



uxi



Revista de Software Libre de la UCI

## “Nuestro 1er Aniversario”



Ya Estamos por todo el “Mundo”

“El buscador, la solución contra el tiempo”

Noticias

Solución de esta edición

“Encriptar el sistema de ficheros en Debian & Ubuntu”

Migración

“Aplicación de la estrategia de migración a Software Libre”

Programación

“Programación de Socket en GNU/Linux con C++.”

Humor Libre

Informática 2007

“CMS Bajo Software Libre”

“El Software Libre y la educación en Colombia”



“El buscador, la solución contra el tiempo”  
pagina 1



1 año

10 Números

“La ciencia tiene derecho a ser libre, a ser respetada, a gobernarse por sí misma”



# Editorial



Estimado Lector:

UXi completa su primer volumen de publicaciones con esta edición especial: **UxiX**. Es especial porque con esta décima publicación nuestra revista alcanza su primer año de existencia. Desde entonces hemos cambiado mucho, lo que comenzó con 3 estudiantes de primer año de Ingeniería en Ciencias Informáticas, miembros de la Comunidad de Software Libre de la UCI, hoy día cuenta con un Grupo Editorial más variado y numeroso. Inicialmente UXi surge como medio de divulgación de las actividades de la comunidad, poco a poco nuestros objetivos se han ido ampliando, de manera que aspiramos a convertirnos en una revista nacional de Software Libre. Apoyar el proceso de migración es y seguirá siendo nuestro principal objetivo.

Estamos conscientes que la tarea no es fácil y cada día debemos ganar en calidad y eficiencia. Lograr el aprecio y respeto de nuestros lectores, colaboradores y amigos son factores imprescindibles para nuestro trabajo.

Esta edición se la dedicamos además al 155 Aniversario del Natalicio de nuestro Héroe Nacional José Martí el pasado 28 de enero, quien luchó hasta la muerte por la libertad y la emancipación del hombre. Usted podrá encontrar al pie de cada página una frase diferente de Martí, todas extraídas del libro "José Martí Aforismos" (2004) del autor Jorge Sergio Balle. Sirva este sencillo homenaje para que lleguen a nuestros lectores algunos pensamientos de nuestro héroe nacional.

En la próxima edición publicaremos un artículo relacionado con los resultados del Festival de Software Libre organizado en la UCI por el Grupo Editorial de UXi y el Proyecto Unicornios para celebrar nuestro Primer Aniversario.

Usted puede suscribirse a UXi a través de la dirección electrónica [softwarelibre@uci.cu](mailto:softwarelibre@uci.cu) con el asunto "Suscribir UXi", así como enviarnos sus comentarios, críticas y sugerencias. Esperamos este número le guste y le sea útil.

Grupo Editorial

**Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre de GNU v1.2 o cualquier otra versión posterior publicada por la FSF; sin Secciones Invariantes, Textos de Cubierta Delantera ni Trasera. Puede consultar una copia de la licencia en:**

<http://www.gnu.org/copyleft/fdl.html>



**“En cada artículo debe verse la mano enguantada que lo escribe, y los labios sin mancha que lo dictan”**



**Jefe Consejo Editorial**  
Abel García Vitier  
avitier@estudiantes.uci.cu

**Editores**  
Jorge Luis Betancourt González  
jlbetancourt@estudiantes.uci.cu

Abel García Vitier  
avitier@estudiantes.uci.cu

**Redactor:**  
Yosbel Brooks Chávez  
ybrooks@estudiantes.uci.cu

**Arte y Diseño:**  
Angel Alberto Bello Caballero  
aabello@estudiantes.uci.cu

David Padrón Álvarez  
dpadron@estudiantes.uci.cu

**Revisión y Corrección:**  
MSc. Clara Gisela Scot Bigñot  
claragisela@uci.cu

MSc. Graciela González Pérez  
gracielagp@uci.cu

**Coordinadores:**  
Evelio Clavel Rosales  
eclavel@estudiantes.uci.cu

Ing. Abel Meneses Abad  
abelma@uci.cu

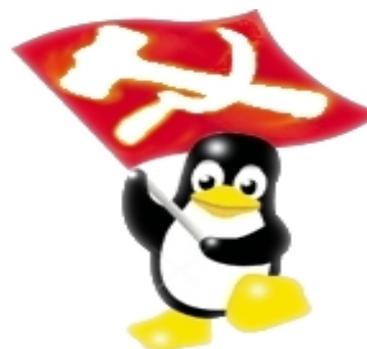
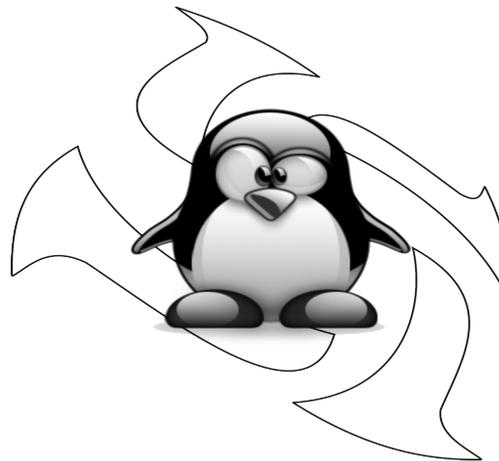
**PROYECTO  
UNICORNIOS**



Servicios Especializados  
para la Migración a SWL



Movimiento de SWL  
Facultad 2



**“Con todos y para el bien de todos”**



# Contenido



## El buscador, la solución contra el tiempo

... Página 1

El mundo del Software Libre ofrece alternativas, libres y además gratis que nos permiten indexar la información de nuestras intranets...



## Noticias

... Página 3



## Solución de esta edición

... Página 4

La protección de datos sensibles tiene la prioridad siempre y cuando el entorno lo exija. El objetivo de este artículo es, dotar al lector de las herramientas necesarias para proteger su sistema mediante la encriptación...



## Humor Libre

... Página 6



## Migración

... Página 7

A finales del curso 2006-2007 y como resultado de una tesis de grado, que acumuló en su contenido todo lo aprendido por los integrantes del Proyecto de Servicios Especializados para la migración a Software Libre se lanzó la Estrategia de Migración hacia Software Libre de la UCI...



## Programación

... Página 8

La programación de sockets muchas veces tiende a ser algo un poco complicado, pero cuando llegamos a entenderla puede en realidad sernos de mucha utilidad. Un socket explicado de manera simple, se puede decir que es una manera de comunicarse con otra computadora o con la nuestra...



## Informática 2007

... Página 12

Como es habitual, ofrecemos a los lectores dos resúmenes de las ponencias presentadas en Informática 2007...



