

SL

EL SOFTWARE LIBRE
HECHO REVISTA

www.revista-sl.org

Edición 5, octubre 2006

XML

A la conquista del mundo

LAMP .NET

No todo es PHP

Web Services

Haciendo versatil la Web

Proyectos SL

Suricata y Free Fall

FS 2006

La aventura continua

En la mira....

FCK Editor

DESARROLLO WEB



En entrevista: Fabien Mannisser
el experto nos platica de Ruby on Rails



Attribution 2.5 Mexico

You are free:

- ◆ to copy, distribute, display, and perform the work
- ◆ to make derivative works
- ◆ to make commercial use of the work

Under the following conditions:



Attribution. You must attribute the work in the manner specified by the author or licensor.

- ◆ For any reuse or distribution, you must make clear to others the license terms of this work.
- ◆ Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 



RevistaSL

El software libre hecho revista

~# Editorial SL



La comunicación entre individuos es una de las principales necesidades de la sociedad desde muchos cientos de años atrás, esta necesidad nos ha llevado a crear soluciones de comunicación a lo largo del tiempo y de esta manera hoy en día Internet es uno de los medios de comunicación indispensables. Internet esta compuesto de una gran diversidad de protocolos, herramientas, tecnologías y mas.

Los miembros de la Revista SL creemos que es indispensable el conocimiento de tecnologías de Internet para la comunidad, todas aquellas que son aplicables al comercio electrónico, así como todas aquellas que son aplicables a la educación y al gobierno.

El desarrollo Web en la actualidad es algo muy sencillo, contamos con muchas posibilidades para obtener resultados que resuelvan problemas desde los mas sencillos, hasta los mas difíciles, y el software libre tiene un protagonico en esto, pues no solo los estadísticos figuran en favor del uso del software libre para web, sino que es la tendencia a seguir de una gran cantidad de organismos, empresas e instituciones educativas a lo largo del mundo.

Héctor Leal Morales, "etosh"
hector.leal@revista-sl.org

a nombre de...

Gonzalo González Rodríguez

Dirección Editorial SL

Tec. Gonzalo J. González Rodríguez

Coordinación Editorial SL

Carlos Augusto Lozano Vargas

Editores SL

Lic. Julio Acuña Carrillo

Tec. Víctor H. Cordova Madrid

Lic. José L. Galicia Sánchez

Lic. Jesus A. Balam Jiménez

Tec. Julio César Sosa Yeladaqui

Diseño & Imagen SL

DG Edgar Guerra

TSU Josue Gutierrez Hernández

Tec. Eyden Barboza Barela

Mirones SL

ICE Héctor Leal Morales

LIA David Arroyo Arias

Lic. Luis F. Peniche Novelo

Lic. Mauro Parra Miranda

Lic. Raquel Hernández Padilla

Buzón SL

buzon@revista-sl.org

Colaboraciones

colaboraciones@revista-sl.org

Publicidad

publicidad@revista-sl.org

contenido

Desarrollo Web



► Nuevas SL:

¿Sabias qué? pag. 3
Java Libre pag. 8

► Vistazo SL:

FCK Editor pag. 9

► Programación SL:

Desarrollando aplicaciones web pag. 11
Desarrollando web con LAMP .NET pag. 14
Transacciones a una BD con PearBD pag. 19

► Web SL

Web services pag. 30

► Proyectos SL:

Suricata pag. 43
Free Fall pag. 45

XML

a la conquista del mundo
pag. 27

En entrevista:

Fabien Manniser
pag. 49

Graficación

JPGraph y PHP
pag. 23

Desarrollo Web

Mejorando la web
pag. 33

Comunidades:

Chetulinuxeros
pag. 46

¿sabias que?

Tec. Julio César Sosa Yeladaqui
julio.cesar@revista-sl.org



1.- ¿Sabias que la mascota del navegador Mozilla Firefox® o simplemente Firefox no es un Zorro Rojo si no que en realidad es un Panda Rojo?

Así es, por increíble que parezca dicho personaje es en verdad un Panda Rojo a diferencia de lo que todos a primera vista creemos al ver el logotipo que lo incluye y al leer o escuchar el nombre.

Panda Rojo: (*Ailurus fulgens*), Es un pequeño mamífero que se alimenta principalmente de hierbas este singular amigo es nativo de los Himalayas pero también se puede encontrar en China algunas partes de Nepal e India. También es conocido como Panda menor, Gato Oso, Wah y Firefox (Zorro de fuego).

Esta afirmación queda de manifiesto por la Mozilla Corporation® y Mozilla Foundation al cuestionar, ¿Qué es un Firefox? y responder que hace referencia a otra forma de nombrar al Panda Rojo. Cuya respuesta se incluye en este pequeñísimo extracto obtenido del "Mozilla Firefox - Brand Name FAQ"

"What's a Firefox?
A "Firefox" is another name for the red panda."



2.- ¿Sabias que el Kernel - Linux en un principio se llamó FREAX-kernel?

Linus Benedict Torvalds creador del Linux-kernel y padre junto con la Free Software Foundation organización creada por Richard Stallman del sistema operativo Linux o GNU/Linux. Al comienzo Torvalds peso en llamar FREAX al kernel en cuestión, basándose en las palabras Free "libre" y "gratis", Freak "diferente - raro" sin olvidar la X que hace alusión al hecho de estarse tratando de un sistema operativo tipo UNIX.

El FREAX-kernel dejo de llamarse como tal debido a la determinante intervención de Ari Lemmke

quien fue el que puso por primera vez a disposición de todo el mundo el Kernel-Linux por medio de un servidor de FTP y quien a su vez ayudó a convencer a Torvalds de publicarlo bajo la licencia GPL, Lemmke opinaba que el nombre de FREAX era poco atractivo comercialmente hablando por lo que al momento de publicar el kernel decidió nombrar Linux a la carpeta que contenía las fuentes del mismo dando así origen al nombre por todos conocido "LINUX".

Esta afirmación queda de manifiesto de viva voz por parte de Torvalds y Lemmke en el documental Codename: LINUX así como también en el Oldlinux Cross Reference, sitio dedicado a recopilar por cuestiones de conservación histórica todo aquello referente a linux, principalmente las versiones de kernel anteriores a la versión 1.0, distribuciones antiguas y documentación en general que pueda significar un aporte a la memoria histórica de todo aquello que se considera "Old Linux".

En este extracto obtenido del Archivo Makefile utilizado para automatizar el proceso de compilación de las fuentes del kernel, se muestra la versión incluida en el Kernel 0.01, la primer versión del kernel que Torvalds publico el 17 de Septiembre de 1991 fecha en que Linux como Núcleo o Kernel vería la luz.

```
#
# Makefile for the FREAX-kernel.
#
# Note! Dependencies are done
automagically by 'make dep', which also
# removes any old dependencies. DON'T
put your own dependencies here
# unless it's something special (ie not
a .c file).
```

<http://oldlinux.org/Linux.old/kernel/0.00/linux-0.01/kernel/>
<http://oldlinux.org/>

3.- ¿Sabias que existe una Cerveza Open Source?

Pues así es, un grupo de entusiastas daneses inspirados en la corriente del software libre, decidieron crear una receta para fabricar cerveza

con la intención de hacerla libre. La decidieron llamarla Vores Øl, que en danés significa "Nuestra cerveza". Dicha cerveza fue creada por estudiantes de TI de la Universidad de Copenhague, junto con apoyo de Superflex un grupo de artistas dedicados a la realización de proyectos de producción. Cabe destacar que los ingredientes no son del tipo "se consiguen en la tienda de la esquina", es probable que se tenga que recurrir a una tienda especializada por ejemplo para obtener los diferentes extractos de malta que la componen o algunos frutos como el guaraná agregado en algunas subversiones de de la receta.

También existe el proyecto FREE BEER basado en la Vores Øl pero con la idea de mantener el concepto de cerveza libre así como recopilar nuevas recetas derivadas o totalmente nuevas. Actualmente el proyecto se encuentra en la versión FREE BEER v 3.0 Free beer vendría siendo algo así como una Meta-distribución creando una especie de analogía con respecto a las distribuciones o Metadistros Linux.

La Cerveza Libre Vores Øl así como la FREE BEER derivada de la anterior son publicadas bajo la licencia Creative Commons bajo el formato "Atribución-Licenciamiento Recíproco" la cual nos brinda las siguientes facultades.

- * copiar, distribuir y comunicar públicamente la obra
- * hacer obras derivadas
- * hacer un uso comercial de esta obra

Receta para la "Vores Øl"

Esta receta se ha obtenido de la página web oficial (en septiembre de 2005), aunque es posible que existan versiones más recientes y mejoradas, dado su carácter de código abierto.

Receta para unos 85 litros (aproximadamente, un volumen de alcohol del 6%):

- Extracto de malta

Se emplearán cuatro tipos diferentes de cebada malteada:

- o 6 kg de malta de Pilsen
- o 4 kg de malta münster
- o 1 kg de malta de caramelo
- o 1 kg de malta rubia (lager)

- Se machaca la malta y se pone en agua caliente (55–60 °C) durante 1 ó 2 horas.

- Se filtra la mezcla y el líquido resultante contiene unos 10 kilos de extracto de malta.

- Sabor y dulzura (aquí se pudieran omitir algunos de los ingredientes como el lúpulo de Hallertau y lúpulo Tettang agregando únicamente el guaraná o en su defecto utilizar simplemente azúcar para endulzar).

- o 50 g de lúpulo de Hallertau
- o 60 g de lúpulo Tettang
- o 300 g de fruto de la guaraná
- o 4 kg de azúcar

- Se pone el extracto de malta a hervir en un recipiente grande con el lúpulo de Hallertau y aproximadamente 70 l de agua.

Tras hora y media, agregar la guaraná y el azúcar.

Calentar la mezcla a fuego lento durante una hora más o menos, apagar el fuego agregar el lúpulo Tettang. Dejar reposar 10 minutos.

- Filtrar la mezcla y enfriar en un recipiente cerrado.

Fermentación:

Agregar la levadura y dejar fermentar la cerveza a temperatura ambiente durante unas dos semanas.

Cuando esté fermentada, embotellar. Añadir antes 4 g de azúcar por cada litro, y una pizca de levadura del fondo del recipiente de fermentación para cebarla.

Dejar la cerveza en botellas a temperatura ambiente durante 8 o 10 días para que se carbonate.

- Salud!!

<http://www.voresoel.dk/>
<http://www.freebeer.org/>
http://es.wikipedia.org/wiki/Vores_%C3%98l

4.- ¿Sabías que La Maquina del Tiempo Existe?

Bien pues no es mentira, realmente existe ;) sólo



que ésta, es una maquina del tiempo virtual que únicamente viaja al pasado. Corría el año de 1996 cuando Brewster Kahle un empresario dedicado a los negocios en Internet, abrigó la idea de hacer una recolección de los sitios web existentes en la red, dando origen a uno de los mayores, si no es que el mayor acervo cronológico alusivo al contenido en la Red, indexando y recopilando hasta nuestros días al rededor de 55,000,000,000 (Cincuenta y cinco mil millones) de paginas web, y eso no es todo, también guarda registro de textos importantes, audio, vídeo, software e imágenes. Para muestra basta un botón observemos el siguiente ejemplo.

- Ingresamos al sitio <http://www.archive.org/>

- Ubicamos The Wayback Machine e ingresamos la url que deseamos explorar para nuestro caso es <http://www.unam.mx> sitio oficial de la Universidad Nacional Autónoma de México. (no siempre enviará resultados ya que el funcionamiento depende de la recopilación e indexado periódico de la base de datos de Internet Archive, por lo tanto, es posible que determinados sitios aun no hayan sido incluidos).

5.- ¿Sabías que en 1961 el mundo que conocemos pudo haber llegado a su fin? (en sentido figurado errrrmmmmmm!!! o tal vez no tanto)

El 30 de Octubre de 1961 fue detonada la bomba nuclear más grande jamás construida, cuando menos de la que se tenga registro, dicha bomba fue desarrollada por la antigua Unión Soviética (URSS) hoy conocida como Rusia en tiempos de la “Guerra fría” con el único objetivo de demostrar la superioridad tecnológica y militar soviética o lo que en política se conoce como “Mostrar el musculo”, la bomba en cuestión fue llamada “**Tsar Bomba**” o Bomba Zar (Bomba Rey, Bomba Emperador, algunos lo traducen como “Bomba del Zar”) haciendo alusión a la grandeza de dicho artefacto así como al inmenso orgullo Ruso, la bomba también recibió el nombre código de “**Iván**”.

Iván fue una bomba de hidrógeno con un poder explosivo en base a su diseño inicial de alrededor de 100 megatones, poder entre 5000 y 6600 veces mayor que las bombas de Iroshima y Nagasaki, momentos antes de la detonación decidieron disminuir su poder a 50mt, debido que algunas personas involucradas en el proyecto consideraron que detonar una bomba de tal magnitud podría resultar peligroso para la región ocasionando una serie de problemas ambientales aunado a que habría una mínima posibilidad de huir de la explosión por parte de la tripulación de la Aeronave Tupolev Tu-95 avión en donde fue transportado y liberado Iván.

Océano Glacial Ártico, Octubre 30 – 1961
Archipiélago de Nueva Zembla (URSS)
11:30 a.m.

El avión de Carga Tupolev Tu-95 sobrevuela el ártico a 10.5 km de altura cuando de pronto tal vez el arma mas poderosa que jamás el hombre ha concebido era expulsada desafiante, el paracaídas se abre para retrasar la inminente caída, la tripulación del Tu-95 abandona la escena a toda marcha mientras el tiempo parece haberse detenido, los segundos se hacen horas hasta que repentinamente a 4200 metros sobre la superficie marina se suscita la implacable explosión.

Una columna de aproximadamente 64km de altura con forma de hongo se yergue imponente sobre el Océano Ártico con un diámetro aproximado de 35km de ancho. El efecto luminoso pudo verse a mas de 1000 km de distancia, el “infierno en la tierra” ya que el poder calorífico de la Bomba pudo haber causado quemaduras de tercer grado incluso a 100 km de distancia del epicentro, las ondas sísmicas producidas por la explosión fueron registradas en todas partes del mundo.

Recordemos un poco y saquemos conclusiones: (La Siguiete Información pudiera no ser exacta, así como perturbadora más sin embargo nada lejana a la realidad.)

El 6 de Agosto de 1945 en tiempos de la segunda guerra mundial Los Estados Unidos de América decidieron bombardear las Ciudades de Iroshima y Nagasaki basta mencionar que dichos bombardeos causaron la perdida de entre 400,000 y 500,000 vidas humanas entre ellos El ejercito aunado a hombres, mujeres y niños inocentes quienes fueron la mayoría en ese numero de bajas algunos murieron al momento de la detonación otros otros tantos murieron paulatinamente por la gravedad de sus heridas así como por los efectos la radiación que les causó infinidad de cánceres entre otros motivos relacionados con los bombardeos. Cabe mencionar que las dos bombas usadas tuvieron un poder de 16 y 25 kilotonnes respectivamente, Little Boy en Iroshima y Fat Man en Nagakazi.

Ahora que ya tenemos una referencia, no es difícil deducir que una bomba Como “Iván” tiene el poder de acabar con una ciudad como Los Ángeles en California Estados Unidos, una ciudad de aprox. 17.2 millones de habitantes y una extensión geográfica de 1200 km² o viéndolo de otro modo para su fácil visualización un circulo de aproximadamente 42.72 km de diámetro tomando la idea de un circulo como referencia al campo de acción geográfico de la ciudad. El “infierno en la tierra” ya que el poder calorífico de la Bomba pudo haber causado quemaduras de tercer grado incluso a 100 km de distancia del epicentro. Creo que no hace falta entrar en detalles para conocer el final. =(

La tecnología ha avanzado aún más y es posible que existan o se puedan desarrollar armas de mayor poder destructivo. Hoy en día cualquier país que posee la tecnología y conocimiento del uso de la energía atómica o nuclear es capaz cuando

cuando menos en teoría, de poder crear armas nucleares. 🤖

<http://nuclearweaponarchive.org/>

http://es.wikipedia.org/wiki/Bomba_Tsar

The largest nuclear weapon ever: Tsar Bomba 50 Megatons

<http://www.youtube.com/watch?v=16cewjeqNdw>

Esta es una nueva sección con la que queremos poner un poco de humor, curiosidades y mucho más. Esperamos sus comentarios, ideas o aportaciones a buzon@revista-sl.org

java libre

Julio Acuña Carrillo
julio.acuna@revista-sl.org



El pasado 13 de noviembre Sun Microsystems anunció un cambio de licencia para Java, una de las tecnologías con mayor uso en los últimos años.

La comunidad del software libre había abogado durante mucho tiempo para que Java se liberara, incluso hubo varios intentos de implementar máquinas virtuales y compiladores libres usando las especificaciones de Sun, pero ninguno de esos intentos alcanzó todas las especificaciones o superó en rendimiento a la máquina original, es por eso que cada vez que se anunciaba que Java sería libre todo el mundo miraba el anuncio con escepticismo.

La noticia de que Java por fin era libre fue impactante, pero lo fue aún más cuando se anunció que la licencia escogida era la GPL versión 2 y no la CDDL que el mismo Sun inventó para OpenSolaris, ahora también se está considerando cambiar la licencia de este último por la GPL.

Para los más puristas la trampa de Java ha dejado de existir, el dilema de programar software libre en una tecnología no libre como lo era Java hace algunos días se ha disipado y sin duda esto atraerá a mucha gente hacia el software libre, y muchos lenguajes libres con máquinas virtuales tendrán la oportunidad de beneficiarse de lo que hotspot tenga que ofrecer, algunas ideas podrían implementarse para mejorar su desempeño.

pasará con proyectos como Kaffe y gcj, sin mencionar todo el software hecho por Apache Software Foundation con especial interés en Geronimo, sin duda será interesante ver su evolución, pero al final todo el mundo saldrá beneficiado.

Sin duda estamos ante acontecimientos que cambiarán la forma en que la gente ve al software incluso cuando haya recelo en algunas acciones tomadas por las grandes empresas como la asociación entre Novell y Microsoft para dar soporte en SUSE y Windows respectivamente, al sistema operativo del otro, que en este momento nos parece como si el mundo estuviera de cabeza pero no podemos más que esperar a ver cómo se dan las cosas, por supuesto estando muy atentos en proteger nuestras libertades. 🇪🇺

FCK editor

Ing. Héctor Leal Morales
hector.leal@revista-sl.org



Fue hace unos años que leí acerca de las aplicaciones de oficina en línea, encontré algunos textos y también pude probar alguna, redacté algunos documentos en el editor, a modo de pruebas, y el comportamiento fue muy aceptable. Me agradó la posibilidad de tener el software en línea y no tener que tenerlo instalado en mi máquina.

Al poco tiempo un cliente me preguntó si podía hacer eso mismo pero en los formularios con los cual él generaba sus páginas, Yo le dije que si se podía, pero que jamás no lo había intentado, entonces me solicitó algunas mejoras a una interfaz que él ya tenía, y entre ellas me habló de su inquietud por una herramienta como esa, con la cual él tuviera la capacidad de darle formato a sus páginas web, tal y como él lo hacía en su aplicación de oficina.

