uso de directivas de control de acceso (`AuthDBMGroupFile`, `AuthDBMUserFile`, `AuthGroupFile`, `AuthName`, `AuthType`, `AuthUserFile`, `require`, etc.).
  - **FileInfo:** Permite usar las directivas de control de tipos de archivos (`AddEncoding`, `AddLanguage`, `AddType`, `DefaultType`, `ErrorDocument`, `LanguagePriority`, etc.).
  - **Indexes:** Permite el uso de directivas para el control de indexación de directorios (`AddDescription`, `AddIcon`, `AddIconByEncoding`, `AddIconByType`, `DefaultIcon`, `DirectoryIndex`, `FancyIndexing`, `HeaderName`, `IndexIgnore`, `IndexOptions`, `ReadmeName`, etc.).
  - **Limit:** Permite el uso de directivas para el control de acceso por servidores (`allow`, `deny` and `order`).
  - **Options:** Permite el uso de directivas para controlar el uso de algunas características especiales de los directorios (`Options` and `XbitHack`).
- **Order:** El orden de las directivas define el orden en que éstas son evaluadas:
- `allow, deny:` Las directivas `allow` son evaluadas primero que las `deny`, su estado inicial es `FORBBIDEN` (Acceso Prohibido).
  - `deny, allow:` Las directivas `deny` son evaluadas primero que las `allow`, su estado inicial es `OK`.
  - `Mutual-failure:` Solo los servidores que están en la lista de `allow` y no en la de `deny` pueden acceder al directorio.

**Configuración de Servidores Virtuales (Virtual Hosts)** Existen dos tipos de servidores virtuales que se van a explicar más adelante a fondo, los basados en direcciones ip's diferentes y los basados en nombres diferentes. Los basados en IP's diferentes necesitan una dirección IP por cada servidor virtual que se piensa hospedar en la máquina, mientras que los que están basados en nombres (dominios registrados en InterNIC) solo necesitan una ip que esté asignada a todos los nombres de servidores que se piensan hospedar. Al principio con el protocolo HTTP/1.0 solo era posible implementar el primer método, pero con la aparición de HTTP/1.1 se introdujo la posibilidad de que el servidor fuera capaz de distinguir a qué nombre se le estaba haciendo referencia.

**Servidores Virtuales basados en nombres** Las ventajas de usar servidores virtuales basados en nombres es que se puede tener un número ilimitado de éstos, son más fáciles de configurar y no se necesita ningún hardware o software adicional. La única desventaja es la falta de compatibilidad con navegadores viejos que no soportan la versión 1.1 del protocolo http.

Vamos a explicar este método con un ejemplo sencillo. Supongamos que tenemos los nombres `www.servidor.com` y `www.otroservidor.edu`, ambos apuntan a la dirección ip `111.222.333.444`. Entonces se agrega al final del `httpd.conf` las siguientes líneas:

```
<VirtualHost 111.222.333.444>
  ServerName www.servidor.com
  DocumentRoot /home/httpd/html/www.servidor.com
</VirtualHost>

<VirtualHost 111.222.333.444>
  ServerName www.otroservidor.edu
  DocumentRoot /home/httpd/html/www.otroservidor.edu
</VirtualHost>
```

Se pueden agregar más directivas que se necesiten dentro de la sección de `<VirtualHost>`. Para que esto funcione solo hay que estar seguro que `www.servidor.com` y `www.otroservidor.edu` apunten a la misma dirección IP `111.222.333.444`.

**Nota:** Cuando se usa la directiva `NameVirtualHost` el servidor solo se va a fijar en las secciones de `<VirtualHost>` y va a ignorar por completo el sitio principal. Si usa servidores virtuales tiene que colocar una entrada de `<VirtualHost>` por cada sitio que quiera publicar, incluido el principal.

También se pueden usar alias con los servidores en caso de que se necesite que alguno de ellos necesite ser accesado por nombres diferentes, por ejemplo que `www.servidor.com` también pueda ser accesado como `www2.servidor.com`. Por ejemplo:

```
ServerAlias www.servidor.comwww2.servidor.com
```

Se pueden usar wildcards (\* o ?) Con esta directiva (e.g. \*.servidor.com).

**Servidores Virtuales basados en IP's diferentes** Como el nombre lo indica, debe haber una IP por cada servidor que se quiere hospedar en la máquina; esto, ya sea teniendo varias interfaces físicas o siendo interfaces virtuales (IP aliasing).

Hay dos formas de configurar Apache para el uso de múltiples IP's, corriendo un demonio de http por cada servidor o corriendo uno solo que puede manejar a todos los servidores virtuales.

Para usar múltiples demonios lo único que hay que hacer es una instalación por cada servidor virtual y especificar la directiva `List` en a la ip en cada archivo `httpd.conf` de cada servidor, por ejemplo: `Listen 200.131.223.4:80`.

En el caso de un solo demonio de http se usan nuevamente las secciones de `<VirtualHost>`:

```
<VirtualHost 200.131.222.2>
  ServerAdmin webmaster@servidor.com
  DocumentRoot /home/httpd/html/www.servidor.com
  ServerName www.servidor.com
  ErrorLog /home/httpd/html/www.servidor.com/logs/error_log
  TransferLog /home/httpd/html/www.servidor.com/logs/access_log
```

```
</VirtualHost>

<VirtualHost 200.131.222.2>
  ServerAdmin webmaster@www.otroservidor.edu
  DocumentRoot /home/httpd/html/www.otroservidor.edu
  ServerName www.otroservidor.edu
  ErrorLog /home/httpd/html/www.otroservidor.edu/logs/error_log
  TransferLog /home/httpd/html/www.otroservidor.edu/logs/access_log
</VirtualHost>
```

Aquí la diferencia es que no se usa la directiva `NameVirtualHost` y por lo tanto el servidor principal si funciona, solo hay que agregar las entradas extra de los servidores virtuales.

### 3.2.2. ¿Cómo configurar Sendmail?<sup>3</sup>

#### Introducción

Con este documento voy a explicar como poner a funcionar a las carreras el Sendmail 8.9.3 para que reciba y envíe correo teniendo un dominio registrado en InterNIC, además de mostrar también como se configuran los alias de correo.

#### Resumen

La verdad es que Sendmail es un servidor de correo bastante difícil de configurar; precisamente por lo flexible que puede ser, se vuelve demasiado complejo pero las cosas importantes para poner a funcionar un servidor de correo con sus funciones más básicas se van a explicar acá. Los archivos que hay que tener en cuenta son: `/etc/HOSTNAME`, `/etc/sysconfig/network`, `/etc/sendmail.cf`, `/etc/domainalias` y `/etc/aliases`.

#### Configuración de Sendmail para enviar correo

Para poder enviar correo desde su maquina, de lo único que se tiene que cerciorar es que el demonio de sendmail esté corriendo, esto lo puede revisar con el siguiente comando:

```
ps x | grep sendmail
```

, esto le debe botar un mensaje parecido a éste:

```
366 ? S 0:00 sendmail: accepting connections on port 25
1986 ttyp0 S 0:00 grep sendmail
```

---

<sup>3</sup>Por Juan Luis Baptiste. [jbaptiste@merlinux.org](mailto:jbaptiste@merlinux.org). Septiembre 17 de 1999.

Si le aparece la primera línea, todo está bien, sendmail está funcionando correctamente. Si sólo le aparece la segunda quiere decir que el demonio de Sendmail no está corriendo, entonces tiene que ponerlo a correr. Puede hacerlo con el comando `/usr/sbin/sendmail bd`. Pero para que el demonio se cargue automáticamente cada vez que reinicie el sistema, lo puede hacer fácilmente entrando al programa de Setup (si está usando RedHat) desde el shell o desde una terminal, escogiendo System Services y seleccionando Sendmail para que sea cargado cada vez que inicie el sistema. Si todo está correcto puede probar el Sendmail mandando un correo desde una cuenta de su máquina a una cuenta externa, por ejemplo en hotmail.

### Configuración de Sendmail para recibir correo

Primero tiene que cerciorarse de que el hostname de la máquina esté bien configurado (esto es, que cuando entre al shell le aparezca algo así como

```
[root@maestro /root]#
```

), para esto, tiene que colocar en el archivo `/etc/HOSTNAME` el nombre de la máquina con su respectivo dominio (e.j. `maestro.dhis.org`). También revise que en el archivo `/etc/sysconfig/network` aparezcan las siguientes líneas:

```
NETWORKING=yes
FORWARD_IPV4=true #Sólo si va a usar su máquina como enrutador o
                  #puente entre dos redes, si no ponga false.
HOSTNAME=maestro.dhis.org
DOMAINNAME=dhis.org.
GATEWAY=xxx.yyy.zzz.www
GATEWAYDEV=eth0
```

Puede comprobar que todo está correcto con el comando

```
uname --nodename
```

, si todo está bien, debe mostrar el nombre de la maquina con su dominio (e.j. `Maeztro.dhis.org`).

Después de esto, busque en el archivo `/etc/sendmail.cf` una línea que dice `Cwlocalhost`, cámbiela por `Cwmaquina.dominio.com`, por ejemplo `Cwmaestro.dhis.org`. Listo, es todo, ahora reinicie el demonio de Sendmail, por ejemplo con

```
kill -HUP < /var/run/sendmail.pid
```

Ahora pruebe que todo esté funcionando correctamente, mande un correo desde una cuenta que no pertenezca a la máquina (e.j. Una en hotmail) a una cuenta en la maquina, mande un correo desde una cuenta en la maquina a otra en la maquina y revise que éste correo llegue de parte del dominio de su máquina (e.j. `usuario@maestro.dhis.org`) y no algo así como `<usuario@localhost.localdomain>`. Lo mismo para los correos que usted mande desde su máquina a una cuenta de afuera.

Para poder hacer que los usuarios puedan revisar su correo remotamente via pop3, descomente en el archivo `/etc/inetd.conf` una línea parecida a esta:

```
pop-3 stream tcp nowait root /usr/sbin/tcpd ipop3d
```

Ahora tiene un servidor de correo funcionando a las carreras. =).

### Configuración de los alias

Existen dos tipos de alias que usted puede definir en Sendmail, los alias a cuentas de usuarios y los alias a dominios completos. Los alias a cuentas de usuarios se usan cuando por ejemplo Usted necesita recibir el correo para diferentes personas pero no quiere crear una cuenta por cada persona, entonces crea una sola cuenta y crea alias a la cuenta que creó con los nombres de las otras personas que necesitaba.

En el caso de los alias de dominios, si de pronto usted necesita recibir el correo no sólo para un dominio sino para varios (por ejemplo que usted tenga un negocio de web hosting), sería conveniente poder hacerlo con la misma máquina y no tener una por cada dominio que quiere administrar. Para eso se crearon los alias de dominio en sendmail. Voy a empezar con los alias a usuarios para seguir con los de dominios.

**Alias a cuentas de correo** Sendmail para esta tarea utiliza generalmente un archivo llamado `aliases` que se encuentra en el directorio `/etc`, aunque como veremos más adelante pueden ser otros, tantos como alias se quieran crear.

En el archivo `/etc/aliases` se encuentran todos los alias del sistema, por ejemplo todas las cuentas de `bin`, `daemon`, `ingres`, `games`, `nobody`, `system`, `uucp`, etc, tienen un alias a la cuenta del `root`, esto quiere decir que cualquier correo dirigido a cualquiera de estas cuentas será redirigido a la cuenta del `root`.

El formato para definir los alias es el siguiente:

```
alias: cuenta
```

Por ejemplo, `merlinux: maestro`

Esto quiere decir que todos los correos enviados a `<merlinux@servidor.com>` van a ser reenviados a `<maestro@servidor.com>`. La otra forma es definiendo un archivo de alias por cada alias o grupo de alias que necesite tener, como por ejemplo `/etc/misalias`, en el que puedo colocar:

```
merlinux: maestro
grupomerlinux: maestro
otracuenta: otracuenta2
```

Lo único que hay que hacer si se usan archivos diferentes del `/etc/aliases` es indicar en el `sendmail.cf` debajo de la línea en donde se indica el nombre del archivo de alias (`/etc/aliases`), coloque una línea que arranque con "OA" (sin las comillas), por ejemplo `OA/path/a/archivo.aliases`. No se le olvide la extensión `.aliases`.

Después de colocar los alias en cualquiera de los archivos tiene que correr el comando

```
newaliases
```

para que se actualice la base datos de alias.

