AÑO 0
NÚMERO 2
2012-12-31

#2

<Champagne>

Edición Especial
Con regalos y sorpresas

Despedimos el año con un
INVITADO DE LUJO:
Richard Stallman

HD Hackers & DEVELOPERS

Magazine digital de distribución mensual
sobre Software Libre, Hacking
y Programación

Staff

Celia Cintas	Licenciada en Informática
Eugenia Bahit	Arquitecta GLAMP & Agile Coach
Indira Burga	Ingeniera de Sistemas
Laura Mora	Adm. de Redes y Sistemas
María Jose Montes	Tec. en Informática de Gestión
Milagros Infante	Est. Ingeniería de Sistemas
Yecely Díaz	Maestra en Inteligencia Artificial

Responsable de Proyecto:
Eugenia Bahit

Responsable de Comunicación:
Indira Burga

Colaboradora invitada:
Rosa María Orellana

Descubre tus regalos

HD Hackers & DEVELOPERS #2

"... Hacker es alguien que disfruta jugando con la inteligencia.."

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

Acerca de

Hackers & Developers es un Magazine digital de distribución libre y gratuita, sobre Software Libre, hacking y programación.

Se distribuye mensualmente bajo una licencia Creative Commons.

Envía tu artículo

¿Quieres colaborar con HD Magazine?
pide el instructivo y la plantilla en
colabora@hdmagazine.org

U! Dilo en público

¿Quieres enviar un comentario para que se publique en la zona **U!**?
Escríbenos a:
lectores@hdmagazine.org

Contáctanos

¿Quieres enviarnos un comentario en Privado? Envíanos un e-mail a:
contacto@hdmagazine.org

Haz un donativo

¿Quieres apoyarnos económicamente?
Envíanos un e-mail a:
donaciones@hdmagazine.org

Merchandising

Visita nuestro **Web Store y apóyanos económicamente** adquiriendo merchandising de Hackers & Developers
http://store.hdmagazine.org

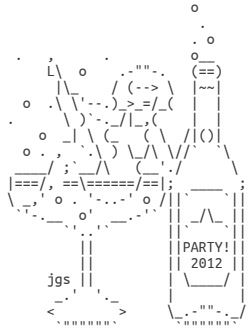
Despedimos el año junto a un invitado de lujo:

Richard Stallman

en exclusivo, para Hackers & Developers Magazine

Pág. 48

Artículos de este mes en Hackers & Developers...



ASCII ART
Este mes: «Woman in Champagne Glass»
By **Joan Stark** (http://www.ascii-art.com)

>> Pág. 94

Prevención de ataques por fuerza bruta y Man in the Middle

SEGURIDAD INFORMÁTICA

Los ataques de fuerza bruta así como los llamados «*Man In The Middle*», son dos de las violentas agresiones informáticas -tan temidas como frecuentes-, a las que todos aquellos que tenemos a cargo algún servidor, estamos expuestos pudiendo convertimos en posibles víctimas de las mismas. Tomar las medidas necesarias para prevenirlas, es la forma más segura de minimizar los riesgos a los cuáles nos enfrentamos.

Escrito por: **Eugenia Bahit** (Arquitecta GLAMP & Agile Coach)

Eugenia es **Arquitecta de Software**, **docente** 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** e integrante del equipo de **Debian Hackers**.

Webs:
Cursos de programación a Distancia: www.cursosdeprogramacionadistancia.com
Agile Coaching: www.eugeniabahit.com

Redes sociales:
Twitter / Identi.ca: **@eugeniabahit**

Los ataques de fuerza bruta, no son los únicos de los cuáles nuestro servidor o aplicación, pueden ser víctimas (pensándolo bien, si cuando me roban la billetera, la víctima no es la billetera sino yo, no debería minimizar la situación, pues cuando mi servidor o mi aplicación es atacada y/o agredida, la víctima también soy yo).

Pero volvamos a los tipos de ataques de los cuáles nosotros (seres humanos) podemos ser víctimas. Si bien existe una gran cantidad de tipos de ataques informáticos (de la misma forma que existen infinidad de tipificaciones para los delitos no informáticos), en

este momento nos ocupan dos de los más temidos: los ataques de fuerza bruta y los ataques llamados "Man in the Middle". Pero, actuando desde ahora, podremos intentar prevenir a fin de minimizar los riesgos.

Man in the Middle

Mientras que los ataques de fuerza bruta podrían compararse al delincuente que intenta por todos los medios posibles, abrir una caja fuerte para robar su contenido, los ataques de tipo *«Man in the Middle»* (hombre en el medio -o como me gusta llamarlos a mi, *"mad" in the middle*), representan un modo de agresión mucho más "psicópata" que la anterior. En este caso, el agresor actúa como un psicópata que observa los movimientos de su víctima. Se posiciona en medio de nuestro ordenador y el servidor, "leyendo nuestra correspondencia" antes de enviarla. Podemos comparar un ataque *«Man in the Middle»* a la siguiente situación:

> *Cada vez que escribimos una carta, la depositamos en el buzón de correo, el psicópata nos la intercepta, la abre, la lee, le hace algunos cambios y la entrega él mismo al destinatario original. Luego, hace lo propio con la carta que nos retorne la otra persona.*

En esto mismo consiste el ataque *«Man in the Middle»*. Cuando intentamos conectar al servidor mediante el protocolo SSH, el agresor intercepta nuestros paquetes de datos, los acapara para sí y es él mismo, quien actúa como intermediario para enviar dicha información al verdadero servidor. Cuando el servidor envía una respuesta, el agresor vuelve a interceptarla y nos la entrega modificada, haciéndonos creer que ha sido el servidor quien nos la ha enviado, convirtiéndonos en rehenes de su ordenador.

Medidas preventivas

Contexto: En los ataques de este tipo, cuando el agresor intercepta nuestros "mensajes" al servidor, su ordenador (otro servidor) actúa de forma engañosa, haciéndose pasar por el verdadero servidor.

Cuando te conectas desde tu ordenador a un servidor por primera vez utilizando el protocolo SSH, el servidor te envía una huella digital única (*fingerprint*) que debes aceptar para poder *loguearte* o rechazar, negando así la posibilidad de acceso. Esta huella digital, el servidor la almacena en una clave pública y al ser aceptada por tu ordenador (cliente SSH), se guarda en el archivo know_hosts (hosts conocidos).

Las siguientes veces que te conectes, cuando el servidor envíe su huella digital, el cliente SSH buscará dicha clave en el archivo know_hosts que se encuentra en el directorio `.ssh` de tu home. Al encontrarla, verificará que ambas claves coincidan evitando así, volver a

pedirte su aceptación o rechazo.

El problema: La única posibilidad de que la huella digital del servidor cambie, es que se genere una nueva clave -de forma *ex profesa*- en el servidor, ya sea porque se efectuó una reinstalación completa o porque se eliminó la clave actual y se creó una nueva. Es decir, que si ya te has conectado al servidor anteriormente y aceptado su huella digital, si te vuelves a conectar y te pide aceptar una nueva clave sin que la huella original se haya modificado, deberás rechazarla.

Tomar las precauciones necesarias para evitar el problema: Para estar seguro de cuál es la verdadera huella digital de tu servidor, deberás verificarla en el directorio `/etc/ssh/` del servidor:

```
$ cd /etc/ssh
$ ls -lh
total 164K
-rw-r--r-- 1 root root 123K Apr  2  2012 moduli
-rw-r--r-- 1 root root 1.7K Apr  2  2012 ssh_config
-rw-r--r-- 1 root root 2.6K Dec 10 19:47 sshd_config
-rw------- 1 root root  668 Dec  8 16:44 ssh_host_dsa_key
-rw-r--r-- 1 root root  604 Dec  8 16:44 ssh_host_dsa_key.pub
-rw------- 1 root root  227 Dec  8 16:44 ssh_host_ecdsa_key
-rw-r--r-- 1 root root  176 Dec  8 16:44 ssh_host_ecdsa_key.pub
-rw------- 1 root root 1.7K Dec  8 16:44 ssh_host_rsa_key
-rw-r--r-- 1 root root  396 Dec  8 16:44 ssh_host_rsa_key.pub
-rw-r--r-- 1 root root  302 Jan 10  2011 ssh_import_id
```

Los archivos `.pub` son las llaves públicas generadas por SSH para tu servidor.

Los archivos `ssh_host_`*`formatoclave`*`_key.pub` son aquellos que contienen las huellas digitales públicas que nos interesan.

Para saber cuál de los archivos `.pub` debes mirar, tendrás que verificar el formato de clave especificado en la huella digital que el servidor te está ofreciendo (se resalta el formato en negritas):

```
$ ssh usuariocomun@123.456.78.90
The authenticity of host '123.456.78.90 (123.456.78.90)' can't be established.
ECDSA key fingerprint is 91:bf:9d:7b:b9:9f:15:54:07:fb:d4:23:c6:fa:fd:b0.
Are you sure you want to continue connecting (yes/no)?
```

En el caso anterior deberás mirar la clave de tu archivo `ssh_host_`**`ecdsa`**`_key.pub` . Para visualizarla en el mismo formato -hexadecimal legible- en el que te es ofrecida por el servidor al intentar conectarte, puedes utilizar el comando `ssh-keygen`:

```
$ ssh-keygen -lf ssh_host_ecdsa_key.pub
```

El mismo, arrojará una salida similar a la sguiente:

```
2048 91:bf:9d:7b:b9:9f:15:54:07:fb:d4:23:c6:fa:fd:b0  root@localhost (ECDSA)
```

> *Antes de aceptar la clave, verifica que coincida con la que se te está ofreciendo aceptar o rechazar.*

Para verificar que ambas claves son idénticas, puedes escribirlas en dos archivos y utilizar el comando `diff` para que sean evaluadas con precisión. Si éste no arroja nada, significará que no existen diferencias entre ambas claves (es decir, ambas claves son idénticas). Ergo, puedes aceptar la huella con absoluta confianza:

```
$ echo "91:bf:9d:7b:b9:9f:15:54:07:fb:d4:23:c6:fa:fd:b0" > a
$ echo "91:bf:9d:7b:b9:9f:15:54:07:fb:d4:23:c6:fa:fd:b0" > b; diff a b
```

Claro que también podrías compararlas manualmente y confiar en tu buen ojo =)

Ataques de fuerza bruta

Un ataque de fuerza bruta, es aquel mediante el cual, se intenta descubrir una contraseña de forma agresiva, ya sea de modo sencillo (probando diferentes combinaciones de caracteres) o mediante el uso de un diccionario (probando combinaciones de nombres de usuarios y contraseñas de uso frecuente).

> *Utilizar contraseñas que incluyan letras en minúsculas, mayúsculas, números y otros caracteres con una longitud mayor a 16 caracteres, **ayuda a minimizar los riesgos** frente a intentos de robo por ataques de fuerza bruta.*

Medidas preventivas para conexiones por SSH

Lo primero que debemos hacer, es configurar ciertas medidas de seguridad en el demonio de SSH (en el servidor). El archivo de configuración se encuentra en

```
/etc/ssh/sshd_config
```

Las **medidas que nos ayudarán a prevenir ataques de fuerza bruta**, son las siguientes:

1) No permitir el login remoto del usuario root

```
PermitRootLogin no
```

Esta medida evitará que en caso de una agresión con resultado favorable para el atacante, puedan acceder al servidor con permisos absolutos (es decir, como súper usuario).

Está muy claro que necesitamos contar con un usuario que cuente con dichos permisos. Solo bastará con acceder mediante un usuario común y *loguearnos* como *root* una vez dentro del servidor.

La alternativa, es **crear un usuario común y agregarlo al grupo de administradores**:

```
# adduser usuariocomun
# usermod -a -G sudo usuariocomun
```

2) Limitar la cantidad de intentos de logueo fallidos

```
MaxAuthTries 3
```

En este caso, indicamos que el máximo número de veces que podemos equivocarnos al intentar *loguearnos*, es de tres. **Esta directiva no bloqueará al usuario ni su IP**. Simplemente le cerrará la conexión, haciendo más "tedioso" el trabajo del agresor.

Esto ayuda a prevenir los ataques de fuerza bruta que utilizan diccionarios, ya que el proceso de agresión se hará más lento al tener que reiniciar la conexión en cada oportunidad que el servidor desconecte al atacante.

3) Limitar la cantidad de conexiones simultáneas

```
MaxStartups 2
```

Mediante esta directiva, estamos indicando que no puede haber más de dos conexiones simultáneas desde una misma IP. Esto es muy útil -al igual que el caso anterior-, para ralentizar el trabajo del agresor, puesto que la cantidad de intentos simultáneos se verán limitados, generando una una menor tasa de acierto.

4) Limitar la cantidad máxima de tiempo durante la cual se mostrará la pantalla de logueo

```
LoginGraceTime 45
```

45 segundos, es tiempo más que suficiente para introducir tu contraseña. Al igual que en los dos casos anteriores, esta medida hará que el trabajo del agresor se haga cada vez más lento y tedioso. Menos tiempo para los intentos, generará una tasa de acierto cada vez más baja.

5) Limitar el acceso SSH a un usuario determinado

```
AllowUsers usuariocomun
```

Si solo habilitamos el acceso por SSH a un usuario, los intentos del agresor por ingresar al servidor, serán prácticamente imposibles, siempre y cuándo éste, desconozca por

completo el nombre del usuario que cuenta con dicho permiso.

Si se desea habilitar dicho acceso a más de un usuario, se colocarán sus nombres en la misma línea, separados por un espacio en blanco:

```
AllowUsers usuariocomun usuario2 usuario3
```

Si se cuenta con una IP fija en el cliente, se puede restringir aún más el acceso de este usuario, solo desde la IP del mismo. En este caso, a no ser que el agresor logre acceder al ordenador del usuario (ya sea de forma virtual o física), **será imposible que logre acceder por SSH**:

```
AllowUsers usuariocomun@123.456.78.90
```

Se puede necesitar ser un poco menos radical con las autorizaciones y, limitar el acceso a un grupo determinado de usuarios:

```
AllowGroup grupohabilitado
```

6) Cambiar el puerto por defecto

```
Port 372
```

Una medida menos efectiva pero algo astuta, es modificar el puerto por el cual se accede mediante el protocolo SSH. Por defecto, este puerto es el 22.

Una gran cantidad de "armas" (perdón, debí decir "herramientas"), se dedican a intentar realizar los ataques de forma predeterminada (y automatizada) directamente a través de este puerto. Cambiar el número de puerto, no garantiza absolutamente nada. Simplemente retrasará unos pocos minutos el ataque, dado que existen formas -también agresivas- de escanear los puertos remotamente y ver cuál es el que está a la escucha de SSH.

Vale aclarar que el puerto, debería ser inferior a 1024.

7) La alternativa más segura: no permitir el acceso mediante contraseña

A esta altura, no debes estar entendiendo nada y hasta me debes haber catalogado de demente irreversible. Y créeme: lo entiendo. Pero déjame que te explique de que se trata.

Existe una alternativa para iniciar sesión por SSH, mediante la cual, en vez de utilizar una contraseña, el servidor verifica tu identidad de la misma forma que tu verificas la suya: a través de una huella digital única.

Esta técnica, consiste en generar una clave RSA en el ordenador desde el cual se le permitirá el acceso a un determinado usuario y luego, enviar una copia de la clave pública generada al servidor.

De todas las medidas, ésta es quizás, la menos vulnerable de todas.

Para implementarla, el primer paso consistirá en generar una clave RSA para tu usuario, en tu ordenador local. Para ello, deberás ejecutar el comando `ssh-keygen` y seguir los pasos conforme vayan apareciendo en pantalla:

```
$ ssh-keygen
```

Una vez generada la clave, **en el directorio `.ssh` de tu home**, encontrarás dos archivos:

```
id_rsa       esta es tu clave privada. No debes compartirla con nadie
id_rsa.pub   es tu clave pública y solo ésta, deberás enviar a tu servidor
```

Enviarás dicha clave al servidor, mediante el comando `scp`, reemplazando -obviamente- tu usuario e IP:

```
scp ~/.ssh/id_rsa.pub usuariocomun@123.456.78.90:
```

Una vez enviada al servidor, ingresas en éste y realizas los siguientes cambios:

Creas un directorio (dentro de la home de tu usuario en el servidor) para almacenar la clave que has enviado anteriormente:

```
mkdir .ssh
```

Mueves la clave pública dentro del directorio creado:

```
mv id_rsa.pub .ssh/authorized_keys
```

Modificas los permisos del directorio y del archivo, de a uno a la vez:

```
chown -R usuariocomun:usuariocomun .ssh
chmod 700 .ssh
chmod 600 .ssh/authorized_keys
```

Y finalmente, desactivas el acceso SSH mediante contraseña, desde el archivo `/etc/ssh/sshd_confg`:

```
PasswordAuthentication no
```

> *Recuerda que cada vez que realices cambios en el archivo* `/etc/ssh/sshd_config` *deberás reiniciar el servicio SSH:* `service ssh restart`

8) Bloquear las IP tras varios intentos fallidos

Puedes instalar una herramienta que se encargue de detectar los intentos de acceso fallidos (verificando los *logs* de autenticación) y automáticamente, cree reglas que

bloqueen temporalmente, las IP que hayan realizado dichos intentos. Una de estas herramientas, es **fail2ban**. Puedes encontrarla en los repositorios oficiales de tu distribución GNU/Linux con suma facilidad o descargarla ingresando en:

http://www.fail2ban.org/wiki/index.php/Downloads

Una excelente guía en español sobre **cómo configurar correctamente fail2ban**, puede encontrarse en el sitio Web oficial, de la mano de Manuel Aróstegui Ramírez, ingresando en: http://www.fail2ban.org/wiki/index.php/**HOWTO_fail2ban_spanish**

Medidas preventivas en aplicaciones Web

Es posible que en tus aplicaciones Web, se realicen ataques de fuerza bruta. Las medidas de seguridad a implementar, dependen solo y exclusivamente de cada aplicación en particular. Sin embargo, tener en cuenta las siguientes pautas, ayudará a un desarrollo mucho más seguro:

- **Utiliza una política de contraseñas estricta:** obliga a tus usuarios a utilizar contraseñas de no menos de 12 caracteres, que incorporen tanto números, como letras en mayúsculas y minúsculas y al menos 1 o 2 caracteres que no sean alfanuméricos;
- **Limita la cantidad de intentos de acceso:** procura permitir un máximo de 4 intentos. Genera *logs* de aplicación propios, con cada uno de los intentos de acceso a la aplicación. Estos *logs* te servirán para que tu aplicación pueda analizar de forma rápida y confiable, aquellos usuarios e IP que hayan fallado varias veces sucesivas en su intento de acceso;
- **Bloquea los usuarios que hayan fallado al menos 4 veces seguidas en su intento de acceso:** procura avisar previamente que serán bloqueados e inhabilítales el reingreso por al menos 60 minutos;
- **Bloquea aquellas IP desde las cuáles, se haya intentando acceder de forma errónea:** ten presente que desde una misma IP, el atacante podría intentar probar con diferentes combinaciones de usuario y contraseña. Es muy importante que verifiques esto en los *logs* de tu aplicación y bloquees por intentos fallidos por IP y no solo por usuario. En estos casos, no será al usuario al que debas inhabilitar, sino a la IP;
- **Minimiza la posibilidad de que los agresores lleguen a tu aplicación a través de los buscadores:** procura no utilizar nombres comunes para los sistemas de administración. Evita URIs que contengan palabras como *admin*, *panel*, *login* y otros términos que puedan asociarse fácilmente, a formularios de *logueo*.

Tip by Commander in Chief:

Una forma de utilizar contraseñas fáciles de recordar y complejas de descubrir, es *hashear* el nombre de tu canción favorita, grupo de música o deportista preferido y utilizar ese *hash* como *password*. Si tienes **PHP-CLI** instalado, desde la terminal ejecutas:

```
php -r 'print md5("creedence clearwater revival");'
```

o con **Python**:

```
python -c "import hashlib; print hashlib.md5('creedence
clearwater revival').hexdigest()"
```

Backups: ¡Siempre me acuerdo de ellos cuando no están!

SEGURIDAD INFORMÁTICA

Todos sabemos lo importantes que son los backups. En nuestras empresas tenemos grandes suites, pero ¿y para nuestros datos personales?

Escrito por: **Laura Mora** (Administradora de Redes y Sistemas GNU/Linux)

Laura es administradora de **Redes y Sistemas GNU/Linux** en Cataluña. Tiene una larga trayectoria en la consultoría de soluciones telemáticas **opensource** para movimientos sociales. La base de sus proyectos es el **empoderamiento tecnológico** de las personas.

Webs:
Blog: http://blackhold.nusepas.com/
Web: http://delanit.net

Redes sociales:
Twitter / Identi.ca: **@Blackhold_**

Este artículo quiero dedicárselo a una de las redactoras de esta revista que tras un problema de hardware tuvo una pérdida de datos de varios meses, en éste artículo pues vamos a ver algunas ideas para mantener nuestros archivos a salvo o recuperarlos en caso de desastre.

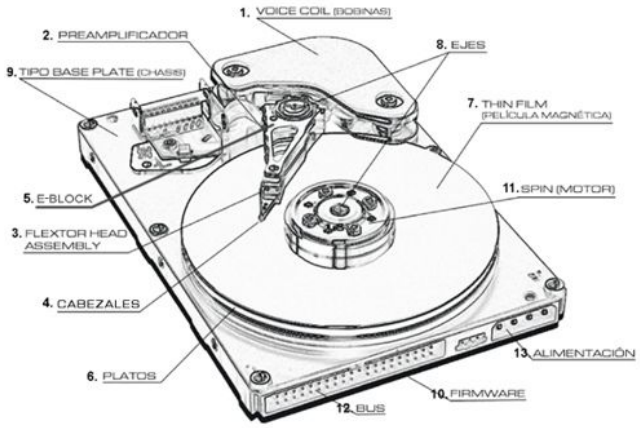
Muchos de nosotros nos hemos encontrado con el robo del ordenador, una mala caída o simplemente una muerte súbita del disco. Ante estos casos simplemente la mejor solución al problema es tener un **backup** lo más reciente posible. En el caso de un fallo físico del disco la broma puede salirte realmente cara si optas por empresas de recuperación de discos.

Quizás podemos ser unos maniáticos de las copias de seguridad y tener nuestros datos multiplicados varias veces en decenas de discos duros, los cuales vamos dejando olvidados bajo montones de papeles, desperdigados dentro de un armario, o peor aún, ¡backups dentro de backups! La caída de alguno de estos pequeños discos de 2,5" puede ser fatal por algunos de los datos que guardas ahí y sobretodo si no están en ningún otro lugar.

Así que vamos a ver a distintos niveles como recuperar nuestros datos y sobretodo algunas recomendaciones para no olvidarse de una cosa tan importante como los backups.

Recuperación de un disco "muerto"

Los discos duros pueden morir de dos formas: muerte mecánica o muerte electrónica.



Una **muerte mecánica** puede ser ocasionada por varios motivos. La más común, una caída. Esto puede hacer que el cabezal se salga de su eje y en este caso lo peor que se puede hacer es poner en funcionamiento el disco, ya que esto rallaría los platos e implicaría la pérdida de gran parte de la información. También nos podemos encontrar con que alguna de las partes mecánicas se haya bloqueado o dejado de funcionar (normalmente oiremos un cleck! cleck! cleck!).

Por software, identificaremos que estamos ante un fallo mecánico porque nuestro sistema detectará el disco pero no podrá acceder a él.

En estos casos si realmente la información que contiene el disco es realmente valiosa, se debe contemplar la opción de llevar el disco duro a una empresa de recuperación de datos, las cuales tienen las llamadas salas blancas: espacios totalmente liberados de polvo y otras partículas que al abrir el disco podrían dañarlo.