Antes de intentar programar algo, me puse a buscar ¿quién utilizaban otras personas para cubrir esta necesidad?, y bueno, en la búsqueda me encontré con una herramienta muy fácil de utilizar y muy agradable para el usuario final.

FCKeditor es un editor de HTML que se puede incluir de una forma muy sencilla en sitios web. Maneja diferentes tipos de licencia, LGPL y Comercial. Entre sus características destacan el soporte de diferentes agentes de usuario, salida XHTML, CSS, cortar y pegar, soporte de links, soporte de imágenes, soporte de tablas, entre otras, y también tiene algunas características como diferentes barras de herramientas,

personalizables, soporte de skins y plugins, soporte de idioma y se puede implementar con diferentes lenguajes como ASP, JAVA, PHP, etc Se puede descargar la última versión desde <http://www.fckeditor.com> y obtener mucha más información, también hay un wiki en <http://wiki.fckeditor.net>

La implementación de FCKeditor en una aplicación es muy sencilla. Una vez que se descarga el software debes de poner los archivos necesarios dentro del alcance de tu aplicación. Generalmente dentro de `public_html`

Entonces si el software quedó de la siguiente forma:

`/directorio/directorio/public_html/FCKeditor`

Podemos utilizar el siguiente ejemplo en php:

```
<?php
include("../FCKeditor/fckeditor.php");

if(!empty($_POST["FCK_CODE"])){
    $valor=$_POST["FCK_CODE"];
}else{
    $valor='Valor dentro del area de
edicion';
}

$oFCKeditor = new FCKeditor('FCK_CODE');
$oFCKeditor->BasePath = '../FCKeditor/';
```

```

$oFCKeditor->Value = $valor;
$oFCKeditor->Width  = '600' ;
$oFCKeditor->Height = '400' ;

?>
<html>

    <head>

        <title>Titulo de la
pagina</title>
        <style type="text/css">
            body, td, input {font-
family:sans-serif}
        </style>

    </head>

    <body>

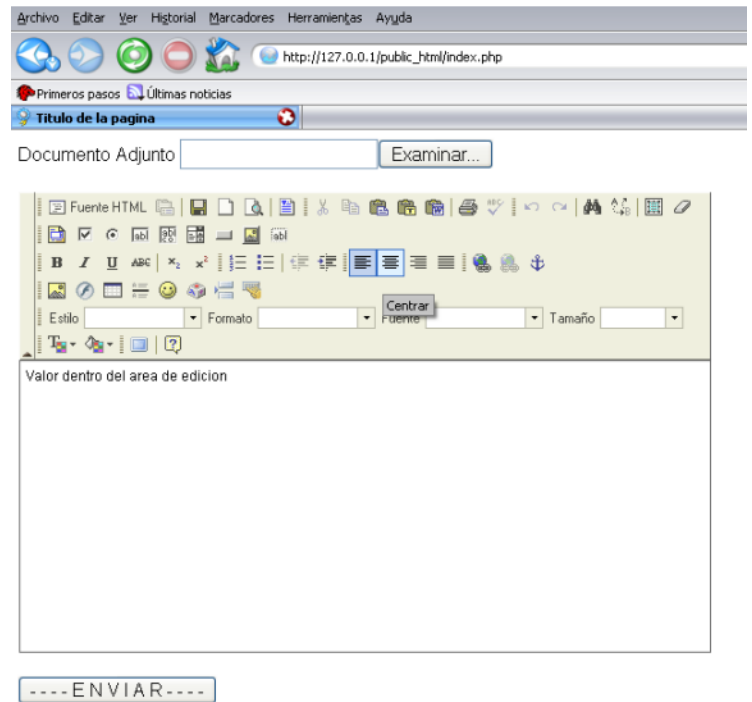
        <form method="post"
enctype="multipart/form-date"
action="<?php echo
$_SERVER["PHP_SELF"]; ?>">
            Documento Adjunto <input
type="file" name="adjunto"><br><br>
            <?php $oFCKeditor->Create();
?><br>
            <input type="submit"
name="boton" value=" - - - - E N V I A
R - - - - ">
            </form>
        </body>
</html>

```

En las primeras líneas de este ejemplo se incluye el software y también se pregunta por una variable.

Después está la parte del FCKeditor en donde se crea el objeto, se establece el path base, se le asigna un valor al área de edición y se establecen los valores de ancho y alto.

Después tenemos un formulario en donde tenemos un campo file, que podría servirnos para adjuntar un archivo, después está el editor y al final un botón. El ejemplo se ve de esta manera.



Es un ejemplo sencillo que sólo se manda la información a sí mismo, si se observa el código HTML de resultado se puede ver que FCKeditor arroja XHTML en un campo de tipo hidden nombrado como FCK_CODE que fue el valor que le dimos al constructor al crear el objeto.

En este ejemplo se pueden hacer unas modificaciones sencillas, se agrega una función para registrar en base de datos y se obtiene algo funcional.

Es muy sencillo aplicar FCKeditor, en el wiki hay muchísima información para explotar al máximo esta excelente herramienta. 🤖

Héctor Leal es parte del Consejo Editorial SL. Profesionalmente se dedica al desarrollo web, usando aplicaciones libres.

desarrollando aplicaciones web

Adib Saavedra Bocanegra
adib.saavedra@gmail.com



En esta ocasión revizaremos el desarrollo de aplicaciones web usando herramientas como Bluefish, Pear (DB, HTML_QF) y MySQL.

Actualmente el uso de clases para el desarrollo de software esta muy extendido, pues estas hacen que el código sea mas limpio, elegante, simple y promueven la reutilización de códigos, por lo tanto el desarrollo se hace mas rápido.

Para el desarrollo web, la tecnología LAMP, es muy popular, y el manejo de clases también, esto hizo posible el nacimiento de PEAR.

PEAR: PHP Extension and Application Repository es "un framework y un sistema de distribución de componentes PHP reusables".

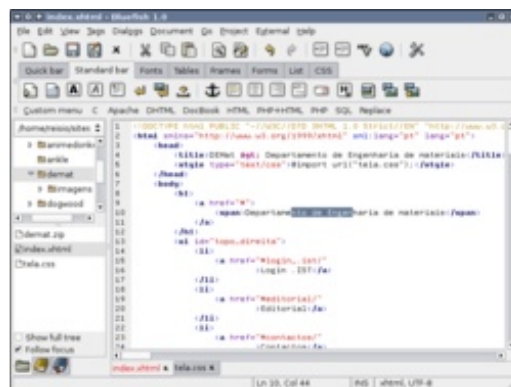
En la práctica es una colección de 265 paquetes de clases desde Autenticación hasta Web Services, sólo comparable en su misión al CPAN de Perl. Aunque todavía le queda mucho para equipararse a éste último.

PEAR DB es una clase de abstracción de acceso a base de datos, es decir un módulo de PHP que permite el control de una base de datos sin necesidad de especificar ningún servidor de bases de datos en concreto.

Esto significa que el mismo código se puede utilizar para acceder a MySQL, Postgres u Oracle.

Pear HTML_QuickForm provee metodos para crear, validar y procesar forms de HTML de una forma rapida, basado en diversos para metros se pueden contruir desde un formulario simple hasta un complejo sistema de registro validando con eventos de Javascript.

Bluefish es un editor, liberado bajo GPL, para programadores y diseñadores web experimentados. Reconoce diversos lenguajes de programación y de marcas, pero se centra en la edición de sitios web dinámicos e interactivos



Como servidor LAMP entendemos un servidor GNU/Linux con Apache Web Server, MySQL y PHP instalados.

Instalando el Framework

En una distribución basada en Debian, (Ubuntu, Knoppix, Kubuntu), la forma de instalar el framework sería la siguiente:

```
#apt-get install php4-pear
```

o de la forma común:

```
$lynx -source http://go-pear.org/ | php
```

y el instalador, expondrá unas preguntas básicas, después agregar el path del repositorio PEAR en su `php.ini`, ejemplo:

```
include_path = ".:usr/share/php"
```

Interactuando con el Framework

Ya instalado el framework podemos interactuar con él desde la línea de comandos, a continuación mostraremos las más comunes y una breve descripción de ellas.

Mostrar las clases que tenemos instaladas

```
#pear list
```

Mostrar las clases que están activas, para pear

```
#pear list-all
```

Instalar una clase

```
#pear install nombre del paquete
```

Desinstalar una clase

```
#pear uninstall nombre del paquete
```

Actualizar una clase

```
#pear upgrade nombre del paquete
```

A continuación pasare a explicar un ejemplo sencillo de lo que podemos hacer con todas estas herramientas.

Para ello se agregaron algunas capturas de pantalla a fin de que ustedes puedan realizar los ejemplos de forma exitosa.

Con estas herramientas podemos desarrollar todo tipo de aplicaciones que nos podamos imaginar, para este caso he desarrollado una agenda que tiene formas para insertar un elemento...

The screenshot shows a web form titled "Insertar nuevo registro" with a "Inicio" link. It contains four required fields: "Nombre" (text input with "Rafael"), "Apellidos" (text input with "Marquez"), "Pais" (dropdown menu with "Mexico"), and "Club" (dropdown menu with "Barcelona"). Below the fields are two buttons: "Limpiar" and "Insertar".

Eliminar un registro...

The screenshot shows a web form titled "Eliminar un registro" with a "Inicio" link. It contains one required field: "Nombre" (text input with "Alvaro"). Below the field are two buttons: "Limpiar" and "Eliminar". A note at the bottom states "* denotes required field".

Modificar un registro existente...

The screenshot shows a web form titled "Modificar un registro" with a "Inicio" link. It contains two required fields: "Nombre" (text input with "Ronaldo") and "Club" (dropdown menu with "Inter de Milan"). Below the fields are two buttons: "Limpiar" and "Modificar". A note at the bottom states "* denotes required field".

Y al final tener la capacidad de imprimir todos nuestros registros en una tabla, claro recordando que esto es solo un pequeño ejemplo de todas las posibilidades que tenemos.

[Inicio](#)

Nombre	Apellidos	País	Club
Carlos	Sabido	Mexico	PSV Eindhoven
Francisco	Fonseca	Mexico	Sporting Lisboa
Rafael	Marquez	Mexico	Barcelona

[Atras](#)

Por fines de este artículo solo he incluido algunas capturas, sin embargo les invito a descargar el [Bonus Pack](#) de RevistaSL 5 para encontrar el código de esta agenda de ejemplo.

Espero este artículo les motive a probar todas las posibilidades que el mundo del software libre nos ofrece para desarrollo web. 🤖

¡Anúnciate en Revista SL!

publicidad@revista-sl.org



¿Te interesa dar a conocer tus productos o servicios de forma rápida, eficaz y a una gran cantidad de gente?....

Revista SL "El software libre hecho revista es la solución"...

Somos una publicación bimestral, cuyo único objetivo es divulgar el software libre, proyectos, trabajos e ideas de la comunidad de software libre....

Llegamos a todos los países de habla hispana del mundo, países en donde tendrá presencia tu empresa...

Nuestro número 3 "Debian, hijos, primos y arrimados" tuvo más de 20,000 descargas, por lo que tendrás miles de clientes potenciales....

Con tu contribución ayudas al crecimiento del proyecto :)

desarrollando web con LAMP .NET

Martin Marquez
xomalli@gmail.com



Durante los últimos años las herramientas de desarrollo para aplicaciones en Internet han evolucionado, desde los comienzos de las primeras páginas que ofrecían un procesamiento a las solicitudes de los usuarios usando un formulario HTML y un programa CGI del lado del servidor, pasando por los applets de Java, los lenguajes server-side scripting y las páginas ASP, se ha buscado una manera más segura, más rápida y más robusta de desarrollar tales aplicaciones.

Se necesita la creación de Frameworks que no solo ofrecieran el complemento a los formularios HTML (como los CGI's), sino que además pudieran ofrecer herramientas, protocolos y lenguajes de programación para facilitar el desarrollo de aplicaciones como si fuesen hechas para el escritorio.

Dentro de estos frameworks se encuentra .NET, la cual podemos utilizar desde Linux usando el proyecto mono (<http://go-mono.com>).

En ese sitio hay una descripción más amplia de la plataforma .NET y mono.

Recomiendo además también leer la revista software libre número 2 <http://www.revista-sl.org/> ahí hay una introducción a .NET, por el momento sólo nos enfocaremos a las herramientas para el desarrollo de aplicaciones web, usando software libre.

Creando la estructura de nuestra aplicación Web

Asumiendo que tenemos instalados los paquetes PostgreSQL, Mono y Apache, descargaremos el `mod_mono` que es el plugin que nos permite el soporte de páginas aspx en Apache, vamos al sitio de descarga <http://www.mono-project.com/ASP.NET> sea que lo instalemos desde los paquetes RPM o bien desde un (.tar.gz), en la página de descarga están las instrucciones a detalle.

Teniendo el software instalado, crearemos nuestra web application. Primero vamos a crear una estructura para nuestro proyecto, esto puede hacerse de manera manual o bien usando una herramienta visual, en mi caso particular uso el Quanta plus <http://quanta.kdewebdev.org/>.

Bajo un directorio (public_html,htmls) haremos una carpeta con el nombre de nuestra web application en este caso se llamará "computo".


```

dragon@system32:/...ublic_html/computo - Terminal N° 2 - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda
dragon@system32:~/public_html> pwd
/home/dragon/public_html
dragon@system32:~/public_html> mkdir computo
dragon@system32:~/public_html> ls -l | grep computo
drwxr-xr-x  2 dragon users  48 2006-10-18 12:42 computo
dragon@system32:~/public_html> cd computo
dragon@system32:~/public_html/computo> pwd
/home/dragon/public_html/computo
dragon@system32:~/public_html/computo> █

```

Crearemos la siguiente estructura, bajo el directorio "computo":

- bin/ Carpeta donde se colocaran los binarios de la aplicación
- images/ Carpeta de imagenes
- Web.config Archivo de configuración de nuestra aplicación
- index.aspx Archivo de la vista inicial
- index.aspx.cs Archivo controlador de nuestra vista inicial

El software de mono viene con un pequeño servidor web que sirve para probar nuestras páginas aspx, usaremos el siguiente código muy sencillo, lo guardamos con el nombre de index.aspx:

```

<%@Page language="C#"%>
<%@import Namespace="System"%>
<script language="C#" runat="server">
void Page_Load() {
    if(Session["c"] == null){
        int c = 0;
        Session.Add("c",c.ToStr
ing());
    }else{
        int c =
Int32.Parse(Session["c"].ToString());
        c++;
        Session["c"] =
c.ToString();
    }
}
</script>

```

```

<html>
<body>
No de veces que ha llamado a la pagina
durante esta sesión
<%=Session["c"].ToString()%><br>
<a href="index.aspx">Refrescar</a><br>
<a href="index.aspx" target="_new">Abrir
en otra ventana</a>
</body>
</html>

```

Usen un editor de texto, en mi caso yo utilizo kdevelop <http://www.kdevelop.org/>

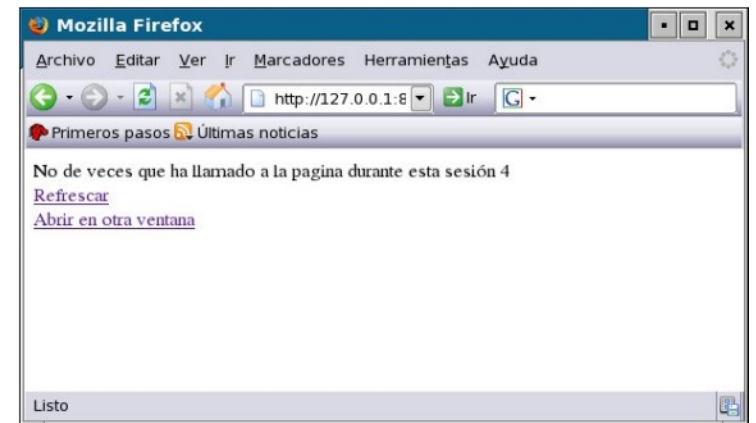
Teniendo el código escrito en un archivo llamado index.aspx, usamos el comando xsp en el directorio donde se encuentra nuestra aplicación.

```

dragon@system32:/...ublic_html/computo - Terminal N° 4 - Konsole
Sesión Editar Vista Marcadores Preferencias Ayuda
dragon@system32:~/public_html/computo> ls
bin  images  index.aspx
dragon@system32:~/public_html/computo> xsp
xsp
Listening on port: 8000 (non-secure)
Listening on address: 0.0.0.0
Root directory: /home/dragon/public_html/computo
Hit Return to stop the server.
█

```

Después lo visualizamos con un navegador:



Scripting language versus Code-behind

Scripting language

Las páginas ASPX se pueden programar de 2 maneras, la primera se llama *Server Side Scripting*

Language, es como nuestra primera página de ejemplo, es la manera más usual de programar para empezar con ASP.NET, es la clásica mezcla de código HTML con código scripting, la sintaxis comienza con:

```
<%@Page language="C#"%>
```

Esta línea nos indica el lenguaje scripting que vamos a utilizar además de C#, esta debe ser siempre nuestra primera línea en todas las páginas ASP.NET sigue la parte de la declaración de bibliotecas que vamos a usar en nuestra página, por ejemplo si utilizamos las bibliotecas System.Data, Npgsql, System.IO, etc.

```
<%@import Namespace="System"%>
<%@import Namespace="Npgsql"%>
<%@import Namespace="System.IO"%>
<%@import Namespace="System.Data"%>
```

Cada línea por cada biblioteca que utilicemos, el siguiente bloque es exclusivamente vamos a poner todo el código C# que vamos a utilizar en nuestra página, dentro de este bloque declararemos funciones.

```
<script language="C#" runat="server">
    string foo(){
    }
</script>
```

no se pueden declarar funciones fuera de este bloque, ahora para imprimir en nuestra página se utiliza el <% como inicio de un bloque scripting y un %> como el final de un bloque, aquí podemos imprimir de la siguiente manera.

```
<%=foo()%> escribe el valor de foo()
<% Response.Write(foo()); %> escribe el valor de foo
```

También podemos intercambiar código HTML con código scripting, como en el siguiente ejemplo:

```
<%@Page language="C#"%>
<html>
<body>
<p>
```

```
<%for(int i = 0;i < 6;i++){%>
<font face="arial">Valor de i =
<%=i></font>
<%Response.Write("</P>"); } //fin del
for%>
</body>
</html>
```

y el resultado será...



Code behind

Una de las principales desventajas de las páginas hechas con scripting language es la dificultad para darle mantenimiento al código, a esa manera de programación se le conoce como código spaghetti, ya que es una mezcla de uno o varios lenguajes. El modo Code behind, es un modelo donde hay una separación del código y la vista que es generada por ese código.

