



Como hacer código seguro

Programación Segura PHP



Código de Conduta UBUNTU

SEO en WordPress



Archlinux

Distribución
simple y ligera



Proyecto factible en el campo
de las tecnologías de la educación.



www.vaslibre.org.ve



Y más...

Todo el contenido está bajo licencia de Creative Commons.

Puede copiar, distribuir, mostrar públicamente su contenido y hacer obras derivadas, siempre y cuando

- a) Reconozca los créditos de la obra
- b) No la use de forma comercial
- c) Comparta bajo la misma licencia.



Twitter: @vaslibre



03 Editorial

04 Publicidad & Patrocinantes

05 Seo en Wordpress

07 Evento Tic's & Educacion

08 ArchLinux: Simple y Ligera

10 Sitios Recomendados

11 Entendiendo los JOINS

13 Evento Flisol 2011

17 Proyecto Factible

19Codigo de Conducta UBUNTU

Staff:

Juan C. Karroum (JCK)
Héctor A. Mantellini (Xombra)
Jhon Vera (jjedi_ve)

Colaboradores:

Héctor A. Mantellini (Xombra)
Eduardo Echeverria(echevemaster)
Angel Cruz (Abr4xas)
Cesar Rojas (ovejorock)
David Emerling Rondon
Alexander Garzon

En este año 2011, el grupo VaSlibre está centrado en llevar a la mayor cantidad de público las bondades del software libre en diferentes ámbitos.

Colaboraremos con la Universidad Abierta (UNA) de Valencia - Carabobo, instalando en sus equipos de computación la Distribución Canaima, igualmente se dictaran charlas y talleres de capacitación.

Participaremos en el evento coordinado y organizado por la Asociación Webmaster Venezuela y patrocinado por la Universidad de Carabobo, I Jornada de Formación para el Uso de Herramientas de Software Libre aplicadas a las TICs y a la Educación.

Organizaremos y coordinaremos con la colaboración de Asociación Webmaster Venezuela y de Ubuntu Carabobo el Flisol 2011.

Estamos en conversaciones para organizar el 7º CNSL (Séptimo Congreso Nacional de Software Libre.)

Además, en el horizonte hay más eventos y charlas en Universidades y Liceos en los cuales participaremos para difundir el Software Libre.

Participa con nosotros, unete a la Fuerza!





Seo en WordPress

Antes de todo debemos saber que es SEO: El posicionamiento en buscadores o posicionamiento web es el resultado de la recuperación de información en la base de datos de los grandes motores de búsqueda de Internet por el uso de algoritmos de búsqueda en el software.

La tarea de ajustar la información de las páginas que se pretenden hacer aparecer en primeras posiciones de los resultados es conocida como SEO, sigla en inglés de Search Engine Optimization, o sea, 'Optimización para motores de búsqueda'. Consiste en aplicar diversas técnicas tendientes a lograr que los buscadores de Internet sitúen determinada página web en una posición y categoría alta (primeras posiciones) dentro de su página de resultados para determinados términos y frases clave de búsqueda. También es entendido como las técnicas de desarrollo web que tengan como objetivo mejorar la posición de un determinado sitio web por sus páginas en la lista de resultados de los motores de búsqueda que en inglés se identifican como SERPs y forma parte de la jerga de los webmasters hispanohablantes. Via wikipedia

Ahora bien el SEO es muy importante debido a:

Muchas compañías saben que aparecer en las primeras posiciones dentro de estos resultados puede reportarles excelentes beneficios. Hay ocasiones en los que el usuario

busca un determinado servicio, y estar en la primera o segunda posición puede significar un cliente casi seguro. Por ello, la inversión realizada en los servicios de un 'SEO' profesional puede ser recuperada en pocas semanas o meses. <http://google.dirson.com/>

El sitio

<http://www.woorank.com> te muestra muchas cosas que debemos tener en cuenta a la hora de aplicar el SEO a nuestro sitio. Ok, hasta aquí todo bien no? Que pasa cuando tenemos un

blog gestionado con WordPress? Como todos sabemos wordpress es:

Un sistema de gestión de contenido enfocado a la creación de blogs (sitios web periódicamente actualizados). Desarrollado en PHP y MySQL, bajo licencia GPL y código modificable, tiene como fundador a Matt Mullenweg. WordPress fue creado a partir del desaparecido b2/cafeleg y se ha convertido junto a Movable Type en el CMS más popular de la blogosfera. Las causas de su enorme crecimiento son, entre otras, su licencia, su facilidad de uso y sus características como gestor de contenidos.

Otro motivo a considerar sobre su éxito y extensión, es la enorme comunidad de desarrolladores y diseñadores, que se encargan de desarrollarlo en general o crear plugins y temas (themes) para la comunidad, siendo usado hasta

ahora por más de 210 millones de usuarios. [via wikipedia]

Existen

muuuuuuuuuchos plugins orientados al SEO pero yo solo uso este que me parece superior: All in One SEO Pack Out-of-the-box SEO for your WordPress blog. <http://semperfiwebdesign.com/>

También existe una

forma de aumentar las visitas a nuestra pagina/blog y es mediante el uso de RSS.

RSS son las siglas de Really Simple Syndication, un formato XML para syndicar o compartir contenido en la web. Se utiliza para difundir información actualizada frecuentemente a usuarios que se han suscrito a la fuente de contenidos. El formato permite distribuir contenidos sin necesidad de un navegador, utilizando un software diseñado para leer estos contenidos RSS (agregador). A pesar de eso, es posible utilizar el mismo navegador para ver los contenidos RSS. Las últimas versiones de los principales navegadores permiten leer los RSS sin necesidad de software adicional. RSS es parte de la familia de los formatos XML desarrollado específicamente para todo tipo de sitios que se actualicen con frecuencia y por medio del cual se puede compartir la información y usarla en otros sitios web o programas. A esto se le conoce como redifusión web o sindicación web (una traducción incorrecta, pero de uso muy común). Fuente: wikipedia.



Seo en WordPress

En twitter:

* [twitterfeed](#) feed your blog to twitter – twitterfeed lets you post any RSS or Atom feed to twitter automatically.

* [dlvr.it](#) Syndicate your content to the social web.