Esta operación no baja de los 800-900€

Una **muerte electrónica**, significa que la electrónica que hace funcionar el disco ha dejado de funcionar, comúnmente por una subida de tensión o por la más que odiosa y molesta *obsolescencia programada*.

En estos casos siempre nos queda la opción de buscar un disco duro, misma marca y modelo que el disco duro estropeado y con una llave Torx[1], remplazar cuidadosamente la electrónica.

Esta operación tiene la complejidad de encontrar un disco duro idéntico, pero hay tiendas on-line que venden electrónicas para los modelos de discos duros más comunes. Su coste puede ir de los 20 a los 60€ aproximadamente.

1 http://es.wikipedia.org/wiki/Torx

Una buena prevención salva datos!

Normalmente cuando un disco duro va a fallar, antes nos avisa. Pero debemos tener y usar adecuadamente algunas herramientas para identificar estos avisos. En GNU/Linux tenemos dos herramientas básicas:

SMARTMONTOOLS: Con ésta, podremos verificar el estado de nuestro disco duro. Normalmente un disco antes de morir suele avisarnos mediante errores I/O.

Smartmontools suele encontrarse en los repositorios de la gran mayoría de distribuciones GNU/Linux. Si no es así, siempre tenemos la opción de descargarlo desde su página web[2].

Una vez lo tenemos instalado, modificamos el fichero `/etc/default/smartmontools` y verificamos que existan estas dos líneas:

```
enable_smart="/dev/sda /dev/sdb"
start_smartd=yes
```

En la primera debemos especificar los discos físicos que queremos auditar con esta herramienta. Smartmontools está como servicio, así que si lo modificamos deberemos (re)iniciarlo:

```
service smartmontools start
```

Para verificar que funciona correctamente usaremos el comando `smartctl -i` *`disco`*

```
# smartctl -i /dev/sda
[...]
Device supports SMART and is Enabled
Temperature Warning Enabled
```

Para proceder al test de un disco duro de los listados anteriormente, ejecutamos lo siguiente como `root` (sustituid "`sdb1`" por vuestro disco duro si procede):

```
smartctl -d scsi -H /dev/sda
```

Si todo va bien, en ese disco duro deberíais obtener un mensaje parecido a éste:

```
=== START OF READ SMART DATA SECTION ===
SMART overall-health self-assessment test result: PASSED
```

Si algo va mal, aparecerá un aviso "FAILING_NOW". Para obtener más info al respecto, de nuevo como `root` ejecutaremos lo siguiente:

```
smartctl --attributes --log=selftest /dev/sda
```

2 http://sourceforge.net/apps/trac/smartmontools/wiki

Si queremos que el sistema nos mande regularmente más información sobre el estado de los discos, lo recomendable es crear un *script* y ejecutarlo con `cron`:

```
[/etc/crontab]
#report smartmontools
0 7     * * 1   root    /root/scripts/report_smartmontools.sh


[/root/scripts/report_smartmontools.sh]
#!/bin/bash

MAIL="usuario@dominio.net"

smartctl -a /dev/sda |mail -s "[SMART: SDA `hostname`]" $MAIL
smartctl -a /dev/sdb |mail -s "[SMART: SDB `hostname`]" $MAIL
```

FSCK: Esta otra herramienta nos puede servir para verificar discos o incluso "salvar" a aquellos que tengan sectores defectuosos, para poder extraer los datos ¡antes de que sea demasiado tarde!

El funcionamiento es como la gran mayoría de las herramientas en sistemas GNU/Linux:

```
fsck [-opciones] /dev/sda
      -a confirmar automáticamente. No recomendado.
      -c comprobar bloques en el disco.
      -f forzar el chequeo aunque todo parezca ok.
      -v (verbose) despliega más información.
      -r Modo interactivo. Espera nuestra respuesta.
      -y asume yes de respuesta.
```

Borrado accidental de datos

También nos podemos encontrar con la posibilidad de que alguien (o simplemente uno mismo), haya decidido instalar un sistema operativo encima de un disco duro con datos preciados, o ¡que le haya dado demasiado a la tecla suprimir! Pero *don't panic*!

Cuando borramos una parte de nuestro disco, en realidad lo que estamos haciendo es marcar aquellos sectores del disco como vacíos, pero ¡los datos siguen ahí! así que ante todo y lo más importante, es que en este caso, si te has percatado que la has pifiado, lo mejor que puedes hacer es dejar de añadir o quitar datos de aquel disco e ir a herramientas de recuperación de datos borrados.

Cuando queremos hacerlo para particiones basadas en sistemas GNU/Linux tenemos algunas herramientas como autopsy[3].

Lo primero que deberemos hacer es generar una imagen que la analizaremos con autopsy:

```
dd if=/dev/sda1 of=/media/hd/archivo_con_disco_a_recuperar.dd
```

3 http://www.sleuthkit.org/autopsy/

Descargaremos[4] e instalaremos el programa (también lo encontraremos en el repositorio de las distribuciones de GNU/Linux más comunes) y, accederemos a la herramienta vía web:

```
http://localhost:9999/autopsy
```

En la interfaz creamos un nuevo caso con cualquier nombre; seleccionaremos -en "add image"- el fichero que hemos creado con la imagen de nuestra partición, e incorporaremos el fichero de imagen al proyecto.

Seleccionamos "Analizar" y luego en "File Analysis", veremos ya los ficheros que se borraron, podremos visualizarlos e incluso si no se han machacado, recuperarlos.

Hay varios programas para recuperar ficheros en GNU/Linux, para conocer algunos de ellos os recomiendo leer este artículo[5].

Programas para hacer Backups

Tras ver pues, varias formas de recuperar datos a varios niveles, lo más recomendable es realizar copias de seguridad y sobretodo si el soporte es un ordenador o un disco duro externo el cual no desplazamos en absoluto, mucho mejor, ya que de esta forma reducimos los riegos para un fallo mecánico.

Si tenemos muchos datos, quizás un `cp` no nos servirá así que lo recomendable será usar herramientas destinadas a realizar copias de seguridad.

De copias de seguridad podemos tener tres tipos:

Completas: El contenido del backup es exactamente el mismo que el contenido de los directorios donde tenemos nuestros datos.

Diferenciales: El contenido del backup contiene los archivos modificados a partir de una copia completa, cuando mas vieja sea dicha base, más grande será esta copia diferencial.

Incrementales: Es similar a la diferencial pero contiene todas las versiones de los ficheros modificados desde la última base.

Si queremos hacer copias de seguridad para entornos profesionales, se recomienda el uso de copias diferenciales o incrementales. Escogeremos una u otra dependiendo de la infraestructura que tengamos y/o la exigencia de nuestro cliente/jefe.

Para ello podremos usar algunas herramientas de backup como bacula[6], backup pc[7] o grsync[8] entre otros.

4 http://www.sleuthkit.org/autopsy/
5 http://www.bairesnortelug.com.ar/2007/01/31/recuperacion-de-datos-perdi-todo-y-ahora-que-hago/
6 http://www.bacula.org/
7 http://backuppc.sourceforge.net/
8 http://www.opbyte.it/grsync/

Mi recomendación pero, para archivos personales es la realización de copias completas. Algunos tenemos Diógenes tecnológico y tendemos a guardar muchos datos, sólo los borramos cuando necesitamos espacio en el disco ;)

Para ello tenemos 2 opciones: o usar `copy`, que nos podemos morir en el intento si estamos hablando de muchos ficheros o, crear un script de rsync que lo podemos ejecutar manualmente o añadirlo en cron.

Lo primero será instalar rsync en la maquina a la cual queremos hacer backup de los datos y la máquina que va a guardar nuestros datos con un disco mas grande que el de la maquina a hacer el backup.

Una vez hecho esto creamos un fichero con este contenido (éste es el ejemplo de mi "superscript" de backup!):

```
#!/bin/bash
rsync --delete -av /home/laura/ laura@ipserverbackups:/home/laura/
```

En algún momento, queremos excluir algún directorio. Añadiremos la opción `--exclude` en la zona de opciones del comando, por ejemplo:

```
rsync --delete --exclude .VirtualBox -av /home/laura/ laura@ip:/home/laura/
```

Si nos fijamos, la transferencia de datos la haremos sobre `scp` (ssh).

A veces la solución mas sencilla es la mas efectiva :)

Ahora es acordarse y tomar consciencia de la importancia de tener una buena política de Backups. Una de ellas es asociar a hacer backups a tareas domésticas, como no salir de casa sin las llaves, o si te vas varios días de asegurarte de que todas las ventanas estén cerradas. ¿Tienes tus backup al día?

escuela IT

Formamos personas,
Certificamos profesionales.

En cursos que inician en Enero '13

50 %
de descuento

El descuento aplica en cualquiera de los siguientes cursos, en pagos efectuados hasta 24 horas antes de la fecha de inicio indicada:

- 10 enero: **Diseño Responsive Web Design**
- 17 enero: **PHP Básico**
- 24 enero: **Desarrollo de aplicaciones Android**

Para más información o **canjear este cupón**, comunícate con **escuela.IT** ingresando en www.escuela.it/contacto. CÓDIGO DE DESCUENTO: **EIT-2013-HDMAG**

NO ACUMULABLE CON OTRAS OFERTAS Y/O PROMOCIONES. **VÁLIDO** DESDE EL 31/12/2012 **HASTA 24 HORAS ANTES DE LA FECHA DE INICIO DEL CURSO ELEGIDO.**

Gentileza de Hackers & Developers Magazine: www.hdmagazine.org

Arduino: Filosofía OpenSource

HARDWARE

En esta entrega vamos a adentrarnos en el maravilloso mundo de Arduino. Hablaremos de su hardware y software, con el que podrás hacer tus propios diseños.

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://archninfa.blogspot.com.es/

Redes sociales:
Twitter: **@MMontesDiaz**

Nació en el Instituto de Diseño Interactivo Ivrea (Italia), a partir de un proyecto en el año 2005. La idea surgió por la necesidad de tener una herramienta para los estudiantes que fuera más moderna que las que estaban en el mercado y de bajo coste. Además, que funcionase con cualquier sistema operativo y que contase con documentación adaptada a gente que quiera empezar de cero.

Los inicios de Arduino contados por **Massimo Banzi***:*

> *"Cuando estaba trabajando en esto conocí a David Cuartelles y comenzó a echarme una mano con el proyecto…Hicimos juntos el primer hardware de Arduino, luego vino David Mellis, un estudiante mío, que se unió para escribir el software, luego Tom Igde entró como consejero y Gianluca Martino que era el que producía las placas. Así se formó el equipo, añadiendo gente según sus habilidades".* ***Massimo Banzi***

Arduino se implementó, no obstante, sobre los cimientos de Wiring. Verán, en Ivrea también daba clases Casey Reas, uno de los fundadores de la plataforma de programación Processing. Banzi pensó en cómo hacer un Processing para hardware. Comenzó, entonces, a trabajar con un estudiante suyo, que había hecho una tesis sobre el tema, Hernando Barragán.

> *"Después de que Hernando hiciera Wiring pensamos en cómo hacer toda la plataforma más simple, más barata y sencilla de usar. Se comenzó a reimplementar todo como un proyecto open source para que todo el mundo pudiera venir y ayudar, contribuir".* ***Massimo Banzi***

En la actualidad se pueden fabricar infinidad de prototipos y su uso se está expandiendo por todo el mundo.

> *"El hardware abierto significa tener la posibilidad de mirar lo que hay dentro de las cosas, que eso sea éticamente correcto, y que permita mejorar la educación. Educar en cómo funcionan las cosas…El hardware, aunque sea libre, no puede ser gratuito, es físico y cuesta dinero, lo que hicimos fue buscar el precio justo. Arduino no fabrica nada, diseña y mantiene un sitio web."* ***Massimo Banzi***

HARDWARE

Es una placa de circuito impreso donde va instalado el microcontrolador, las conexiones de entrada y salida, la conexión para el puerto USB y el botón reset.

Es decir, un pequeño circuito que contiene un ordenador completo en un chip (microcontrolador).

Microcontrolador.- realiza las instrucciones almacenadas en el programa de forma cíclica.

Algunos microcontroladores usados en las placas son el Atmega328, Atmega168, Atmeg1280, Armega168V, ATmega328p por su sencillez y bajo coste.

Puerto USB.- a través de él cargamos el programa (sketch) que hemos realizado en el entorno de programación de Arduino (IDE).

Pines de entrada/salida digital.- Estos pines pueden ser de entrada o salida, lo especificaremos mediante el sketch creado en el IDE.

Pines de entrada analógica.- Estos pines

aceptan valores analógicos y los convierten en un número comprendido entre 0 y 1023.

Pines de salida analógica.- Realmente son pines digitales que se han reprogramado para salida analógica usando el sketch en el IDE.

Botón reset.- Permite resetear el programa y cargar uno nuevo.

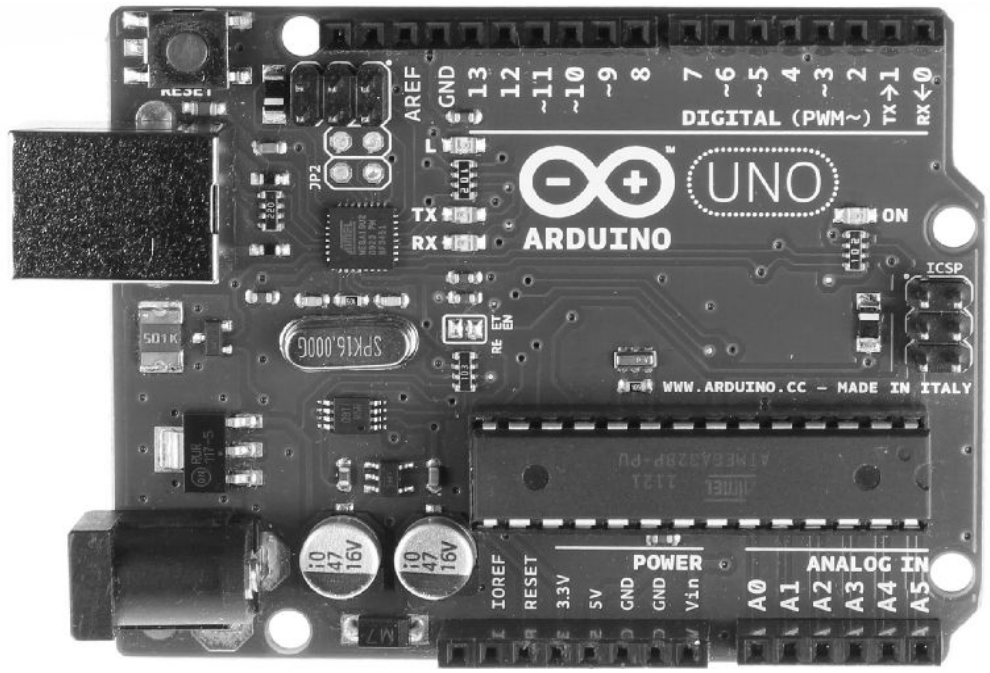
Tenemos la posibilidad de adquirir nuestra placa Arduino o hacerla a mano. En esta dirección encontramos lo necesario:

http://arduino.cc/es/Main/ArduinoBoardSerialSingleSided3

> *"Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos. "*

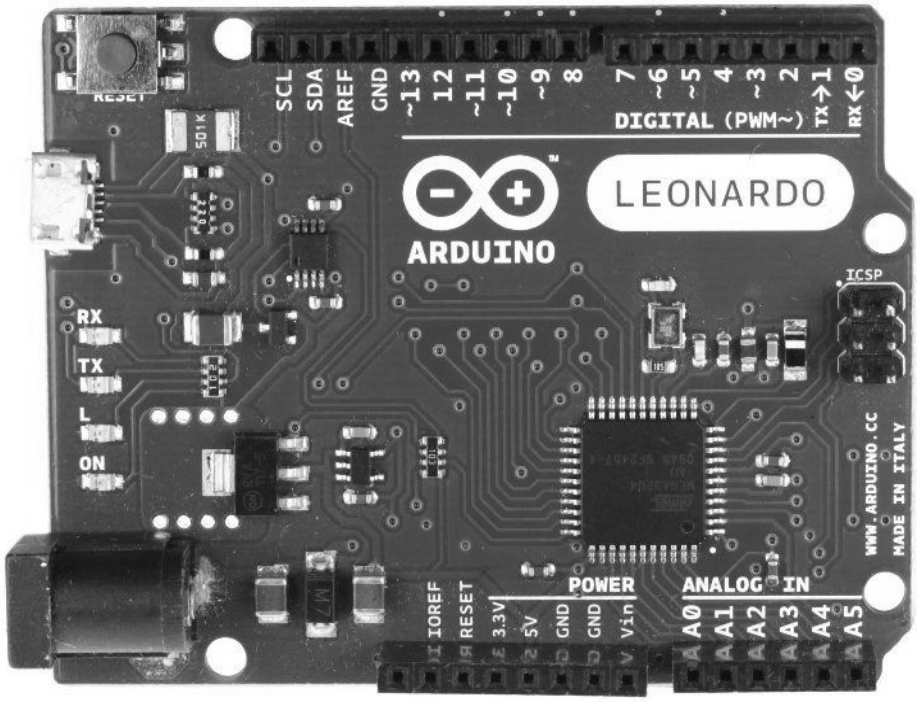
Arduino Uno

Arduino Uno es una placa electrónica basada en el ATmega328. Cuenta con 14 pines de entrada/salida digitales (de las cuales 6 se pueden utilizar como salidas PWM), 6 entradas analógicas, un resonador cerámico 16 MHz, una conexión USB, , entrada de corriente, conector ICSP y botón de reset. Contiene todo lo necesario para hacer funcionar el microcontrolador; simplemente conectándolo al ordenador con el cable USB o aliméntalo con un trasformador o batería para empezar.



Arduino Leonardo

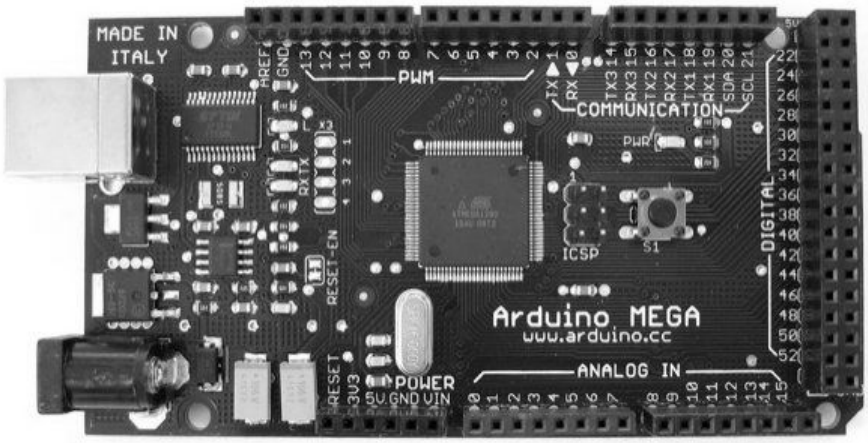
El *Leonardo Arduino* es una placa electrónica basada en el ATmega32u4 . Tiene 20 pines de entrada/salida digitales (de los cuales 7 se pueden utilizar como salidas de PWM y 12 como entradas analógicas), un oscilador de cristal de 16 MHz, una conexión micro USB, entrada de corriente, conector ICSP y botón de reset. Destaca por la incorporación de un único procesador encargado tanto de la conversión de Serie a USB, cómo de las funciones principales de procesamiento y de una nueva distribución de los pines. Esta última característica se aplicará también a los modelos anteriores. Entre las nuevas incorporaciones, destaca también la llegada del nuevo pin, IOREF, con el que podremos ajustar el voltaje del procesador y la compatibilidad con teclado y ratón.



Arduino Mega

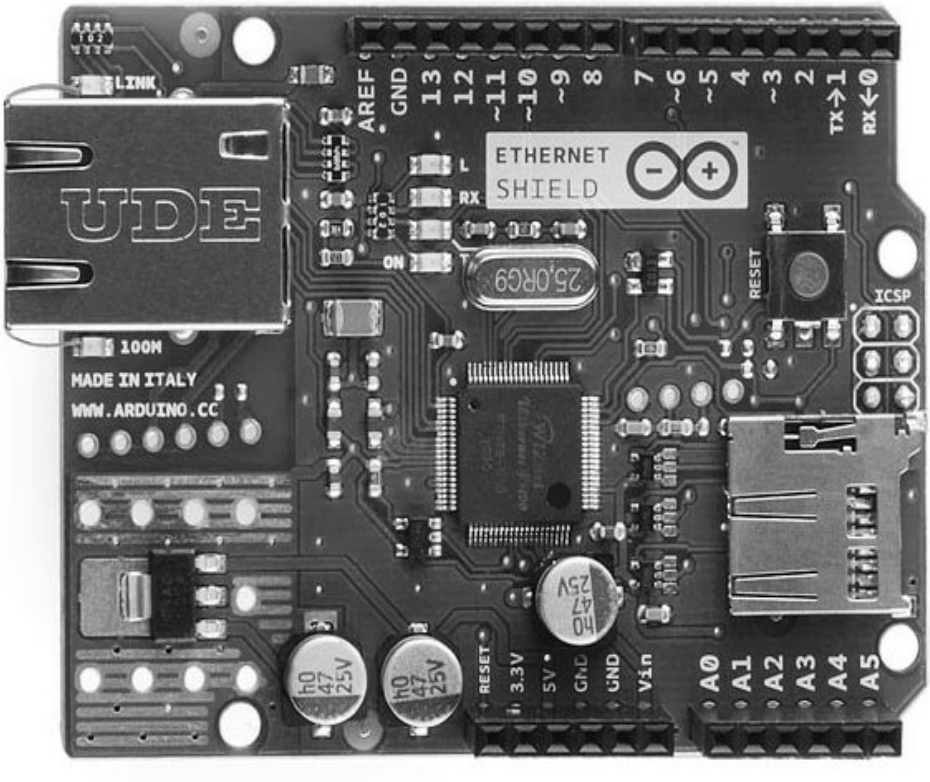
El *Arduino Mega* está basado ATmeg1280. Tiene 54 entradas/salidas digitales (de las cuales 14 proporcionan salida PWM), 16 entradas digitales, 4 UARTS (puertos serie por hardware), un cristal oscilador de 16MHz, conexión USB, entrada de corriente, conector ICSP y botón de reset. El Mega es compatible con la mayoría de *shields*[9] diseñados para el Arduino Duemilanove o Diecimila.

9 **Shields:** placas que se colocan encima de la placa Arduino y añaden una nueva función, para controlar diferentes aparatos, adquirir datos, etc.



Arduino Ethernet Shield

El *Arduino Ethernet Shield* permite -a una placa Arduino- conectarse a Internet. Se basa en el chip Wiznet W5100 Ethernet. Nos proporciona una red IP, con soporte para cuatro conexiones socket simultáneas.



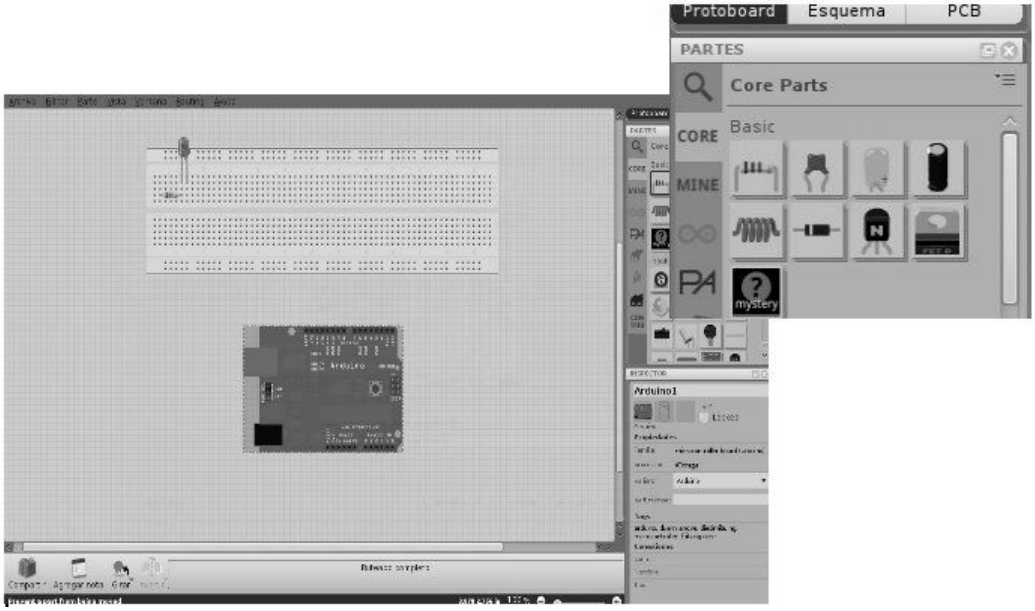
Fritzing nos ofrece un entorno gráfico para proyectos Arduino. Podemos usarlo para diseñar trazados de placa, esquemas y tarjetas de circuito impresas (PCB).