## Evento

Primer Taller Universitario de Blender

... Página 13



# ARTÍCULO PRINCIPAL



Yoandy Pérez Villazón  
yvillazon@estudiantes.uci.cu  
Proyecto UNICORNOS  
Universidad de las Ciencias  
Informáticas

## Resumen:

Este artículo muestra de manera breve la importancia de los buscadores en la web y la necesidad de implantación de estos en lugares donde los volúmenes de datos crecen a tal ritmo que es imposible dominar toda la información que se tiene, además de ello muestra un estudio sobre los diferentes buscadores libres que existen y expone el modo de configuración de uno de ellos que ha sido seleccionado después de una profunda investigación como el más adecuado para su instalación y puesta en marcha.

## Introducción :

La arquitectura de la información y su organización son temas sumamente importantes cuando los volúmenes de esta aumentan exponencialmente, los niveles investigativos actuales de las empresas e instituciones necesitan no sólo formas de organizar bien la información, además de ello necesitan herramientas para encontrarlas fácilmente; empresas como Google o Yahoo brindan dichos servicios pero para ello debemos de entregar nuestra documentación a ellos y además de esto pagar por el servicio.

El mundo del Software Libre ofrece alternativas, libres y además gratis que nos permiten indexar la información de nuestras intranets servidas en un servidor web y encontrarlas rápidamente.

## Desarrollo :

Realmente sería imposible realizar una búsqueda de cualquier información a través de la red de redes, tal proceso se lleva a cabo de esta manera actualmente aquí en la Universidad de las Ciencias Informáticas (UCI) en el servidor de documentación libre montado desde febrero del año 2005 , cuando se iniciaba el proceso de migración tecnológica de Software Privativo a Software Libre en la facultad 10.

A partir del montaje de este servidor y hasta nuestros días no existía un buscador, de ahí surge la tarea de estudiar varias aplicaciones de búsqueda que son capaces dada una pagina inicial y partir de una aplicación araña (spider) comenzar a realizar una búsqueda en un servidor web indexar la información creando un buscador para agilizar los procesos de gestión de la información.

## “El buscador, la solución contra el tiempo”

Han sido estudiadas un total de 5 aplicaciones de las más usadas en internet y una quedó por estudiar debido a su complejidad de instalación y configuración, recomendamos para una institución profesional realizar un estudio sobre NUTCH, un buscador muy prometedor, las principales aplicaciones estudiadas se muestran a continuación:

### Site Search Pro [1]:

*Descripción:* Sistema comercial para añadir un motor de búsqueda a un sitio web. Disponible tanto en ASP como en PHP. Dispone de varias versiones con funcionalidades más o menos avanzadas a precios distintos.

*Licencia:* Licencia comercial

*Plataformas:* Multiplataforma

*Idiomas:* Inglés

*Experiencia personal:* Este sistema no lo hemos instalado debido a que no lo poseemos, es comercial y su código no está accesible en la web.

### TSEP: The Search Engine Project [2]:

*Descripción:* TSEP es un motor de búsqueda para un sitio web, creado en PHP. Puedes poner un botón "Buscar en este Sitio" y permitir que la gente encuentre fácilmente lo que están buscando.

*Licencia:* Software Libre, GPL

*Plataformas:* Multiplataforma

*Idiomas:* Multidioma

*Experiencia personal:* Este sistema no pudo ser instalado satisfactoriamente debido a problemas en la implementación del mismo , se le fueron instalados todos los requerimientos que eran necesarios pero por errores de la implementación del mismo no funcionó correctamente.

### PHPDig [3]:

*Descripción:* Spider o robot y motor de búsqueda creado en PHP y con base de datos MySQL. Crea un glosario con palabras encontradas en las páginas indexadas.

*Licencia:* Software Libre, GPL

*Plataformas:* Multiplataforma

*Idiomas:* Inglés

*Experiencia personal:* Sistema más trabajado, instalado correctamente e indexa correctamente los formatos html y txt , pero posee grandes problemas con la indexación de otros formatos y problemas en la base de datos.

#### **Blasten blt-SEARCH [4]:**

*Descripción:* Un buscador adaptado a usted y sus visitantes. Utilizando un sistema escaneador de enlaces, basado en la tecnología PageRank con un poder de penetración superior a los 10 puntos, logra la búsqueda de imágenes.

*Licencia:* Freeware

*Plataformas:* multiplataforma

*Idiomas:* Español

*Experiencia personal:* Sistema instalado satisfactoriamente , mejor interfaz de todos lo que se han probado , libre, además en español , pero código muy confuso , a la hora de la indexación presenta un problema que impide este proceso, es muy configurable, tal vez con conocimientos de PHP se puedan solucionar los problemas que este posee.

#### **Htdig [5]:**

*Descripción:* Sistema creado utilizando CGI's , son un conjunto de aplicaciones programadas en C++ , encargadas de buscar e indexar información de la Web , modular y posee complementos adicionales programados en lenguajes como C,C++ y Perl.

*Licencia:* Freeware

*Plataformas:* Multiplataforma

*Idiomas:* Multilingüa

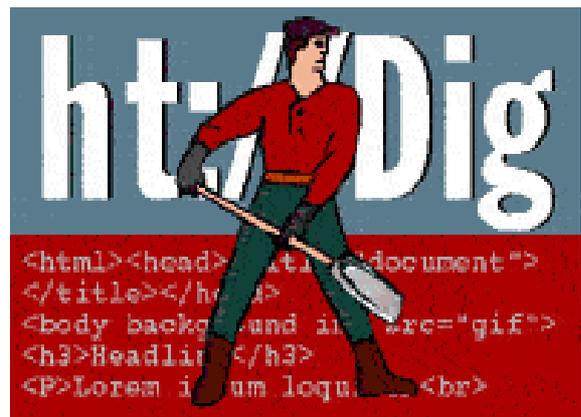
*Experiencia personal:* Sistema instalado satisfactoriamente , funcionalidades trabajando de manera total , probado 100% sobre el entorno GNU/Linux Debian 4.0 con la versión 3.1.6 de este producto compilada para esta arquitectura, sistema elegido para hacer el buscador del servidor de documentación libre de la UCI.

#### **Htdig, el buscador seleccionado**

Durante este proceso de investigación hemos seleccionado el Htdig debido a que se ajusta a nuestras características de una mejor manera, a continuación mostramos algunas ventajas a modo de características de este software:

- Es Software Libre completamente liberado bajo GPL.
- Está programado en C++ utilizando CGI's.
- La indexación es mucho más rápida.
- La búsqueda de datos es mucho menor que los otros que están programados en PHP.
- Es un sistema modular lo que le permite adicionar funcionalidades.
- Interfaz para las búsquedas configurable.
- Capaz de indexar muchos formatos, pdf, html, doc, txt, etc.