Como nota adicional y quizás un poco fuera del contexto les puedo comentar que también debemos tener en cuenta la nueva era tecnológica en la que vivimos, ya que existe una gran cantidad de personas accediendo al internet desde sus teléfonos celulares, bien sean blackberry, android, nokia en fin, también debemos ofrecerles un buen diseño y para ello nos encontramos con:

WP Super Cache Very fast caching plugin for

WordPress. ocaoimh.ie/wp-super-cache/ que va tomado de la mano con

WPTouch A plugin which formats your site with a mobile theme for visitors on Apple iPhone / iPod touch, Google Android,

Blackberry Storm and Torch, Palm Pre and other touch-based smartphones.

<http://www.bravenewcode.com/products/wptouch>

Mejorar rendimiento de WordPress

0.- Optimizar la base de datos: desde PHPMYAdmin: antes de hacer algo en su base de datos Realice una copia de seguridad, desde el PHPMYAdmin.

1.- Realice una copia de seguridad y empieza por eliminar comentarios y espacios en blanco de todos sus archivos, puede utilizar alguna de estas herramientas web. Instalar extensiones para Firefox, al imprescindible Firebug le puedes añadir YSlow que le aconseja qué

debe hacer para aumentar la velocidad de su sitio web según las reglas del equipo de desarrolladores de Yahoo

2.- use el compresor de css <http://wordpress.org/extend/plugins/css-compress/>

3.- instalar en su WP el plugin de cacheo, de esa forma cachea sus páginas para incrementar la velocidad de respuesta y minimizar la carga del servidor. <http://1-blog-cacher.es/javier-garcia.com/> también puede usar <http://ocaoimh.ie/wp-super-cache/> (recomendado)

4.- Elimine los plugin innecesarios es decir aquellos que no esta usando.

5.- Minimice y optimizar las imágenes, verifique que las imágenes posean 72 dpi y que al mostrarla posea en el código html generado el tamaño a mostrar ajustado a las dimensiones reales.

6.- Plugin optimizador de base de datos

<http://wordpress.org/extend/plugins/optimize-db/>

Por cierto, mi blog posee un woorank de 52,9 xD ¿Cual es el tuyo?



Angel Cruz

Twitter: [@abr4xas](#)

<http://abr4xas.org>



Enlaces de Interés para saber más de SEO:

<http://www.leemiblog.com/>

<http://www.kabytes.com/>

<http://vagabundia.blogspot.com/>

<http://www.webmasterlibre.com/>

<http://www.awven.com/>

EVENTO



Universidad de Carabobo
Facultad de Ciencias de la Educación
Dirección de Tecnologías de Información y Comunicación de la FaCE
Dirección de Gestión Administrativa de la FaCE
con apoyo de
Asociación Webmaster de Venezuela

TE INVITAN A LA:

I JORNADA DE FORMACIÓN PARA EL USO DE HERRAMIENTAS DE SOFTWARE LIBRE APLICADAS A LAS TIC'S Y A LA EDUCACIÓN



Fecha : 04 de Marzo de 2011

Lugar: Auditorio de la Facultad de Ciencias de la Educación

Horario : de 8:00 a.m a 4:40 pm

1era Etapa de Inscripciones 100 Bs. (13/12/2010 al 25/01/2011)

2da Etapa de Inscripciones 120 Bs. (26/01/2011 al 02/03/2011)

Incluye Certificado y Refrigerio

Deposito Bancario a Cuenta Corriente Banco Banesco Nro.
0134-0319-82-3191098478 a nombre de
SOLUCIONES SAEF, C.A.

Para mayor información, dirigirse a la Dirección de Tecnologías de Información y Comunicación de la Facultad de Ciencias de la Educación Universidad de Carabobo o a través de la página Web www.face.uc.edu.ve y los teléfonos: 0241-6004000 Ext. 325458 /0416-7418882/0414-4304448/0414-0447060 o en Twitter: @awvene

FORMALICE SU INSCRIPCIÓN EN <http://registro.awven.com>

ArchLinux: simple y ligera

Arch es una distribución que destaca por su sistema 'rolling release' es decir que sus repositorios se actualizan constantemente y no existen versiones ni lanzamientos como es habitual en la distribuciones GNU/Linux.

Esta distribución está dirigida a usuarios de GNU/Linux con cierta experiencia que deseen conocer una alternativa diferente pero muy potente. Verán que no hay que ser un Jedi un “geek” para usar esta distribución pero eso sí. Hay que disponer de mucho TIEMPO LIBRE y sobre todo PACIENCIA.

Distribuciones Frozen Release

contrario el sistema base está desprovisto de todo el desorden que puede ocultar partes importantes del sistema ó hacer su acceso más difícil o complicado. Posee un conjunto de archivos de configuración organizados para que su acceso y edición sea rápido y sin herramientas de configuración gráficas que tienden a ocultar muchas opciones al usuario. Por tanto un sistema ArchLinux es fácilmente configurable hasta el más mínimo detalle.

Precisión de código por encima de la comodidad. El sistema ArchLinux da prioridad a la elegancia del diseño en lugar de parches



Una distro Frozen Release es aquella que consta de “ciclos” de lanzamientos; es decir los desarrolladores se encargan de armar una serie de aplicaciones y bibliotecas para una “versión” de su distribución. Ejemplo de estas distribuciones son: Fedora Ubuntu Debian (Versiones estables).

Distribuciones Rolling Release

Una distribución Rolling Release es aquella que el software que contiene está en constante actualización. Ejemplo de estas distribuciones son: ArchLinux Gentoo Debian (Versiones inestables).

Modelo de Desarrollo (The Arch Way)

El modelo de desarrollo de ArchLinux:

Simple

Precisión de código por encima de la comodidad

Centrado en el usuario

Libre

(Extraído de la wiki de archlinux.org)

Simple

ArchLinux define simplicidad como una ligera estructura base sin agregados innecesarios modificaciones o complicaciones que permite a un usuario modelar el sistema de acuerdo a sus propias necesidades. Una ligera estructura base no significa que el sistema base sea pequeño. Por el

innecesarios automatización ó mejoras visuales. Los parches de software se mantienen a un mínimo absoluto y sólo se proporcionan para evitar un fallo grave. Idealmente nunca. Los conceptos diseños y características están generados e implementados usando los principios de la Filosofía de Arch como guías en lugar de obedecer influencias externas.