**Alias de dominios** Primero que todo, los alias que vaya a querer usar TIENEN que estar registrados ante InterNIC y tienen que apuntar a la máquina en donde los piensa alojar, sin esto, nada de lo que voy a explicar no servirá. Teniendo esto listo procederemos a lo importante. Lo primero que tiene que hacer es crear un archivo llamado `domainaliases` en el directorio `/etc`. Coloque ahí los alias a los dominios que necesite de la siguiente forma:

```
*@dominio1.com usuariol@dominioreal.com
*@dominio2.com otrousuario@dominioreal.com
webmaster@dominio1.com otrousuario@otroservidor.com
```

Después desde una terminal corra el siguiente comando para crear la base de datos de dominios:

```
makemap btree /etc/domainaliases.db < /etc/domainaliases
```

Ahora en el archivo de configuración `/etc/sendmail.cf` agregue una línea que comience con `Cw` por cada dominio virtual que vaya a usar:

```
Cwdominio1.com
Cwdominio2.com
```

Y por último agregue esta línea una sola vez:

```
Kdomainaliases btree /etc/domainaliases.db
```

Listo, todo debe estar listo. Para probar la nueva configuración de `sendmail` ejecute el siguiente comando:

```
sendmail -v -bv info@domain1.com
```

El destinatario final debe aparecer ahora.

## 3.3. Periféricos

### 3.3.1. ¿Cómo montar una unidad de diskette o cdrom?

Frecuentemente necesitamos transportar datos de una máquina a otra y no estamos en red por lo que la única solución es usar un diskette o un CDROM. El lío es que de pronto una de las dos máquinas funciona con Windows y la otra con Linux así que el problema en principio nos puede parecer imposible de solucionar. Sin embargo, la cuestión es increíblemente simple como vamos a ver.

Todo se reduce a “montar” la unidad de diskette adecuadamente. Con “montar” me refiero a la acción de asignar el dispositivo a un punto en la jerarquía del sistema o directorio para que pueda ser accedido. Fíjese que en los sistemas como DOS (o Windows y compañía) siempre se asigna una letra a cada unidad y es la forma de acceder a ellos, de modo que el disco duro generalmente se conoce como la unidad C y la unidad de CDROM como D, etc. En los sistemas Unix y en Linux no existen este tipo de restricciones que en realidad son un lastre de DOS. Lo que se hace es usar el comando `mount` para montar la unidad en un punto específico usando un tipo de sistema de archivos particular. Resulta que cuando se quiere montar un dispositivo, hay que verificar qué tipo de sistema de archivos usa para montarlo adecuadamente, pues de lo contrario los datos no se transmitirán en la forma correcta. Por ejemplo, el sistema de archivos de los primeros sistemas Windows era FAT 16<sup>4</sup> y el sistema de archivos de Windows 95 y familia es FAT 32, mientras que el sistema de archivos de Win NT es NTFS<sup>5</sup> y el sistema de archivos de Linux es el Extended 2<sup>6</sup>. Por esta razón es que en un disco duro se habla de particiones FAT32, ó swap, ó Linux, ó NT, etc. Entonces, si está usando una máquina cuyo sistema de archivos nativo es Extended 2 (es decir, Linux) y trata de montar un diskette formateado en Windows (es decir, cuyo sistema de archivos es FAT32), hay que especificar el tipo de sistema de archivos del dispositivo para que los datos sean transferidos correctamente porque de lo contrario, Linux no haría la conversión respectiva.

El comando `mount` tiene una opción `-t` que permite especificar el tipo de sistema de archivos que usa el dispositivo que se quiere montar. La sintaxis del comando es la siguiente:

```
mount -t <tipo de sistema de archivos> <dispositivo> <punto destino>
```

donde el parámetro `<tipo de sistema de archivos>`; es el tipo de sistema de archivos del dispositivo a montar. Linux soporta una gran variedad de tipos de sistema de archivos que reconoce con los siguientes nombres: `minix`, `xiafs`, `ext`, `ext2`, `msdos`, `umdos`, `vfat`, `proc`, `autofs`, `devpts`, `nfs`, `iso9660`, `smbfs`, `ncpfs`, `adfs`, `affs`, `coda`, `hfs`, `hpfs`, `ntfs`, `qnx4`, `romfs`, `ufs`, `sysv`, `xenix`, `coherent`. Entonces, si queremos especificar el sistema de archivos FAT32, debemos usar el nombre `vfat` y si queremos especificar FAT16 entonces usamos el nombre `msdos`, etc. El tipo de sistema de archivos que usan los CDROMs es `iso9660`. Cuando se usa el comando `mount` sin la opción `-t`, el sistema trata de montar el dispositivo usando el tipo de sistema de archivos que más se ajuste al formato que tiene, osea que intenta

---

<sup>4</sup>File Allocation Table 16 bits

<sup>5</sup>NT File System

<sup>6</sup>JTo: No es el único sistema de archivos de Linux! Corregir!

adivinar cómo fue formateado el dispositivo. Aunque esta utilidad es bien cómoda, el problema al usar `mount` así es que no siempre se montan los dispositivos como se debe. La prueba más sencilla es tratar de montar sobre Linux un diskette formateado en Windows sin la opción `-t` que resulta en que el diskette queda montado como `msdos` y por lo tanto estarán vigentes las restricciones de DOS como recortar los nombres largos de archivos a 8 caracteres y usar 3 caracteres para la extensión. Si se usa la opción `-t` con el valor `vfat`, entonces los archivos se leerán y se escribirán en la forma correcta.

Ahora, respecto al parámetro `<dispositivo>` que es el nombre del dispositivo, se debe especificar el nombre que el sistema le ha asignado. En los sistemas Unix y Linux hay un directorio llamado `/dev` en el que se encuentran las referencias a los dispositivos que el sistema reconoce. Una de estas referencias es `fd0`<sup>7</sup> que es el nombre con el que el sistema reconoce a la primera unidad de diskettes. Si hay una segunda unidad de diskettes, se reconocerá con el nombre `fd1` y así sucesivamente. Entonces, cuando vamos a montar un diskette en la primera unidad de diskettes, el valor del parámetro `<dispositivo>` será `/dev/fd0`. Por otro lado, cuando se quiere montar un CDROM, Linux generalmente reconoce a la unidad de CDROM como `/dev/cdrom` que en realidad es un enlace a la referencia real al dispositivo (usualmente `/dev/hdd`).

Finalmente, hay que asignar un valor coherente al parámetro `<punto destino>` que especifica el nombre del directorio o punto de la jerarquía del sistema en donde se va a montar el dispositivo. Dicho directorio debe existir desde antes del procedimiento porque `mount` no lo puede crear. Así que lo primero es crear ese punto sobre el que vamos a montar con la directiva `mkdir` que permite crear un directorio. Por ejemplo, (suponiendo que el usuario tiene permisos para escribir en la raíz del sistema) si previamente hemos creado el punto con `mkdir /mi_diskette` entonces podemos montar un diskette formateado en Windows así:

```
mount -t vfat /dev/fd0 /mi_diskette
```

Si accedemos al directorio `/mi_diskette`, encontraremos los datos que habían sido almacenados en el diskette en el formato especificado y todo lo que se escriba en este directorio, será transferido en formato FAT32. Ahora, si lo que quiere es montar un CDROM, primero creamos el punto en el que se va a montar por ejemplo con `mkdir /cdrom` y luego se monta con:

```
mount /dev/cdrom /cdrom
```

Fíjese que aquí no se ha usado la opción `-t`. Esto es porque por defecto, los CDROMs se montan con el tipo `iso9660` y no hay necesidad de especificarlo. Tenga en cuenta que no se pueden escribir datos sobre un CDROM y al montarlo, el sistema avisa que lo montó exclusivamente para lectura :)

El último paso es muy importante. Cuando termine de usar el dispositivo, es NECESARIO “desmontarlo”. Esto se hace porque el sistema mantiene referencias tanto al punto en el que se montó el dispositivo como a los archivos que están contenidos allí y si no se desmontara, tales referencias permanecerían activas pero inconsistentes. Para desmontar un dispositivo, se

---

<sup>7</sup>floppy disk 0

usa el comando `umount` al que se le puede pasar como parámetro el nombre del punto sobre el que está montado el dispositivo. Entonces, cuando termine de usar el diskette en el ejemplo anterior, se desmonta así:

```
umount /mi_diskette
```

y cuando termine de usar el CDROM, lo puede desmontar con:

```
umount /cdrom
```

Para poder sacar el CDROM de la unidad, use la directiva `eject` que expulsa la bandeja de la unidad de CDROM.

### 3.3.2. ¿Cómo configurar una impresora PPA (aka. winprinter)?<sup>8,9</sup>

Este proceso de instalación fue probado en la distribución Linux Mandrake versión 7.0.

#### Conseguir los paquetes en Internet

Dos paquetes RPM son distribuidos por el proyecto `pnm2ppa` para la instalación en sistemas Linux Red Hat, y sobre distribuciones “compatibles” como Mandrake. Estos se pueden encontrar en <http://www.sourceforge.net>.

- `pnm2ppa`: `pnm2ppa-0.8.9pre1-1.i386.rpm`
- `rhs-printfilters`:
  - `rhs-printfilters-1.57-4ppa1.i386.rpm` para Red Hat 6.1
  - `rhs-printfilters-1.58-4ppa1mdk.i386.rpm` para Mandrake 7.0

El segundo RPM (`rhs-printfilters`) integra los drivers PPA con los filtros de impresión Red Hat, usados por distribuciones Red Hat y “compatibles”.<sup>10</sup>

*Todos los siguiente pasos deben ser realizados como usuario root.*

---

<sup>8</sup>Respuesta por Herman Fabián Sandoval. Revisada por Offray Vladimir Luna Cárdenas.

<sup>9</sup>Específico para las siguientes impresoras: HP710, HP720, HP820 y HP1000.

<sup>10</sup>No instalar el RPM Red Hat sobre distribuciones Mandrake. Para Mandrake utilizar el paquete `*mdk.i386.rpm`.

## Instalar los paquetes

*Antes de instalar los paquetes es necesario asegurarse de desinstalar cualquier impresora PPA que se tenga actualmente instalada, para ello se debe utilizar la herramienta PrinterTool (“printtool” desde línea de comandos), seleccionar la impresora y presionar el botón eliminar.*

1. Utilizando algún programa de administración de paquetes (RpmDrake, gnoRPM, rpm, kpackage, etc.), instalar el paquete pnm2ppa-0.8.9pre1-1.i386.rpm. Si al momento de la instalación indica que hay conflictos con otros paquetes ignorarlos y continuar la instalación.

*Si se presentan problemas que no se puedan solucionar con la actual guía, después de haber instalado el paquete mencionado es posible acceder al archivo `/usr/doc/pnm2ppa-0.8.9pre1/INSTALL.REDHAT.txt`, el cual contiene información más completa.*

2. Desde consola ejecutar la utilidad `detect_ppa` la cual se encarga de detectar la impresora PPA. Si esta es encontrada se mostrará un mensaje como este:

```
*** Script for testing installation of PPA printers on Linux (2.2.x kernel)
Printer found at parallel port 0 (usually /dev/lp0 )
IEEE-1284 autoprobe reports:
CLASS:PRINTER;
MODEL:DESKJET 710C;
MANUFACTURER:HEWLETT-PACKARD;
DESCRIPTION:Hewlett-Packard DeskJet 710 Series;
COMMAND SET:SCP,VLINK;
Supported PPA printers found on parallel ports:
type=710, port=0;
printers on parallel port are by default accessed by printing to /dev/lp0
(unless this was explicitly changed when loading the parport module)
```

3. Enviar una página de prueba a la impresora. Para ello utilizar el comando de consola `test_ppa` el cual pedirá algunos datos acerca de la impresora, el tipo de papel, el puerto al cual está conectada, el controlador que desea probar (blanco y negro o color) para luego imprimir la página de prueba de impresión (color o blanco y negro) y la página de prueba de calibración. Con esto se sabrá que la impresora ppa está lista para imprimir archivos en formato PostScript.
4. Utilizando algún programa de administración de paquetes (RpmDrake, gnoRPM, rpm, kpackage, etc.), desinstalar el paquete `printtool` (ubicado en “Applications - Publishing” dentro del árbol de paquetes) y el paquete de filtros de impresión que se encuentra en la misma ubicación.
5. Utilizando algún programa de administración de paquetes (RpmDrake, gnoRPM, rpm, kpackage, etc.), instalar el paquete `rhs-printfilters-1.58-4ppa1mdk.i386.rpm` para Mandrake 7.0. Si al momento de la instalación indica que hay conflictos con otros paquetes ignorarlos y continuar la instalación.

