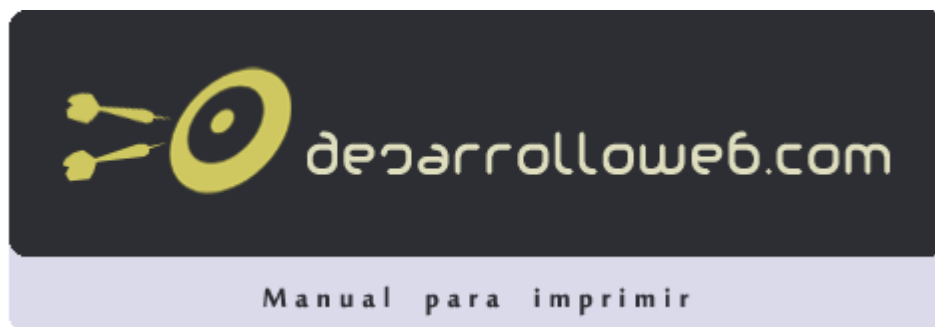


Taller de PHP



Autores del manual

Este manual ha sido realizado por los siguientes colaboradores de DesarrolloWeb.com:

Miguel Angel Alvarez
Director de DesarrolloWeb.com
<http://www.desarrolloweb.com>
(53 capítulos)

Rubén Alvarez
Redactor de DesarrolloWeb.com
<http://www.desarrolloweb.com>
(4 capítulos)

Eloi de San Martin Lagranje
<http://www.programacionweb.net>
(2 capítulos)

Serafin Vazquez
<http://www.phpisam.com>
(2 capítulos)

Carlos Luis Cuenca
<http://www.helloworldsolutions.com/>
(2 capítulos)

Antonio Melé
Estudiante de informática
<http://www.ciberseleccion.com>
(1 capítulo)

Angel Deblás

Emmanuel García De Caro
Webmaster de Blasten.com
<http://www.blasten.com>
(13 capítulos)

Alfredo Anaya
Director Técnico de ADR Formacion
<http://www.adrformacion.com>
(3 capítulos)

Roberto Bárcenas
<http://www.chicosyescritores.org>
(2 capítulos)

Pablo González
<http://www.xski.net>
(2 capítulos)

Raúl Avilés
<http://www.isens.es/>
(2 capítulos)

Gema Maria Molina Prados
Equipo DesarrolloWeb.com
<http://www.desarrolloweb.com/>
(1 capítulo)

Juliana Monteiro Lazaro

Andrés Fernández
<http://www.disegnocentell.com.ar>
(4 capítulos)

Miguelo
Desarrollador freelance
<http://www.keleke.com>
(3 capítulos)

Juan Edgardo Jorquera Uribe
Analista Programador
<http://www.aplicacionesweb.cl>
(2 capítulos)

Jorge De Freitas
<http://www.jjcreaciones.com>
(2 capítulos)

Sergio Rodriguez Rios
<http://www.guaser.info/ipsrr>
(1 capítulo)

Daniel Guajardo
(1 capítulo)

Luis Vega M.

<http://www.demasiadopersonal.com>
(1 capítulo)

Josue Gutiérrez Olivares
(1 capítulo)

Jorge Jesús Churruca shock dude
<http://www.nixlink.com.ar/>
(1 capítulo)

Agustin Jareño.
<http://www.levanteweb.com/>
(1 capítulo)

Eugenia Bahit
Desarrolladora ASP y PHP
<http://www.cmzk.com.ar>
(1 capítulo)

Manu Gutierrez
<http://www.tufuncion.com>
(1 capítulo)

José Lorenzo Escudero
Traductor y desarrollador de CriarWeb.com
<http://www.criarweb.com>
(1 capítulo)

Sunflower
<http://www.sunflower.es>
(1 capítulo)

Directora de CriarWeb.com
<http://www.criarweb.com>
(1 capítulo)

Diego Pinilla
Responsable relaciones exteriores de MercadoProfesional.com
<http://www.mercadoprofesional.com>
(1 capítulo)

Jorge Pinedo
Creador de Paginator
<http://nuestrophp.tk>
(1 capítulo)

José Valle
(1 capítulo)

Francisco J. Matias
<http://www.pclandia.com>
(1 capítulo)

William Wong Garay
Profesional en Computación e Informática. Especialista en desarrollo web.
<http://billysite.net>
(1 capítulo)

irv.
<http://www.ignside.net/>
(1 capítulo)

Raúl Cano
<http://infoapuntes.com>
(1 capítulo)

Webmaster
<http://fodsitewebcindario.com/>
(1 capítulo)

Alejandro Suazo
contador auditor
<http://triple.chileasesores.com>
(1 capítulo)

Héctor A. Pinto F
Ingeniero en Informática. Santiago de Chile.
(1 capítulo)

Mario Juárez
Estudiante
<http://mkarra.zepsilon.com/>
(1 capítulo)

Aaron Gomez Perez
<http://www.elwey.com/>
(1 capítulo)

Raúl Jiménez Ortega
Desarrollador web
<http://www.quickwebdeluxe.com>
(1 capítulo)

Diego Villar
(1 capítulo)

José Pino
(1 capítulo)

Introducción al taller de PHP

En este manual vamos a desarrollar diversas técnicas que se utilizan en PHP para realizar procesos un poco más complejos, los artículos están tratados con un enfoque práctico y son independientes unos de otros. Para leerlos con posibilidad de asimilar los conceptos y técnicas de programación es necesario tener unos conocimientos previos sobre el lenguaje. Por ello, si aun no has aprendido PHP es aconsejable que empieces por el [Manual de programación en PHP](#) antes de dedicar tus esfuerzos a este Taller de PHP.

De todos modos, antes de entrar con los capítulos prácticos vamos a ver una brevísima introducción al lenguaje

Qué es PHP

PHP Es un lenguaje de programación de páginas web del lado del servidor cuyas características principales son la independencia de plataforma y su gratuidad.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para

crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP.

PHP es uno de los lenguajes que sirven para la programación de scripts del lado del servidor, otros lenguajes muy utilizados son ASP o JSP, que tienen características similares.

Para poder programar en PHP se requiere de un servidor preparado para ello. Como el lenguaje de programación es multiplataforma, cualquiera de los principales servidores web nos servirán para ello. Lo único que tenemos que hacer para preparar nuestro servidor para entender el PHP es descargar de la página de inicio de la tecnología, www.php.net, el módulo específico para el sistema que estamos utilizando. Una vez descargado tendremos que instalarlo en la computadora, siguiendo las instrucciones que podremos encontrar en la misma página. A partir de ese momento nuestro servidor web podrá entender las páginas PHP y nos podremos lanzar a programar en este potente lenguaje.

Las personas que tienen un sistema Windows 98 también pueden instalar un servidor web y prepararlo para PHP. El servidor web más accesible para estos sistemas es el Personal Web Server, que se encuentra en el mismo CD de windows 98, en el directorio AddOns. Desde la página de PHP se puede encontrar el modulito que permitirá al Personal Web Server, entender las páginas PHP.

Encontrar un proveedor que soporte PHP, para publicar las páginas en Internet es relativamente fácil y barato. PHP está ampliamente difundido entre los servidores Linux o Unix, que no permiten la programación con ASP, lo que hace de PHP el complemento ideal para que sus servidores puedan realizar la programación de páginas con scripts de servidor. Podemos encontrar algunos [sitios gratuitos donde permiten programación en PHP](#). Si deseamos un servicio más profesional y con soporte a mejores recursos y funcionalidades, sería interesante que contratásemos un espacio de pago. En DesarrolloWeb.com [ofrecemos servicio de alojamiento](#) en asociación con la mejor empresa española del sector.

*Artículo por **Miguel Angel Alvarez***

Tutorial de sesiones en PHP

Veamos un sencillo ejemplo de uso de sesiones para ilustrar a los usuarios más inexpertos el manejo de sesiones con PHP en sus procesos más básicos, como son la inicialización de sesiones, comprobar si existen variables de sesión, crearlas y modificarlas.

En este ejemplo vamos a hacer un sistema para llevar la cuenta de las páginas que ha visto un visitante. Es decir, si entra a una página del sitio por primera vez, contamos que ha visto una página, luego con cada página adicional que visite, seguiremos acumulando 1 en la cuenta de páginas vistas dentro del sitio. Del mismo modo, si se recarga una página o se vuelve a la misma página que ya se estuvo, también haremos que se acumule 1 en su cuenta.

Para aprender a trabajar con sesiones recomendamos leer el artículo [sesiones en PHP](#).

Lo que tendremos que hacer en este ejercicio es lo siguiente:

- Inicializar la sesión
- Si no tenemos el contador de páginas creado, lo inicializamos al valor 1
- Si tenemos el contador de páginas creado, lo incrementamos en 1.

```
<? session_start();  
if (!isset($_SESSION["cuenta_paginas"])){
```

```
$_SESSION["cuenta_paginas"] = 1;
}else{
    $_SESSION["cuenta_paginas"]++;
}
?>
<html>
<head>
<title>Contar páginas vistas por un usuario en toda su sesión</title>
</head>

<body>
<?
echo "Desde que entraste has visto " . $_SESSION["cuenta_paginas"] . " páginas";
?>
<br>
<br>
<a href="otracuenta.php">Ver otra página</a>
</body>
</html>
```

Como se ha podido ver, lo primero que debemos hacer es inicializar la sesión con `session_start()`.