Centrado en el usuario

Muchas distribuciones GNU/Linux intentan ser más amigables al usuario ArchLinux siempre ha sido y seguirá siendo 'centrado al usuario'. Dándole a los usuarios el control absoluto a todo el sistema administrando completamente por sí mismos.

Libre

Entre los principios del desarrollo de ArchLinux está la libertad de elección.

Los usuarios no sólo están autorizados a hacer todas las decisiones relativas a la configuración del sistema sino también escoger lo que su sistema será.

ArchLinux: simple y ligera

ArchLinux comparada con otras distros

-Arch vs. Ubuntu: Si te gusta compilar tus propios kernels testear proyectos solo-CVS ó compilar un paquete desde sus fuentes Arch esta mejor equipado. Si quieres encender el computador y poder usar rápidamente el software del sistema sin configurar mucho ni 'aprender' Ubuntu está mejor equipado. En general los desarrolladores y usuarios avanzados probablemente prefieran Arch sobre Ubuntu.

-Arch vs. Debian: Arch posee menos paquetes sin embargo, Arch provee mejor soporte que Debian para construir paquetes propios. Arch es muy permisivo cuando un paquete es definido como 'non-free' (no libre) por GNU. Está optimizado para arquitecturas i686 y x86_64. Los paquetes de Arch son más nuevos que los de Debian (incluso en su rama unstable). ArchLinux es una distribución 'andante' mientras que Debian Stable se libera con paquetes 'congelados'.

Puedes consultar mas comparaciones en la wiki de Arch:

[http://wiki.archlinux.org/index.php/Arch_Compared_to_Other_Distributions_\(Español\)](http://wiki.archlinux.org/index.php/Arch_Compared_to_Other_Distributions_(Español))

ArchLinux en servidores: Quizás ArchLinux sea el método ideal para los entornos de escritorio pero sin duda para un servidor no lo es. Imagina tener que parar y reiniciar el servidor cada dos por tres por actualizar el kernel. Sería catastrófico además de muy poco práctico.

El equipo de desarrolladores de Arch decidió incorporar a los repositorios un kernel LTS (Long Time Support). Este kernel esta basado en la versión 2.6.32.x y además agrega algunos parches que añade el equipo de Ubuntu a su distribución Ubuntu Server.

Paquetes Oficiales:

core Contiene los paquetes más importantes y críticos de la distribución.

extra Contiene paquetes muy requeridos para la comunidad, sin embargo no todos son necesarios para hacer andar el sistema.

community Contiene paquetes binarios que fueron contribuidos por la comunidad en este repositorio colaboran otra escala de desarrolladores (Trusted Users).

unstable Contiene software experimental e inestable especialmente si la versión de desarrollo de un paquete se ha vuelto muy popular.

testing Contiene paquetes que son candidatos a entrar en los repositorios [core]

[extra] ó [community]. Sólo usuarios experimentados deben usarlo.

release Es una instantánea del repositorio [core] cuando se usa una release. Está aquí por motivos históricos y no está pensado para uso general.

unsupported No es realmente un repositorio. A diferencia de los demás no provee paquetes binarios. Es usado para referirse a la colección de PKGBUILD (Archivo descriptivo de construcción de un paquete para ArchLinux) en AUR (Arch User Repository) que son enviados por usuarios comunes. Por lo tanto no es oficial.

Paquetes NO-Oficiales

AUR es una plataforma en la que los usuarios de ArchLinux que no son desarrolladores pueden contribuir agregando paquetes NO oficiales estos luego pueden ser incluidos en repositorios binarios (como [community]) y ser mantenidos por un desarrollador.

Reconstrucción de paquetes oficiales con ABS (Arch Build System)

Es un sistema tipo 'ports' para construir y empaquetar los programas a partir de su código fuente. Mientras pacman (Package Manager) es la herramienta especializada de Arch para paquetes binarios (incluyendo paquetes construidos con ABS). ABS es la herramienta especializada de Arch para compilar las fuentes y crear un paquete .pkg.tar.gz instalable estos ABS o subdirectorios no contienen el paquete del software ni las fuentes en cambio contiene un archivo PKGBUILD que es el componente mas importante para construir el paquete. El árbol de ABS se encuentra en /var/abs.

Comunidad ArchLinux Hispana

Wiki: <http://www.archlinux-es.org>

Canal IRC: #archlinux-ve en freenode

Grupo de usuarios en Venezuela:

<http://www.archlinux.org.ve>

César A. Rojas

Twitter: @ovejorock

<http://www.ovejro.com>



Sitios Recomendados



Asociación Webmaster Venezuela

<http://www.awven.com>

Sitio orientado a servir de punto de encuentro e intercambio de información de servicios profesionales en tecnología, programación, seguridad informática, software libre ente otros en forma de directorio comercial, además de brindar apoyo educativo en tecnologías

Twitter: @awvene

curioseandito.blogspot.com

<http://curioseandito.blogspot.com>

Una de las bloggers Venezolana más activas centrada en la web 2.0, consejos en el uso de las redes sociales

Twitter: @curiosa



abr4xas.org

<http://www.abr4xas.org>

Sitio con notas de interés sobre Ubuntu y otras bondades del sistema Operativo GNU/Linux

Twitter: @abr4xas



Entendiendo los JOINS

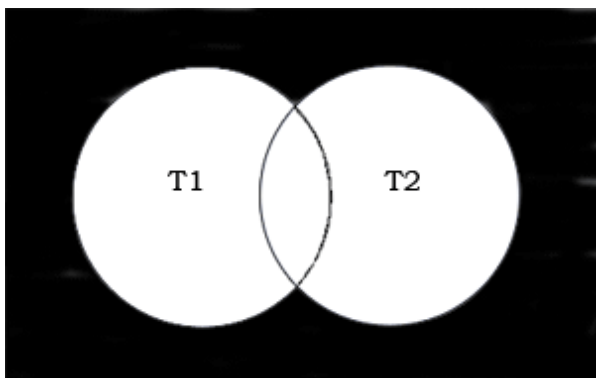
Entendiendo los JOINS con gráficos

No es de sorprender que son muchos los programadores o administradores de bases de datos quienes recién se inician, que el comprender las consultas relacionales entre distintas tablas puede resultar algo difícil de comprender, de visualizar.