Este artículo ha sido el resultado de una investigación sobre los buscadores libres más importantes disponibles en internet, se han probado y estudiado las características de cada uno de ellos, quedando como constancia una guía completa de instalación del htdig [6].



#### **Referencias:**

- [1] Site Search Pro  
<http://www.desarrolloweb.com/scripts/site-search-pro-motor-busqueda.html>
- [2] TSEP: The Search Engine Project  
<http://www.desarrolloweb.com/scripts/step-motor-busqueda-php.html>
- [3] PhpDig  
<http://www.desarrolloweb.com/scripts/phpdig-motod-busqueda-php.html>
- [4] Blasten blt-SEARCH  
<http://www.desarrolloweb.com/scripts/blasten-motor-busqueda.html>
- [5] Htdig  
<http://www.desarrolloweb.com/scripts/blasten-motor-busqueda.html>
- [6] Yoandy Perez Villazon, Htdig, Indexando la red, 2007  
[http://10.128.50.121/Documentacion/Publicaciones/15\\_Noviembre\\_2007/](http://10.128.50.121/Documentacion/Publicaciones/15_Noviembre_2007/)





## LliureX ahorro de 15 millones de euros

La nueva versión cuenta con un mejor soporte de hardware y versiones actualizadas. La directora general de Innovación Tecnológica Educativa, Sofía Bellés, ha manifestado, durante su participación en la II edición de las jornadas SourcePyme, que "la distribución LliureX en poco más de dos años ha supuesto un gran avance y un ahorro de 15 millones de euros para todos los valencianos".

## Red Hat lanza programa Advance Business Partner

En el marco de la convención Imagina+ de Team, Redhat presentó a distribuidores el programa Advance Business Partner mediante el cual busca el apoyo del canal para facilitar la adopción de Linux en las organizaciones.

## ¿Por qué Linux es más seguro que Windows?

La reputación de Linux como sistema operativo seguro es legendaria, y siempre se ha mantenido que en general las vulnerabilidades y fallos de seguridad eran más reducidos y menos importantes que los que aparecen en Windows.

- Mejores herramientas de gestión
- Mejor configuraciones de serie
- Diseño modular
- Mejores herramientas para la protección contra ataques Zero-Day
- Arquitectura Open Source
- Entorno muy diverso

## Gutsy Gibbon: el escritorio de Linux más fácil

El escritorio de Linux siempre ha sido un reto; por eso es que la versión del sistema operativo Linux presentada por Ubuntu creó tanta sensación cuando debutó. Ubuntu 7.10, llamado Gutsy Gibbon, sobrepasa a sus predecesores porque ofrece mejores gráficos, instalación simplificada del software y mejor manejo de los archivos de multimedia.

## Torvalds asegura que Google impulsará la llegada de Linux en los móviles en 2008

El sistema operativo Linux, que hasta ahora no ha tenido demasiado éxito en los teléfonos móviles, se verá impulsado gracias a la iniciativa Open Handset y a Android de Google, según afirmó el creador de Linux, Linus Torvalds.

Google está trabajando con Motorola y otros grandes operadores de telecomunicaciones, incluyendo T-Mobile, Telefónica y Qualcomm, para desarrollar una plataforma de 'software' libre llamada Android para servicios móviles.

## Tres ventajas distintivas de Firefox

### 1. Seguridad

Firefox es la forma más segura de usar Internet

### 2. Personalización

Cuando hablamos de navegar por la web, creemos que un navegador no sirve para todos

### 3. 100% software orgánico

Firefox es bueno para vos: es sinónimo de apertura, innovación y libertad en Internet.

## Bazaar 1.0 ¿adiós a Subversion?

Según Mark Shuttleworth, "Bazaar está diseñado para equipos globales de desarrolladores colaborativos" y tiene la ventaja sobre los sistemas similares centralizados como Subversion o CVS en que no requiere un servidor dedicado. Así como antes Subversion reemplazó a CVS como el sistema de control de versiones preferido por los proyectos de código abierto ¿Será otra vez KDE el primero en hacer el gran cambio que otros imitarán después?

## Nova sale oficialmente en la UCI!

La tan esperada distribución UCI hace su primera salida oficial para estudiantes de la UCI. En la noticia van contenidas las aplicaciones de las que dispone este instalador y el LiveCD para su uso y prueba en todo el campus universitario. Disponibles también los enlaces para su descarga en el sitio de ISOS.

## Gentoo 2007.1 cancelado, 2008.0 confirmado

Y sin duda como una parte más en la actual crisis en Gentoo muchos ya habrán caído en cuenta de que durante todo el año pasado sólo hubo una actualización en esa distribución: Gentoo 2007.0, lanzada a principios de Mayo. Ahora se anunció de que no habrá ningún Gentoo 2007.1, a pesar del vasto número de vulnerabilidades de seguridad entre Septiembre y Diciembre del 2007. En su lugar, el equipo de *Release Engineering de Gentoo* decidió *anar todos sus esfuerzos para la próxima versión 2008.0.*

## 1er Aniversario de UXi. Festival de Software Libre, en grande!!!

El jueves 31 de Enero se estará celebrando en el docente 3 el 1er Festival de Software Libre, auspiciado por el proyecto UNICORNIOS, con la dirección del equipo editorial de la revista UXi. Conferencias durante todo el día, y exposiciones desde la noche del 30 de enero el evento permitirá conocer lo más importante desarrollado durante el 2007 en materia de SWL, y nos acercará a las perspectivas del 2008.



Eiger Mora Moredo  
[emora@estudiantes.uci.cu](mailto:emora@estudiantes.uci.cu)  
 Yunier Vega Rodríguez  
[yvrodriguez@estudiantes.uci.cu](mailto:yvrodriguez@estudiantes.uci.cu)  
 Proyecto Unicornios  
 Universidad de las Ciencias  
 Informáticas

A medida que se desarrollan herramientas, muchas de estas pueden utilizarse con objetivos diferentes para los que inicialmente fueron creadas, un vivo ejemplo es el potente Knoppix o cualquier otra compilación LiveCD de GNU/Linux, que permiten fácilmente a un tercero acceder a la información existente en una PC y modificarla a su antojo.

La protección de datos sensibles tiene la prioridad siempre y cuando el entorno lo exija. El objetivo de este artículo es, dotar al lector de las herramientas necesarias para proteger su sistema mediante la encriptación.