Para instalarlo en Arch:

```
$ yaourt -S fritzing
```

En distribuciones basadas en Debian:

```
# apt-get install fritzing
```



En el panel lateral tenemos las tres vistas principales organizadas en pestañas. Esta aplicación incluye varios proyectos de ejemplo que pueden ser útiles para familiarizarse con la misma.

Protoboard o Placa de prototipos: En ella se mostrará la visión real del proyecto. Podremos realizar las conexiones de manera muy sencilla, colocando los componentes y uniéndolos entre sí. Permite añadir notas para una mejor compresión del diseño.

La Vista de Esquema: Presenta una forma más abstracta de ver los componentes y las conexiones. Esta vista nos permite comprobar las conexiones realizadas anteriormente y es muy útil para crear la documentación del proyecto.

Vista de PCB o placa del circuito impreso: Desde esta vista iremos diseñando la forma en la que se acoplarán los componentes dentro de la placa de circuito impreso. Se pude cambiar el tamaño y la complejidad de las conexiones, según sea necesario. Cuando están colocados los componentes, se puede utilizar el botón `autorregulo` para que se generen las líneas de cobre.

SOFTWARE: el lenguaje y entorno de programación

La programación se realiza en un entorno de desarrollo integrado (IDE) basado en **Processing**, el cuál nos permite escribir ***sketches*** en lenguaje de programación Arduino.

Lenguaje Arduino

Este lenguaje está basado en C/C++. Un programa Arduino se ejecuta en dos partes:

1. `void setup()`

Es donde colocamos el código de inicialización, es decir las instrucciones que configuran la placa antes de que inicie el bucle principal del sketch.

2. **`void loop()`**

   Contiene el código principal del sketch.

Operadores de comparación

```
== igualdad                 != distinto
>  mayor que                <  menor que
>= mayor o igual que        <= menor o igual que
```

Operadores booleanos

```
&& (AND)              || (OR)               !(NOT)
```

Estructuras de control

`if (condición) {}`

Sirve para discriminar si se dio una determinada condición. Utilizaremos para la condición operadores de comparación. Todo lo que figure entre las llaves será ejecutado si la condición es verdadera.

`if (condición) {} else {}`

Si la condición es verdadera, ejecuta lo que figure entre las llaves y sino, lo que está a continuación del **else**.

`switch case (selector) {}`

Nos permite ejecutar un bloque de instrucciones, según sea el valor de **selector**. La instrucción **break**, termina la ejecución del **case**. Es necesaria incluirla al final de cada **case** pues, una vez se cumple la condición, se ejecutan las instrucciones linealmente hasta el final (aunque ya pertenezcan a otros **case**) o hasta encontrar un **break**.

```
switch case (selector)
{
     case etiqueta1:
         instrucciones1
         break;
     case etiqueta2:
         instrucciones2
         break;
      ........................
      default:
           instrucciones3
}
```

`while (condición) {}`

Repite las instrucciones que estén entre las llaves mientras se cumpla la condición.

`do { } while (condición)`

Su funcionamiento es similar a `while` pero con una diferencia: al menos una vez ejecuta las instrucciones aunque la condición no se cumpla.

`For (instruccion_inicial; condicion; instruccion_ciclica) {}`

Nos permite repetir las instrucciones una cantidad especificada de veces. En el siguiente ejemplo se repetirán cinco veces:

```
for (int i=0; i<5; i++){
    <instrucciones>
}
```

`continue`

se utiliza dentro de los bucles para que, una vez se ejecuta, volver a testear la condición, saltándose el resto de instrucciones que haya a continuación de él.

Funciones de entrada y salida

`pinMode (pin, INPUT|OUTPUT)`

Nos permite declarar un pin digital como entrada (INPUT) o como salida (OUTPUT). Ejemplo:

```
pinMode(8, OUTPUT); // setea el pin 8 como pin de salida
```

`digitalWrite (pin, HIGH|LOW)`

Nos permite activar o desactivar el pin digital. Ejemplo:

```
digitalWrite(8, HIGH); // activa el pin digital 8
digitalWrite(5, LOW); // desactiva el pin digital 5
```

`digitalRead (pin)`

Nos permite conocer el estado de activación de un pin. Ejemplo:

```
digitalRead(8); // Retorna HIGH
digitalWrite(5); // retorna LOW
```

`analogRead(pin)`

Lee el voltaje aplicado a un pin analógico de entrada y devuelve un número comprendido entre 0 y 1023.

`analogWrite(pin, valor)`

Cambia la velocidad del PWN en uno de los pines marcados como PWN. El ancho se codifica en 256 niveles, es decir, podremos introducir un número del 0 (apagado) al 255 (encendido) en el parámetro `valor`.

> ***PWM (Pulse width Modulation):*** *La modulación por anchura de pulso es una técnica que se emplea para producir señales digitales que filtradas, se comportarán como señales analógicas. El PWN en Arduino funciona a una frecuencia bastante próxima a 500Hz, lo que equivale a periodos de 2 milisegundos cada uno.*

Temporizadores

`millis ()`

Devuelve el tiempo en milisegundos transcurridos desde que la placa activó el programa actual.

`micros ( )`

Es lo mismo que `millis` pero en microsegundos.

`delay(milisegundos)`

Sirve para pausar los procesos de la placa durante un cierto tiempo, especificado por el parámetro milisegundos.

`delayMicroseconds(microsegundos)`

Es lo mismo que `delay()` pero en microsegundos.

Para consultar más instrucciones: http://arduino.cc/es/Reference/HomePage

Ejemplo para hacer que un LED aumente y disminuya su intensidad de brillo de forma cíclica:

```
const int LED = 9;  // definimos el pin para el led
int i = 0; // vamos a usar esta variable como contador

void setup() {
  pinMode(LED, OUTPUT);  // indicamos que el led es una salida
}

/*
    loop () es donde especificamos como debe comportarse nuestro
    dispositivo. Éste, no dejará de repetirse hasta  que apaguemos
    la placa.
*/
void loop() {
    // Aumenta la intensidad de brillo
    while(i < 255) {
        analogWrite(LED, i);
        delay(15);
```

```
        i++;
    }

    // Disminuye la intensidad del brillo
    for(i=255; i>0; i--) {
        analogWrite(LED, i);
        delay(15);
    }
}
```

El entorno de desarrollo integrado (IDE)

El IDE estándar de Arduino está escrito en Java. El comando necesario para instalar en **ArchLinux** es **`yaourt -S arduino`** y en distribuciones basadas en **Debian** **`apt-get install arduino`**. Gnoduino es una alternativa realizada en Python que puede instalarse en ArchLinux mediante **`yaourt -S gnoduino`**.

Referencias :

http://arduino.cc/
http://playground.arduino.cc/Learning/Linux
http://processing.org
http://wiring.org.co/
http://www.creativeapplications.net

NdR: *Arduino se auto-define como hardware de código abierto (open source) que comparte principios con el Software Libre y también con el Open Source (recordemos que Software Libre y Open Source NO son sinónimos). No obstante, todo el software Arduino disponibles, se encuentra liberado bajo una licencia GPL (Software Libre).*

Tip by Meri:

A veces creamos entornos chroot y necesitamos ejecutar aplicaciones gráficas en ellos. Para lograrlo, antes de entrar en el chroot, ejecutamos:

```
$ xhost +
$ export DISPLAY=:0.0
```

vinco orbis

Curso presencial de Introducción a Grails (México, DF)

25 %
de descuento

Vinco Orbis Proyectos inicia sus cursos de capacitación en **marzo del 2013**. A los lectores de Hackers & Developers Magazine en México, Distrito Federal les ofrecemos un 25% de descuento en el curso presencial de "Introducción a Grails", con duración de 8 semanas, horarios viernes (tardes) y sábado (mañanas). **Cupo limitado a 10 personas.** Informes, reservas e inscripción: grails@vincoorbis.com | CÓDIGO DE CUPÓN: **VO-GRAILS-HDMAG**

NO ACUMULABLE CON OTRAS OFERTAS Y/O PROMOCIONES. **VÁLIDO** DESDE EL 31/12/2012 **HASTA EL 22/02/2013** INCLUSIVE O HASTA AGOTAR LÍMETE DE 10 VACANTES (LO QUE SUCEDA PRIMERO).

Gentileza de Hackers & Developers Magazine: www.hdmagazine.org

Especial NoSQL: Introducción a NoSQL y servicios en la nube

ESPECIAL NOSQL

Las bases de datos NoSQL no pretenden reemplazar a las relacionales únicamente son otra opción de almacenamiento de datos. Cabe destacar que su arquitectura distribuida, flexibilidad en su estructura, escalabilidad y el manejo de enormes cantidades de datos son algunas de las características por las cuales se han vuelto cada vez más populares.

Escrito por: **Yecely Díaz** (M.I.A. & Desarrolladora Web)

Yecely es **Maestra en Inteligencia Artificial** y **Desarrolladora Web**. Aficionada de la Tecnología, Videojuegos y la Web. Ansiosa de aprender, contribuir y programar todas sus ideas. Ama lo que hace y el código es su vida.

Webs:
Blog: http://silvercorp.wordpress.com

Redes sociales:
Twitter: **@silvercorp**

El término NoSQL ("not only SQL") fue utilizado por primera vez en 1998 para referirse a una base de datos (BD) de código abierto que no ofrecía una interfaz SQL pero sí utilizaba el modelo relacional y, fue en el 2009 que Eric Evans retomó el término cuando Johan Oskarsson quiso organizar un evento en San Francisco para discutir el tema de las bases de datos distribuidas de código abierto; este tipo de BD suelen no proveer garantías ACID, término que hace referencia al conjunto de características necesarias para que ciertas instrucciones se consideren como una transacción.

ACID es un acrónimo de **A**tomicity, **C**onsistency, **I**solation y **D**urability que en español es Atomicidad, Consistencia, Aislamiento y Durabilidad.

El primer concepto es la propiedad que asegura que todas las operaciones hayan sido terminadas (*commit*). La consistencia es cuando se asegura que solo se inicien transacciones que se puedan terminar, es decir, se pasará de un estado válido a otro

válido sin "romper" con reglas o directrices que afecten la integridad de la BD.

El aislamiento es la propiedad que verifica que una operación no afecte a otras, siendo que dos transacciones ejecutadas al mismo tiempo sean completamente independientes sin afectar a alguna otra que se encuentre en ejecución.

Y por último la durabilidad se refiere a que una vez realizada la operación ésta persista y exista sin importar los fallos del sistema.

La característica más importante de NoSQL es que no impone una estructura de datos en forma de tabla y relaciones entre las mismas, es decir debemos olvidarnos de un esquema rígido y sin cambios, en este caso es más flexible ya que permite almacenar información de forma clave-valor (Cassandra, Riak, etc), documentos (CouchDB, MongoDB, BaseX, etc), mapeo de columnas o grafos (Neo4j, InfoGrid, HyperGraphDB, etc).

Actualmente las BD de tipo NoSQL han tomado mayor popularidad debido a que son utilizadas por las principales redes sociales como Facebook y Twitter por mencionar algunas, ¿y a qué se debe esto?, principalmente porque pueden manipular grandes cantidades de datos de una forma muy rápida y están preparadas para escalarse horizontalmente sin perder rendimiento alguno.

La escalabilidad es un término utilizado para referirse a la propiedad de aumentar la capacidad de trabajo o tamaño de un sistema sin que sea comprometida su funcionamiento y calidad, lo cual puede ser de manera vertical u horizontal. El primero se refiere a actualizaciones o modernización de los componentes que ya existen y el segundo, a cuando se aumenta el número de componentes; para el caso de NoSQL podemos incrementar el número de nodos.

¿Cuando se debe utilizar una base de datos NoSQL?

Aunque tenemos ventajas con NoSQL no debemos intentar reemplazar las bases de datos relacionales ya que dependiendo de las necesidades del proyecto es aquella que debemos utilizar o en algunos casos ambas; a continuación te planteo ejemplos reales para que quede más claro este punto.

Punto de venta. Para este sistema se requieren vínculos comprador-venta, costos, historiales, facturación, etc, por lo que no sería posible eliminar las relaciones y tampoco es necesario cambiar la estructura de nuestras tablas. En conclusión, NoSQL no es una opción.

Control escolar. De la misma manera que la anterior contamos relaciones alumno-materias, maestro-grupos entre otros, por lo que NoSQL tampoco sería una opción.

Red Social. Es una aplicación de gran escala, lectura/escritura de enormes cantidades, servicio a millones de usuario, información cambiante; cabe mencionar que pueden existir relaciones usuario-usuario, siendo posible almacenar cierta información en NoSQL y otra en SQL.

> ***Mi recomendación para saber si es factible utilizar NoSQL** es preguntarte ¿existirán relaciones? ¿necesito una estructura fija? ¿que cantidad de información almacenaré?, estos cuestionamientos te ayudarán a aterrizar tu proyecto y plantearte si es una buena opción para tu desarrollo.*

Sólo recuerda algo, **no porque sean tecnologías "nuevas" tu proyecto será innovador**, **utiliza únicamente aquello que te ayude a resolver problemas** y no ocasionarte más de los que puedas tener.

> *"Los buenos programadores resuelven problemas. Los mejores los evitan"*

NoSQL en la nube

Si deseas iniciarte en el mundo de las BD no relaciones te recomiendo empezar a utilizar los servicios en la "nube" de tal forma que analices como prefieres almacenar tus datos. Es por eso que te dejo los siguientes enlaces donde podrás crear tus primeras BD NoSQL en la nube.

Cassandra.io: Puedes utilizar su versión de prueba y API, su ventaja es que cuenta con librerías en los lenguajes Java, Ruby, PHP y Obj-C, lo cual es más sencillo realizar la comunicación con este servicio en la nube.

MongoHQ: Personalmente es un servicio que he usado desde hace un par de años, cuenta con un panel de administración y la opción de crear diferentes conexiones y bases de datos.

Couchbase: Escalabilidad, flexibilidad, consistencia son algunas de las características que podrás encontrar en su servicio. Cabe mencionar que su comunidad es amplia y actualmente tiene soporte para PHP, C, Python, Ruby y Java.

Redis: Almacenamiento llave-valor donde podrás tener la opción de un servidor maestro, otro esclavo y un panel donde puedes configurar tus datos.

En un siguiente artículo me enfocaré en mongoDB donde te mostraré como instalarlo, crear tu BD, documentos y las operaciones básicas.

Tip by Yesi Days:

Para posicionarte al inicio de tu Sitio Web al dar click a un botón o enlace, agrega al evento la siguiente instrucción en jQuery:

```
$('html, body').animate({scrollTop: 0}, 0);
```

Especial NoSQL: estructura interna, código y contexto

En el artículo anterior, se ha hecho una introducción a las bases de datos NoSQL. Aquí nos enfocaremos en tratar de comprender la estructura y comportamiento interno de esta nueva generación de DDBB que permite manipular grandes cantidades de información, haciendo una comparación con otros tipos de bases de datos, con el fin de encontrar nuevas arquitecturas de solución que ofrecer en alguna implementación de Software en la que nos encontremos envueltos y se requiera.

Escrito por: **Rosa María Orellana** – Colaboradora invitada

Rosa María es **Ing. Informática** de Perú. **Especialista en Soluciones con Tecnologías Web.** Co Fundadora de la Comunidad **Python Perú**, promotora de **Agile Girls, GDG Girls** en el Perú, Coordinadora del grupo **LinuxIDES**. Analista de Arquitecturas y Gestora de Proyectos de Software con Tecnologías Web y Móvil, **Lider Agile**, Analista, amante de Python y gran difusora del FOSS en General

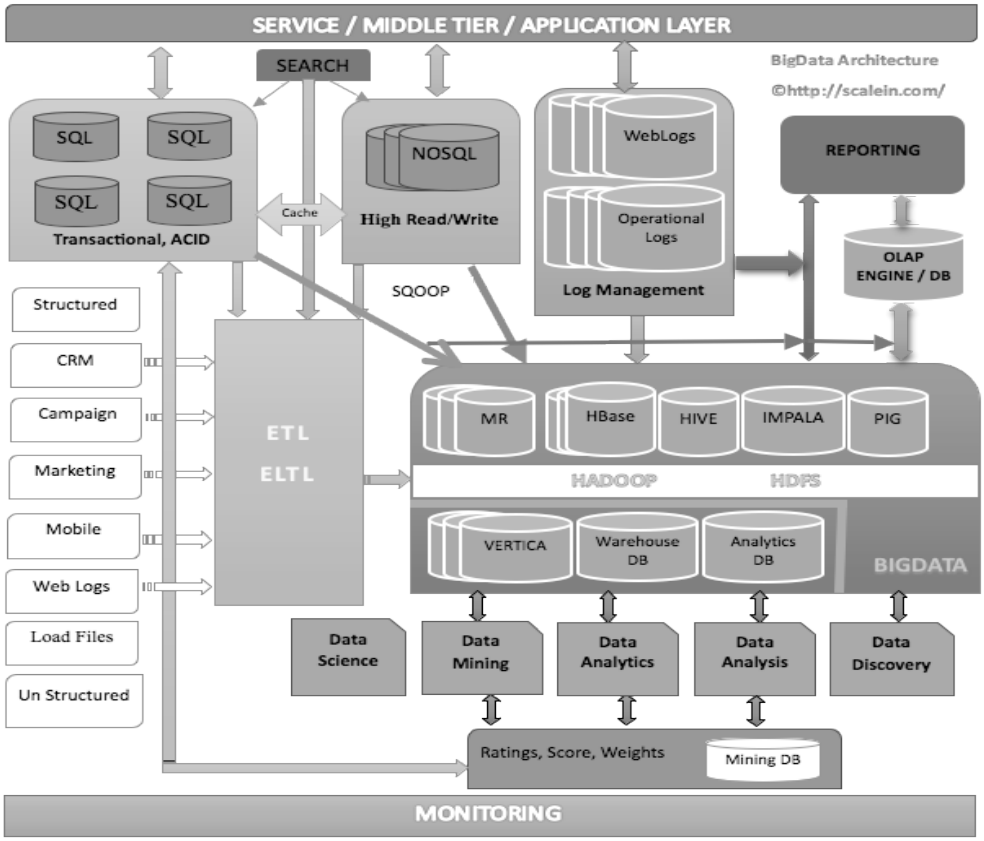
Webs:
http://about.me/rosamaria

Redes sociales:
Twitter: **@assoritam**

La gran cantidad de información que va apareciendo año tras año y que exigen a los sistemas de software que mejoren sus métodos y modelos en la manipulación, análisis y administración de datos que almacenan, han dado paso a plantear nuevas soluciones y a atacar problemas que existen con las bases de datos relacionales tradicionales; problemas como la complejidad de consultas, la baja eficiencia y problemas de alta escalabilidad. Para poder entender un poco más las bases de datos NoSQL es conveniente explicar previamente, el concepto de *Big Data.* Éste se refiere a aquellos sistemas que deben procesar, almacenar y analizar grandes cantidades de datos con una alta velocidad de respuesta.

Las soluciones de sistemas con estructuras Big Data con mayor éxito, provienen de plataformas Open Source.

La arquitectura y los componentes que involucran este tipo de soluciones *Big Data*, se puede ver resumida en la siguiente imagen:



Fuente de Imagen: http://scalein.com/images/bigdata-scalein-architecture.png

En la parte superior izquierda se observa el área de donde proceden los datos. Pueden ser fuentes de datos estructuradas, semi-estructuradas o sin estructuras. Recordemos que el presente artículo centra la importancia de las NoSQL y el proceso de transformación y almacenamiento que éstas ofrecen.

También se observa en la parte central izquierda de la imagen los canales de transformación y migración de datos, como son ETL (Extraer, Transformar y Cargar) ó ETLT (Extract, Transform, Load y Transform). Entre estos, podemos citar como ejemplo algunas tecnologías libres y open-source como bash, python, perl, javascripts o Kettle-Pentaho Project, las cuáles ayudan a este tipo de procesamiento.

Aquí está lo interesante. SQOOP -que es una herramienta diseñada para transferir datos- puede importar datos desde un sistema de bases de datos relacional (RDBMS) tal

como MySQL u PostgreeSQL al sistema de archivos distribuido Hadoop[10] (HDFS), transformar los datos en Hadoop MapReduce y, exportarlos de nuevo en un RDBMS.

Luego, en la parte central del esquema, ubicamos una tercera sección compuesta por las áreas de procesamiento e integración de datos, las cuales contienen las fuentes de datos -estructurados y no estructurados - más grandes, en un mismo lugar, para facilitar el proceso. Hadoop y Ecosistemas (Hadoop / HDFS, MapReduce, HBase, Colmena, Impala, Pig, entre otros) utilizan HDFS[11] como almacenamiento nativo.

Este tipo de arquitecturas considera los almacenamientos de datos como DatawareHouse y soluciones de Analytics las cuáles son útiles para ser consumidas por los componentes de consumo de datos (ej. Ikanow[12]).

Finalmente en la parte inferior del esquema se observan los datos de consumo para ser utilizados por los usuarios finales en capas internas (ad-hoc) o externas a través de APIs, para minería de datos, Ciencia de Datos, Análisis para predecir o estimar el rendimiento general.

Ahora bien, luego de haber descrito un poco la estructura general y el proceso para la manipulación de grandes cantidades de datos nos enfocamos en revisar las estructuras que bajo este esquema, proponen las bases de datos NoSQL.

Los sistemas web son los principales sistemas que presentan estos grandes retos de almacenamiento, por la alta velocidad, eficiencia y disponibilidad con que deben funcionar. Es así que aparece un conjunto nuevo de base de datos no relacionales (**N**ot **O**nly **SQL**).

> *NoSQL, permite el almacenamiento de datos en altas frecuencias de lecturas y escrituras, así como el soporte de almacenamiento de millones de usuarios.*

Estructura básica de una Base de Datos NoSQL

Veamos una estructura básica para comprender los términos principales que utilizan las Bases de Datos NoSQL en similitud a una Base de Datos SQL para comprender con mayor claridad.

```
----------- NoSQL -----------
Servidor
|__Base de datos
   |__Colección
      |__Documento (BSON)
         |__Campos
```

```
----------- SQL -----------
Servidor
|__Base de datos
   |__Tabla
      |__Fila
         |__Columnas
```

10 Hadoop: http://hadoop.apache.org/#What+Is+Apache+Hadoop%3F
11 http://www-01.ibm.com/software/data/infosphere/hadoop/hdfs/
12 http://www.ikanow.com/

Entonces podemos decir que las colecciones son análogas a las tablas (las consultas se hacen a este nivel), los documentos a las filas de las tablas (por tanto los documentos tienen un único ID) y los campos a las columnas de una tabla que corresponde a una estructura de base de datos relacional.