Luego de inicializar la sesión podemos trabajar con variables de sesión en cualquier lugar del código PHP. Nosotros vamos a crear y utilizar una variable de sesión para llevar la cuenta del número de páginas que hemos visto.

```
$_SESSION["cuenta_paginas"]
```

Para ver si una variable de sesión ha sido creada podemos utilizar la función `isset()` pasándole como parámetro la variable que deseamos saber si ha sido creada. Si no fue creada anteriormente, simplemente la creamos (inicializando su valor a 1). Si fue creada, pues que tenemos que incrementar en 1. Eso se consigue con este código:

```
if (!isset($_SESSION["cuenta_paginas"])){
    $_SESSION["cuenta_paginas"] = 1;
}else{
    $_SESSION["cuenta_paginas"]++;
}
```

Eso es todo! Ahora lo que se supone que tenemos son otras páginas en el mismo sitio que también tienen que incrementar en 1 el contador de páginas vistas por ese usuario en esa sesión.

Eso lo podríamos hacer con el mismo trozo de código, como el que hemos visto antes, colocado al principio del archivo PHP.

```
<? session_start();
if (!isset($_SESSION["cuenta_paginas"])){
    $_SESSION["cuenta_paginas"] = 1;
}else{
    $_SESSION["cuenta_paginas"]++;
}
?>
```

Luego, en cualquier momento, si deseamos mostrar el número de páginas vistas en un momento dado, tendríamos que hacer algo como esto:

```
<? echo "Has visto " . $_SESSION["cuenta_paginas"] . " páginas, contando actualizaciones de paginas"; ?>
```

Podemos [ver el ejemplo relatado en funcionamiento](#).

*Artículo por **Miguel Angel Alvarez***

Acceso a base de datos con PHPLIB

El acceso a base de datos es una de las tareas más comunes, en cualquier aplicación sobre Internet. Y una de las primeras que cualquier programador trata de automatizar. Y como casi todo en esta vida se les ha ocurrido antes a algunos miles, y en algunos casos lo han resuelto francamente bien.

Como muestra, las clases de base de datos que incluye <http://phplib.netuse.de/> PHPLIB entre otras, como las que incorpora para la gestión de sesiones, autenticación, etc.

Si vienes de la programación estructurada, probablemente las clases y la programación orientada a objetos, OOP, te resulten un tanto intimidatorias. Pero francamente el uso de las clases aporta tantas ventajas que merece la pena perder un poco de tiempo en comprender su funcionamiento.

Procuraré no hacer uso de la terminología que se utiliza en la OOP, aunque algunas veces será inevitable. En estos casos pasaré de puntillas sobre estos temas, ya que no es finalidad de este tutorial introducirte en el mundo de los objetos.

Para el uso de DB_Sql, solo tienes que <http://phplib.netuse.de/download/index.php3> descargarte la librería de clases y descomprimirlas. No te asustes entre todos esos ficheros, solo nos interesan unos pocos, en concreto los que empiezan por db_. Como podrás ver PHPLIB incluye clases para manejar las bases de datos más populares, entre las que se cuentan las de MySQL, en la que nos centraremos, Oracle y Postgress entre otras.

Para utilizar una clase lo primero que hacemos es crear una instancia de la misma y para ello debemos informar a PHP donde encontrar la clase. Lo haremos de la siguiente forma:

```
include ('/ruta_al_fichero/db_mysql.inc');  
$q= new DB_Sql;
```

Como puedes ver para ello hemos utilizado la palabra new y lo hemos asignado, a una variable. En realidad esto es un objeto, una instancia de la clase DB_Sql.

Ahora inicializaremos los parámetros de conexión a la base de datos, modificando algunas variables definidas dentro de la clase:

```
$q->Host = "tuHost";  
$q->Database = "tuBaseDeDatos";  
$q->User = "tuUsuario";  
$q->Password = "tuPassword";
```

Por supuesto, puedes asignar estos parámetros dentro de la clase a las correspondientes variables, de esta forma los datos de conexión siempre serán los mismos.

Hasta el momento deberías tener algo como esto:

```
<?php  
include ('/ruta_al_fichero/db_mysql.inc');  
$q= new DB_Sql;  
  
$q->Host = "tuHost";  
$q->Database = "tuBaseDeDatos";  
$q->User = "tuUsuario";  
$q->Password = "tuPassword";
```

?>

Y llego el momento de hacer algo práctico con nuestra clase. Supongamos una base de datos con la siguiente estructura:

```
create table amigos (  
id int(11) default '0' not null auto_increment,  
nombre varchar(25),  
apellidos varchar(25),  
dirección varchar(50),  
ciudad varchar(50),  
pais varchar(50),  
primary key (id)  
);
```

Vamos a realizar un query a la base de datos, usando la clase. La construcción del query se hace de la misma forma que si atacamos directamente a la base de datos, con las funciones de PHP:

```
$query='select * from amigos';  
$q->query($query);
```

Con esto, si todo fué bien, habremos conseguido los registros que cumplen la condición de la tabla amigos. Pero, ¿como podemos asegurarnos de que hemos obtenido algún registro?. Bien, por un lado deberemos asegurarnos de que la conexión con la base de datos se ha establecido. DB_sql establece una conexión persistente al efectuar la consulta, por lo tanto si query devuelve algún valor, sabremos que la conexión se ha establecido. Para asegurarnos de que la consulta ha sido exitosa, deberemos tener en cuenta que algunas veces una consulta puede ser verdadera, pero no devolver ningún resultado, caso muy común al realizar un select. Es por lo tanto una buena práctica verificar este punto, así es como quedaría nuestro código:

```
$query='select * from amigos';  
if (!$q->query($query)){  
echo 'Lo siento no se pudo establecer la conexión<br>';  
}else{  
if (!$q->num_rows()){  
echo 'Lo siento no se ha obtenido ningún resultado<br>';  
}else{  
echo 'Hey! hay '.$q->num_rows().' registros que cumplen la condición';  
}  
}
```

Esto esta muy bien, veamos ahora como manejar los registros y campos de forma individual. Para ello recorreremos los resultados e iremos mostrando los datos que nos interesen uno a uno:

```
$query='select * from amigos';  
if (!$q->query($query)){  
echo 'Lo siento no se pudo establecer la conexión<br>';  
}else{  
if (!$q->num_rows()){  
echo 'Lo siento no se ha obtenido ningún resultado<br>';  
}else{  
echo '<pre>';  
echo 'Hey! hay '.$q->num_rows().' registros que cumplen la condición';  
while ($q->next_record()!=0){
```

```
echo 'El id es: '.$q->f('id').'\n';
echo $q->f('nombre').' '.$q->f('apellidos').'\n';
}
echo '</pre>';
}
}
```

De la misma forma que hacemos un select, haremos un insert, update o delete. Obviamente estos tres últimos no devuelven resultados, por lo tanto para saber si nuestros queries han funcionado recurriremos a la función `affected_rows` de la clase `DB_Sql`, de esta forma:

```
$query='delete from amigos where ciudad=\'Medellín\'';
if (!$q->query($query)){
echo 'Lo siento no se pudo establecer la conexión<br>';
}else{
if (!$q->affected_rows()){
echo 'Lo siento no se encontro ningún registro que cumpla la condición<br>';
}else{
echo 'Se han eliminado '.$q->affected_rows().' registros';
}
}
```

Como puedes comprobar, el uso de una clase no es nada complicado. Solo debes preocuparte por conocer los métodos, las funciones de la clase que tienes disponibles y lo que hacen. Habitualmente las clases están suficientemente documentadas y te muestran los valores de entrada y salida requeridos. Además es una buena forma de mejorar tu estilo de programación y adquirir nuevos conocimientos.

El uso básico de las clases que incorpora PHPLIB para otras bases de datos es básicamente igual, puede que más o menos funciones, aunque las básicas son iguales en todos los casos. Es importante que revises la documentación, para comprobar las posibles diferencias.

Ten en cuenta que cualquier llamada del tipo `$q->loquesea()`, es una llamada al método de una clase y que algo del tipo `$q->otracosa` es una variable de la clase. Con paréntesis, un método, como una función normal de PHP y sin ellos una variable. Dentro de la clase la llamada a las funciones y variables se hace con la palabra reservada `this`. Por lo tanto cuando te encuentres un `$this->algo`, fíjate si tiene o no paréntesis a continuación, y así sabrás si es una llamada a un método de la clase o alguna operación con variables.

Y esto es todo de momento, espero que te sirva de ayuda. Hasta pronto.

Artículo por [José Valle](#)

Verificar la existencia de una URL

Ya hemos visto las [funciones de gestión de archivos](#) más comúnmente utilizadas en PHP. Sirvámonos de la más clásica de todas ellas, *fopen*, para crear un script que verifique la existencia de una URL.

Este tipo de script puede ser utilizado para múltiples propósitos: Detectar si los enlaces están rotos, verificar una etapa de inscripción en un formulario...

En este caso, hemos simplificado al máximo su contenido de manera a poner evidencia su funcionamiento. Otro tipo de mejoras tales como la verificación de la extensión del archivo (.asp, .php, ...) o del protocolo de transferencia (http, ftp...) pueden ser introducidas afín de personalizar su uso para distintas aplicaciones.