Nuestro primer ejemplo usando un modelo Code behind, sería de la siguiente manera, primero modificamos la vista:

```
<%@ Page language="c#"
Codebehind="index.aspx.cs"
Inherits="computo.classes.index"
"%>
<html>
```

```
<body>
```

No de veces que ha llamado a la pagina durante esta sesión

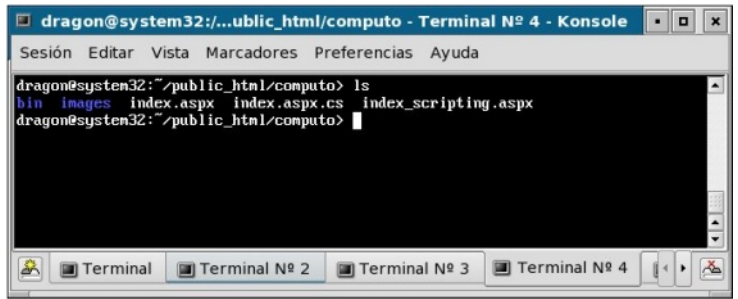
```
<%=Session["c"].ToString()%><br>
<a href="index.aspx">Refrescar</a><br>
<a href="index.aspx"
target="_new">Abrir en otra ventana</a>
</body>
</html>
```

```
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
```

```
namespace computo.classes{
    public class index :
System.Web.UI.Page{
        private void Page_Load(object
sender, System.EventArgs e){
            if(Session["c"] == null){
                int c = 0;
                Session.Add("c",c.ToString()
);
            }else{
                int c =
Int32.Parse(Session["c"].ToString());
                c++;
                Session["c"] = c.ToString();
            }
            override protected void
OnInit(EventArgs e){
                this.Load += new
System.EventHandler(this.Page_Load);
            }
        }
    }
}
```

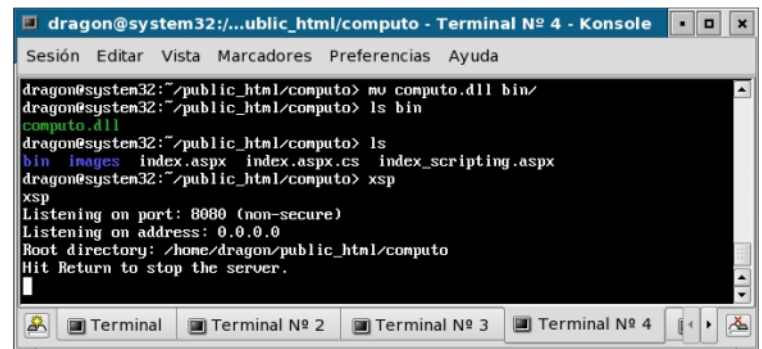
tendremos los archivos de nuestra web aplicación como se muestra en la imagen.

Ahora debemos compilar nuestra aplicación para generar un archivo .dll el cual deberá de estar en nuestro directorio bin para que pueda funcionar nuestra aplicación, lo compilamos usando el siguiente comando:



```
mcs -r:System.Web /t:library
index.aspx.cs -out:computo.dll
index.aspx.cs
```

ahora movemos el archivo computo.dll al directorio bin, en donde van los binarios, ejecutamos el servidor xsp abrimos el navegador y probamos.



Ejemplo 2, usando code-behind ejemplo2.aspx.cs

```
using System;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;

namespace computo.classes{
    public class ejemplo2 :
System.Web.UI.Page{

        protected string s = " ";
        private void Page_Load(object
sender, System.EventArgs e){
            for(int i = 0;i < 6;i++){
                s += "<p><font
face=\"arial\">Valor de i = " +
i.ToString() + "</font></p>";
            }
        }
    }
}
```

```
    }  
    override protected void  
OnInit(EventArgs e) {  
    this.Load += new  
System.EventHandler(this.Page_Load) ;  
    }  
    }  
}
```

Ejemplo 2 vista ejemplo2.aspx

```
<%@Page language="c#"  
Codebehind="ejemplo2.aspx.cs"  
Inherits="computo.classes.ejemplo2"%>  
<html><body>  
<%Response.Write(s);%>  
</body></html>
```

En la siguiente entrega usaremos los controles WebForm y postgreSQL para crear un formulario y guardar nuestros datos.

Referencias en línea:

<http://www.go-mono.com>

<http://dotgnu.org/> 

Este artículo se compone de tres entregas y esta protegido bajo la licencia Creative Commons

transacciones a una una BD con Pear BD

Adib Saavedra Bocanegra
adib.saavedra@gmail.com



[PHP Extension and Application Repository](#) es un framework y un sistema de distribución de componentes PHP reusables".

En la práctica es una colección de 265 paquetes de clases desde Autenticación hasta Web Services, sólo comparable en su misión al CPAN de Perl. Aunque todavía le queda mucho para equiparse a este último, su progreso nunca se ha detenido, desde que hace 5 años, Stig Bakken contribuyera con el primer paquete de PEAR: DB, la clase "oficial" de abstracción de acceso a base de datos de PHP.

PEAR::DB es una clase de abstracción de acceso a base de datos, es decir un módulo de PHP que permite el control de una base de datos sin necesidad de especificar ningún servidor de bases de datos en concreto.

Esto significa que el mismo código se puede utilizar para acceder a MySQL, Postgres u Oracle.

Instalando Pear

En una distribución basada en Debian, (Ubuntu, Knoppix), será:

```
#apt-get install php-pear
```

o de la forma común

```
$lynx -source http://go-pear.org/ | php
```

y el instalador, expondrá unas preguntas básicas, después agregar el path del repositorio PEAR en su php.ini, ejemplo:

```
include_path = ".:usr/share/php"
```

Interactuando con Pear

Comandos básicos del front-end, de PEAR:

```
debian:/# pear list (muestra las clases que  
tenemos instaladas)
```

```
debian:/# pear list-all (muestra las clases  
que están activas, para pear)
```

```
debian:/# pear install nombre del paquete  
(instala una clase)
```

```
debian:/# pear uninstall nombre del  
paquete (desinstala una clase)
```

```
debian:/# pear upgrade nombre del paquete  
(desinstala una clase)
```

ahora instalaremos la clase DB...

Instalando la clase DB

En una distribución basada en Debian, (Ubuntu, Knoppix), sería:

```
#apt-get install php-db
```

o de la forma común:

```
#pear install db
```

Si todo salió bien ya estamos listos para trabajar, con nuestra clase.

MBD soportados por PEAR DB

dbase: dBase
fbsql: FrontBase (funcional desde DB 1.7.0)
ibase: InterBase (funcional desde DB 1.7.0)
ifx: Informix
msql: Mini SQL (funcional desde DB 1.7.0)
mssql: Microsoft SQL Server (NOT for Sybase. Compile PHP --with-mssql)
mysql: MySQL (for MySQL <= 4.0)
mysqli: MySQL (for MySQL >= 4.1) (requires PHP 5) (since DB 1.6.3)
oci8: Oracle 7/8/9
odbc: ODBC (Open Database Connectivity)
pgsql: PostgreSQL
sqlite: SQLite
sybase: Sybase

Creando nuestra BD

Primeramente vamos a crear nuestra BD que usaremos para realizar nuestras "queries".

```
mysql -p "mipass";
create database escuela;
use escuela;
create table alumnos (ncontrol int (4),
                      nombre
                      varchar (10),
                      edad
                      tinyint (2));
show tables;
describe alumnos;
grant select, insert, update, delete
on escuela.* to adib@localhost
identified by "ladib";
```

Insertando registros....

Para insertar registros entraremos con el usuario, que acabamos de crear:

```
use escuela;
insert into alumnos values (1234,"
adib",24);
insert into alumnos values (1235,"
mike",24);
insert into alumnos values (1236,"
laura",23);
insert into alumnos values (1237,"
ana",25);
insert into alumnos values (1238,"
crisrina",24);
```

Conectando y desconectando a la BD

Para realizar estas tareas usaremos el siguiente código:

```
<?php
require_once 'DB.php';
$user = 'adib';
$pass = 'ladib';
$host = 'localhost';
$db_name = 'escuela';
$dsn =
"mysql://$user:$pass@$host/$db_name";

$db = DB::connect($dsn);

if (DB::isError($db)) {
die ($db->getMessage());
}

echo "<center><h3>Aquí estoy conectado
con la BD Escuela</h3></center>";

$db->disconnect();
?>
```

Ejecutando una “query”, usando el metodo DB::getAll()

Para realizar esta tarea usaremos el siguiente código:

```
.....
if (DB::isError($db)) {
die ($db->getMessage());
}

$results = $db->getAll('SELECT
nombre,edad
        FROM alumnos');

    foreach($results as $row) {
        print "Nombre:
$row[0]&nbsp;&nbsp;  Edad: $row[1]<br>";
    }

$db->disconnect();
?>
```

Ejecutando una “query”, usando el metodo DB::getCol()

Para ejecutar la "query" ahora en este método usaremos el siguiente código:

```
.....
if (DB::isError($db)) {
die ($db->getMessage());
}

$nombres = $db->getCol('SELECT nombre
FROM alumnos');
    print "Nuestros alumnos: <ul> <li>
" . join('<li>', $nombres)
    . "</ul>";

$db->disconnect();
?>
```

Ahora veremos un último método para ejecutar una "query" con el método DB::getAssoc()

Ejecutando una “query”, usando el metodo DB::getAssoc()

Para realizar esta tarea usaremos el siguiente código:

```
.....
if (DB::isError($db)) {
die ($db->getMessage());
}

    $nombre= $db->getAssoc('SELECT
ncontrol,nombre FROM
        alumnos');
    print '<select name="nombre">';
    foreach ($nombre as $ncontrol =>
$nnombre) {
        print "<option
value=\"\$ncontrol\">$nombre</option>";
    }
    print '</select>';

$db->disconnect();
?>
```

Obteniendo información de una “query”, usando el método DB_Result::numRows()

Para realizar esta tarea usaremos el siguiente código:

```
.....
if (DB::isError($db)) {
die ($db->getMessage());
}

$result = $db->query('SELECT edad FROM
alumnos
        WHERE edad < 24');
if ($result->numRows() > 0) {
    print "Edad: <ul>";
    while($row = $result->fetchRow())
        { print "<li> $row[0]</li>"; }
    print "</ul>";
}
```

```

} else {
    print "no existen alumnos menores
de 24 años.";
}
$db->disconnect();
?>

```

Obteniendo informacion de una “query”, usando el metodo DB::affectedRows()

Ahora usaremos otro método para obtener información de la "query"

```

.....
if (DB::isError($db)) {
die ($db->getMessage());
}

$db->query("UPDATE alumnos SET edad =
23 WHERE
    nombre LIKE 'adib%");
print 'se ha relizado ' . $db-
>affectedRows()
    . ' movimiento, al campo edad-
adib.';

$db->disconnect();
?>

```

Ejecutando multiples “query's”, usando el metodo DB::prepare() and DB::execute()

Bien, ahora veremos una de las tareas mas útiles, el ejecutar multiples "query's"

```

.....
if (DB::isError($db)) {
die ($db->getMessage());
}

$result = $db->prepare('INSERT INTO
alumnos VALUES (?, ?, ?)');

```

```

$db-
>execute($result,array(1239,'oscar',25));
$db-
>execute($result,array(1230,'paco',25));

$db->disconnect();
?>

```

Y con esto finalizaríamos, les invito a realizar los ejemplos que aunque muy sencillos les serán sumamente utiles. 🤖

Adib es egresado de la carrera de Licenciatura en Informatica del Tecnológico de Coatzacoalcos, México. Y a participado en diversos foros y congresos en México con un alto nivel técnico.

introducción a las graficas dinamicas con jpgraph y php



M en C Longino Jacome
longino@php.com.mx

La biblioteca JpGraph es un conjunto de funciones OO (orientadas a objetos) que permiten la generación sencilla de diferentes tipos de elementos gráficos. Mediante unas pocas instrucciones podemos hacer gráficos con mínimo esfuerzo y de calidad.

JpGraph puede se puede obtener del sitio <http://www.aditus.nu/jpgraph/> La licencia de JpGraph es QT, es decir, existe una licencia de libre distribución para fines no comerciales y existe una licencia comercial.

La instalación es muy sencilla, teniendo un sistema PHP con soporte para GD. Utiliza `phpinfo()` para determinar si tu instalación tiene el soporte.

Del sitio <http://www.aditus.nu/jpgraph/> bajar la biblioteca para PHP4 o PHP5, según el caso. El directorio contiene las funciones necesarias para hacer una gran cantidad de gráficos, la documentación necesaria y una buena colección de ejemplos.

Lo único que se necesita es incluir los archivos según el tipo de gráfica a realizar.

Por ejemplo, si vamos a hacer una gráfica de líneas, podemos empezar así:

```
<?
include("graficas/jpgraph.php" );
include("graficas/jpgraph_line.php" );
?>
```

En el directorio “graficas” tenemos el código de la biblioteca de JpGraph. `jpgraph.php` es el archivo básico para empezar, y después se agrega el tipo específico de gráfico a realizar, en este caso, gráficos de líneas. Si fuésemos a realizar una gráfica de pie, utilizaríamos el archivo `jpgraph_pie.php`

```
<?
include("graficas/jpgraph.php" );
include("graficas/jpgraph_line.php" );

// Algunos datos para graficar
$ydata = array(11,3, 8,12,5 ,1,9, 13,5,7
);

// Creamos la gráfica. Los valores se
refieren a las dimensiones
$graph = new Graph(500, 340,"auto");
$graph->SetScale( "textlin"); // El tipo
de escala

// Creamos la gráfica para los datos
$lineplot =new LinePlot($ydata);

// Adicionamos el gráfico
$graph->Add( $lineplot);
```

```
// Desplegamos la gráfica
$graph->Stroke();
```

```
?>
```

En este caso estamos dando los valores de forma manual, pero el arreglo de datos se puede obtener de una llamada a una base de datos, por ejemplo.

La gráfica, como está no dice gran cosa, ya que le faltan elementos.

```
$graph->title->Set ('Ejemplo'); //
Titulo
$graph->xaxis-> title->Set("Eje X" );
// Título para el eje X
$graph->yaxis-> title->Set("Eje Y" );
// Título para el eje Y
```

Con esos elementos, ya tenemos más idea de lo que la gráfica representa. De igual forma, se pueden añadir más datos.

```
$ydata2 = array(1,19, 15,7,22 ,14,5,
9,21,13 ); // Segundo arreglo de datos
$lineplot2 =new LinePlot($ydata2); //
Se generan las líneas
$graph->Add( $lineplot2); //
```

Añadir una segunda serie de datos puede ser útil para realizar gráficas comparativas, pero tendremos que trabajar un poco en la “estética”.

```
$lineplot2 ->SetColor("orange");
$lineplot ->SetColor("blue");
$lineplot->SetLegend ("Linea 1");
$lineplot2 ->SetLegend("Linea 2");
```

La caja que contiene los nombres de las series, se puede ubicar en un mejor lugar

```
$graph->legend->SetLayout (LEGEND_HOR);
$graph->legend-
>Pos (0.5,0.99,"center", "bottom");
```

Y además, hemos puesto las líneas más gruesas

```
$lineplot ->SetWeight(2);
$lineplot2 ->SetWeight(2);
```

Vamos a quitar el marco y la sombra gris del fondo, a redimensionar la gráfica para que no se encime la caja que contiene los nombres de las series.

```
<?
include("graficas/jpgraph.php" );
include("graficas/jpgraph_line.php" );

$ydata = array(11,3, 8,12,5 ,1,9, 13,5,7
); // Datos de la serie 1
$ydata2 = array(1,19, 15,7,22 ,14,5,
9,21,13 ); // Datos de la serie 2

// Creamos la gráfica de 500 X 340 pixeles
$graph = new Graph(500, 340,"auto");

$graph->SetMarginColor('white'); //
Quitamos el fondo gris
$graph->SetFrame(false); // Quitamos el
marco de la gráfica

$graph->SetScale( "textlin"); //
Definimos la escala

$graph->title->Set ('Ejemplo'); // Título
de la gráfica
$graph->xaxis-> title->Set("Eje X" ); //
Título del eje X
$graph->yaxis-> title->Set("Eje Y" ); //
Título de eje Y

// Añadimos u grid alternado de gris y
azul para resaltar la escala del eje Y
$graph->ygrid-
>SetFill(true, '#EFEFEF@0.5', '#BBCCFF@0.5')
;
$graph->xgrid->Show();


// Creamos la gráfica a partir de los
datos
$lineplot =new LinePlot($ydata); //
Primer conjunto de datos
$lineplot ->SetColor("blue"); // Color de
la línea del primer conjunto de datos
$lineplot2 =new LinePlot($ydata2); //
Segunda línea de datos
$lineplot2 ->SetColor("orange"); // Color
de la segunda línea
```

```
// Armamos la gráfica para ambos
conjuntos de datos
$graph->Add( $lineplot);
$graph->Add( $lineplot2;

// Títulos para las series
$lineplot->SetLegend ("Linea 1");
$lineplot2 ->SetLegend("Linea 2");

// Posición de la caja que contiene los
nombres de las series
$graph->legend->SetLayout(LEGEND_HOR);
$graph->legend-
>Pos(0.5,0.99,"center","bottom");

// Desplegamos la gráfica terminada
$graph->Stroke();

?> 
```

CONOCE PIENSA SORPRENDETE TU MEJOR ELECCION

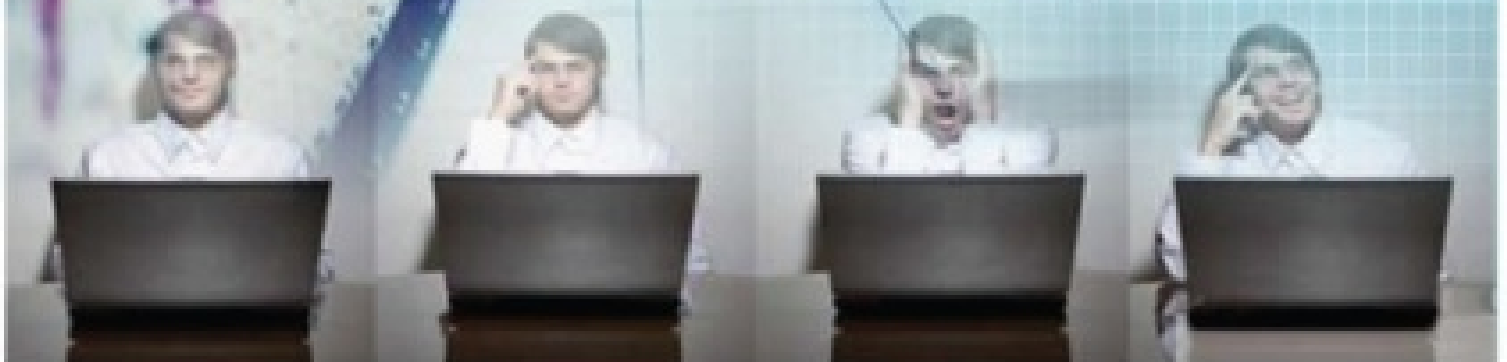
INFO-@

innovaciones informaticas del sureste

www.info-e.com.mx

- Servidores dedicados GNU/Linux
- Resellers
- Instalacion de servidores GNU/Linux
- Hospedaje web

LLEVANDOLA TECNOLOGIA A OTRO NIVEL



XML a la conquista del mundo

Sonia Sánchez
sonny.taz@gmail.com



“A casi 10 años de su creación, todos aquellas personas que pensamos en postergar el aprendizaje de XML en ese entonces, hoy, se ha vuelto el tema que se debe aprender. El simple hecho de tener sindicados los sitios de noticias o blogs que leemos a diario en un lector de feeds, nos involucra en el uso y aprendizaje de una tecnología que lejos de estar en decadencia, se ha convertido en motor de la evolución del mundo de las IT's”

Si bien es cierto que el mundo de las Tecnologías de la Información es cambiante y día a día se actualiza, el año de 1998 como tormenta, nos sorprende con el nacimiento del Extensible Markup Language o mejor conocido como XML.