Características Principales

Entre las ventajas y características particulares que tienen estos modelos de datos, frente a las base de datos tradicionales (RDBMS) podemos considerar las siguientes:

- Como se ha mencionado, una de las principales características de una base de datos NoSQL, es la capacidad de poder manejar grandes cantidades de datos.
- Este tipo de almacenamiento de datos no-relacionales, no requiere esquemas de información fija.
- No realizas un diseño de tablas y estructuras por adelantado (como normalmente haces con las bases de datos).
- El modelo de almacenamiento de datos no mantiene una estructura relacional entre sus datos, lo cuál explica que no se utilizan mecanismos rígidos de consistencia.
- Alto rendimiento.
- Mantienen una estructura distribuida.
- Escalabilidad Horizontal
- La Velocidad

¿Cómo se clasifican?

Existe una clasificación de base de datos no-relacionales considerando la forma en como éstas almacenan la información. Aquí solo haremos una descripción general con un ejemplo básico de esta clasificación:

Bases de datos documentales.

Son Bases de Datos que permiten almacenar, recuperar y administrar la información orientada a una noción abstracta de un documento. Esta clasificación envuelve uno de los principales tipos de base de datos NoSQL.

Las bases de datos orientadas a documentos son similares a los registros o filas en bases de datos relaciones, pero su estructura es bastante simple y menos rígidas. Es decir ,no se encuentran definidos bajo un esquema estándar sino que es dinámico permitiendo la

evolución de esquemas. Ej. MongoDB[13], CouchDB[14].

A continuación se muestra un breve ejemplo en CouchDB, de como se representa un registro en una base de datos documental:

```
{
  "Subject": "Yo amo el basketball"
  "Author": "Rosa María"
  "PostedDate": "12/21/2012"
  "Tags": ["Jordan", "basketballl", "Deporte"]
  "Body": "Yo he decidido practicar basketball."
}
```

Bases de datos en grafos.

Recordemos nuestras clases de estructuras discretas donde veíamos la definición inspirada por Euler y la teoría de grafos. Este tipo de base de datos utiliza este concepto, donde se presenta la información en nodos. También se les llamada BDOG y algunos no la consideran del todo un modelo de datos NoSQL. Ej. Neo4j[15]

A continuación, se muestra un breve ejemplo del manejo de Base de Datos orientado a grafos como es Neo4j con la creación de un grafo en Java [16].

Variables que vamos ha emplear:

```
GraphDatabaseService graphDb;
Node firstNode;
Node secondNode;
Relationship relationship;
```

La sintaxis para comenzar el servidor de la Base de Datos y realizar la conexión en Neo4j es:

```
graphDb = new GraphDatabaseFactory().newEmbeddedDatabase( DB_PATH );
registerShutdownHook( graphDb );
```

Ahora vamos a crear un grafo pequeño que consta de dos nodos, conectados con una relación y algunas propiedades:

```
firstNode = graphDb.createNode();
firstNode.setProperty( "message", "Hola, " );
secondNode = graphDb.createNode();
secondNode.setProperty( "message", "El Mundo se acaba!" );
relationship = firstNode.createRelationshipTo( secondNode,
RelTypes.KNOWS );
relationship.setProperty( "message", "es 21 de Diciembre. " );
```

13 http://www.mongodb.org/
14 http://couchdb.apache.org/
15 http://www.neo4j.org
16 https://github.com/neo4j/neo4j/blob/1.9.M02/community/embedded-examples/src/main/java/org/neo4j/examples/EmbeddedNeo4j.java

Luego imprimimos el resultado:

```
System.out.print( firstNode.getProperty( "message" ) );
System.out.print( relationship.getProperty( "message" ) );
System.out.print( secondNode.getProperty( "message" ) );
```

Donde tenemos como salida:

```
Hola, es 21 de Diciembre. El Mundo de acaba!
```

Bases de datos clave-valor.

Las Base de Datos clave-valor se relacionan mediante una llave. Este tipo de base de datos sirve para manejar grandes cantidades de datos donde a pesar de tener filas y columnas como una base de datos tradicional, distribuye la información en valores no únicos y asociados entre sí. No están pensadas para el análisis de datos, sino para el almacenamiento puro y duro. Ej. Cassandra[17].

Para entender mejor como funcionan las Base de Datos clave valor observemos las siguientes tablas:

ID	Usuario	Nombre	Email	Fecha
1	rosamaria	Rosamaría Pérez	rmp@tp.com	02/04/1979
2	chivi	Silvia García	chivi@hatmail.com	18/02/1985
3	jessica	Jessica Robles	jrobles@micorreo.es	03/04/2012

En un modelo de datos clave-valor tenemos contenedores a los que también se les llama *cabinets*. En cada contenedor podemos tener tantas parejas de clave-valor como se desee distribuir; se puede además, tener datos del mismo tipo o diferentes, según se requiera. La eficiencia está basada en que a cada clave se le asocia un valor. Por ejemplo tenemos:

El primer contenedor relaciona los nombres de los usuarios como clave y y las contraseñas, como el valor asociado a cada clave.

```
[users.cab]
rosamaria=srfgfd234gf
chvi=partisimd24r5
jessica=jeka34fte
```

En otro contenedor podríamos tener los datos de los usuarios:

```
[user_data.cab]
rosamaria_nombre=Rosa María
```

17 http://cassandra.apache.org/

```
rosamaria_email=rmp@tp.com
rosamaria_fecha=02/04/1979
chivi_nombre=Silvia García
chivi_email= chivi@hatmail.com
chivi_fecha=18/02/1985
```

Como se ve, la clave se forma por el usuario+dato (usuario_dato), de esa forma, al *loguearse* un usuario podemos preguntar al primer contenedor que nos devuelva la contraseña: si no devuelve nada, el usuario no existe; si devuelve algo verificamos la contraseña y si es correcta podemos recuperar el resto de valores.

Base de datos multi-valor.

Este modelo de datos esta relacionado a una lista de datos. Comprende tres dimensiones de estructuras de datos: campos, valores y subvalores. Algunos Ej. ArangoDB[18], AlchemyDB[19].

A continuación vemos la estructura de datos para un usuario con una lista de e-mails (lista de datos), relacionados a un solo usuario.

```
{
      "first-name" : "Rodrigo",
      "last-name" : "Catter",
      "dob" : "26-Ago-1985",
      "addres" : {
            "unit-number" : "15",
            "street-numer" : "23",
            "street-name" : "JuanXXIII",
      },
      "emai-addrs" :
      {
            "7dfcb445-74d3b4f17-aa66-dgfbe3dbd982":  "rcatter@lapositiva.com.pe",
            "5bd1003-454df3456-9cb4-4rrertvdlksdfg": "cbro87@gmail.com",
            "8dsewe4-56fgjee321-ft42-76effgkwlrt52": "mani23@hotmail.com.pe",
      }
}
```

Bases de datos orientada a objetos.

Son las que incorporan todos lo conceptos importantes de la estructura "objeto".

Este modelo de base de datos representa la información en forma de los objetos que normalmente se usa en la programación orientada a objetos. Es importante resaltar que las Bases de Datos orientada a objetos objetos son diferentes de las bases de datos relacionales. Ej. ZODB[20].

A continuación describimos un ejemplo para insertar un nuevo usuario, creando el objeto `user`, llenarlo con datos e insertándolo en la instancia `Btree`, se deje preparado para

18 http://www.arangodb.org/
19 http://code.google.com/p/alchemydatabase/
20 http://www.zodb.org/

realizar una transacción. El ejemplo se basa en la guía de ZODB escrito en el lenguaje de Programación Python.

```
# Se crea la instancia de un Nuevo usuario
import transaction
newuser = User()
# Aquí se añade un atributo al objeto :)
newuser.id = 'amk'
newuser.first_name = 'Laura'
newuser.last_name = 'Ramirez'

# se agrega el objeto al BTree con el ID
userdb[newuser.id] = newuser

# Se realiza el cambio
transaction.commit()
```

Base de datos tabular.

También conocida como SSAS tabular, la estructura que maneja los datos es en tablas con filas y columnas y están diseñadas para manejar gran cantidad de información. Ej. HBase[21], Hypertable[22].

HBase está construido sobre HDFS y ofrece búsquedas rápidas de registro. Los usuarios almacenan filas de datos en las tablas marcadas. Una fila de datos tiene una clave que puede ordenar un número arbitrario de columnas.

Tip by Rose:

Para entender bien el concepto base de datos relacionales y no relacionales como leí una vez, pensemos en nuestras manos izquierdas y derecha, cada una ofrece fortalezas individuales para cada tarea en específico. Por ejemplo, un beisbolista sabe que una de sus manos es mejor para lanzar la pelota y la otra para atraparla; puede ser que cada mano intente hacer la actividad de la otra, mas sin embargo, el resultado no será el más óptimo.

21 http://hbase.apache.org/
22 http://hypertable.org/

Scratch: Imagina, programa, comparte

EDUCACIÓN

Empezar en el mundo de la programación divirtiéndose, es una gran y atractiva manera. Esto puedes hacerlo usando un entorno de programación visual basado en *Squeak* para realizar secuencias animadas (historias, juegos, creaciones artísticas, etc) con sonido (o sin él), teniendo como gran ventaja, aprender o mejorar tus *skills* al programar.

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

Estudiante de Ingeniería Informática. Miembro de **APESOL** (Asociación Peruana de Software Libre) y 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**

Scratch[23] es un entorno de programación de mucha ayuda para principiantes de este maravilloso mundo y arte de la programación, ya que obtienes resultados sin tener que escribir el código con la sintaxis exacta. Fue creado por Lifelong Kindergarten Group del MIT Media Lab[24]; está escrito en Squeak[25], contando así con potentes facilidades multimedia.

El *"scratching"* -que es la técnica usada en la música en el Turntablism (el Djing), considerado como el arte de arreglar o crear música mediante efectos de sonido y manipulación de la rotación de discos de vinilo-, es de donde deriva el nombre de Scratch, la similitud de ambos "scratch" está en la reutilización de piezas: objetos, gráficos, sonidos, secuencias de comandos, etc ya que estos pueden importarse o combinarse en nuevos programas.

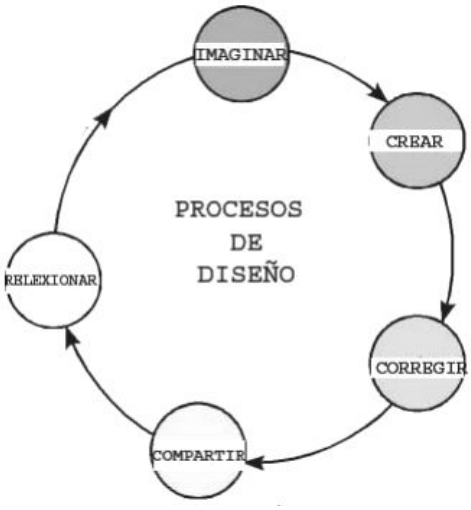
23 http://scratch.mit.edu/
24 http://www.media.mit.edu/news/fact-sheet
25 http://www.squeak.org/

Los avances en diseño de interfaces hacen que Scratch sea más atractivo para los que asumen el reto de aprender a programar; permite construir y probar la mayoría de procesos táctiles ya que la prioridad de los creadores fue hacerlo tan fácil de aprender hasta incluso por niños, ya que al arrastrar y soltar bloques de condiciones (con parámetros) y de consecuencias (acciones), Scratch permite crear fácilmente programas que controlan y mezclan imágenes, animaciones, música y sonido. En pocas palabras podemos decir que le permite aprender a programar mientras se divierte.

> *Cada nueva idea debería tener al menos un momento en el que parezca una locura.*

Ingredientes básicos de un proyecto de Scratch

Los proyectos en Scratch[26] deben tener *Sprites* (Objetos móviles programables), conocidos solo como objetos. Éstos, pueden modificarlos dándoles un disfraz diferente (una abeja, un niño o cualquier cosa), este disfraz puede ser cualquier imagen (dibujada en un editor de pinturas, importada de su disco, de un sitio web, etc). Estos objetos pueden tener instrucciones para que se muevan, suenen, reaccionen y demás; para eso debe encajar bloques formando pilas, llamadas programas (scripts). Finalmente se ejecutarán en orden, los bloques desde la parte superior hacia la inferior.



De esta manera se comprenden fácilmente conceptos que están integrados como procesos interactivos (bucles), criterios condicionales (if, then, if-not), las coordenadas en un plano, las variables, etc. y entendiéndolo para el control de la visualización de una animación que uno mismo construye; uno mismo parte de una idea y crea el prototipo funcional (el modelo) con las soluciones que considere necesarias ya que se podría seguir creando proyectos de forma indefinida.

Interfaz

En la pantalla del programa al ejecutarlo se muestran las siguientes partes principales[27]:

26 http://scratched.media.mit.edu/resources/new-scratch
27 http://vimeo.com/29457909

- **Escenario:**

  Es la zona de trabajo, la zona blanca más grande donde se desarrollarán las acciones.

- **Botones de nuevos sprites:**

  Hay tres botones (debajo del escenario) que permitirán que busquemos o creemos nuevos actores para nuestra acción.

- **Lista de sprites:**

  Zona donde aparecerán los sprites en miniatura que vayan a actuar, se pulsa sobre el que se desee editar.

- **Barra de herramientas:**

  Es la barra que contendrá lo necesario para mover objetos, copiar, cortar, aumentar tamaño, etc.

- **Bandera verde:**

  Se ejecutan los guiones o programas que hayamos creado.

- **Botón rojo:**

  Sirve para detener la acción.

- **Notas del proyecto:**

  Sirve para añadir comentarios al proyecto.

- **Paleta de bloques:**

  Es la caja de instrucciones para que los actores hagan cosas, son los bloques para crear scripts o programas. Son 8 cajas diferentes de piezas para distintas acciones, cada una de un color: movimiento, apariencia, sonido, lápiz, control, sensores, números y variables.

- **Área de scripts:**

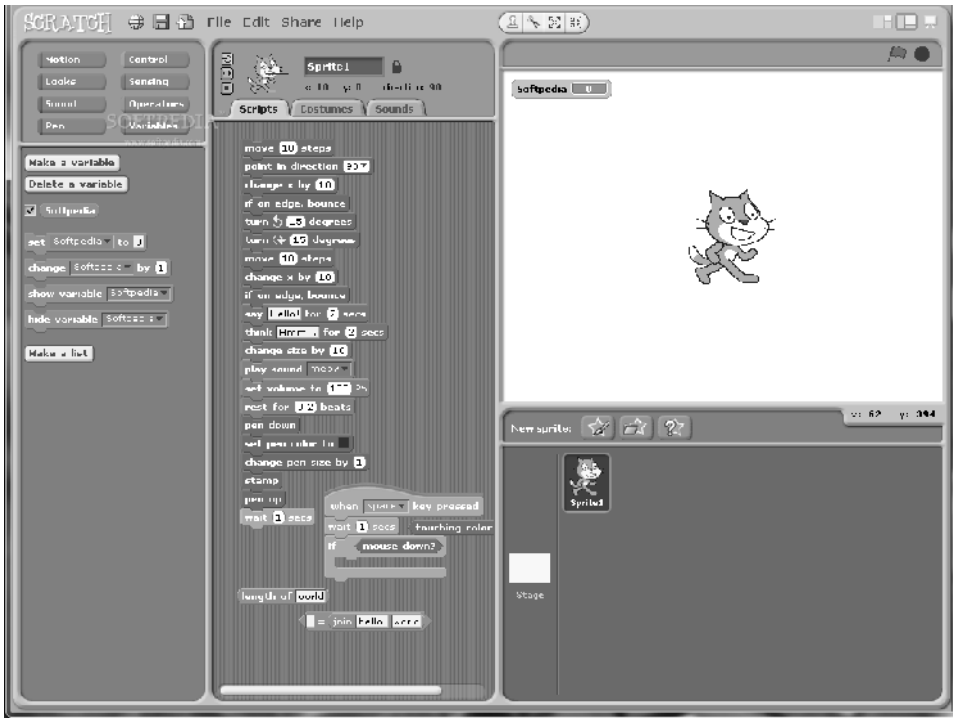
  Aquí se arrastran los bloques de nuestras cajas de piezas para ir construyendo guiones o programas.

- **Pestañas:**

  Nos permiten añadir guiones, disfraces o sonidos al guión que construyamos.

- **Información sprite seleccionado:**

  Aparecerá el nombre del sprite y una miniatura del mismo.



Comunidad de usuarios de Scratch

El programa se usa en distintas partes del mundo y así como su eslogan lo dice: Imagina, programa, comparte; indica que el compartir y la creatividad son partes importantes detrás de la filosofía de Scratch. Los programas creados pueden subirse al sitio web por cualquier miembro de la comunidad y luego cualquier persona puede descargar el código completo para estudiarlo o añadirlo en otros proyectos, puedes tener como favoritos los programas creados por otros miembros y compartir sus ideas, todos los proyectos están bajo una licencia Creative Commons y se sabe que aproximadamente 10 millones de personas visitan la página cada mes, Scratch incluso cuenta con una comunidad para educadores llamada ScratchEd[28].

¿Por qué Scratch?

Para aprender más sobre el desarrollo de habilidades de aprendizaje del siglo XXI, el reporte sobre "Aprendiendo en el siglo XXI" identifica nueve tipos de habilidades de aprendizaje divididos en tres áreas claves y veremos como Scratch soporta el desarrollo[29].

Área 1: Habilidades de información y comunicación

28 http://scratched.media.mit.edu/
29 http://learnscratch.org/resources/why-learn-scratch

1. **Información y medios de las habilidades de lectura**

   Al trabajar en proyectos de Scratch, los estudiantes aprenden a seleccionar, crear y gestionar múltiples formularios de medios, incluyendo texto, imágenes, animación, grabaciones, etc. Mientras los alumnos ganan experiencia creando con medios, también son más perceptivos y críticos en analizar los medios que los rodean.

2. **Habilidades de comunicación**

   La comunicación efectiva en el mundo actual requiere más la habilidad de leer y escribir texto. Scratch involucra a la gente joven a escoger, manipular e integrar una variedad de medios para expresar su propia creatividad y persuasión.

Área 2: Habilidades para pensar y resolver problemas

3. **Pensamiento crítico y pensamiento sistémico**

   A medida que aprenden a programar en Scratch, la gente joven también se involucra en el razonamiento crítico y el pensamiento sistémico para construir proyectos, necesitan coordinar el tiempo e interacción entre múltiples objetos ("sprites"). La habilidad de programar entradas interactivas dan a los estudiantes experiencias directas con detección, captación y otros conceptos fundamentales de los sistemas.

4. **Identificación del problema, formulación y solución**

   Scratch soporta el problema buscando y resolviendo un significativo contexto de diseño. Crear un proyecto requiere pensar en una idea e implementarla usando los bloques de programa; ha sido diseñado para que los alumnos puedan cambiar dinámicamente piezas de código e inmediatamente ver resultados, de esta manera los alumnos experimentan una resolución de problemas iterativo.

5. **Creatividad y curiosidad intelectual**

   Scratch refuerza el pensamiento creativo, una habilidad importante que está en incremento en el rápido cambio del mundo actual, la gente joven se involucra buscando soluciones innovadoras a problemas inesperados, no solo aprendiendo como resolver un problema predefinido sino estando preparado para enfrentar a las nuevas soluciones como nuevos retos.

Área 3: Habilidades interpersonales y auto-direccionales

6. **Habilidades interpersonales y colaborativas**

   Debido a que los programas en Scratch están construidos en bloques gráficos, el código es más fácil de leer y compartir que otros lenguajes de programación. Los objetos visuales y el código modular soporta la colaboración y permite que los alumnos trabajen juntos en proyectos e intercambien objetos y código.

7. **Auto-dirección**

Tomar una idea y pensar como programarla en Scratch requiere persistencia y práctica. Cuando la gente joven trabaja en ideas de proyectos encuentran un significado personal, ya que sus ideas darán motivación interna y superarán retos y frustraciones encontradas en el diseño y en el proceso de resolución de problemas.

8. **Responsabilidad y adaptabilidad**

   Cuando los estudiantes crean proyectos en Scratch tienen una audiencia en mente y necesitan pensar sobre como otras personas reaccionarán y responderán a sus proyectos. Desde que los proyectos son fáciles de cambiar y revisar, los alumnos pueden modificarlos basados en la retroalimentación de otros.

9. **Responsabilidad social**

   Debido a que los programas se comparten, los alumnos pueden provocar discusión de temas importantes con otros miembros de su entorno de aprendizaje inmediato como con la amplia comunidad internacional de Scratch

> *Para el creativo lo imposible no es más que aquello que aún no se ha intentado.*

Tip by Miracle:

Otro modo de recordar contraseñas seguras para distintos servicios: usa la técnica "base + variable". ¿Cómo funciona? (ejemplo) **Base:** hdmagazine → **hDm@g@z1n3**. **Variable:** hDm@g@z1n3**gmail** | hDm@g@z1n3**identica**

EN EXCLUSIVO PARA HACKERS & DEVELOPERS

Despidiendo el año junto a Richard Stallman

por Eugenia Bahit

Inteligente, comprometido con sus ideales; divertido y amable, con la humildad que solo las personas de una grandeza espiritual distinguida pueden tener. Luchador, perseverante; honesto, directo y sin tapujos a la hora de expresarse y dar sus opiniones. Fiel amante de la libertad, con una distinguida trayectoria y una actitud coherente entre "el dicho y el hecho", de la cuál pocos pueden darse el lujo. Richard Matthew Stallman: un hombre que inspira.

Preparativos...

Corría el 4 de diciembre. Se acercaba el momento de hacer planes para el especial de fin de año. Recién terminábamos de publicar la segunda edición del Magazine y solo sabíamos que teníamos que hacer algo "verdaderamente especial". Algo que como equipo, nos ayudara a marcar "un antes y un después". Algo que nos fortaleciera como grupo y que nos hiciera sentir orgullosas de trabajar las unas con las otras.

Fuimos arrojando ideas y así surgieron los "tips" (de finales de artículos), los cupones de descuento y algo **tan ambicioso como utópico que jamás creímos podía convertirse en realidad**: conseguir una entrevista con Richard Stallman.

Utópica y ambiciosa (o no), semanas después, me animé y le escribí un e-mail en un inglés que seguramente daba pena. Largo. Larguísimo, como si no hubiese sido consciente del tiempo que le llevaría leerlo. Como si ni siquiera hubiera tenido consciencia, del poco conocimiento que tengo de su idioma natal.

Le conté sobre el Magazine; le hablé de lo que hacíamos y le ofrecí leer la revista. Para mi sorpresa, la respuesta estaba en mi bandeja de entradas, en menos de 24 horas: **"It sounds like a good goal** (...)**"**, fue la primera línea de su respuesta.

*Foto: **Martin Kozák** (bajo dominio público, via Wikimedia Commons)*

Intercambiamos algunos correos más, hasta que en un perfecto español, me dijo: **"¿Quieres llamarme el sábado?"** y me adjuntaba su número de teléfono.

Decir emoción, me suena a "poco" para describir el cúmulo de sensaciones que se han cruzado en mi, desde ese día, hasta hoy, sábado 29 de diciembre a las 17:30 HS de Argentina, momento en el cual, tuvimos la siguiente charla:

Richard, ¿cómo es un día tuyo en la fundación dentro de la Free Software Foundation? Mi trabajo es mayormente leer y contestar mensajes y son de muchos tipos. Por ejemplo hay noticias, me piden decidir algo, hay mensajes desde los usuarios... entonces son muchos tipos de trabajo pero casi todos llegan por correo y envío mi decisión o respuesta por correo.

Por principios, en la FSF decidí no aceptar dinero

Y tu trabajo en la Free Software Foundation ¿es 100% voluntario? ¿No cobras un sueldo? Si, no recibo salario y la FSF no paga mis viajes. Es por principios. Decidí no aceptar dinero de la FSF porque tenía que pedir a otros que trabajen como voluntarios (…).