A continuación podéis ver como quedaría el script:

```
<?
function verificar_url($url)
{
    //abrimos el archivo en lectura
    $id = @fopen($url,"r");
    //hacemos las comprobaciones
    if ($id) $abierto = true;
    else $abierto = false;
    //devolvemos el valor
    return $abierto;
    //cerramos el archivo
    fclose($id);
}
?>
<html>
<head>
    <title>Verificacion de URL</title>
</head>
<body>
<?
if (!isset($url))
{
?>
    <form action="enlace.php" method="post">
        Indica tu URL: <br>
        <input type="Text" size="25" maxlength="100" name="url"
value="http://">
        <input type="Submit" value="Verificar!"
        </form>
<?
}
else
{
    $abierto = verificar_url($url);
    if ($abierto) echo "La URL existe!";
    else echo "La URL no existe o es inaccesible...";
}
?>
</body>
</html>
```

[Ejecutar ejemplo](#)

Hemos introducido en el mismo script que se encarga de verificar la URL el formulario que se

encarga de recogerla. Así, podemos dividir el script en dos partes: Una primera que se encarga de recoger la URL en un campo texto y una segunda que es la que verdaderamente evalúa la existencia de la URL.

Para hacer más aplicable el script hemos dejado la evaluación propiamente dicha en forma de función que podréis copiar y pegar en vuestra librería particular. Como podéis ver, el modo de operar es extremadamente sencillo:

- Abrimos el archivo remoto por medio de la función *fopen* en modo solo lectura. A notar que precediendo a la función *fopen* hemos introducido un símbolo arroba @ cuyo cometido es el de ignorar el posible error producido por la sentencia. Esto nos evita ver el mensaje de error que es mostrado cuando la URL no existe.
- Verificamos que el identificador de apertura *\$id* no está vacío. Es en este punto donde podemos implementar a nuestra función las mejoras de las que hemos hablado, las cuales le confieren la verdadera utilidad.
- Devolvemos un valor *true* o *false* dependiendo del éxito de la conexión.

Como podéis ver el script no reviste ninguna dificultad y puede ser muy práctico.

Artículo por **Rubén Alvarez**

Usuarios activos con PHP

En nuestro manual de PHP abordamos en su momento el [uso de sesiones](#) y dimos algún ejemplo práctico en el que este tipo de variables pueden ser utilizadas para dar a nuestro sitio un aspecto más dinámico.

Muchos de vosotros habéis podido ver en ciertos sitios un contador de usuarios que se encuentran en ese momento navegando por las mismas páginas que vosotros. Si os habéis fijado bien, habréis podido observar que, para la mayoría de los casos (sino la totalidad), el sitio en cuestión está programado con ASP como lenguaje de servidor.

Efectivamente, ASP permite una gestión más accesible de las sesiones por medio del famoso archivo [global.asa](#). Esto no quiere decir sin embargo, que PHP es incapaz de realizar el mismo tipo de tareas sino que, más bien, hemos de trabajar un poco más para conseguirlas. En efecto, todos los lenguajes tienen sus pros y sus contras y, en este caso particular, ASP resulta más versátil aunque hay que admitir que, con su versión 4, PHP ha mejorado mucho en todo lo que respecta al tratamiento de sesiones.

Aquí os mostramos una forma sencilla de contabilizar los usuarios activos de vuestro sitio usando para ese propósito una tabla. Dentro de dicha tabla, iremos almacenando los distintos números IP de los visitantes de nuestro sitio y la hora y fecha en la que el visitante ha ejecutado por última vez el script.

Asimismo, la tabla ha de ir borrando progresivamente las sesiones que no hayan sido renovadas en un tiempo que nosotros consideremos límite. Nosotros hemos fijado dicho límite en 24 minutos que es el tiempo máximo tomado por defecto por PHP para suprimir los datos

de una sesión. Para modificar este tiempo de vida máxima de una sesión puede hacerse en el *php.ini* a partir del parámetro *session.gc_maxlifetime* donde expresaremos dicho plazo en segundos. Ojo, este tiempo máximo es restaurado a su valor inicial cada vez que el usuario realiza una petición al servidor, esto quiere decir que un visitante podrá navegar cuanto tiempo quiera por el sitio guardando la misma sesión siempre y cuando no se quede más de 24 minutos sin realizar ningún tipo de acción.

Para el correcto funcionamiento del script, es necesario antes de nada crear una tabla en nuestra base de datos. Esta sentencia SQL puede ayudaros en la tarea:

```
CREATE TABLE control_ip (  
ip VARCHAR(15) NOT NULL,  
fecha INT(14) UNSIGNED NOT  
NULL,  
INDEX (ip)  
);
```

Como veis, el campo *ip*, que almacena el número IP del visitante, está indexado. Esto nos permitirá una selección rápida. En contrapartida, como todo campo indexado, su tamaño en memoria será doblado lo cual no tiene mucha importancia ya que la tabla tendrá un número de registros bastante limitado. Lo importante en efecto es que el script se ejecute rápidamente sin consumir demasiados recursos del servidor, sobretodo teniendo en cuenta que se trata de un código que será sistemáticamente ejecutado en cada una de las páginas del sitio.

Pasemos a continuación a mostrar el script que utilizaremos:

```
<?  
////////////////////////////////////  
//USUARIOS ACTIVOS  
//Calcula el numero de usuarios activos  
////////////////////////////////////  
  
function usuarios_activos()  
{  
    //permitimos el uso de la variable portadora del numero ip en nuestra funcion  
    global $REMOTE_ADDR;  
  
    //asignamos un nombre memotecnico a la variable  
    $ip = $REMOTE_ADDR;  
    //definimos el momento actual  
    $ahora = time();  
  
    //conectamos a la base de datos  
    //Usad vuestros propios parametros!!  
    $conn = mysql_connect($host,$user,$password);  
    mysql_select_db($db,$conn);  
  
    //actualizamos la tabla  
    //borrando los registros de las ip inactivas (24 minutos)  
    $limite = $ahora-24*60;  
    $ssql = "delete from control_ip where fecha < ".$limite;  
    mysql_query($ssql);
```

```
//miramos si el ip del visitante existe en nuestra tabla
$sql = "select ip, fecha from control_ip where ip = '$ip'";
$result = mysql_query($sql);

//si existe actualizamos el campo fecha
if (mysql_num_rows($result) != 0) $sql = "update control_ip set fecha = ".$ahora." where ip = '$ip'";
//si no existe insertamos el registro correspondiente a la nueva sesion
else $sql = "insert into control_ip (ip, fecha) values ('$ip', $ahora)";

//ejecutamos la sentencia sql
mysql_query($sql);

//calculamos el numero de sesiones
$sql = "select ip from control_ip";
$result = mysql_query($sql);
$usuarios = mysql_num_rows($result);

//liberamos memoria
mysql_free_result($result);

//devolvemos el resultado
return $usuarios;
}
?>
```

Podéis observar, como viene siendo norma, que el script es expresado en forma de función. Después de definir la IP y el momento en el que el script está siendo ejecutado, pasamos a interactuar con la base de datos. Dentro de este proceso, podemos distinguir distintas fases:

- Conexión con la base de datos
- Barrido de la tabla para eliminar los registros obsoletos.
- Inserción o actualización de un registro dependiendo de si el visitante es nuevo o no.
- Cómputo del número de registros de la tabla

La función finaliza liberando el espacio de memoria utilizado en sus consultas y enviando el resultado del cálculo efectuado.

En líneas generales el script es de fácil comprensión. Puede que alguna de las funciones empleadas os sea desconocida. Si es así, acudid al la [pagina oficial de PHP](#) donde podréis encontrar detalles en cuanto a su empleo.

Para sacar el valor proporcionado por la función a nuestro script principal tendremos que realizar una llamada clásica del tipo:

```
$active_users = usuarios_activos();
```

He aquí en definitiva un script sencillo que puede dar a vuestro sitio una imagen un poco más dinámica. Además, podéis utilizarlo y mejorarlo para crear vuestro propio sistema de estadísticas internas. Eso os lo dejamos a vosotros...

Artículo por **Rubén Alvarez**

Taller de paso de variables por URL

Este taller resultará un poco básico para algunas personas, ya que no vamos a explicar nada exclusivamente nuevo y que no hayamos comentado ya en otros manuales. Sin embargo, seguro que muchas otras personas agradecerán esta explicación para contestarse algunas dudas a la hora de hacer páginas avanzadas con ASP o PHP.

Resulta que me han llegado algunas dudas a mi correo de personas que no comprendían muy bien la manera de realizar páginas que, dependiendo de la variable que reciban por la URL presenten una información u otra. A continuación podemos ver estas dudas para aclararnos.

Duda 1: (Sobre el lenguaje ASP, aunque da igual, ya que la metodología de uno y otro lenguaje son las mismas. Sólo difiere la sintaxis.)

Hola me llamo Orel y necesito si me pueden dar una ayudita con mi pagina web.
Bueno primero que nada estoy creando mi pagina en asp y no entiendo cómo poner un enlace por ejemplo:

Cuando colocan varios enlaces y todos llevan a la misma pagina ver.asp lo que cambia es ?
Id=emuladores o ?Id=Roms etc.

No se cómo hacer eso.