Este artículo está basado en la configuración de Debian Etch con kernel linux-image-2.6.18-4-686.

El primer paso es hacer todas las salvadas necesarias de los datos sensibles.

## Instalación de sistema base

Lo ideal sería una instalación desde cero con el particionamiento más cómodo al usuario. No obstante, si están creadas las particiones de antemano no tiene problema alguno.

El en caso de estudio se presenta el siguiente particionamiento:

boot	/dev/sda1	50MB
root_inicial	/dev/sda2	512MB
root_final	/dev/sda3	10GB
home	/dev/sda5	70GB

Se hará una instalación base ("sistema mínimo") en la que se definirá:

- La partición BOOT con el sistema de fichero deseado (ext3, reiserfs, xfs) que no se encriptará.
- La partición ROOT\_INICIAL desde la que se harán las configuraciones y al final del día será la partición SWAP.
- Las particiones ROOT\_FINAL y HOME que se marcarán como "no utilizar".
- Otras particiones que el usuario desee crear que se marcarán como "no utilizar".

*Nota:* No se define ninguna partición SWAP.

## Encriptar el Sistema de Ficheros en Debian & Ubuntu

### Encriptando

Instalado todo lo necesario, empezamos a encriptar las particiones ROOT\_FINAL y HOME para esto se hace uso de la herramienta cryptsetup.

```
# cryptsetup -y -c aes-cbc-essiv:sha256 \
-h sha512 luksFormat /dev/sda3

# cryptsetup -y -c aes-cbc-essiv:sha256 \
-h sha512 luksFormat /dev/sda5
```

Para explicar las opciones:

- y verificar contraseña
- c especificación de Sistema de cifrado
- h hash a usar para la contraseña
- luksFormat crea una partición LUKS y establece la contraseña inicial

El sistema le advertirá al usuario que los datos de la partición serán borrados y le pedirá que escriba YES como confirmación, seguido le pedirá la contraseña y la verificación de la misma.

```
# cryptsetup luksOpen /dev/sda3 root
# cryptsetup luksOpen /dev/sda5 home
```

- luksOpen Abre la partición luks y establece el nombre de un nuevo dispositivo /dev/mapper/
- root, home nombre del nuevo dispositivo: /dev/mapper/root & /dev/mapper/home respectivamente

Le pedirá la contraseña al usuario, el comando tendrá éxito luego de una correcta verificación. A continuación se da formato a los dispositivos creados y se montan

```
# mkfs -t ext3 /dev/mapper/root
# mkfs -t ext3 /dev/mapper/home
# mount /dev/mapper/root /mnt
# mkdir /mnt/home
# mount /dev/mapper/home /mnt/home
```

Se copia el contenido de ROOT\_INICIAL a la partición ROOT encriptada(/dev/mapper/root):

```
# cp -axv / /mnt
```

## CHROOT: Configuraciones finales

El sistema base en /dev/mapper/root y /dev/mapper/home ya está encriptado pero es necesario configurarlo para que logre cargar, por esa razón el siguiente paso es un chroot.

Se montan los dispositivos para acceder a ellos desde el chroot:

```
# mount -o bind /dev/ /mnt/dev
```

Se realiza en chroot y los montajes necesarios:

```
# chroot /mnt /bin/bash
# mount /boot
# mount /proc
# mount /sys
```

Se editan los siguientes ficheros de configuración:

/etc/fstab  
/etc/kernel-img.conf  
/etc/crypttab  
/boot/grub/menu.lst

- El /etc/fstab para estas particiones quedaría así:

```
#<file system> <mount point> <type> <options> <dump> <pass>
/dev/mapper/root / ext3 defaults 0 0
/dev/mapper/home /home ext3 defaults 0 0
```

y ahora al fichero /etc/crypttab:

- En el fichero /etc/kernel-img.conf se agrega la siguiente línea:

```
ramdisk = /usr/sbin/mkinitramfs
```

- En el /etc/crypttab se escribe lo siguiente:

```
# <target name> <source device> <key file> <options>
root /dev/sda3 none luks
home /dev/sda5 none luks
```

- En el archivo /boot/grub/menu.lst se pone la línea:

```
# kopt=root=/dev/mapper/root devfs=mount ro
```

**Nota:** Nótese que el # se mantiene al inicio de la línea.

Se reconfigura el initrd.img del kernel

```
# dpkg-reconfigure linux-2.6.18-4-686
```

Ahora, se desmonta todo, salimos de chroot y reiniciamos

```
# umount -a
# exit
# reboot
```

## Encriptando la SWAP

Ahora se trabaja desde el sistema encriptado, pero falta la swap, para hacerlo seguimos los pasos anteriores:

```
# cryptsetup -y -c aes-cbc-essiv:sha256 \
-h sha512 luksFormat /dev/sda2
# cryptsetup luksOpen /dev/sda2 swap
# mkswap -v1 /dev/mapper/swap
```

Agregamos la línea correspondiente al fichero /etc/fstab

```
#<file system> <mount point> <type>
/dev/mapper/swap none swap
<options> <dump> <pass>
sw 0 0
```

y ahora al fichero /etc/crypttab:

```
# <target name> <source device>
swap /dev/sda2
<key file> <options>
none luks
```

## Ultimando detalles

Ya tenemos nuestro sistema de ficheros encriptado pero hay que poner la contraseña para cada partición en la carga del sistema, para evitar esto se pueden definir ficheros llaves para las particiones home y swap.

Se hace el directorio destino(en el caso de estudio: cryptkeys)

```
# mkdir /etc/cryptkeys
```

Se genera el fichero llave:

```
# cryptsetup luksAddKey /dev/sda2 /etc/cipherkeys/swap
# cryptsetup luksAddKey /dev/sda5 /etc/cipherkeys/home
```

Se le cambian los permisos a cada fichero para que solo root los pueda leer

```
# chmod 600 /etc/cryptkeys/*
```

Se modifican las líneas correspondientes en el fichero /etc/crypttab

```
# <target name> <source device> <key file> <options>
home /dev/sda5 /etc/cryptkeys/home luks
swap /dev/sda2 /etc/cryptkeys/swap luks
```

