

AÑO ----- 0
NÚMERO ----- 9
FECHA: 2013-07-29

#9

“ESCORPIÓN”

HD

Hackers & DEVELOPERS

Magazine digital de distribución
mensual sobre Software Libre, Hacking y Programación
para profesionales del sector de Tecnologías de la Información

Staff

Eugenia Bahit	Arquitecta GLAMP & Agile Coach
María José Montes Díaz	Técnica en Informática de Gestión
Milagros Infante Montero	Est. Ingeniería de Sistemas
Indira Burga	Ingeniera de Sistemas
Sergio Infante Montero	Ingeniero de Software
Mario Iván Ruvalcaba	Profesional Técnico en Informática



Hackers & Developers Magazine se distribuye bajo una licencia **Creative Commons Atribución NoComercial CompartirIgual 3.0 Unported**. Eres libre de copiar, distribuir y compartir este material.
FREE AS IN FREEDOM!

Hackers & Developers Magazine, es una iniciativa sin fines de lucro destinada al fomento y difusión de las tecnologías libres presentes o futuras, bajo una clara óptica docente y altruista, que resulte de interés técnico y/o científico a profesionales del sector de Tecnologías de la Información. Hackers & Developers Magazine se sostiene económicamente con el apoyo de la comunidad, no recibiendo subvención alguna de ninguna empresa, organización u organismo de Gobierno. Necesitamos de tu apoyo para poder mantener este proyecto.

Ayúdanos a continuar con este proyecto

Puedes hacer un donativo ahora, de 10, 15, 25, 50, 100 o 150 USD para ayudar a que Hackers & Developers Magazine pueda seguir publicándose de forma gratuita, todos los meses. Puedes donar con PayPal o Tarjeta de Crédito a través del siguiente enlace:

www.hdmagazine.org/donar

“Hacker es alguien que disfruta jugando con la inteligencia”

Richard Stallman
Free Software, Free Society
(Pág. 97), GNU Press 2010-2012

En esta edición:

Recupera la contraseña de root en GNU/Linux.....	4
Conexiones SSH y SFTP desde Python con paramiko.....	7
Piwik: Una plataforma de análisis web alternativa.....	12
Página Man ¿cómo crearlas?.....	24
Monitorización del Sistema con Conky.....	33
Creando capturas de pantalla para tus manuales con Shutter.....	39
Trabajando con GIT II.....	45
KILL & KILLALL: Asesinato en Primer Grado.....	51
El poderoso virtualenvwrapper.....	55

Y LAS SECCIONES DE SIEMPRE:

ASCII Art.....	Pág. 61
Este mes: Scorpion	
Zona U!.....	Pág. 62
La comunidad de nuestros lectores y lectoras	

Créditos

Hackers & Developers Magazine es posible gracias al compromiso de:

Responsable de Proyecto
Eugenia Bahit

Responsables de Comunicación

Indira Burga (Atención al Lector) - Milagros Infante (Difusión)

Staff

Eugenia Bahit
Arquitecta GLAMP & Agile Coach
www.eugeniabahit.com

María José Montes Díaz
Técnica en Informática de Gestión
archninfo.blogspot.com.es

Indira Burga
Ingeniera de Sistemas
about.me/indirabm

Sergio Infante Montero
Ingeniero de Software
neosergio.net

Milagros Infante Montero
Estudiante de Ingeniería en Sistemas
www.milale.net

Colabora en esta edición:
Fabio Durán Verdugo

Difusión

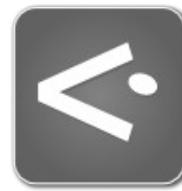
Hackers & Developers Magazine agradece a los portales que nos ayudan con la difusión del proyecto:



www.debianhackers.net



www.desarrolloweb.com



www.desdelinux.net

E-mail de Contacto:
contacto@hdmagazine.org

Web Oficial: www.hdmagazine.org
Cuenta Twitter Oficial: [@HackDevMagazine](https://twitter.com/HackDevMagazine)

Recupera la contraseña de root en GNU/Linux

Aquí os dejo un pequeño truco que nos permitirá restablecer la contraseña de root.

Escrito por: **María José Montes Díaz** (Archera & Programadora)



Estudiante de Grado Ingeniería en Tecnología de la información. Técnico en informática de gestión. Monitora FPO. Docente de programación Python y Scratch para niños de 6-12 años. Activista del software libre y cultura libre.

Webs:

Blog: <http://blog.archninfo.org/>

Redes sociales:

Twitter: [@MMontesDiaz](https://twitter.com/MMontesDiaz)

Utilizamos muchas contraseñas y, por temas de seguridad, no usamos la misma para todo, pero ¿qué ocurre si olvido la contraseña de root? Pues veamos un método para restablecerla.

Lo primero que debemos hacer es decirle a nuestro gestor de arranque que, en lugar de arrancar un modo normal, inicie sólo un terminal con Bash. Para ello, en el inicio del sistema, tenemos la posibilidad de editar las entradas del gestor de arranque. Veamos cómo se haría para Lilo y Grub:

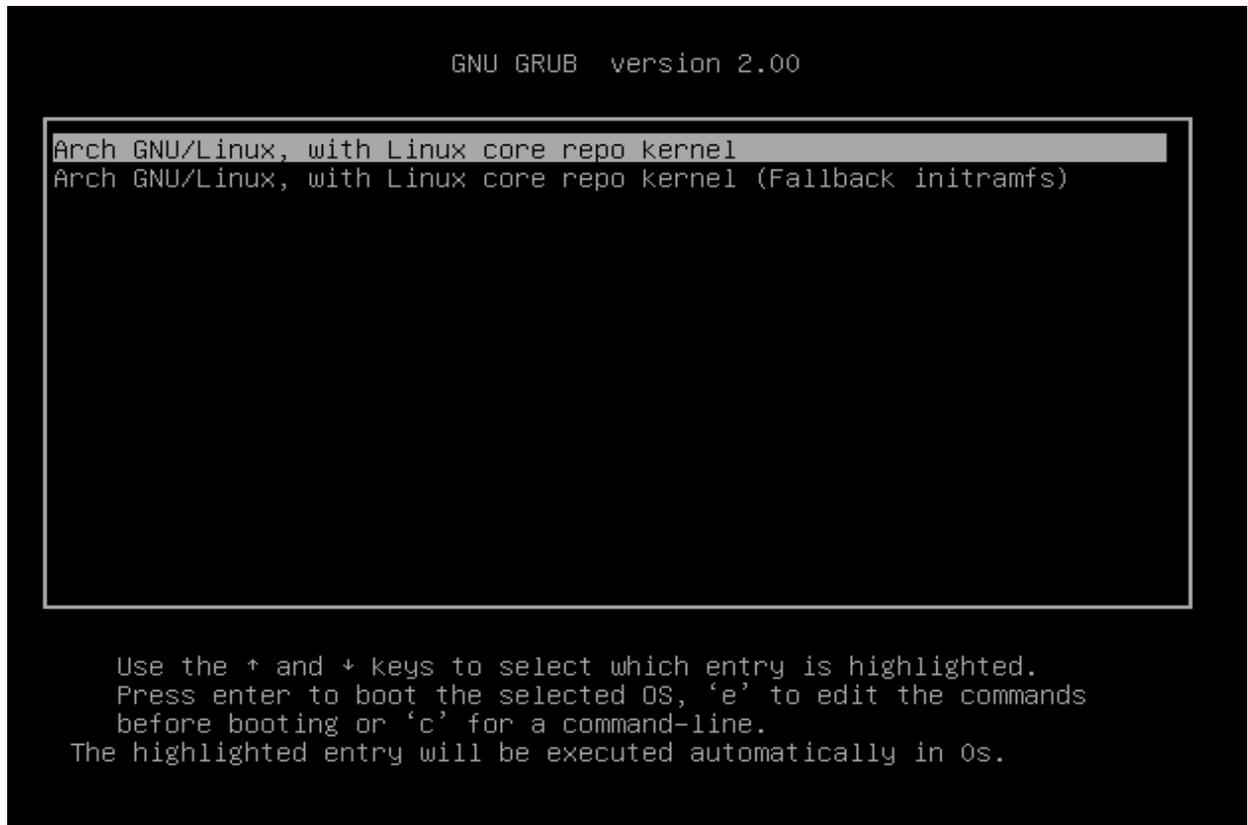
LILO



Presionamos **[Tab]**, y añadimos a la entrada **init=/bin/bash**

```
Linux
boot:Linux init=/bin/bash
```

GRUB



Podemos editar la entrada seleccionada pulsando **e**. Buscamos la entrada **linux**. Podemos ver un ejemplo en la imagen:

```
GNU GRUB version 2.00

insmod ext2
set root='hd0,msdos1'
if [ x$feature_platform_search_hint = xy ]; then
  search --no-floppy --fs-uuid --set=root --hint-bios=hd0,ms\
dos1 --hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 7cb2ba1c-a\
bcf-45b8-9f07-0943c96c9425
else
  search --no-floppy --fs-uuid --set=root 7cb2ba1c-abcfc-45b8\
-9f07-0943c96c9425
fi
echo          'Loading Linux core repo kernel ...'
linux         /boot/vmlinuz-linux root=UUID=7cb2ba1c-abcfc-45b\
8-9f07-0943c96c9425 ro quiet
echo          'Loading initial ramdisk ...'
initrd        /boot/initramfs-linux.img

Minimum Emacs-like screen editing is supported. TAB lists
completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for
a command-line or ESC to discard edits and return to the GRUB
menu.
```

Una vez localizada la entrada, al final de línea, en este caso detrás de quiet, añadimos:

```
init=/bin/bash
```

Una vez editada la entrada, pulsamos F10 para continuar.

Nuestro sistema arrancará en un terminal con Bash, como usuario root y con el sistema de archivos montado como sólo lectura. Lo siguiente, cambiar el modo a lectura/escritura:

```
mount -o remount -rw
```

Y ya podemos restablecer la contraseña ejecutando:

```
passwd root
```

Sólo nos queda reiniciar el sistema:

```
/sbin/shutdown now -arn
```

Scripting para SysAdmins:

Conexiones SSH y SFTP desde Python con paramiko

Crear conexiones SSH con Python, es tan simple como escribir unas pocas líneas de código gracias al módulo paramiko¹ y con él, comenzamos la nueva serie de artículos «*Scripting para SysAdmins*», especialmente diseñada para Administradores de Sistemas que deseen incursionar en la programación.

Escrito por: **Eugenia Bahit** (GLAMP Hacker & eXtreme Programmer)



Eugenia es **Arquitecta de Software**, docente e instructora de tecnologías **GLAMP** (GNU/Linux, Apache, MySQL, Python y PHP) y **Agile coach** (UTN) especializada en Scrum y **eXtreme Programming**. Miembro de la **Free Software Foundation**, **The Linux Foundation** y **Debian Hackers**. Creadora de **python-printr**, **Europio Engine** y colaboradora de **Vim**.

Webs:

Cursos de programación: www.cursosdeprogramacionadistancia.com

Web personal: www.eugeniabahit.com

Redes sociales:

Twitter / Identi.ca: [@eugeniabahit](https://twitter.com/eugeniabahit)

Combinar el poder de SSH con el de Python es un sueño muy fácil de cumplir: solo basta con recurrir al módulo paramiko para que con unas pocas líneas de código, podamos crear *scripts* que nos permitan crear conexiones SSH y ejecutar comandos de forma remota.

```
import paramiko
```

paramiko permite que el usuario se valide tanto por contraseña como por par de llaves,

¹ <https://pypi.python.org/pypi/paramiko>

por lo cual es ideal para autenticar usuarios más allá de las políticas del servidor.