Por que si yo pongo la pagina ver.asp?Id=emuladores Me aparece la pagina de emuladores mientras que si pongo ver.asp solo me muestra la pagina de inicio ¿Como hacen eso de que con la misma pagina puedan crear varias?

Duda 2: (Esta duda es de PHP, pero da igual el lenguaje para el que estemos trabajando, en el fondo es la misma duda)

Como se que vosotros sabéis mucho de php, os quería comentar una duda que tengo. ¿Que necesito para crear una web de carátulas en la cual las carátulas salgan en una página que sea una especie de "plantilla", en la cual solo cambie la carátula? Éste tipo de sistemas los he visto en webs de carátulas y son de la forma:

<http://nombredelaweb/caratula.php?id=587>

<http://nombredelaweb/caratula.php?id=351>

También cambia el texto y otras cosas.

Respuestas

Pues bien, planteadas las dudas, primero decir que en DesarrolloWeb tenemos todos los contenidos necesarios para aprender técnicamente a utilizar estas variables que se pasan por la URL. Es importante que conozcamos la sintaxis para poder hacer los ejercicios donde dudamos.

Aquí tenemos los enlaces:

- Para [aprender a pasar y recoger variables por la URL en ASP](#)
- [Lo mismo pero en PHP](#).

Una vez aprendida la parte técnica podemos pensar en cómo hacer estos sistemas en ambos lenguajes. Lo primero que debemos saber es que esas variables que recibimos por parámetro las podemos utilizar para cualquier cosa. En la duda 1 las vamos a utilizar para mostrar una información de una sección y en la duda 2 las vamos a utilizar para extraer la información de la carátula que se desea ver. Creo que lo mejor sería ver cada uno de los dos casos, en el lenguaje que se preguntan.

Respuesta de la Duda 1, para ASP

Necesitamos extraer el valor de la variable que nos pasan por parámetro y hacer unas cosas u otras en función de dicho valor. Veamos a grandes rasgos el código que utilizaría.

```
<%  
'extraigo el valor de la sección que se desea ver  
seccion_ver = request.QueryString("id")  
  
'Dependiendo de la sección muestro unos u otros contenidos  
if seccion_ver = "emuladores" then  
'entonces coloco todo lo que queramos mostrar en la sección emuladores  
%>  
<h1>Sección de emuladores</h1>  
<p>Bienvenidos a mi sección de emuladores, donde puedes conocer todos los principales  
emuladores del mercado y descargar los programas.  
<p>... (En fin, todo lo que queramos escribir en esta sección)  
<%  
elseif seccion_ver = "Roms" then  
'entonces coloco todo lo que queramos en la sección de Roms  
%>  
<h1>Sección de Roms</h1>  
<p>Bienvenidos a mi sección de Roms, tenemos más de 500 roms de juegos de todos los  
tiempos. Cada uno con enlace para descarga e instrucciones de uso.  
<p>... (En fin, todo lo que queramos escribir en esta sección)  
<%  
else  
'paso por aquí si no era la sección emuladores ni Roms  
'entonces hago lo que desee, que en la duda era simplemente mostrar la página principal.  
Response.redirect("index.asp")  
'esta instrucción te redirecciona a la página principal de tu sitio web, si es que se llamaba  
index.asp y está en el mismo directorio que la página actual.  
end if  
%>
```

Como se puede comprobar, dependiendo de lo que se reciba por parámetro se mostrará una sección u otra. Si no se recibe nada por parámetro o el valor no corresponde a ninguna de las secciones que hemos definido, se redirecciona a la página `index.asp`, que suponemos que es la página principal del sitio web.

Respuesta de la Duda 2, para PHP

Este ejemplo es, si cabe, más simple que el primero. La forma de trabajar típica que se realiza con un ejemplo como este es que todas las carátulas estén en una tabla de una base de datos. Entonces, cada vez que se accede a la página `caratula.php` se recibe el identificador de la carátula que se quiere visualizar en una variable en la URL, de esta manera: `caratula.php?id=12`. Así estamos accediendo la página que muestra carátulas y le decimos que la carátula que deseamos ver es la que tiene el identificador (id) igual a 12.

Supongamos que metemos la información de la carátula en una tabla que tiene esta forma:

Tabla caratula

id	Identificador de la carátula
titulo	Título del disco
imagen	Nombre del archivo imagen de la carátula
info	Descripción complementaria de la carátula

El código en PHP para hacer algo así sería el siguiente. En este código vamos a obviar algunas instrucciones de conexión con la base de datos que realmente no vienen al caso. Recordamos que en el [Manual de PHP I](#) tenemos todas las explicaciones para aprender a manejar bases de datos.

```
<?
//Extraigo el identificador de la carátula a mostrar
$id_caratula = $_HTTP_GET_VARS["id"];

//creo la sentencia SQL que extrae datos de esa carátula
$sql = "select * from caratula where id=" . $id_caratula;

//obtengo los datos de la carátula
$rs = mysql_query($sql);
$fila = mysql_fetch_object($rs);

//muestro los datos de esa carátula
?>
Disco: <?echo $fila->titulo;?>
<br>
Imagen: 
<br>
Descripción: <?echo $fila->info;?>

<?
//cierro la base de datos
mysql_free_result($rs);
//... todo lo que haga falta para terminar la página...
?>
```

Conclusión

Con esto acabamos. Esperamos que os sirva de ayuda este artículo o, por lo menos, haya resuelto vuestras dudas sobre la utilización de variables en la URL tanto en ASP como en PHP. Sólo pedir disculpas porque los ejemplos que he utilizado no son páginas completas al 100%,

ya que faltan las cabeceras de las páginas y otras instrucciones que no venían al caso. Además, como son ejemplos parciales, se me habría podido escapar alguna falta de sintaxis, al no poderse poner en marcha los scripts así, de manera parcial.

Lo importante es que hayáis cogido la idea y entendido mejor para qué puede servir pasar variables en la URL y cómo hacen los desarrolladores para que la página sea distinta dependiendo de los valores de las variables. Espero que haya sido así.

Artículo por Miguel Angel Alvarez

Programas de libre distribución en PHP

El lenguaje de programación de páginas dinámicas del servidor multiplataforma y totalmente gratuito, [PHP](#), tiene varias ventajas que no nos hemos cansado de señalar constantemente. Entre estas ventajas hay una que ahora mismo nos interesa comentar. Se trata de que existen en Internet multitud de programas gratuitos creados en PHP para realizar una buena cantidad de tareas habituales. Estos programas los podemos instalar en nuestro sistema libremente, por ser de libre distribución, y disfrutar de sus funcionalidades en nuestros servidores, insistimos, gratuitamente.

Hay muchos programas de libre distribución en PHP. Ejemplos típicos son servidores de banners, servidores de sistemas de afiliación, servidores orientados al control de bases de datos, sistemas de agenda, sistemas de chat, comercios electrónicos... y un largo etcétera que no vamos a alargar en estos momentos. Cualquier persona que desee informarse de la lista completa de scripts disponibles en PHP puede acercarse a la dirección http://www.hotscripts.com/PHP/Scripts_and_Programs/ que corresponde con la página de programas PHP que ofrece el sitio [Hotscripts.com](http://www.hotscripts.com). En Hotscripts nos presentan multitud de programas, unos gratuitos y otros de pago, pero en el caso de PHP la mayoría son gratuitos, al contrario de lo que ocurre en ASP, donde la mayoría tienen un precio, en ocasiones excesivamente elevado.

Hay un lugar muy típico de Internet donde fomentan y ayudan al desarrollo de programas de libre distribución es [SourceForge](http://sourceforge.net), con dirección [http://sourceforge.net/](http://sourceforge.net). Se trata de un servicio gratuito para desarrolladores de código abierto que ofrece fácil acceso a herramientas para trabajo en grupo, tableros de anuncios, foros, listas de discusión, seguimiento de errores, archivado de versiones, libre alojamiento de páginas web... como podemos ver todo lo que necesite un desarrollador para la publicación a todo lujo de sus aplicaciones gratuitas. En [SourceForge](http://sourceforge.net) "viven" muchos proyectos interesantes, que merecería la pena explicar por separado.

[SourceForge](http://sourceforge.net) es, por tanto, un lugar ideal para encontrar programas para PHP, ya que la mayoría de desarrollos en PHP son de libre distribución. Veamos rápidamente algún ejemplo:

phpMyAdmin

Es un gestor de base de datos MySQL, a través del cual podemos administrar todos los procesos de nuestra base de datos. Muy útil para tenerlo instalado en nuestro alojamiento PHP, ya que, la base de datos MySQL se tiene que administrar por consola y por línea de comandos y a veces el proveedor no nos ofrece esa posibilidad de conexión. Además, porque nos hace más fácil administrar los datos que con sentencias SQL, como se hace por línea de comandos.

Podemos [aprender más sobre phpMyAdmin y su instalación](#) en un artículo de DesarrolloWeb.com

phpAdsNew

Un servidor de banners que tiene las funcionalidades de los mejores productos comerciales. Posibilidad de crear clientes, campañas, banners, y administrar las impresiones de banners de nuestro sitio de la manera que mejor queramos. Mantiene unas excelentes estadísticas de impresiones y clicks, hace reports, etc.