XML está diseñado para incorporar las siguientes características al mundo Web.

o Fácil de usar. Los documentos XML son fáciles de leer y entender aún para aquellos usuarios con pocos conocimientos de cómputo o nulos. Esta simplicidad es quizás, el más grande punto de venta de XML.

o Una estructura formal. Aunque nos ofrece flexibilidad para el nombramiento y uso de las etiquetas, éstas deben ser nombradas y jerarquizadas correctamente; las etiquetas deben ser abiertas y cerradas correctamente y los namespaces deben ser definidos en dónde sean requeridos.

¿XML o HTML?

HTML es un lenguaje básico de marcado, compuesto por un número de etiquetas (tags) y uno que otro pequeño conjunto de reglas, por ejemplo, algunas etiquetas deben ser cerradas al declararse, otras, simplemente declararse. Es simplemente un lenguaje de marcado.

XML en contraste al HTML está formado por reglas consistentes, cuándo se crea un documento XML no solamente se está preparando para ser mostrado en el Web, se está codificando una serie de etiquetas que pueden ser interpretadas en cualquier ambiente XML.

¿Cómo sé si un documento XML es correcto?

Los documentos de tipo XML son considerados correctos cuándo respetan la sintaxis de XML

sintaxis que establece, entre otras reglas, que una etiqueta de inicio debe tener una etiqueta de cierre, si en una de ellas falta la etiqueta de cierre no se puede decir que el documento está bien formado porque no respeta la sintaxis.

Además, debe ser válido porque cumple con las reglas que establece un esquema XML.

Las extensiones XML.

Hablar de XML es hablar también de sus extensiones, es decir, de aquellos elementos que nos permiten extender (como su nombre lo dice) las capacidades básicas del lenguaje y por lo tanto generar una nueva funcionalidad.

Las siguientes, son extensiones XML:

► **XPath**: Las referencias a partes específicas de un documento para ser accedidas por otras herramientas y tecnologías son permitidas a través de esta extensión. Las expresiones de XPath pueden referirse a todo o una parte del texto, datos y valores de los elementos, atributos instrucciones de proceso, comentarios, etc., de un documento XML.

► **XQuery**: ¿Conoces las Bases de Datos Relacionales? Bueno, entonces basta decir que XQuery es a XML lo que SQL a una Base de Datos Relacional.

► **XML namespaces**: Los namespaces permiten a un mismo documento XML contener elementos y atributos con un mismo nombre los cuales se refieren a conceptos diferentes.

► **XML Signature**: define la sintaxis y las reglas de procesamiento para crear firmas digitales en el documento.

► **XML Encryption**: Al igual que el anterior, define las reglas de procesamiento y la sintaxis para crear las reglas de cifrado de un documento.

► **XPointer**: es un estándar del W3C, cuya finalidad es proporcionar una forma de identificación única hacia fragmentos de un documento tipo XML para la formación de vínculos.

Se divide en cuatro especificaciones:

► Como framework: forma las bases para la identificación de los fragmentos XML.

► Como esquema de direccionamiento de un elemento.

► Como esquema para los namespaces.

► Y como esquema para el direccionamiento de Xpath.

La expresión: `#xpointer(id("valor id"))` hace referencia al documento, específicamente a la sección que cumpla con el valor del id marcado.

Por lo tanto la referencia en xlink quedaría marcada de la siguiente manera:

```
xlink:href="http://mi_sitio.com/documento.xml#xpointer(id('valor id'))"
```

La manera corta del uso de la misma referencia sería:

```
xlink:href="http://mi_sitio.com/documento.xml#valorid"
```

XML generador de invenciones.

Los siguientes ejemplos muestran la aceptación de XML en el Web y sus razones para ser considerado como un tópico más de aprendizaje.

XSLT

Extensible Stylesheet Language Transformations puede extraer y ordenar datos XML y darles un formato como HTML o XHTML listo para ser visualizado.

El procesador XSLT toma dos archivos como entrada, uno del tipo documento XML y otro una hoja de estilo del tipo XSLT, produciendo un archivo de salida.

Por ejemplo:

El archivo XML:

```
<?xml version="1.0"?>
<personas>
  <persona nombreusuario="CVE1234">
    <nombre>Luisa</nombre>
```

```

    <apellido>Ruiz</apellido>
  </persona>
  <persona nombreusuario="CVE5678">
    <nombre>Marco</nombre>
    <apellido>Flores</apellido>
  </persona>
</personas>

```

El archivo XSLT:

```

<?xml version="1.0"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Tr
ansform" version="1.0">
<xsl:output method="xml" indent="yes"/>
<xsl:template match="/">
  <transform>
    <xsl:apply-templates/>
  </transform>
</xsl:template>
<xsl:template match="persona">
  <registro>
    <nombreusuario>
      <xsl:value-of
select="@nombreusuario" />
    </nombreusuario>
    <nombre>
      <xsl:value-of
select="nombre" />
    </nombre>
  </registro>
</xsl:template>
</xsl:stylesheet>

```

Tendríamos como resultado:

```

<?xml version="1.0" encoding="UTF-8"?>
<transform>
  <registro>
    <nombreusuario>CVE1234</nombreusua
rio>
    <nombre>Luisa</nombre>
  </record>
  <registro>
    <nombreusuario>CVE5678</nombreusua
rio>
    <nombre>Marco</nombre>
  </record>
</transform>

```

RDF

Resource Description Framework provee una estructura para aplicaciones que intercambian metadatos en el Web. Basado bajo el concepto de crear sentencias sobre los recursos en expresiones de la forma sujeto-predicado-objeto.

El sujeto se basa en el recurso, el predicado en los aspectos del recurso expresando la relación entre el sujeto y el objeto. A esto se le conoce como triples, dentro de la terminología de RDF.

Este tipo de mecanismo para la descripción de recursos es el componente más grande el cuál ha propuesto la actividad web de la W3C, no siendo más que un Software autómatas que puede almacenar, intercambiar y leer información distribuida a través de la Web.

RSS

Rich Site Summary, Really Simple Syndication o RDF Site Summary, son formatos especificados bajo la tecnología XML, los RSS's liberan dicha información como un archivo de tipo XML llamado "RSS feed", "RSS stream" o "RSS channel". Las RSS's permiten a los usuarios suscribirse a sus sitios web favoritos que, los webmasters al colocar en su contenido este formato, los usuarios podrán visualizar desde un lector de feeds.

XML-RPC

Protocolo basado en XML para codificar sus llamadas, además de utilizar HTTP como mecanismo de transporte. Un mensaje XML-RPC es una petición del tipo HTTP-POST, el cuerpo de la petición un documento XML, la ejecución en el servidor de la petición genera y regresa un valor también en formato XML.

Conclusiones.

En los inicios del uso de XML muchos agregamos este tópico en la lista de to-do's para su aprendizaje, alguno más quizás lo pasamos por desapercibido, hoy, es un hecho que tanto la evolución de XML como el uso que muchas tecnologías la han adaptado a su aprendizaje se ha vuelto un must para, al menos, todos los involucrados en los ambientes Web. Ha casi 10 años de su inicio, XML se ha mantenido evolucionando en el mundo de las Tecnologías de la Información. 🤖

web services

Ivan González Zamora
djnavi@gmail.com



En los últimos años, la forma de desarrollo de software ha sufrido cambios haciendo que Internet sea uno de los recursos principales, los sistemas conectados deben sacar provecho a Internet.

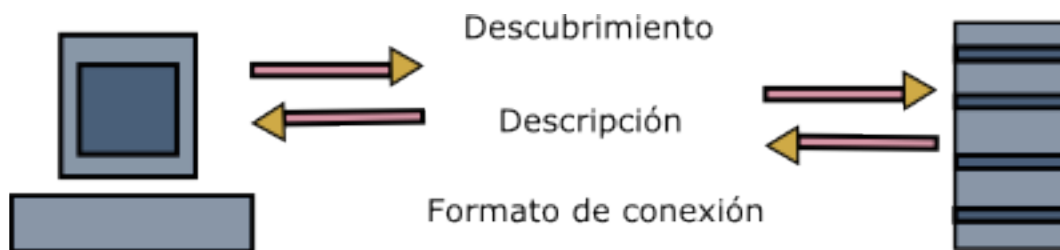
En ocasiones es necesario seguir usando aplicaciones antiguas y combinarlas con nuevos sistemas, esto hace que nos preguntemos ¿Cómo haremos para mezclar aplicaciones que han sido desarrolladas en diferentes sistemas operativos, lenguajes de programación, paradigmas, etc.. y que estas se comuniquen satisfactoriamente? La respuesta se encuentra en un formato en el cual han trabajado diferentes organizaciones, empresas, personas para lograr interoperabilidad en las aplicaciones, este formato es por supuesto XML (eXtensible Markup Language).

En este artículo hablaremos de Web Services que son y para que sirven, trataré de que sea teórico y práctico ya que existe variedad en los lectores de esta revista y no cuento con que todos sean programadores :) así que comencemos.

Un Web Service es una parte de software que proporciona una funcionalidad determinada por programación. Siendo no tan técnicos un Web Service es un fragmento de código que responde devolviendo datos de acuerdo a una petición. Es como pedir información en un módulo turístico, una pregunta, una respuesta :)

Veamos como funciona esto, la imagen muestra la infraestructura de un Web Service

Hablemos de cada uno de los pasos mostrados en la imagen, primero el descubrimiento, este paso no es obligatorio ya que por varias razones no es necesario publicar el Web Service, por ejemplo para uso privado. Descripción, esta fase es importantísima ya que debemos tener una 'descripción del servicio' es decir debemos saber como comunicarnos para obtener una respuesta satisfactoria, como en un restaurant nos dan una carta para saber qué platillo pedir, aquí de manera similar debemos saber qué pedir y cómo, para eso esta WSDL (Lenguaje de descripción de servicios Web) que determina el formato para el Web Service.



Decíamos que una de las razones por la que usar un formato como XML es para interoperabilidad de sistemas, lenguajes, etc... por lo tanto usar una comunicación no estandar o no abierta haría que estos Web Services no estén disponibles universalmente, por esa razón usaremos protocolos abiertos como lo es HTTP conocidos para todos y SOAP (<http://www.w3.org/TR/soap>) soap es un protocolo simple y ligero basado en XML para intercambiar datos. Soap se puede usar en combinación de otros protocolos como lo es HTTP, FTP, SMTP.

Crearemos un Web Service y un Cliente que lo consuma, usaremos Mono para este ejemplo, aunque esto lo escribiré en C# no está limitado solo a este lenguaje, como decíamos podemos usarlo en un sin fin de SO, Lenguajes, Plataformas, con decirles que hasta en VB6 se puede :P

Creemos un archivo con el nombre de MiWebService.cs (Que clásico :P) y escribiremos el siguiente código:

```

/* Nombres de espacios utilizados */
using System;
using System.Web;
using System.Web.Services;
using System;
using System.Collections.Generic;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            WSHolaMundo.Service
CsHolaMundo = new WSHolaMundo.Service();
            string cadena =
CsHolaMundo.HolaMundo();
            Console.WriteLine(cadena);
            Console.Read();
        }
    }
}
using System.Web.Services.Protocols;

```

```

/* Esta clase sera un Webservice */
[WebService(Namespace =
"http://tempuri.org/")]
public class ServicioWeb : Webservice
{
    public ServicioWeb ()
    {
    }

    /* Un metodo Web que devuelve la
cadena "Hola Mundo" */
    [WebMethod]
    public string HolaMundo()
    {
        return "Hola Mundo";
    }
}

```

Ahora el cliente que lo consume
CWebService.cs:

```

using System;
using System.Collections.Generic;
using System.Text;

namespace ConsumidorWeb
{
    class Program
    {
        static void Main(string[] args)
        {
            WSHolaMundo.ServicioWeb
CsHolaMundo = new
WSHolaMundo.ServicioWeb(); /* Creamos una
instancia de la clase */

            string cadena = CsHolaMundo.HolaMundo();
            /* Ejecutamos el metodo que nos devuelve
la cadena "Hola Mundo" */
            Console.WriteLine(cadena); /*
Y lo mostramos en pantalla */
            Console.Read();
        }
    }
}

```

Esto sólo fue un pequeño y muy breve resumen de lo que es un Webservice, ahora les queda a ustedes aprender mas de eéto y ver las grandes ventajas que trae usar esta tecnología.

Yo me despido agradecinédole a Revista-SL por darme este espacio para compartir un poco de información. 🙏



Interlegit.com.mx
DESARROLLO Y SOLUCIONES WEB

100 Mb de Cuota en Disco
Dominio .COM, .US, .NET, .ORG,
.INFO, .NAME, .BIZ, .WS, .TV o .CC
Cualquiera de estos diseños
(solicite el catalogo de plantillas)
Un año de hospedaje 100Mb
100 cuentas de correo
Webmail y POP3
Alias y autocontestadores
PHP, Perl, CGI, MySQL
Administrador de Archivos
Panel de Control
Pago único Anual
De **\$450.00 M.H**

50 Mb de Cuota en Disco
Dominio .COM, .US, .NET, .ORG,
.INFO, .NAME, .BIZ, .WS, .TV o .CC
Cualquiera de estos diseños
(solicite el catalogo de plantillas)
Un año de hospedaje 50Mb
80 cuentas de correo
Webmail y POP3
Alias y autocontestadores
PHP, Perl, CGI, MySQL
Administrador de Archivos
Panel de Control
Pago único Anual
De **\$ 350.00 M.H**

Promociones de Noviembre para los lectores de la Revista SL

**15% de descuento en otros
planes de hospedaje**

**50% de descuento en dominios
MX en la compra de otros
planes de hospedaje**

desarrollo web

Sara Mostajo Rojas
linuxchix,peru@gmail.com



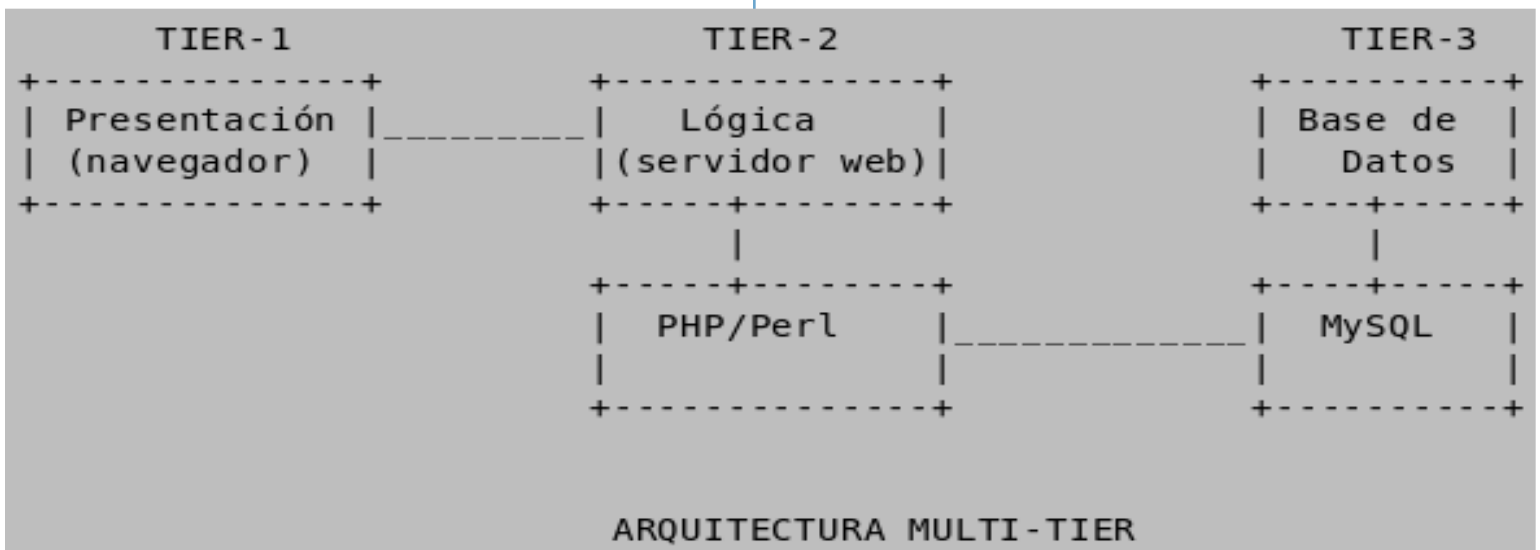
Uno de las preguntas más importantes que tenemos que hacernos a la hora de desarrollar aplicaciones o sitios web con base de datos y cualquier lenguaje de scripting, es: ¿existe una herramienta que permita integrar código y consultas en la base de datos? La respuesta es SI

Cuando se desarrollan aplicaciones web separamos muy claramente las distintas partes del sistema para un mejor desempeño (arquitectura multi-tier):

- ▶ Presentación,
- ▶ Lógica de aplicación,
- ▶ Almacenamiento de datos.

Esta separación es necesaria y nos dará muchas ventajas:

- ▶ Primera y más importante, evitamos hacer dependiente todo un sitio web (scripts, diseño gráfico) de una sola tecnología o aplicación.
- ▶ Podemos utilizar cualquier software para el desarrollo del diseño de la página web, por ejemplo un sistema gestor de contenidos (CMS).
- ▶ No se puede exigir a los programadores que aprendan a usar un determinado software de diseño, que normalmente están orientados a diseñadores y con una curva de aprendizaje bastante elevada.



► Si permitimos que el trabajo de los programadores sea lo más independiente del diseño posible, y dejamos que usen sus propias herramientas, aumentaremos considerablemente la productividad.

Las herramientas de desarrollo están evolucionando muy rápidamente, lo que es bueno hoy, podría ser muy malo mañana.

El desarrollo de un sitio web tiene por lo menos cuatro fases distintas:

1. Diseño conceptual.
2. Diseño gráfico y árbol de navegación.
3. Desarrollo.
4. Producción.

El problema radica en que los grupos de desarrollo, sobre todos aquellos que trabajan con presupuestos limitados, no suelen tener la fase de desarrollo claramente divididas, ni una especificación de las herramientas de software que se usarán en cada una de dichas fases. Aquí radica una de las barreras para el desarrollo de sitios Web con Linux: se justifica la decisión de usar Windows, ASP y SQL Server en el hecho de que existen “mejores herramientas de integración”, pues esto, fuera de ser una gran equivocación, nos hace dependientes de tecnologías propietarias (y dominantes), y hacemos que todo el negocio dependa en realidad de 2 o 3 proveedores de software, ya que siempre tendremos que pagar para poder acceder a las últimas herramientas desarrolladas por esas empresas o por copias de ellas; que obviamente no permitirá que el trabajo sea eficiente.