Conexión con autenticación por contraseña

El siguiente, es un ejemplo de conexión estándar autenticada por contraseña:

```
import paramiko

# Inicia un cliente SSH
ssh_client = paramiko.SSHClient()

# Establecer política por defecto para localizar la llave del host localmente
ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())

# Conectarse
ssh_client.connect('123.456.78.90', 22, 'user', 'secretpassword')

# Ejecutar un comando de forma remota capturando entrada, salida y error estándar
entrada, salida, error = ssh_client.exec_command('ls -la')

# Mostrar la salida estándar en pantalla
print salida.read()

# Cerrar la conexión
ssh_client.close()
```

Como se puede observar, utilizar paramiko es sumamente simple y no tiene demasiada ciencia. El único “truco” a tener en cuenta, es establecer la política por defecto para la localización de la llave del host en el ordenador del cliente. De lo contrario, si no se encontrara la llave del host (*host key*, usualmente localizada en el archivo `~/.ssh/known_hosts`), Python nos arrojaría la siguiente excepción de paramiko:

```
raise SSHException('Unknown server %s' % hostname)
paramiko.SSHException: Unknown server 123.456.78.90
```

Un consejo: para evitar dejar la contraseña almacenada en texto plano (lo cual es de sumo riesgo y sin sentido), éste se le puede solicitar al usuario mediante `getpass`.

```
from getpass import getpass
import paramiko

clave = getpass('Clave: ')

HOST = '123.456.78.90'
PUERTO = 372
USUARIO = 'juanperez'
```

```
datos = dict(hostname=HOST, port=PUERTO, username=USUARIO, password=CLAVE)

ssh_client = paramiko.SSHClient()
ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
ssh_client.connect(**datos)
entrada, salida, error = ssh_client.exec_command('ls -la')
print salida.read()
ssh_client.close()
```

Conexión autenticada por par de llaves

Para conectarse evitando la utilización de una contraseña, se pudo haber creado previamente, una llave pública que el usuario debió enviar al servidor, para que el servidor la pueda contrarrestar contra la llave privada del usuario. En este caso, solo podrá bastar con no enviar la contraseña para autenticar al usuario:

```
import paramiko

HOST = '123.456.78.90'
PUERTO = 372
USUARIO = 'juanperez'
datos = dict(hostname=HOST, port=PUERTO, username=USUARIO)

ssh_client = paramiko.SSHClient()
ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
ssh_client.connect(**datos)
entrada, salida, error = ssh_client.exec_command('ls -la')
print salida.read()
ssh_client.close()
```

Ejecución remota de comandos como súper usuario

Si el usuario `root` no tiene permitidas las conexiones remotas ¿cómo ejecutar comandos con privilegios de súper usuario si nos estamos conectando con `paramiko`? Ésta, suele ser una de las dudas más frecuente y sin embargo, la solución no está en `paramiko` sino en los comandos ejecutados en el propio *shell*. Vémoslo con un ejemplo en el que listamos las reglas establecidas en `iptables` con `sudo -S`:

```
from getpass import getpass
import paramiko

HOST = '123.456.78.90'
PUERTO = 372
USUARIO = 'juanperez'
datos = dict(hostname=HOST, port=PUERTO, username=USUARIO)

password = getpass('Clave: ')
ssh_client = paramiko.SSHClient()
ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
ssh_client.connect(**datos)
```

```
comando = 'echo %s | sudo -S iptables -nL' % password
entrada, salida, error = ssh_client.exec_command(comando)

print salida.read()
ssh_client.close()
```

Guardando los *logs* de paramiko

Con paramiko también es posible registrar toda la actividad que paramiko realice desde el *script*:

```
from getpass import getpass
import paramiko

HOST = '123.456.78.90'
PUERTO = 372
USUARIO = 'juanperez'
datos = dict(hostname=HOST, port=PUERTO, username=USUARIO)
password = getpass('Clave: ')

paramiko.util.log_to_file('paramiko.log')

ssh_client = paramiko.SSHClient()
ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
ssh_client.connect(**datos)
comando = 'echo %s | sudo -S iptables -nL' % password
entrada, salida, error = ssh_client.exec_command(comando)

print salida.read()
ssh_client.close()
```

Manipulando archivos remotos mediante SFTP

Con paramiko es posible crear conexiones SFTP directas (sin recurrir al cliente SSH) por medio del objeto `SFTPCClient()`. Aunque también, podremos obtener una instancia de `SFTPCClient` desde el propio cliente SSH (`SSHClient`) recurriendo al método `open_sftp()`.

```
import paramiko

paramiko.util.log_to_file('paramiko.log')

HOST = '123.456.78.90'
PUERTO = 372
USUARIO = 'juanperez'
datos = dict(hostname=HOST, port=PUERTO, username=USUARIO)

ssh_client = paramiko.SSHClient()
ssh_client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
ssh_client.connect(**datos)
entrada, salida, error = ssh_client.exec_command('pwd')
ruta = salida.read().replace('\n', '')

sftp = ssh_client.open_sftp() # Crea un objeto SFTPCClient()
```

```

# Descargando archivos
archivos = sftp.listdir()

for archivo in archivos:
    archivo_remoto = "%(ruta)s/%(nombre)s" % dict(ruta=ruta, nombre=archivo)

    print "Descargando: %s" % archivo_remoto

    try:
        sftp.get(archivo_remoto, "paramii/%s" % archivo)
        print "copiado archivo %s." % archivo
    except:
        print "Fallo al intentar copiar %s. Tal vez es un directorio." % archivo

sftp.close()
ssh_client.close()

```

El cliente SFTP de paramiko, provee de los mismos métodos que un cliente FTP normal. Todos los métodos pueden ser consultados desde el manual oficial ingresando en <http://www.lag.net/paramiko/docs/paramiko.SFTPClient-class.html>

Entre los métodos provistos por el cliente SFTP, podremos encontrar:

Método	Descripción
get(remoto, local)	Trae un archivo remoto a un directorio local
put(local, remoto)	Envía un archivo local al servidor
chdir(ruta)	Cambia el directorio de trabajo actual
chmod(ruta, modo)	Cambia permisos en un archivo
mkdir(ruta, modo=511)	Crea un directorio
rename(anterior, nuevo)	Renombra un archivo o directorio
remove(archivo)	Elimina un archivo
rmdir(ruta)	Elimina un directorio
symlink(fuente, destino)	Crea un enlace simbólico

Piwik: Una plataforma de análisis web alternativa

El análisis web es un conjunto de técnicas relacionadas con el estudio de datos relativos al tráfico en un sitio web. La principal finalidad de este tipo de análisis es la de comprender de mejor forma el impacto y la tendencia que nuestros servicios producen sobre los visitantes del sitio, ayudándonos a emprender una estrategia más eficaz de cara a nuestro público o nicho de mercado.

Escrito por: **Mario Iván Ruvalcaba** (Profesional Técnico en Informática)



Mario Iván Ruvalcaba (**a.k.a. Perseo**) es estudiante de Ingeniería en Computación, Profesional Técnico en Informática y Cofundador de **DesdeLinux**. Usuario y amante de GNU/Linux, idealista, entusiasta y defensor del Software y Cultura Libre; activista y blogger. Linux user: 547468.

Webs:
www.perseosblog.com

Redes sociales:
Twitter / Identi.ca: [@icaroperseo](https://twitter.com/icaroperseo)

Cuando publicamos un sitio web, ya sea un blog personal o corporativo, o tal vez una página donde ofrecer nuestros productos y servicios, lo hacemos principalmente con dos objetivos en mente: Tratar de llegar a la mayor cantidad posible de personas y procurar crear o publicar contenido de interés para seguir contando con sus visitas de forma recurrente, pero ¿Cómo saber que estamos cumpliendo de forma efectiva nuestros objetivos? ¿Cómo conocer cuáles son los temas de verdadero interés para nuestros lectores y cuales son intrascendentes? Una de las mejores maneras de medir este impacto es por medio del análisis web.

Las herramientas de análisis web nos proporcionan información tan diversa como: el comportamiento de los usuarios en el sitio, el país de donde proceden, cuáles son las páginas que han visitado o la página que están visualizando actualmente, si han accedido al sitio directamente desde un motor de búsqueda como Google, Bing, ente otros o si han sido redirigidos desde una página externa, cuanto tiempo permanecen en el sitio, en qué punto abandonan el sitio, qué páginas cuentan con una mayor cantidad

de tráfico y cuáles no, obtención de reportes de visitas diarias, semanales y mensuales, cuales son los criterios de búsqueda o palabras clave más utilizados por los usuarios, siendo estos solo algunos de los tantos beneficios que nos ofrecen este tipo de análisis. Existen servicios muy conocidos que nos ayudan en este tipo de tareas como *Google Analytics*, pero en esta ocasión les presentaré una plataforma mucho más completa y que además tiene como plus el ser software libre.

Piwik² es una plataforma de análisis web que cuenta con un sinnúmero de funcionalidades y características que lo convierten en una herramienta bastante flexible y mucho mejor adaptada a las necesidades del usuario. Posee una amplia comunidad que le brinda un desarrollo activo y constante (actualmente se encuentra en la versión 1.12 y próxima al lanzamiento de la versión 2.0, misma que promete un sinfín de mejoras³), dispone de una gran cantidad de complementos propios y desarrollados por terceros, además de estar disponible en 48 idiomas incluido el español. Por otra parte también es posible integrar Piwik con *frameworks* como **Ruby On Rails** o **Django** y una gran cantidad de gestores de contenido (CMS) tales como **Wordpress**, **Drupal**, **Joomla!**, entre otros.



Ilustración 1: CMS y Frameworks soportados por Piwik entre otros.

En esta ocasión únicamente veremos una breve reseña de las funcionalidades y particularidades de Piwik, ya que mostrar plenamente todas sus capacidades daría para más de un artículo; sin más, comencemos.

Panel de Control

La mayoría de funcionalidades de Piwik se pueden apreciar directamente desde su panel de control, desde ahí podremos obtener rápidamente la siguiente información en tiempo real:

Gráfica de las últimas visitas

² <http://piwik.org/>

³ <http://piwik.org/roadmap/>



Ilustración 2: Gráfica de visitas.

En este caso el criterio empleado en la gráfica ha sido todas las visitas, pero también se pueden mostrar los siguientes datos: visitantes únicos, promedio de tiempo en el sitio, porcentaje de rebote, etc. Otras funcionalidades interesantes son: el poder realizar anotaciones en la gráfica, exportar el conjunto de datos en otros formatos o como imagen.

Visitantes en tiempo real

Fecha	Visitas	Páginas vistas
Últimas 24 horas	250	447
Últimos 30 minutos	10	16
Jue 27 Jun - 10:46:01 - IP: ██████████		
desde Google - "Palabras clave no está definido"		
Páginas: █		
Jue 27 Jun - 10:46:01 - IP: ██████████		
desde Google - "Palabras clave no está definido"		
Páginas: █		
Jue 27 Jun - 10:45:04 - IP: ██████████		
desde Google - "linux-backports-modules linux mint" # 8		
Páginas: █		
Jue 27 Jun - 10:43:20 (44 min 40s) - IP: ██████████		
desde Google - "what is the ? for running in boot-up manager of mint" # 1		
Páginas: █ █ █ █ █ █		

Ilustración 3: Visitantes en tiempo real.

Desde aquí se puede obtener suficiente información del usuario como: hora y fecha, país de origen, tipo y versión del navegador de internet, sistema operativo y/o dispositivo móvil, dirección IP, procedencia (página externa o motor de búsqueda –en caso de que el usuario haya ingresado al sitio por medio de un buscador, también se muestran las palabras clave que empleó, así como la posición en la que se encuentran dentro del gráfico “Listado de palabras clave”, mismo que se mostrará más adelante–), si

es o no un visitante recurrente, páginas del sitio que se encuentra visitando o ha visitado, duración de su visita, recursos (archivos, imágenes, etc.) que el usuario a descargado directamente del sitio.



Ilustración 4: Información detallada de las visitas al sitio.

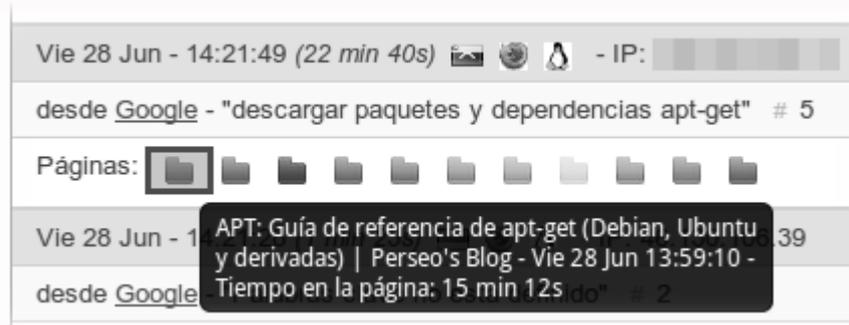


Ilustración 5: Descripción de las páginas visitadas.

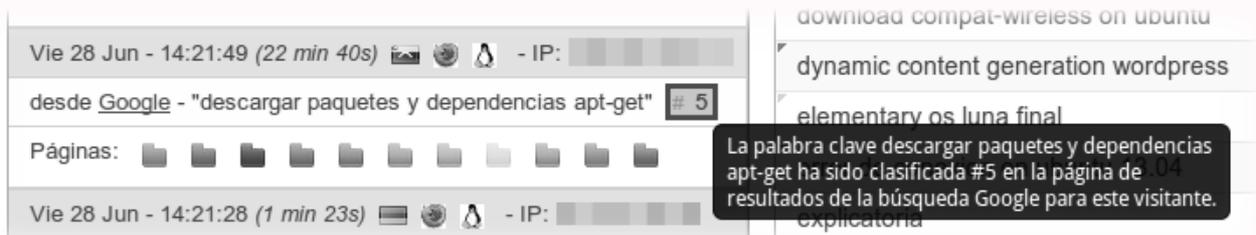


Ilustración 6: Palabras clave utilizadas y posición ocupada (ranking) en el gráfico "Listado de palabras clave".

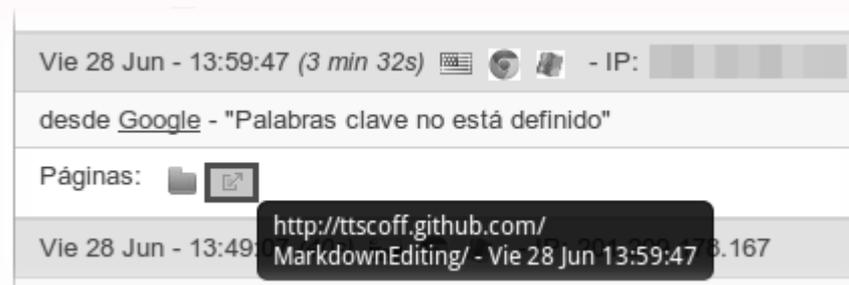


Ilustración 7: Descripción de los recursos descargados o accedidos directamente del sitio.