Y más allá de la Free Software Foundation ¿qué otro trabajo tienes o qué otro trabajo "remunerado"? Me pagan por quizás la mitad de mis conferencias.

¿Te gusta realmente viajar o te estresa? Me gusta. No siempre, pero mayormente me gusta.

¿Cómo surgen las campañas dentro de la FSF? Alguien tiene la idea, a veces yo, a veces alguien del equipo. Porque tenemos a dos empleados para las campañas (…) **y si por ejemplo algún usuario envía un e-mail a la FSF proponiendo una campaña...** si nos gusta, lo haremos. **¡Es muy bueno saberlo!** Siempre valoremos las buenas sugerencias.

Debian en sus foros, aconseja solucionar problemas instalando Software privativo

Con respecto a la FSF, el Software Libre y las distribuciones GNU/Linux aceptadas ¿en qué punto se encuentran actualmente con Debian? Debian tiene problemas. Un problema es que distribuye y recomienda paquetes no libres. Pero también en sus foros hay preguntas "¿Cómo puedo hacer esto?" y la respuesta es instalar ese programa privativo. Por eso, no endosamos Debian. Saca algunos programas privativos como soluciones y tratarlos como soluciones es negar... es rehusar a considerarlos como problemas. Entonces esa contradicción entre su práctica y nuestra filosofía es nuestro motivo de no avalar. **Leí hace poco en un foro, que Debian había presentado otra versión a la Free Software Foundation para ver si era aceptada ¿es cierto?** No recuerdo (...) me sorprendería. No comprendo como una nueva versión de algo podría resolver este problema. Porque otro criterio para aprobar una distribución GNU con Linux es que no tenga relación visible con ningún programa privativo. Es decir que si el mismo proyecto publica dos versiones y una contiene software privativo, no podemos aprobar ninguna. Tiene demasiada relación con software privativo. Si Debian echa los paquetes privativos; si corta su relación con esos paquetes, podríamos aprobarlo.

Con Linus Torvalds, no somos amigos

Con Linus Torvalds ¿tienes alguna relación? De disputa. No somos amigos. **No son amigos... ¿puedo preguntar por qué?** Nunca estuvo de acuerdo con la filosofía del Software Libre. Políticamente nunca estuvo de acuerdo con nosotros. Pero también tiene la tendencia de enojarse y decir cosas muy duras a quien quiera. Incluso a nosotros -que no es amigable-. Pero también nos critica por nuestro intento de proteger a los usuarios de la práctica de "tivoización" (NdR: se refiere a la castellanización del término inglés "Tivoization[30]"), qué significa fabricar computadoras con Software bajo la GPL versión 2 (usualmente), de manera que el código fuente es libre pero que el ejecutable en la computadora es privativo. Y hoy en día muchos dispositivos Android lo hacen con Linux, porque Linux se publica bajo la versión 2 de la GPL que escribí en el año '91. En aquel año no existía el problema de la tivoización y no anticipaba este problema entonces no hice nada para evitar este problema. Y Torvalds quiere mantener Linux bajo la versión 2 de la GPL para que permita la tivoización. Es lo que dice. El está en favor de permitir que los usuarios pierdan su libertad, así. Decidió quedarse con la versión 2

30 http://es.wikipedia.org/wiki/Tivoizaci%C3%B3n

que es un problema. Pero el aspecto central de este problema es la tivoización que quita la libertad a muchos usuarios. Pero, otra cosa es, que escribió un programa importante. El núcleo que se usa con el sistema operativo GNU en la combinación GNU con Linux.

Muchas personas llaman Linux al sistema entero, dándole el crédito de nuestro trabajo a otros

Pero erróneamente muchos se acostumbraban a llamar el sistema entero como Linux, dándole todo el crédito para nuestro trabajo anterior. Entonces, comencé a pedir siempre que la gente reconozca nuestro trabajo. Pero esto no es una crítica de Torvalds. Durante unos años no se oponía a esta campaña nuestra pero más tarde, desde hace unos años, intenta argumentar que es incorrecto llamar al sistema como GNU con Linux. Intenta persuadir a la gente que no reconozca nuestro trabajo. **Pero ¿qué fue lo que sucedió con Linus Torvalds para que de repente, primero estaba a favor y no criticaba y ahora, hace unos años empezó?** No se. No se porqué pensaba "así" y luego "pensaba asá". Uno tendría que conocerlo mucho más que yo (…). **Y ¿alguna vez le preguntaste?** No. Porque cuando alguien llega a ese punto, no tengo porque hablar con él. Cuando alguien intenta convencer a la gente de que no reconozca nuestro trabajo, es tratarnos muy mal.

No admiro a Dennis Ritchie

Y hablando de Linus Torvalds y "famosos" en el ámbito de la informática, alguna vez ¿tuviste oportunidad de encontrarte con Dennis Ritchie? No. **¿nunca tuviste oportunidad, nunca la buscaste...?** No, ni ganas. No me interesó. **No te interesó... ¿puedo preguntar por qué?** ¿Qué hizo que me interesaría? **A ver...** solo hizo trabajos técnicos. No lo admiro por eso. Desarrolló un sistema operativo no ético que tuve que reemplazar. Eso no es lo que admiro.

Hablando de privativo... Python actualmente, tiene una licencia compatible con la GPL. Pero tuvo una especie de ida y vuelta con si era compatible o no era compatible, en versiones creo que anterior a la 2. Fue hace mucho tiempo. No recuerdo los detalles. Hablamos con el desarrollador y por fin... **¿Con Guido Van Rossum?** Si. Pero no fue solo él, porque fue su empleador el que hizo el problema y no él. **Ah, mira que interesante...** trabajaba, quizás para una Universidad, no lo recuerdo, pero su empleador deseaba cambiar la licencia y creó un problema, que por fin, resolvimos.

*Foto: **Lionel Allorge** - CC-BY-SA-3.0*

Yendo a algo más personal... ¿eres de esas personas que se aburren con facilidad...? Si. ¿Si? **Y ¿eres de los que necesita estar siempre haciendo algo productivo?** No necesariamente productivo. A veces hago algo por placer y también a veces necesito divertirme. **Y ¿Qué cosas haces para divertirte?** A veces cuando tengo quizás un poco de sueño y no sostengo concentrar más en algo difícil, me divierto con algún videojuego por algún tiempo. **¡Te gusta jugar videojuegos! ¿Qué videojuego juegas?** Prefiero no decirlo, pero es Software Libre.

Y con respecto a la programación, ¿sigues programando a menudo? No. **¿estás más avocado a lo que es las charlas y demás...?** Si. Mi trabajo es el avance del movimiento. **Y ¿no extrañas sentarte a programar durante horas?** No. **¿No?** No. **Bueno, bien. Y hablando de tus proyectos, GNUPedia ¿en que quedó?** Se fusionó con Nupedia para lanzar Wikipedia. **Ah, no sabía.** Así nació Wikipedia, con la fusión de dos proyectos: Nupedia y GNUPedia...

Y lo divertido es, que hablaba con ellos durante más o menos un año, pero solo raras veces y entonces, no sabía que había dos. Y por fin llegaron dos mensajes dentro de bastante poco tiempo para que supiera que eran dos y luego los puse en contacto entre ellos.

Ser un buen programador es hacer programas útiles que funcionen bien

¿Qué significa para vos ser un buen programador? Hacer programas útiles, que funcionen bien y en hacerlo rápidamente. Para ser programador ético, hace falta distribuirlos respetando la libertad de sus usuarios. **Y a los programadores que hasta hoy en día no se animaron a liberar sus códigos y a desarrollar Software Libre ¿qué les dirías para que se animen?** Ah, no se. Porque depende de cuáles son sus valores. Pero lo que digo generalmente, es que un programa ofrecido sin libertad es una trampa y aceptarlo es ser tonto. Y crear trampas para meter a la gente en la trampa, es un abuso. No se debe.

Y hablando de trampas y demás: Open Source es una contrapartida a lo que es el movimiento del Software Libre. No tienen como principio la libertad sino que se refieren más al código fuente pero no hacen hincapié en lo que socialmente significa el Software Libre... ese término fue inventado como una manera de hablar del Software Libre, de los mismos programas, pero sin plantearlo como un asunto ético (…) si sabiendo esto, miras lo q' dicen, verás que siempre evitan decir "El Software no libre es injusto". Este es el punto, que no quieren decirlo y nunca lo dicen. **Y también sucede que muchas compañías desarrolladoras de Software, utilizan el término Open Source para referirse a que simplemente colocan el código fuente de un programa, a disposición de cualquier usuario, pero sin embargo, no respetan las cuatro libertades que sí propone el Software Libre.**

La definición de Open Source es más o menos equivalente a la de Software Libre, pero el término genera problemas

Eso es un abuso del término Open Source. Porque no siguen la definición que es más o menos equivalente a la de Software Libre. Pero este mal entendido es muy común y empeora el problema que el término causa. **Y con la gente de la Iniciativa Open Source, ¿tienes algún tipo de trato?** Muy poco, muy poco.

*Foto: **Victor Hermida Prada** – CC-BY-2.0*

A los programadores que quieren contribuir con el movimiento,

les sugiero participar en la Ingeniería Inversa

Richard ¿algo que nos quieras decir a todos los programadores, seamos programadores de Software Libre o los que están en la duda? Es muy amplio. Pero a los que quieren contribuir éticamente al movimiento, lo que sugiero es participar en la ingeniería inversa. Es un campo absolutamente importante para nuestro progreso, porque nos encontramos bloqueados por los periféricos cuyo modo de empleo es secreto. Sobre todo, por ejemplo, los aceleradores de vídeo. Hace falta descubrir su modo de empleo para poder escribir Software Libre y entonces, hace falta la ingeniería inversa.

En la Free Software Foundation, uno ¿de qué forma puede aportar para colaborar con ustedes? Hay varios grupos de trabajo. Por ejemplo hay un grupo que contesta preguntas sobre las licencias. Y hay otro grupo que sugiere proyectos de programación a los voluntarios. Hay otros trabajos que también hacemos en los cuales los voluntarios pueden participar. Dice en fsf.org como participar. Otras maneras de contribuir al movimiento es organizar grupos activistas locales y en Argentina hay que luchar contra la distribución de Windows *(NdR: en las Netbooks que otorga el Estado Nacional)* (…)

Cuando hay errores reproducibles en el código, lanzar el depurador es la manera más fiable de detectarlos

¿Quieres un poco de consejo para los programadores? **¡Me encantaría!** Cuando hay un error lanza inmediatamente el depurador, para saber que pasa dentro del programa. No intentes adivinar. Si no es obvio no puedes adivinar. Pero puedes detectar el problema con el depurador. Y la otra cosa que no hay que hacer, es intentar varias maneras de invocar el programa para buscar un patrón de los casos del error, porque no puedes deducir donde está el error, así. La manera fiable de detectar el error, es con el depurador, porque así puedes hacer que el programa pare en algunos puntos para mirar los datos intermediarios y así puedes localizar en el programa y en la ejecución de este caso, dónde sucede un valor erróneo (…) suponiendo que es un error reproducible. Para un error infrecuente, irreproducible, es más difícil siempre, pero cuando es reproducible, siempre es posible de manera, conceptualmente sencilla, localizar el error en una línea de código.

Lisp, es el lenguaje más elegante y poderoso; tiene capacidades ausentes en los otros lenguajes

Muy buen consejo Richard y, aprovecho y te hago una pregunta más: ¿cuál es tu lenguaje de programación, el que más te apasione programar? Lisp. Lisp es el lenguaje más elegante y poderoso. Porque Lisp tiene capacidades que faltan en todos los otros lenguajes. **Por ejemplo ¿cuáles?** Cuando lanzas el sistema de Lisp, hace un bucle de read-eval-print. Read, significa leer una expresión y convertirla en datos. Porque en Lisp, cualquier expresión tiene una representación natural y sencilla como datos. Luego hace eval, es la evaluación de una expresión como datos, para poder producir un resultado que también es datos. Luego print que convierte los datos en representación textual. En otros sistemas, no hay read, no hay eval y no hay print. Porque no hay conversiones generales entre texto y datos y porque un programa no tiene representación natural como datos. Entonces read no tiene sentido y tampoco eval tiene sentido. Entonces, casi todo en Lisp es ausente en otro lenguaje.

"C" no tiene el poder de Lisp pero me gusta bastante

Hay una variante de Lisp que se llama Scheme, puedes aprender uno u otro. Porque tienen en común a la mayoría de las cosas interesantes (…) pero en cuanto a los lenguajes algebraicos, me gusta C... **Y sí, C es como el lenguaje por excelencia, ¿no?** Si, un amigo tenía una chapa que decía "El lenguaje C combina el poder del ensamblador con la comodidad del ensamblador" *(NdR: Richard Stallman me tuvo que explicar el chiste, porque para variar, mi espíritu Sheldon Cooper, me impidió entender el chiste por mis propios medios)*. Pero de todos modos (…) me gusta bastante C. No tiene el poder de Lisp: un programa no tiene forma como datos, en C, pero para compilar algo, si aceptas que no vas a poder cambiar el programa durante su ejecución, entonces me gusta.

Bueno Richard, nuevamente, un millón de gracias. Ha sido un enorme placer hablar contigo. Happy Hacking! **Happy Hacking, Richard!** Bye! **Bye Bye!**

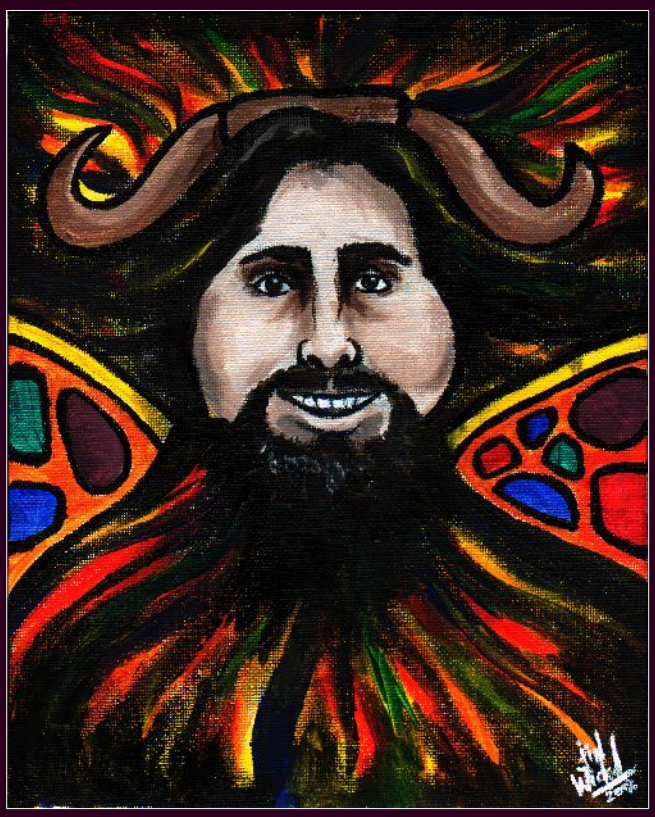


Ilustración creada por ***Jin Wicked*** *con permiso para distribución exacta (sin modificaciones).*
http://www.jinwicked.com/

Información adicional

Richard Stallman, nació en Estados Unidos en el año 1953. Preside la fundación sin ánimo de lucro, **Free Software Foundation.** En el sitio Web de la fundación, **www.fsf.org** puedes encontrar suficiente información para participar activamente en proyectos, grupos de voluntarios o haciendo un donativo.

GNU, es el sistema operativo que utiliza el núcleo creado por Linus Torvalds. La mayor parte de las distribuciones que utilizan este núcleo, están basadas en el sistema operativo GNU en combinación con el núcleo Linux. De allí, que no es correcto decir "distribución Linux", sino **"distribución GNU/Linux"**, puesto que **muchísimos otros sistemas operativos utilizan el mismo kernel (núcleo Linux) y sin embargo, no son libres.** Información completa sobre esto, puedes encontrar en el sitio Web del proyecto GNU, ingresando en **www.gnu.org**.

Richard Stallman, dedica parte de su tiempo, a mantener su **sitio Web personal** www.stallman.org actualizado con noticias, información personal y artículos de opinión propios, no solo referidos al Software Libre, sino también, a temas como el arte y la política.

Guía de seguridad en aplicaciones Web PHP

SEGURIDAD - PHP

Algunos *tips* sobre seguridad en aplicaciones Web bajo PHP, nunca vienen mal. En la primera edición de la revista, hablamos sobre como prevenir inyecciones SQL utilizando mysqli. Si bien cada aplicación "es un mundo", algunas medidas básicas suelen ser más que útiles, inevitables en toda aplicación. Hablaremos de ellas en este artículo.

Escrito por: **Eugenia Bahit** (Arquitecta GLAMP & Agile Coach)

Eugenia es **Arquitecta de Software**, **docente** 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** e integrante del equipo de **Debian Hackers**.

Webs:
Cursos de programación a Distancia: www.cursosdeprogramacionadistancia.com
Agile Coaching: www.eugeniabahit.com

Redes sociales:
Twitter / Identi.ca: **@eugeniabahit**

Si debe tener permisos de escritura, no debe estar servido

Sin dudas, este es uno de los puntos fundamentales a tener en cuenta al momento de desarrollar nuestra aplicación. Frecuentemente los CMS, plataformas de e-Learning e incluso de comercio electrónico -y me refiero a los más populares-, al momento de su instalación, nos piden asignar un directorio con permisos de escritura a fin de poder cargar archivos de imágenes (o cualquier otro contenido estático), mediante su plataforma de administración. Mayormente, es requisito que este directorio se encuentre servido dentro del `DocumentRoot`.

Un directorio Web con permisos de escritura, es una puerta sin llave ni cerrojo en un barrio lujoso: algo tentador para aquellas personas a quiénes les gusta ir de visita a casas ajenas, sin invitación y tomando como propios aquellos objetos de terceros (por si no se entendió, "ladrón", solo que dicho en plan Neruda de supermercado).

El principal inconveniente que se le presenta a algunos programadores al momento de optar por no "servir" un directorio con permisos de escritura, es **¿cómo haré accesibles**

los archivos estáticos mediante el navegador si el directorio no se encuentra servido[31]? La respuesta a esta pregunta es simple: engañando al navegador.

La forma más eficiente, segura, genérica y reutilizable para lograrlo, es hacer que los archivos estáticos que se encuentran en un directorio no accesible mediante el navegador, se sirvan a través de un único archivo PHP encargado de leer y mostrar dichos archivos modificando previamente el *Content-Type* (tipo de contenido) en los encabezados HTTP.

Desde la versión 5.3 de PHP se dispone del módulo **`fileinfo`**[32], quien mediante sus funciones nos ayudará con este proceso. En versiones anteriores, puede utilizarse la **función obsoleta** `mime_content_type()`. Veamos un ejemplo con `fileinfo`.

Supongamos que nuestro *script* de carga de archivos, guarda los mismos en el directo privado `/miswebs/example.com/uploads/` mientras que el `DocumentRoot` se encuentra en `/miswebs/example.com/www/`. Dentro de **`www`**, crearemos nuestro pequeño *script* encargado de servir archivos estáticos: **`showfile.php`**. La tarea de este *script*, consistirá entonces, en:

```
<?php

# Definimos la ruta del directorio privado.
# Este es el directorio con permisos de escritura.
$ruta = '/miswebs/example.com/uploads';

# El nombre del archivo que se desea servir, lo obtendremos
# mediante el parámetro 'file' que luego será pasado por la URI
$file = isset($_GET['file']) ? "$ruta/{$_GET['file']}" : NULL;

# Verificamos el valor de $file
# Si no es NULL, verificamos si el archivo existe antes de proceder
if(!is_null($file)) {
    if(file_exists($file)) {
        # Creamos un recurso fileinfo para obtener el tipo MIME
        $resource = finfo_open(FILEINFO_MIME_TYPE);
        # Obtenemos el tipo MIME
        $mimetype = finfo_file($resource, $file);
        # Cerramos el recurso
        finfo_close($resource);

        # Modificamos los encabezados HTTP
        header("Content-Type: $mimetype");
        # Leemos y mostramos el archivo
        readfile($file);
    }
}

?>
```

Para ver el archivo (como si en realidad estuviese servido), solo bastará con ingresar al

31 Un directorio “servido” se refiere al hecho de poder ingresar en éste, mediante el navegador Web, a través de una URL simple mediante el protocolo HTTP/HTTPS.

32 http://php.net/manual/es/book.fileinfo.php

archivo `showfile.php` pasándole el nombre del estático como parámetro mediante el argumento `file`: **`http://example.com/showfile.php?file=mi_imagen.jpg`**

Vale aclarar que este *script*, servirá para cualquier tipo MIME que pueda ser entendido por el navegador: desde archivos HTML, CSS y JavaScript, hasta imágenes, vídeos, archivos de audio y cualquier otro contenido multimedia.

Si viene por $_GET o $_POST se filtra pero en el usuario nunca se confía

Nada tan conocido, pero nunca está demás recordarlo. No interesa si un dato se guardará o no en una base de datos, en un archivo o nada más se quiere mostrar de forma "volátil" en la pantalla. Eliminar etiquetas HTML y PHP es una buena forma de **evitar los ataques XSS (*Cross Site Scripting*)** y solo lleva menos tiempo que un suspiro:

```
$dato_loco = strip_tags($_GET['maldito_parametro']);
```

Y, convertir caracteres "conflictivos" como comillas dobles y simples, paréntesis angulares y otros caracteres estrafalarios, es solo agregar una función más:

```
$dato_final = htmlentities($dato_loco);
```

Para **validar y sanear datos** (sí, sanear. La palabra "sanitizar" no existe en español), desde la versión 5.2 de PHP, se incorpora la extensión **Filter**[33], la cuál provee funciones de validación y saneamiento, que nos permiten validar si los datos se corresponden a los esperados y limpiarlos, respectivamente.

Esta extensión viene instalada por defecto desde la versión 5.2 de PHP y se puede configurar su aplicación, previamente desde las directivas `filter.default` y `filter.default_flag` del archivo `php.ini`. (Ver más detalles en la Web oficial: http://php.net/manual/es/filter.configuration.php)

Mediante la función **`filter_var($dato, FILTRO[, OPCIONES])`**[34] se puede limpiar y una gran variedad de datos (desde enteros y booleanos hasta correos electrónicos y direcciones IP). Una completa **lista de filtros** posibles, puede encontrarse en http://php.net/manual/es/filter.filters.validate.php.

El saneamiento de los datos también nos provee un gran abanico de filtros que pueden ser consultadas en http://php.net/manual/es/filter.filters.sanitize.php, mientras que las **opciones de filtro y sanación**, pueden obtenerse visitando la URL: http://php.net/manual/es/filter.filters.flags.php

33 http://php.net/manual/es/intro.filter.php
34 http://php.net/manual/es/function.filter-var.php

Vale aclarar que estos filtros y sanadores, son válidos para otras variables superglobales (además de `$_GET` y `$_POST`, como `$_COOKIE` y `$_SERVER`).

Un "Disculpe las molestias" siempre es mejor que un Warning

Nada más agradable para un programador que ver los errores generados por PHP en la pantalla del navegador, como si los logs de Apache y PHP no existieran. Pero eso, no debe hacerse en el servidor de producción.

Los errores de PHP dan demasiada información al usuario. Más información de la que un usuario común puede entender y de la que un usuario "no tan común" debería conocer. ¿Para qué mostrarla entonces?