Los scripts en PHP, o en cualquier otro lenguaje, son parte de la lógica. No tienen que ser conocidos y modificados por la misma persona que hace el diseño (es el encargado de sólo una parte de la presentación), es un trabajo totalmente distinto y no puede producir los mismos resultados.

Tampoco se puede admitir que una herramienta que es apta para diseñadores con pocos conocimientos de programación sea apta para un programador experto. Debemos tener definido claramente lo que se desea desarrollar, solo así identificaremos la herramienta adecuada para cada trabajo.

El desarrollo, profesional, de un sitio web se divide por lo menos en cuatro fases

Metodología a seguir

Se propone aquí 4 pasos básicos a seguir para el desarrollo de un sitio Web de forma independiente a las herramientas que usen los diseñadores gráficos. También pongo énfasis en la optimización de las estructuras de las tablas para dar mayor sensación de agilidad al usuario y como estimar y reducir los tiempos de descarga de la páginas.

Estos pasos fueron usados de manera estricta pero no formal en el desarrollo de varios sitios Web complejos.

1. Diseño conceptual

Se fijan las pautas con el cliente, como será el sitio, que datos tendrá, como se navegará, tipo de cliente al que irá orientado, etc. De esta fase se tiene una idea del diseño y la navegación que tendrá el sitio. El resultado ideal de esta fase es una prueba de concepto. La prueba de concepto puede tener dos partes:

1. Diseño gráfico del sitio: El diseño gráfico de una página web es tan solo una parte del diseño de la misma, ya que, además, hay que considerar un conjunto más o menos extenso de condicionantes que van a limitar la libre creatividad del diseñador a quien le puede bastar con los dibujos hechos en aplicaciones de diseño como el Illustrator, FreeHand, Killustrator o con software de imágenes como el Gimp

2. Consulta y carga de datos: Esta parte es preparada principalmente por los programadores y consistirá en unas páginas HTML sencillas generadas por los scripts y que accedan a un prototipo simplificado de la base de datos.

La prueba de concepto no sólo es una buena herramienta para negociar y convencer al cliente, sino también para detectar que información es la que hace falta para completar el diseño de la base de datos y aplicaciones, como así también detectar de antemano los problemas que nos encontraremos en el desarrollo.

2. Diseño gráfico y árboles de navegación

Se hace un diseño detallado de las diferentes páginas que tendrá el Web. Si es un sitio dinámico se necesita tener las pautas y restricciones técnicas de consultas a la base de datos pues los elementos gráficos, ya tengan formato de mapa de bits o vectorial, suelen traducirse en ficheros de bastante peso, dependiendo del tamaño de la imagen y del formato en que se guarde. Esto origina que las páginas que contienen en su diseño muchas imágenes, o pocas pero de gran tamaño, tarden mucho tiempo en ser descargadas desde el servidor web y presentadas en la ventana del ordenador del usuario, que no suele ser muy paciente.

Por ejemplo, si una consulta a la base de datos pueden generar más de 30 o 40 resultados, debe preverse la paginación de la salida, lo que incluye el diseño y enlaces para cada página. También es importante que la longitud y formato de los datos de la base de datos sean adecuados para el diseño que se presenta. Si se generan textos largos, no se pueden poner los mismos en una columna muy angosta y de poca altura.

Un factor clave que hay que tener en cuenta a la hora del diseño conceptual de la parte gráfica y lógica del sitio es *el tiempo que tardará el cliente para visualizar el resultado (tiempo de visualización)*. Hay muchos parámetros que influyen, por ejemplo:

► **Tiempo de respuesta:** cuanto tiempo tarda en servidor en empezar a devolver resultados al navegador. Este es el parámetro más importante que debe tener en cuenta el o los programadores. De él depende que se pueda ajustar el diseño para que los usuarios puedan empezar a visualizar la página lo más rápido posible. Si las primeras consultas a la base de datos son muy complejas o tenemos habilitado el buffering en el lado servidor (en PHP4 se hace con `ob_start()`), podemos afectar negativamente el resultado. Por el contrario, si el resultado se obtiene muy rápidamente el uso del buffering puede ayudar bastante al envío eficiente de los datos a través de la red. Para que la página de una sensación de agilidad, es importante que el tiempo de respuesta no supere los 2 o 3

segundos pues estudios realizados demuestran que el tiempo máximo de aguante de una persona que espera la descarga de una página suele ser de unos 10 segundos, pasados los cuales prefiere abandonar nuestro sitio web y buscar otro más rápido.

► **Tiempo de retorno:** cuanto tiempo tarda el servidor en terminar de ejecutar los programas en el servidor y entregar todos los datos y será siempre superior al tiempo de respuesta ($T_{retorno} > T_{respuesta}$). El tiempo que tarde el programa en terminar de generar todos los datos no sólo influirá en la conexión con un usuario en particular, sino con el rendimiento de todo el sistema. A mayor tiempo de retorno, menor cantidad de conexiones simultáneas posibles y mayor carga de todo el sistema. Si el tiempo de retorno de un script es superior a un 1 segundo, hay que estudiarlo detenidamente. El primer estudio a hacer es el consumo de CPU. Si ésta es baja, tenemos problemas de latencia, posiblemente con la conexión a la base de datos. Si por el contrario el consumo es elevado, la lógica del programa es muy compleja o usamos muchas llamadas de sistema. En estos casos puede ayudar el uso de sistemas de cache de código.

► **Tiempo de descarga:** es el tiempo que tarda el cliente en bajarse todos los datos a su ordenador. Este tiempo es siempre mayor al tiempo de respuesta ($T_{descarga} > T_{retorno}$) y depende de la velocidad de conexión.

► **Tiempo de visualización:** el primer aspecto a tener en cuenta es la cantidad de código HTML que debe ser bajado para cada página. Esto depende del tipo de conexión que tendrán los visitantes. Un buen parámetro es que el tamaño máximo del HTML no debe superar los 30-40 KBytes por página genérica. En el caso de imágenes es un poco más flexible, ya que se puede acelerar la previsualización de la página indicando los tamaños en pixels de cada imagen.

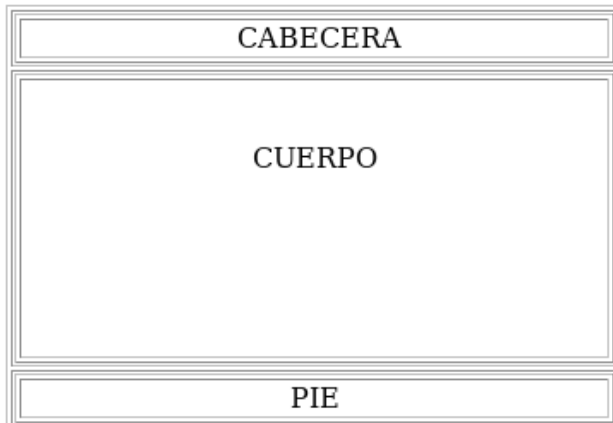
Las páginas de sitios dinámicos suelen tener una estructura bien definida:

1. **Cabecera:** aquí se suelen poner el logo de la empresa, menú de opciones de navegación, banners de publicidad.

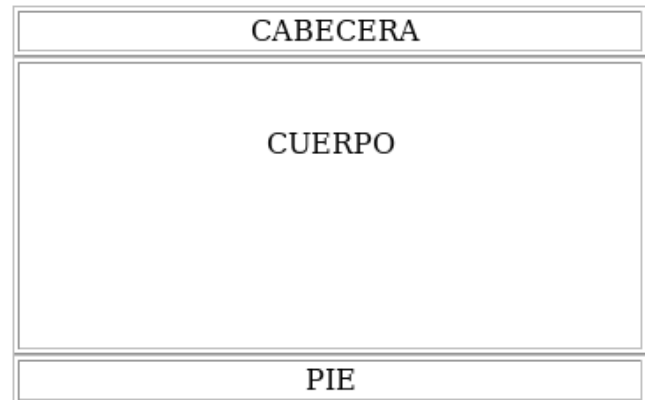
2. **Cuerpo:** en esta parte se pone la parte principal de la página.

3. **Pié:** aquí suele ir información para contactar a la empresa, direcciones, correos, teléfonos y otros enlaces corporativos.

Un error común es poner esas tres partes anidadas es una sola tabla:



La forma correcta de solucionar el problema es de intentar anidar la menor cantidad de tablas, y evitar tener una sola tabla principal que engloba a toda la página. Para ello lo mejor es hacer una **división horizontal de las tablas**:



Las paginas web suelen estar divididas en dos a tres columnas que sirven para un menú de navegación y otra para un índice de temas o artículos relacionados

Este tipo de formato hace que el navegador no pueda generar la página hasta que se reciba el último byte (cercano al tiempo de descarga) que define a la tabla (normalmente el cierre de tabla: </table>).

Supongamos que la página tiene 40KB de tamaño (en HTML), que el tiempo de respuesta del servidor es de 0.5 segundos y que su tiempo de retorno es lo suficientemente bajo para alimentar de datos continuamente. Si la conexión del cliente es un RDSI (64 kbps \approx 7.5 KB/seg) y en condiciones ideales, el tiempo que tardará el navegador en empezar a generar la página en pantalla será igual al tiempo de descarga:

$$T_{vis} = T_{descarga} = 0.5 \text{ seg} + 40 \text{ KB} / 7.5 \text{ KBseg} = 0.5 \text{ seg} + 5.33 \text{ seg} \approx 6 \text{ segundos}$$

Aunque este tiempo parezca razonable, no es aceptable, porque en la mayoría de las conexiones no tendremos los 64kbps sólo para bajar el HTML, sino también las imágenes, saturación de la red, etc., por lo que fácilmente dicho valor se duplica.

Con la forma de especificar las tablas en la figura anterior, se puede obtener los mismos resultados visuales con una mejora sustancial en el tiempo de visualización. Supongamos que la tabla CABECERA está definida por los primeros 3KB (más que suficiente en la mayoría de los casos), por lo tanto el tiempo que esperará el cliente para empezar a pre-visualizar la página será:

$$T_{vis} = 0.5 \text{ seg} + 4 \text{ KB} / 7.5 \text{ KBseg} = 0.5 \text{ seg} + 0.53 \text{ seg} \approx 1 \text{ segundo}$$

Como se puede ver, obtendríamos una mejora de casi un 700% con sólo estructurar mejor las tablas. Si la estructuras de las páginas son más complejas, conviene seguir con la técnica de sub-dividir horizontalmente a las tablas:

3. Desarrollo

En esta fase se desarrollan los gráficos, el código HTML y los scripts necesarios. Aquí aparece el problema, los responsables de proyectos buscan que todo esto se haga con una sola aplicación.

Es mejor dividir el desarrollo en dos partes claramente diferenciadas y que se podrán desarrollar en paralelo.

1. Desarrollo gráfico y HTML
2. Desarrollo de programas

3.1 Desarrollo gráfico y HTML

Con las herramientas existentes, normalmente es el mismo diseñador el que se encarga del diseño gráfico (sobre todo en proyectos pequeños). Cada diseñador tiene sus propias herramientas y distintas plataformas (como por ejemplo trabajar con CMS)

Lo mejor que se puede hacer es que el o los diseñadores desarrollen todas las páginas con HTML estático y texto falso con su herramienta favorita, luego el texto falso se reemplazará por las partes de código (PHP) que generen la salida deseada. Para evitar problemas de integración, es mejor que los diseñadores trabajen asistidos por los programadores para decidir que tipo de texto salida se puede poner en cada página.

Para facilitar la posterior integración con los programas, a la hora de la maquetación en HTML hay que tener en cuenta lo siguiente:

1. Poner comentarios en HTML del tipo de salida o consulta en cada sección.
2. Usar desde el principio nombres de fichero con las extensiones correspondientes que se usarán en la versión definitiva (php, phtml, pl, cgi, etc.). De esta forma se evitarán problemas de enlaces por inconsistencia en los nombres.
3. Usar siempre enlaces relativos, de esta forma se podrá mover fácilmente los programas a distintos directorios o servidores virtuales.
4. Poner todo lo que sea código JavaScript o imágenes en subdirectorios y usar referencias relativas.
5. Grabar todos los ficheros en formato UNIX y con hora GMT.

3.2 Desarrollo de programas

Al mismo tiempo que el o los diseñadores

preparan el código HTML, el o los programadores deben preparar los programas usando HTML muy simples para probar la base de datos y ejecución de los programas.

Las herramientas que se usen en estas fases serán normalmente aquellas que use el programador y que son básicamente un editor de texto y acceso al servidor por medio de TELNET (o SSH). La ventaja principal es que permiten a los programadores libertad de horarios y de ubicación, ya que la mayoría de este trabajo puede ser realizado remotamente. En caso de trabajo en grupo se recomienda el uso del CVS, ya que permitirá el control de concurrencia, cambios, parches y versiones con mucha facilidad.

Lo más importante es tener acabada la mayor parte de la programación antes que se acabe el diseño e integrar paulatinamente los módulos acabados en aquellas páginas ya finalizadas por los diseñadores. Además se debe controlar periódicamente que los enlaces y referencias de las páginas HTML sean válidos y relativos.

Pruebas

Además de los cambios inevitables que surgirán cuando el cliente vea como funcionan las páginas hay que ir realizando pruebas para verificar que los tiempos de respuesta, retorno y de descarga son adecuados y controlar que el sistema soporta la cantidad de conexiones simultáneas que puede haber.

Pruebas de tiempos de respuesta y retorno

Esta prueba no permitirá saber cuanto tiempo tarda el servidor en generar la salida HTML. Podemos tener una aproximación bastante buena si medimos el tiempo total que tardan en bajarse las páginas en una red local (como el ancho de banda es elevado, el tiempo de descarga es prácticamente igual al tiempo de retorno).

Hay otro dato importante al hacer este tipo de prueba, Transfer rate nos indica el tráfico que ha sido capaz de generar el servidor en Kilobytes por segundo. Si llevamos eso a kilobits por segundo (multiplicar el valor por 10 para obtener una aproximación debido a los overhead del TCP/IP), veremos que el servidor (un Linux PIII, 500 Mhz, con Apache, PHP y MySQL) **es capaz de saturar una línea de ¡6 mbps!**

En este caso no habrá problemas de saturación de CPU a menos que tengamos una línea de Internet con una capacidad superior.

El segundo parámetro a investigar es la cantidad de conexiones simultáneas que puede soportar. La estimación de conexiones simultáneas es compleja e incierta, pero una buena aproximación sería estimar las conexiones promedio por día y multiplicar ese valor por 3 a 5 para obtener los picos que veríamos en el servidor.

Si calculamos que se servirán unas 200.000 páginas diarias, tendremos que por segundo puede haber unas 2.3 conexiones (200.000/86.400 segundos día). Si multiplicamos ese valor por 5 nos queda unas 11.5 conexiones por segundo de pico.

No confiar en la velocidad de la red local: el tiempo de visualización será ínfimo comparado con los mejores tiempos que se puedan obtener a través de Internet. En el caso de pruebas de red local, la página estará descargada en menos de la mitad de una décima de segundo (0.04 seg), que está muy lejos de los 6 segundos del caso "normal". No olvidar que hay que hacer pruebas y estimaciones no sólo en la red local donde se hace el desarrollo, sino con también con una navegador estándar y conexiones con modem. Las diferencias son enormes y pueden generar sorpresas y muchas de ellas desagradables.

Cálculo del ancho de banda necesario

La forma más común (y empírica) para calcular los requerimientos máximos es tomar el promedio de tráfico diario y multiplicarlo por 3 (como mínimo) para obtener el tráfico máximo aproximado. En el ejemplo mencionamos 200.000 páginas por día (2.3 por segundo). Si el promedio de cada página es de unos 28.000 bytes (suelen ser un poco más debido a las imágenes que se descargan una vez), entonces el tráfico promedio y máximo serán:

Trafpromedio = 2.3 páginas/seg * 28.000 * 10 = 644 kbps

Trafmaximo = 2.3 páginas/seg * 28.000 * 3 * 10 = 1.932 kbps ≈ 2 Mbps

Esto nos da una idea del tipo de línea a Internet (o ancho de banda) que tendremos que contratar para atender adecuadamente los picos de tráfico.

4. Producción

El sitio ya entra en producción y su acceso es público. Aunque esta parece la más sencilla de las fases, es la más cara y difícil de conseguir. Hay que actualizar los datos para que el sitio sea interesante y visitado y además asegurarse que funcione las 24 horas del día, los 7 días de la semana (24x7)...un constante mantenimiento.

Si usamos Linux, Apache, MySQL/Postgres y PHP/Perl oPython (LAMP), no tendremos ningún problema en sobrevivir muy tranquilamente a esta fase. 🇵🇪

Referencias:

<http://es.wikipedia.org/wiki/CVS>
<http://www.desarrolloweb.com/>
http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web
<http://bulma.net/body.phtml?nIdNoticia=734&nIdPage=2>
<http://www.maestrosdelweb.com/editorial/desarrolloweb/>
<http://soporte.tiendalinux.com/portal/Community/Web>
http://www.artedinamico.com/ad/ad_estaticas_mostrar.php?id=21
http://livedocs.macromedia.com/studio/8_es/exploring/wwhelp/wwhimpl/common/html/wwhelp.htm?context=LiveDocs_Parts&file=04_workf.htm

Sara Mostajo es un experta desarrolladora web, promotora del software libre quien a fundado el Grupo de Usuarios GNU/Linux Linux Chix en Peru.

www.linuxchixperu.org

se lo que hicieron el chetumal pasado

Tec. Victor Hugo Cordova Madrid
victor.hugo@revista-sl.org



Bueno antes que todo, permítanme extenderles una disculpa a todos ustedes – nuestros lectores que hacen posible esta publicación y que nos motivan a seguir trabajando – por no haber publicado a tiempo esta segunda y última parte de la reseña de lo que fue el “1er Congreso de Software Libre Chetumal, Q.R. – Belice”, pero por cuestiones que se salieron un poco de mis manos estuve algo alejado del movimiento del SL, afortunadamente ya estamos de vuelta y bueno, a seguir trabajando ;)

Segundo Round.

El segundo día del congreso (Sábado 8 de Abril) comenzó con el taller llamado “[Un vistazo al proyecto mono: .Net en Linux](#)” impartido por uno de los diseñadores gráficos de RevistaSL Josué Gutiérrez Hernández (BlackNash) donde se expuso el desarrollo de interfaces con Glade, C# y Mono.

Alrededor de las 11 de la mañana, en el teatro, dio inició la conferencia “[The Cerros Model](#)”, seguida de otra conferencia llamada “[Markus: Desarrollo de Software Libre para Escuelas](#)”, ambas impartidas por los miembros de la Comunidad Linux de Belice, estas conferencias trataron acerca de unos proyectos que nuestros hermanos Beliceños están llevando a cabo en ese país.