PHP-Nuke

Un sistema completo para la administración de un portal web. Con este sistema podemos administrar un buscador de enlaces, noticias y artículos en general, usuarios registrados, encuestas, etc. Se trata de un software muy popular, ya que existen muchos desarrolladores que han adoptado esta solución para crear su propio portal sin necesidad de muchos esfuerzos en el tiempo de desarrollo.

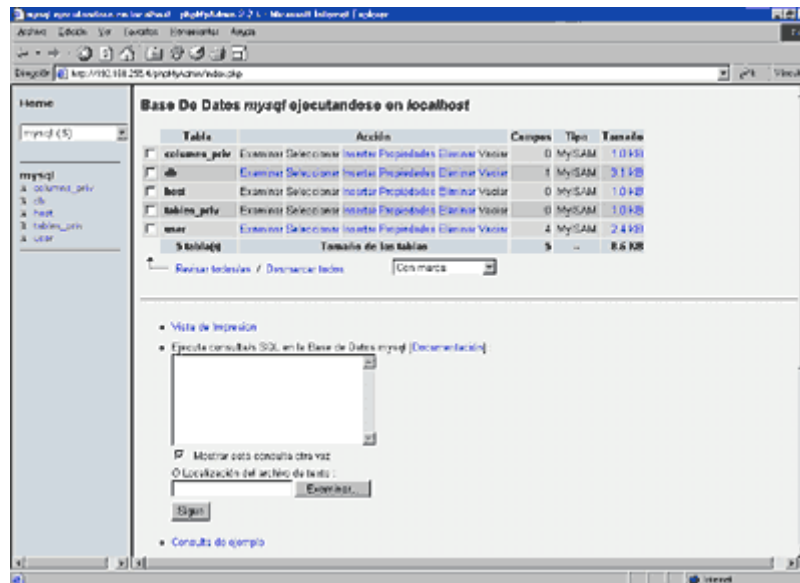
La lista es interminable y los programas que se ofrecen son realmente interesantes. Sin duda en PHP hay muchos desarrolladores generosos que han trabajado duro para presentarnos estas joyas gratuitas, con las que podemos olvidarnos del desarrollo de determinadas funcionalidades de nuestras páginas avanzadas. Además, sabiendo que dichas funcionalidades están cubiertas de una manera muy profesional y con cientos de horas de investigación que nosotros no deberíamos dedicar, a no ser que deseemos reinventar la rueda, que nunca es aconsejable.

Antes de acabar, apuntaremos la posibilidad de encontrar más directorios de scripts y programas listos para usar en general y de cualquier lenguaje, en nuestro [buscador en la categoría de colecciones de scripts](#). En el futuro y para continuar con esta exposición, comentaremos detenidamente alguno de estos programas de libre distribución para PHP.

*Artículo por **Miguel Angel Alvarez***

phpMyAdmin

phpMyAdmin es un programa de libre distribución en PHP, creado por una comunidad sin ánimo de lucro, que sólo trabaja en el proyecto por amor al arte. Es una herramienta muy completa que permite acceder a todas las funciones típicas de la base de datos MySQL a través de una interfaz web muy intuitiva.



La aplicación en sí no es más que un conjunto de archivos escritos en PHP que podemos copiar en un directorio de nuestro servidor web, de modo que, cuando accedemos a esos archivos, nos muestran unas páginas donde podemos encontrar las bases de datos a las que tenemos acceso en nuestro servidor de bases de datos y todas sus tablas. La herramienta nos permite crear tablas, insertar datos en las tablas existentes, navegar por los registros de las tablas, editarlos y borrarlos, borrar tablas y un largo etcétera, incluso ejecutar sentencias SQL y hacer un backup de la base de datos.

Página de phpMyAdmin

La página de inicio del proyecto es <http://www.phpmyadmin.net/>. Desde allí podemos descargar los ficheros de la última versión de la aplicación, que posteriormente debemos colocar en nuestro servidor web. También podemos encontrar a phpMyAdmin dentro de la red Sourceforge.net, que es un sitio que recoge multitud de proyectos "Open Source" (código abierto).

Hay varias versiones disponibles, pero es recomendable escoger la que nos aconsejen como la última versión estable (The last stable versión). En el momento de escribir este artículo era la 2.2.6. De modo que, si nuestro sistema es Windows, descargaremos el archivo phpMyAdmin-2.2.6-php.zip

Los archivos que hemos descargado son de la versión 4 de PHP, aunque también ofrecen la posibilidad de bajarse los archivos que guardan compatibilidad con la versión 3 de PHP, para que aquellos que no dispongan del motor de PHP más actual.

La página de inicio del programa también nos ofrece la posibilidad de ver un demo online, aunque nos avisan de que el servidor donde se aloja puede estar caído.

<http://www.phpmyadmin.net/phpMyAdmin/>

Home

desarrolloweb (9)

desarrolloweb

- academia_seccion
- seccion_curso
- tacademia
- tcertificados
- tcursos
- tmetodologia
- tprovincias
- tseccion
- ttipos

Base De Datos desarrolloweb - tabla tacademia ejec

Mostrando campos 0 - 2 (3 total)

SQL-query : [Editar]
SELECT * FROM tacademia LIMIT 0, 30

Mostrar: 30 filas empezando de 0
en modo horizontal y repite encabezados cada 100 celdas

	id_academia	nombre	contrasena	persona	em
Editar	1	academia12	password12	persona12	em
Editar	2	academia2	password2	persona2	em
Editar	9	333	333	33	33

Mostrar: 30 filas empezando de 0
en modo horizontal y repite encabezados cada 100 celdas

Instalando phpMyAdmin

Una vez descargada la última versión la tenemos que descomprimir, con lo que obtendremos los ficheros PHP que conforman la herramienta y colocarlos dentro del directorio de publicación de nuestro servidor web.

Nota: recordamos que phpMyAdmin es un proyecto escrito en PHP, por lo que necesitaremos colocar los archivos en un servidor web que permita programación de páginas PHP. Además, deberemos acceder a la herramienta a través de la dirección del servidor web, seguida del directorio en el que tenemos los archivos que hemos descomprimido. Por ejemplo, si nuestro servidor es el [PWS](http://www.pws.com) y hemos colocado los archivos dentro del directorio de publicación (Generalmente C:\inetpub\wwwroot), en el subdirectorio phpMyAdmin, debemos escribir algo como <http://localhost/phpMyAdmin>

Si tuviéramos instalado un servidor Apache los colocaríamos en la carpeta que hayamos indicado como "documentRoot", que suele ser htdocs.

Lo primero que podemos leer es el archivo de la documentación, que encontramos junto con los archivos de phpMyAdmin. Explica datos generales del programa, como sus requerimientos, instrucciones de instalación, configuración, preguntas frecuentes, etc.

Posteriormente, tal como explica la documentación, hay que editar el archivo config.inc.php para cambiar los valores de host de la base de datos (ordenador que tiene instalado el MySQL) y el usuario y password con el que nos conectamos. Se pueden configurar muchos aspectos en la herramienta, aunque ahora solo comentaré los que he encontrado esenciales para hacerla funcionar, en la documentación tenemos un apartado dedicado por completo a especificar el sentido de cada variable.

\$cfgPmaAbsoluteUri

Debemos asignarlo a la ruta completa necesaria para acceder a phpMyAdmin. Podría ser algo como <http://localhost/phpMyAdmin> o <http://www.midominio.com/phpMyAdmin>

\$cfgServers[\$i]['host'] string

El nombre del host de la base de datos. Por ejemplo localhost, si es que es el mismo ordenador donde estamos instalando phpMyAdmin y la base de datos. También podría ser la dirección IP

del ordenador al que nos conectamos.

```
$cfgServers[$i]['user'] string  
$cfgServers[$i]['password'] string
```

El par usuario/contraseña que debe utilizar phpMyAdmin para conectarse con el servidor MySQL.

Con estas sencillas configuraciones ya podemos acceder a phpMyAdmin y trabajar con nuestra base de datos a golpe de ratón, que resulta muy de agradecer teniendo en cuenta que, en caso de no tener esta herramienta u otra parecida, la otra opción consistiría en utilizar el lenguaje SQL, y, en caso de que la base de datos esté alojada remotamente en Internet, no podríamos hacerlo sino es con acceso por TELNET al servidor de la base de datos.

Referencias: En DesarrolloWeb puedes conocer más cosas de PHP y MySQL.

En la [sección de PHP](#), podras aprender mucho de PHP, algo sobre MySQL e incluso sobre el lenguaje SQL. En el [directorio dedicado a MySQL](#) hay algunas referencias a artículos y enlaces externos.

*Artículo por **Miguel Angel Alvarez***

phpMyAdmin sobre Debian

Esta es una mini guía de instalación de phpMyAdmin en Debian GNU/Linux, con el objetivo de que funcione o se integre con Apache, MySQL y PHP. Se da por hecho que Apache, MySQL y PHP están funcionando correctamente en el sistema, es decir, php con su modulo para Apache, etc.

Lo primero es instalar el paquete, en los repositorios oficiales debería estar disponible el paquete para ser instalado.

```
# aptitude install phpmyadmin
```

Una vez que la descarga esté lista, se pedirá por la configuración de Apache, es decir, el sistema reconocerá automáticamente instalaciones de Apache en nuestro sistema. Si usas Apache 1.3.x o Apache2, solo debes seleccionarlo con la barra espaciadora y luego aceptar.