Los errores pueden ser silenciados con una simple @ delante de una instrucción:

```
@fopen('/home/user/templates/archivo_que_no_existe.html', 'r');
```

Pero mucho más efectivo y cómodo para el desarrollo, es modificar las directivas de `php.ini` en tiempo de ejecución, dependiendo de si se está trabajando en desarrollo u operando en producción:

```
const PRODUCCION = False;  # En producción, se establece en True

if(!PRODUCCION) {
    ini_set('error_reporting', E_ALL | E_NOTICE | E_STRICT);
    ini_set('display_errors', '1');
} else {
    ini_set('display_errors', '0');
}
```

Si es divertido y fascinante para el programador, es una mala idea para el servidor

Sí, es fascinante ponerse a jugar con la función `shell_exec()` y ver como desde un simple *script* Web, puedes hacer cosas como ejecutar locos comandos del sistema operativo. Pero ya. Es un juego... y peligroso. Permitir la ejecución de funciones como `exec()`, `shell_exec()`, `popen()`, `system()` y "compañía" es un claro síntoma del *síndrome del Kamikaze* (sí, es un síndrome que acabo de inventar). Se puede -y debe- evitar que dichas funciones sean ejecutadas, mediante la directiva **`disable_functions`** del archivo `php.ini`.

Una buena configuración de esta directiva, podría ser la siguiente:

```
disable_functions = proc_open, popen, disk_free_space, diskfreespace,
set_time_limit, leak, tmpfile, exec, system, shell_exec, passthru,
show_source, system, phpinfo, pcntl_alarm, pcntl_fork, pcntl_waitpid,
pcntl_wait, pcntl_wifexited, pcntl_wifstopped, pcntl_wifsignaled,
pcntl_wexitstatus, pcntl_wtermsig, pcntl_wstopsig, pcntl_signal,
pcntl_signal_dispatch, pcntl_get_last_error, pcntl_strerror,
```

```
pcntl_sigprocmask, pcntl_sigwaitinfo, pcntl_sigtimedwait, pcntl_exec,
pcntl_getpriority, pcntl_setpriority
```

Si lo recibes desde un formulario, no hagas nada hasta no estar seguro

Te matas filtrando datos con JQuery y haciéndole miserable la vida a los usuarios que no entienden que todos los malditos campos del maldito formulario son obligatorios. Y ni te cuento si esto mismo lo venías haciendo desde la época en la que para *John Resig*, los Pitufos eran más importantes e imprescindibles que un ordenador y, todo lo que hoy hace JQery en una sola instrucción, a ti te demandaba unas 744 mil 288 coma 73 líneas de código (no es que yo tenga años para repartir, solo me lo han contado...).

JavaScript por aquí, alertas por allá, bonitos *layers* que se despliegan lentamente con unos mágicos *fadeIn*, *fadeOut* y *fadeWTF*. Todo muy lindo, muy mágico, muy *Alicia en el País de las Maravillas*, pero basta con un simple clic derecho > inspeccionar elemento > editar... para que tu mágico arte se diluya cual chocolate en leche hirviendo (perdón, me apetece una rica *chocolatada*) y te envíen el formulario como se les de la maldita gana. Y peor aún cuando el mismo formulario se copia, edita y sube en un servidor que no es el tuyo o más frustrante aún, ni siquiera se sube a un servidor y se le hace un simple "doble click" al maldito HTML editado.

Los datos importantes, se filtran y validan desde PHP, no desde JavaScript. JavaScript se ejecuta en el cliente y éste, es quien tiene todo el control sobre aquel. Y nunca está demás, verificar desde la URL de la cual proviene dicho formulario:

```
$uri_donde_muestras_tu_form = 'http://example.com/index.php?page=contacto';

if(isset($_SERVER['HTTP_REFERER'])) {
    if($_SERVER['HTTP_REFERER'] != $uri_donde_muestras_tu_form) {
        // El form no se está enviando desde el archivo que creías
    }
}
```

En caso de tener accesible el formulario en más de una URI, bastará con definir las URI permitidas en un *array* (una **lista blanca**) y validar el *referer* con la función `in_array()`:

```
$form_uris = array(
   'http://example.org/contacto.php',
   'http://www.example.org/contacto.php',
   'https://example.org:8000/contacto.php'
   'http://otrodominio.com/contacto'
);

if(isset($_SERVER['HTTP_REFERER'])) {
    if(!in_array($_SERVER['HTTP_REFERER'], $form_uris)) {
        // El form no se está enviando desde el archivo que creías
    }
}
```

Vale aclarar que las listas blancas son también útiles cuando se requieren llamadas de retorno (callback) dinámicas:

```
$funcion = $_GET['pagina'];
$allowed_functions = array('mostrar_form', 'enviar_form');

if(in_array($funcion, $allowed_functions)) {
   # función permitida
   call_user_func($funcion);
}
```

Otra forma de mantener a salvo los formularios, es cerciorándote de que que los datos recibidos (nombres de tus campos de formulario) son los esperados. Si se recibe algún campo extra o menos cantidad de los esperados, algo no huele bien.

```
$campos_esperados = array('nombre', 'apellido', 'edad');

foreach($campos_esperados as $campo) {
    # Obtenemos una variable con el mismo nombre de campo
    $$campo = isset($_POST[$campo]) ? $_POST[$campo] : NULL;
}
```

También puedes aprovechar el paso anterior, para "matar dos pájaros de un tiro", validar y sanear los datos con `filter_var()`.

Si es una contraseña, se hashea. Si no está hasheado, no es una contraseña.

Nunca guardes ni tus contraseñas ni las de tus usuarios en texto plano. Las contraseñas deben guardarse *hasheadas*, vayan o no a estar almacenadas en una base de datos, pues **una contraseña debe ser irrecuperable**.

Guardar contraseñas *hasheadas* es tan simple como recurrir, por ejemplo, a la función `md5()`:

```
$clave = md5($_POST['clave']);
```

Luego, para comparar la contraseña del usuario con la almacenada, simplemente se vuelve a *hashear* el *input* de la misma forma que se hizo para guardarlo.

En un próximo artículo, intentaremos centrar el foco en la prevención de ataques CSRF (Cross Site Request Fogery) y la captura de sesiones implementando identificadores de sesión secretos (tokens) y regenerando identificaciones únicas de sesión (session ID) de forma más profunda.

Ahora **¡a asegurar tu aplicación!**

Patrones y anti-patrones de diseño ¿Para que sirven?

PHP

Cuando uno desarrolla POO suele encontrarse mucho con las palabras "patrones" y "anti-patrones" de diseño, pero ¿qué significan en realidad estas palabras?, ¿para qué me sirven?, ¿tiene algo que ver con el patrón de la Hacienda San José[35]?, definitivamente no tiene nada que ver con el patrón de ninguna hacienda. Los patrones de diseño nacieron para solucionar los problemas comunes en el diseño de software y por supuesto, dichos patrones no son netamente de un lenguaje. En este artículo hablaremos un poco de historia y específicamente sobre los patrones de diseño aplicados en PHP.

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**

35 Gran Hacienda ubicada en Chincha - Peru

Un poco de historia: corría el año 1995, cuando un grupo de gurúes llamado "Gang of four - GoF" compuesto por Erich Gamma, Richard Helm, Ralph Johnson y John Vlisides, tuvieron la genial idea de lanzar el primer libro de patrones de diseño Head First: Patterns Design de O'Reilly. Fueron 23 patrones agrupados en 3 categorías. Por supuesto, no era la primera vez que se usaba la palabra patrones, ésta ya había sido acuñada un par de décadas antes por el Arquitecto Christopher Alexander en un intento de formalizar y plasmar de forma práctica, generaciones de conocimiento arquitectónico. Esto mismo se dio con los patrones de diseño de software.

¿Que significan en realidad estas palabras?

Patrones: son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces.

Anti-patrones: es un patrón de diseño que invariablemente conduce a una mala solución para un problema.

> *"El lenguaje de patrones brinda a todo el que lo utilice el poder de crear una infinita variedad de construcciones nuevas y únicas, de la misma forma que su lenguaje común le brinda el poder de crear una infinita variedad de oraciones."*
> *Christopher Alexander*

¿Para que me sirven?

- Los patrones de diseño nos dan una visión holística del tratamiento de los problemas.
- Estandariza la forma en que se diseña.
- Evita desperdiciar el tiempo en problemas que ya cuentan con soluciones probadas.

Para ser considerado un patrón, tiene que cubrir -al menos- los siguientes requisitos:

- **Efectivo:** Se debe comprobar su éxito resolviendo problemas anteriores.
- **Reusable:** Poder aplicarlo a otros problemas similares a las circunstancias definidas por el patrón.

Los patrones de diseño se dividen básicamente en 3 categorías:

1. **Creacional:** Cómo se crean instancias de objetos.
2. **Estructural:** Cómo se relacionan y combinan las clases para dar una estructura más compleja.

3. **De comportamiento:** Cómo se comunican los objetos, cooperan y distribuyen las responsabilidades para lograr sus objetivos.

Estructura de un patrón

Para describir un patrón se usan plantillas más o menos estandarizadas, de forma que se expresen uniformemente y puedan constituir de forma efectiva un medio de comunicación uniforme entre diseñadores. Varios autores eminentes en esta área han propuesto plantillas ligeramente distintas, aunque la mayoría de ellas, definen los mismos conceptos básicos.

La plantilla más común es la utilizada precisamente por el GoF y consta de los siguientes apartados:

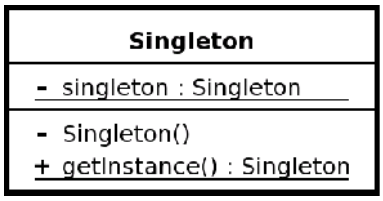
- **Nombre del patrón:** nombre estándar del patrón por el cual será reconocido en la comunidad (normalmente se expresan en inglés).
- **Clasificación del patrón:** creacional, estructural o de comportamiento.
- **Intención:** ¿Qué problema pretende resolver el patrón?
- **También conocido como:** Otros nombres de uso común para el patrón.
- **Motivación:** Escenario de ejemplo para la aplicación del patrón.
- **Aplicabilidad:** Usos comunes y criterios de aplicabilidad del patrón.
- **Estructura:** Diagramas de clases oportunos para describir las clases que intervienen en el patrón.
- **Participantes:** Enumeración y descripción de las entidades abstractas (y sus roles) que participan en el patrón.
- **Colaboraciones:** Explicación de las interrelaciones que se dan entre los participantes.
- **Consecuencias:** Consecuencias positivas y negativas en el diseño, derivadas de la aplicación del patrón.
- **Implementación:** Técnicas o comentarios oportunos de cara a la implementación del patrón.
- **Código de ejemplo:** Código fuente de ejemplo para la implementación del patrón.
- **Usos conocidos:** Ejemplos de sistemas reales que usan el patrón.
- **Patrones relacionados:** Referencias cruzadas con otros patrones.

> *Recuerden que un buen patrón de diseño debe poder aplicarse en la mayoría -sino en todos- los lenguajes, sólo limitado por las capacidades de éstos.*

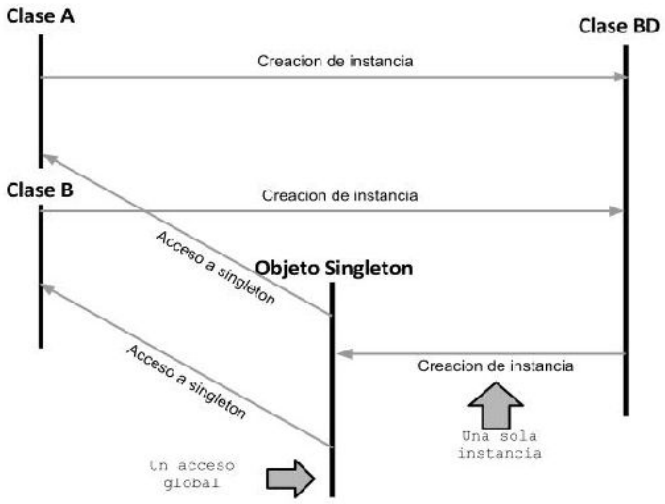
Ahora veremos algunos patrones aplicados en PHP. Vale aclarar que esta, es una selección a criterio personal.

Singleton

El patrón de diseño Singleton es un patrón de diseño creacional que se asegura de tener una sola instancia de una clase particular durante su tiempo de ejecución y, proporciona un punto de acceso global a ella.



Veamos el siguiente gráfico:



Como podemos ver en la imagen, cualquier clase puede instanciar a la clase BD, sin embargo, ésta solo creará una vez dicha instancia. En las demás llamadas a crear, se les asignará el objeto singleton devolverá el acceso, o mejor dicho: “no me importa cuantas veces me llames yo solo respondo una vez y lo que digo no cambia nunca” (si lo se, es caprichoso singleton).

Algunas características:

- El constructor de la clase debe ser privado (private).
- Para almacenar nuestra instancia debemos crear una propiedad estática privada.
- Esta data será persistente y se mantendrá a lo largo de todas las llamadas getInstance.

Ejemplo básico de como implementarlo:

```
class Contador
{
   private static $instancia;
   private $contador;

   //Constructor privado evitando que la clase sea instanciada directamente
   private function __construct()
   {
      echo "Se ha creado un " . __CLASS__ . "\n";
      $this->contador = 0;
   }

   // Método encargado de instanciar la clase
   public static function getInstance()
   {
      if (  !self::$instancia instanceof self)
      {
         self::$instancia = new self;
      }
      return self::$instancia;
   }

   public function incrementar()
   {
      return ++$this->contador;
   }

   public function disminuir()
   {
      return --$this->contador;
   }
}
```

Como vemos, `getInstance()`, es el que instancia la clase, al no encontrarla crea la instancia y la almacena en `$instancia`, si ésta ya existe retoma la instancia.

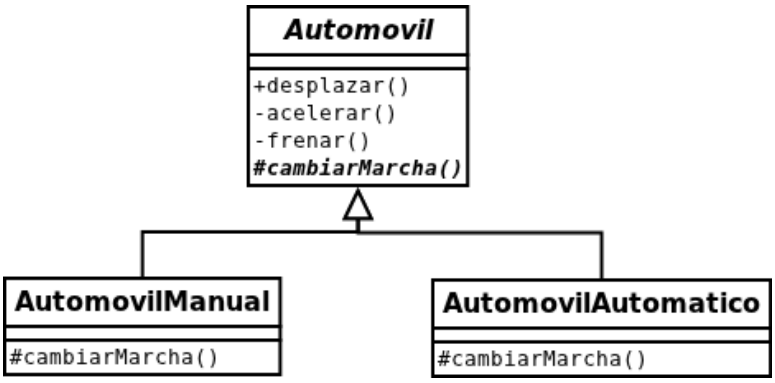
Lectura Recomendada: Patrón Singleton y herencia en PHP[36]

Factory

Es un patrón de diseño creacional que sirve para instanciar múltiples objetos de similares características, está basado en la instancia de objetos en tiempo de ejecución.

Personalmente creo que es uno de los patrones más importantes dado que ayuda a la centralizar muchas clases con características similares, en pocas palabras “pasa por mi y yo te digo a donde debes de ir (yo mando)”.

36 http://juanjofr.com/blog/patron-singleton-y-herencia-en-php/



El principal objetivo de este patrón es encapsular el procedimiento creacional que diferentes clases pueden tener, en una sóla función, como lo veremos en el siguiente ejemplo con varias clases para conectarse a diferentes base de datos, sin importar donde desee conectarme concentraré la instancia en una clase llamada `BaseDatos`.

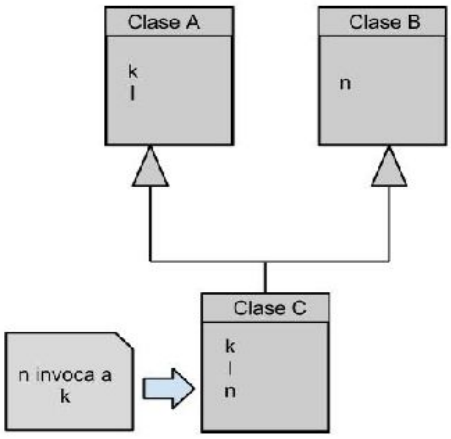
```
class MySQL{
  public function __construct(){
    print_r("Instancio la BD - MySQL");
  }
}

class PostgreSQL{
  public function __construct(){
    print_r("Instancio la BD - PostgreSQL");
  }
}

abstract class BaseDatos{
    public static function load($bd){
        try {
            if (class_exists($bd)) {
                return new $bd;
            } else {
                throw new Exception("No existe la clase “.$bd);
            }
        } catch (Exception $e) {
            echo 'Excepción: ',  $e->getMessage(), "\n";
        }
    }
}
```

Adapter

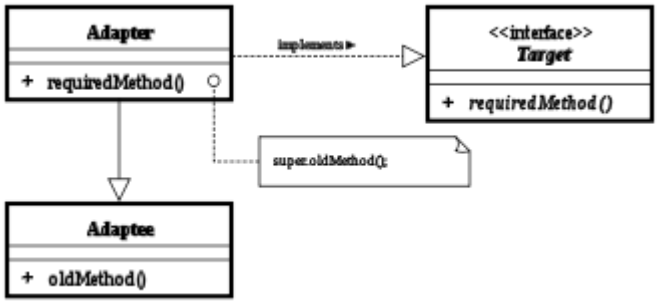
Es un patrón de diseño estructural que nos permite reutilizar una clase con una interfaz diferente, por tanto las clases trabajan juntas, lo que de otra manera no podría hacerlo debido a sus interfaces incompatibles.

Se utiliza para evitar cambiar el código cuando cambia una interfaz, o para futuras modificaciones o implementaciones cuando diseñas clases generales.

También conocido como Wrapper (Envoltorio).



La clase A contiene un método k que se quiere incluir como una implementación más del método n, declarado en la clase abstracta pura, B. Entonces, se diseña una clase C que herede de las clases A y B, el método k y la declaración del método n respectivamente. Como ambos elementos se han juntado en la clase C, basta con definir que la implementación del método n consiste en invocar al método k para conseguir que k sea una n más. Esta última técnica se denomina envolver.



Ejemplo: Nos permite envolver acciones en una clase de tal modo que estas puedan ser usadas en el momento adecuado.

```
$account_domain = new Account();
$account_domain->NewAccount( //inputs );

class Account()
{
    public function NewAccount( //inputs )
    {
        $user = new User();
        $user->CreateOrUpdate( //subset of inputs );

        $profile = new Profile();
        $profile->CreateOrUpdate( //subset of inputs );
    }
}
```

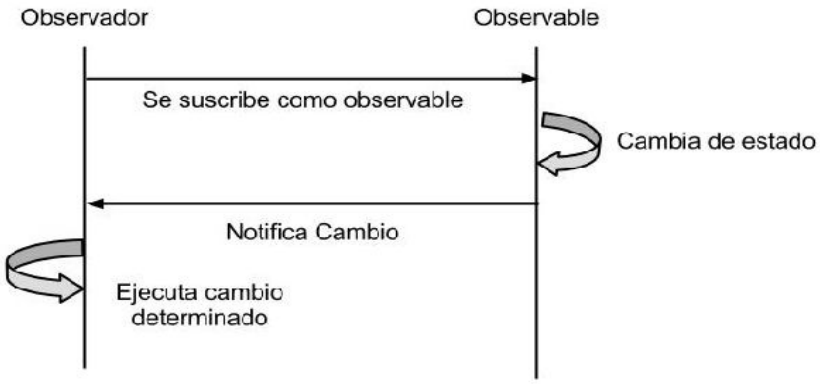
Esto también nos permite cambiar algunos campos que están siendo recibidos de la

clase cliente, convirtiéndolos en algo compatible con las funciones adaptadas.

Observer

El patrón Observer, no es más que un manejador y despachador de eventos, en el cual uno o varios "observadores" se suscriben a un "sujeto", el cual les "notifica" cuando un evento es llevado a cabo, de tal forma que dichos observadores puedan realizar una determinada operación.

Veámoslo en el siguiente gráfico.



Para mayor entendimiento recomiendo esta lectura Patrón de diseño Observer en PHP[37]

Ya hemos aprendido uno cuantos patrones, nos quedan muchos más, pero ahora, cuando uno aprende sobre patrones, suele pasar que el entusiasmo por implementarlo nos desborda, es así que empieza la búsqueda implacable de ¿donde implementarlos?, ahí podemos caer en muchos errores porque como mencioné anteriormente, los patrones son buenos pero hay que tener criterio.

> *"El aprendizaje más importante proviene de la experiencia directa."*
> *Nonaka & Takehuchi*

Anti-patrones

¿Y como no sentirse identificado con esto?, todos hemos pasado por ello, es un mal del

37 http://arturoweb.wordpress.com/2010/03/08/patron-de-diseno-observer-en-php/

que todo programador a padecido en algún momento, pero tranquilos: siempre se puede cambiar -eso espero-. Como habíamos definido anteriormente, estas son las malas prácticas que todo buen programador debe evitar.

El libro Anti-Patterns (de William Brown, Raphael Malveau, Skip McCormick y Tom Mowbray, junto con la más reciente incorporación de Scott Thomas) describe los antipatrones como "la contrapartida natural al estudio de los patrones de diseño". El estudio formal de errores que se repiten, permite reconocer y reconducir los elementos involucrados hacia una mejor solución.

Los antipatrones de diseño se clasifican en tres grandes grupos, debido a que son 3, los actores principales en todos los proyectos de desarrollo de software.

- Desarrollo de Software
- Arquitectura de Software
- Gestión de Proyectos de Software

Cada anti-patron cuenta con un nombre que curiosamente, la mayoría de veces, da risa, y la forma de explicarlo suele ser muy clara y sencilla. Tengo que admitir que me río cada vez que las leo y es que como dice el dicho -recordar es volver a vivir-.

La lista de anti-patrones es muy extensa. Aquí solo expondré algunos por categoría:

Desarrollo de software

- **The Blob:** Objeto todopoderoso.

  Este es un mal común puesto que se suele centralizar todos los métodos en una misma clase, esta clase se termina volviendo indispensable, lo que es peor aún hacer un cambio es demasiado engorroso, no reutilizable y definitivamente no es POO.

- **Lava Flow:** Código muerto e información de diseño olvidada permanecen congelados en un diseño cambiante. Esto es análogo a un flujo de lava en el que se van endureciendo pedazos de roca conforme avanzan.

- **Golden Hammer:** Asumir que nuestra solución favorita es universalmente aplicable, haciendo bueno el refrán "a un martillo, todo son clavos".

  Esta relacionado con aferrarse a un paradigma, por ejemplo, algunos programadores solo programan en PHP o Java, así que todo lo quieren hacer con ese lenguaje, sin entender que cada lenguaje tiene sus limitaciones.

- **Spaghetti Code:** Da referencia a código mal estructurado.

  Por ejemplo es muy conocido es cuando encontramos en un mismo archivo código PHP y HTML sazonado con Javascript: es de temer esta mezcla.

Arquitectura de Software

- **Reinvent the wheel:** Se refiere a reimplementar componentes que se pueden conseguir prefabricados de antemano y hacer poco reuso en el código.
- **Copy and paste programming:** Programar copiando y modificando código existente en lugar de crear soluciones genéricas.
- **Improbability factor:** Asumir que es improbable que un error conocido cause verdaderos problemas.

Gestión de Proyectos de Software

- **The Mythical Month Man:** Consiste en la creencia de que asignar más personal a un proyecto, acotará el tiempo de entrega. Normalmente para corregir retraso del proyecto. Llega un punto donde entre más personal se asigne, más se retrasa el proyecto.
- **Corncob:** Es habitual que en el desarrollo de un proyecto software, ciertas personas dificulten su desarrollo. Se ha calculado que al menos la mitad del tiempo en un proceso de desarrollo de software se invierte en la comunicación entre personas.

Espero que este artículo les sirva de base para un mejor entendimiento de patrones y anti-patrones.

Referencias:

http://msdn.microsoft.com/es-es/library/bb972242.aspx
http://es.wikipedia.org/wiki/Patr%C3%B3n_de_dise%C3%B1°

Tip by Doña Giberish:

Cuantas veces hemos querido probar algo nuevo, ¿qué me dicen de node.js, Django o Ruby on Rails? si su caso es como el mío les recomiendo ir a Turnkey: en esta página encontraras los .iso de servidores configurados con diferentes aplicaciones listos para correr en tu virtualbox, por supuesto esto es para que juegues con todo ese conocimiento y si deseas subir a producción tienes que tunear tu propio server.