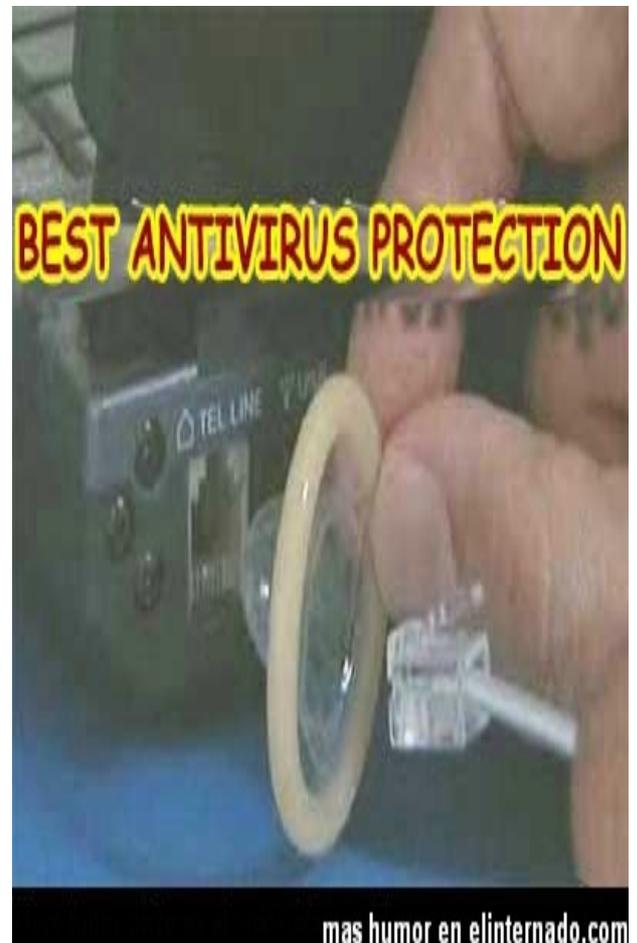
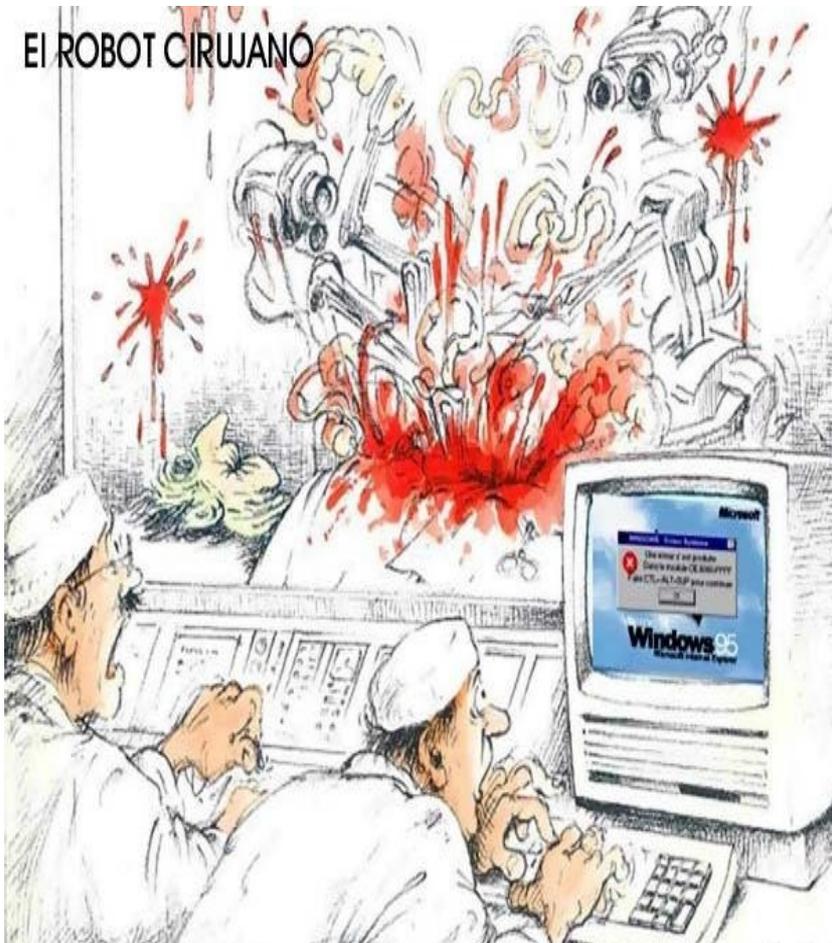
Se reconfigura el initrd.img

```
# dpkg-reconfigure linux-2.6.18-4-686
```

Se reinicia y listo, a instalar lo que le falta al desnudo sistema base inicial, para disfrutar de GNU/Linux sin temor a que los datos sensibles sean comprometidos.



## HUMOR LIBRE



“Los débiles parafrasean: los poderosos, crean”



# MIGRACION



Yoandy Pérez Villazón  
[yvillazon@estudiantes.uci.cu](mailto:yvillazon@estudiantes.uci.cu)  
Proyecto UNICORNIOS  
Universidad de las Ciencias  
Informáticas

## Resumen:

Este artículo es una recopilación de las principales aplicaciones de la metodología de migración hacia Software Libre creada en la UCI, identifica los lugares y eventos en el que ha sido expuesta dicha guía por los distintos miembros de la Comunidad de Software Libre de la UCI. Esta guía fue elaborada por el Ing. Ramón Paumier Samón como parte de su trabajo de curso del pasado año y recopila la información de los avances del proyecto durante su tiempo de trabajo.

## ¿Que es la guía de migración ?

A finales del curso 2006-2007 y como resultado de una tesis de grado, que acumuló en su contenido todo lo aprendido por los integrantes del Proyecto de Servicios Especializados para la migración hacia Software Libre se lanzó la Estrategia de Migración hacia Software Libre de la UCI, la cual tuvo desde el mismo comienzo de su elaboración gran impacto en la soberanía tecnológica en los lugares donde fue aplicada.

Esta estrategia consiste en la acumulación de conocimiento de personas dedicadas a la noble tarea del estudio de las tecnologías libres con el afán de aplicarlas a nuestra universidad e instituciones que la necesitan para migrar a Software Libre, contiene un resumen de las experiencias de los principales intentos de migración de distintas universidades y otras organizaciones del mundo estudiando los principales éxitos y fracasos de estas para que no sean cometidas en el futuro, constituye una guía donde se estudian los principales procesos de la UCI y cómo se debe emprender la migración de los mismos de una manera que no existan grandes "traumas" en el desarrollo del cambio. Se describen los distintos pasos por los que se debe transitar para lograr una migración de manera eficiente proponiendo alternativas basadas en modelos. Se exponen además los principales momentos por los que transcurre el proceso y qué hacer en cada momento, resaltando el apoyo que debe brindar la institución para el mismo, la formación de recursos humanos.

También es descrita una propuesta en cuanto a las herramientas alternativas de código abierto libres a utilizar con la implantación de los nuevos Sistemas GNU/Linux, contiene un conjunto de plantillas

## “Aplicación de la estrategia de migración a Software Libre”

necesarias para realizar un levantamiento de requisitos asociados a la tecnología previo a la migración.