Duración de las visitas

Por medio de este gráfico podremos conocer cuál es el tiempo aproximado que las visitas permanecen en el sitio, dicho gráfico se puede representar ya sea como una nube de etiquetas, gráfica de barras, circular o en forma de tabla. Como casi la mayoría de los reportes, también se puede exportar a otros formatos.

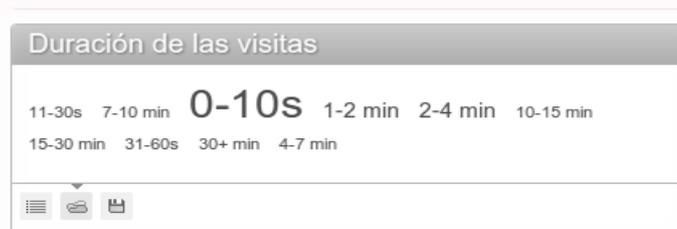


Ilustración 8: Duración de las visitas - Nube de etiquetas.

Lista de palabras clave

A través de este listado podremos analizar y obtener las palabras clave que los usuarios emplean con mayor regularidad en los motores de búsqueda al momento de efectuar sus consultas. Este listado ordena las *keywords* según su trascendencia, indicando también la cantidad de veces que han sido utilizadas por los usuarios.

Palabras clave	Visitas
Palabras clave no está definido	58
sublime text markdown	2
"pro git" "español" "pdf"	1
apt get de firefox	1
como entrar a la carpeta themes suse	1
compat-wireless ubuntu13	1
elementary os extension	1
el uso del apt-get	1
en otra ocasion tuve la oportunidad de entrevistar	1
error de conexion en ubuntu 13.04	1
firefox for debian apt pro centrum softwaru	1
instalar rvm ubuntu	1
kernel 3.5 ubuntu 12.04	1
netstat servicios puertos abiertos	1
que es gambas	1
testear redess wifi ubuntu 13.04	1
ubuntu atajo de teclado	1

Ilustración 9: Palabras clave.

Lista de páginas web externas

Este gráfico muestra las páginas web externas que hacen referencia o enlazan a nuestro sitio, así como la cantidad de veces que los visitantes han visitado nuestra web por medio de estas.

Sitio web	Visitas
planetubuntu.es	11
t.co	5
www.root.cz	5
plus.url.google.com	2
translate.googleusercontent.com	2
askubuntu.com	1
deblinux.wordpress.com	1
ubuntuforums.org	1
www.lasombradelhelicoptero.com	1
www.ranklinux.tk	1

1-10 de 10

25

Ilustración 10: Páginas externas

Visor Map

Por medio de este widget podemos ver gráficamente el porcentaje de visitas que nuestro sitio recibe desde las diferentes partes del mundo.



Ilustración 11: Porcentaje de visitas por países.

Navegadores de los visitantes y Motores de búsqueda

Ambos indicadores muestran las estadísticas correspondientes a su finalidad por lo que no hay mucho que agregar al respecto.

Navegadores de los visitantes	
Navegador	Visitantes únicos
Chrome	62
Firefox	49
Safari	10
Internet Explorer	2
Konqueror	1

1-5 de 5

Reporte relacionado: [Versión del navegador](#)

Ilustración 12: Estadísticas de los navegadores web.

Mejores motores de búsqueda	
Motores de búsqueda	Visitas
Google	73
DuckDuckGo	1
Google Images	1

1-3 de 3

Ilustración 13: Estadísticas de los motores de búsqueda.

Visitas por hora

Este apartado nos mostrará los horarios de mayor afluencia de visitantes en nuestro sitio. Gracias a este tipo de información podremos determinar cuáles son los horarios idóneos para publicar nuestro contenido, realizar promociones, etc.



Ilustración 14: Visitas por hora.

Estas son solo algunas de las funcionalidades que Piwik muestra por defecto en el panel de control, y como se puede sospechar, también podemos añadir o quitar widgets según nuestras necesidades o preferencias.

Otras funcionalidades

Algunas de las funcionalidades que no aparecen a simple vista pero que se encuentran presentes son:

Información al detalle

Piwik ofrece diversas opciones que nos permiten analizar con mayor detalle la información obtenida. Algunos ejemplos de esto son los siguientes:

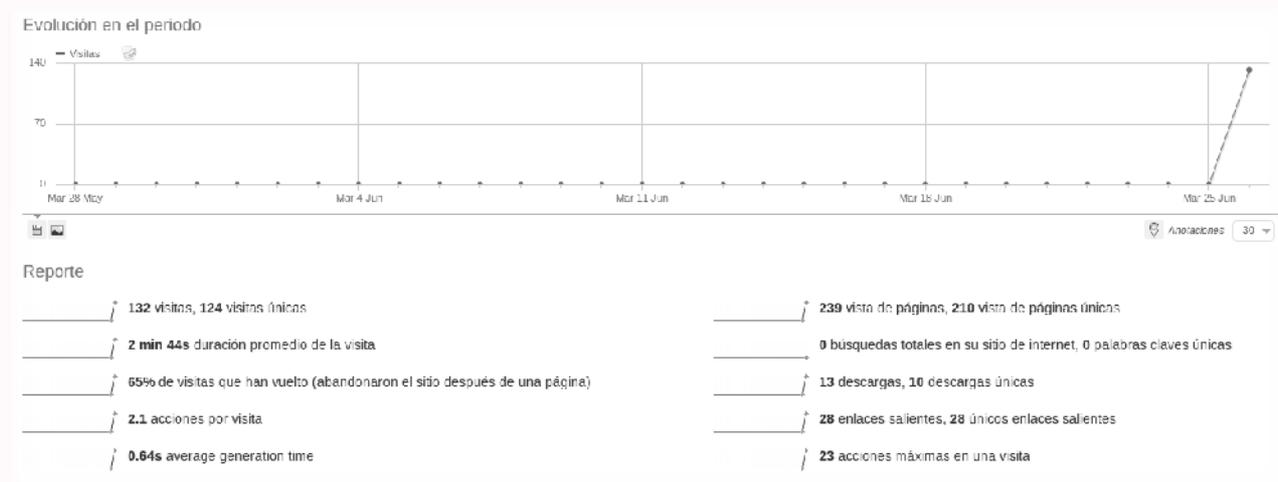


Ilustración 15: Evaluación del periodo.

Títulos de página

Nombre de Página	Páginas vistas	Páginas vistas únicas	Porcentaje de rebote	Tiempo promedio en la página	Porcentaje de salida	Avg. generation time
Perseo's Blog (posts antiguos página 0) Perseo's...	23	18	60%	50s	56%	0.92s
How To: Monta tu propio Feedly de forma privada y ...	11	11	67%	3 min 48s	73%	0.71s
Markdown: The perfect config for Sublime Text 2 ...	15	11	55%	5 min 42s	100%	0.18s
elementary OS: A quick look at Scratch Text Editor...	11	8	63%	12s	100%	0.13s
¿Qué es Gambas y Cómo instalarlo en Ubuntu	8	8	67%	43s	63%	0.47s
Administración: Gestiona todos los servicios de fo...	6	6	100%	17s	17%	2.17s
How To: Instalar el driver de tu wireless con comp...	8	6	40%	7 min 35s	83%	0.53s
How To: Install Ruby via RVM in Ubuntu	6	6	100%	0s	100%	0.21s
How To: Install the driver for your wireless with ...	6	6	75%	53s	83%	1.19s
Administración: Listar aplicaciones o servicios qu...	7	5	67%	48s	60%	0.96s

Ilustración 16: Títulos de página,

Registro de visitantes

Fecha	Visitantes	URL de referencia	Acciones
Mié 26 Jun - 23:48:04 IP: 181.31.179.98 Proveedor: Libertel	Plugins:	Sitio web: translate.google.com.ar/	1 Acción 1. Solucionar problemas de conexión con WiFi Realtek Card RTL8188CE (RTL8192CE, ... http://www.perseosblog.com/en/posts/solving-connection-problem-with-realtek-w...
Mié 26 Jun - 23:01:30 IP: 190.96.97.11	Plugins:	Sitio web: planetubuntu.es	4 Acciones - 7 min 18s 1. Gigggle. Un excelente cliente para Git con interfaz gráfica Perseo's Blog http://www.perseosblog.com/posts/gigggle-un-excelente-cliente-para-git-con-int... 2. ¿Qué es Gambia y cómo instalarlo en Ubuntu/Mint? Perseo's Blog http://www.perseosblog.com/posts/que-es-gambias-y-como-instalarlo-en-ubuntu-mint/ 3. Seguridad: Deshabilitar el soporte para IPv6 en Ubuntu, Mint y elementary OS ... http://www.perseosblog.com/posts/seguridad-deshabilitar-el-soporte-para-ipv6-... 4. Perseo's Blog (posts antiguos página 0) Perseo's Blog http://www.perseosblog.com/
Mié 26 Jun - 23:05:13 IP: 201.1.215.27 Proveedor: Telese	Plugins:	Google Palabras clave: "Palabras clave no está definido" [6]	1 Acción 1. Solving connection problem with Realtek WiFi Card RTL8188CE (RTL8192CE, RTL81... http://www.perseosblog.com/en/posts/solving-connection-problem-with-realtek-w...
Mié 26 Jun - 23:00:39 IP: 181.163.198.218 Proveedor: Movistar	Plugins:	Entrada directa	1 Acción 1. How To: Instalar el driver de tu wireless con compat drivers (compat wireless... http://www.perseosblog.com/posts/how-to-instalar-el-driver-de-tu-wireless-con...
Mié 26 Jun - 22:48:21 IP: 189.29.214.186 Proveedor: Virtua	Plugins:	Entrada directa	1 Acción 1. PPA: How to install Metadata anonymisation toolkit (MAT) in Ubuntu 13.04/12.1... http://www.perseosblog.com/en/posts/ppa-install-metadata-anonymisation-toolk...

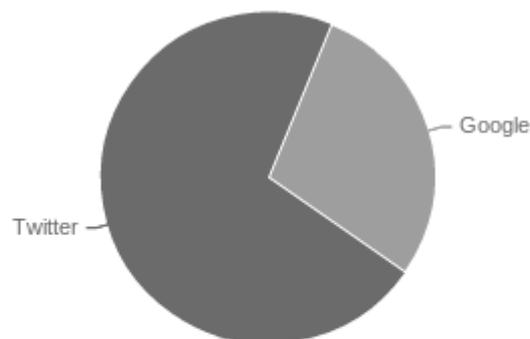
Ilustración 17: Registro de visitantes.

Redes Sociales

Por otra parte, también podemos observar el impacto que tiene nuestro sitio en las redes sociales.

Redes sociales

— Visitas



A la izquierda, este es un subconjunto de los informes de los sitios de internet. Filtra respecto de otros sitios de internet, así puede comparar sus referentes de las redes sociales directamente.

Ilustración 18: Redes sociales.

Administración de Plugins

Por defecto Piwik incorpora varios plugins mismos que le añaden una gran flexibilidad.

Administración de Plugins

Los plugins extienden y amplían las funcionalidades de Piwik. Una vez que el plugin esté instalado, usted puede activarlo o desactivarlo desde aquí.

Plugin	Versión	Descripción	Estado	Acción
Actions	1.12	Reportes de páginas vistas, enlaces externos y descargas. Los Enlaces Externos y las Descargas se rastrean automáticamente! <i>By Piwik</i> .	Activo	Desactivar
Annotations	1.12	Te permite adjuntar notas en los días para marcar cambios que hayas realizado en tu sitio, guardar los análisis que hayas hecho sobre los datos y compartir tus ideas con tus colegas. Al agregar anotaciones a tus datos te asegurarás de recordar el por qué de los datos arrojados. <i>By Piwik</i> .	Activo	Desactivar
AnonymizeIP	1.12	Ocultar la dirección IP del visitante para cumplir las leyes/directrices de privacidad de su localidad. <i>By Piwik</i> .	Inactivo	Activar
CustomVariables	1.12	Las Variables Personalizadas son pares del estilo nombre,valor que puede configurar en una visita usando la función <code>setVisitCustomVariables()</code> de la API de Javascript. Piwik reportará cuantas visitas, páginas y conversiones han usado cada una de estos nombres y valores personalizados. Required to use Ecommerce Analytics feature! <i>By Piwik</i> .	Activo	Desactivar
Dashboard	1.12	Tu Panel de Análisis Web. Puedes personalizar su Panel; agregar	Activo	Desactivar

Ilustración 19: Plugins.

Pero esto no es todo, lo mejor aún están por venir.