Sabiendo que las imágenes se fijan mejor en nuestro cerebro que sólo leer, traigo a ustedes una adaptación del artículo escrito por mi colega Ligaya Turmelle en el año 2007 pero que cuya vigencia es válida aun hoy.

A continuación observemos estos diagramas de Venn y conozcamos los tipos de relaciones JOIN entre tablas de una base de datos.

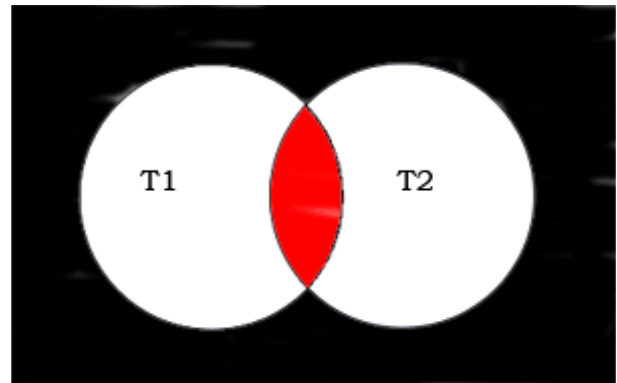
La base: un diagrama vacío:



El círculo T1 representa a todos los registros de la tabla 1 y el círculo T2 representa todos los registros de la tabla 2. Note como se sobrepone un poco en el medio, donde ambos círculos coinciden.

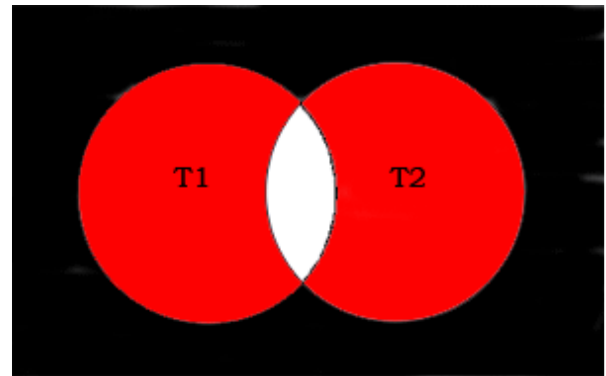
INNER JOIN

Retorna los registros donde “conciden” entre las dos tablas. Así que por cada registro devuelto de T1 habrá uno enlazado por el foreign key con T2. En programación esto equivale al operador lógico AND. El círculo T1 representa a todos los registros de la tabla 1 y el círculo T2 representa todos los registros de la tabla 2. Note como se sobrepone un poco en el medio, donde ambos círculos coinciden. al operador lógico AND.



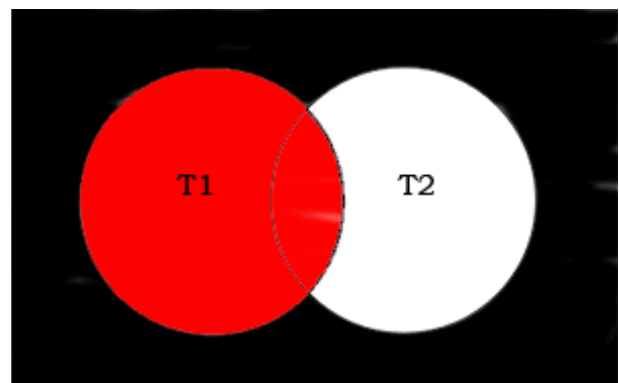
OUTER JOIN

Es lo opuesto al inner join. Sólo retorna los registros que no están en T1 y T2, es decir: donde no coinciden. En programación esto equivale al NOT AND.



LEFT JOIN

Devolverá todos los registros de la tabla izquierda (T1), sin importar si coincide o no con la tabla derecha (T2). Si llega a suceder una coincidencia con los registros en la tabla T2, pues los devolverá también. Si no, los rellenará con NULL.

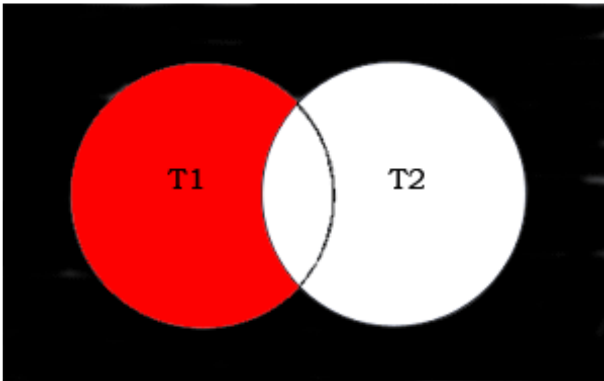


Entendiendo los JOINS

Como podrán apreciar se puede realizar exactamente lo mismo con la tabla derecha T2, y se conoce como RIGHT JOIN, sin embargo en la mayoría de los casos sólo se usa el LEFT JOIN.

LEFT OUTER JOIN

Combina los conceptos del left join y del outer join. Básicamente se obtendrán todos los registros de la tabla izquierda (T1) menos donde coincida con la tabla derecha (T2).

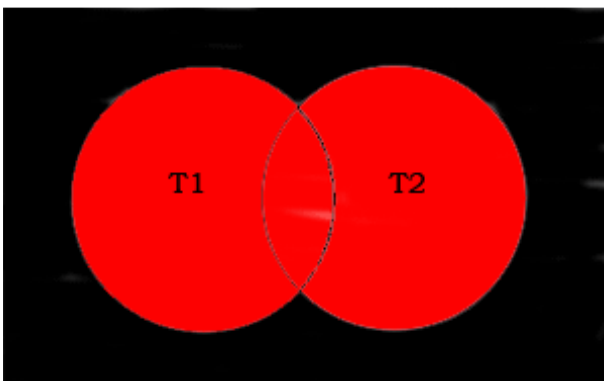


Observe que lo mismo puede ser aplicado con la tabla derecha (T2), conocido entonces como RIGHT OUTER JOIN. Aunque la mayoría prefiera referirse a la tabla izquierda siempre.