De ahí siguió la plática “[Infraestructura de](#)

[servicios con FreeBSD: Red ITC](#)” la cual expuso Alejandro Acosta Arjona en donde se mostró la infraestructura de servicios con FreeBSD de la Red ITC como: Antecedentes Infiltración - NAT Posicionamiento - Oportunidad - Web cache Sustitución - Software discontinuado - Web - DNS Crecimiento - Servicios de correo, entre otros...

Cerca de la 1:00 p.m. el IcenetX Team llevó a cabo en el Salón Latinoamericano el taller “[Servidores LAMP](#)” en donde se trató de Explicar de manera teórica y práctica, la manera correcta de implementar un servidor LAMP (Linux Apache, MySQL y PHP). Contenido: Introducción a LAMP's, qué las forma, por qué y para qué sirven. Campo de aplicación (Internet, Intranet y Extranet), instalación de Apache, MySQL y PHP sobre Linux, configuraciones de Seguridad. Este taller fue muy interesante y por lo mismo se extendió casi una hora más de lo programado, pero los muchachos de IcenetX se portaron a la altura =)

Después de un breve descanso reanudamos labores a las 16:00 horas entrando de lleno a la conferencia impartida por el Ing. Héctor Leal Morales (Etosh), miembro del Consejo Editorial SL, llamada “[Tu gusto por la Web, llévalo a producción](#)”, en donde se trataron puntos acerca de cómo poder sacar provecho de los talentos que muchas personas y desarrolladores web suelen tener, como un hobby se puede convertir en una fuente de ingresos y un buen trabajo.

Al mismo tiempo en el salón latinoamericano se



impartía el taller “[Introducción a la programación con Python](#)” que nos hizo el favor de dar el Lic. Julio Acuña Carrillo (urkonn) en donde se desarrollaron pequeñas aplicaciones tanto en modo texto como con interfaces gráficas de usuario que ejemplificaron el uso de Python.

A las 17:00 horas se llevo a cabo la conferencia “[Cómo vender software libre a empresas corporativas](#)”, nuestro buen amigo Leo Utskot nos explicó como es un verdadero reto entrar en una empresa corporativa y vender software libre. Por un lado tienes que pelear en contra de Microsoft, Oracle y otras empresas grandes de software que tienen miles de empleados y que destinan millones de dólares en publicidad y propaganda. Por otro lado hay que enfrentarse con los dogmas de las áreas de sistemas.

A las 18:00 horas presenciamos una plática acerca de la [Conformación de un cluster con apoyos ProSoft](#), impartida por Alejandro Bautista, en donde se explicó de que forma se puede formar un cluster con el apoyo de ProSoft, cuáles son los pasos y metodología que se tienen que seguir y respetar, y se invitó a empresas y desarrolladores presentes a formar parte del proyecto de la conformación de un cluster en el estado de Quintana Roo.

Finalmente llegamos al momento más esperado del día, la conferencia magistral, donde tuvimos como invitado especial a una de las personas más importantes, activas y conocidas en el mundo del SL de México, nuestro queridísimo y respetado amigo Gunnar Wolf Iszaevich, el cual nos hizo el favor de hablarnos sobre lo que es el [Proyecto Debian](#), su filosofía, estatutos, formas

trabajar, versiones, y muchísimas cosas más. Una plática sumamente constructiva donde el público se animó mucho a preguntar y donde también tuvimos una primicia internacional, ya que Gunnar dio a conocer el nombramiento del nuevo líder del Proyecto Debian, ya que las elecciones acababan de terminar tan solo unos minutos antes. Con esto terminamos un muy movido y productivo segundo día del congreso, así que nos dispusimos a ir a cenar y prepararnos para el último y tercer día, el cual prometía ser maratónico!!!



Seguridad y más seguridad, Round Final ;)

Empezamos el último día del congreso a las 09:00 horas con lo que desde mi particular punto de vista fue el taller más esperado de todos, el taller de “[Seguridad en aplicaciones Web](#)” impartido por José Carlos Nieto Jarquin (xiam - Gekkkoware) abarcando: Introducción (Ética hacker) - Entendiendo protocolos (descripción y prácticas sobre HTTP y SMTP) - Read The Fucking Manual/RFC! - Formas "brutas" de ataque. (Descripción y prácticas de D.D.o.S., fuerza bruta, obtención automatizada de proxies, construcción de peticiones) - Encontrando errores en código fuente. - ¿Cómo escribir exploits? - Explotando SQL Injection, XSS. - Salto de protecciones (captcha, falseo de sesiones, falseo de IP normal y vía proxy) - Encontrando errores en aplicaciones de forma ciega. - Formas de protección contra ataques.

A las 11:00 horas el IcenetX Team (BRIO, gaper y ayzax) dieron la conferencia “[Seguridad informática](#)” en donde el objetivo principal fue Conocer los riesgos actuales y métodos de defensa y protección

más importantes para la información. Seguridad informática Sobre Hacking y sus ramas Virus, phishing y spyware Firewalling, NAT y routing.

Continuamos con la conferencia “[Configuración de un Servidor FTP seguro](#)” a las 12:00 horas, en donde Adib Saavedra Bocanegra explicó que en la actualidad es muy común la transferencia de archivos usando el protocolo FTP sobre un ambiente *nix y ocupando cuentas del sistema, esto hace engorroso el trabajo de la administración de usuarios, ensucia el archivo passwd y convierte en vulnerable e inseguro el sistema, aquí es cuando se hace necesario un método de autenticación externo y que no involucre cuentas shell en esta plática nos explicaron como instalar un servidor FTP que se autentifique contra un servidor de bases de datos.

Y pues, antes de irnos a degustar de una buena comida, como entremés presenciamos la plática “[Firewall's y el port knocking](#)” impartida por nuestro gran amigo (y jefe =P) Gonzalo Javier González Rodríguez (chalo), en donde se mostró al público las ventajas de tener un firewall con GNU/Linux y hacer más seguro nuestro acceso mediante la técnica del port knocking.



Después del almuerzo reanudamos actividades a las 16:00 horas con la conferencia “[Un host bajo la mira](#)”, donde Carlos Augusto Lozano Vargas (vendetta) nos dio una idea general de qué es lo que busca un atacante en un host, como es que sucede, qué herramientas se pueden usar para realizarlo y algunas sugerencias de protección.

A la misma hora, pero en el Salón Latinoamericano Alejandro Acosta estaba dando



el taller de [Instalación y configuración de FreeBSD](#), en donde se llevó de la mano a los asistentes para poder realizar una correcta instalación del sistema operativo antes mencionado y poder configurarlo de tal forma que se logre una total compatibilidad con el hardware.

Acercándonos al final del Congreso, tuvimos la oportunidad de presenciar la plática “[Conoce a tu Enemigo: HONEYPOTS](#)” con Miguel José Hernández y López (el buen Mike) en donde vimos que los HONEYPOTS han ganado en popularidad y aceptación, presentándose como una herramienta eficaz y poderosa ante los hackers. En esta plática se abordó qué son, qué tipo de Honeypots hay, para qué sirven, y se platicó acerca de HoneyD importante software (open source) para la creación de Honeypots.



Y bueno como plato fuerte y final tuvimos la fenomenal presencia de Sandino Araico Sánchez quien impartió la conferencia magistral “[Más allá del falso sentido de la seguridad](#)”, por mucho, fue la conferencia con más asistencia e interés de parte de la audiencia, llenando casi por completo el Teatro Constituyentes (con una capacidad de cerca de 500 gentes), en esta plática Sandino nos explicó el concepto de un “Zero Day”, y con su ya conocida y particular forma de decir las cosas nos adentró en el mundo de la seguridad informática, desvaneciendo muchos mitos que se tienen en el entorno informático.

Así dimos por finalizado el “[1er Congreso de Software Libre Chetumal, Q.R. – Belice](#)” y no nos resta más que agradecer a todas y cada una de las personas que hicieron posible esto, desde los ponentes, miembros de comunidades linuxeras de otras partes de la república, el staff, hasta todos los asistentes, estudiantes, profesionistas y patrocinadores que nos estuvieron apoyando durante todo el evento de forma incondicional. Sabemos que tuvimos muchos errores de organización, hubieron cosas que no tomamos en cuenta a tiempo y que desembocaron en errores, al fin y al cabo humanos, además de que fue la primera vez en que nos animamos a organizar un evento de tal magnitud. Públicamente quiero hacerles llegar, de parte de la CLC, una atenta disculpa por todas las molestias que les pudimos haber causado con nuestros errores y esperamos poder contar de nuevo con su apoyo en futuros eventos.



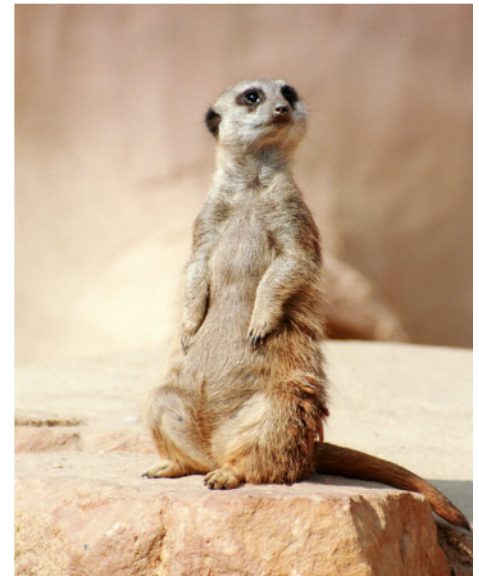
Actualmente, ya estamos empezando a ver algunos detalles para organizar la segunda edición de este congreso, animados por muchas personas que se quedaron con un muy grato sabor de boca de la edición pasada y pues los mantendremos informados a través de esta que es su Revista y su sección de EventosSL.

Gracias!!! 🇬🇧

suricata

un desarrollo mexicano libre

Quetzalcoatl Pantoja Hinojosa
 quetzalcoatl.pantoja@gmail.com



Suricata es un administrador de proyectos que nació en la UAEM (Universidad Autónoma del Estado de México) como tesina, actualmente se encuentra liberado bajo licencia LGPL en el sitio de Sourceforge.

El objetivo principal de Suricata, es convertirse en una herramienta utilizable en una PyME para llevar a cabo actividades de administración dentro de la organización, desde un punto de vista colaborativo, por lo que esta herramienta fue pensada para usarse dentro de un ambiente web.

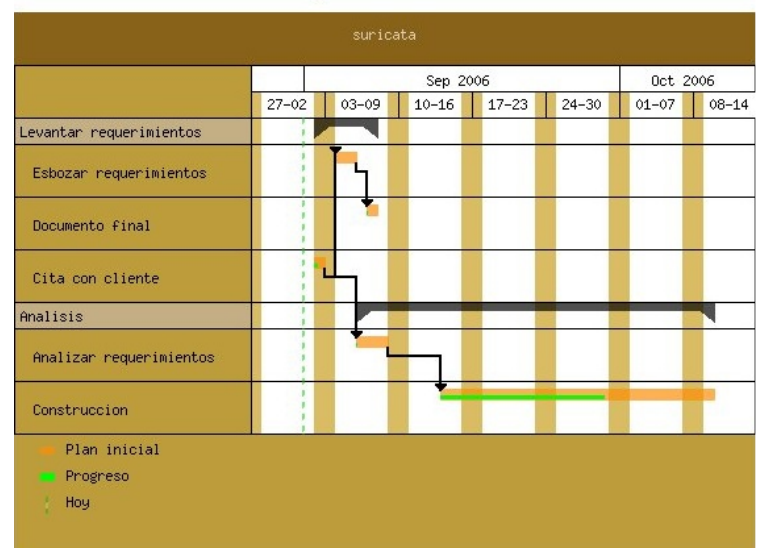
Para llevar a cabo el desarrollo, se utilizó PHP5 en conjunto con PEAR::DB y PostgreSQL, gracias que se usó PEAR::DB, es fácilmente portable a otros manejadores de bases de datos; otra tecnología que también se implementa en el desarrollo es el uso de CSS, lo que permite cambiar el aspecto con relativa sencillez. Por otra parte, está pensado para portarse a otros idiomas y soportar diversas nomenclaturas por proyecto.

Dentro del paquete descargable de Suricata, se encuentra un pequeño tutorial sobre su funcionamiento en formato de screencasts.



Presentación de Suricata en el EnLi

Diagrama de Gantt



La tesina, cuenta con licencia CC, ya que en ella se documenta la estructura utilizada y un pequeño apartado sobre administración básica de proyectos, se espera que esta documentación crezca proporcionalmente con el proyecto, de esta forma la incorporación de voluntarios en el proyecto será fácil. 🤖

**Para mas información sobre Suricata
contacta al Autor o visita:**

<http://www.sourceforge.net/projects/suricata>

free fall

Leo Utskot
leo@copyleft.com.mx



Copyleft México junto con su socio comercial Inteligencia Productiva lanzan la versión 1.0 de una herramienta de subastas inversas en línea llamada **Free Fall**. Disponible en <http://freefall.copyleft.com.mx/>, el software forma parte de la estrategia de ambas empresas de "productizar" varios de sus desarrollos para apoyar la comunidad de software libre en México y brindar un mejor servicio a sus cliente corporativos.

"La necesidad de proveer herramientas robustas y fáciles de usar crece cada día en México y tenemos varios clientes listos para usar Free Fall" comenta Miguel Angel Pereyra, Director de Inteligencia Productiva. Respecto a la necesidad en el mercado, comenta: "Durante los últimos 7 años, entre otros servicios, hemos liderado para nuestros clientes en Argentina, Brasil y México distintos procesos de procura con adjudicación por subasta inversa. Siempre hubo una diferencia entre los procesos reales del mercado y lo que los sistemas disponibles ofrecían, convirtiéndose más en un obstáculo que en un facilitador. Finalmente al aliarnos con Copyleft encontramos un interlocutor dispuesto a realmente hacernos más fácil la vida, y fue entonces que nos sentamos, diseñamos Free Fall y ellos lo desarrollaron." Sobre el enfoque con que se atacó el problema, Leo Utskot, Director General de Copyleft agrega: "En base a la premisa de que "menos es más", tomamos las necesidades que conocíamos y desarrollamos una solución simple y robusta.

Antes que sobrecargar a Free Fall con funcionalidades bonitas pero poco prácticas o de uso poco habitual, decidimos incluir solo aquellas que sabemos que son realmente imprescindibles."

Respecto a la decisión de liberar Free Fall como software libre, comenta Miguel Angel Pereyra: "Estamos convencidos de que el valor agregado a un cliente que decida empezar a comprar por subasta inversa es el servicio que podamos darle. Nos respaldan los varios años de experiencia en el desarrollo de estos procesos y los muchos pesos ahorrados a nuestros clientes, un sistema como Free Fall es una ayuda muy importante pero no es la clave para el éxito de todo un proceso de procura con adjudicación por subasta inversa en línea. No vemos ningún problema en que el sistema esté disponible libremente y el que lo desee pueda tomarlo para ofrecer servicios como el nuestro. No le tenemos miedo a competir en igualdad de condiciones, y además cuantos más seamos más fuerte será la "evangelización", hoy aún incipiente".

Desarrollado con tecnología de software libre, PHP y MySQL, y basado en los nuevos conceptos conocidos como Web 2.0, Free Fall es una solución de bajo costo pero al mismo tiempo de vanguardia.

Copyleft México (<http://www.copyleft.com.mx>) es la filial de Copyleft Industries (Oslo, Noruega) y desarrolla soluciones de sistemas sobre plataforma web como software libre.



FREE FALL REVERSE AUCTION

Subastas inversas, simplificado.

Inteligencia Productiva (<http://www.inteligenciaproductiva.com>) es una empresa mexicana que ofrece al mercado corporativo soluciones de sistemas para sus procesos en formato ASP.

Subastas inversas en línea son procesos de selección de proveedores en los cuales varias

empresas invitadas compiten a través de Internet por bajar el precio de adjudicación de un contrato de servicios o provisión de insumos.

Para más información visita <http://freefall.copyleft.com.mx/>

Software Libre

Hospedaje Web

Libros

Sourveniers

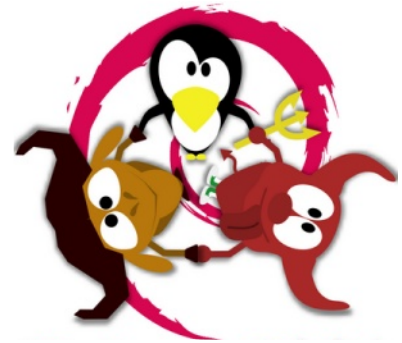
.... tu tienda libre en Internet



BAZARLIBRE.COM

comunidad linux chetumal

Tec. Gonzalo González Rodríguez
gonzalo.gonzalez@revista-sl.orgx



Primero unas definiciones:

Comunidad: Una comunidad es un grupo o conjunto de personas (o agentes) que comparten elementos en común, elementos tales como un idioma, costumbres, valores, tareas, visión de mundo, edad, ubicación geográfica (un barrio por ejemplo), estatus social, roles, etc. Por lo general una comunidad se crea una identidad común, en base a diferenciarse de otros grupos o comunidades (generalmente signos o acciones), que es compartido y elaborado entre sus integrantes y socializado. Uno de los propósitos de una comunidad es unirse alrededor de un objetivo en común como puede ser el bien común. Aun cuando se señalo anteriormente basta una identidad en común para conformar una comunidad sin la necesidad de un objetivo específico.

Fuente: <http://es.wikipedia.org/wiki/Linux>

Linux: Estrictamente se refiere al núcleo Linux, pero es más comúnmente utilizado para describir un sistema operativo similar a Unix pero libre, también conocido como GNU/Linux, formado mediante la combinación del núcleo Linux con las bibliotecas y herramientas del proyecto GNU y de muchos otros proyectos/grupos de software libre.

El término "Linux" se utiliza actualmente para distribuciones Linux completas, las cuales suelen contener grandes cantidades de software

además del núcleo, partiendo de servidores web como Apache a entornos gráficos como GNOME o KDE y suites ofimáticas como OpenOffice.org.

Fuente: <http://es.wikipedia.org/wiki/Linux>

SOFTWARE LIBRE

El "Software Libre" es un asunto de libertad, no de precio. Para entender el concepto, debes pensar en "libre" como en "libertad de expresión", no como en "cerveza gratis" [N. del T.: en inglés una misma palabra (free) significa tanto libre como gratis, lo que ha dado lugar a cierta confusión]. "Software Libre" se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

La libertad de usar el programa, con cualquier propósito (libertad 0). La libertad de estudiar cómo funciona el programa, y adaptarlo a tus necesidades (libertad 1). El acceso al código fuente es una condición previa para esto. La libertad de distribuir copias, con lo que puedes ayudar a tu vecino (libertad 2). La libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. (libertad 3). El acceso al código fuente es un requisito previo para esto.

Fuente: <http://www.gnu.org/philosophy/free-sw.es.html>



Comunidad Linux Chetumal

Chetumal: Ciudad Chetumal. Capital del estado de Quintana Roo, México. Se localiza en el sur del Estado de Quintana Roo, entre los 19° 14' y 70° 50' de latitud norte y los 87° 15' y 89° 25' de longitud oeste. Al sur hace frontera con Belize.

La Comunidad Linux Chetumal

Definición.