Se creara un link simbólico en el directorio /var/www/ apuntando al phpmyadmin, que realmente está instalado en /etc/phpmyadmin.

Ahora tocará configurar phpmyadmin para poder entrar a él mediante password. Es una medida de seguridad que se debe tener en cuenta.

Lo primero es cambiarle el password al usuario root de MySQL (no del sistema), esto se hará para tener mas seguridad en nuestras bases de datos.

Como root ejecutar:

```
# mysql -u root mysql
```

Con esto lo que hace es entrar en MySQL como usuario root a la base de datos que se llama mysql (es la base de datos que viene por defecto junto a la test).

Luego dentro de MySQL, escribir:

```
mysql> UPDATE user SET Password = PASSWORD('contraseña')
WHERE User = 'root';
```

Sustituyendo "contraseña" por la contraseña que se quiere asignar.

Con esto lo que hace es actualizar una tabla de configuración que hay en MySQL llamada user con los valores que se indican. Esta tabla posee todos los usuarios de las firentes bases de datos. Podremos agregar la cantidad de usuarios que queramos con diferentes permisos.

Ahora modificaremos el fichero de configuración de phpmyadmin.

Utilizamos vi para editar el archivo:

```
# vi /etc/phpmyadmin/config.inc.php
```

Se deben descomentar las siguientes líneas.

Esta línea es para dejar el tipo de autenticación, es decir, mediante cookies, http o entrar sin pedir contraseña (config). Por defecto la dejaremos en cookie.

```
$cfg['Servers'][$i]['auth_type'] = 'cookie';
```

Aquí se indica el usuario que se conectará a las bases de datos. Por defecto es el usuario root de MySQL. Se recomienda agregar otro usuario con algunos privilegios de root, y así manejar las bases de datos con él, ya que con root podrá ser peligroso.

```
$cfg['Servers'][$i]['user'] = 'usuario';
```

Aquí especificamos la contraseña que hemos puesto al usuario que hemos modificado en la base de datos mysql.

```
$cfg['Servers'][$i]['password'] = 'contraseña';
```

Guardamos el archivo presionado "Esc" y luego ":wq".
Por último se deben reiniciar los servicios de MySQL y Apache.

Para Apache:

```
# /etc/init.d/apache restart
o
# apachectl restart
```

Para MySQL:

```
# /etc/init.d/mysql restart
```

Un dato a agregar, es que podemos manejar los servicios que se inician al encender el computador con un programa llamado rconf, se puede instalar con APTITUDE, luego solo se ejecuta con rconf y seleccionamos o deseccionamos los servicios con la barra espaciadora.

Instalar rconf:

```
#aptitude install rconf
```

Ejecutar rconf:

```
#rconf
```

Luego debes aceptar los cambios.

Mas Información:

http://www.phpmyadmin.net/home_page/index.php

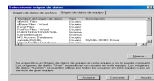
Artículo por **Luis Vega M.**

Exportar datos de MySQL a Microsoft Access 2000

Migrar datos de una base de datos a otra es algo a lo que muchos de nosotros hemos tenido que confrontarnos en algún momento. A continuación os explicamos cómo recuperar información almacenada en un servidor de datos Mysql hacia una base Access 2000.

Referencia: Para realizar esta tarea es necesario que hayamos descargado el driver ODBC y lo hayamos instalado en nuestro sistema Windows. Esta labor se puede conocer en un artículo de DesarrolloWeb.com: [Instalar el driver ODBC para MySQL](#).

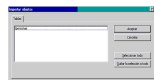
Para importar una tabla de Mysql a Microsoft Access, desde Access, y con la base de datos en la que se quieren importar los datos abierta, seleccionar el menu Archivo->Obtener datos Externos->Importar. En la pantalla de Importar datos, en la opción Tipo de archivo seleccionar ODBC databases().



Seleccionar origen de datos de equipo, y dentro de esta, el nombre de la fuente de datos que hemos creado anteriormente. Una vez la has seleccionado, y has hecho clic sobre "Aceptar", aparecerá la pantalla de configuración del driver por si deseas marcar para esta acción en concreto, algunas de las opciones de configuración que aparecen en el driver ODBC, si no deseas marcar ninguna, clic sobre "OK".

Nota: pudiera ser en algún caso que los tipos de los datos de la base en los sistemas MySQL y Access no sean totalmente compatibles y se produzca alguna anomalía al exportarlos. Realmente es una posibilidad que pensamos, aunque en las pruebas que hemos realizado no hemos visto ningún tipo de problema, bien es cierto que los campos que hemos trabajado no eran muy raros.

Aparecerá una ventana donde pregunta qué tabla de Mysql se desea exportar a Access:



Selecciona la tabla , y haz clic sobre "Aceptar"

Nota: si estamos exportando los datos hacia o desde un servidor de bases de datos alojado en algún proveedor de Hosting, tenemos que tener en cuenta que estos no siempre incluyen en su paquete básico el acceso remoto al servidor de base de datos, o requiere de un aviso explícito por parte del cliente para su configuración.

Referencia: si deseamos realizar una migración de datos en el otro sentido, es decir, desde Access hacia MySQL, será muy indicado leer otro artículo en DesarrolloWeb que explica el proceso detalladamente. [Exportar datos de Access 2000 a MySQL](#).

Artículo por **Carlos Luis Cuenca**

Exportar datos de Access 2000 a MySQL

No es de extrañar que hayamos comenzado a hacer nuestros pinitos en la web sirviéndonos de una base de datos sencilla como Access. Tampoco es de extrañar que, llegado el momento, pasemos a cosas más serias y nos pasemos a un servidor de datos como MySQL. Aquí os mostramos una manera bastante práctica de migrar los datos de la una a la otra.

Referencia: Para realizar esta tarea es necesario que hayamos descargado el driver ODBC y lo hayamos instalado en nuestro sistema Windows. Esta labor se puede conocer en un artículo de DesarrolloWeb.com: [Instalar el driver ODBC para MySQL](#).

Para exportar una tabla a Mysql, hay que abrir la base de datos y seleccionar la tabla. Después, hacer clic sobre Archivo->Exportar. En la pantalla de exportar, en la opción Guardar como tipo, seleccionar ODBC databases().

Una vez se ha hecho esto, aparece una ventana que nos pregunta el nombre que le queremos dar a la tabla en Mysql, por defecto aparece el mismo.



Haz clic sobre "Aceptar", y aparecerá la pantalla en la que se pide que selecciones el origen de datos ODBC:

Nota: pudiera ser en algún caso que los tipos de los datos de la base en los sistemas MySQL y Access no sean totalmente compatibles y se produzca alguna anomalía al exportarlos. Realmente es una posibilidad que pensamos, aunque en las pruebas que hemos realizado no hemos visto ningún tipo de problema, bien es cierto que los campos que hemos trabajado no eran muy raros.



Seleccionar origen de datos de equipo, y dentro de esta el nombre de la fuente de datos que hemos creado anteriormente. Una vez la has seleccionado y has hecho clic sobre "Aceptar", aparecerá la pantalla de configuración del driver por si deseas marcar para esta acción en concreto algunas de las opciones de configuración que aparecen en el driver ODBC. Si no deseas marcar ninguna, haz clic sobre "OK" y los datos comenzarán a exportarse.

Nota: si estamos exportando los datos hacia o desde un servidor de bases de datos alojado en algún proveedor de Hosting, tenemos que tener en cuenta que estos no siempre incluyen en su paquete básico el acceso remoto al servidor de base de datos, o requiere de un aviso explícito por parte del cliente para su configuración.

Referencia: si deseamos realizar una migración de datos en el otro sentido, es decir, desde MySQL hacia Access, será muy indicado leer otro artículo en DesarrolloWeb que explica el proceso detalladamente. [Exportar datos de MySQL a Microsoft Access 2000](#).

Artículo por **Carlos Luis Cuenca**

Contador simple para páginas PHP

Hice una modificación al Script publicado en el artículo [Escritura de archivos con PHP](#), en el que se enseña a escribir archivos de texto mediante PHP, tocando los temas de lectura y escritura.

El objetivo es llevar un conteo de las veces que se ha visitado una página.

Puse el siguiente script PHP al final de la página, se entenderá bien si se lee el artículo señalado antes.

```
<?
$archivo = "contador.txt";
$contador = 0;

$fp = fopen($archivo,"r");
$contador = fgets($fp, 26);
fclose($fp);

++$contador;

$fp = fopen($archivo,"w+");
fwrite($fp, $contador, 26);
fclose($fp);

echo "Esta página ha sido visitada $contador veces";
?>
```

Además, creé un archivo llamado "contador.txt" que lo guardé en el mismo directorio que la página. Dicho archivo fue inicializado con un cero (0) como único texto.

Nota: si tenéis problemas a la hora de escribir en un archivo, casi con toda probabilidad, estará protegido contra escritura. O bien el archivo o bien el directorio.

Si tenéis vuestro propio servidor tendréis que modificar los permisos de tal archivo o directorio por vosotros mismos. Sin embargo, si estáis publicando en un alojamiento contratado en un proveedor tendréis que enteraros de qué mecanismo hay que poner en marcha en ese proveedor para conseguir los permisos. En muchos casos existirá un panel de control donde modificar esas cosas, en otros casos tendréis que escribir a soporte técnico para que lo hagan a mano ellos o os digan cómo hacerlo, si es que es posible.