THETA JOIN

Es el producto cartesiano entre las dos tablas, y seguramente no es lo que se desea, ya que puede provenir del resultado de una mala consulta y entonces obtener más registros de los que realmente esperaba.

```
SELECT t1.*, t2.* FROM table1 t1, table2 t2  
WHERE t1.id = 5;
```



Y allí lo tienen, quizá en el futuro cuando necesiten realizar relaciones entre tablas y necesiten una pista.

Alexander Garzon
Certificado Zend PHP5
PHP de Venezuela
<http://www.php.org.ve>
Twitter: @alexandergarzon



Flisol 2011



¿Asististe al FLISoL 2010 y ahora quieres ayudar? ¿Tienes ideas para mejorar el evento? ¿Eres parte de una comunidad de Software Libre o de un Colectivo y quieres participar? ¿Eres un usuario que quiere poner su grano de arena con el evento? ¿Eres un profesional y puedes colaborar con las tareas de mantenimiento? ¿Eres un simple entusiasta y quieres aprender más?

Este año queremos crecer, pero necesitamos de tu ayuda sin importar cuanto sepas o que disponibilidad de tiempo tengas, seguro habrá una tarea simple en la que nos puedes ayudar. para conectarte al IRC: irc.freenode.org – #flisol-ve

Recuerda que lo importante es ayudar y formar parte de la experiencia...

¿Cual es la ganancia? **Disfrutar!!!**

Invita a tus amigos, familiares, compañeros de trabajo y de comunidad a esta reunión, mientras más seamos; mejor será el FLISoL 2011, Contamos contigo!!!

Flisol 2011 en Valencia - Carabobo - Venezuela

Fecha: Sábado 09 de Abril del 2011

Ubicación: Biblioteca Central de la Facultad de Ciencias de la Educación de la Universidad de Carabobo

Contacto:

Jhon Vera	john.vera1@gmail.com
Angel Cruz	webmaster@abr4xas.com
Eduardo Echeverria	echevemaster@gmail.com
Hector Mantellini	webmaster@xombra.com

Programación Segura en PHP



La programación segura de PHP es esencial para no comprometer la seguridad del servidor donde este alojada nuestra aplicación web. Una aplicación con mal diseño de seguridad es vulnerable usualmente a:

XSS: Cross-site scripting es un tipo de inseguridad informática o agujero de seguridad basado en la explotación de vulnerabilidades del sistema de validación de HTML incrustado. Esta falla suele producirse por varias razones: Variables no inicializadas correctamente, Ausencia de control de datos, entre otras.

CSRF: Cross-site request forgery o falsificación de petición en sitios cruzados, es un tipo de exploit malicioso de un sitio web en el que comandos no autorizados son transmitidos por un usuario en el sitio web que confía.

Esta vulnerabilidad es conocida también por otros nombres como XSRF, enlace hostil, ataque de un click, cabalgamiento de sesión, y ataque automático.

SQL-injection: Es una vulnerabilidad informática a nivel de la validación de las entradas a la base de datos de una aplicación. El origen es el filtrado incorrecto de las variables utilizadas en las partes del programa con código SQL. Es, de hecho, un error de una clase más general de vulnerabilidades que puede ocurrir en cualquier lenguaje de programación o de script que esté incrustado dentro de otro. Esta falla suele producirse por razones tan básicas como: Ausencia de control de datos.

RFI: Este tipo de ataque es relativamente desconocido entre desarrolladores, aunque se está extendiendo entre los entendidos de la materia, lo que hace que sea especialmente dañino. La inclusión de archivo remoto, o RFI, involucra un ataque de una locación remota que explora una aplicación PHP vulnerable e inyecta código malicioso para lograr spamming o incluso acceso a

la carpeta ruta del servidor desde la barra de navegación.

Para tener obtener un código seguro hay que tener en cuenta básicamente 2 cosas:

Sanitizar y Validar

Sanitizar y Validar los datos, son los puntos de más importancia al momento de desarrollar y programar una aplicación segura, además de las que más impacto tienen a la hora de analizar la seguridad, debido que estos datos pueden ser manipulados por cualquier usuario, en muchos casos malintencionados.

Entendamos estos dos (02) conceptos:

Que es Sanitizar? Es el proceso de aplicar una limpieza exhaustiva a un dato o grupos de datos para su uso. Por ejemplo, `FILTER_SANITIZE_EMAIL` quita caracteres que no son los apropiados que contiene una dirección email.



Que es Validar? Es el proceso utilizado para validar o comprobar si los datos cumplen con ciertos requisitos predefinidos. Por ejemplo, `FILTER_VALIDATE_EMAIL` determinar si los datos contienen una dirección válida de correo electrónico, pero sin cambiar los datos en sí.

Más sobre filtros:

<http://ve2.php.net/manual/es/book.filter.php>

Normas para un código PHP seguro

00.- Jamás confiar en los usuarios.

01.- Siempre hay que validar los datos que están bajo el control de los usuarios. Hay que tener en cuenta que un usuario malicioso puede realizar una conexión HTTP válida con un sencillo "telnet" e introducir los datos que desee y en el formato que él quiera. Nunca se debe pasar al sistema ningún dato que provenga de una entrada de usuario sin ser

Programación Segura en PHP

validada antes, y luego sanitizada. Usar las funciones `htmlspecialchars()`, `htmlentities()`, `strip_tags()` y otras funciones para limpiar el HTML y Javascript.

02.- Usar siempre `$_POST` - `$_GET` - `$_COOKIE` - `$_FILE` para capturar los datos que provengan de otras páginas de nuestra aplicación. Debemos verificar que `register_globals` este en OFF. Usar solo `$_REQUEST` a menos que la variable que se este usando no sea crítica y que la data que va en ella no sea sensible y no afecte al sistema.

03.- No usar solamente Javascript para validar datos. Javascript no ofrece una seguridad real. Siempre validen todos sus datos del lado del servidor (con PHP en este caso).

04.- Aún cuando PHP es un lenguaje permisivo es buena costumbre definir siempre las variables antes de usarlas `settype()`.

05.- Verificar que `safe_mode` este en ON (verifica si el dueño del script actual coincide con el dueño del fichero a ser operado por una función de fichero) de esta forma se comprueba que los archivos sobre los que se trabaja desde otro archivo pertenecen al mismo propietario, evitando de esta forma que otra aplicación obtenga datos sin autorización.