6. Utilizando algún programa de administración de paquetes (`RpmDrake`, `gnorPM`, `rpm`, `kpackage`, etc.), instalar el paquete `printtool`, esto permitirá que después de instalar el paquete de filtros, en el paso anterior, la herramienta `printerTool` reconozca los nuevos filtros para programas cuya salida no es PostScript, por ejemplo editores de texto livianos como el `kwrite`.
7. Ejecutar la herramienta `PrinterTool` (escribiendo `printtool` en consola).
  - Presionar el botón `Add` para agregar la impresora.
  - Seleccionar `Local Printer`.
  - Presionar el botón `Select`.
  - En la parte izquierda (`Printer Type`) de la ventana que aparece, mirar si la impresora está seleccionada (normalmente `PrinterTool` hace autodetección), si no lo está es necesario seleccionar la impresora `PPA`.
  - En la parte derecha `Color Depth/Uniprint Mode`, seleccionar la calidad y el tipo de impresión a realizar (Recomendadas: `Color Econofast Mode` y `Color High Quality`, tal vez sea necesario hablar con los usuarios para determinar la selección correcta).
  - En la parte derecha seleccionar la opción “`Send EOF after job to eject page`” y asegurarse de que la opción “`Fast Text Printing`” NO esté seleccionada, si es necesario (por lo general lo es) deseleccionarla.
  - Presionar `Ok`.
  - Presionar `Ok`, nuevamente.
  - Seleccionar la impresora y enviar una página de prueba. (Opcional)
8. Imprimir un documento desde un editor de texto básico. Si imprime, pasar a configurar la impresión en `StarOffice`.

### Configurar la Impresora para StarOffice:

1. Entrar como `root`. Ejecutar el archivo `/bin/spadmin`, donde es el directorio donde instaló los archivos compartidos del programa `Star Office`, por ejemplo `/usr/share/soffice`  
*Si no se accede como root no se podrán realizar las siguientes actividades.*
2. En el cuadro de diálogo (Instalación de Impresora) que aparece, seleccionar la impresora instalada: `default_queue=lpr`.
3. Presionar el botón `Conectar...`
4. En el cuadro de diálogo `Conectar` seleccionar `lp0=lpr -P lp0` (o en su defecto el que tenga la indicación `lp=...`) y presionar `Aceptar`.
5. Seleccionar `Configurar...`
6. En este cuadro seleccionar todas las opciones por defecto de configuración de la impresora (para evitar que cada usuario deba realizarlas cada vez que vaya a imprimir).

7. Por cuestiones de estética y facilidad para el usuario seleccionar el botón Cambiar Nombre y colocar el nombre de la impresora (por ejemplo: Impresora HP DeskJet 710C) y aceptar.
8. Seleccionar el botón Página de Prueba . . . . Se debe imprimir la página de prueba correctamente (esta imprime un texto, unas barras en color y finalmente un caracol en líneas, en la misma hoja).
9. Si hay varias impresoras configuradas es necesario Predeterminar la impresora, seleccionándola y luego presionando el botón Predeterminar.
10. Cerrar la ventana de Instalación de Impresora.

### 3.3.3. ¿Cómo configuro un Winmodem?

#### ¿Cómo puedo saber si el modem PCI que tengo actualmente es soportado en Linux?<sup>11</sup>

En primer lugar, es necesario averiguar si existe un controlador (*driver*) para el *chipset* del modem correspondiente. El *chipset* es un grupo de microchips diseñados para realizar una o varias funciones relacionadas, en este caso, con el proceso de modulación y demodulación de la señal. Para lograr esto, la información que se encuentra en el sitio Linmodems.org (<http://linmodems.org>) puede ayudar bastante, por cuanto ofrece una base de datos bastante completa sobre modems soportados en Linux.

Una primera aproximación a la búsqueda de información sobre el *chipset* es el uso del comando `lspci -vv`. Sin embargo, es posible que la búsqueda del controlador sea algo tediosa, partiendo de que sólo obtendremos el nombre del *chipset*. En Linmodems.org se encuentra información sobre el script `scanModem` (<http://linmodems.technion.ac.il/packages/scanModem.gz>), que al ser ejecutado nos brinda información sobre el *chipset* y el *driver* correspondiente, incluida su localización en Internet, si este efectivamente existe. Muchos modems de diferentes marcas utilizan los mismos *chipsets*, así que es probable que se tenga éxito. Si se da este caso, se debe proceder a instalar el driver y seguir las recomendaciones brindadas por sus desarrolladores. Este paso y los siguientes se deben realizar con privilegios de `root`.

Generalmente el proceso de instalación se encarga de configurar el sistema para que cargue los módulos del controlador automáticamente, al ser usado por primera vez el modem en una sesión de trabajo. Si no es así, sería necesario utilizar el comando:

```
modprobe modulo_correspondiente
```

Es importante verificar qué archivo de dispositivo corresponde al modem. Comúnmente es de la forma `/dev/tty??`. Esta información usualmente se da en el proceso de instalación y generalmente se crea de forma automática un enlace `/dev/modem` al archivo respectivo. Si no es así, es necesario hacerlo:

---

<sup>11</sup>Respuesta de Sergio Roa Ovalle [s.roa@computer.org](mailto:s.roa@computer.org), publicada originalmente en el FAQ del grupo LUGUNAR <http://www.lugunar.net>

```
ln -s
/dev/tty?? /dev/modem
```

de tal forma que podemos configurar fácilmente programas como `kppp` y `wvdial`, usados para la conexión a Internet.

### PCTel mini-COMO Version 1.1<sup>12</sup>

Esta sección describe como instalar un modem PCTel en Linux.

1. **Instalación.** Todo lo siguiente lo haremos como `root`, el superusuario de Linux.

a) *Descompresión:* Descomprimimos el archivo usando el comando `tar`, de esta forma:

```
tar xvzf pctel-X.X.X.tar.gz
```

b) *Ir a:* Accedamos al directorio descomprimido, llamado `pctel-X.X.X`, haciendo `cd pctel-X.X.X`.

c) *Encontrando su chipset:* Para encontrar el tipo de modem PCTel que usted posee, utilizaremos el comando `lspci` y la opción `-v`,

```
lspci -v
```

, o bien utilizando `cat` de esta forma:

```
cat /proc/pci
```

, lo cual muestra algo similar a:

```
00:01.6 Modem: Silicon Integrated Systems [SiS] 56k Winmodem
(Smart Link HAMR5600 compatible) (rev a0) (prog-if 00 [Generic])
Subsystem: Silicon Integrated Systems [SiS] 56k Winmodem (SmartLi
Flags: bus master, medium devsel, latency 64, IRQ 3
I/O ports at d800 [size=256]
I/O ports at d400 [size=128]
Capabilities: [48] Power Management version 2
```

d) *Compilando:* Al saber el tipo de modem PCTel que tenemos, procederemos a compilar nuestro módulo así:

```
./configure --with-hal=list
```

muestra el tipo de modems que son soportados por este software. Mostraría algo así:

---

<sup>12</sup>Respuesta de Juan F. A. Saldarriaga. [nebiros@gigax.org](mailto:nebiros@gigax.org). Grupo de Implementación GNU y Arquitecturas \*X (GiGAX) - <http://www.gigax.org>. 15 de Abril del 2003.

```
--with-hal=hal
Select one of: pct789, cm8738, i8xx, sis, via686a
```

Dependiendo de nuestro tipo de modem escogeríamos una de las opciones mostradas, en el caso mas común, sería tener un modem HSP56, entonces escogeríamos la opción `sis`, así:

```
./configure --with-hal=sis
```

`make` compila el módulo según el tipo de modem que tengamos. `make install` instala los módulos que acabamos de compilar.

- e) *Instalando los módulos:* Hacemos `insmod pctel` y luego `insmod ptserial`, así cargamos los módulos ya instalados.
- f) *Probando los módulos:* Checkeamos con `tail` en los logs del sistema si los módulos fueron cargados exitosamente, `tail /var/log/messages` mostrará algo como esto:

```
May 16 23:28:17 suba kernel: PCTel initialization. Country code i
May 16 23:28:17 suba kernel: PCTel device[00:11.0](0x88) found "P
May 16 23:28:17 suba kernel: PCTel driver version 0.9.6 [5.05c-4.
May 16 23:28:17 suba kernel: PCTel driver built on [Linux 2.4.8 i
May 16 23:28:17 suba kernel: ttyS15 at 0xe800 (irq = 9) is a PCTe
```

- g) *Iniciando los módulos automáticamente:* En `/etc/rc.d/rc.local` agregamos las siguientes líneas:

```
insmod pctel
insmod ptserial
```

Así cada vez que entremos a Linux, los módulos del PCTel se cargaran por defecto.

## 2. Problemas.

- a) *Modem no responde:*

Si al utilizar `wvdial`, `kppp`, u otro programa de marcación telefónica, nos dice que el modem no responde, es posible que tengamos un conflicto con nuestra tarjeta de sonido, este problema se presenta muy comunmente en tarjetas de sonido *on-board* o integradas, entre ellas las famosas SiS 7018 y últimas.

El problema es que muchas distribuciones cargan los módulos de estas tarjetas de sonido por defecto, y al estar cargados estos módulos de sonido el modem no puede funcionar. Para arreglar esto muchas veces tendremos que recompilar nuestro kernel, o editar el archivo `/etc/modules.conf` y desabilitar la línea donde carga nuestra tarjeta de sonido.

- *Recompilando nuestro kernel*

Recompilaremos nuestro kernel de ésta forma, desabilitando ciertas opciones en el kernel para que los módulos de nuestra tarjeta de sonido SiS 70XX no se carguen automáticamente:

```
cd /usr/src/linux
make menuconfig
```

Escogemos Sound --->

Seleccionamos [\*] Trident 4DWave DX/NX, SiS 7018 or ALi 5451 PCI Audio Core y con la barra espaciadora cambiamos de la opción [\*] a la [M].

Para salir seleccionamos Exit ---> y guardamos nuestra configuración.

Luego:

```
make dep
make bzImage
make modules && make modules_install
make install
```

Acá instalaremos la nueva imagen del kernel, ésta imagen va por defecto a /, pero muchas distribuciones manejan su imagen en /boot, por lo que necesitaras mover ésta imagen /vmlinuz a /boot, así: `mv /vmlinuz /boot`

- *Editando el archivo /etc/modules.conf*

Si al entrar a la configuración del kernel, y vemos que la opción mencionada arriba está bien [M], entonces necesitaremos editar el archivo /etc/modules.conf, buscando una línea como ésta:

```
alias sound-slot-0 trident
```

La comentaremos, poniendo un # al inicio de la misma:

```
# alias sound-slot-0 trident
```

### 3. Fuentes.

Fuentes utilizadas para este mini-COMO: <http://linmodems.technion.ac.il/pctel-linux/><sup>13</sup>

### FAQ sobre Modems On-Board con Chipset PC-TEL Version 0.1<sup>14</sup>

1. *Introducción:* Decidí hacer ésta FAQ ya que navegando en la red pude encontrar algo que les sera útil a todos los usuarios de motherboards PC-CHIPS con modems integrados: un driver para hacer andar el modem en Linux. Ojo, lo que si quiero aclarar es que si bien en mi sistema el driver perfectamente, no puedo garantizar lo mismo en otros sistemas. Hago esta FAQ basandome solo en mi experiencia. Igualmente me ofrezco a responder cualquier duda que tengan sobre la instalación.
2. *Equipo en el que funciona el driver:* El driver me funciona perfectamente en mi motherboard PC-CHIPS M598LMR, que posee el chipset SiS 530. El chip del modem es el CMI 8738, que en realidad no solo es de modem sino tambien de sonido.
3. *¿De donde bajo el driver?* Yo lo encontré en [http://www.ecs.com.tw/download/dw\\_spec.asp?product\\_id=200](http://www.ecs.com.tw/download/dw_spec.asp?product_id=200)<sup>15</sup> pero por las dudas en los proximos dias lo voy a poner yo en algun otro sitio.

<sup>13</sup>Link parece estar muerto, favor actualizar!

<sup>14</sup>Respuesta de Hernan Pablo Alvarez. [hernanalvarez@arnet.com.ar](mailto:hernanalvarez@arnet.com.ar). 25/1/2000.

<sup>15</sup>JTo: Enlace muerto en el original. Cambiado por uno que funciona.

4. *¿Cómo instalo el driver?* El driver viene comprimido en un archivo auto-extraíble para Windows. Voy a detallar paso a paso como instalarlo:
  - a) Descomprimir el driver en un directorio de Windows, digamos C:\modem.
  - b) Bootear en Linux y montar dicha partición.
  - c) Dentro del directorio que creamos, encontramos el directorio `linux`. Entramos en ese directorio, y encontramos 3 directorios mas: `include`, `lib` y `src`.
  - d) Entramos en el directorio `src`, y dentro de ese directorio encontramos otro mas: `serial`. Entramos en ese también.
  - e) Ejecutamos `make`. Si tenemos el `gcc` bien instalado, en segundos tendríamos que tener el driver compilado en el directorio `lib` que les mencionaba antes.
  - f) Nos movemos hacia el directorio `lib`, y ejecutamos

```
insmod pctel.o
```
  - g) Si el driver funciona, te va a aparecer el siguiente mensaje:

```
PCTel Device (61) found irq = 11 base = 0x0dc00
```

*(ojo, en tu PC puede ser distinto, en la mía el modem esta en el IRQ 11).*
  - h) Para completar la instalación, hay que ejecutar el comando `mknod /dev/ttyS15 c 62 79`. Eso va a crear el dispositivo `/dev/ttyS15` que de ahora en adelante va a ser el modem.
  - i) Recomiendo copiar el archivo `pctel.o` a algun otro directorio, y cada vez que se quiera usar el modem ejecutar `insmod pctel.o`.
5. *Para terminar:* Puede leer también el archivo `readme.txt` incluido con el driver, en éste archivo se explica muy bien como instalar el driver.