Con esto ya está hecho un contador muy simple, pero muy funcional.

Referencia: Hemos publicado un artículo que amplía esta práctica sobre el contador. En concreto se hace un contador que no sólo registra las visitas, sino también las visitas en un mes y el mes de la última visita. El artículo se llama [Contador mejorado para páginas PHP](#).

También está publicado un artículo sobre [un contador PHP que utiliza imágenes](#) para mostrar el número de visitas.

*Artículo por **Daniel Guajardo***

Gestión de archivos por PHP

El tratamiento de archivos resulta ser una práctica muy común en cualquier sitio web. Muy a menudo nos vemos en la necesidad de procesar un texto para cambiarle el formato, buscar una cadena en su interior o cualquier otro tipo de operación.

PHP propone un sinfín de [funciones para la gestión de archivos](#) que van desde las más elementales de apertura, lectura y cierre a otras más rebuscadas como el cálculo de espacio en el disco duro, tamaño del archivo, gestión de derechos de acceso...

En este artículo pretendemos mostraros cuáles son las funciones más esenciales para el

tratamiento de archivos para posteriormente ejemplificarlas en un par de scripts que os pueden resultar útiles:

Funciones de gestión de archivos

Función	Descripción	Sintaxis
copy	Copia un archivo	copy(\$origen,\$destino)
rename	Cambia el nombre del archivo de <i>\$antes</i> a <i>\$despues</i>	rename(\$antes,\$despues)
unlink	Borra el archivo	unlink(\$archivo)

Funciones para la lectura de archivos

Función	Descripción	Sintaxis
fopen	Abre un archivo y le asigna un identificador id. Veremos el modo más adelante	\$id = Fopen(\$archivo,\$modo)
fgets	Lee una línea de un archivo hasta un numero máximo de caracteres	fgets(\$id,\$max)
fwrite	Escribe una cadena dentro del archivo	fwrite(\$id,\$cadena)
fseek	Avanza o retrocede el puntero del archivo un cierto numero de posiciones	fseek(\$id,\$posiciones)
feof	Comprueba si el puntero que lee el archivo ha llegado al final	feof(\$id)
fpassthru	lee completamente el archivo y lo muestra	fpassthru(\$id)
fclose	Cierra el archivo abierto previamente	fclose(\$id)

Las operaciones más elementales, copia, borrado y cambiar el nombre, requieren únicamente el nombre (y path) del archivo sobre el cual se ejerce la operación. Para operaciones más complejas, como la lectura de líneas o la escritura de texto dentro del archivo, se requiere de una previa apertura del archivo al cual le asignaremos un identificador *\$id*.

Una vez abierto el archivo, podremos desplazarnos a lo largo de él por medio de un puntero imaginario que avanza o retrocede por las líneas de texto y mediante el cual nos situaremos en el lugar escogido para insertar, modificar o simplemente copiar una cadena.

Existen distintos modos de apertura que nos permiten definir las acciones que podemos realizar sobre el archivo. Aquí os mostramos los diferentes modos que, como veréis, son de lo más variado:

Modos de apertura de archivos

Sintaxis	Descripción
'r'	Sólo lectura
'r+'	Lectura y escritura
'w'	Sólo escritura
'w+'	Lectura y escritura. Suprime el contenido anterior si se escribe. El archivo es creado si no existe.
'a'	Sólo escritura. El archivo es creado si no existe y el puntero se coloca al final.
'a+'	Lectura y escritura. El archivo es creado si no existe y el puntero se coloca al final.

A notar que si tratamos con archivos en binario hemos de colocar una *b* delante del modo (ej.

ba, bw+,...)

Recordamos que esta lista no es más que una recopilación y que muchas [otras funciones relacionadas](#) pueden sernos también útiles.

Nota: También puedes profundizar más en la gestión de archivos con nuestro manual [Gestión de archivos en PHP](#)

Artículo por **Rubén Alvarez**

Gestión de directorios por PHP

Siguiendo con la saga de artículos referentes a la explotación de archivos por medio de PHP, vamos a presentar algunas funciones que nos pueden ser muy útiles en la navegación por directorios. Este tipo de funciones podrían, por ejemplo, servirnos para crear exploradores de archivos en nuestro navegador.

gestión de directorios

	Descripción	Sintaxis
Abre un directorio situado en \$path y le asigna un identificador \$dir		\$dir = opendir(\$path)
Lee un elemento del directorio \$dir abierto previamente con opendir y desplaza el puntero al elemento siguiente		readdir(\$dir)
Elimina el directorio \$dir		rmdir(\$dir)
Creación de un directorio situado en \$path con los derechos de acceso \$derechos (entero)		mkdir(\$path, \$derechos)
Rebobina el puntero de lectura del directorio \$dir al primer elemento		rewinddir(\$dir)
Cierra el directorio \$dir abierto previamente con opendir		closedir(\$dir)

La forma de tratar con estas funciones es la similar a la que ya hemos visto para la lectura secuencial de archivos. Podemos distinguir tres etapas elementales:

- Apertura del directorio por medio de la función *opendir* asignándole al mismo tiempo un identificador
- Realización de las tareas necesarias en relación con ese directorio
- Clausura del identificador por medio de la función *closedir*

A notar que, para que un directorio pueda ser borrado, hace falta previamente haber eliminado cada uno de los elementos contenidos dentro del directorio. Para esto, nos podemos servir de la función *unlink*, presentada en otro [artículo](#).

Por otra parte, la creación de un directorio por medio de la función *mkdir* requiere la definición de los derechos de acceso por medio de un número entero. Esperamos poder explicar en qué consisten estos derechos próximamente.

Como ejemplo sencillo de lo que podemos hacer con estas funciones, aquí os presentamos un pequeño script de lectura que os permite visualizar el contenido de un directorio:

```
<?php
//definimos el path de acceso
```

```

$path = "mi/camino";

//abrimos el directorio
$dir = opendir($path);

//Mostramos las informaciones
while ($elemento =
readdir($dir))
{
    echo $elemento."<br>";
}

//Cerramos el directorio
closedir($dir);
?>

```

Otra forma de abordar la gestión de directorios es por medio de la clase *dir* que permite la creación de un objeto sobre el cual podremos aplicar toda una serie de métodos equivalentes a las funciones previamente vistas. Si estas familiarizado con la programación orientada a objetos, puede que esta modalidad te resulte mas intuitiva. Si no sabes en qué consiste la programación orientada a objetos, puedes visitar este [artículo](#).

En este caso, la forma de operar es análoga a la ya vista:

- Creamos un objeto *\$dir* con la instrucción: `$dir = dir($path)`
- Realizamos las tareas necesarias llamando a los métodos de la clase *dir*
- Cerramos el directorio con el método *close*

Algunos de los métodos que podemos utilizar con esta clase son los siguientes:

Métodos de la clase *dir*

Método	Descripción	Sintaxis
path	Indica el path del directorio	<code>\$objeto->path</code>
read	Lee un elemento del directorio	<code>\$objeto->read</code>
rewind	Vuelve el puntero de lectura del directorio al primer elemento	<code>\$objeto->rewind</code>
close	Cierra el directorio	<code>\$objeto->close</code>

Como ejemplo, he aquí el script equivalente al abordado para el caso de las funciones, esta vez usando la clase *dir*:

```

<?
//definimos el path de acceso
$path="mi/camino/";

//instanciamos el objeto
$dir=dir($path);

//Mostramos las informaciones
echo "Directorio ".$dir-
>path.":<br><br>";

```

```
while ($elemento = $dir->read())
{
    echo $elemento."<br>";
}
//Cerramos el directorio
$dir->close();
?>
```

Los scripts propuestos no son más que ejemplos sencillos de lo que estas funciones pueden ofrecernos. En vuestras manos queda el combinar estas funciones con otras vistas en este mismo taller de manera a crear aplicaciones que gestionen los archivos y directorios de vuestro servidor.

Recordamos que esta lista no es más que una recopilación y que muchas [otras funciones relacionadas](#) pueden sernos también útiles.

Artículo por [Rubén Alvarez](#)

Mandar mails desde PHP

Para el envío de correos electrónicos utilizando PHP disponemos de una función bastante potente, incluida en todas las versiones de PHP, sin necesidad de instalar ningún añadido, en contra de lo que ocurría con ASP.

Referencia: En caso de que necesitemos programar el envío de correo electrónico en nuestra página utilizando ASP, también hemos publicado un artículo en DesarrolloWeb.com llamado [Mandar mails desde ASP](#).

En concreto, en PHP disponemos de una función llamada mail() que permite configurar y enviar el mensaje de correo. La función se llama mail() y recibe tres parámetros de manera obligada y otros dos parámetros que podemos colocar opcionalmente. Devuelve true si se envió el mensaje correctamente y false en caso contrario.

Parámetros necesarios en todos los casos

Destinatario: la dirección de correo o direcciones de correo que han de recibir el mensaje. Si incluimos varias direcciones debemos separarlas por una coma.

Asunto: para indicar una cadena de caracteres que queremos que sea el asunto del correo electrónico a enviar.

Cuerpo: el cuerpo del mensaje, lo que queremos que tenga escrito el correo.