06.- Evitar el uso de permisos 777, para archivos y carpetas, en su lugar use 766. Para casos especiales use archivos `.htaccess`

para colocar los permisos necesarios.

07.- Almacenar las claves en formato `md5()` o `sha1()` o cualquier otro tipo de cifrado. Si es combinado mejor.

08.- Magic Quotes fue descontinuada a partir de la versión PHP 5.3.0, por ello usar `addslashes()` para minimizar el riesgo de SQL Injection o un ataque XSS.

09.- Si necesita usar tener archivos `.inc` (haciendo referencia a `includes`) o `.ini.cfg` (haciendo referencia a archivos de configuración) coloque final `.php` ejemplo: `config.ini.php`, esto es debido que algunos servidores no procesan bien ese tipo de ficheros y pueden mostrar su contenido en modo texto.

10.- Utiliza el `===` para verificar valores de entrada (identidad de datos y de tipos).

11.- Limitar las entradas solamente a los tipos de datos que deba recibir, usar funciones tales como.

`empty()` Determina que una variable está vacía
`isset()` Determina si una variable

ha sido declarada
`ctype_alnum()` Chequea posibles caracteres alfanuméricos
`ctype_alpha()` Chequea posibles caracteres alfabéticos
`ctype_digit()` Chequea posibles caracteres numéricos

12.- Deshabilitar el mostrar errores de ejecución o código. Solo se debe mostrar los errores en fase de desarrollo de la aplicación, jamás en producción:
`ini_set('display_errors', 0);`
También puede usar archivos `.htaccess` y colocar:
`php_value display_errors 0`
`php_value display_startup_errors 0`

13.- Usar expresiones regulares (Las expresiones regulares son una serie de caracteres que forman un patrón, normalmente representativo de otro grupo de caracteres mayor, de tal forma que podemos comparar el patrón con otro conjunto de caracteres para ver las coincidencias.). Por ejemplo usar para sanitizar:
`preg_match()` Realiza una comparación con una expresión regular
`preg_replace()` Realiza una búsqueda y sustitución de una expresión regular.

14.- Para evitar el RFI verificar las directivas:

`allow_url_fopen`
`allow_url_include`

La directiva `allow_url_fopen` esta activada por defecto, y `allow_url_include` está desactivada. Estas dos directivas simples protegerán adecuadamente tu sitio de cualquier ataque RFI. También pueden usar archivo `.htaccess`



Programacion Segura en PHP

15.- Verificar la data en las sentencias SQL
(MySQL)mysql_real_escape_string() Escapa
ciertos caracteres de una cadena en una sentencia
SQL

Error e inseguro:

```
$query = "SELECT * FROM users WHERE  
user='{$_POST['user']}' AND  
pass='{$_POST['pass']}'";
```

Correcto y seguro:

```
$query = sprintf("SELECT user,name FROM users  
WHERE user='%s' AND  
pass='%s'",mysql_real_escape_string($_POST['user']  
''),mysql_real_escape_string($_POST['pass']));
```

16.- Evitar el uso el comodín * en las sentencias
SQL

17.- Sesiones y Cookies. Es preferible usar
SESSION por 2 razones: las cookies requieren que
el navegador soporte cookies; y en las cookies
estas enviando todos los datos del servidor al
navegador y de vuelta en cada visualización, con lo
que ello conlleva (seguridad, limite de tamaño de
cabeceras de peticiones, y un largo etc.).

Con las sesiones solo tienes que tener cuidado de
que no se pierda el ID de session. El contenido de
la session no sale del servidor.

18.- Usar urlencode y urldecode

Codifica/Decodifica como URL una cadena

```
$url =  
"http://www.ejemplo.com/index.php?q=".urlencode  
($_GET['var']);
```

19.- Agregar estas línea en el .htaccess

RewriteEngine on

Block out any script that includes a <script> tag
in URL

```
RewriteCond %{QUERY_STRING}  
(\<|%3C).*script.*(\>|%3E)[N,C]
```

Héctor A. Mantellini (Xombra)

Twitter: @xombra

<http://www.xombra.com>



Proyecto Factible

Tomando como referencia a lo expuesto como concepto de proyecto factible en el Manual de Trabajos de Grado y Maestría y Tesis Doctoral de la Universidad Pedagógica Experimental Libertador (1998: 7):

El Proyecto Factible consiste en la elaboración de una propuesta de un modelo operativo viable para solucionar problemas, requerimientos o necesidades de organizaciones o grupos sociales; puede referirse a la formulación de políticas, programas, tecnologías, métodos o procesos. El proyecto debe tener apoyo en una investigación de tipo documental, de campo o un diseño que incluya ambas modalidades.



Extrapolándolo al desarrollo de propuestas o productos en el ámbito de la tecnología educativa, el autor del presente texto expositivo conceptualiza mediante autores los vocablos que se involucran en esta relación.

Desarrollo (Proyecto): Como lo define Álvarez (2002: 49) tiene que ver con: El contenido del proceso de planeación (identificación de problemas y necesidades, formulación de objetivos o metas) y con la participación de los sectores interesados en esta identificación y formulación, con la instrumentación (selección de medios, acciones y estrategias), con la organización y gestión del proceso (coordinación y distribución de funciones) y con la evaluación (verificación de logros alcanzados y recuperación de experiencias).

Producto: Según Kotler (2003: 6) define que un producto es “cualquier ofrecimiento que puede satisfacer una necesidad o un deseo, y podría ser una de las 10 ofertas básicas: bienes, servicios, experiencias, eventos, personas, lugares, propiedades, organizaciones, información e ideas.

Tecnología Educativa: Según como lo conceptualiza la UNESCO (1984: 3) El nuevo significado dado al término “tecnología educativa”, en el más amplio sentido como “el modo sistemático de concebir aplicar y evaluar el conjunto de procesos de enseñanza y aprendizaje teniendo en cuenta los recursos técnicos y

humanos y las interacciones entre ellos, como forma de obtener una más efectiva educación”.

Y pretende exhibir su propia definición de Proyecto Factible centrado en el desarrollo de productos tecnológicos adaptados a la educación el cual se expone a continuación.

Consiste en la planeación, selección de estrategias, organización y evaluación de una propuesta viable que busque satisfacer las necesidades de un grupo u organización educativa con el fin de aplicarlos en los procesos de enseñanza y aprendizaje utilizando recursos tecnológicos y humanos.