Experimentando con matplotlib y otros "yuyos" ...

PYTHON

En estas líneas veremos rápidamente como manejar imágenes y *plotear* -de alguna forma sencilla (y *naive*)- superficies simples en 3D.

Escrito por: **Celia Cintas** (Licenciada en Informática)

Licenciada en Informática (UNPSJB), actualmente realizando **Doctorado en Ingeniería** (Procesamiento de Imágenes - UNS), Docente (UNPSJB), Intento de sysadmin (CC) y code monkey el resto de las horas :). Pythonera por defecto con alarmantes inclinaciones hacia Prolog y otras herejías.

Webs:
Blog: http://yetanotherlog.wordpress.com/

Redes sociales:
Twitter / Identi.ca: **@RTFMcelia**

En esta ocasión usaremos imágenes simples para tratar de darle altura (en el eje z) desde nuestro ploteo. ¿Para qué nos puede servir esto? Si estamos tratando de distinguir distancias quizás en un gráfico 3D tendremos mejor apreciación que una imagen en 2D.

Encendiendo Motores ...

Sabiendo que ya tienen python corriendo en sus máquinas, solo necesitaremos:

- numpy (numpy.org)
- matplotlib (matplotlib.org) y
- PIL (www.pythonware.com/products/pil)

Todos nuestros ingredientes están a un ***pip***[38] de distancia.

Imágenes desde Python

Para manipular imágenes desde Python tenemos varias bibliotecas. Entre las más populares se encuentran OpenCV[39], SimpleCV[40], scikit-image[41], PIL[42] o simplemente trabajarlas como arreglos de numpy, las cuales quedarán pendientes para próximos artículos. En este caso al ser simple y llano lo que deseamos hacer, trabajaremos solamente con PIL.

```
from PIL import Image
```

Y para abrir una imagen simplemente debemos escribir:

```
im = Image.open(file_name).convert("L")
```

La función *convert* con parámetro L nos retorna nuestra imagen en escala de grises, por lo que los diferentes grises nos darán la profundidad de nuestra imagen.

Ahora tenemos nuestra imagen en escala de grises pero… ¿cómo plotearla con *matplotlib*? Primero que nada debemos llevar nuestra imagen a un arreglo de *numpy*, estructura que *matplotlib* entiende muy bien.

```
import numpy as np

array_im = np.asarray(im)
```

Si observan la variable *array_im* verán algo asi:

```
[[  19.   19.   19. ...,  188.  188.  188.]
 [  19.   19.   19. ...,  182.  188.  188.]
 [  19.   27.   24. ...,  188.  182.  188.]
 ...,
 [  19.   19.   19. ...,  183.  187.  188.]
 [  19.   24.   19. ...,  188.  183.  187.]
 [   0.    0.    0. ...,  208.  208.  208.]]
```

Creando nuestro mesh 3D

En esta parte veremos los pasos necesarios para graficar funciones en 3D. Vamos por partes -*como diría Jack*-. Para plotear esta clase de gráficos debemos importar los siguientes módulos:

```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
```

38 Pip es una herramienta para administración de paquetes para python
39 http://opencv.willowgarage.com/documentation/python/index.html
40 http://www.simplecv.org/
41 http://scikit-image.org/
42 http://www.pythonware.com/products/pil/

```
import matplotlib.cm as cm
```

Cm nos permite determinar el *colormap* de nuestro gráfico. Habiendo importado nuestros módulos ahora podemos construir los valores que deseamos dibujar.

Teniendo nuestra imagen en un arreglo, en el cual sus valores representarán nuestro *z*, debemos generar los valores correspondientes a *x, y* que será la posición de cada píxel de nuestra imagen. Esto se logra de la siguiente manera:

```
fig = plt.figure()
ax = fig.add_subplot(121, projection='3d')
Z = array_im
x = np.arange(0, im.size[0])
y = np.arange(0, im.size[1])
X, Y = np.meshgrid(x, y)
```

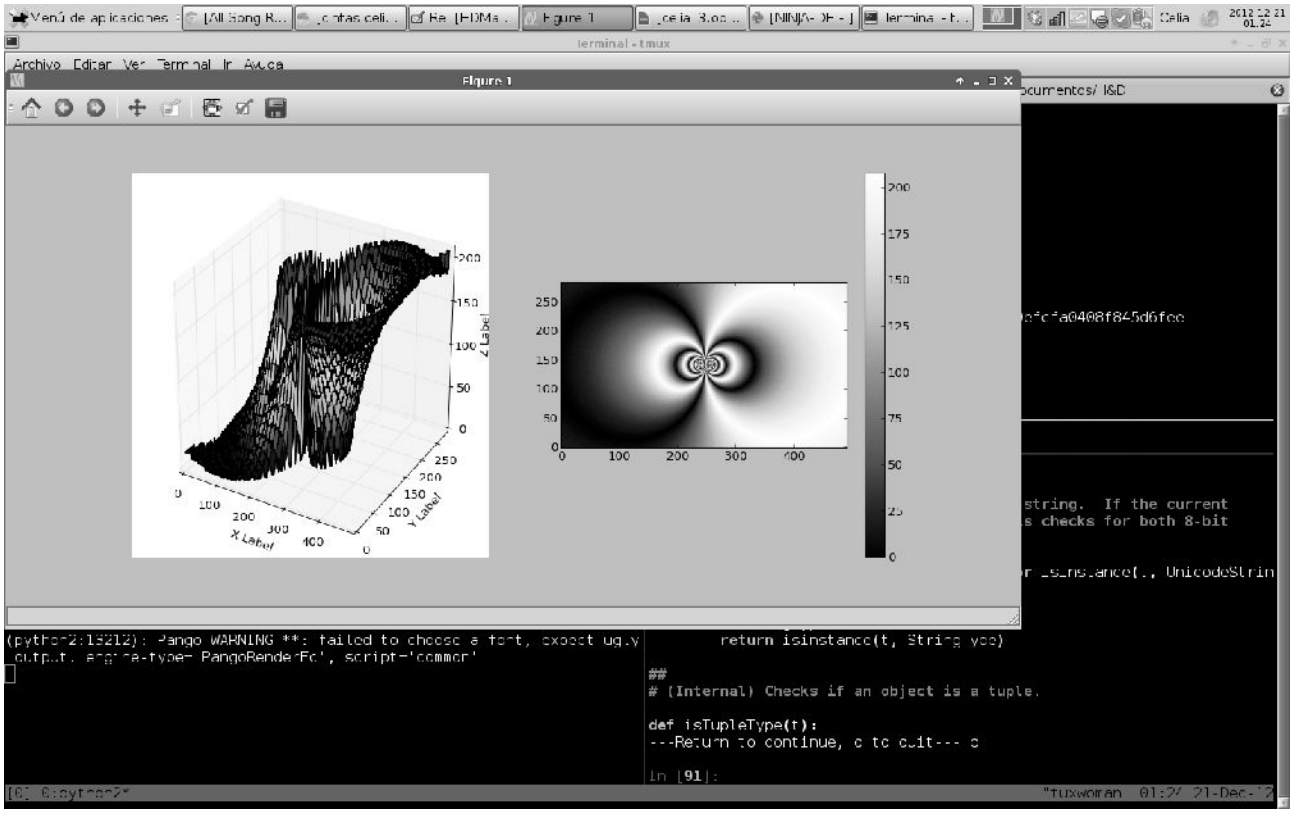
im.size nos da el ancho y alto de nuestra imagen, por lo que *x* e *y* irán desde la posición 0 a sus respectivos ancho (im.size[0]) y alto(im.size[1]).

Y luego de que contamos con nuestros x, y, z dibujamos la superficie, el parámetro *cmap* simplemente nos da distintos tipos de configuración de colores para nuestro plot.

```
ax.plot_surface(X, Y, Z, cmap=cm.jet)
```

Ultimando detalles...

En nuestro experimento mostramos la imagen al lado del plot generado como se ve a continuación:



para mostrar la imagen y la barra de colores (bondades de *matplotlib*) se debe teclear:

```
imaplot = plt.imshow(im, origin='lower', cmap=cm.Greys_r)
bar = plt.colorbar(imaplot)
```

Para ver el código completo del experimento pueden ingresar en

https://github.com/celiacintas/Frankepy/blob/master/surface/surfaceIm.py

Referencias

1. PIL - http://www.pythonware.com/products/pil/
2. Matplotlib - http://matplotlib.org/examples/

Tip by Miss Cuelgue:

Al utilizar *imshow* de pyplot recordad colocar *origin='lower'* si desean mostrar una imagen sin pasarla previamente a un arreglo de *numpy,* de otra forma les mostrará la imagen dada vuelta.

Manual de MVC: (2) Vistas dinámicas y Templates

MANUAL DE MVC

Tanto en Python como en PHP, es posible no embeber diferentes lenguajes en las plantillas. Incluso, cuando se trate de resultados traídos de una base de datos, que deban ser sustituidos de forma iterativa. En la entrega de hoy, aprenderemos las técnicas que nos convertirán en "Magos del MVC".

Escrito por: **Eugenia Bahit** (Arquitecta GLAMP & Agile Coach)

Eugenia es **Arquitecta de Software**, **docente** 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** e integrante del equipo de **Debian Hackers**.

Webs:
Cursos de programación a Distancia: www.cursosdeprogramacionadistancia.com
Agile Coaching: www.eugeniabahit.com

Redes sociales:
Twitter / Identi.ca: **@eugeniabahit**

Creo que la consulta más frecuente que he recibido desde que publiqué mi libro *«POO y MVC en PHP»*, ha sido "cómo evitar embeber código de programación en el HTML para completar una tabla o lista de selección (`select`) con los resultados de una consulta a base de datos". Y créanme: es sumamente sencillo.

Primero, es necesario saber, que existen tres tipos de sustituciones que pueden efectuarse en las vistas:

1. **Sustituciones estáticas:** es el caso de una plantilla HTML en la que se deben sustituir ciertos datos de forma estática. Es decir, plantillas HTML en las que cada indicador de sustitución, debe ser reemplazado por un valor concreto.
2. **Sustituciones dinámicas:** son aquellas plantillas HTML en las cuáles un mismo indicador de sustitución, debe ser reemplazado de forma iterativa por más de un dato. Por ejemplo, una lista de selección que deba reemplazarse con los resultados devueltos por una consulta de selección múltiple a base de datos.

3. **Sustituciones combinadas:** son aquellas plantillas HTML en las que se debe realizar tanto una sustitución estática como una (o más) sustituciones dinámicas.

Pasemos lo anterior a ejemplos prácticos:

Plantilla de sustitución estática:

```
<!doctype html>
<html lang="es">
    <head>
        <charset="utf-8">
        <title>[INDICADOR_DE_SUSTITUCION: TITULO]</title>
    </head>

    <body>
        <header>
            <h1>[INDICADOR_DE_SUSTITUCION: TITULO]</h1>
        </header>

        <article>
            [INDICADOR_DE_SUSTITUCION: CONTENIDO]
        </article>
    </body>
</html>
```

Plantilla de sustitución dinámica:

```
<!doctype html>
<html lang="es">
    <head>
        <charset="utf-8">
        <title>Listado telefónico</title>
    </head>

    <body>
        <header>
            <h1>Listado telefónico</h1>
        </header>
        <table>
            <tr>
                <th>Nombre</th>
                <th>Apellido</th>
                <th>Teléfono</th>
            </tr>
            <!--SUSTITUCION-DINAMICA-->
            <tr>
                <td>[INDICADOR_DE_SUSTITUCION: NOMBRE]</td>
                <td>[INDICADOR_DE_SUSTITUCION: APELLIDO]</td>
                <td>[INDICADOR_DE_SUSTITUCION: TELÉFONO]</td>
            </tr>
            <!--SUSTITUCION-DINAMICA-->
        </table>
    </body>
</html>
```

Plantilla de sustitución combinada:

```
<!doctype html>
<html lang="es">
    <head>
        <charset="utf-8">
        <title>[INDICADOR_DE_SUSTITUCION: TITULO]</title>
    </head>

    <body>
        <header>
            <h1>[INDICADOR_DE_SUSTITUCION: TITULO]</h1>
        </header>

        <article>
            [INDICADOR_DE_SUSTITUCION: CONTENIDO]
        </article>

        <h3>Listado telefónico</h3>
        <table>
            <tr>
                <th>Nombre</th>
                <th>Apellido</th>
                <th>Teléfono</th>
            </tr>
            <!--SUSTITUCION-DINAMICA-->
            <tr>
                <td>[INDICADOR_DE_SUSTITUCION: NOMBRE]</td>
                <td>[INDICADOR_DE_SUSTITUCION: APELLIDO]</td>
                <td>[INDICADOR_DE_SUSTITUCION: TELÉFONO]</td>
            </tr>
            <!--SUSTITUCION-DINAMICA-->
        </table>
    </body>
</html>
```

Indicadores de sustitución en las GUI

Los indicadores de sustitución, son marcas de texto plano que se colocan en los archivos HTML para identificar las zonas de la plantilla, en las cuáles deben reemplazarse ciertos datos.

En **Python**, los identificadores de sustitución se expresan con **palabras concatenadas antecedidas del signo dólar ($)**:

```
<!doctype html>
<html lang="es">
    <head>
        <charset="utf-8">
        <title>$titulo</title>
    </head>

    <body>
        <header>
```

```
            <h1>$titulo</h1>
        </header>

        <article>
            $contenido
        </article>
    </body>
</html>
```

Mientras que en **PHP**, no existe ninguna regla al respecto, quedando al libre albedrío del diseñador, la elección de estilo de los identificadores de sustitución. Un clásico en los identificadores de sustitución en PHP, es **encerrar las palabras identificadoras entre dos llaves, de apertura y cierre respectivamente {}**:

```
<!doctype html>
<html lang="es">
    <head>
        <charset="utf-8">
        <title>{titulo}</title>
    </head>

    <body>
        <header>
            <h1>{titulo}</h1>
        </header>

        <article>
            {contenido}
        </article>
    </body>
</html>
```

Lógica de sustitución en MVC

La lógica de las vistas en MVC, es única y exclusiva para cada GUI en particular. No obstante, existe un procedimiento estándar, el cual consta de tres pasos:

1. Obtener el HTML (**GUI**);
2. Crear un **diccionario** con los identificadores de sustitución y los datos asociados a su reemplazo;
3. Realizar la **sustitución** de datos sobre la plantilla HTML e imprimirla en pantalla (es decir, mostrar el resultado al usuario).

Obtener la GUI en PHP:

```
$plantilla = file_get_contents('/ruta/a/template.html');
```

Obtener la GUI en Python:

```
with open('/ruta/a/template.html', 'r') as archivo:
```

```
    plantilla = archivo.read()
```

Crear un diccionario de sustituciones en PHP:

```
$diccionario = array(
    '{titulo}'=>'Página de muestra',
    '{contenido}'=>'Hola Mundo'
);
```

Crear un diccionario de sustituciones en Python:

```
diccionario = dict(
    titulo='Página de prueba',
    contenido='Hola Mundo'
)
```

> *Por favor, notar que en Python, el nombre de las claves del diccionario es el identificador de sustitución SIN el signo dólar, mientras que en PHP, las claves del diccionario deberán verse exactamente igual a los identificadores de sustitución utilizados en la plantilla HTML.*

Sustitución de diccionarios en PHP:

```
$render = str_replace(array_keys($diccionario),
    array_values($diccionario), $plantilla);
```

Sustitución de diccionarios en Python:

```
from string import Template
render = Template(plantilla).safe_substitute(diccionario)
```

Sustituciones estáticas

Tanto en Python como en PHP, las sustituciones estáticas se realizan siguiendo los tres pasos estándar mencionados en el punto anterior: traer la plantilla, crear el diccionario y realizar la sustitución.

Si los datos sustitutos debieran traerse desde una consulta a base de datos, primero se realizará la consulta SQL, luego se almacenarán los datos retornados en variables y, finalmente, dichas variables se asignarán como valores de las claves del diccionario, sin más complejidad que ésta.

Sustituciones dinámicas

En las sustituciones dinámicas, la mayor complejidad radica en la obtención del código HTML sobre el cual la sustitución debe realizarse.

La plantilla HTML se deberá obtener de forma estándar, mientras que de ella, antes de proceder con los dos pasos siguientes, se deberá recuperar sólo la fracción de código HTML sobre la cuál realizar los reemplazos. Dicha fracción de código, se obtendrá definiendo previamente, la siguiente **expresión regular**:

```
<!--NOMBRE-DE-LA-SUSTITUCION-DINAMICA-->(.|\n){1,}<!--NOMBRE-DE-LA-SUSTITUCION-DINAMICA-->
```

Dónde `NOMBRE-DE-LA-SUSTITUCION-DINAMICA` será el texto utilizado en el código HTML dentro del comentario identificador:

Dado el siguiente identificador de sustitución dinámica:

```
<!--TELEFONOS-->
    ... código html a iterar + indicadores de sustitución
<!--TELEFONOS-->
```

La expresión regular debería verse como la siguiente:

```
<!--TELEFONOS-->(.|\n){1,}<!--TELEFONOS-->
```

Obtención del código HTML iterativo en PHP:

```
$regex = "/<!--TELEFONOS-->(.|\n){1,}<!--TELEFONOS-->/";
preg_match($regex, $plantilla, $matches); # $matches se define al vuelo
$match = $matches[0];
```

Obtención del código HTML iterativo en Python:

```
import re
regex = re.compile("<!--TELEFONOS-->(.|\n){1,}<!--TELEFONOS-->")
match = regex.search(plantilla).group(0)
```

Una vez obtenido el `match` (fragmento de código HTML coincidente con la expresión regular), tanto el diccionario como las sustituciones, deberán realizarse de manera estándar pero dentro de un bucle, condicionado por la cantidad de registros obtenidos de la consulta SQL. La única salvedad, es que ambos pasos deberán realizarse en la misma estructura de control cíclica y, el resultado de cada sustitución, *sumarse* dentro de una misma variable. La sustitución, en este caso, se efectuará sobre el `match` y no sobre la plantilla.

Sustitución iterativa en PHP:

```
$render = "";
foreach($registros as $array) {
    $diccionario = array(
        "{nombre}"=>$array[0],
        "{apellido}"=>$array[1],
        "{telefono}"=>$array[2]
    );
    $render .= str_replace(array_keys($diccionario),
```

```
        array_values($diccionario), $match);
}
```

Sustitución iterativa en Python:

```
render = ""
for tupla in registros:
    diccionario = dict(
        nombre=tupla[0],
        apellido=tupla[1],
        telefono=tupla[2]
    )
    render += Template(match).safe_substitute(diccionario)
```

Finalmente, deberá sustituirse el `match` por el `render` en la plantilla.

Sustitución del match por el render en PHP:

```
$render_final = str_replace($match, $render, $plantilla);
```

Sustitución del match por el render en Python:

```
render_final = plantilla.replace(match, render)
```

Si no se desea conservar los comentarios HTML identificadores de la sustitución dinámica (lo más recomendado), los mismos, se podrán reemplazar mediante la función `str_replace` en PHP y el método `replace` del objeto *string* de Python.

Sustituciones combinadas

Cuando en una misma plantilla se necesite efectuar una sustitución estática y una o más sustituciones dinámicas, el proceso se hará paso a paso. Esto significa, que en primer lugar, se efectuará la sustitución estática. El resultado obtenido en ese `render`, deberá ser aquel sobre el cuál se realice el `match` para la sustitución dinámica. Es decir, que cuando se llegue al proceso de sustitución dinámica, **no se volverá a traer la plantilla**, sino que **se utilizará el código** ya *renderizado*, **obtenido en el proceso de sustitución estática**.

Tip by Commander in Chief:

Evita utilizar servicios en línea para obtener un *hash* determinado. Estos servicios guardan la cadena ingresada asociándola al *hash* resultante y se utilizan para efectuar el proceso de ingeniería inversa sobre un *hash*, a fin de obtener la cadena original.

Pásate a GNU/Linux con Arch Linux: Parte II

MANUAL DE ARCH

En el artículo anterior el sistema base quedó instalado y con nuestro usuario creado. Ahora vamos a instalar el escritorio y dar los últimos retoques al sistema.

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://archninfa.blogspot.com.es/

Redes sociales:
Twitter: **@MMontesDiaz**

Una vez reiniciado el sistema, debemos levantar la red. Sólo hará falta hacerlo una vez y en el siguiente arranque lo hará desde el perfil seleccionado. Para ello, nos *logueamos* en el sistema como root y, suponiendo que al perfil lo hayamos llamado **mired**, ejecutamos:

```
# netcfg mired
```

En caso de haber instalado la versión **x86_64**, editamos el archivo `/etc/pacman.conf` y descomentamos la sección **[multilib]**:

```
[multilib]
SigLevel = PackageRequired
Include = /etc/pacman.d/mirrorlist
```

Ahora, continuamos con la instalación *(NdR: la numeración de los pasos es continuación de la entrega anterior)*:

9. Instalación de sudo

Sudo es una utilidad que permite a ciertos usuarios del sistema ejecutar tareas de administrador.

```
# pacman -Sy sudo
```

Configuramos la utilidad:

```
# EDITOR=nano visudo
```

Nos vamos a la línea `# %wheel ALL=(ALL)ALL` y la descomentamos, quedando así:

```
%wheel ALL=(ALL)ALL
```

Añadimos nuestro usuario al grupo **wheel**:

```
# gpasswd -a Nuestro_user wheel
```

Salimos de la sesión root.

```
# exit
```

Y ahora, entramos con nuestro usuario.

A partir de este momento, para realizar tareas administrativas, utilizaremos la siguiente sintaxis:

```
$ sudo <comando>
```

Por ejemplo, para sincronizar la lista de paquetes y actualizar nuestro sistema:

```
$ sudo pacman -Syu
```

10. Configurando el sonido.

Instalamos los paquetes necesarios:

```
$ sudo pacman -S pulseaudio-alsa alsa-plugins alsa-utils
```

Para **x86_64**:

```
$ sudo pacman -S pulseaudio-alsa alsa-plugins alsa-utils lib32-alsa-plugins
```

11. Instalando las "X"[43]

Procedemos a la instalación de los paquetes necesarios:

```
$ sudo pacman -S xorg-server xorg-xinit xorg-server-utils xf86-input-evdev
xorg-twm xorg-xclock xterm ttf-dejavu
$ sudo pacman -S mesa mesa-demos
```

Editamos el archivo `/etc/X11/xorg.conf.d/10-evdev.conf` y establecemos la distribución del teclado a español:

```
Section "InputClass"
 Identifier "evdev keyboard catchall"
 MatchIsKeyboard "on"
```

43 NdR: "X" se refiere al protocolo **X Window Sistem** (muchas veces llamado X11 haciendo referencia a la última versión dek mismo), el cuál es el que facilita la ejecución interfaces gráficas en sistemas operativos basados en Unix.

```
    MatchDevicePath "/dev/input/event*"
    Driver "evdev"
    Option "XkbLayout" "es"
  EndSection
```

Instalamos el driver apropiado para nuestra tarjeta gráfica.