Características especiales

El verdadero potencial de Piwik se aprecia de mejor forma a través de las siguientes características:

Administración de múltiples sitios

Piwik es capaz de administrar múltiples sitios desde una sola interfaz/servidor, para hacerlo solo es necesario agregar los sitios en cuestión e incluir el *Tracking Code* o código de seguimiento a las diferentes páginas web, este código se genera de forma independiente para cada sitio administrado.

Administrador de Acceso (multiusuarios)

Piwik cuenta con la posibilidad de gestionar permisos y usuarios.

Privacidad

Este apartado es de lo más interesante, ya que desde aquí podremos administrar el nivel de privacidad al obtener las distintas métricas de nuestros visitantes, bien sea por cuestiones éticas o legales.

Las opciones que Piwik nos ofrece dentro de este apartado son:

- Hacer anónimas las direcciones IP de los visitantes.

- Borrar los registros y reportes de los visitantes después de un cierto periodo de tiempo.
- Soporte para *Do Not Track*⁴.
- Mostrar un mensaje en el sitio con la opción de no seguimiento (rastreo) de Piwik a sus visitantes (eliminación de cookies).

Mensajero Móvil

Esta característica nos permite crear informes SMS y recibir mensajes de texto cortos con las estadísticas de tus sitios.

Crear y Recibir Informes por Correo Electrónico

Al igual que el punto anterior, Piwik nos permite recibir informes vía email de todos nuestros sitios.

APIs y Widgets

Piwik ofrece la posibilidad de incluir widgets con las estadísticas de tus sitios en blogs, sitios y aplicaciones web, intranet, entre otros. También puedes utilizar sus APIs⁵ para obtener esta información en tus aplicaciones.

Requisitos mínimos para la implementación de Piwik

Para poder ejecutar Piwik en tu servidor necesitaras contar con lo siguiente:

- PHP 5.13 o superior.
- MySQL 4.1 o superior.
- Extensiones **pdo** y **pdo_mysql** o la extensión **mysqli**, habilitadas de forma predeterminada. En servidores tipo Debian estas extensiones se incluyen de forma predeterminada instalando el paquete **php5-mysql**.
- Extensión **GD de PHP** para la generación de gráficos (*sparklines*). En servidores tipo Debian esta extensión se incluye de forma predeterminada instalando el paquete **php5-gd**.

Instalación de Piwik

Realmente la instalación de Piwik en el servidor no representa un problema significativo gracias a su intuitivo asistente, el cual nos permite conseguir nuestro objetivo en unos cuantos simples pasos.

4 https://en.wikipedia.org/wiki/Do_Not_Track

5 <http://piwik.org/docs/analytics-api/>



Ilustración 20: Asistente de instalación de Piwik.

Una vez finalizada la instalación y preconfiguración, es necesario incluir el código de rastreo en nuestro sitio web para comenzar a utilizar las distintas métricas que esta plataforma nos ofrece.

Documentación

Piwik cuenta con una documentación⁶ bastante completa misma que se encuentra traducida parcialmente al español. En ella encontraremos información relacionada a los aspectos básicos de instalación, configuración, optimización, uso de las API's, desarrollo de complementos y por supuesto, el manual de usuario. Por otra parte, también es posible adquirir desde su tienda virtual el ebook: *Piwik Web Analytics Essential*⁷ donde se profundizan a detalles todas sus funcionalidades.

6 <http://es.piwik.org/documentacion/>

7 <http://piwik.org/shop/>

Página Man ¿cómo crearlas?

Si alguna vez creaste al menos una herramienta para GNU/Linux, muy probablemente se te haya cruzado la pregunta «¿Cómo se hará para crear una página Man?» y la idea de este artículo, es darte una respuesta paso a paso.

Escrito por: **Eugenia Bahit** (GLAMP Hacker & eXtreme Programmer)



Eugenia es **Arquitecta de Software**, docente e instructora de tecnologías **GLAMP** (GNU/Linux, Apache, MySQL, Python y PHP) y **Agile coach** (UTN) especializada en Scrum y **eXtreme Programming**. Miembro de la **Free Software Foundation**, **The Linux Foundation** y **Debian Hackers**. Creadora de **python-printr**, **Europio Engine** y colaboradora de **Vim**.

Webs:

Cursos de programación: www.cursosdeprogramacionadistancia.com
Web personal: www.eugeniabahit.com

Redes sociales:

Twitter / Identi.ca: [@eugeniabahit](https://twitter.com/eugeniabahit)

Man es la herramienta utilizada por el sistema operativo GNU con Linux -así como por otros basados en *nix-, para documentar desde comandos hasta aplicaciones. Generalmente, como usuarios del sistema operativo GNU, cuando necesitamos aprender sobre un determinado comando ejecutamos `man comando` para acceder a la **página del manual** específica para el comando correspondiente.

Crear nuevos comandos es tan simple como crear un script ejecutable y almacenarlo en el directorio `/sbin` (o `/usr/sbin`), sin embargo, crear una página del manual (*man page*) para dicho comando, muchas veces suele ser “una receta misteriosa”, tal vez, por la escasa -o difícilmente accesible- información al respecto.

Siguiendo la filosofía de “erradicación de misterios” que nos acompaña edición tras edición, en esta oportunidad, aprenderemos a crear nuestras propias *man pages*.

Aclarando el panorama

Las páginas del manual son archivos de texto plano que sorprendentemente, pueden ser escritos con cualquier editor de texto, sin requerir de herramientas específicas para su creación (muy a pesar de los tutoriales que afirman lo contrario).

Una forma de “descubrir” por tus propios medios de qué se trata una página del manual, es localizar cualquier *man page* de un comando al azar. Por ejemplo, para localizar la ubicación del archivo de la página del manual para el comando `mkdir`, podemos ayudarnos del comando `whereis`:

```
whereis mkdir
```

Este comando, no solo nos arrojará la ruta del binario, sino además, la de la página del manual:

```
/usr/share/man/man2/mkdir.2.gz
```

Para visualizar el archivo, solo bastará con copiarlo y descomprimirlo con `gzip`:

```
# cp /usr/share/man/man2/mkdir.2.gz /home/user/  
# gzip -d mkdir.2.gz
```

Si miramos el contenido del archivo descomprimido, podremos ver que se trata de texto plano puro:

```
(...)  
The newly created directory will be owned by the effective user ID of the  
process.  
If the directory containing the file has the set-group-ID  
bit set, or if the file system is mounted with BSD group semantics  
.RI ( "mount -o bsdgroups"  
or, synonymously  
.IR "mount -o grpuid" ),  
the new directory will inherit the group ownership from its parent;  
otherwise it will be owned by the effective group ID of the process.  
  
If the parent directory has the set-group-ID bit set then so will the  
newly created directory.  
.SH "RETURN VALUE"  
.BR mkdir ()  
returns zero on success, or -1 if an error occurred (in which case,  
.I errno  
is set appropriately).  
.SH ERRORS  
.TP  
.B EACCES  
The parent directory does not allow write permission to the process,  
or one of the directories in  
.I pathname  
did not allow search permission.  
(See also  
.BR path_resolution (7).)  
.TP  
.B EEXIST  
.I pathname
```

(...)

Identificar el número de sección

Si observaste detenidamente la salida del `whereis` que hicimos anteriormente, habrás notado que tanto la carpeta donde se almacena la página del manual, como la página del manual propiamente dicha, tienen un número:

```
/usr/share/man/man2/mkdir.2.gz
```

El número en cuestión, varía dependiendo de la herramienta o comando. Por ejemplo, en el caso de `iptables`, el número correspondiente es el 8:

```
$ whereis iptables
iptables: /sbin/iptables /usr/share/iptables /usr/share/man/man8/iptables.8.gz
```

Estos números, representan la **sección del manual** a la cuál pertenece el comando o herramienta. El manual se encuentra dividido en 8 secciones a las cuáles a cada una le corresponde un número:

```
1 Programas ejecutables y guiones del intérprete de órdenes
2 Funciones provistas por el núcleo (kernel) del Sistema Operativo
3 Funciones de la biblioteca del propio sistema
4 Archivos especiales (se encuentran generalmente en /dev)
5 Formato de ficheros y convenios p.ej. I/etc/passwd
6 Juegos
7 Paquetes de macros y convenios p.ej. man(7), groff(7).
8 Órdenes de administración del sistema (generalmente solo son para root)
```

Por lo tanto, para crear una página del manual, lo primero que debemos hacer es identificar la sección a la cuál pertenece el comando o herramienta que estamos a punto de documentar.

El número de sección nos servirá para:

1. Saber **en qué directorio** dentro de `/usr/share/man/` tendremos que guardar nuestra propia página;
2. Establecer el **nombre de archivo**, ya que el número de sección, formará parte del nombre del archivo.

Estableciendo un nombre

Si haces un listado de cualquiera de las carpetas dentro de `/usr/share/man/` o ejecutas

el comando `whereis` para varios comandos, podrás notar que el nombre de los archivos de las páginas del manual, siempre guardan el mismo formato:

```
comando.numero-seccion
```

Por ejemplo, para el comando `adduser` el archivo de la página del manual se llama `adduser.8` mientras que para el comando `cp`, se llama `cp.1`.

Contenido del archivo

Cuando hicimos un `cat` del archivo `mkdir.2`, como bien pudo observarse, algunas líneas comenzaban por un punto seguido de una o dos letras mayúsculas:

```
.I errno
is set appropriately).
.SH ERRORS
.TP
.B EACCES
The parent directory does not allow write permission to the process,
or one of the directories in
.I pathname
```

Las mismas, se denominan **macros de formato** y forman parte de los sistemas de formato de texto para plataformas tipo `*nix`, como es el caso del sistema operativo GNU, el cuál utiliza a Linux como Kernel.

roff fue el programa original que permitía dar formato al texto. Luego, fueron apareciendo programas similares como nroff y troff (entre otros), hasta que en 1990, James Clark creó groff, la alternativa libre del proyecto GNU

Las **macros de formato** pueden consultarse accediendo a la página del manual `man` de la sección 7, mediante la ejecución de: `man 7 man`

Título

La línea de título debe ser la primera luego de los comentarios sobre el archivo que eventualmente pudiesen colocarse. Un título es indicado con la macro `.TH` y debe indicarse con la siguiente sintaxis:

```
.TH nombre-comando numero-sección fecha fuente-origen título-del-manual
```

Por ejemplo, la siguiente línea:

```
.TH blockip 8 2013-06-29 "JackTheStripper 1.0 beta 3" "Manual de blockip"
```

Producirá:

blockip(8)	Manual de blockip	blockip(8)
JackTheStripper 1.0 beta 3	2013-06-29	blockip(8)

Secciones

Las secciones se indican con la macro `.SH` seguida del nombre de sección:

```
.SH nombre de la sección
```

Entre las secciones de uso más frecuente que podremos indicar, encontramos las siguientes:

NAME /NOMBRE

Nombre del comando, archivo o herramienta

SYNOPSIS /SINOPSIS

Sintaxis de uso

DESCRIPTION /DESCRIPCION

Descripción del comando, archivo o herramienta

OPTIONS/OPCIONES

Descripción de las opciones aceptadas en la línea de comandos por el comando o herramienta.

Normalmente `splo` se implementa en las secciones 1 y 8.

ENVIRONMENT /ENTORNO

Muestra una lista de las variables de entorno utilizadas por el comando o herramienta, describiendo cómo éstas son utilizadas

FILES /ARCHIVOS

Una lista de todos los archivos utilizados por el comando o herramienta.

BUGS /ERRORES

Descripción de errores o problemas conocidos.