Tomando en cuenta el proyecto factible como modalidad de investigación esencial en el desarrollo de productos en el campo de la tecnología aplicada a la educación se pueden definir las siguientes características o rasgos:

Resolver problemas educativos de tipo práctico promoviendo y proponiendo esquemas de acción de tipo tecnológico fruto de experiencias teóricas anteriores.

Según Carr (2002:109) “Los problemas educativos por ser mayormente de tipo práctico no se rigen por normas de la investigación teórica sino que surgen cuando las prácticas utilizadas en las actividades educativas resultan de algún modo inadecuadas con respecto a su fin. En otras palabras aparecen cuando hay una discrepancia entre una práctica educativa y las expectativas con las que se emprende la práctica en cuestión”. Por tanto, todos los problemas educativos son origen de un distanciamiento entre la teoría y la práctica; el uso de herramientas tecnológicas como apoyo y no como un fin, para actividades que impliquen “hacer” dará como resultado la efectiva aplicación de la teoría.

Esta resultante teoría “aplicable”, es producto de los resultados de cualquier proyecto factible que implique la resolución de un problema o necesidad de índole educativa-tecnológica. Tratar un tema implicando todos sus componentes no obviando sus relaciones perceptibles o tangibles.

Proyecto Factible

Proporcionar una visión holística de las variables existentes en los problemas o necesidades educativas tomándolos con carácter global y no haciendo tratamientos reduccionistas e inconexos.

La obtención de un producto o investigación educativa como consecuencia final y factible de beneficio social.

El beneficio de un producto educativo siempre es el aprendizaje y la obtención de nuevos constructos que impliquen la transformación social de los individuos objetos de la enseñanza.

Haciendo una reflexión sobre la importancia de la utilización del Proyecto Factible en el campo de la tecnología aplicada a la educación se manifiesta como principal objetivo el realizar cambios mediante la investigación de fenómenos en el contexto educativo en el cual estamos inmersos, apoyándonos como docentes en las posibilidades que brinda la tecnología como entorno de adaptación y como vínculo de satisfacción de las necesidades de nuestros alumnos mediante competencias y destrezas generales que los ayuden en su vida cotidiana.

Referencias Bibliográficas:

- Álvarez, I (2002): Planificación y desarrollo de proyectos sociales y educativos. Editorial Limusa, 2002.
- Carr, W (2002): Una teoría para la educación: hacia una investigación educativa crítica. Ediciones Morata, 2002
- Kotler, P: Dirección de marketing: Conceptos esenciales. Pearson Educación, 2003
- Unesco (1986): Glossary of educational technology terms. Unesco. Methods, Materials, and Techniques of Education Section, International Bureau of Education
- UPEL (1998): Manual de Trabajos de Grado y Maestría y Tesis Doctoral de la Universidad Pedagógica Experimental Libertador. Caracas: UPEL.

Por:

Eduardo Echeverría (Echevemaster)

Twitter: @echevemaster

Código de Conducta UBUNTU

Hola a todos, aunque he querido realizar un artículo para esta revista desde hace algún tiempo, por razones adversas y diversas no lo he podido concretar hasta este momento; Pero entonces nació una interrogante... ¿Acerca de que puedo escribir?... Pues por mi mente pasan miles de ideas y cosas a las cuales me gustaría enfocar en esta revista. Pero hace poco sostuve una conversación con un usuario novel de GNU/Linux interesado en buscar características en las distribuciones GNU/Linux que le permitan obtener un mayor conocimiento del tema, y fue así cuando puso “en mesa” el “Código de Conducta de Ubuntu”.

Más o menos sus palabras fueron las siguientes:

...Es resaltante que una organización de una distribución GNU/Linux se tomará como necesario y normativa crear e implementar a sus afines una serie de “Recomendaciones”, para soportar su organización entre otras cosas...

... Y mas aún cuando lees el Código de Conducta, ves que no es otra cosa que normativas de convivencia y educación que todos deberíamos ya de saber, por ende no pensé que fuese algo tan simple y sencillo de comprender. Y que para sus mas entusiastas lleven a cabo un procedimiento con sus sistemas para “firmar” dicho Código de Conducta...

Pues, a pesar que nuestra amena conversación el amigo se tuvo que retirar, y pues yo aprovecho de responderle mientras escribo este artículo.

Si bien es cierto que el Código de Conducta ha sido adoptado de una manera muy significativa por la organización de la distribución Ubuntu (Canonical Ltd). Es un tema largo de hablar, pues tendríamos que explicar muchos aspectos diferentes para comprender tal decisión, y hasta entrevistar a quienes decidieron el mismo. A la vez es simple con saber lo que significa la palabra Ubuntu. Que en su diversidad de traducciones posibles se reduce al compartir, convivir y saber aprovechar esas características.

Entonces la otra interrogativa es ¿Es funcional, Sirve de algo?. Sin duda alguna que sí.

A veces como humanos y seres tan diversos solemos ignorar ciertas cosas que deberían estar plantadas en nuestras bases como seres maduros y racionales. Sin embargo, cuando perteneces a un grupo de personas y tienes estatutos los cuales rigen en menor o mayor grado un comportamiento que deberías cumplir. Es simple tienes algo que lo recuerde y quienes te lo recuerde, y si en algún momento estas de uno u otro lado, pues tienes una base para hacerte oír, escuchar a otros y respetarse todos.

En mi experiencia personal, entre miembros de la comunidad Ubuntu-Venezuela en el Estado Carabobo, hemos tenido momentos no muy gratos por la diversidad de opiniones y demás influyentes. En donde no sabemos como detener una división por alguna discordia de algún grupo o algunas personas, y es allí donde nos apegamos al Código de Conducta para seguir los lineamientos que haga cesar la discordia en cuestión de una manera

racional justa y balanceada y responsable, donde todos tienen el mismo derecho y deber de opinar, escuchar y trabajar en alguna labor. Y si por alguna casualidad la discordia o problemática continua, el mismo CoC. Indica Renuncie Consideradamente.