### 3.3.4. ¿Cómo configuro el mapa de caracteres de mi teclado?

Son muy conocidos los típicos problemas debido a la mala configuración del teclado y se pueden llegar a convertir en un verdadero karma cuando no encontramos el símbolo adecuado, o cuando al presionar una tecla simplemente sale otra completamente distinta en la pantalla. Si sumamos a esto las diferentes distribuciones de teclados que existen (Inglés, Inglés internacional, Español, etc.) donde todas las teclas tienen diferentes posiciones.

O si su problema es simplemente cambiar la acción de las teclas, lo que voy a describir le va a ser útil.

Linux nos ofrece una potente herramienta de configuración - `xmodmap` que convierte los eventos del teclado en símbolos específicos. De aquí nos podemos valer para poder poner en la salida estándar lo que deseamos.

¿Cómo funciona? Bueno la idea es leer una lista de expresiones dadas y colocar la correcta antes de ejecutarla; esta lista está compuesta por los códigos de cada una de las teclas (`keycode`), el

número del código y el código o nombre del símbolo (`KEYSYMNAME`) el cual puede estar dado en forma hexa, decimal, octal o como constante.

El programa básicamente lee ya sea desde el mismo shell o de un archivo de texto plano las cadenas dadas y las interpreta; por ejemplo si deseamos cambiar el `DELETE` por el `BackSpace` para lograr un mayor acercamiento al usuario, digitamos

```
xmodmap -e "keysym BackSpace=Delete"
```

Podemos cargar todo lo que necesitamos desde un archivo en forma ordenada y luego ejecutarlo a nuestra orden o cargarlo ya sea en un `.init` si se es `root` o simplemente colocarlo en su perfil de usuario (`.profile`) lo cual ahorra mucho tiempo.

Entonces veamos:

Para empezar a realizar cualquier cambio debemos definir lo que el programa llama los modificadores. Estos no son mas que `CapsLock`, las teclas de `Ctrl`, las teclas `Alt`, `Tab`, es decir teclas de función. Entonces como un primer paso debemos limpiar o borrar la función de estos modifiers y en un segundo paso cargamos la lista de códigos: `keycode NUMBER = KEYSYMNAME...`

Por ejemplo

```
keycode 26=e
```

Y con esto ya esta definido que el 26 es e minúscula, y terminamos definiendo de nuevo nuestros modifiers.

Cómo limpiar los modifiers? Bueno los modificadores son `shift`, `lock`, `control`, `mod1`, `mod2`, ..., `mod5`; los cuales hacen referencia comúnmente a:

- `shift`: a las mayúsculas izquierda y derecha.
- `lock`: a la acción de asegurar mayúsculas es decir `CapsLock`.
- `control`: a ambas teclas `Ctrl` (izquierda y derecha).

Para definir izquierda y derecha se hace de la siguiente manera:

- `Shift_R`: shift derecho.
- `Shift_L`: shift izquierdo.
- `Control_R`: Ctrl derecho.
- `Control_L`: Ctrl izquierdo.

Las palabras claves reservadas son `clear`, para limpiar, `add` para agregar y `remove` para quitar, y para realizar comentarios antecedemos un `!`; luego para limpiar los modifiers hacemos:

```
clear shift
clear lock
clear control
clear mod1
clear mod2
clear mod3
clear mod4
clear mod5
```

Luego cargamos el mapa para cada tecla teniendo en cuenta ya sea su código hexa, octal, decimal o usamos algunas constantes como:

```
Esc = Escape
! i = exclam exclamdown
@ = at
# = numbersign
$ = dollar
% = percent
^ = asciicircum
& = ampersand
* = asterisk
( = parenleft
) = parenright
- _ = minus underscore
= = equal plus
BackSpace = BackSpace
Tab = Tab Tab
[ = bracketleft braceleft
] = bracketright braceright
Return = Return
Ctrl izquierdo = Control_L
```

Para las teclas del teclado numérico se añade el prefijo KP\_, es decir KP\_1 se refiere al número 1 del teclado numérico (ubicado a su mano derecha). Aquí están las otras:

```
keycode 80 = KP_8
keycode 81 = KP_9
keycode 82 = KP_Subtract
keycode 83 = KP_4
keycode 84 = KP_5
keycode 85 = KP_6
keycode 86 = KP_Add
keycode 87 = KP_1
keycode 88 = KP_2
```

```
keycode 89 = KP_3
keycode 90 = KP_0
```

Ahora algo más interesante es colocar tildes y eñes, eso se logra añadiendo al código de las vocales la siguiente línea, por ejemplo lo que originalmente se veía solo como:

```
keycode 38 = a A
```

agregando la línea, queda:

```
keycode 38 = a A aacute Aacute
```

Y para las eñes:

```
keycode 57 = n N ntilde Ntilde
```

Ahora si se colocan de nuevo los modificadores, con la instrucción add

```
add shift = Shift_L Shift_R
add lock = Caps_Lock
add control = Control_L Control_R
add mod1 = Alt_L Alt_R
```

Si no funciona el NumLock, agregue la siguiente línea

```
add mod2 = Num_Lock
```

Y Listo!, ahora puede empezar a crear su mapa. Aquí abajo adiciono el que yo uso, que con solo dar Alt derecho y vocal coloca tildes o Alt derecho y n coloca ñ.

```
! Key mapping for XFree86.
!
! With this file you can get tildes and acutes with the english layout.
! Just hit AltGr + .
!
! Load it using xmodmap.
!
!
! First, clear the modifiers
!
clear shift
```

```
clear lock
clear control
clear mod1
clear mod2
clear mod3
clear mod4
clear mod5

!
! Set the mapping for each key
!
keycode 9 = Escape Escape
keycode 10 = 1 exclam exclamdown
keycode 11 = 2 at
keycode 12 = 3 numbersign
keycode 13 = 4 dollar
keycode 14 = 5 percent
keycode 15 = 6 asciicircum
keycode 16 = 7 ampersand
keycode 17 = 8 asterisk
keycode 18 = 9 parenleft
keycode 19 = 0 parenright
keycode 20 = minus underscore
keycode 21 = equal plus
keycode 22 = BackSpace
keycode 23 = Tab Tab
keycode 24 = q
keycode 25 = w
keycode 26 = e E eacute Eacute
keycode 27 = r
keycode 28 = t
keycode 29 = y
keycode 30 = u U uacute Uacute
keycode 31 = i I iacute Iacute
keycode 32 = o O oacute Oacute
keycode 33 = p
!keycode 26 = dead_grave asciicircum bracketleft
keycode 34 = bracketleft braceleft
keycode 35 = bracketright braceright
keycode 36 = Return
keycode 37 = Control_L
keycode 38 = a A aacute Aacute
keycode 39 = s
keycode 40 = d
keycode 41 = f
keycode 42 = g
```

```
keycode 43 = h
keycode 44 = j
keycode 45 = k
keycode 46 = l
keycode 47 = semicolon colon
!keycode 48 = dead_acute dead_diaeresis
!keycode 49 = dead_grave dead_tilde
keycode 48 = apostrophe quotedbl
keycode 49 = grave asciitilde
keycode 50 = Shift_L
keycode 51 = backslash bar
keycode 52 = z
keycode 53 = x
keycode 54 = c
keycode 55 = v
keycode 56 = b
keycode 57 = n N ntilde Ntilde
keycode 58 = m
keycode 59 = comma less
keycode 60 = period greater
keycode 61 = slash question questiondown
keycode 62 = Shift_R
keycode 63 = KP_Multiply
!keycode 64 = Alt_L Meta_L
keycode 64 = Alt_L
keycode 65 = space space
keycode 66 = Caps_Lock
keycode 67 = F1 F11
keycode 68 = F2 F12
keycode 69 = F3 F13
keycode 70 = F4 F14
keycode 71 = F5 F15
keycode 72 = F6 F16
keycode 73 = F7 F17
keycode 74 = F8 F18
keycode 75 = F9 F19
keycode 76 = F10 F20
keycode 77 = Num_Lock
keycode 78 = Scroll_Lock
keycode 79 = KP_7
keycode 80 = KP_8
keycode 81 = KP_9
keycode 82 = KP_Subtract
keycode 83 = KP_4
keycode 84 = KP_5
keycode 85 = KP_6
```

```
keycode 86 = KP_Add
keycode 87 = KP_1
keycode 88 = KP_2
keycode 89 = KP_3
keycode 90 = KP_0
keycode 94 = less greater
keycode 95 = F11 F11
keycode 96 = F12 F12
keycode 108 = KP_Enter
keycode 109 = Control_R
keycode 112 = KP_Divide
keycode 113 = Mode_switch
keycode 114 = Break
keycode 110 = Find
keycode 98 = Up
keycode 99 = Prior
keycode 100 = Left
keycode 102 = Right
keycode 104 = Down
keycode 105 = Next
keycode 106 = Insert
keycode 107 = Delete
!keycode 116 =
keycode 117 = Alt_R

!
! Set the modifiers
!
add shift = Shift_L Shift_R
add lock = Caps_Lock
add control = Control_L Control_R
add mod1 = Alt_L Alt_R

!
! If you don't have ServerNumlock set in your Xconfig, uncomment the following
!
add mod2 = Num_Lock
!add mod5 = Meta_L
!add mod5 = Mode_switch
!keysym Multi_key = Multi_key Meta_L

!
! If you use any of the special default key mappings in Xconfig, they should
! duplicated in this file. Mappings should be added before the section above
! which sets the modifiers.
!
```

```
! For the key specs:
! LeftAlt => keycode 64
! RightAlt => keycode 113
! AltGr => keycode 113
! ScrollLock => keycode 78
! RightCtl => keycode 109
!

! For the mappings:
! Meta => Alt_L Meta_L
! Alt_R Meta_R
! Compose => Multi_key
! ModeShift => Mode_switch
! ModeLock => Mode_switch X386Mode_Lock
! ScrollLock => Scroll_Lock
! Control => Control_R

!
! If you use ModeShift or ModeLock, the following modifier must be set:
!
!add mod5 = Mode_switch
!
! For example, to get the equivalent of:
!
! ScrollLock ModeLock
! RightAlt ModeShift
! LeftAlt Meta
! RightCtl Compose
!
! use the following:
!
!keycode 78 = Mode_switch X386Mode_Lock
!keycode 113 = Mode_switch
!keycode 64 = Alt_L Meta_L
!keycode 109 = Multi_key
!
!add mod5 = Mode_switch
```

## 3.4. Otras preguntas sin clasificar aún...

### 3.4.1. ¿Cómo configuro el X Window System?

Lo más básico que hay que saber, es que el manejo ambiente gráfico está a cargo de todo un sistema llamado X-Window que está por encima del sistema operativo como tal. Además, X (como se le conoce familiarmente), puede funcionar en esquema cliente/servidor lo cual permite que una máquina en Villavicencio pueda desplegar su interface de usuario en otra máquina en Edimburgo haciendo de X todo un protocolo. La versión de X que encontramos en Linux se llama XFree86.

En las nuevas versiones de Linux, X se levanta automáticamente y corre uno de los dos entornos de escritorio gráficos más conocidos: KDE ó GNOME, pero por supuesto, esto se puede modificar y vamos a ver cómo. En las versiones anteriores lo normal era que al bootear, la consola (nombre con el que se conoce la terminal o pantalla principal) se desplegaba en modo texto y uno tenía que levantar X con el comando `startx`, más o menos como cuando había que llamar a Windows desde DOS usando el comando `win`. Para decirle a Linux que levante X por defecto o que no lo haga y deje la consola en modo texto, sólo hay que modificar el archivo llamado `inittab` (que define algunos parámetros que indican hasta qué nivel el sistema debe bootear). Este archivo generalmente se encuentra en el directorio `/etc` en las distribuciones como RedHat. Al editarlo, hay que buscar una línea como:

```
id:5:initdefault:
```

que nos dice que el nivel al que llega el procedimiento de boot es el 5. En las líneas superiores deben estar comentados los niveles que tiene el sistema, pero normalmente el nivel 3 es el normal y el nivel 5 incluye levantar X así que todo lo que tiene que hacer es cambiar el 5 por 3 y rebootear.

### Sesiones gráficas

Bueno, ahora conviene saber que en modo texto por supuesto podemos abrir varias terminales lo cual equivale a varios procesos de sesión de usuario, usando la tecla Alt y las teclas de función (por ejemplo, para abrir otra terminal en modo texto, use Alt+F2). Desde una terminal modo texto podemos entonces levantar X usando el comando `startx` que arranca una sesión X para un "display" ó número de terminal gráfica específico, de manera que si su sesión es la primera, corresponde al display número cero.