Ya que tenemos entendido las definiciones, podemos concluir que la Comunidad Linux Chetumal es un grupo o conjunto de personas que comparten un elemento común: Linux.

Nuestra definición es: La Comunidad Linux Chetumal es un grupo o conjunto de personas que tienen como objetivo y propósito difundir el uso de Linux como sistema operativo por excelencia; ser parte del movimiento Open Source, apoyando su difusión y colaboración en los proyectos, teniendo como base la Ciudad de Chetumal, Quintana Roo, México.

Creación.

Esta comunidad nace por la necesidad de difundir el sistema operativo Linux y el movimiento Open Source y de reunir la gran información que podemos recopilar de la Internet y de las experiencias propias de los miembros de esta comunidad.

Unir a las personas que ya siguen en el movimiento del Open Source para apoyarnos mutuamente y ayudar a los novatos.

Para esta comunidad el Open Source, permitirá que el software mexicano sea cada día más común, a nivel nacional e internacional. También permitirá que dicho software se cree con una gran calidad, ya que tendrá influencias de todas partes del mundo.

Fundadores.

Los fundadores de esta comunidad, inicialmente, somos cuatro compañeros. En seguida los enlistamos en orden alfabético:

Carlos Ricardo Villanueva Partida (Nopal)
Christian Peraza Molina (xeneixe)
Gonzalo Javier González Rodríguez (chalo)
Víctor Hugo Cordova Madrid (ViChO)

Nopal, xeneixe y ViChO son de la ciudad de Chetumal, Quintana Roo, México; chalo es de Calderitas, Quintana Roo, México. Somos usuarios activos del sistema operativo GNU/Linux, ya sea en nuestra casa, escuela o/y trabajo.

Comunidades SL es una nueva sección en donde mostraremos a la gente que está en las trincheras del movimiento de software libre.

fabien manniser

Lic. Julio Acuña Carrillo
julio.acuna@revista-sl.orgx



RSL - Fabien, para los que no te conocen, pláticanos acerca de tí

FM – Bueno, pues yo nací en el 68 no en mayo pero en noviembre allá en Francia en la ciudad de Lens, tal vez algunos la puedan ubicar porque el equipo ya a veces anda en las copas europeas, allí pasé casi toda mi vida, estudié la ingeniería industrial, siempre me ha gustado la computación pero decía que no me gustaría estudiar cosas que ya domino, igual uno se aburre en clase y también tenía interés en para qué se utilizan las computadoras por eso me metí en una escuela en que la base es ingeniería industrial pero con una especialidad en ciencias computacionales y en Europa existe un programa que se llama Erasmus que favorece mucho el intercambio entre universidades y me lancé a Londres a estudiar un año en King's College y allí conocí a la que sería mi esposa, mexicana, allá nos casamos en el 91 vivimos tres años y el 28 de junio del 94, nunca me olvido de la fecha, llegamos aquí a México, realmente no dominaba mucho el español y para encontrar trabajo no era tan fácil y uno de mis amigos en Francia me había comentado que existía algo que se llamaba Linux, él se había comprado su primer distribución y le dije mándame una copia, me la mandó por paquetería y pues de los ahorros que teníamos me compré una computadora y empecé a explorar y dije a ver si me pongo a vender eso, yo veía en México que había mas posibilidad de ser empresario que ser empleado y me lancé

con la empresa; tenía mi impresora de inyección de tinta a color y sacaba los folletos y todo, pero realmente nadie lo quería y allí es cuando caí con el Tec de Monterrey y me dijeron ven aquí a recibir el internet, UNIX es necesario para recibir el internet, Linux puede ser una alternativa y es como entré en el Tec, allí me encargué durante ocho años de los servicios computacionales aquí en el campus Toluca, no tengo la certeza al 100% pero creo que pusimos los primeros servidores Linux en producción contra todos los dictámenes que podía haber y pues muchs veces nos ha funcionado bastante bien, mas que nada para lo que es DNS, servicio web, el DHCP y todo eso nos ha dado muy buen resultado. Siempre lo he tratado de hacer de la manera mas barata posible, es decir la gratuita y he tenido que migrar a través de muchas versiones, pusimos Red Hat pero cuando dan la versión de descarga es mas enfocado al escritorio no tanto a servidor, después pasé por Mandrake, después hubo Debian, no me gustó mucho porque no estaba suficientemente optimizado para mi en la parte de servidor y después me metí por Gentoo pero ahí es compila todo, ahorita creo que están con la de Ubuntu la versión para servidor, están quitando toda la parte gráfica y creo que está dando un buen resultado y a fin de cuentas Ubuntu también es Debian. Y bueno yo dentro del Tec estudié la maestría en comercio electrónico, ya tenía la parte de cómo utilizar la compu, la parte ingenieril **** estudiar la parte de administración y por eso estudié la parte de comercio electrónico y también por todo el interés que tenía por todo lo que es web, aquí en el campus viví las desveladas cuando teníamos que

instalar el sistema de inscripción y los alumnos allí haciendo fila porque había un número de computadoras limitada y pude descubrir cuando pasamos todo eso a la web cómo ha ayudado a que hoy en día incluso los alumnos se van de vacaciones a Acapulco y desde cualquier cybercafé se inscriben; incluso ahora lo toman como algo muy natural, pero es allí donde yo viendo esta transición pude descubrir lo que la web nos puede aportar, en movilidad, en facilidad de acceso, entonces me enfoqué mucho a aplicaciones web. En el año 2000 fundamos con algunos compañeros una empresa que se llamó Operaciones digitales de Latinoamérica pero creo que el plan de negocios no era muy sólido y ellos lo dejaron, sin embargo yo quiero retomar eso y encontrar la manera de lograr dos cosas, apoyar al software libre, su uso, lograr que las empresas se concienticen de que no siempre porque pagas tienes la mejor solución y también poder vivir de eso.

[RSL – En todos los congresos que te hemos visto has dado una plática que se llama Desarrollo Ágil de Aplicaciones Web con Ruby on Rails, ¿nos podrías platicar un poco de Ruby on Rails?](#)

FM – Bueno un poco de métodos ágiles primero. La idea es: en las universidades enseñan lo que en muchas empresas hacen donde pasamos por un proceso largo de diseño de sistemas, el problema es que este proceso largo se hace porque no es fácil adaptar un sistema a cambios, cuando utilizamos una metodología hacemos análisis de requerimientos, diseño de interfaz y todo eso y después se programa, el problema es que este programa muchas veces son miles y miles de líneas de código y cualquier cambio que nos dé el cliente no es fácil, entonces para evitar esto de que implementamos el sistema y el cliente nos dice: no esto no corresponde a lo que quiero, se enfoca mucho a que el análisis de requerimientos debe ser muy fuerte, muy robusto para que no pase eso después, el problema es que eso funcionaba muy bien mientras estamos en sistemas tradicionales o sea cuando estábamos en los 80s ponme un sistema que va a durar veinte años incluso porque espero que mi mainframe me dure veinte años, pues todo eso funcionaba muy bien el problema otra vez es que en la parte de web todo cambia constantemente, nacen nuevas tecnologías y llegó el AJAX y de repente ya es la moda y todo el mundo redescubre que hay nuevas facilidades en la web, entonces si tu tienes una parte muy

estructurada de diseño de interfaz pues allí no se te acopla, tienes muchos problemas porque tu cuando empiezas con tu análisis de requerimientos y llegas a la aplicación final y si pasa un año, en un año el mundo ya cambió, entonces necesitamos ser mas ágiles para hacer las cosas de manera mas rápida y eso implica dos cosas, una que tenemos que trabajar diferente, a lo mejor no tardarnos tanto tiempo en el análisis de requerimientos, entrarle al código lo mas pronto posible y también tener herramientas que nos favorezcan el cambio, que nos favorezcan adaptarnos a nuevas cosas y de hecho algo que siempre comentaba en las pláticas es que necesitamos ir hacia nuevas herramientas, no podemos pasar por ejemplo por este proceso tradicional de hacer las cosas, es pérdida de tiempo, entonces Ruby es en parte eso por ejemplo el hecho de que una clase no se cierre y se pueda abrir y tengo algo que tenga que evolucionar, eso se soluciona poniendo un archivo anexo retomo la clase y le adiciono cosas, no tengo que enredar clases, crear nuevas que son abstractas y que no sirven de nada, entonces esta facilidad hace que yo pueda adaptar el cambio, tener estos métodos ágiles que yo justamente pueda tener estos métodos ágiles que es vámos a trabajar de la mano tu cliente y yo y vamos a hacer que tú tengas éxito porque si tú tienes éxito yo como tu proveedor de tecnología yo también, y la clave para tener éxito es ser ágiles, sacar tu producto lo mas seguido posible como Google siempre está en beta dando el mensaje de que siempre estamos mejorando nuestros sistemas y siempre tratando de llegar a lo mejor entonces de allí siempre la idea de decir: bueno, hay que adoptar estos métodos ágiles un poco mas a la parte de aplicaciones web mas que a la parte de aplicaciones tradicionales de escritorio y el hecho de que Ruby es una muy buena alternativa por las opciones que ofrece de soportar esta agilidad.

[RSL - ¿Qué hace tan especial a Ruby on Rails?, hemos visto a través de los años aplicaciones en PHP, ASP, java, ¿qué tiene de especial este framework?](#)

FM -Lo que hace especial a este framework es que no es una nueva propuesta de lo mismo, lo que ofrece ASP, JSP, PHP todo es lo mismo, es tu puedes mezclar las cosas, por ejemplo yo he visto código en PHP horrible y puesto en producción, en una sola página tienes todo mezclado, el acceso a la base de datos, te pones a analizar el código y dices: eso lo van a hackear en 5 minutos. Empiezas desde la hoja blanca, tienes que rehacer todo y a

veces por las presiones no hay un buen diseño al inicio y no separó bien las cosas y haces un sistema que saca la bronca en el momento, el problema es que si tenemos que evolucionar ya se vuelve invivible, yo creo que el mejor ejemplo, y ahí a lo mejor me voy a echar enemigos, es PHP-Nuke, cuando yo entré en el código de esta cosa yo dije: para su autor mis respetos ¿cómo lo logra mantener?, es un código spaghetti tremendo, tu empiezas a ver que ciertas funciones se llaman en otra pero no tiene sentido que esté así, nada más por el tiempo seguramente como ha evolucionado se ha dado, pero yo digo: cómo le hace para mantener eso y sacar una nueva versión, todo está mezclado, de hecho para mí el mejor ejemplo para mí es PostNuke, se tardaron 3 o 4 años en tener algo más estructurado siguiendo la base de PHP-Nuke pero estructurado y poder evolucionar de manera más racional, eso es lo que nos dan esos lenguajes, podemos hacer lo que queremos y muchas veces podemos hacer lo peor que se puede hacer; entonces Ruby on Rails si nos ayuda bastante, aprovecha el lenguaje Ruby, su dinamismo, pero también nos da la posibilidad de que las mejores prácticas estén puestas allí. Una de las mejores prácticas es por ejemplo es adoptar el modelo MVC, en Java no estás obligado, si tu lo quieres hacer lo puedes hacer a través de algo como Struts, pero por ejemplo aquí cuando yo busqué libros no los hay, los tienes que importar, no es una práctica común, entonces cuando veo eso yo pienso que la gente que programa en Java igual no programa separando bien las capas de la aplicación web, y Ruby on Rails ya no necesitas libro, está puesto, su estructura, su base como está al inicio es esa, dividir las cosas, permitir que tu aplicación pueda crecer porque todo está tomado, también igual, los programas de una empresa, si yo tengo un empleado que hace un programa en PHP, se me va la persona y llega otro y dice: si es PHP pero no es mi estilo, no le entiendo a la aplicación incluso siento que tengo que reescribir las cosas porque no es mi estilo; cuando en Ruby on Rails al encuadrarnos en cómo estamos haciendo la aplicación permite que si haya la posibilidad de uno retomar la aplicación de manera más sencilla porque sabemos dónde están las cosas, nos impone una estructura, muchos van a decir la imposición no es buena, pero a veces cuando la imposición nos da que con eso vengan las mejores prácticas y cuando está apoyado en el hecho de que es software libre y puedo influir en su desarrollo, en su diseño y hacer que las mejores

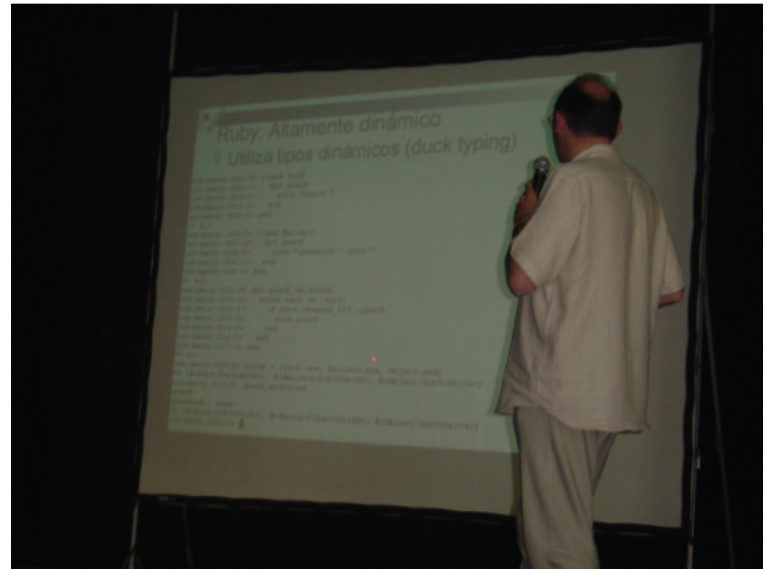
prácticas que no están allí y deberían estar allí puedo influir en que estén allí porque tengo un círculo con los desarrolladores que no tengo necesariamente con software privado, cuando tengo esto yo creo que es posible, es aguantable; entonces allí la idea es que ya no reinvento la rueda, la conexión a la base de datos está hecha, los métodos que te permiten meter javascript en los campos, ya todo está hecho, entonces tu ganas productividad en esta parte. También hay otros principios que son principios ágiles que se adoptan allí, por ejemplo uno fuerte que es el Don't Repeat Yourself (DRY) que es que si tu pones la misma cosa en varios lugares y si tienes que realizar un cambio entonces lo vas a tener que repetir y si lo haces 50 veces lo más probable es que lo hagas 49 veces y si te lo auditan una entonces introduces un bug allí, el hecho de que en Ruby on Rails venga este principio, se favorece este principio de no repetirse de que las cosas estén en un sólo lugar pero te ayuda a que cuando tengas que hacer un cambio es un solo cambio, entonces tu aplicación guarda su consistencia, hay otro también por ejemplo la convención sobre la configuración allí atacan un poquito más la parte de Java, el hecho de que cuando tu instalas Tomcat tienes que configurar una aplicación y generar archivos XML o crearlos tu a mano, pues es algo pesado y muchas veces pierdes tiempo en satisfacer las necesidades del sistema y no realmente hacer cosas que para tí son útiles, que aportan a la aplicación, bueno Ruby on Rails también ataca este problema por ejemplo uno es: vamos a hacer las clases y las tablas que correspondan a las clases simplemente son los nombres de las clases en su lugar, por ejemplo tengo una clase de personas, tengo una tabla que contiene personas, puedo cambiar eso, pero me conviene respetar la convención porque me quita la parte de configuración y entonces tengo un desarrollo más rápido.



RSL - ¿Qué proyectos has hecho en Ruby on Rails?

FM – La verdad no he hecho grandes proyectos porque me he dedicado mas a tratar de difundirlo, de darlo a conocer y pues muchas veces no te da tiempo de poder hacer todo. Hace dos años participé en Rails Day, algo muy interesante que no he visto en otros lenguajes, con Ruby on Rails hacen un concurso cada año, con premios y todo, donde la idea es que uno desarrolle un proyecto en un solo día, montan un servidor con Subversion y arranca un día a la media noche, generalmente nos favorece allí porque es hora de Estados Unidos, y tienes 24 horas para hacer tu proyecto, y la idea cuando nació este proyecto dijeron: queremos demostrar que en 24 horas con Ruby on Rails podemos hacer cosas que en PHP o Java no lo van a poder hacer porque va a ser muy difícil que tengan un proyecto muy visible en tan sólo 24 horas, entonces participé en el primero, no fueron muy buenos resultados porque allí apenas había comprado el libro y el libro no llegó a tiempo por la aduana, etc. porque habían liberado el libro como tres semanas antes, se apresuraron los de Pragmatic Programmer, dijeron: tenemos que sacar el libro porque si no cómo va a programar la gente en el primer concurso y a mi por problemas de aduana me llegó después y no me fue muy bien y bueno también lo que permiten son equipos de hasta tres personas y yo estaba solo y aquí en México no sabía dónde encontrar otras personas, y allí es justamente donde me vino la idea, dije: ojalá pueda difundirlo, pueda lograr que la gente lo conozca, igual el próximo concurso logro que seamos tres, igual hasta dos equipos y también me preocupa mucho cuando veo las noticias que dicen que la productividad en México está bajando cada año y a veces pienso que en la parte computacional no estamos utilizando las herramientas adecuadas, por ejemplo si se me enseña PHP, pues PHP tiene sus virtudes pero también sus debilidades, Java lo mismo y siempre hago una analogía con el transporte, en el transporte hay de todo, hay bicicletas, hay trailer, autobús, cada uno tiene su funcionalidad y para qué lo quiero, no voy a transportar dos toneladas con una bicicleta, tampoco voy a ir a comprar mi pan a la esquina con un trailer, lo mismo pasa con la computación pero yo creo que en la computación a veces nos equivocamos en el sentido de que Java a lo mejor es un trailer,

me permite hacer aplicaciones pesadas, pero si tengo la tienda en la esquina y quiero una aplicación rápida, una librería pues a lo mejor no necesito montar un Oracle, un Java, etcétera, es meter algo muy pesado, entonces allí hay que tener mejores herramientas, PHP venía como una alternativa, yo creo que Ruby on Rails puede ser una todavía mejor, donde fácilmente pueda encontrar un hosting, monto eso, en un día desarrollo la aplicación y listo y eso pues si aumenta la productividad y aumenta también el hecho de que yo pueda desarrollar una aplicación tu empresa que también está evolucionado, entonces también me vino esta inquietud de que se conozca, que se utilice esta herramienta en México, a mi me duele ver que se venden quinientos mil libros y tal vez en México se vendieron quinientos, entonces otros países están evolucionado mas rápido que nosotros, entonces por lo menos contribuyo en eso, de que la gente conozca esta herramienta, vea que es buena, y que trate de utilizarla, pero proyectos realmente no, no muy fuertes la verdad, a veces me he metido en código de otros proyectos existentes, he empezado a analizar cómo funcionan, he hecho algunas propuestas de parches, a veces aceptadas, a veces no. Ahorita me he dedicado mucho a promocionarlo.



RSL – A parte del desarrollo web, ¿a qué te dedicas?