EXAMPLE /EJEMPLO

Ejemplos de uso del comando o herramienta

AUTHORS/AUTORES

Lista de los nombres de los autores.
Generalmente se utiliza el formato: Nombre del Autor <email@example.org>

SEE ALSO/VÉASE TAMBIÉN

Una lista de páginas man sugeridas, separadas por coma y ordenadas primero, por número de sección y luego, por orden alfabético

Veamos un pequeño ejemplo y su salida:

```
.TH blockip 8 2013-06-29 "JackTheStripper 1.0 beta 3" "Manual de blockip"  
.SH NOMBRE  
blockip – bloquea una IP de forma permanente
```

El archivo anterior, produciría una salida como la siguiente:

```
blockip(8)                Manual de blockip                blockip(8)  
NOMBRE  
    blockip – bloquea una IP de forma permanente  
JackTheStripper 1.0 beta 3    2013-06-29                blockip(8)
```

Una completa descripción de las posibles **secciones** que pueden implementarse en una página del manual, podemos obtenerla consultando la página man-pages de la sección 7, ejecutando: **man 7 man-pages**

Formato de fuente

Entre los formatos más usados podemos encontrar:

```
.B Negrita  
.B hola mundo  
produce: hola mundo  
  
.I Itálica  
.I hola mundo  
produce: hola mundo  
  
.R Romana  
.R hola mundo  
produce: hola mundo
```

Los formatos anteriores pueden además, utilizarse en cualquier combinación, produciendo así un alternado de formato. Veamos algunos ejemplos:

```
.BI negrita alternado itálica
    negritaalternadoitálica

.IB itálica alternado negrita
    itálicaalternadonegrita

.BR negrita alternando Romana
    negritaalternandoRomana

.RI romana alternando itálica
    romanaalternandoitálica
```

Párrafos y saltos de línea

Un nuevo **párrafo** se indica con la macro **.PP** mientras que un **salto de línea**, con **.br**

```
.TH micomando 8 2013-06-29 "MiHerramienta 1.0" "Manual de micomando"
.SH NOMBRE
micomando – ejemplo para página del manual
.SH DESCRIPCIÓN
.B micomando
lo escribí en negritas y fui a la siguiente línea para finalizar las negritas
.PP
Ahora escribo un nuevo párrafo
.br
que tiene un salto de línea
```

La salida del ejemplo anterior, será:

```
micomando(8)                Manual de micomando                micomando(8)

NOMBRE
    micomando – ejemplo para página del manual

DESCRIPCIÓN
    micomando lo escribí en negritas y fui a la siguiente línea para
    finalizar las negritas

    Ahora escribo un nuevo párrafo
    que tiene un salto de línea

MiHerramienta 1.0          2013-06-29          micomando(8)
```

Con **.TP** es posible indicar un párrafo con **columnas**, ideal para listar argumentos:

```
.TH micomando 8 2013-06-29 "MiHerramienta 1.0" "Manual de micomando"
.SH NOMBRE
micomando – ejemplo para página del manual
.SH DESCRIPCIÓN
.B micomando
lo escribí en negritas y fui a la siguiente línea para finalizar las negritas
.PP
```

```
Ahora escribo un nuevo párrafo
.br
que tiene un salto de línea
.SH OPCIONES
.TP
.B -a
describe lo que hace la opción -a
.TP
.B -b
describe lo que hace la opción -b
```

Lo anterior, se verá así:

```
micomando(8)                Manual de micomando                micomando(8)
NOMBRE
    micomando – ejemplo para página del manual
DESCRIPCIÓN
    micomando lo escribí en negritas y fui a la siguiente línea para
    finalizar las negritas
    Ahora escribo un nuevo párrafo
    que tiene un salto de línea
OPCIONES
    -a    describe lo que hace la opción -a
    -b    describe lo que hace la opción -b
MiHerramienta 1.0          2013-06-29          micomando(8)
```

Cuando se observa todo lo que hasta ahora ha sido probado, se puede notar que los saltos de línea no son respetados a no ser que se implemente la macro `.br`. Sin embargo, esto puede ser muy tedioso si lo que se desea es, por ejemplo, indicar un fragmento de código fuente.

Para respetar los saltos de línea (e incluso los espacios y tabulaciones), disponemos de las macros `.nf` y `.fi`, para dar inicio y fin a un **bloque de texto con pre-formato**:

```
.nf
for elemento in tupla:
    print elemento * 2
.fi
```

Con las macros `.nf` y `.fi`, nuestro texto con que pre-formato, ha sido respetado:

```
for elemento in tupla:
    print elemento * 2
```

Comentarios

Los comentarios del archivo en general, deberán indicarse al comienzo del archivo. Sin embargo, comentarios sobre el propio texto, pueden hacerse en cualquier lugar que sea necesario.

Para escribir comentarios se utiliza la macro `.\`"

Un ejemplo de comentarios, es el que vimos a la hora de hacer un cat a la página del manual para `mkdir` en la sección 2:

```
.\ " Hey Emacs! This file is -*- nroff -*- source.
.\ "
.\ " This manpage is Copyright (C) 1992 Drew Eckhardt;
.\ "                               1993 Michael Haardt
.\ "                               1993,1994 Ian Jackson.
.\ " You may distribute it under the terms of the GNU General
.\ " Public License. It comes with NO WARRANTY.
.\ "
```

Implementando nuestra nueva página man

Finalmente, cuando nuestro archivo está terminado solo restará comprimirlo con `gzip`:

```
gzip -q /usr/share/man/man8/micomando.8
```

Por favor, notar que la opción `-q` del comando `gzip`, reemplazará el archivo `micomando.8` por `micomando.8.gz`.

Como puedes ver, las opciones de formato de `groff` son inmensas y aquí, solo hemos mencionado una mínima parte. Una buena forma de entender más a fondo tanto las macros de formato como la estructura interna de una página de manual, es mirando el fuente de los archivos, compararlo con la página del manual e ir tratando de implementar formatos similares y viendo el reflejado el resultado visual del mismo.

Y ahora ¡a documentar se ha dicho!

Monitorización del Sistema con Conky

En este artículo os voy a explicar cómo instalar y empezar a jugar con una de las herramientas más versátiles y configurables para monitorizar nuestro sistema: Conky.

Escrito por: **María José Montes Díaz** (Archera & Programadora)



Estudiante de Grado Ingeniería en Tecnología de la información. Técnico en informática de gestión. Monitora FPO. Docente de programación Python y Scratch para niños de 6-12 años. Activista del software libre y cultura libre.

Webs:

Blog: <http://blog.archninfo.org/>

Redes sociales:

Twitter: [@MMontesDiaz](https://twitter.com/MMontesDiaz)

Conky es un monitor de sistema para los sistemas X Windows. Se encuentra disponible para GNU/Linux, FreeBSD, OpenBSD. Está desarrollado bajo la licencia GPL. Es capaz de monitorizar CPU, memoria, swap, espacio de disco, temperaturas, mensajes de sistema, redes, plugins, y mucho más. Su configuración puede resultar difícil pero no imposible. Es un fork⁸ de [tormo](http://tormo.sourceforge.net/)⁹.

Instalamos:

Debian/Ubuntu

```
$ sudo apt-get install conky
```

Gentoo

```
# emerge app-admin/conky
```

8 Es la creación de un proyecto en una dirección distinta de la principal u oficial tomando el código fuente del proyecto ya existente

9 <http://tormo.sourceforge.net/>

O, si usamos [Paludis](#)¹⁰:

```
# paludis -i app-admin/conky
```

Fedora

```
$ sudo yum install conky
```

ArchLinux

```
$ pacman -S conky
```

Slackware

Buscamos en [Slackbuilds](#)¹¹ el Conky para nuestra versión, en mi caso 14.0. A continuación:

```
#Descargamos  
wget http://slackbuilds.org/slackbuilds/14.0/system/conky.tar.gz  
#Descomprimos  
tar xvfz conky.tar.gz  
#Nos situamos dentro de conky  
cd conky  
#Descargamos el código de Conky  
wget http://downloads.sourceforge.net/conky/conky-1.9.0.tar.bz2  
#Dentro del directorio conky ejecutamos:  
./conky.SlackBuild  
#Instalamos el paquete  
installpkg /tmp/conky-1.9.0-i486-1_Bo.tgz
```

10 <http://paludis.exherbo.org/>

11 <http://slackbuilds.org/>

Configuración

Lo primero que tenemos que hacer es una copia del archivo de configuración en nuestro home, con el nombre `.conkyrc`. El archivo por defecto se encuentra `/etc/conky/conky.conf`. Ejecutamos:

```
$ cp /etc/conky/conky.conf ~/.conkyrc
```

Como podéis ver, el archivo se divide en dos partes: La parte de configuración donde se indica el comportamiento y apariencia por defecto y, en la segunda, especificamos qué queremos mostrar: texto, imágenes, objetos, etc.

```
alignment top_left
background no
border_width 1
cpu_avg_samples 2
default_color white
default_outline_color white
default_shade_color white
draw_borders no
draw_graph_borders yes
draw_outline no
draw_shades no
use_xft yes
xftfont DejaVu Sans Mono:size=12

#Posición respecto alineación
gap_x 900
gap_y 300

minimum_size 5 5

net_avg_samples 2
no_buffers yes
out_to_console no
out_to_stderr no
extra_newline no
own_window yes
own_window_class Conky
own_window_type desktop

#Activamos transparencias
own_window_argb_visual yes

# yes: Forzamos argb_value a 0
own_window_transparent no

#Si no queremos transparencia total
own_window_argb_value 100 #valor entre 0-255

double_buffer yes
stippled_borders 0
update_interval 1.0
```

```

uppercase no
use_spacer none
show_graph_scale yes
show_graph_range yes

#Empezamos con el contenido
TEXT
${scroll 35 $nodename - $sysname $kernel on $machine | }
$hr
${color grey}Uptime:$color $uptime
${color grey}Frequency (in MHz):$color $freq
${color grey}Frequency (in GHz):$color $freq_g
${color grey}RAM Usage:$color $mem/$memmax - \
$memperc% ${membar 4}
${color grey}Swap Usage:$color $swap/$swapmax - \
$swapperc% ${swapbar 4}
${color grey}CPU Usage:$color $cpu% ${cpubar 4}
${color grey}Processes:$color $processes ${color grey}\
Running:$color $running_processes
$hr
${color grey}File systems:
 / $color${fs_used /}/${fs_size /} ${fs_bar 6 /}
$hr
${color grey}Networking:
${if_up eth0} \
eth0 Up:$color ${upspeed eth0} ${color grey} - Down:$color ${downspeed eth0}
${endif}\
${if_up wlan0} \
${color grey}wlan0 Up:$color ${upspeed wlan0} ${color grey} - Down:$color \
${downspeed wlan0}
${endif}\
$hr
${color grey}Name          PID   CPU%   MEM%
${color red}  ${top name 1}  ${top pid 1}  ${top cpu 1}  ${top mem 1}
${color lightgreen}  ${top name 2}  ${top pid 2}  ${top cpu 2}  ${top mem 2}
${color blue}  ${top name 3}  ${top pid 3}  ${top cpu 3}  ${top mem 3}
${color lightblue}  ${top name 4}  ${top pid 4}  ${top cpu 4}  ${top mem 4}
${color yellow}  ${top name 5}  ${top pid 5}  ${top cpu 5}  ${top mem 5}
${color lightyellow}  ${top name 6}  ${top pid 6}  ${top cpu 6}  ${top mem 6}

```

La configuración anterior es una modificación de la original. Además de añadir más procesos (los seis primeros), he utilizado la cláusula **if_up** para controlar, en el apartado de **Networking**, si las interfaces existen y están levantadas.

Para una mejor lectura, si queremos hacer un salto de línea en el archivo, podemos utilizar el carácter de escape \.

Todas las opciones de configuración las podemos encontrar en http://conky.sourceforge.net/config_settings.html

Las variables que podemos utilizar en cuerpo de Conky, es decir, el texto que queremos que se muestre, las podemos encontrar en <http://conky.sourceforge.net/variables.html>

Para ver y descargar diferentes configuraciones, podéis visitar <http://conky.sourceforge.net/screenshots.html> , que nos darán ideas para personalizar nuestro Conky.

Evitando el parpadeo

Esto ocurre porque necesita el soporte Double Buffer Extension (DBE) del servidor X, ya que sin este soporte no logrará actualizar la pantalla con suficiente rapidez.

La solución:

Primero comprobamos el contenido del archivo Xorg.0.log

```
# cat /var/log/Xorg.0.log | grep dbe
```

nos debe salir algo así,

```
# [ 78.931] (II) LoadModule: "dbe"  
# [ 78.933] (II) Loading /usr/lib/xorg/modules/extensions/libdbe.so  
# [ 78.936] (II) Module dbe: vendor="X.Org Foundation"
```

A partir de la versión 1.8.x, este módulo se carga automáticamente. Si usamos una versión anterior, bastará añadir al **/etc/X11/xorg.conf** la línea **Load "dbe"** dentro de la sección **"Module"**.

Después, editamos el archivo **/etc/conky/conky.conf** y escribimos la siguiente opción debajo del resto, pero antes de **TEXT** o **XY**

```
double_buffer yes
```

Para que se ejecute al inicio de la sesión:

```
nano ~/.config/autostart/conky.desktop
```

Añadimos lo siguiente:

```
[Desktop Entry]  
Encoding=UTF-8  
Version=0.9.4  
Type=Application  
Name=conky  
Comment=  
Exec=conky -d  
StartupNotify=false  
Terminal=false  
Hidden=false
```

Al iniciar nuestro sistema tendremos el Conky funcionando:



Enlaces de interés:

<http://conky.sourceforge.net/>

<http://torsmo.sourceforge.net/>

<https://wiki.archlinux.org/index.php/Conky>

<http://slackbuilds.org/>

Creando capturas de pantalla para tus manuales con Shutter

Realizar capturas de pantalla de calidad no es algo tan simple si no cuentas con una herramienta que te lo permita y sobretodo de manera sencilla, Shutter es una gran alternativa para lograrlo.

Escrito por: **Milagros Alessandra Infante Montero** (Est. Ing. Informática)



Estudiante de Ingeniería Informática. Miembro de la comunidad de software libre [Lumenhack](#). Miembro del equipo de traducción al español de [GNOME](#). Apasionada por el desarrollo de software, tecnología y gadgets. Defensora de tecnologías basadas en software libre y de código abierto.

Webs:

Blog: www.milale.net

Redes sociales:

Twitter / Identi.ca: [@milale](#)

Cuántas veces hemos tenido dificultades para sacar screenshots de calidad de diferentes maneras por no contar con una buena herramienta, Shutter nos ayuda enormemente a ello. Este programa tiene muchas características interesantes que lograrán que se convierta en una herramienta de bastante ayuda para todos.