En realidad, `startx` es un script que invoca al comando `xinit` que es quién realmente levanta a X. Cuando X está corriendo, para alternar entre sesiones de usuarios ya no se usan las teclas Alt y las teclas de función, sino Ctrl + Alt y las teclas de función. Como se imaginará, esto quiere decir que en una misma máquina se pueden tener varias sesiones X, una por cada sesión de usuario y para retornar a la primera sesión se usa Ctrl + Alt + F7. Ahora, cuando un usuario ya está corriendo su propia sesión X e intenta abrir otra, va a haber un problema porque ya hay una sesión X corriendo (para hacer la prueba, abra una nueva terminal con Ctrl + Alt +

F8 y luego Alt + F1, entre como su usuario y ejecute `startx`). Lo que pasa, es que ya se está usando el display 0 y ahora hay que especificar otros atributos para aclarar que queremos usar otro display. Para hacer esto, es mejor usar el comando `xinit` que ofrece más posibilidades que `startx`.

El comando `xinit` admite muchas opciones pero lo primero que hay que tener en cuenta es que sirve para levantar un servidor X y una ó varias aplicaciones cliente sobre este nuevo servidor. Estas opciones cliente/servidor están separadas por doble guión (`--`) y lo más importante que se debe especificar en las opciones del servidor es el número del display. Entonces, para hacer la prueba, suponga que ya se tiene una sesión X corriendo y ahora se quiere abrir una nueva sesión; así que cambie de sesión de usuario con Ctrl + Alt +F8 y ahora abra una nueva terminal con Alt + F1, entre como su usuario y digite:

```
xinit -- :1
```

y observe el resultado. Lo más probable es que se haya desplegado simplemente una terminal en modo texto pero en una ventana pequeña. Lo que pasa es que en el comando `xinit` solamente se especificaron opciones para el servidor (el número de `display=1`) pero no se especificaron opciones para el cliente, que deberían ir antes del doble guión. Cuando no se incluyen opciones cliente, es decir, no se han especificado las aplicaciones que se ejecutarán en el nuevo servidor X, `xinit` las lee de un archivo llamado `.xinitrc` y si tal archivo no existe, entonces `xinit` ejecuta la aplicación `xterm`, que despliega una terminal modo texto en una ventana aparte. Si realmente quiere abrir otra sesión de usuario como la que normalmente se despliega entonces debe invocar como aplicación cliente al manejador de ventanas (window manager) que usualmente utiliza.

### Ambientes gráficos de escritorio.

Un manejador de ventanas es un programa que controla su ambiente de escritorio y los más sofisticados son KDE y GNOME, que generalmente son clasificados en una categoría superior a un "simple" manejador de ventanas porque ofrecen un gran conjunto de servicios muy potente. De modo que si el ambiente de escritorio con el que Usted se siente más cómodo es, por ejemplo, KDE y quiere abrir otra sesión gráfica con dicho ambiente, entonces ahora modifique la anterior instrucción y digite:

```
xinit /opt/kde/bin/startkde -- :1
```

Debe tener en cuenta que el nombre de la aplicación cliente debe comenzar con un slash (/) y contener toda la jerarquía de ubicación del programa, porque de lo contrario, `xinit` tomará el nombre como una opción de `xterm` y al no reconocerla, abortará la sesión X. Es probable que en su directorio raíz (`$HOME`) tenga su propio archivo `.xinitrc` que puede ser un script ejecutable que corre la aplicación cliente ó ambiente de escritorio de su preferencia cuando ejecuta `xinit` o `startx`. Si no lo tiene, puede hacer un link a un `.xinitrc` genérico que viene por default, con la siguiente instrucción:

```
ln -s /etc/X11/xinit/xinitrc .xinitrc
```

Si examina este script `.xinitrc`, se dará cuenta de que lo que hace principalmente es mirar otro archivo llamado `.Xclients` que a su vez es otro script que revisa el valor de `.wm_style` a ver si contiene el nombre de un manejador de ventanas. Si tiene un script `.Xclients` en su `$HOME`, podrá verificar que lo único que hace es invocar a la aplicación `/usr/X11R6/bin/RunWM` con el parámetro que contiene el archivo `.wm_style` para ejecutar un manejador de ventanas. Puede hacer la prueba de crear el archivo `.wm_style` con una sola línea que contenga la palabra `AfterStep` y observar el resultado cuando levante una sesión X. `AfterStep` es un manejador de ventanas muy liviano y agradable que a pesar de no ser tan sofisticado como KDE, ofrece más o menos el mismo conjunto básico de servicios. Hay muchos otros manejadores de ventanas disponibles muy buenos con los que no estoy familiarizado pero uno de los más populares es el `fvwm95` que es el que da la apariencia de un terriblemente desagradable ambiente `Windows95`.

### Despliegues remotos

Ahora que ya conoce las intimidades de la configuración de un ambiente gráfico, conviene pasar a un tema un poco más arcano que es el de cómo desplegar un cliente en otra máquina para poder manipular una aplicación desde una pantalla remota. Primero, veamos cómo se hace para poder abrir una sola aplicación de una máquina A en la única sesión X de otra máquina B. Como en B tenemos una sola sesión X, quiere decir que está usando su `display=0` y ahora debemos decirle a A que no use el display actual sino el cero de B. Para hacer esto, simplemente reasignamos el valor del `display` en la máquina A con la siguiente instrucción (si quiere comprobar que shell está usando actualmente, digite `echo $0`)

para shells como `tcsh`:

```
setenv DISPLAY B:0
```

Para shells como `bash`, `csh`:

```
set DISPLAY=B:0
```

Con esto le decimos a A que en adelante las aplicaciones gráficas se desplegarán usando el `display` cero de la máquina B. Del lado de B, hay que darle permiso a A para que pueda desplegar sus ventanas en la sesión X con la instrucción:

```
xhost +A
```

que indica que hemos agregado a A a la lista de hosts que tienen permitido desplegar sus clientes en B (si quiere quitar a A de la lista, simplemente digite `xhost -A`). Eso es todo. Para probarlo, ejecute por ejemplo `xterm` en A y comprobará que se ha desplegado en la pantalla de

B. Esto es muy útil por ejemplo para labores de administración remota aunque hay que tener cuidado con la seguridad por lo que para este tipo de tareas es conveniente revisar opciones disponibles para encriptación.

Claro, algunos podrían decir que esto no es “big deal” porque estaban esperando abrir el ambiente completo del usuario de A en la pantalla de B! Pues claro que también se puede. Como sabemos, el ambiente de escritorio es un cliente, así que lo primero será abrir otro servidor X en B que tendrá el `display=1` con `Ctrl + Alt + F8`, `Alt + F1`, `xinit -- :1` y en la nueva terminal darle permisos a A con `xhost +A` para finalmente disparar el ambiente completo desde A haciendo primero `setenv DISPLAY B:1` (para decirle que se va a usar el display uno de B) y luego `startkde` ó `afterstep` con lo cual logramos abrir toda la sesión del usuario de A en B. Fíjese que en este caso quedó abierta la terminal de B que se abrió inicialmente y quedará visible en la sesión gráfica del usuario de A.

Fíjese que en el caso anterior, tocó primero abrir el nuevo servidor X en B y la aplicación cliente se abrió desde A para ser desplegada en el nuevo servidor X de B. De todos modos sí podemos disparar un nuevo servidor X de A en B de la siguiente forma:

```
X :1 -query A
```

Al hacer esto, todo el servidor X de A corre remotamente y se despliega en B (en el display 1) por lo cual aparecerá la pantalla que pide el login y el password de usuario, contrariamente al esquema que se vió antes en donde se desplegaba en B el ambiente de el usuario de A que había invocado su ambiente de trabajo gráfico como aplicación cliente. Tenga en cuenta que el último esquema es más peligroso porque para ingresar al sistema Usted tendrá que mandar su login y password no?

Y listo, creo que con esto es suficiente por ahora.

### 3.4.2. ¿Cómo restringir los directorios públicos en Apache?<sup>16</sup>

En esta ocasión se me presentó un problema con la publicación de unas páginas que contenían información confidencial. Era prioritario que únicamente ciertos usuarios pertenecientes a la compañía pudieran consultar la información contenida en las páginas, pero únicamente se contaba con un servidor Web público. Aunque a estas páginas no se llegaba utilizando ningún hipervínculo, el hecho de ponerles nombres que desconocieran la mayoría de las personas, tanto al interior como al exterior de la compañía, o ubicarlas en un directorio no conocido no brindaba ninguna seguridad de que `_alguien_` se diera cuenta de que existían.

Para todos los que utilizan el servidor Apache/1.3.6 (o una versión posterior a la 1.2) existe la opción de restringir, por medio de un login y un password el acceso a cada directorio y a los subdirectorios contenidos en éste.

Para verificar la versión que usted esta utilizando puede utilizar el comando `httpd` (`httpd - Apache hypertext transfer protocol server`, que generalmente está ubicado en el directorio `/usr/sbin/`) con el parámetro `-v`. El programa se ejecutará, le entregará la versión y se terminará.

---

<sup>16</sup>Repuesta por Jaime Torres, MerLinux

```
[root@myserver sbin]# httpd -v
Server version: Apache/1.3.6 (Unix) (Red Hat/Linux)
Server built: Apr 7 1999 17:17:41
```

Para que los usuarios se autentiquen ante el servidor es necesario crear un archivo que contenga los logins con los que se puede ver la información y sus respectivos passwords. Para crearlo: elegí el archivo `/usr/local/etc/httpd_users`. Para agregar cada uno de los usuarios utilicé el comando `htpasswd` que permite crear el archivo, adicionar y modificar usuarios.

```
[root@myserver sbin]# htpasswd -c /usr/local/etc/httpd jaime
New password:
Re-type new password:
Adding password for user jaime
```

El parametro `-c` indica que quiero crear el archivo. El comando pide que se digite el password que utilizará el usuario `jaime`. El archivo creado tendrá el siguiente aspecto:

```
carlos:yT29qUV8I1uVY
juan:z/WTuIYruAJr2
jaime:cgYHRjXeTAaBE
```

Tiene la misma estructura que el archivo `/etc/passwd`. La primera columna es el login y la segunda el password encriptado. Si se quieren adicionar nuevos usuarios se vuelve a utilizar el comando `htpasswd` pero omitiendo el parámetro `-c`. Si se vuelve a utilizar se sobrescribirá el contenido del anterior archivo. Para cambiar el password de un usuario se utiliza:

```
[root@myserver sbin]# htpasswd /usr/local/etc/httpd jaime
New password:
Re-type new password:
```

y de esta forma, si el usuario `jaime` ya existía dentro del archivo, se le asignará un nuevo password.

Configurar el servidor para que utilice la restricción es el siguiente paso. Se debe crear un archivo llamado `.htaccess` en el directorio que se quiere proteger. El archivo deberá llevar: Un nombre de autenticación que aparecerá la ventana del navegador cuando éste indique que se necesita un login y un password.

```
AuthName "Paginas restringidas"
```

Un tipo de autenticación (yo utilizo el tipo `Basic` porque aunque existe un tipo `Digest` que brinda más seguridad, éste no fue implementado para algunos navegadores de viejas versiones).

```
AuthType Basic
```

El archivo de usuarios permitidos.

```
AuthUserFile /usr/local/etc/http_users
```

Y una especificación de que es necesario ser un usuario válido para ingresar a este directorio del servidor web.

```
require valid-user
```

Con estas cuatro especificaciones basta. Ahora hay que configurar el servidor Apache para que trabaje con las restricciones. Para esto se edita el archivo `access.conf` (generalmente ubicado en `/etc/httpd/conf/`) para que permita hacer la autenticación de usuarios a través del archivo `.htaccess`. Se debe agregar el mandato de `AllowOverride AuthConfig` para este efecto. Generalmente, al interior del archivo, existe una documentación que permite establecer exactamente en dónde se debe ubicar la línea requerida. El archivo debe tener en su interior:

```
# This controls which options the .htaccess files in
# directories can override. Can also be "All", or any
# combination of "Options", "FileInfo", "AuthConfig", and
# "Limit"
```

```
AllowOverride AuthConfig
```

Allí se debe colocar la línea. Note que existen otros métodos de restricción de directorios, pero para el caso utilizaremos `AuthConfig`.

En este instante se debe re-arrancar el servidor web (utilizando el comando `httpd`) y las restricciones empezarán a limitar la entrada de usuarios al directorio elegido (y a los subdirectorios contenidos en éste).

Se puede llegar a limitar el acceso utilizando políticas de grupos y otros medios de autenticación pero los demás temas no serían abordados en este documento.



## Capítulo 4

# Desarrollo

El propósito de éste capítulo es incluir información relacionada con el desarrollo de software utilizando programas de software libre.