FM -Durante varios años estaba a cargo de los servidores aquí en Tec de Monterrey, allí estuve aprendiendo bastante, de hecho cuando yo llegué casi todo estaba basado en Novell, entonces viví Netware 4, Netware 5, Netware 6 y también viví el

hecho de que cuando nació Windows 2000 no pudimos evitar tener servidores Windows 2000 por allí, dos formas de convivencia y siempre también hubo Linux por allí, incluso en su auge manejaba algo como 35 servidores, bueno aquí en Toluca, otros 15 creo que en Morelia, otros dos o tres allá en Atlacomulco, era algo bastante fuerte y allí aprendí bastante y me dí cuenta de que cada plataforma tiene sus buenos y sus malos, Linux no creo que sea la panacea, creo que es equivocado creerlo, si es un buen complemento que nos permite solucionar problemas que otros no solucionan. Ahora después de ocho años en el centro de cómputo dije: ya tengo nuevos conocimientos y experiencias, sería bueno pasarlos al aula, los alumnos muchas veces están a la espera de que alguien les dé una plática de cómo es el trabajo, no cómo está en el libro y para mí fue explorar otros aspectos del Tec, conocer otros departamentos, me pasé seis meses en la enseñanza, mis directivos no me aguantaron mucho tiempo y me invitaron a hacer un proyecto, tal vez lo vieron cuando en el messenger apareció maestrías en línea del Tec, bueno pues yo estuve a cargo de eso, allí descubrí justamente otros aspectos de la parte del web, CRMs, CRPs, por ejemplo tenemos un exámen de admisión, hicimos un sistema para hacer un exámen en línea antes para prepararlo y allí si es donde batallamos algo con PHP, después este proyecto se cerró, pase simplemente al nivel del campus Toluca igual haciendo mercadotecnia, me encargué del portal y nuevamente era mantener un servicio que tenía mas de 2 millones de accesos al mes y era algo no necesariamente sencillo y mas cuando traté de que fuera mas una comunidad que un portal donde los maestros pudieran de manera sencilla promocionar sus eventos, comentar lo que están haciendo. Bueno pues ya mejoramos esta parte, ahora incluso el Tec a nivel nacional todo su sistema es un sistema de portal y ya me regresé a la parte académica, al principio porque íbamos a ofrecer una nueva carrera aquí en el Tec que se llamaba ITIC (Ingeniero en Tecnologías de la Información y Comunicaciones), no necesariamente ingeniero en sistemas computacionales, entonces me gustaba esta parte porque era justamente lo que había vivido y yo creo que la sociedad otra vez no estaba muy lista, incluso yo siento que hay un desencanto hacia la computación ahorita, entonces cuando creamos la nueva carrera menos hubo respuesta y pues finalmente

decidimos limitar los daños, otra vez me regresé a la parte academica y ahora soy director de carrera en Tecnologías Computacionales aquí en el Tec de Monterrey y tratar de motivar a los alumnos porque hay mucho campo en la computación y desafortunadamente yo veo a muchos ahorita que están pensando que casi todo está hecho y si hay cosas por hacer no es para mí o es muy complicado, como que ven el reto demasiado grande y no lo es tanto, pero si uno realmente le entra se puede apasionar con esto, no es necesariamente estar encerrado en un cubículo como lo creen, o sea eso tal vez se deba a las películas de Hollywood donde pintan el nerd donde hay pizzas y café y es programador día y noche, yo creo ellos piensan que es así el trabajo y dicen: no me quiero ver así, entonces huyo de la computación; de hecho cuando he participado en los congresos, los ponentes que participamos tampoco ayudamos a quitar mucho esta idea, de repente les traigo una presentación y la actualicé ayer a las dos de la mañana y dicen: este está loco, entre nosotros decimos que es algo normal y nos gusta, pero cuando ellos que son externos y no están apasionados en esto dicen: están locos, empiezan a hablar en un lenguaje que para mí parece chino, ni le entiendo, o sea cuando nosotros que si estamos muy metidos tenemos una tendencia a perder la confianza de los chavos que le quieren entrar, incluso en el último congreso comentaba que cuando andamos en los congresos los damos a un nivel muy elevado y si, entre nosotros nos sirve, pero no pensamos en el chavo que ni siquiera sabe prender la compu y su esperanza es: si voy a este congreso chance pueda prender la compu pero con Linux y yo he visto a mucha gente en los congresos que dice: ya instalé Linux pero ahora ¿qué hago?, y nosotros estamos allí hablando de la última tecnología, entonces si creo que nos equivocamos en darle muy arriba y no pensar en la gente que le quiere entrar y hay mucha gente que le quiere entrar pero encuentra el hueco, entre lo que sabe y lo que nosotros los gurus manejamos, muy grande; entonces yo ahorita quiero disminuir la brecha, por ejemplo creo que Ruby on Rails puede ser una. Siempre he hecho el intento en los congresos de hacer la propuesta de explicar qué es software libre y yo entiendo siempre ha sido rechazada, dicen: no, eso es para los vagos, pues si pero a veces es un error que se niegue este tipo de presentación, hay que atender a todos los públicos, por ejemplo para mí un install fest no me gusta porque lo que acabamos por hacer es que nosotros instalamos ya quedó instalado y que te vaya bien, si yo alego que es sencillo pues mejor te enseño.



RSL – Me imagino que has tenido resistencia al proponer software libre frente a otras plataformas, ¿a qué es a lo que te has enfrentado?

FM -Bueno, es chistoso porque yo he vivido la transición, ahorita no creo que haya tanta resistencia, al inicio si era el ¿qué es esta bestia rara?, no era confiable, siempre estaba la idea de que lo gratuito no puede ser bueno y pues no había empresa de soporte, yo creo que muchas veces las empresas intervienen en el lado político, si yo introduzco software libre no hay empresa a la que pueda echarle la culpa cuando las cosas no jalan, si yo instalo Windows pues le echo la culpa a Bill Gates, es culpa de Microsoft y no puedo hacer nada y está cerrado y no puedo meter mano, que en software libre cuando uno lo instala no tiene este respaldo, cuando uno tiene una bronca te dicen: tu me dijiste que tienes el código fuente, métete y hazlo, eso a lo mejor si genera mucho temor a algunas personas que están en la empresa entonces si yo meto software libre yo voy a ser el responsable al 100% y si meto algunas aplicaciones propietarias pues entonces tenemos una responsabilidad compartida, yo elegí al proveedor pero el proveedor es el que va a desarrollar esto y él tiene la culpa y también yo siento que tengo una organización atrás de mí, mucha gente dice: cuando instalo Microsoft tengo a diez mil programadores atrás de mí, tal vez en la marcha se dan cuenta que eso no es cierto, se ha escuchado muchas cosas del soporte técnico de Microsoft, y en el software libre dicen: si el hacker se enferma y que no quiere hacer nada mañana y entonces que

ayuda tengo, no sentían que había compromiso, yo creo que el cambio de Novell es algo bastante importante porque en el mundo empresarial tiene mucho renombre, nadie cuestiona hoy en día la fiabilidad de los Netware y el hecho de que Novell ahorita promocione SUSE ha ayudado bastante en el mundo empresarial a que la gente considere Linux, si está esta empresa que tiene toda su red de partners, de distribuidores y todo lo que tiene y me ofrece Linux entonces la gente dice: hay un respaldo, también IBM se ha pronunciado pro-Linux, Dell, todavía hay dudas, HP también es pro-Linux y hasta Sun que te vende los servidores con Linux o con Solaris ya no es nada mas certificación en Solaris, también hay certificación de Linux de Sun y eso ha logrado crear confianza y eso ha ayudado a todo lo demás, como Linux está soportado y se asocia a todo tipo de software libre y sí se acepta su consideración, sin embargo yo he batallado bastante con el Office, para mi el OpenOffice es un cambio demasiado drástico y yo creo que hay una oportunidad allí porque el nuevo Office lo bajé en beta y cambiaron totalmente la presentación del Office de Microsoft y yo creo que va a desubicar a mucha gente, entonces si se desubica mejor que se desubique hacia el opensource y allí puede haber una oportunidad siempre y cuando el soporte del OpenOffice esté muy bien asentado porque hay deficiencias. Bueno por ejemplo yo en la Mac al contrario, utilizo software de Apple, no estoy con OpenOffice y la oferta de Apple no trae la hoja de cálculo y yo digo que debería de traerlo y estoy en la búsqueda de algo que lo haga porque en Excel como es emulado es algo lento y todo mundo me dice: pásate a OpenOffice y si lo tengo instalado pero a veces cuando abro ciertas páginas de Excel dentro de OpenOffice no me aparecen correctamente, incluso una vez me pasó que alguien me mandó un correo con un Powerpoint, lo abrí en OpenOffice y no veía nada y le mando a la persona: oye, me lo mandaste mal, me lo vuelve a enviar y otra vez no lo puedo abrir hasta que me paso a Windows y lo abre perfectamente, entonces si te pones a usar OpenOffice en una empresa tu eres como una isla y todo mundo alrededor de ti va a utilizar el Office y casi te obliga a utilizarlo y estas incompatibilidades que existen a veces si te perjudican, en la Mac igual no he logrado quitar el Excel, no uso ya Word ni Powerpoint, pero no he logrado quitar el Excel aún cuando hay alternativas libres, no lo logras quitar porque hay problemas allí, también por eso entre otras cosas la gente no te acepta tan fácilmente la migración y te voy a poner un ejemplo muy sencillo, yo logré que aquí el centro de cómputo aquí en el

instalara Linux completo, el Ubuntu, nadie lo arranca porque es diferente, la lap que compré ya venía con todo instalado, el Office me lo compré pirata, también algo de lo que perjudica aquí es que a todo mundo le vale que se esté pirateando, un Office cuesta 50 pesos, entonces es difícil de vender un Linux con el pretexto de decir: mira, es gratis, el Office casi lo tuve gratis, yo creo que por eso en países europeos el software libre crece mucho mas, porque hay mucha represión contra el software pirata, entonces allí es un software gratuito contra cinco años de cárcel, entonces ya lo pienso, que aquí hay una impunidad y todo mundo lo sabemos, te vas a la esquina y hay gente vendiendo películas piratas como también hay gente vendiendo software pirata y la gente dice: mira por 200 pesos equipo toda mi compu, tengo el Photoshop, el Office y tengo todo, para qué me molesto. Yo siempre he pensado que ojalá que Microsoft trabaje más contra la piratería porque ayudará al software libre.

RSL – Haz platicado un poco de tu experiencia con los usuarios finales, ¿qué crees que le falta al software libre para que pueda llegar a ese sector y no tengan que estarle preguntando a los gurus?

FM – Yo siempre que lo he mencionado, que lo he propuesto, siempre es la polémica porque yo creo que hay un orgullo en el software libre de tratar de hacer algo diferente a Windows, pero si la gente no nota el cambio de repente no sabe que está en Linux y de repente le dices al usuario un año después y no te lo cree, mientras eso suceda lo que yo he hecho es favorecer proyectos de software libre trabajando sobre Windows, yo creo que Linux da temor, es la imagen de que es el guru, son los nerds, son los geeks y esos cuates están bien locos y están fumados y yo no soy de este tipo de gente y entonces los ubico muy lejos y me dá temor y la línea de comando ni siquiera la de Windows sé manejar. Es chistoso ver que pones Firefox en Windows y la gente dice: ¿eso es software libre?, y de repente tu ves mucha gente utilizando Firefox alrededor de tí, yo creo que aquí un veinte por ciento lo prefieren a Internet Explorer pero no tienen todavía conciencia de que es software libre, yo creo que lo podemos lograr después, pero si tu empiezas a tener software libre bajo Windows de calidad, si tu llegas a que la gente lo único que utiliza en Windows es Windows, que todo lo demás es

software libre, entonces su paso a Linux es más sencillo. Y también algo que siempre he comentado y es algo equivocado que damos es que Linux es parte del software libre pero no todo el software libre es Linux y siempre queremos que la gente para utilizar software libre tenga Linux y yo me conformo con que la gente siga utilizando Windows pero que nada mas Windows, si la gente tiene un OpenOffice, un Gaim, un Gimp, un Firefox, yo creo que vamos de gane, o sea está utilizando puras aplicaciones libres, gratuitas, con su código fuente disponible, bajo Windows y va a poder incluso aprender a programar bajo Windows porque tiene acceso a su código fuente y a lo mejor se puede volver mas exigente hacia Windows porque aprendió a programar y va a decir: oye es que hay cosas que no veo que cuando analizo el código fuente en Linux si tiene, y entonces puede crear presión sobre el mismo Microsoft o desespcionarse y cambiarse a Linux, yo he visto por ejemplo el Ruby on Rails que algo que para mi es muy notable y ha hecho que nada del desarrollo de Ruby on Rails lo hago en Windows es que tiene un manejador de archivos muy lento y en Ruby on Rails por cualquier cosa se incorporan como unos 100 o 150 archivos, está muy dividido en muchos pequeños archivos y allí se siente la lentitud del sistema de archivos de Windows, te penaliza, cuando tu te pasas a la Mac o a Linux carga luego luego, entonces eso es bastante interesante porque teniéndolo en Windows tarde o temprano te llega la queja de que oye es muy lento, ¿es culpa de Ruby, verdad?, no, no lo es, mira instala Linux y vas a ver que si; entonces hubo una motivación para pasarse de Windows a Linux, empezar a descubrir a través del software libre sus deficiencias, y eso favorece a que digan: Ok, acepto el cambio.

RSL – Algo mas que quieras agregar

FM – Pues la verdad me ha dado mucho gusto este año me había puesto como propósito lograr participar un poco mas en el mundo del software libre, favorecerlo, decía: híjoles, tal vez soy de los primeros en el país en haberlo utilizado, yo empecé en otoño del 94 y era algo triste que no participara y pues algo tenía que hacer, lo había hecho en la práctica, el hecho de utilizarlo y de que por allí se mencionara que se utilizara en producción en una empresa como el Tec, bueno pues eso ayudaba pero pues ahorita lo retomé un poquito, ojalá se pueda colaborar mas todavía, yo creo que allí México tiene una oportunidad fuerte, tiene posibilidad de aprender cómo se hacen las mejores

prácticas de software, en Francia, en Alemania, en todos ellos, y ojalá que algún día tengamos una distribución mexicana tan fuerte como Red Hat, SUSE o Mandriva, cada una de ellas representa para todos claramente un país, Ruby es Japón, pues ¿qué nos toca a México?, hay que buscar la manera en que tal vez dejemos de hacer todos los del software libre aquí en México esfuerzos aislados y tal vez empezar a hacer un poquito mas de ruido y que México sea considerado también como una opción en los demás países, si alguien se quiere lanzar yo estoy listo a participar en la distribución mexicana de Linux, con la parte de Ruby on Rails obviamente o en la parte de web, Apache, PHP, integrarlos correctamente, de hecho la tesis que yo hice de maestría era un comparativo de plataformas de web con una metodología científica, la idea original era que si uno lee de internet cada uno dice que es lo mejor, Microsoft decía que era lo mejor, Java decía que era lo mejor, PHP decía que era lo mejor, pero bueno si uno aplica un método científico, pues allí debería salir qué es lo mejor en ciertas condiciones, lo que no me gustó al final es que finalmente ASP .Net y

ADO .Net si era lo mejor que PHP, Perl y todo eso, pero también lo que hice es que una de las condiciones en las cuales se hizo el estudio es que yo utilizaba la versión como estaba preinstalada y yo creo que la gente se ubica en hacer distribuciones libres hacia el escritorio, no hay una distribución gratuita muy enfocada a servidor o por lo menos no la conozco, y con parámetros muy bien optimizados que casi que dices: meto el CD, lo instalo y corre mejor que Windows, y lograr todo lo que hice en la maestría, porque después parametrizando se lograba que PHP tuviera un aspecto similar, si es una diferencia de 5 milisegundos a 5.1, ya no es significativo pero había que configurar, cuando en Windows ya venía preconfigurado todo esto, entonces creo que hay oportunidad allí, de hecho yo lo viví cuando estaba en el centro de cómputo, tu instalas un servidor pero por lo general están pensados para el escritorio, tienes que meterte para que sea servidor, tal vez allí haya oportunidad de hacer algo interesante. 🤖

Mas información del trabajo de Fabien:

www.rubylandia.com
www.odilat.com/news



Calendario SL

La agenda de los Geeks

Conferencia Rails Hispana

iNext promoviendo las tecnologías de desarrollo de software Libre y Abierto en México, se complace en invitar a los desarrolladores mexicanos de “Ruby on Rails” y en general a la comunidad mexicana de desarrollo de software libre y abierto, a la “Conferencia Rails Hispana” la cual se llevará al cabo los días: 24 y 25 de Noviembre en Madrid España y la cual iNext hará accesible en tiempo real en la ciudad de México D.F. al ofrecer sus instalaciones e infraestructura tecnológica para que los asistentes a la misma puedan participar en ella de manera interactiva a través de la infraestructura de videoconferencia y ancho de banda con los que cuenta en sus instalaciones.

La cuota de recuperación será de: \$300.00 pesos por persona.

Estos deberán ser pagados días previos a la conferencia
informes en la sección de registro en: <http://conferenciarails.org>

Los participantes podrán traer su laptop al evento se ofrecerá acceso inalámbrico gratuito a internet en el salón de la conferencia, durante las sesiones de la misma, así como la posibilidad de participar en ella via videoconferencia en tiempo real.

El cupo es limitado y se les recomienda inscribirse lo más rápido posible para poder garantizar su asistencia al evento.

Email Informes:
Victor Rosillo
fractal@fractalgrafics.com.mx

Ruby On Rails en la Ciudad de México

Acude a la reunión para conformar el Grupo de Usuarios de Ruby On Rails en México. Además aprende sobre esta creciente tecnología... se parte del comienzo.

Ivan Vega
<http://entren.ivanyvenian.com>
<http://oc.hemac.com.mx:81/cc/default.asp?ror-mexico>



RevistaSL

El software libre hecho revista

Colaboradores SL

Adib Saavedra Bocanegra
Martin Marquez
Longino Jacome
Sonia Sánchez
Ivan González Zamora
Sara Mostajo Rojas
Quetzatcoatl Pantoja Hinojosa
Leo Utskot
Fabien Manniser

Patrocinadores SL



~# Bytez SL



Una nueva edición de RevistaSL a visto la luz. Al interior del equipo de edición creemos que hemos madurado el proyecto, sin embargo estamos concientes de la existencia de varios detalles que pueden mejorarse. Esos detalles esperamos que vayan desapareciendo con la experiencia que número a número el equipo editorial vaya ganado, y por supuesto siempre tenemos abiertas las puertas a toda ayuda, colaboración u opinión que nos deseen hacer llegar.

No me queda mas, que en nombre de todo el equipo editorial, agradecer sus colaboraciones, comentarios, felicitaciones, visitas, lecturas y descargas. Desde el Número 0 de RevistaSL ustedes nos han apoyado, y siempre trabajamos pensando en ustedes.

Y no me quiero despedir sin antes felicitar a todo el equipo editorial, todas y cada unas de las personas que ponen su granito de arena para que este proyecto bimestre con bimestre reuna las ideas de muchos y las muestre al mundo entero.

Gracias a todos!!

Carlos Augusto Lozano Vargas
carlos.augusto@revista-sl.org