Otro uso que se le da al “CoC” (Código de Conducta). Es si eres parte de algún proyecto o participas de alguna manera en actividades que por alguna razón no puedes continuar, puedes delegar ese trabajo a otr@(s) y así no dejar huérfano alguna labor o proyecto.

En Ubuntu-ve, por ser la Comunidad Local aprobada por Canonical Ltd para representar a Ubuntu en mi país, pues sin duda alguna estamos apegados a ese CoC. Y la cual nos ha permitido, coexistir entre miembros, comunidades y como organización. Si alguna persona mas allá de ser un entusiasta o usuario de Ubuntu quiere pertenecer a Ubuntu o algunos de sus proyectos o a algunas de sus comunidades, lo mas probable es que ese entusiasta adquiera un compromiso un poco mas formal y que a su vez mas participativa. Por consiguiente es posible que para lograrlo se exija la firma digital del CoC. Que a demás de ser unas normativas que aceptas cumplir es una firma digital que te hace parte de Ubuntu y su filosofía, y te permite realizar y coordinar algún proyecto



Código de Conducta UBUNTU

o labor como un usuario de mayor “Jerarquía”.
Que el entusiasta común.

De como firmar el CoC. Les hablaré en mi próxima entrega.

Adjunto traducción del CoC. Disponible en el portal de Ubuntu-ve <http://www.ubuntu-ve.org>

Este código de conducta cubre su comportamiento como miembro de la comunidad Ubuntu, en cualquier foro, lista de correo, wiki, sitio web, Canal IRC, install-fest (festival de instalación), reunión pública o correspondencia privada. El Consejo de la Comunidad Ubuntu arbitrará en cualquier discusión sobre la conducta de un miembro de la comunidad.

* Sea considerado. * Su trabajo será usado por otras personas, y usted a su turno dependerá del trabajo de otros. Cualquier decisión que usted tome afectará a usuarios y colegas, y esperamos que usted tome en cuenta las consecuencias al tomar decisiones. Por ejemplo, cuando estemos en congelamiento de características, por favor no suba bruscamente nuevas versiones de software crítico del sistema, ya que otras personas estarán probando el sistema congelado y no esperan grandes cambios.

* Sea respetuoso. * La comunidad Ubuntu y sus miembros se tratan unos a otros con respeto. Cualquiera puede hacer una valiosa contribución a Ubuntu. Podemos no siempre estar de acuerdo, pero los desacuerdos no son excusa para el mal comportamiento y los malos modales. Todos podemos experimentar algo de frustración de vez en cuando, pero no podemos dejar que la frustración se vuelva en ataque personal. Es importante recordar que una comunidad donde la gente se siente incómoda o amenazada no es productiva. Esperamos que los miembros de la comunidad Ubuntu sean respetuosos cuando se traten con otros contribuidores y también con personas fuera del proyecto Ubuntu, y con los usuarios de Ubuntu.

* Sea Colaborador. * Ubuntu y Software Libre son acerca de la colaboración y el trabajo mutuo. La colaboración reduce redundancia de trabajo

hecho por el mundo del Software Libre, y mejora la calidad del software producido. Usted debería apuntar a colaborar con otros mantenedores Ubuntu, así como también con la comunidad más avanzada que está interesada en el trabajo que usted hace. Su trabajo debería ser hecho transparentemente y los parches de Ubuntu deberían ser devueltos a la comunidad cuando estos sean hechos, no solo cuando hay liberaciones de la distribución. Si usted desea trabajar en un código nuevo para proyectos existentes mas avanzados, al menos mantenga esos proyectos informados de sus ideas y progreso. Podría no ser posible llegar al consenso de mas avanzado o incluso de sus colegas sobre la correcta implementación de una idea, así que no sienta la obligación de tener la aprobación antes de empezar, pero al menos mantenga al mundo exterior informado de su trabajo, y publique su trabajo de manera que permita a personas ajenas probar, discutir y contribuir a sus esfuerzos.

* Cuando tenga desacuerdos. * Consulte a los demás. Los desacuerdos, tanto políticos como técnicos, ocurren todo el tiempo y la comunidad Ubuntu no es la excepción. La meta importante no es evitar los desacuerdos o los diversos puntos de vista sino resolverlos constructivamente. Usted debería dirigirse a la comunidad y al proceso comunitario para buscar consejos y resolver desacuerdos. Tenemos la Mesa Técnica y el Consejo Comunitario, ambos de los cuales le ayudarán a decidir el camino correcto para Ubuntu. También hay muchos Equipos de Proyecto y Líderes de Equipo, quienes podrían ayudarle a encontrar cual dirección sería la más aceptable. Si usted en verdad quiere tomar un camino distinto, entonces le animamos a crear una distribución derivada o hacer disponible un conjunto alternativo de paquetes usando la estructura de Administración de Paquetes Ubuntu, de manera que la comunidad pueda probar sus cambios e ideas para si misma y contribuir a la discusión.

* Cuando tenga inseguridad. * Pida ayuda. Nadie sabe todo, y no se espera que nadie sea perfecto en la comunidad Ubuntu (exceptuando por supuesto al ANDBPV [SABDFL]). Hacer preguntas evita



Código de Conducta UBUNTU

luego muchos problemas en el camino, por lo tanto se le anima a preguntar. Aquellos a quienes se les pregunte deben ser receptivos y colaboradores. Sin embargo, al hacer una pregunta, debe tomarse cuidado para hacerlo en el foro apropiado. Las preguntas fuera de tópico, como peticiones de ayuda en una lista de correo para desarrollo, restan valor a la discusión productiva.

* Renuncie consideradamente.

* Los desarrolladores en todo proyecto van y vienen y Ubuntu no es diferente. Cuando deje o se suelte del proyecto, en parte o en todo, le pedimos que lo haga de manera que minimice la interrupción al proyecto. Esto significa que usted debería decirle a las personas que se va y tomar los pasos para lograr que otros puedan tomarlo donde usted lo deja.

Traducido en Ubuntu-ve desde la versión original en inglés:

<https://launchpad.net/codeofconduct/1.0.1>

<http://ubuntu.org.ve>



Gracias nos vemos en la próxima entrega

Ing. David Emerling Rondón
Coord. Region Centro
Team Carabobo - Ubuntu Venezuela
emerling@ubuntu.org.ve
<http://carabobo-ubuntu.org.ve>
Twitter: @emerling