Incluimos un amplio espectro de información, con el propósito de incentivar y facilitar el desarrollo de software.

Este capítulo aún se encuentra en construcción y aún falta incluir y mejorar muchas respuestas. Lo invitamos a contribuir escribiendo a <colibri-www@bachue.com>.

El encargado de éste capítulo es Alejandro Forero Cuervo <bachue@bachue.com>.

### 4.1. ¿Qué ambientes de programación basados en software libre se encuentran disponibles? <sup>1</sup>

Existe una buena cantidad de ambientes de desarrollo de software libre para diferentes lenguajes.

Una alternativa, quizá la más común, consiste en utilizar un editor de texto integrado a las diversas herramientas del proyecto GNU.

Los editores de texto utilizados con mayor frecuencia para desarrollar software son vim (<http://www.vim.org>) y emacs (<http://www.gnu.org/software/emacs/emacs.html>). Ambos editores suelen ser un poco difíciles de utilizar al comienzo (sus comandos son poco intuitivos) pero muy eficientes una vez que el usuario se ha familiarizado con ellos. Ambos editores ofrecen las principales características que se pueden esperar en un editor de software moderno. Por ejemplo, tienen coloración de sintaxis para una gran cantidad de lenguajes de programación, permiten editar archivos remotamente (sobre FTP, Secure Shell u otros protocolos), etc..

Existe una gran cantidad de herramientas que se integran fácilmente con estos editores:

---

<sup>1</sup>Esta pregunta fue propuesta por Jhair Tocancipá y editada, con base en sugerencias de Juan Luis Baptiste, Gustavo Gonzalez, y Sergio Roa, por Alejandro Forero Cuervo.

- Para compilar los archivos se suele ejecutar GCC, el compilador del proyecto GNU, capaz de compilar código en C, C++, Fortran, Java y otros lenguajes. La integración entre GCC y estos editores les permite a los últimos reportar los errores detectados en el proceso de compilación de manera eficiente.
- Para manejar programas con varios archivos de código fuente se suelen utilizar archivos Makefile, manejados por GNU Make, en los que se describen las dependencias entre los archivos. Para proyectos de gran tamaño es recomendable utilizar Autoconf y Automake, dos herramientas también del proyecto GNU, que ayudan con la generación de los archivos Makefile y a que el software funcione en la mayor cantidad de plataformas posibles.
- Para depurar código de C o C++ se suele utilizar GDB, el depurador del proyecto GNU. La integración de Emacs con GDB permite controlar el proceso de depuración enteramente desde Emacs de manera bastante práctica.
- Para ayudar a la navegación del código se suelen utilizar archivos especiales (de “tags”), que contienen una lista de todos los símbolos (funciones, variables, estructuras, etc.) definidos en todos los archivos de código que hacen parte del proyecto.

Otra alternativa diferente es utilizar un ambiente de desarrollo integrado (a éstos se les suele llamar IDEs, del inglés “integrated development environment”). Entre los IDEs disponibles en software libre se encuentran los siguientes:

- Anjuta (<http://www.anjuta.org/>), un ambiente de desarrollo pensado especialmente en C y C++. Fue escrito por el proyecto GNOME utilizando GTK+ y provee manejo de proyectos, guías (wizards) para guiar la creación de aplicaciones, depuración interactiva y edición de código.
- Eclipse (<http://www.eclipse.org/>), un avanzado ambiente de desarrollo, que suele ser muy utilizado para proyectos basados en Java. Es bastante modular y existe una gran cantidad de plugins que aumentan su funcionalidad drásticamente, probablemente convirtiéndolo en la mejor herramienta libre para trabajar J2EE.
- KDevelop (<http://www.kdevelop.org/>), un ambiente desarrollado por el proyecto KDE. Provee manejo de proyectos (generación de archivos Makefile), integración con herramientas de traducción e internacionalización, depuración interactiva, control de versiones (apoyado en CVS), edición de código y generación de interfaces gráficas (tanto para KDE/Qt como para GNOME/GTK+). Soporta proyectos basados en C, C++, Java, Add, Perl, Ruby y Pascal.

## 4.2. ¿Qué significa un paradigma de programación?<sup>2</sup>

Un paradigma de programación, como su nombre lo implica, refleja la manera de pensar de los desarrolladores, la manera en que afrontan los problemas y construyen sus soluciones.

---

<sup>2</sup>Esta respuesta fue redactada por Alejandro Forero Cuervo.

Los principales paradigmas de programación son la programación orientada a objetos, la programación funcional, la programación orientada a tablas, la programación lógica, la programación orientada a aspectos y la programación imperativa.

Algunos paradigmas son muy profundos y tienen muchas implicaciones sobre la actividad de desarrollar software, mientras que otros se centran en aspectos mucho menores pero que, según sus proponentes, hacen que este proceso sea mucho más eficiente.

### 4.3. ¿Qué quiere decir que un lenguaje sea funcional? <sup>3</sup>

Al decir que un lenguaje de programación es funcional, se suele hacer referencia al paradigma de programación funcional.

En términos general, al desarrollar software en el paradigma de programación, en contraste con el paradigma de programación imperativo, se tiende a enfatizar más la evaluación de expresiones que la ejecución secuencial de comandos.

En el fondo, para ejecutar programas los computadores ejecutan, de manera secuencial, instrucciones de código de máquina. De cierta forma, ésto hace que los lenguajes imperativos sean más cercanos a la manera en que funcionan los computadores, al forzar a los programadores a formular sus algoritmos como una serie de pasos que se realizarán de manera secuencial.

Los lenguajes funcionales, por otro lado, son más cercanos a la manera en que funciona la mente humana, pues tienden a permitirle a los programadores describir sus algoritmos como expresiones que serán evaluadas.

Por ejemplo, cualquier matemático definiría la función factorial de la siguiente manera:

Pendiente: Traducir de LaTeX a SGML:

```
\begin{displaymath} f(x) = \left\{ \begin{array}{ll} 1 & x = 0 \\ x \times f(x - 1) & x > 0 \end{array} \right. \end{displaymath}
```

En un lenguaje imperativo como C o Java, esta función se programaría de una manera similar a la siguiente:

```
int
fact ( int x )
{
    int total = 1;

    while ( x < 0 )
    {
        total *= x;
        x--;
    }
}
```

---

<sup>3</sup>Esta respuesta fue escrita por Alejandro Forero Cuervo, con base en información extraída del documento Erbis Scheme de su autoría.

```
    }  
    return total;  
}
```

Como se ve, el algoritmo para calcular la función es formulado como una secuencia de pasos que se ejecutan en orden.

En contraste, una implementación funcional, en un lenguaje como Haskell o Scheme, la función factorial tendería a ser programada de una forma similar a la siguiente:

```
let rec fact x = if x = 0 then 1 else x * fact(x - 1);;
```

Vale la pena anotar que ambos ejemplos podrían ser mejorados: en el primero se podrían ahorrar algunas líneas de código y el segundo podría modificarse para utilizar una cantidad de memoria constante.

Hay varias consecuencias de el énfasis en la evaluación de expresiones que son comunes a la mayoría de lenguajes funcionales:

- *Los procedimientos son ciudadanos de primera categoría* Los procedimientos y funciones son objetos visibles que pueden ser almacenados dentro de estructuras complejas, pasados como argumentos, construidos en tiempo de ejecución y manipulados al igual que otros tipos como los números y las cadenas.
- *Manejo automático de memoria* El programador no necesita llevar un registro manual de la memoria utilizada por cada objeto y liberarla; la implementación del lenguaje se encarga de eso de manera automática.

Los lenguajes funcionales han permanecido en uso por mucho tiempo y han mostrado todo el poder de su gran nivel de expresividad. Al permitir la creación de procedimientos en tiempo de ejecución, permiten un gran nivel de modularidad que, de acuerdo con sus proponentes, difícilmente puede alcanzarse en otros paradigmas de programación.

Dentro de los lenguajes funcionales se encuentran Lisp, Haskell, CaML y muchos otros.

## 4.4. Programación orientada a objetos <sup>4</sup>

### 4.4.1. ¿Qué es la programación orientada a objetos?

La programación orientada a objetos, frecuentemente llamada OOP por su sigla en inglés, es aquella en la que el software es construido buscando simular el comportamiento de objetos

---

<sup>4</sup>Las respuestas de esta sección fueron escritas por Sergio Roa [s.roa@computer.org](mailto:s.roa@computer.org) y Alejandro Forero Cuervo [bachue@bachue.com](mailto:bachue@bachue.com).

reales de manera directa. La idea central consiste en identificar los objetos concretos que intervienen en las situaciones que el software debe manejar y organizar la modularidad del software en torno a ellos. Una vez identificados los objetos importantes, se identifican las acciones que se pueden realizar con ellos y su información de estado relevante para el sistema.

Los lenguajes orientados a objetos son aquellos que han sido pensados explícitamente para dar soporte a este paradigma de programación. En la mayoría, cada tipo de objetos suele llamarse una “clase”, las variables en las que se almacenará la información sobre su estado “atributos” o “propiedades” y las funciones que simularán las acciones que se pueden realizar con los objetos “métodos”.

De acuerdo con sus seguidores, la programación orientada a objetos permite pensar las cosas con un gran nivel de abstracción, lo que en muchos casos hace que los programas sean más fáciles de entender y de modificar. Además, la implementación de cada clase corresponde a un modelo concreto que puede ser utilizado de manera relativamente independiente de los otros, lo que facilita la reutilización de código en múltiples aplicaciones.

#### 4.4.2. ¿Qué lenguajes orientados a objetos existen?

Tal vez los dos lenguajes orientados a objetos más populares son Java (<http://java.sun.com>) y C++ (<http://www.research.att.com/~bs/C++.html>). Dentro de los lenguajes orientados a objetos imperativos se cuentan, en adición a los anteriores, Objective C, Smalltalk, ADA 95, Object Pascal e Eiffel. Existen varios lenguajes funcionales orientados a objetos como Lush (<http://lush.sourceforge.net/>), un lenguaje basado en Lisp, y Ocaml (<http://www.ocaml.org/>), basado en Caml (<http://caml.inria.fr/>) (Haskell (<http://www.haskell.org>), por su parte, incorpora algunas ideas importantes de la programación orientada a objetos pero no las suficientes como para poder ser considerado orientado a objetos).

#### 4.4.3. ¿Qué quiere decir que un objeto hereda de otro?

La herencia es un mecanismo común a la mayoría de lenguajes orientados a objetos a través del cual se pueden crear clases derivadas de otras, extendiendo su funcionalidad pero conteniendo funcionalidad única.

Por ejemplo, se puede crear la clase Persona, con funcionalidad básica común a todas las personas, y la clase Estudiante, con cierta funcionalidad adicional a la anterior. En este ejemplo, la herencia permite al programador dotar a la clase Estudiante de toda la funcionalidad de la clase Persona sin tener que duplicar código. Cada instancia de la clase Estudiante tendrá todas las propiedades de la clase Persona y cuando se ejecute uno de los métodos de la segunda sobre una instancia de la primera, se utilizará automáticamente el código de la clase Persona. En este caso se dice que la clase Estudiante hereda los atributos y métodos de la clase Persona (o, de manera más breve pero equivalente, que Estudiante hereda de Persona).

El mecanismo de herencia no está limitado a un solo nivel. En el ejemplo anterior podría adicionarse una tercera clase EstudiantePregrado que herede de la clase Estudiante, en cuyo caso se tendría una jerarquía de objetos que permitiría a la clase EstudiantePregrado utilizar la funcionalidad tanto de Estudiante como de Persona.

#### 4.4.4. ¿Qué quiere decir que un lenguaje permite la herencia múltiple?

Se dice que un lenguaje orientado a objetos permite la herencia múltiple cuando una clase puede heredar directamente de más de una clase. En algunos lenguajes no se permite la herencia múltiple, lo que significa que las clases no pueden heredar directamente las propiedades de más de una clase.

Esto no incluye los casos en que una clase hereda las propiedades de otra y ésta a su vez las de una tercera. Un caso como éste es perfectamente válido en un lenguaje que no soporte la herencia múltiple.

#### 4.4.5. ¿Qué es el encapsulamiento en un lenguaje orientado a objetos?

La mayoría de lenguajes orientados a objetos ofrecen técnicas especiales que permiten controlar qué porciones de código tienen acceso a los diferentes atributos y métodos de las clases.

Una práctica bastante común en ciertos lenguajes es asegurarse de que únicamente los métodos de cada clase tengan acceso directo a sus atributos. En este caso, para obtener información de los atributos de una clase desde otra es necesario invocar a uno de los métodos de la que contiene su información para que éstos la retornen.

Estas técnicas, conocidas como de encapsulamiento de información, fuerzan a los programadores a seguir ciertas prácticas de programación y algunos las consideran muy útiles para simplificar la complejidad del código al simplificar la interfaz de cada módulo escondiendo en su interior todos los detalles de implementación (atributos y métodos) que no deban ser visibles desde afuera.