Al realizar capturas de pantalla podemos necesitar que sea de la pantalla completa, de una ventana o de alguna área específica; después de tenerla podemos necesitar aplicar algún efecto en particular, dibujar o resaltar algo en la imagen y podemos querer finalmente subirla a algún host de imágenes; realizar todo esto es posible con Shutter, esta herramienta es libre, de código abierto y con licencia GPL v3.

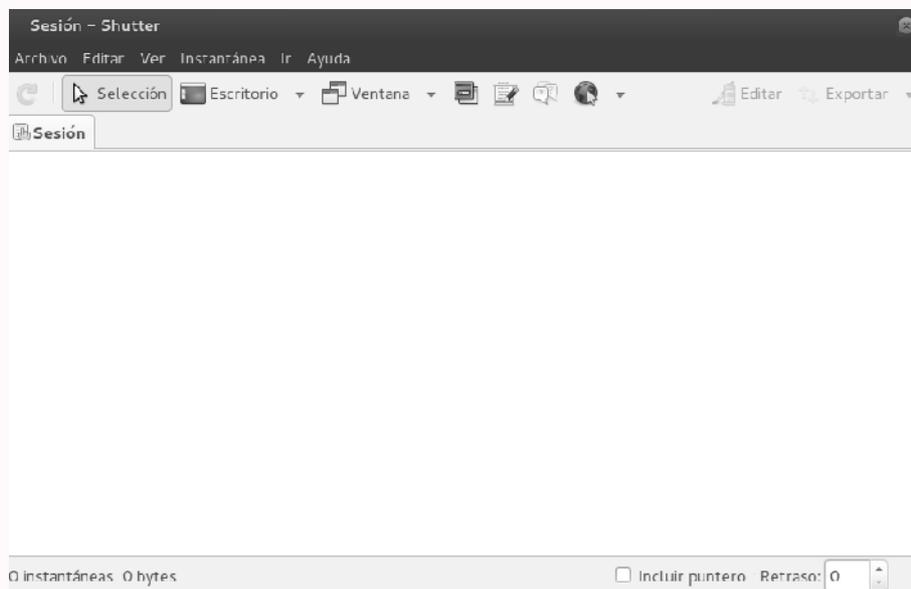
¿Cómo conseguimos Shutter?

Shutter¹² está disponible para la mayoría de distribuciones de GNU/Linux, en mi caso estoy usando Ubuntu 13.04 Raring Ringtail y puedes conseguirla directamente desde el centro de software de Ubuntu o escribiendo estas líneas en la terminal

¹² <http://shutter-project.org/downloads/>

```
sudo add-apt-repository ppa:shutter/ppa
sudo apt-get update && sudo apt-get install shutter
```

Después de esto, ya Shutter está listo para usar:



Capturas de pantalla

Shutter permite hacer capturas de pantalla de diferentes maneras que veremos a continuación:

- **Área específica**

Al elegir la opción “Selección”, Shutter nos permite elegir cierta área arbitrariamente en la pantalla, al capturarla podemos editarla, quizás añadirle imágenes para llamar la atención sobre algo o resaltar algún dato importante como se muestra a continuación.

Últimas Ediciones de Hackers & Developers Magazine		
Fecha	Nro. de Edición	Codename
24.06.2013	 NEW Hackers & Developers Magazine Nº 8	Horse
27.05.2013	Hackers & Developers Magazine Nº 7	She
22.04.2013	Hackers & Developers Magazine Nº 6	Faraón
25.03.2013	Hackers & Developers Magazine Nº 5	Sweet
25.02.2013	Hackers & Developers Magazine Nº 4	Narciso
28.01.2013	Hackers & Developers Magazine Nº 3	Elvis
31.12.2012	Hackers & Developers Magazine Nº 2	Champagne (Ed. Especial)
03.12.2012	Hackers & Developers Magazine Nº 1	Cobra
05.11.2012	Hackers & Developers Magazine Nº 0	Butterfly

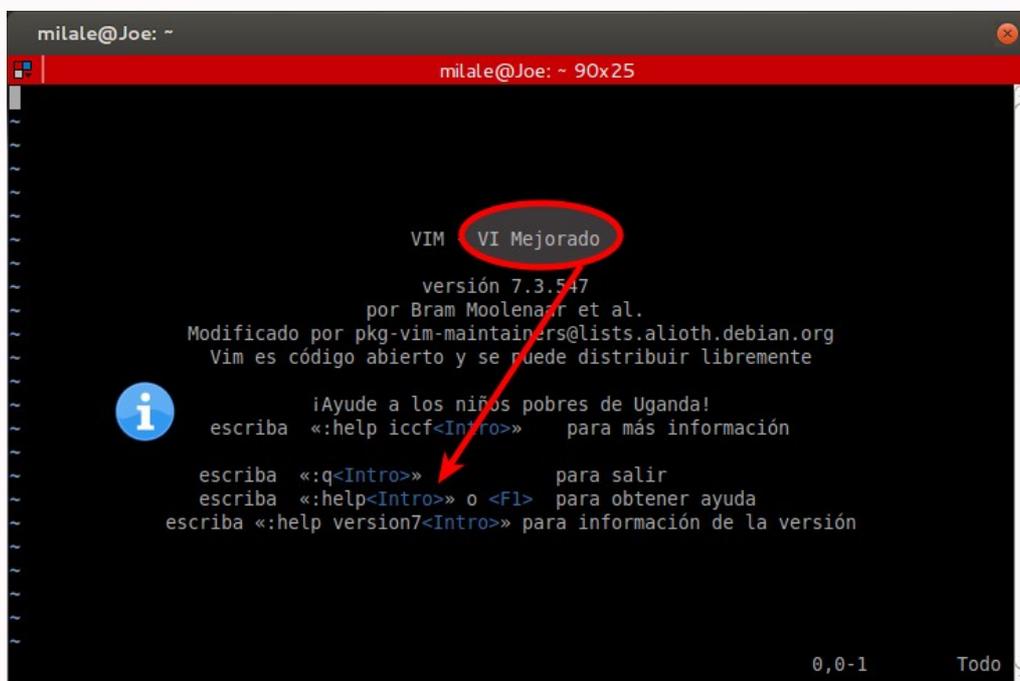
- **Escritorio**

Shutter tiene la opción de poder capturar todos los contenidos de el o los escritorios con los que trabajemos, cuenta con opciones para seleccionar el escritorio actual o seleccionar un área de trabajo específica. La siguiente captura por ejemplo es de todas las áreas de trabajo que tengo ahora.

- **Ventana**

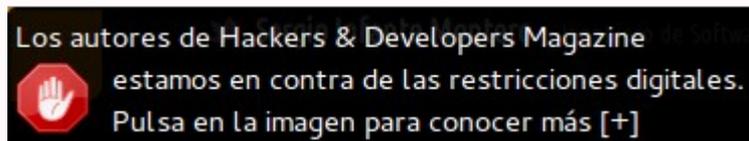
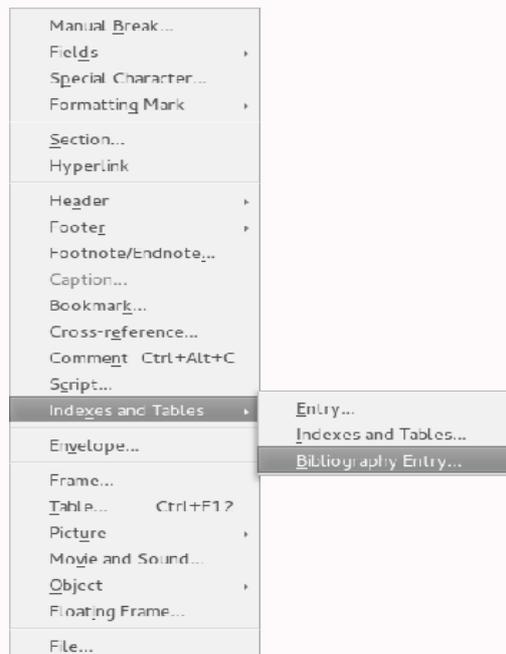
Cuando se realizan capturas de ventanas específicas, por ejemplo al momento de escribir un manual y necesitar de screenshots para complementar el texto, generalmente se necesita recortar solo la ventana que deseamos.

Podríamos realizarlo con una captura de área específica, sin embargo, Shutter nos da la posibilidad de elegir la ventana que se quiere capturar en un listado y el programa automáticamente recortará solo el área deseada.



- **Menú o tooltip**

Otro tipo de captura de pantalla que podemos necesitar al momento de escribir un manual o dar alguna indicación es la captura de algún menú o tooltip, por ello Shutter también nos ayuda a realizar este tipo de imagen. A continuación podemos ver la captura que saque con Shutter del menú "Insert" en LibreOffice y en la imagen siguiente podemos ver el aviso emergente que aparece al pasar el cursor del mouse sobre la imagen de "Authors against DRM" en nuestra página.



- **Sitio web**

Shutter nos permite realizar la captura de pantalla de un sitio web sin tener que abrir la página en un navegador necesariamente, ingresamos la página, por ejemplo: www.hdmagazine.org/donar y listo la captura se realizará automáticamente en un instante.

Shutter y la edición

Ya hemos visto las distintas formas que Shutter nos ofrece al momento de realizar capturas de pantalla y como coloqué en algunos ejemplos se pueden añadir elementos interesantes que permitan una mejor visión de las capturas que realicemos.

Cuando estés trabajando con Shutter no es necesario que hagas uso de un editor de gráfico externo como GIMP por ejemplo, ya que este programa cuenta con su propio editor incorporado, los detalles que pueden añadirse son por ejemplo:

- **Añadir texto, flechas y demás formas**

Shutter como hemos visto permite añadir flechas y otras formas (rectángulos,



círculos, etc) a la captura que hemos realizado e incluso se puede personalizar más cambiando el color, la fuente o el ancho de línea.

- **Censurar o pixelar para esconder datos privados**

Al realizar capturas de pantalla podemos necesitar esconder alguna información que no podamos compartir, para estos casos se pueden censurar o pixelar datos.

- **Formas auto-incrementadas**

Al escribir un manual de pasos, las personas suelen agregar dígitos incrementados usualmente como texto, Shutter permite lograrlo de manera sencilla dando clic en las áreas específicas que se desea y aparecerá una forma con dígitos que nos hará todo más simple.

- **Recortado**

Shutter permite que se recorte determinada parte de la captura realizada para quizás ingresar las dimensiones en los campos y obtener lo que se desea.

EUROPIO engine

Huscar...

DESCRIPCIÓN | DESCARGAS | EMPRESAS | Versión actual

Motor MVC para aplicaciones modulares PHP

europioengine es un motor de software para crear aplicaciones modulares con PHP y MySQL bajo el patrón arquitectónico modelo-vista-controlador permitiendo un estilo 100% orientado a objetos así como la posibilidad de diseñar clases orientadas a datos (pueden o no el programador lo desea) no sujeta a una DDD estricta. EuropioEngine es desarrollado y mantenido por Eugenia Bahit y se distribuye bajo licencia GNU GPL v3.0.

Documentación

Guía completa paso a paso y manual de referencias para iniciarse con europioengine y explorarlo en profundidad.

[Aprende](#) 1

Código Fuente (GPL v3)

EuropioEngine es Software Libre. Descarga el código fuente completo y comienza a crear tu propio Framework o desarrollar tus apps.

[Descarga](#) 3

Live Demo

En este enlace podrás ver una aplicación (a modo de ejemplo) creada con europioengine como motor de su arquitectura MVC.

[Prueba](#) 2

Créditos

Arquitectura & programación:
Eugenia Bahit

Testing:
Jimmy Daniel Sarranco

Mis otros proyectos
[python print](#)
Módulo que emula la función print() de PHP para imprimir objetos en pantalla de forma recursiva.

[Mas información](#)

Historial de Cambios

Version 3.2 Beta 3
Bugfixing

- CollectorObject no verificaba si el objeto heredaba de ComposerObject, generando un error en la colección de subtipos de éste.
- El helper del ORM generaba conflicto con propiedades cuyos nombres contenían guiones bajos (_).
- El AppEngine no hacía la importación de composerobject.

Version 3.2 Beta 2
Agregado Composer Object: objeto composer especialmente diseñado para crear compositores exclusivos en relaciones muchos a uno.

[+] Ver historial de cambios completo

© 2009-2013 Eugenia Bahit
Contenido publicado bajo una licencia GNU Free Documentation License
Website Template by TemplateMonster.com | <http://html5xss3.com>

En esta imagen capture el website colocando la dirección www.europio.org, en la parte de la versión use la opción de censurar, en el menú pixelee el menú Changelog, use la forma auto-incrementada para numerar 3 pasos, coloqué un rectángulo en la parte de créditos y subraye las distintas versiones para que se puedan percatar de las diferencias.