En primer lugar, identificamos nuestra tarjeta,

```
$ lspci | grep VGA
```

Ahora instalamos el controlador apropiado:

Drivers privativos:

NVDIA:

```
$ sudo pacman -S nvidia
```

Si pacman pide eliminar `libgl` y falla debido a dependencias no satisfechas, eliminamos `libgl` y volvemos a instalar el paquete.

```
$ sudo pacman -Rdd libgl
$ sudo pacman -S nvidia
```

Para Arch x86_64, debemos instalar el paquete `lib32`:

```
$ sudo pacman -S lib32-nvidia-utils
```

Editamos el archivo `/etc/X11/xorg.conf.d/10-evdev.conf`:

```
$ sudo nano /etc/X11/xorg.conf.d/10-evdev.conf
```

ATI

```
$ sudo pacman -S catalyst-dkms catalyst-utils
$ sudo pacman -S lib32-catalyst-utils
$ sudo systemctl enable dkms.service
```

Drivers libres:

NVDIA

```
$ sudo pacman -S xf86-video-nouveau
```

INTEL

```
$ sudo pacman -S xf86-video-intel
```

ATI

```
$ sudo pacman -S xf86-video-ati
```

Genérico

```
$ sudo pacman -S x86-video-vesa
```

“La meta de Arch no es ser grande, es estar bien hecho”

14. Instalación del Escritorio

Para la instalación de KDE:

Los paquetes se encuentran divididos en varios grupos, para ver cuales son:

```
$ sudo pacman -Ssq kde-meta
```

Para instalar todos:

```
$ sudo pacman -Sy $(pacman -Ssq kde-meta) kde-l10n-es
```

De esta forma, si no estamos interesados en las opciones de accesibilidad, por ejemplo, bastará eliminar el paquete `kde-meta-kdeaccessibility`:

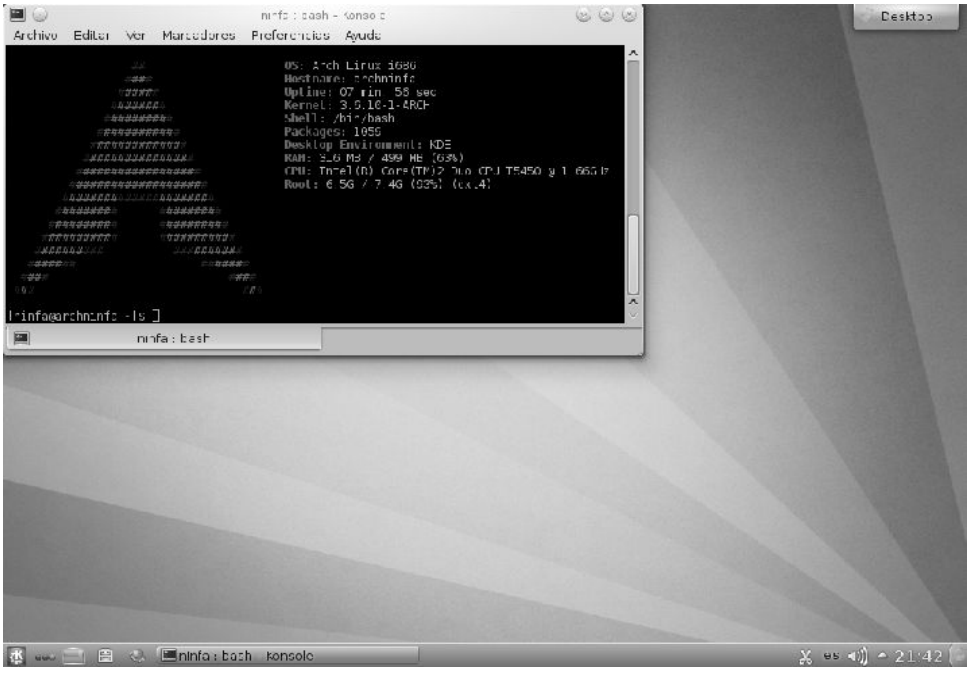
```
$ sudo pacman -Rsnc kde-meta-kdeaccessibility
```

Activamos el login gráfico:

```
$ sudo systemctl enable kdm.service
```

Si queremos tener un gestor gráfico de paquetes, podemos optar por PackageKit

```
$ sudo pacman -S apper
```



Para la instalación de GNOME 3:

Está dividido en dos grupos: Básico y aplicaciones estándar.

```
$ sudo pacman -Syu gnome
```

Varias herramientas opcionales:

```
$ sudo pacman -Syu gnome-extra
```

Si queremos tener todos los paquetes

```
$ sudo pacman -Syu gnome gnome-extra
```

Instalamos la utilidad de configuración de gnome:

```
$ sudo pacman -S gnome-tweak-tool
```

Activamos el login gráfico:

```
$ sudo systemctl enable gdm.service
```

El gestor de paquetes para GNOME sería:

```
$ sudo pacman -S gnome-settings-daemon-updates
```

Para ambos:

Activamos el modo gráfico por defecto:

```
$ sudo systemctl enable graphical.target
```

Ahora, debemos reiniciar nuestro sistema:

```
$ sudo reboot
```

15. Instalación de yaourt

Para tener acceso a los paquetes disponibles en el AUR (Arch User Repository), una buena herramienta es `yaourt`. Para instalarla, podemos hacerlo así:

```
$ sudo pacman -S wget
$ wget https://aur.archlinux.org/packages/pa/package-query/package-query.tar.gz
$ tar zxvf package-query.tar.gz
$ cd package-query
$ makepkg -si
$ cd ..
$ rm -r package-query

$ wget https://aur.archlinux.org/packages/ya/yaourt/yaourt.tar.gz
$ tar zxvf yaourt.tar.gz
$ cd yaourt
$ makepkg -si
$ cd ..
$ rm -r yaourt
```

Otro método es añadiendo el repositorio `archlinuxfr`. Para hacer esto, editamos el archivo `/etc/pacman.conf`:

```
$ sudo nano /etc/pacman.conf
```

y añadimos estas líneas al final del archivo:

```
[archlinuxfr]
Server = Server = http://repo.archlinux.fr/$arch
```

después, actualizamos la lista de paquetes e instalamos:

```
$ sudo pacman -Sy yaourt
```

Una vez instalado, desactivamos el repositorio, bien eliminando las líneas o bien añadiendo una # al principio de cada una para comentarlas:

```
#[archlinuxfr]
#Server = Server = http://repo.archlinux.fr/$arch
```

16. Instalación de codecs, suite ofimática y navegador

Codecs

```
$ sudo pacman -S gstreamer0.10-plugins
$ sudo pacman -S firefox firefox-i18n-es-es
```

Para reproducir vídeos desde el navegador, debemos instalar uno de los siguientes paquetes:

```
$ yaourt -S rosa-media-player-plugin
$ sudo pacman -S totem-plugin
$ sudo pacman -S gecko-mediaplayer
```

Para java:

```
$ sudo pacman -S icedtea-web-java7
```

o

```
$ sudo pacman -S icedtea-web
```

Para flash:

```
$ sudo pacman -S gnash-gtk
```

o

```
$ sudo pacman -S flashplugin
```

Cómo **suite ofimática**, podemos instalar LibreOffice:

```
$ sudo pacman -S libreoffice libreoffice-es
```

Si no queremos tener paquetes correspondientes a escritorios que no utilizamos, podemos proceder así:

```
$ sudo pacman -S libreoffice-base  libreoffice-calc  libreoffice-common
libreoffice-draw libreoffice-impress libreoffice-math libreoffice-
postgresql-connector  libreoffice-sdk  libreoffice-sdk-doc  libreoffice-
writer libreoffice-es
```

Para escritorios GTK cómo GNOME:

```
$ sudo pacman -S libreoffice-gnome
```

Para escritorios QT cómo KDE:

```
$ sudo pacman -S  libreoffice-kde4
```

En el próximo artículo veremos otros escritorios más ligeros, que nos permiten poder utilizar equipos antiguos.

Creando un túnel cifrado SSH: Regalo especial fin de año, by María José (a.k.a. "Meri")

Esto es útil si utilizamos redes abiertas o no fiables. Lo primero que se necesita es un servidor SSH. Si nuestra IP es dinámica, necesitaremos un servicio DNS dinámico como NO-IP[44].

Paso1:

```
$ ssh -ND PUERTO user@host
```

Podemos añadir un alias en el archivo `~/.bashrc`

```
alias tunel="ssh -ND PUERTO ser@host"
```

De esta forma, para crear la conexión, sólo habrá que ejecutar:

```
$ tunel
```

user: nuestro usuario

host: la dirección IP o dominio del equipo al que queremos conectarnos

PUERTO: El puerto al que vamos a redirigir el navegador.
No es el del servidor ssh. Por ejemplo: 3333

Paso 2:

Configurar el navegador Firefox :

Editar → Preferencias → Avanzadas → Red → Conexión → Configuración:

Marca la casilla "configuración manual de proxy" y escribe `localhost` en el campo "servidor SOCKS" y en el puerto, `3333`.

¡Ya puedes navegar seguros desde esa red!

44 http://www.no-ip.com

Analizando los logs de acceso de Apache

SERVIDORES

Entender y analizar los logs de acceso de Apache, nos puede resultar útil para cosas tan triviales como obtener estadísticas de visitas como para adoptar políticas de seguridad tanto preventivas como paliativas y en casos aún más graves, judiciales. En este artículo, haremos un breve resumen de como interpretar los logs de Apache y ver de qué forma nos pueden ser útiles.

Escrito por: **Eugenia Bahit** (Arquitecta GLAMP & Agile Coach)

Eugenia es **Arquitecta de Software**, **docente** 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** e integrante del equipo de **Debian Hackers**.

Webs:
Cursos de programación a Distancia: www.cursosdeprogramacionadistancia.com
Agile Coaching: www.eugeniabahit.com

Redes sociales:
Twitter / Identi.ca: **@eugeniabahit**

La primera vez que observas el log de accesos de Apache, decides que lo mejor que podrías hacer con él es... es... en fin... cualquier cosa menos prestarle atención. Pero, *wait a minute little Saltamontes*, que no solo no es difícil entenderlos, sino que además, es mucho más útil de lo que te imaginas.

Los logs de Apache pueden configurarse mediante la directiva `CustomLog`[45] dentro del `VirtualHost` utilizando la sintaxis:

```
CustomLog /ruta/al/archivo formato
```

Donde formato podría ser `common` o `combined`. Por ejemplo:

45 http://httpd.apache.org/docs/2.2/mod/mod_log_config.html#customlog

```
CustomLog /miswebs/example.com/logs/access.log combined
```

La diferencia entre `common` y `combined` se podría decir que es poco sutil:

```
common:   %h %l %u %t \"%r\" %>s %b
combined: %h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"
```

Y es que el segundo, guarda la URI de referencia mediante la cuál se realizó la petición y el User-agent (información relativa al dispositivo utilizado para la conexión: navegador, buscador, etc.).

Los formatos anteriores pueden resultar incomprensibles, pero ¡no te estreses! Es mucho más sencillo de lo que imaginas:

```
%h      es el host que accede. Por ejemplo, una IP como:
        123.456.78.90

%l      el protocolo de identificación del usuario RFC 1413.
        no enloquezcas que seguramente verás un guión como salida a no
        ser que se opere en una red privada

%u      el nombre del usuario (comúnmente la salida será un guión a no
        ser que se trate de un usuario autenticado en el sistema)

%t      marca de tiempo: fecha completa incluyendo hora y UTC. Por ejemplo:
        10/Dec/2012:14:54:58 -0300

%r      (las barras que la envuelven son simples escapes de caracteres para
        que las comillas que le siguen, se impriman). Es la solicitud
        realizada por el usuario.
        Primero, incluye el método (GET, POST, PUT, etc.).
        Luego, el recurso solicitado (archivo al cuál se accede).
        Y finalmente, el protocolo (HTTP 1.1/1.0). Un ejemplo sería:
        "GET /index.php HTTP/1.1"

%>s     El código de respuesta de estado. Por ejemplo:
        200 (OK)
        404 (Not Found), etc.

%b      La cantidad de bytes entregados al usuario. Por ejemplo:
        4108
```

`combined`, suma además:

```
%{Referer}i      la URI de referencia (archivo, sitio o página que contiene
                 el vínculo hacia el recurso solicitado). Por ejemplo:
                 http://example.org/goTo.php?p=http://example.com/index.php

%{User-agent}i   El user-agent del usuario (valga la redundancia). Por ej.:
                 Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:11.0) Gecko/20100101 Firefox/11.0
```

A mi me gusta utilizar el formato `combined`. Es mucho más completo para obtener

estadísticas.

Por ejemplo, **el contador de descargas** que hice **para** la Web del Magazine **www.hdmagazine.org** es mitad un *script* de shell y mitad Python (ya que el mismo *script* de shell lo utilizo para otras cosillas xD), que básicamente lo que hace es un `grep` sobre los recursos coincidentes con una edición particular y los cuenta mediante la opción `-c`.

Algo tan simple como contar la cantidad de solicitudes realizadas al PDF de la edición anterior (edición número 1), se obtiene con solo 1 comando y 3 argumentos:

```
$ grep -c magazine=HackersAndDevelopers\&num=1\ HTTP access.log
```

Pero no solo es útil para obtener estadísticas de "visitas". También nos puede servir para saber si algún usuario ha estado intentando acceder con intensiones "non santas". Por ejemplo, es muy típico que alguien intente ingresar a alguna URL que contenga la palabra `admin`:

```
$ grep -i admin access.log | less
190.188.247.165 - - [03/Dec/2012:10:38:59 -0500] "GET
/admin/scripts/tinymce/jscripts/tiny_mce/plugins/ibrowser/scripts/phpThumb/phpThumb
.php?src=./index.php&fltr[]=blur|
5;echo+082119f75623eb7abd7bf357698ff66c>cache/acunetix; HTTP/1.1" 404 470 "-"
"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"
190.188.247.165 - - [03/Dec/2012:10:39:01 -0500] "GET /admin.htm HTTP/1.1" 404 429
"-" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"
190.188.247.165 - - [03/Dec/2012:10:39:01 -0500] "GET /admin.html HTTP/1.1" 404 429
"-" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"
190.188.247.165 - - [03/Dec/2012:10:39:02 -0500] "GET /admin HTTP/1.1" 404 426 "-"
"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"
190.188.247.165 - - [03/Dec/2012:10:39:02 -0500] "GET /admin/phpThumb/phpThumb.php?
src=./index.php&fltr[]=blur|5;echo+082119f75623eb7abd7bf357698ff66c>cache/acunetix;
HTTP/1.1" 404 438 "-" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64;
Trident/5.0)"
190.188.247.165 - - [03/Dec/2012:10:39:02 -0500] "GET /Admin HTTP/1.1" 404 427 "-"
"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"
190.188.247.165 - - [03/Dec/2012:10:39:02 -0500] "GET /ADMIN HTTP/1.1" 404 426 "-"
"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"
190.188.247.165 - - [03/Dec/2012:10:39:02 -0500] "GET /admin/upload/phpThumb.php?
src=./index.php&fltr[]=blur|5;echo+082119f75623eb7abd7bf357698ff66c>cache/acunetix;
HTTP/1.1" 404 440 "-" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64;
Trident/5.0)"
190.188.247.165 - - [03/Dec/2012:10:39:03 -0500] "GET /adminpanel HTTP/1.1" 404 430
"-" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"
190.188.247.165 - - [03/Dec/2012:10:39:03 -0500] "GET /admin0 HTTP/1.1" 404 427 "-"
"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"
190.188.247.165 - - [03/Dec/2012:10:39:03 -0500] "GET /admin1 HTTP/1.1" 404 426 "-"
"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"
190.188.247.165 - - [03/Dec/2012:10:39:03 -0500] "GET /admin/release HTTP/1.1" 404
431 "-" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"
190.188.247.165 - - [03/Dec/2012:10:39:03 -0500] "GET /admin_ HTTP/1.1" 404 427 "-"
"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"
190.188.247.165 - - [03/Dec/2012:10:39:04 -0500] "GET /_admin HTTP/1.1" 404 427 "-"
"Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"
190.188.247.165 - - [03/Dec/2012:10:39:04 -0500] "GET
/admin/tiny_mce/plugins/ibrowser/scripts/phpThumb/phpThumb.php?
src=./index.php&fltr[]=blur|5;echo+082119f75623eb7abd7bf357698ff66c>cache/acunetix;
HTTP/1.1" 404 464 "-" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64;
Trident/5.0)"
```

```
190.188.247.165 - - [03/Dec/2012:10:39:04 -0500] "GET
/zadmin/tiny_mce/plugins/ibrowser/scripts/phpThumb/phpThumb.php?
src=./index.php&fltr[]=blur|5;echo+082119f75623eb7abd7bf357698ff66c>cache/acunetix;
HTTP/1.1" 404 465 "-" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64;
Trident/5.0)"
190.188.247.165 - - [03/Dec/2012:10:39:04 -0500] "GET /administrator HTTP/1.1" 404
432 "-" "Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)"
```

Y les aseguro que eso, es menos del 0,3% de las solicitudes que me encontré ese famoso 3 de diciembre, desde la misma IP **tras lanzarse la segunda edición de Hackers & Developers Magazine**.

La suma de los accesos realizados por la IP en cuestión, es una buena forma de obtener "el empujón final" para decidir bloquear a dicho usuario:

```
$ grep -c 190.188.247.165 access.log
7155
```

Parece una pavada ¿cierto? Pero no lo es.

Gracias a los logs de acceso de Apache, sabiendo analizarlos podemos tomar medidas preventivas importantes. Y no solo preventivas: en algunos países como la Rep. Argentina (donde Hackers & Developers Magazine tiene su asiento legal), este tipo de "pruebas" -según cada caso- podría ser considerada como un delito de ámbito penal. El "niñato" que anduvo *"jugando a ver si el software que me bajé de somosreguachihackers.hosting-gratuito.lala, hace algo cuando le doy clic al botón CheGuachinApretáAcá"*, ha dejado muchísimos rastros.

> ***En casos mucho más serios que este ejemplo, los datos que Apache provee, pueden ser utilizados a nivel forense*** *ya que rastrear una IP y obtener su localización aproximada no es difícil para los organismos de Justicia:* ***a través de la IP se puede obtener no solo la ubicación geográfica*** *(país, ciudad, provincia o estado), sino además, el proveedor ISP y la latitud y longitud del nodo.*

```
IP Address:  190.188.247.165          Location:  La Plata, Buenos Aires, Argentina
Coordinates: -34.9215, -57.9545       ISP:       Prima S.A.
```

Ejemplo de datos que pueden obtenerse tras un rastreo simple de la IP

Con las coordenadas del nodo, hasta Google Maps nos puede dar incluso, un domicilio bastante aproximado como se muestra en la siguiente imagen:

A nivel forense, estos datos son mucho más útiles que para nosotros: para la Justicia, solo bastará un oficio a la compañía proveedora del servicio, para obtener el nombre del cliente que utilizaba la IP en la fecha y hora especificada.

Es decir, que **gracias a los logs de Apache**, no solo podemos obtener divertidas estadísticas y tomar medidas preventivas, sino que además, **la Justicia puede obtener pruebas legalmente válidas**.

Y ahora ¿sigues pensando que revisar los logs de Apache es una pavada? A ti también te puede servir de mucho, puesto que no cabe duda de que **bloquear una IP en estos casos, no solo reducirá el ancho de banda consumido, sino que además, frenará un poco "la ansiedad" del niñato**:

```
iptables -I INPUT -s 190.188.247.165 -j DROP
```

Vale aclarar que **la revisión de los logs de acceso de Apache** -más allá de contar estadísticas-, debe ser asumida como **una política de seguridad a implementar de forma periódica**. Pues no existe un único usuario "molesto" ni tampoco los niñatos poseen IPs fijas.

La revisión de los logs de acceso de Apache, como política preventiva de seguridad, **SIEMPRE debería ir acompañada de la revisión de los logs de errores. Esto nos ayudará a asegurar aún más nuestras aplicaciones**, dado que, si haciendo un `grep` por la IP en cuestión sobre el log de errores, éste nos arroja algo, en el error (o los errores) arrojado estará la respuesta a la pregunta: ¿qué medida de seguridad se deberá sumar a nuestra aplicación?:

```
grep 190.188.247.165 error.log | less
```

Generalmente, la mayoría de los errores será del tipo 404 (archivo no encontrado):

```
... [error] [client 190.188.247.165] File does not exist:
/ruta/a/document/root/phpmyadmin
```

Pero **el punto de observación**, no **debe centrarse** en los errores 404, sino **en los errores de código** (500 y "compañía"). Son los únicos que nos ayudarán a saber qué parte del código de nuestra app, se debe asegurar. Una forma rápida de encontrar errores de PHP, por ejemplo, sería la siguiente:

```
grep 190.188.247.165 error.log | grep '\(Warning\|Notice\)' | less
```

*Como nota de color y a mero título personal, quiero admitir que **este tipo de "jueguecitos" resulta algo penoso cuando has pasado los últimos 16 años de tu vida, compartiendo con decenas de miles de personas** que no conoces, todo tu conocimiento **de forma absolutamente desinteresada**. Siempre digo que la vida no es ni justa ni injusta, sino que es "el resultado de las decisiones que uno toma". Y sentir que tu decisión de compartir con el mundo todo lo que sabes, trae aparejado que cualquier persona psicológicamente inestable intente dañarte, no es el sentimiento que elegiría para este fin de año.*

¡Gracias a quiénes saben valorar el esfuerzo! ¡Feliz 2013!

Eugenia Bahit
Responsable de Proyecto
Hackers & Developers Magazine

*Y por favor, ten siempre presente **"...que no son las cosas las que aportan significado a un momento determinado, sino que es el momento el que aporta significado a las cosas..."**.-*

*Frase pronunciada por el **Rabino Abraham Joshua Heschel***

¡Feliz año!

2013

ASCII ART

Woman in Champagne Glass

by **joan stark** (jgs)
spunk1111@juno.com
ASCII ART GALLERY
www.ascii-art.com

TU ZONA EXCLUSIVA U!

Para **publicar** en la zona **U!** envíanos tu mensaje a **contacto@hdmagazine.org** indicando en el **asunto**: ZonaU!

Pedidos y Agradecimientos de nuestros lectores (y adictos... xD)

Carlos Ramírez (México): escribo para felicitarlas por su gran trabajo, sus contenidos son fascinantes, la verdad me han gustado bastante. Por otro lado quisiera preguntarles si podrían publicar algo relacionado con transmisión de vídeo en vivo, ademas claro de los hangouts de google.

Ariel Scherman (Argentina): Primero que nada, gracias por cumplir lo que pedí del artículo de MVC, estuvo genial, así como todo el número en general. (...) me gustaría si alguna pudiese escribir en algún momento sobre patrones de diseño, o temas relacionados con la arquitectura, dado que estoy intentando meterme en ese tema y me está costando un poco, debido a la gran cantidad diversa de información que hay en internet, que hace que no sepa "qué camino recorrer" (...) desde ya les agradezco y las felicito por todo lo que están haciendo.

Alejandro Plancarte: me gustaría que hicieran un articulo sobre OpenFlow, sus plataformas y herramientas de depuración más útiles para el desarrollo de aplicaciones de control de red en OpenFlow. Gracias y ÉXITO!!!!

Víctor Martín Rosas García (PIURA, Perú): Hola, muchas gracias por su respuesta y publicación en la revista. Quería comentarles que soy de Piura-Perú, no de Lima-Perú, sólo por aclarar :) Podrían crear un artículo sobre las ventajas y desventajas entre PHP Y Python, y cuando deberíamos optar por uno de ellos.

Respuesta: *Mil disculpas por la confusión, Víctor!*

Juan Rafael Ricabal (Cuba): Muy instructivo el segundo numero de la revista, créanme que casi me estoy haciendo adicto, felicidades a todas por la navidad, espero que el 2013 sea de un rotundo éxito.

Jose Luzón Sanchis (España): Soy un gran fan y lector de vuestra revista. La descubrí por casualidad y me pareció genial. Estáis haciendo un gran trabajo. Espero que este proyecto siga adelante por mucho tiempo. Si pudierais publicar algo más sobre Arch sería de agradecer. Por ejemplo, qué hacer después de instalarlo, programas "imprescindibles", tips, etc. Por cierto, tampoco sería mala idea que se implementara un formulario para suscribirnos a la revista y poder recibirla directamente al correo electrónico. Muchas gracias y enhorabuena por el fantástico trabajo que estáis haciendo.