#### 4.4.6. ¿Qué es el paso de mensajes en un lenguaje orientado a objetos?

El paso de mensajes es una abstracción que en algunos casos ayuda a entender el funcionamiento del software. Cada invocación que el código de una clase hace a un método de otra se puede pensar como un mensaje que es pasado de un objeto a otro y genera una respuesta. Esto permite realizar diagramas en los que se visualice la manera en que cada mensaje se propaga a través de la red de clases.

#### 4.4.7. ¿Qué programas de software libre han sido desarrollando utilizando programación orientada a objetos?

Muchos programas importantes de software libre han sido desarrollados utilizando programación orientada a objetos. Por ejemplo, para los dos proyectos de escritorios gráficos de software libre más importantes, GNOME (<http://www.gnome.org>) y KDE (<http://www.kde.org>), la metodología de programación orientada a objetos ha sido fundamental y está presente en sus librerías más importantes. Otros ejemplos importantes los constituyen Mozilla (<http://www.mozilla.org/>), uno de los más importantes programas de navegación, y OpenOffice (<http://www.openoffice.org>), la suite de herramientas de oficina de software libre más avanzada.

## 4.5. ¿Qué quiere decir que un lenguaje provea recolección de basura automática?<sup>5</sup>

En el transcurso de la ejecución de un programa tienen que reservarse porciones de memoria para almacenar variables de manera temporal. En algunos lenguajes los programas deben liberarlas explícitamente cuando ya no las necesitan. En otros, la implementación del lenguaje se encarga automáticamente de detectar cuáles variables han dejado de ser requeridas y de liberar el espacio en memoria que ocuparan. Se dice que estos últimos proveen recolección de basura automática o manejo automático de memoria (en inglés se suele usar el término “garbage collection”).

En términos generales es deseable que un lenguaje provea recolección automática de basura pues ésto tiende a simplificar el código fuente de los programas. En los lenguajes que no proveen recolección automática los desarrolladores tienen que construir mecanismos para que los programas detecten cuándo liberar las porciones de memoria asociadas con sus variables.

Sin embargo, en algunas situaciones poco comunes el uso de algoritmos genéricos para colección de basura (como los que pueden esperarse en una implementación de un lenguaje con colección de basura automática) puede ser poco aceptable. En éstos casos puede ser preferible un lenguaje que no provea colección automática de basura para tener control sobre el mecanismo particular utilizado para liberar la memoria que ya no es requerida.

Entre los lenguajes que no proveen recolección de basura automática se encuentran C, C++. Entre los que si lo hacen están Haskell, Lisp, APL, Perl, Java y Python.

## 4.6. ¿Qué son las continuaciones?

## 4.7. Desarrollo en C

### 4.7.1. ¿Qué ventajas y desventajas tiene C en comparación con otros lenguajes de programación?

## 4.8. Desarrollo en Haskell

## 4.9. Desarrollo en Java

### 4.9.1. ¿Qué herramientas libres existen para desarrollo en Java?<sup>6</sup>

#### Herramientas básicas

Para desarrollar en Java son necesarios como mínimo los siguientes programas:

---

<sup>5</sup>Pregunta planteada y contestada por Alejandro Forero Cuervo.

<sup>6</sup>Repuesta inicialmente redactada por Jhair Tocancipá

1. Un compilador, que transforme programas escritos en el lenguaje de programación Java (cuya especificación puede leerse aquí (<http://java.sun.com/docs/books/jls/>)) en código binario que puede ser ejecutado por la llamada Máquina Virtual de Java (JVM<sup>7</sup>).
2. Una implementación de la JVM, que puede ejecutar código generado por el compilador. La JVM puede entenderse como un computador emulado en software, con un conjunto de instrucciones y manipulación de áreas de memoria propias. La especificación de la JVM, al igual que la del lenguaje de programación Java, es abierta, y puede leerse en la siguiente dirección: <http://java.sun.com/docs/books/vmspec/>.

Algunas implementaciones libres de la JVM y del lenguaje Java (i.e. el compilador) son:

- SableVM (<http://www.sablevm.org>): Se trata de una implementación de la JVM escrita en C (no incluye un compilador, esto es, puede serle útil si usted solo está interesado en **ejecutar** el código generado por algún compilador de Java).
- Kaffe (<http://www.kaffe.org/>): Otra implementación de la JVM.
- Jikes (<http://oss.software.ibm.com/developerworks/opensource/jikes/>): Un compilador de Java.
- GCJ (<http://gcc.gnu.org/java/>): El compilador GNU de Java.

### Jitters<sup>8</sup>

Además de generar código para ser ejecutado por la JVM, es posible generar código para ser ejecutado por la máquina nativa sobre la cual se ejecuta una JVM determinada.

Dicho código es generado por un *Just-In-Time Compiler* (también conocidos como *jitters*). La distribución oficial de Java incluye un *jitter*, de los cuales también hay versiones libres, como por ejemplo:

- shuJIT (<http://www.shudo.net/jit/>): genera código para procesadores x86.
- ElectricalFire (<http://www.mozilla.org/projects/ef/>): un compilador JIT algo experimental del proyecto Mozilla.

### Ambientes de programación - programación por componentes

*Falta completar ésta sección...*

- JBoss ([www.jboss.org](http://www.jboss.org)): Implementa la EJB Specification (<http://java.sun.com/products/ejb/>).<sup>9</sup>

---

<sup>7</sup>Del inglés *Java Virtual Machine*.

<sup>8</sup>Pregunta formulada por Jeyson D. Polanco [fractal@colombialinux.org](mailto:fractal@colombialinux.org) en [colibri@bachue.com](mailto:colibri@bachue.com) - 19 Sep 2003. Respuesta formulada por Jhair Tocancipá.

<sup>9</sup>EJB=*Enterprise Java Beans*

- Apache Tomcat (<http://jakarta.apache.org/tomcat/>): Implementa la Java Servlet Specification (<http://java.sun.com/products/servlet/index.html>).

#### 4.9.2. ¿Qué ventajas y desventajas tiene Java en comparación con otros lenguajes de programación?

### 4.10. Desarrollo en Lisp

#### 4.10.1. ¿Qué es Lisp? <sup>10</sup>

Lisp es una familia de lenguajes funcionales no estrictos cuyos principales dialectos son Scheme y Common Lisp.

#### 4.10.2. ¿Cuál es la historia de Lisp? <sup>11</sup>

Lisp descende de uno inventado por el matemático John McCarthy a finales de 1950. El lenguaje diseñado por McCarthy tenía interés matemático pues buscaba ofrecer un mecanismo alternativo a las máquinas de Turing que fuera más práctico para formalizar algoritmos. A estudiantes de McCarthy se les ocurrió que sería muy práctico contar con una máquina concreta que permitiera la ejecución de programas escritos en este lenguaje y realizaron la primera implementación de Lisp.

Al cabo de algunos años Lisp evolucionó bastante, incorporando nuevas ideas, como la de hacer visibles las continuaciones al programador, y hoy continua siendo una alternativa importante dentro de los lenguajes de programación.

#### 4.10.3. ¿Qué características particulares diferencian a Lisp de otros lenguajes de programación? <sup>12</sup>

Las siguientes son algunas características de Lisp. La mayoría son comunes a las principales variantes de Lisp pero algunas pueden no encontrarse en determinados dialectos o implementaciones.

- Recolección automática de memoria. La mayoría de implementaciones de Lisp se encargan de manejar la memoria automáticamente, liberando el espacio utilizado por objetos cuando dejan de ser necesitados.
- Evaluación perezosa. Lisp incluye un mecanismo bastante sencillo para utilizar evaluación perezosa de expresiones.

---

<sup>10</sup>Esta respuesta fue redactada por Alejandro Forero Cuervo.

<sup>11</sup>Esta respuesta fue redactada por Alejandro Forero Cuervo.

<sup>12</sup>Esta respuesta fue escrita por Alejandro Forero Cuervo. Partes de la información fueron extraídas del R5RS.

- Tipos dinámicos (débiles). Lisp no posee un sistema de tipos estáticos (de tiempo de compilación). Los tipos en Lisp son asociados a los valores más que a las variables. Esto implica que los errores de tipos son detectados en tiempo de ejecución.
- Recursión por cola. Las implementaciones de Lisp suelen ser programadas de manera que ciertos cálculos iterativos puedan realizarse en espacio (memoria) constante aún cuando sean descritos utilizando recursión.
- Las continuaciones tienen status de primera clase. En Scheme, las continuaciones se encuentran presentes en el lenguaje y pueden ser capturadas y manipuladas por el programador. Esto es muy útil para implementar una gran cantidad de construcciones avanzadas para controlar el flujo de ejecución del programa, incluyendo salidas no locales de funciones, técnicas de backtracking y corutinas.
- Sistema de macros. Scheme cuenta con un avanzado mecanismo para la construcción de macros, que permite construirlas de manera sintácticamente limpia.

Pendiente: mencionar aritmética.

#### 4.10.4. ¿Cuáles son las principales diferencias entre Common Lisp y Scheme?

#### 4.10.5. ¿Qué es Emacs-Lisp? <sup>13</sup>

El editor de texto Emacs está escrito casi enteramente en una variante de Lisp conocida como Emacs-Lisp. El uso de una variante de Lisp para programar este editor ha hecho que sea muy fácil de extender.

De acuerdo con Richard Stallman, el creador de Emacs (quien también es el fundador y líder del proyecto GNU), “Lisp es el lenguaje de programación más poderoso, y si quiere un interprete, Lisp es el mejor. Ninguno de los otros lenguajes llega a acercarse a Lisp en su poder”.

#### 4.10.6. ¿Cómo puedo aprender a programar en Lisp? <sup>14</sup>

Si no sabe programar en Lisp, lo más recomendable es que aprenda primero Scheme, que es un poco más sencillo que Common Lisp.

La especificación definitiva del lenguaje Scheme es el conocido Revised 5th Report on the Algorithmic Language Scheme, conocido por las siglas R5RS. Este documento es muy claro y puede servirle si ya ha programado en otros lenguajes y quiere aprender a hacerlo utilizando Scheme.

Un buen documento para aprender a programar, que de paso utiliza el lenguaje Scheme, es el libro Structure and Interpretation of Computer Programs, abreviado SICP, que ha sido utilizado por años en varias universidades para enseñar a programar a los estudiantes. Este libro

---

<sup>13</sup>Respuesta por Alejandro Forero Cuervo.

<sup>14</sup>Esta inquietud fue planteada por Camilo Uribe y respondida por Alejandro Forero Cuervo.

explica el funcionamiento de Scheme y trae una gran cantidad de ejercicios muy buenos. Puede beneficiarse mucho consultando este libro, inclusive si ya sabe programar en Scheme. El SICP está disponible para ser descargado por internet.

Puede encontrar tanto el R5RS como el SICP en cualquier motor de búsqueda. Ambos documentos están en inglés.

Para aprender a programar en Common Lisp le recomendamos consultar el capítulo 2 del libro *Ansi Common Lisp* (<http://www.paulgraham.com/lib/paulgraham/acl2.txt>) (en inglés).

#### 4.10.7. ¿Qué implementaciones de Lisp me recomiendan utilizar?<sup>15</sup>

La siguiente es una lista incompleta de programas de software libre que permiten la ejecución de código escrito en Lisp:

- Bigloo, una implementación de Scheme.
- CMUCL, una implementación de Common Lisp.
- Emacs, un editor de texto escrito en su mayoría en Emacs-Lisp, incluye un interprete de esta variante de Lisp.
- GNU CLisp, la implementación de Common Lisp del proyecto GNU.
- Guile, el interprete de Scheme del proyecto GNU, es quizá uno de los sistemas más difundidos para ejecutar programas escritos en Scheme. Existe una buena cantidad de código que permite la integración de Guile con librerías particulares como las del proyecto GNOME (lo que permite la creación de programas con interfaces gráficas programando enteramente desde Scheme).
- MIT Scheme, la implementación de Scheme utilizada en el M. I. T. para fines pedagógicos.
- SIOD
- Stalin, un compilador de Scheme a lenguaje de máquina. Introduce algunas limitaciones al lenguaje (por ejemplo, no se pueden utilizar continuaciones ni la función eval y se utilizan los tipos básicos de C para representar la información) con el propósito de incrementar la velocidad de ejecución. Muchas personas la consideran la mejor alternativa cuando la velocidad de ejecución es importante.

#### 4.10.8. ¿Qué programas de software libre hay escritos en Lisp?<sup>16</sup>

Tal vez el principal programa de software libre escrito en Lisp sea Emacs: toda la funcionalidad de alto nivel (la lógica) está escrita en una variante de Lisp conocida como Emacs-Lisp, ejecutado por un interprete escrito en C (que hace parte de Emacs).

<sup>15</sup>Esta respuesta fue escrita por Alejandro Forero Cuervo.