### Lugares donde ha sido aplicada la guía de migración

Aunque la guía de migración fue elaborada inicialmente para la UCI ha sido aplicada en diversos lugares dentro y fuera de la universidad.

- Migración del grupo de multimedia de la UCI mediante la capacitación del personal y cambio del sistema operativo.
- Asesoría de migración y capacitación a la facultades 1, 2 y 7.
- Asesoría de migración a la facultades 4 y 6.
- Asesoría de migración al MES e instalación de varios servicios.
- Asesoría de migración a 1era y B (Oficinas adjuntas al Consejo de Estado) y capacitación.
- Asesoría de migración al centro Martín Luther King.
- Exposición del proceso y experiencias en reunión de trabajo con el grupo FORMATUR.
- Experiencias de la guía aplicándose al proyecto MINPPAL con Venezuela.
- Aplicación de la guía en la MiniUCI de Granma.
- Aplicación de la guía en la MiniUCI de Artemisa.

Esta guía de migración sirve de base a tesis de grado que se aplicarán en distintos lugares del país como empresas y escuelas a distintos niveles. Fue aplicada durante el proceso de formación de profesores de los distintos Institutos Politécnicos de Informática del país.



### Referencias:

Ing. Ramón Paumier Samon, Metodología para la Migración a Software Libre de la Universidad de las Ciencias Informáticas.UCI, Ciudad de la Habana, 2007

[http://10.128.50.121/Documentacion/Publicaciones/09\\_Mayo\\_2007/](http://10.128.50.121/Documentacion/Publicaciones/09_Mayo_2007/)  
(Sólo disponibles desde la UCI)

Marcos Luis Ortiz Valmaseda, Yoandy Pérez Villazón. Elementos para una Migración a Software Libre, Reunión de Migración de las escuelas de Formación del Turismo, 2006



# PROGRAMACIÓN



Dayron Pérez Roldán  
[droidan@estudiantes.uci.cu](mailto:droidan@estudiantes.uci.cu)  
Proyecto UNICORNIO  
Universidad de las Ciencias  
Informáticas.

## Resumen

Los Sockets representan una interfaz de comunicación que ofrece un mecanismo de comunicación general entre dos procesos cualesquiera, que pertenezcan a un mismo sistema o a dos sistemas diferentes. En este artículo se abordarán los aspectos más importantes de la programación de sockets, utilizando Code::Blocks como entorno de Programación para escribir el código, el cual podría ser escrito en cualquier IDE (Entorno de Desarrollo Integrado) de programación, que soporte C++. Este artículo no pretende enseñar a programar socket, tan sólo es una panorámica de la programación de socket ilustrada con pequeños ejemplos.

## Introducción

La programación de sockets muchas veces tiende a ser algo complicado, pero cuando llegamos a entenderla puede ser en realidad de mucha utilidad. Un socket, explicado de manera simple, se puede decir que es una manera de comunicarse con otra computadora o con la nuestra. Para ser más exacto es una manera de comunicarse usando descriptores de ficheros estándar de Unix.

En Unix, todas las acciones de entrada y salida son desempeñadas escribiendo o leyendo en uno de estos descriptores de fichero, los cuales son simplemente un número entero, asociado a un fichero abierto que puede ser una conexión de red, un terminal, o cualquier otra cosa (recuerde uno de los primeros conceptos que aprendemos al usar UNIX "En un sistema UNIX, todo es un fichero").

Muchas veces se programan aplicaciones de escritorio y gustaría que las mismas pudieran comunicarse entre sí, desde diferentes puntos en toda una intranet o incluso internet; como por ejemplo un pequeño chat para conversar con los amigos, o quizás se desarrolla un juego y quiere aportarle la habilidad de ser jugado entre dos personas desde diferentes extremos de la red, etc.

Pueden surgir muchos usos para los sockets y todo se va a resumir en que en ellos reside la herramienta básica orientada al paso de mensajes entre procesos para programar en red sobre TCP/IP.

## “Programación de Socket en Linux con C++”

Ahora bien, sobre los diferentes tipos de sockets se puede hacer mención de tres de ellos:

- SOCK\_STREAM (Sockets de Flujo)
- SOCK\_DGRAM (Sockets de Datagramas)
- SOCK\_RAW (Sockets Puro, solo usados por el root)

No quiere decir que estos sean los únicos sockets que existen; no obstante en este artículo sólo se hablará de los sockets de flujo y los sockets de datagramas.

### Sockets de Flujo:

Generalmente no poseen errores. Por ejemplo si se enviara por el socket tres objetos "A, B, C" llegarán a su destino en el mismo orden. Estos sockets usan TCP (Transmission Control Protocol) y es este protocolo el que asegura el orden de los objetos durante la transmisión. Un ejemplo de aplicación que usa socket de flujo es **telnet**.

### Sockets de Datagramas:

Estos usan UDP ("User Datagram Protocol"), y no necesitan de una conexión accesible como los Sockets de Flujo, tan sólo se construirá un paquete de datos con información sobre su destino y se enviará afuera, sin necesidad de una conexión. Un ejemplo de aplicaciones que usan socket de datagramas son **tftp** y **bootp**.

Mucho más podría explicarse aquí sobre estas dos clases de sockets, pero esto es suficiente como para captar el concepto básico de socket. Entender qué es un socket y algo sobre estos dos tipos de sockets de Internet es un buen comienzo, pero lo más importante será saber cómo trabajar con ellos.

### Desarrollo

En la programación con sockets existen estructuras de datos que son las encargadas de almacenar la información del socket así como la descripción del nodo remoto entre otras funcionalidades. Una estructura de dato es un tipo de dato que puede contener otros tipo de datos y los cuales son agrupados en un único tipo definido por el usuario.

Veamos entonces las siguientes estructuras:  
Contiene información del Socket.

```
struct sockaddr
{
    unsigned short sa_family;
    char sa_data[14];
};
```

Hace referencia a los elementos del Socket.

```
struct sockaddr_in
{
    short int sin_family;
    unsigned short int sin_port;
    struct in_addr sin_addr;
    unsigned char sin_zero[8];
};
struct in_addr
{
    unsigned long s_addr;
};
};
```

Almacena información del nodo remoto. Esto es en el cliente.

```
struct hostent
{
    char *h_name;
    char **h_aliases;
    int h_addrtype;
    int h_length;
    char **h_addr_list;
#define h_addr h_addr_list[0]
};
```

Esta definida en `#include <netdb.h>`.