Compartir con Shutter

En nuestro día a día y sobretodo en las muchas horas que pasamos frente a la pc, navegamos por foros, wikis y muchas páginas más y podemos tener la necesidad de realizar screenshots y luego subirlos a la nube para así poder compartirlo con otras personas. Shutter también ha pensado en ello, cuenta con una función incorporada que la sube a la nube sin necesidad de que abramos una página en el navegador para lograrlo e incluso la última versión ya tiene soporte para Ubuntu One. Al seleccionar la opción de exportar seleccionamos una cuenta de alguna página e inmediatamente después obtenemos los links para compartir; también es necesario indicar como los plugins con los que cuenta ofrecen efectos impresionantes a las capturas de pantalla haciéndolas únicas.

Subí a la nube el screenshot que realice para mostrar la captura de todas las áreas de trabajo mediante Imgur y este es el link que obtuve <http://i.imgur.com/y0Lj6F2.jpg> donde pueden ver la imagen original.

Una imagen vale más que mil palabras. |

Colaborar con Shutter

Así como hemos podido comprobar, Shutter es una herramienta genial que permite realizar diferentes capturas de pantalla con estilos y demás que permiten obtener imagenes de calidad con todo lo que nos sea necesario.

Shutter es un programa de código abierto y disponible totalmente gratis, se puede contribuir con este proyecto de distintas maneras, haciendo preguntas de modo que sirva y quede como una base de conocimiento¹³, también se pueden crear reportes de bugs ya que como ellos dicen no pueden arreglar algo que no sepan que pasa y reportarlo¹⁴ solo tomará un momento, también se puede contribuir mediante traducciones¹⁵ siguiendo las indicaciones que se indiquen y finalmente contribuir con el proyecto de manera económica¹⁶.

13 <https://answers.launchpad.net/shutter>

14 <https://bugs.launchpad.net/shutter>

15 <https://translations.launchpad.net/shutter>

16 <http://flatr.com/thing/23370/Shutter-Screenshot-Tool>

Trabajando con GIT II

En esta segunda entrega hablaremos un poco mas de comandos mas avanzados y que nos dan muchas mas funcionalidades a la hora de trabajar con GIT.

Escrito por: **Indira Burga** (Ingeniera de Sistemas)



Indira es **Ing. de Sistemas** de Perú. Gestora de Proyectos de desarrollo de software, **programadora PHP**, analista, nueva amante de las **metodologías Ágiles**. Ahora envuelta en una nueva aventura: su propia empresa "**IC Projects**" dedicada al desarrollo de Software.

Webs:

About.me: <http://about.me/indirabm>

Redes sociales:

Twitter: [@indirabm](https://twitter.com/indirabm)

En este artículo hablaremos de comandos mas avanzados que nos librarán de más de un problema, imagínense que desean mover los commits de una rama a otra o tal vez desean eliminar archivos de todo el directorio de trabajo, solo enviar un commit a una rama, unir commits o separar commits, pero ¿como lo hacemos? Bueno a continuación daremos un vistazo por los comando que nos ayudaran a hacer lo antes mencionado y mas.

Fetch - Pull

Cuando hacemos "git fetch" lo único que se hace es actualizar la copia local, esto significa que no se modificara ningún archivo que tiene actualmente.

Cuango hacemos "git pull" en realidad hacemos "git fetch" y "git merge".

Ejemplo:

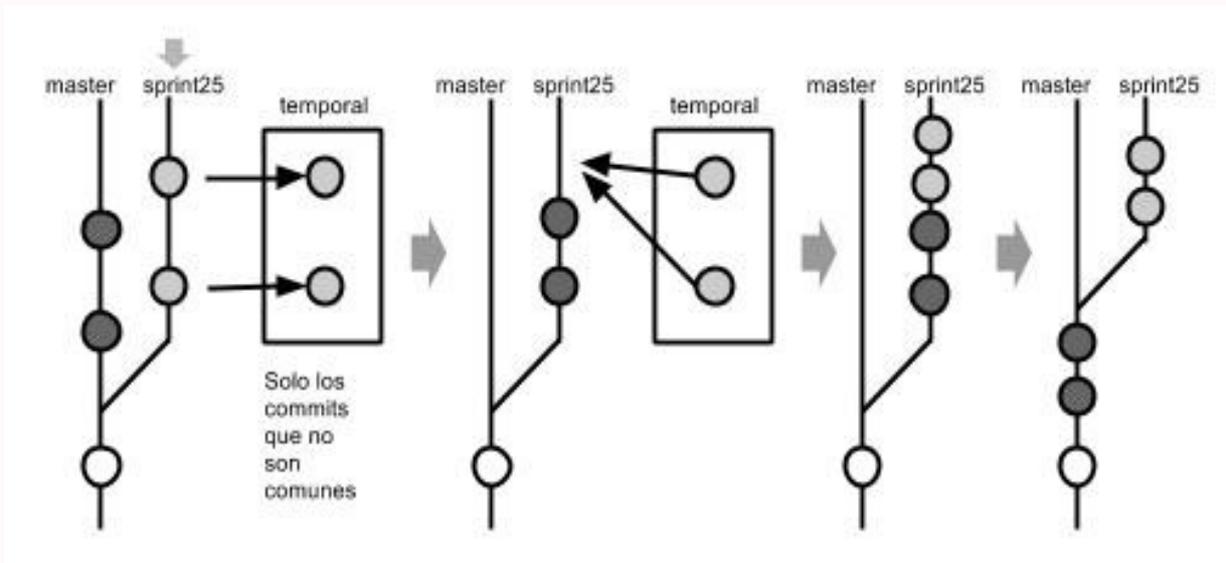
Si deseamos comprobar cuales han sido los cambios reales desde el ultimo push y asi estar seguro de modificar tus archivos.

```
git fetch
//podremos ver las diferencias que hay y realizar un git merge
git diff origin
```

Rebase

Con este comando básicamente podemos reorganizar nuestros commits en diferentes ramas, a continuación explicare que es lo que hace rebase paso a paso.

```
git checkout sprint25
git rebase master
```



Como habrán podido ver este comando es muy útil, ahora bien digamos que lo que deseamos es intercambiar el orden de los commits o separar un commit en dos, para esto se usa el rebase interactivo (-i).

Como reorganizar commits:

```
git rebase -i HEAD~3
// HEAD~3 significa hasta q commit anterior se desea ver
```

Se abrirá una ventana donde vera los commits, ahí tendrá diferentes opciones

```
pick call9a8 Desktop y sus ventanas. Mejo
pick 6d36f75 Implementando la arquitectura
pick 4226f41 Cambios para nuevo usuario
```

- Reordenar los commits, con solo intercambiar el orden en el que aparecen.
- Reword, Editar los mensajes del commit, solo es necesario cambiar pick por el reward
- Squash, sirve para separar o unir uno o mas commits

Ejemplo de separar commits:

```
//cambiar pick por squash, podrá ver los archivos del commit y determinara que  
//archivos enviara en cada commit  
git reset HEAD^  
git add <archivo>  
git commit -m "mensaje"  
git add <archivo 2>  
git commit -m "mensaje 2"  
git rebase --continue
```

Stash

Guarda los cambios hechos desde el ultimo commit.

Ejemplo:

Tu jefe te llama y te dice, hay un error en una funcionalidad y es urgente que se pase a producción, sin embargo aun no has finalizado tu tarea y no puede hacer commit, ahi entra en stash.

```
git stash save  
//Puedes ir a otra rama a arreglar el error  
git stash apply  
//puedes continuar tu trabajo
```

Conflictos con Stash:

Si muestra conflicto al ejecutar stash apply

```
git stash apply  
git reset --hard HEAD  
git stash apply
```

Si muestra conflicto al ejecutar stash pop

```
git stash pop  
git stash list  
git stash drop
```

Si deseamos que los cambios del área de preparación (staging area)

```
//Solo los archivos que no están commiteados  
git stash save --keep-index  
git status  
git commit -m "mensaje"  
git stash pop
```

Comandos importantes al usar stash:

```
// Listar todos los stash actuales
git stash list

// Continuar trabajando en un stash especifico en este caso el penultimo
git stash apply stash@{1}

// Eliminar un stash
git stash drop

// Recupera datos, ejecuta stash apply y drop
git stash pop

// Muestra un stash determinado o el mas reciente
git stash show stash@{1}
git stash show

// Que pasen al stash los archivos que que no están en seguimiento (untracked)
git stash save --include-untracked

// Sumario de cambios
git stash list --stat

// Poner un mensaje al stash
git stash save "mensaje"

// Recuperar rama que a sido borrada después de haber hecho stash save
git stash branch <rama> stash@{1}

// Limpiar la lista de los stash
git stash clear
```

Tree- Filter

Git comprobará cada commit ejecutado en el directorio de trabajo, ejecutara el comando que esta entre comillas y volverá hacer commit.

Ejemplo:

Eliminando archivo de todos los commits.

```
// Reemplazar --all por HEAD si solo se desea eliminar de la rama actual
git filter-branch --tree-filter 'rm -f <archivo>' -- --all
```

Se recomienda siempre hacer un backup.

Index- Filter

Git ejecutará el comando en cada commit, pero sin comprobar primero (por lo que es más rápido).

```
git filter-branch --index-filter 'git rm --cached --ignore-unmatch <archivo>'
```

Al ejecutar *filter-branch* Git deja una copia de seguridad de su árbol en el directorio ". git", para finalizar la eliminación se debe de ejecutar el siguiente comando

```
git filter-branch -f --tree-filter 'rm -f <archivo>'
```

Es probable que al eliminar un archivo de todo el directorio de trabajo deje commits sin contenido, para ello se ejecuta el siguiente comando.

```
git filter-branch -f --prune-empty -- --all
```

Cherry-pick

Enviar un commit de una rama a otra

```
git log --online  
//ver el sha del commit deseado  
git checkout <rama>  
git cherry-pick <sha>
```

Editar el mensaje del sha

```
git cherry-pick --edit <sha>
```

Enviar dos commits unidos a otra rama

```
git log --online  
//ver los shas del commit deseado  
git cherry-pick --no-commit <sha1> <sha2>  
git status  
git commit -m "mensaje"
```

Visualizar el origen del cherry-pick, solo funciona el ramas publicas

```
git cherry-pick -x <sha>
```

Seguimiento de quien fue el que hizo el commit del cherry-pick

```
git cherry-pick --signoff <sh>
```

GNU/Linux para programadores:

KILL & KILLALL: Asesinato en Primer Grado

Cuando recién te pasas a GNU/Linux uno de los problemas más frecuentes con el que te encuentras es que tarde o temprano, algún programa se “cuelga” y comienzas a experimentar el síndrome del Ctr + Alt + Supr heredado de tu anterior sistema privativo. Pero no sufras: existe una cura.

Escrito por: **Eugenia Bahit** (GLAMP Hacker & eXtreme Programmer)



Eugenia es **Arquitecta de Software**, docente e instructora de tecnologías **GLAMP** (GNU/Linux, Apache, MySQL, Python y PHP) y **Agile coach** (UTN) especializada en Scrum y **eXtreme Programming**. Miembro de la **Free Software Foundation**, **The Linux Foundation** y **Debian Hackers**. Creadora de **python-printr**, **Europio Engine** y colaboradora de **Vim**.

Webs:

Cursos de programación: www.cursosdeprogramacionadistancia.com

Web personal: www.eugeniabahit.com

Redes sociales:

Twitter / Identi.ca: [@eugeniabahit](https://twitter.com/eugeniabahit)

Según la *distro* que utilices, los programas que instales, las fuentes desde las cuales los instales, los recursos de tu ordenador y la forma en la que “maltrates” a tu Sistema Operativo, los programas -en modo gráfico generalmente- experimentarán “cuelgues” que, si recién te inicias en GNU/Linux, provocarán la reinstalación de tu sistema al grito de diversos improperios. Pero... tranquilo: no es grave y no necesitas reinstalar el sistema cuando algo no resulta como lo esperabas. Por el contrario, cerrar “por la fuerza” un programa “colgado” es mucho más simple de lo que te imaginas.

Ahora, antes de continuar, te haré la misma recomendación que hago constantemente a mis alumnos: **siempre, deja una terminal abierta y maximizada**.

La diferencia entre un Sistema Operativo privativo popular y uno libre como el Sistema

Operativo GNU con Linux, es que este último te permite mantener un control absoluto sobre él mismo, haciéndolo independiente de su interfaz gráfica. Y no. Que no hayas pagado por tu *distro*, no significa que la diferencia sea de precio. De hecho, cuando hablamos de libre y privativo, simplemente hablamos de tu libertad para controlar lo que tienes y no del precio de lo que adquieres. Y en los Sistemas Operativos GNU, dicho control, lo tienes por intermedio de la terminal en el 99,9% de los casos.

Entonces ¿para qué cambiarte de Sistema Operativo si no vas a hacer uso de sus ventajas? Si intentas utilizar GNU/Linux como tu antiguo Sistema Operativo privativo, fracasarás en la experiencia. Así que lo mejor que puedes hacer, es iniciar una amistad con la terminal y dejarla siempre abierta y maximizada.

Tener la terminal abierta te ayudará a muchísimas cosas, pero por sobre todo, te facilitará la vida cuando un programa, eventualmente, se quede congelado.

¿Se colgó? Entonces ¡mátalo!