<sup>16</sup>Esta respuesta fue añadida por Alejandro Forero Cuervo.

En cuanto a Scheme se podría citar el hecho de ser, junto con Perl, el lenguaje preferido para construir imágenes de manera programática utilizando The GIMP.

## **4.11. Desarrollo en Perl**

**4.11.1. ¿Qué ventajas y desventajas tiene Perl en comparación con otros lenguajes de programación?**

**4.11.2. ¿Cómo puedo aprender a programar en Perl?**

## **4.12. Desarrollo en PHP**

## **4.13. Desarrollo en Python**

**4.13.1. ¿Qué ventajas y desventajas tiene Python en comparación con otros lenguajes de programación?**

## Capítulo 5

# Aplicaciones

### 5.1. Internet

#### 5.1.1. ¿Cuáles son los comandos básicos de *Internet Relay Chat* (IRC)?<sup>1</sup>

##### Introducción

IRC es conocido popularmente como el programa que permite entablar conversaciones simultáneas (*chat*) con personas conectadas a distintos computadores. Fué concebido inicialmente en Finlandia por Jarkko Oikarinen (<http://www.kumpu.org/jto/>) a finales de la década de los 80s.

IRC consiste en realidad de

- Un conjunto de redes (o servidores), como por ejemplo Undernet (<http://www.undernet.org/>) o Dalnet (<http://www.dal.net/>), sobre los cuales se ejecuta un software servidor de IRC.
- Clientes especializados utilizados por los usuarios, que permiten la comunicación con los servidores de IRC. Un cliente es por ejemplo XChat (<http://xchat.org>) (ver más adelante en para una lista con otros clientes libres de IRC).

Los comandos de IRC se pueden clasificar en dos grandes grupos. Por un lado, están los comandos que permiten ejecutar funciones de mantenimiento del software servidor de IRC, como por ejemplo limitar el número de personas conectadas al servidor (para ejecutar éstos comandos se requiere de privilegios especiales).

Por otro lado, están los comandos que no requieren de privilegios especiales y que ofrecen funcionalidad básica, como por ejemplo conectarse a un servidor, o enviar un mensaje a un usuario en particular.

Por convención, cualquier línea que comience con un *slash* '/' es interpretada por los clientes de IRC como un comando.

<sup>1</sup>Respuesta original por Juan F. A. Saldarriaga nebiros@gigax.org, editada y ampliada por Jhair Tocancipá.

## Comandos generales<sup>2</sup>

- `/SERVER <nombre del servidor> [<puerto>]`: Éste comando permite especificar el servidor (primario) de IRC al cual conectarse.

Por ejemplo con:

```
/SERVER irc.debian.org

Connecting to leguin.acc.umu.se (130.239.18.172) port 6667..
--- Connected. Now logging in..
--- AUTH :*** Looking up your hostname...
--- AUTH :*** Found your hostname, welcome back
--- AUTH :*** Checking ident
--- AUTH :*** Got ident response
:Welcome to the freenode IRC Network jto_
```

Nos conectamos al servidor de IRC de Debian. Para ser más precisos en éste caso nos estamos conectando a un servidor primario (`leguin.acc.umu.se` en éste caso) de la red de IRC de Debian.

- `/JOIN #<nombre del canal>`: Para poder conversar con otras personas en IRC no basta con especificar un servidor. Debido al gran número de personas que se conectan a estos servidores y a los distintos intereses de éstas personas, los servidores manejan distintos canales. Éste comando permite especificar un canal al cual conectarse.

Por convención los nombres de los canales comienzan con el símbolo '#'.

Así con el siguiente comando,

```
/JOIN #colibri

You are now talking on #colibri
```

nos conectamos al canal de Colibri.

- `/PART [#<nombre del canal>] [<razón>]`: Nos permite abandonar un canal. Sí queremos por ejemplo salir del canal de Colibri, podemos usar:

```
/PART #colibri hora de almorzar!

You have left channel #colibri ("hora del almorzar!")
```

- `/LIST [#<nombre del canal>] [-MIN n] [-MAX n]`: Lista los canales actualmente disponibles.

Por ejemplo, el siguiente comando nos da una lista de los canales con mínimo 5 y máximo 10 usuarios conectados:

---

<sup>2</sup>Sintaxis tomada del sistema de ayuda del cliente libre de IRC XChat.

```
/LIST -MIN 5 -MAX 20
```

```
*** #justlinux 16      Unofficial Just Linux Channel | Flood in #flood |
+Bots | Channel Guidelines : http://serz.swordmasters.com.ar/jl/ | Gnom
+FINAL today? :)
*** #debian.pl 9       Kto mi napisze driver do karty obsługujacej tylko
+1.2 do BeOS-a? :->
*** #kerneled 12      Are you kerneled? - http://www.kerneled.com ||
```

- `/ME <mensaje>`: Muestra en canal actual una descripción sobre lo que el usuario está haciendo. Por ejemplo:

```
/me Making format c:
```

```
* nickname Making format c:
```

- `/MSG <nick> <mensaje>`: Envía un mensaje privado a un usuario particular.
- `/QUERY <nick>`: Abre una nueva ventana con la cual enviar mensajes privados a un usuario particular.
- `/WHOIS <nick>`: Muestra información sobre un usuario particular, por ejemplo:

```
/WHOIS jto
```

```
[jto] (jto@p508B1427.dip.t-dialin.net) : Jhair Tocancipa Triana
--- [jto] #colibri #debian
--- [jto] leguin.freenode.net :Umeå, SE, EU
--- [jto] idle 00:06:05, signon: Thu Sep 11 22:34:11
--- [jto] End of WHOIS list.
```

- `/NICK <nick>`: Cambia el sobrenombre (en inglés *nickname* o *nick*) del usuario actual. Por ejemplo:

```
/NICK jhair
```

```
You are now known as jhair
```

- `/QUIT [<razón>]`: Éste comando desconecta a un usuario del servidor de IRC y muestra el mensaje opcional en los canales en los que estaba conectado. (Éste mensaje solo le aparece a las personas que están en los mismos canales del usuario que sale).
- `/AWAY <mensaje>`: Escribe un mensaje en los canales actuales que aclara que el usuario no está prestando atención actualmente a los canales. Cuando alguien se comunique con un usuario que haya especificado un mensaje de `/AWAY`, dicho mensaje será mostrado automáticamente.

## Comandos especiales

Éstos comandos requieren de privilegios especiales para ser ejecutados. I.e. pueden ser ejecutados solo por los moderadores de un canal.

- `/TOPIC [<tópico>]`: Cambia el tópico de un canal. Si no se da un `<tópico>`, muestra el tópico actual.

Por ejemplo:

```
/TOPIC
```

```
Topic for #debian is /msg the bots, NOT the people, FLOOD IN #flood NOT
Have question? JUST ASK. || XFree86 4.3 is in (/msg apt) experimental
--- Topic for #debian set by bob2 at Sat Sep 6 19:45:40
```

```
/TOPIC intento cambiar el topico sin privilegios
```

```
#debian :You need to be a channel operator to do that
```

- `/INVITE <nick> #<nombre del canal>`: Invita un usuario particular al canal especificado.
- `/MODE #<nombre del canal> | <nick> [[+|-]<modo> [<parámetros>]]`: Este es un poderoso comando para control de los operadores sobre el canal y los usuarios.

## Software libre de IRC

- BitchX (<http://www.bitchx.org/>)
- ircII (<http://www.eterna.com.au/ircii/>): Cliente de IRC que funciona exclusivamente en la consola.
- erc (<http://delysid.org/emacs/erc.html>): Paquete de extensión para Emacs que permite usar a éste editor como cliente de IRC.

### 5.1.2. ¿Cómo conectarse a Internet?

#### Utilizando `wvdial`

#### Con una conexión ADSL

### 5.1.3. ¿Que programas de noticias/correo electrónico libres pueden recomendar?<sup>3</sup>

Existen programas de correo/(noticias usenet) sumamente poderosos. Podemos recomendar, entre otros:

---

<sup>3</sup>Propuesta por Jhair Tocancipá.

- Gnus (<http://www.gnus.org>)

Gnus es un programa de noticias para Emacs escrito por Lars Magne Ingebrigtsen (<http://quimby.gnus.org/>) basado en el código fuente de GNUS de Masanobu UMEDA. Es parte de la distribución de oficial Emacs.

Existen muchas razones que hacen de Gnus un programa técnicamente superior a muchos programas de noticias (correo electrónico). Entre otras:

- Casi cualquier aspecto del programa es configurable.
- Puede enviar/recibir correo electrónico.
- Sigue estándares rigurosamente, entre los más importantes:
  - RFC 2822 (<http://www.faqs.org/rfcs/rfc2822.html>): Que describe la estructura de mensajes de correo electrónico.
  - RFC 1036 (<http://www.faqs.org/rfcs/rfc1036.html>): Que describe la estructura de mensajes de noticias (usenet).
  - RFC 2045-2049 (<http://www.faqs.org/rfcs/rfc2045.html>): Describe las llamadas *Multipurpose Internet Mail Extensions (MIME)* que especifican cómo un mensaje debe incluir información en sistemas de caracteres distintos a ASCII (en el cuerpo y encabezado del mensaje), incluir información en otros formatos distintos a texto plano (e.g. imágenes y video) y estructurar el cuerpo de un mensaje en partes distintas (*Multipart*).
  - RFC 2440 (<http://www.faqs.org/rfcs/rfc1991.html>): Que especifica el intercambio de mensajes encriptados mediante PGP.
  - RFC 1730 (<http://www.faqs.org/rfcs/rfc1730.html>): Que especifica la version 4 del protocolo IMAP (versiones anteriores también son soportadas por Gnus). Muchas empresas (desafortunadamente) mantienen servidores de correos basados en productos de Microsoft. La buena noticia es que Gnus puede hablar con esos servidores (sólo si éstos ofrecen sus servicios mediante IMAP).
  - Y muchos otros!
- Está escrito en Emacs-Lisp :-)

Dentro de Emacs puede arrancar Gnus con el comando `M-x gnus RET`

El archivo de configuración de Gnus es por defecto `~/.gnus`. Un ejemplo de configuración mínima que le permite leer/enviar correo electrónico/noticias es:

```
;Método (backend) para leer noticias
(setq gnus-select-method '(nntp "servidor.com"))

;Métodos (backends) adicionales, en éste caso para correo
(setq gnus-secondary-select-methods '((nml "private")))

;Servidor de POP3 (correo de entrada)
(setq mail-sources
  '((file :path "/var/spool/mail/usuario"))
```

```
(pop :server "pop3.servidor.com"
     :user "12345678"
     :password "87654321"))

;Información del usuario
(setq user-full-name "Usuario")
(setq user-mail-address "usuario@servidor.com")

;Servidor SMTP (correo de salida)
(setq smtpmail-default-smtp-server "smtp.servidor.com")

;Función para enviar mensajes de correo
(setq message-send-mail-function 'smtpmail-send-it)
```

- Mutt (<http://www.mutt.org>)
- Sven Guckes slrn (<http://slrn.sourceforge.net/>)
- KMail (<http://kmail.kde.org/>)  
KMail es un programa de correo para KDE<sup>4</sup>.
- Sylpheed (<http://sylpheed.good-day.net/>)  
Sylpheed es un programa de correo basado en la librería GTK+ (<http://www.gtk.org>).
- Mozilla (<http://www.mozilla.org/>)  
Mozilla incluye también un programa de correo muy popular. Actualmente se están llevando a cabo esfuerzos para hacer éste componente más modular (hacerlo una aplicación independiente fácilmente integrable). Más información en <http://www.mozilla.org/projects/thunderbird>.

---

<sup>4</sup>Si alguien usa KMail como su programa de correo, podría incluir aquí algunos consejos para su instalación y uso!

## Capítulo 6

# Software Libre y Educación

### 6.1. Organizaciones en Colombia

#### 6.1.1. ¿Existen organizaciones en Colombia dedicadas a tratar el tema de software libre y la educación?

Si. SLEC (<http://ingenieria.sanmartin.edu.co>) comenzó a trabajar desde el año 2000 y maneja estadísticas de instituciones educativas (<http://ingenieria.sanmartin.edu.co/~slec/estadisticas.html>), comparte experiencias y sirve como espacio para que quien lo desee se pueda vincular o generar proyectos (<http://ingenieria.sanmartin.edu.co/~slec/hospedaje.html>). También tiene enlaces con otras comunidades que buscan cómo mejorar la enseñanza y un listado ([http://ingenieria.sanmartin.edu.co/cgi-bin/slec/slec.cgi/slec.software\\_librered](http://ingenieria.sanmartin.edu.co/cgi-bin/slec/slec.cgi/slec.software_librered)) de programas que pueden ser usados en instituciones educativas.



## Capítulo 7

# Instalación

**7.1. No hay preguntas aún en esta sección.**