Las demás librerías que se deben incluir para programar socket son :

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
```

## Conversiones

Existen dos tipos de ordenamiento de bytes: bytes más significativos, y bytes menos significativos. Este es llamado "Ordenación de Bytes para Redes", algunas máquinas utilizan este tipo de ordenación para guardar sus datos internamente. Existen dos tipos a los cuales seremos capaces de convertir: short y long.

htons() : "Nodo a variable corta de Red"  
htonl() : "Nodo a variable larga de Red"  
ntohs() : "Red a variable corta de Nodo"  
ntohl() : "Red a variable larga de Nodo"

## Direcciones IP

Existen algunas funciones que ayudarán a manipular direcciones IP, como son `inet_addr()` y `inet_ntoa()`.

Por un lado, la función `inet_addr()` convierte una dirección IP en un entero largo sin signo (**unsigned long int**), por ejemplo:

```
dest.sin_addr.s_addr=inet_addr("10.33.3.6")
```

Por otro lado, `inet_ntoa()` convierte a una cadena que contiene una dirección IP en un entero largo. Por ejemplo:

```
char *ip;
ip=inet_ntoa(dest.sin_addr);
printf("La dirección es: %s\n",ip);
```

**Nota:** La función `inet_addr()` devuelve la dirección en formato de Ordenación de Bytes para Redes por lo que no necesitaremos llamar a `htonl()`.

Vistos ya elementos de importancia con respecto a socket se pueden mencionar las llamadas al sistemas mas importantes.

## Funciones Importantes en el uso de Socket

**Socket:** Devuelve un descriptor de socket, el cual podrá usarse luego para llamadas al sistema.

```
#include <sys/types.h>
#include <sys/socket.h>
int socket(int domain,int type,int protocol);
```

**Bind:** Su función esencial es asociar un socket con un puerto (de nuestra máquina).

```
#include <sys/types.h>
#include <sys/socket.h>
int bind(int fd, struct sockaddr *my_addr ,int addrlen);
```

**Connect:** Se usa para conectarse a un puerto definido en una dirección IP.

```
#include <sys/types.h>
#include <sys/socket.h>
int connect(int fd, struct sockaddr *serv_addr, int addrlen);
```

**Listen:** Se usa para esperar conexiones entrantes. Después de llamar a `listen()`, se deberá llamar a `accept()`, para así aceptar las conexiones entrantes.

```
#include <sys/types.h>
#include <sys/socket.h>
int listen(int fd,int backlog);
```

**Accept:** Se usa para conseguir establecer la conexión entrante.

```
#include <sys/types.h>
#include <sys/socket.h>
int accept(int fd, void *addr, int *addrlen);
```

La secuencia resumida de llamadas al sistema es:

```
socket()
bind()
listen()
accept()
```

Ahora se hablará de las funciones que se usan para comunicarse a través de socket.

**Send:** Se usa para enviar datos.

```
#include <sys/types.h>
#include <sys/socket.h>
int send(int fd, const void *msg, int len, int flags);
```

**Recv:** Se usa para recibir datos.

```
#include <sys/types.h>
#include <sys/socket.h>
int recv(int fd, void *buf, int len, unsigned int flags);
```

Esta función send y recv se usan para sockets de flujo y socket de datagramas conectados. Puesto que los sockets de datagramas no están conectados a una máquina remota, antes de poder enviar el paquete debemos aportar la dirección de destino. A continuación se abordarán las funciones que se usan para sockets de datagramas desconectados.

**Sendto:** Se usa para enviar datos.

```
int sendto(int sockfd, const void *msg, int len, unsigned int flags, const struct sockaddr *to, int tolen);
```

**Recvfrom:** Se usa para recibir datos.

```
int recvfrom(int sockfd, void *buf, int len, unsigned int flags, struct sockaddr *from, int *fromlen);
```

De nuevo es igual, pero con dos elementos más.

Y por último la función que sirve para cerrar la conexión del descriptor de socket.

**Close:** Se usa para cerrar la conexión con el nodo remoto.

```
close(sockfd);
```

Existen muchísimas funciones que se usan en el manejo de socket, pero estas se las dejamos a usted para que las pueda investigar si le interesa el tema. Hasta este punto sólo se ha hablado de las herramientas que provee el sistema para programar socket, ahora se mostrará un pequeño ejemplo de un cliente-servidor con socket.

## Servidor

```
#include <iostream>
using namespace std;
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#define PORT 3550 /* El puerto que será abierto */

int main()
{
    int fd, fd2; /* los ficheros descriptores */
    struct sockaddr_in server;
    struct sockaddr_in client;
    socklen_t sin_size;
    int BACKLOG=2; /* El número de conexiones permitidas */

    if ((fd=socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        printf("error en socket()\n");
        exit(-1);
    }
    server.sin_family = AF_INET;
    server.sin_port = htons(PORT);
    server.sin_addr.s_addr = INADDR_ANY;
    bzero(&(server.sin_zero), 8);
    if(bind(fd, (struct sockaddr*)&server, sizeof(struct sockaddr))== -1) {
        printf("error en bind() \n");
        exit(-1);
    }
    if(listen(fd, BACKLOG) == -1) {
        printf("error en listen()\n");
        exit(-1);
    }
    while(1) {
        sin_size=sizeof(struct sockaddr_in);
        if ((fd2 = accept(fd, (struct sockaddr*)&client, &sin_size))== -1) {
            printf("error en accept()\n");
            exit(-1);
        }
        printf("Se obtuvo una conexión desde %s\n", inet_ntoa(client.sin_addr) );
        send(fd2, "Bienvenido a mi servidor.\n", 22, 0);
        close(fd2);
    }
    return 0;
}
```

## Cliente

```
#include <iostream>
using namespace std;
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