El tema es que se congeló tu programa ¿entonces...? Entonces, es hora de asesinar a un proceso. Para ello, lo primero que necesitas es conocer el nombre del proceso. Éste, generalmente, es similar al nombre del programa. Por ejemplo, el proceso correspondiente al navegador *Firefox*, se llama *firefox*.

Pero no siempre sabrás el nombre con exactitud y para ello, hay una solución: los procesos que están corriendo en tu sistema, puedes obtenerlos con el comando `ps` con los siguientes argumentos:

```
user@host:~$ ps aux
```

El comando `ps` arrojará una lista similar a la siguiente:

```
root      2073  0.0  0.0 15124 1428 ?        Ss   Apr04  0:02 sendmail: MTA: accepting connections
eugenia   2075  0.0  0.3 76264 7672 ?        Sl   Apr04  0:00 telepathy-indicator
eugenia   2079  0.0  0.3 42152 6112 ?        Sll  Apr04  0:01 /usr/lib/telepathy/mission-control-5
eugenia   2084  0.0  0.2 91208 5284 ?        Sl   Apr04  0:00 /usr/lib/gnome-online-accounts/goa-daemon
root      2147  0.0  0.2 37764 5392 ?        Ss   Apr04  0:02 /usr/sbin/apache2 -k start
root      2188  0.0  0.0 4632  812 tty1    Ss+  Apr04  0:00 /sbin/getty -8 38400 tty1
www-data  2190  0.0  0.3 39108 5948 ?        S    Apr04  0:00 /usr/sbin/apache2 -k start
www-data  2191  0.0  0.3 39108 5948 ?        S    Apr04  0:00 /usr/sbin/apache2 -k start
www-data  2192  0.0  0.3 39108 5948 ?        S    Apr04  0:00 /usr/sbin/apache2 -k start
www-data  2193  0.0  0.3 39108 5948 ?        S    Apr04  0:00 /usr/sbin/apache2 -k start
www-data  2194  0.0  0.3 39108 5948 ?        S    Apr04  0:00 /usr/sbin/apache2 -k start
eugenia   2195  0.0  0.3 55688 6528 ?        Sl   Apr04  0:14 zeitgeist-datahub
eugenia   2201  0.0  0.2 44324 4380 ?        Sl   Apr04  0:01 /usr/bin/zeitgeist-daemon
eugenia   2207  0.0  0.2 52072 5124 ?        Sl   Apr04  0:01 /usr/lib/zeitgeist/zeitgeist-fts
eugenia   2215  0.0  0.0 5448  252 ?        S    Apr04  0:00 /bin/cat
eugenia   2216  0.0  0.5 73068 10056 ?       Sl   Apr04  0:03 gnome-screensaver
```

Solo que la lista, será muchísimo más extensa. Así que un método corto para obtener el programa que buscas, será aplicar un filtro sobre esa salida con el comando `grep` y utilizando como patrón de búsqueda el nombre del programa o parte de él:

```
user@host:~$ ps aux | grep gedit
```

Obtendrás algo como lo siguiente:

```
eugenia 18519 16.5 1.9 121956 38600 ? Sl 15:51 0:01 gedit
eugenia 18542 0.0 0.0 5628 836 pts/0 S+ 15:51 0:00 grep --color=auto gedit
```

El dato que te interesa, será generalmente el del primer resultado que aparezca en la lista:

```
eugenia 18519 16.5 1.9 121956 38600 ? Sl 15:51 0:01 gedit
```

De ese registro, dos columnas serán las que deberás observar: la de ID del proceso (PID):

```
eugenia 18519 16.5 1.9 121956 38600 ? Sl 15:51 0:01 gedit
```

Y la del nombre del comando (COMMAND):

```
eugenia 18519 16.5 1.9 121956 38600 ? Sl 15:51 0:01 gedit
```

Con esa información ya puedes “asesinar” al proceso, ya sea con el comando killall más el nombre del proceso:

```
killall COMMAND
```

O con el comando kill más la ID del proceso:

```
kill PID
```

Siguiendo el ejemplo anterior, para matar el proceso por nombre ejecutaríamos:

```
killall gedit
```

Y para matarlo por su ID:

```
kill 18519
```

Para verificar que un proceso haya terminado efectivamente, puedes ejecutar `killall` con el argumento `-w`:

```
killall -w COMMAND
```

Si el proceso finalizó, verás un mensaje de error como el siguiente:

```
killall -w gedit  
gedit: proceso no encontrado
```

Es muy importante que para esta verificación utilices `killall` y no `kill`, ya que podría suceder que tras matar un proceso se reinicie de forma automática y para entonces, su PID habrá cambiado.

El poderoso virtualenvwrapper

En muchos proyectos con python *virtualenv* hace la vida mas simple y organizada al equipo de desarrollo, *virtualenvwrapper* lo hace aún mejor.

Escrito por: **Sergio Infante Montero** (Ingeniero de Software)



Ingeniero Informático con estudios de Master de Dirección Estratégica en TI. Senior Software Developer en [Belatrix](#), activista, contribuidor y consultor de proyectos **FLOSS**. Miembro de [GNOME Foundation](#). Escritor de artículos y libros técnicos de programación.

Perfiles:

<http://about.me/neosergio>

Twitter: [@neosergio](#)

Virtualenv es una herramienta que permite crear entornos privados para el desarrollo con Python, su uso es bastante extendido y se ha convertido en un estándar de facto para cualquier proyecto serio con este estupendo lenguaje. Su uso permite manejar dependencias, versiones e indirectamente permisos para un proyecto, sin afectar al sistema en general o a otros proyectos en el mismo computador o servidor.

Aprender a usar virtualenv es muy sencillo, y existen muchos tutoriales y artículos por todo internet que permiten aprender a usarlo muy rápidamente.

Virtualenvwrapper es una serie de extensiones para virtualenv, la ventaja importante es que contiene una especie de envolturas, llamadas también **wrappers**, que permiten gestionar entornos virtuales y manejar el flujo del desarrollo, haciendo más fácil el trabajo en mas de un proyecto a la vez, evitando el conflicto entre dependencias de los mismos.

Instalación

La mejor forma de instalar virtualenvwrapper es a través de pip, y bastaría escribir:

```
pip install virtualenvwrapper
```

o también podría ser a través de `easy_install`:

```
easy_install virtualenvwrapper
```

Una vez instalado, se debe agregar las siguientes dos líneas a `.bashrc` para establecer la ubicación de los entornos virtuales y la ubicación del script de `virtualenvwrapper`:

```
export WORKON_HOME = $HOME/.virtualenvs  
source /usr/local/bin/virtualenvwrapper.sh
```

Ahora sólo recargar `.bashrc` para poder usar los **wrappers**:

```
source ~/.bashrc
```

Con estos simples pasos ya estamos listos para usarlo.

Wrappers

El primero de todos los wrappers es el usado para crear un nuevo entorno virtual este sería el siguiente comando:

```
mkvirtualenv nombre_del_entorno
```

Donde el `nombre_del_entorno` se puede reemplazar por el nombre que se desee, por ejemplo podríamos tener uno para `hdmagazine` de la siguiente manera:

```
mkvirtualenv hdmagazine
```

Al hacer esto el terminal imprimiría algo como esto:

```
New python executable in hdmagazine/bin/python  
Installing setuptools.....done.  
Installing pip.....done.
```

Y el prompt tendría el nombre del entorno entre paréntesis al principio algo como esto:

```
(hdmagazine)sergio@foobar:~$
```

Todas las dependencias de python que se instalen permanecerán dentro de este entorno.

Para salir del entorno se usa el siguiente wrapper:

```
deactivate
```

Y el prompt volvería a tener la siguiente forma:

```
sergio@foobar:~$
```

Para volver a ingresar al entorno se usa:

```
workon nombre_del_entorno
```

Y nuevamente volveríamos a entrar en el entorno, para seguir el ejemplo anterior se usaría

```
workon hdmagazine
```

Y se volvería a cambiar el prompt indicando que estamos dentro del entorno listo para trabajar dentro de él.

Si se ha olvidado los nombres de los entornos creados, solamente con `workon`, se encontraría un listado de todas las disponibles en el ordenador, para borrar alguna, se puede usar el siguiente wrapper:

```
rmvirtualenv nombre_del_entorno
```

Es necesario notar que sólo se puede borrar entornos que actualmente se encuentra desactivados, eso quiere decir que hay que salir del entorno si se desea eliminar el actual.

Dos de los wrappers más útiles cuando se trata de configurar el entorno, son los que nos ubican en los directorios donde se encuentran las dependencias y librerías, estos serían

```
cdvirtualenv
```

El anterior lleva a `~/virtualenvs/nombre_del_entorno` y el siguiente:

```
cdsitepackages
```

Lleva a `~/virtualenvs/nombre_del_entorno/lib/python2.7/site-packages` (tener en cuenta que el número de versión de python depende de lo que se tenga instalado en el ordenador, no es una regla que sólo funcione en python 2.7)

Esos son los wrappers más comunes.

Hooks

El uso de hooks es una funcionalidad muy estupenda de virtualenvwrapper permiten extender la funcionalidad, veamos los más importantes.

Para poder configurar acciones luego de activarse el entorno podemos usar `postactivate`:

```
cdvirtualenv
```

Eso nos llevaría a la carpeta del entorno actual y luego usaríamos un editor de texto para modificar el `bin/postactivate`, por ejemplo se puede usar vim y ejecutar lo siguiente:

```
vim bin/postactivate
```

Encontraras un archivo de texto vacío o con una o dos líneas de comentarios, aquí podemos definir las ordenes que se ejecutarán después de activarse, por ejemplo que vaya a un directorio específico (asumiendo que `~/Code` existe):

```
cd ~/Code
```

De tal manera que cada vez que se active el entorno, automáticamente se cambiará al directorio `~/Code`

```
(hdmagazine)sergio@foobar:~/Code$
```

Básicamente lo que tenemos son una especie de atajos a acciones que se ejecutarán dependiendo de la acción de los wrappers, si hacemos un listado dentro de **`bin/`** veremos que se pueden configurar, acciones para:

- postactivate (la del ejemplo anterior)
- postdeactivate
- preactivate
- predeactivate

Pero si vamos a la carpeta donde se encuentran todos los entornos

```
cd ~/.virtualenvs
```

Y hacemos un listado veremos de que se pueden configurar para:

- postmkvirtualenv
- postrmvirtualenv
- premkvirtualenv
- prermvirtualenv
- postmkproject
- postrmproject
- premkproject
- prermproject

Proyectos

La lista anterior tiene 4 hooks que nos dan el indicio de que se puede usar virtualenvwrapper para hacer proyectos, veamos como funciona esto, para poder probarlo vamos a hacer un ejemplo:

Creamos primero una carpeta donde se encontrara todos nuestros proyectos:

```
mkdir ~/proyectos
```

Ahora tenemos que editar **.bashrc** para agregar un nueva línea, para mantener el orden lo haremos entre las otras dos anteriores, no olvidar recargar **.bashrc**:

```
export WORKON_HOME=$HOME/.virtualenvs
export PROJECT_HOME=$HOME/proyectos
source /usr/local/bin/virtualenvwrapper.sh
```

Una vez hecha esta configuración basta con usar el wrapper mkproject de la siguiente

manera:

```
mkproject hdmagazine2
```

Para obtener este resultado, que nos indica que además de crear el entorno, se está creando una carpeta para un proyecto que use el entorno creado:

```
New python executable in hdmagazine2/bin/python
Installing setuptools.....done.
Installing pip.....done.
Creating /home/sergio/proyectos/hdmagazine2
Setting project for hdmagazine2 to /home/sergio/proyectos/hdmagazine2
```

Y nos daremos cuenta que ahora el prompt tiene la siguiente forma:

```
(hdmagazine2)sergio@foobar:~/proyectos/hdmagazine2$
```

Esta redirección a esa carpeta funcionará cada vez que se use `workon`. **Simplemente genial.**

Para profundizar en su uso, es muy recomendable revisar la documentación¹⁷, y dentro de ella darle la importancia a la referencia de comandos¹⁸ y a como extenderlo¹⁹.

Happy coding :)

17 <http://virtualenvwrapper.readthedocs.org/en/latest/index.html>

18 http://virtualenvwrapper.readthedocs.org/en/latest/command_ref.html

19 <http://virtualenvwrapper.readthedocs.org/en/latest/plugins.html>

Serán tres días completos de sesiones con expertos locales, regionales e internacionales donde tendrás la oportunidad de seguir aprendiendo de la comunidad y sumar tu aporte.

VI Jornadas Latinoamericanas de Metodologías Ágiles



10, 11 y 12 de Octubre en la Cámara de Comercio de Lima (Perú)

cobrar con
PayPal

comisión:
0%



transferir a la
**Cuenta Bancaria Virtual
en USA**

**Clic aquí
para registrarte**

costo de apertura:
registrándote
mediante esta
publicidad
y haciendo un
depósito inicial:
USD 100.-

extraer y
comprar con
Payoneer
en todo el mundo!



comisión:
1%



safecreative



1 307295 492695
INFO ABOUT RIGHTS

¡GRACIAS POR LEERNOS!