#define PORT 3550

#define MAXDATASIZE 100

int main(int argc, char *argv[])
{
int fd,numbytes; /*ficheros descriptores */
char buf[MAXDATASIZE];
struct hostent *he;
struct sockaddr_in server;
if (argc !=2) {
printf("Uso: %s <Dirección IP>\n",argv[0]);
exit(-1);
}
if ((he=gethostbyname(argv[1]))==NULL){
printf("gethostbyname() error\n");
exit(-1);
}
if ((fd=socket(AF_INET, SOCK_STREAM, 0))==
-1){
printf("socket() error\n");
exit(-1);
}
server.sin_family = AF_INET;
server.sin_port = htons(PORT);
server.sin_addr = *((struct in_addr *)he-
>h_addr);
bzero(&(server.sin_zero),8);
if(connect(fd, (struct sockaddr *)&server,
sizeof(struct sockaddr))== -1){
printf("connect() error\n");
exit(-1);
}
if ((numbytes=recv(fd,buf,MAXDATASIZE,0)) ==
-1){
printf("Error en recv() \n");
exit(-1);
}
buf[numbytes]='\0';
printf("Mensaje del Servidor: %s\n",buf);
close(fd);
}
```

## Conclusiones

De manera general se han abordado algunos rasgos principales del tema de la programación de socket, los cuales resultan de suma importancia para aquellos que desean hacer que sus aplicaciones se comuniquen entre sí.



Tomado del Portal de Software Libre de la UCI

Visítenos en:

<http://softwarelibre.uci.cu>



<http://documentacionlibre.prod.uci.cu/>



Listo el repo binario de  
NOVA



# INFORMÁTICA 2007



**Julio César Hernández García**  
[juliochg@gmail.com](mailto:juliochg@gmail.com)  
Escuela de Ingeniería  
Universidad Valle del Momboy  
Trujillo, Venezuela

**Yenni A. Méndez Alegría**  
[ymendal@unicauca.edu.co](mailto:ymendal@unicauca.edu.co)  
July E. Jiménez Orjuela  
[jejimenez@unicauca.edu.co](mailto:jejimenez@unicauca.edu.co)  
Universidad del Cauca  
Colombia

## CMS BAJO SOFTWARE LIBRE

El objetivo de esta investigación, es explicar todo lo relacionado con un CMS (Contents Management System) un sistema manejador de contenido, el cual permite tener información en la Web sin mucho esfuerzo, abarca desde la explicación de la Internet y como hoy en día los avances tecnológicos no llevan al uso de esta herramienta, incluye las maneras de poseer una pagina o el contenido necesario para difundir una información, sin saber acerca de la programación o lenguaje WEB, hasta los tipos de CMS existentes. Hoy en día vienen contruidos completamente y en muy diferentes lenguajes pero todos con un fin específico, es por ello que se muestran las diferentes vistas que puede tener un CMS, como son las partes de administración y usuario. La administración de un CMS es completamente gráfica ya que accedes a ella con un respectivo login y password, que se han creado anteriormente, y la vista de usuario que son los que verán el contenido que coloquemos en esta página, el uso de CMS bajo software libre nos ahorra tiempo y trabajo, también son aquellos que tienen la licencia GPLv2 y están protegidos bajo esa licencia, que no tendrá problemas al ser usada por que no estarán incurriendo en ningún delito, la instalación de CMS es fácil debido a que siempre tendrá un sistema de ayuda o un manual dentro del código que vamos a utilizar.

Las configuraciones necesarias como crear y adecuar la base de datos, agregar usuarios y colocar contraseñas, se harán de modo gráfico, el uso de herramientas libres sin mucho esfuerzo y sin mucho conocimiento acerca de lenguaje WEB, nos da la facilidad de tener una página a nuestro alcance, con muchas versatilidades que nos pueden ofrecer y cubrir nuestras necesidades, y adaptarlas para lo que queremos.

## EL SOFTWARE LIBRE Y LA EDUCACION EN COLOMBIA

A la hora de incorporar tecnologías de la información y de las comunicaciones en la educación, se presentan diversos inconvenientes, uno de los más importantes es el costo. Los costos asociados a infraestructura de redes, conectividad, hardware, software, capacitación del personal, instalación y mantenimiento pueden ser una limitante especialmente para países en desarrollo como Colombia.

De los costos mencionados, el Hardware y el Software son los más significativos, sin embargo el costo de la infraestructura de hardware va en descenso mientras que algunas empresas desarrolladoras y comercializadoras de software imponen costos a la Licencia del mismo cada vez más altos.

Esto a su vez ha influido en el aumento excesivo de los índices de piratería de software en el mundo. Sin embargo existe una alternativa que puede disminuir las barreras de acceso a las TIC's a través de una significativa reducción en los costos del software, se trata del Software Libre.



## Ponencias disponibles en:

[http://10.128.50.121/Documentacion/Publicaciones/06\\_Febrero\\_2](http://10.128.50.121/Documentacion/Publicaciones/06_Febrero_2)  
(Sólo disponibles desde la UCI)



<http://www.debiancuba.org>  
**Debian Cuba**

Comunidad de usuarios de Debian GNU/Linux en Cuba

**“Que la enseñanza científica vaya, como la sabia en los árboles,  
de la raíz al tope de la educación pública”**



## BlenderUCi

Primer Taller Universitario

### “Primer Taller Universitario de Blender”

El crecimiento acelerado del software libre no solo ha involucrado a los usuarios de estaciones de escritorio – después de que conquistara casi por completo el mercado de servidores – su alcance ha tocado también un área sensible de la informática: el Diseño de Gráficos por Computadora, en lo que a muchos de sus detractores ha tomado por sorpresa. Blender, una herramienta nacida al calor de este movimiento Universal, es el exponente más representativo en la creación de contenidos tridimensionales, que cuenta además con una comunidad activa a nivel Internacional que le ha merecido el reconocimiento de diseñadores y artistas de la plástica. Es por esta razón que en nuestra Universidad, inmersa en un proceso paulatino de migración tecnológica, Blender vendría siendo la alternativa ideal para tanto software restrictivo y de difícil adquisición. A fin de difundir el conocimiento sobre esta herramienta y sus posibilidades, la Comunidad de Software Libre se complace en invitarles a participar en el Primer Taller Universitario de Blender a realizarse el día 28 de febrero de 2008 en los docentes de nuestro recinto Universitario.

### *Programa*

Como parte de este Evento se realizarán conferencias sobre temas que involucran aspectos tales como motores de renderizado, personalización de blender, animación, programación de juegos, paseos virtuales, etc., así como el empleo de software de diseño de apoyo al proceso creativo como Gimp. Se agrega a esto además una exposición de trabajos hechos por estudiantes de nuestra Universidad, y la exhibición del tráiler del primer film elaborado completamente en Blender: Plumíferos, entre otras actividades.

Desde ya contamos contigo. ¡Haz tuyo este Evento!

Contactos:

Ing. Raul Pérezalejo Neyra, Profesor Facultad 5 ([rperez-alejo@uci.cu](mailto:rperez-alejo@uci.cu))

Yunier Vega Rodríguez, estudiante Facultad X ([yvrodriguez@estudiantes.uci.cu](mailto:yvrodriguez@estudiantes.uci.cu))





## ***NOVA A LA LUZ***



***UXI ya es internacional  
VISITENOS!***

<http://revistauxi.wordpress.com/>



***FORO LINUXERO***

<http://www.espaciolinux.com/foros.html>