Ejemplo de envío de un mail sencillo

```
<?
mail("pepito@desarrolloweb.com,maria@guiartemultimedia.com","asuntillo","Este es el cuerpo del
mensaje")
?>
```

Parámetros opcionales del envío de correo

Headers: Cabeceras del correo. Datos como la dirección de respuesta, las posibles direcciones que recibirán copia del mensaje, las direcciones que recibirán copia oculta, si el correo está en formato HTML, etc.

Additional_parameters: esta opción no suele utilizarse y, además, sólo está disponible a partir de la versión PHP 4.0.5 y desde PHP 4.2.3 está deshabilitado en modo seguro. Puede usarse para pasar parámetros adicionales al programa configurado para enviar el correo, cuando se manda el mail usando la opción de configuración `sendmail_path`. Podemos obtener más información en la [documentación de PHP para la función mail\(\)](#).

Ejemplo complejo de envío de correo

Vamos a enviar un correo con formato HTML a `pepito@desarrolloweb.com`, con copia a `mariano@desarrolloweb.com` y con copia oculta para `pepe@pepe.com` y `juan@juan.com`. La dirección de respuesta la configuraremos a `maria@desarrolloweb.com`.

```
<?
$destinatario = "pepito@desarrolloweb.com";
$asunto = "Este mensaje es de prueba";
$cuerpo = '
<html>
<head>
  <title>Prueba de correo</title>
</head>
<body>
<h1>Hola amigos!</h1>
<p>
<b>Bienvenidos a mi correo electrónico de prueba</b>. Estoy encantado de tener tantos lectores. Este
cuerpo del mensaje es del artículo de envío de mails por PHP. Habría que cambiarlo para poner tu propio
cuerpo. Por cierto, cambia también las cabeceras del mensaje.
</p>
</body>
</html>
';

//para el envío en formato HTML
$headers = "MIME-Version: 1.0\r\n";
$headers .= "Content-type: text/html; charset=iso-8859-1\r\n";

//dirección del remitente
$headers .= "From: Miguel Angel Alvarez <pepito@desarrolloweb.com>\r\n";

//dirección de respuesta, si queremos que sea distinta que la del remitente
$headers .= "Reply-To: mariano@desarrolloweb.com\r\n";

//ruta del mensaje desde origen a destino
$headers .= "Return-path: holahola@desarrolloweb.com\r\n";

//direcciones que reciben copia
$headers .= "Cc: maria@desarrolloweb.com\r\n";

//direcciones que recibirán copia oculta
$headers .= "Bcc: pepe@pepe.com,juan@juan.com\r\n";

mail($destinatario,$asunto,$cuerpo,$headers)
?>
```

Nota: Antes de poner en marcha el script en vuestro servidor, por favor, cambiar los datos de configuración de las direcciones de correo que van a recibir el mensaje y colocar unas direcciones que sean vuestras y donde podáis comprobar si los mensajes se envían correctamente.

Conclusión y descarga

Pensamos y esperamos que después de este artículo compartáis nuestra opinión, que el envío de mails en PHP es una tarea muy sencilla. Además, es muy de agradecer que todas las versiones de PHP incluyan una función para el envío de mails.

Ponemos a vuestra disposición para la descarga el archivo .php con el [código anterior completo para realizar el envío de mails](#).

Nota: Para el envío de correo mediante PHP es necesario que este disponga de una correcta configuración.

Si nuestro web está en un servidor de un proveedor de hosting seguramente ya hayan configurado PHP para el envío de mails. Si estamos trabajando en un servidor propio, sí tendremos que configurar PHP.

PHP se configura en el archivo php.ini, donde debemos especificar datos como el servidor de correo saliente que debe de utilizar PHP para transferir los mensajes.

Dentro del php.ini, debemos buscar el epígrafe [mail function]. Dependiendo de nuestro sistema deberemos configurar de una manera u otra.

En sistemas Windows encontraremos el php.ini en el directorio windows o dentro de este, en el subdirectorio system32 o similar. En este sistema deberemos indicar el dominio del servidor de smtp, algo como smtp.midominio.com. Si es el ordenador local el que hace de servidor, podremos poner "localhost" como máquina que enviará el correo. También podemos especificar la dirección desde donde queremos que parezca que se envía el mensaje en caso de que no se indique otra durante el envío.

Artículo por *Miguel Angel Alvarez*

validar email en PHP

Vamos a ver una función muy útil en PHP que sirve para comprobar la validez de un correo. En realidad comprueba si una dirección de correo electrónico está bien escrita sintácticamente, dejando de lado las comprobaciones de si ese mail existe o no realmente, que no se pueden hacer tan fácilmente.

Vamos a escribir una función que se llama comprobar_email y recibe la cadena de texto con el email que queremos validar. Si dicho email es correcto desde el punto de vista sintáctico, es decir, si tiene un nombre de usuario, una arroba y una terminación con el nombre de un dominio o subdominio, etc, devolverá un 1, es decir, verdadero. En caso de que el email no esté correctamente escrito, la función devolvería 0, que equivale a falso.

La función en si da por hecho inicialmente que el email es erróneo y realiza una serie de comprobaciones que, si todas responden correctamente, dan por conclusión que el email sí estaba bien escrito. Si alguna de esas comprobaciones no era correcta, no se llegaría al final de las comprobaciones y quedaría el resultado como se ha supuesto en un principio, es decir, como incorrecto.

código de la función

```
function comprobar_email($email){  
    $mail_correcto = 0;  
    //compruebo unas cosas primeras
```

```
if ((strlen($email) >= 6) && (substr_count($email,"@") == 1) && (substr($email,0,1) != "@") &&
(substr($email,strlen($email)-1,1) != "@")){
    if ((!strstr($email,"")) && (!strstr($email,"\"")) && (!strstr($email,"\\")) && (!strstr($email,"\$")) && (!
strstr($email," "))) {
        //miro si tiene caracter .
        if (substr_count($email,".")>= 1){
            //obtengo la terminacion del dominio
            $term_dom = substr(strrchr ($email, '.'),1);
            //compruebo que la terminación del dominio sea correcta
            if (strlen($term_dom)>1 && strlen($term_dom)<5 && (!strstr($term_dom,"@")) ){
                //compruebo que lo de antes del dominio sea correcto
                $antes_dom = substr($email,0,strlen($email) - strlen($term_dom) - 1);
                $caracter_ult = substr($antes_dom,strlen($antes_dom)-1,1);
                if ($caracter_ult != "@" && $caracter_ult != "."){
                    $mail_correcto = 1;
                }
            }
        }
    }
}
if ($mail_correcto)
    return 1;
else
    return 0;
}
```

Las comprobaciones

En el primer if compruebo que el email tiene por lo menos 6 caracteres (el mínimo), que tiene una arroba y sólo una y que no está colocada ni al principio ni al final.

En el segundo if comprueba que no tiene algunos caracteres no permitidos. Y los restantes hacen comprobaciones de las distintas partes de la dirección de correo, a saber: Que hay un punto en algún lado y que la terminación del dominio es correcta y que el principio de la dirección también es correcto.

Finalmente, se devuelve la variable local utilizada para guardar la validez o incorrección del correo.

Descarga del script

La persona que lo desee, puede [descargar el archivo con el código fuente de este script](#), para utilizarlo en sus aplicaciones web.

Artículo por ***Miguel Angel Alvarez***

Paginación de resultados con PHP y MySQL

En muchas ocasiones, cuando se presentan en una página web registros de una base de datos, se deberían mostrar demasiados registros como para colocarlos todos en una única página. En estas ocasiones se suele paginar los resultados, quizás cientos, en distintas páginas con conjuntos de registros mucho menos numerosos. Por ejemplo, podríamos presentar los resultados en páginas de 10 elementos o 20, dependiendo de nuestras intenciones y el tipo de datos que se estén presentando. Este efecto lo habremos podido observar repetidas veces en

los buscadores.

Podríamos desarrollar distintos scripts para paginar resultados en PHP. En este artículo vamos a explicar una posibilidad basada en la utilización de una base de datos MySQL y sentencias SQL a las que indicaremos el conjunto de registros que queremos mostrar en cada página. Los enunciados SELECT del lenguaje SQL, en la base de datos MySQL y otras muchas, tienen una cláusula llamada LIMIT, con la que podemos indicar los registros a mostrar, por ejemplo, 10 registros empezando por el registro 180.

```
select * from pais limit 180,10
```

Como vemos LIMIT tiene dos argumentos, el primero es el registro por el que empezar los resultados y el segundo el número de resultados a recoger en el conjunto de registros resultante.

Así pues, en este ejercicio de paginación la cláusula LIMIT será la clave para mostrar los registros en grupos del tamaño deseado.

Podemos [ver el resultado que vamos a conseguir en este artículo](#) ahora y así tendremos más facilidad de identificar las distintas partes del código que vamos a comentar.

Código de paginación

Hay varias partes del código que servirán específicamente para implementar la paginación. Lo primero es saber qué página se desea mostrar. En principio se mostraría la primera página de resultados, pero si un visitante selecciona con los enlaces de abajo otra página distinta de la primera, habría que mostrarla también. El índice de la página a mostrar, si es que no es la primera vez que se accede, se recibe por parámetro en la URL.

```
//Límite la búsqueda
$TAMANO_PAGINA = 10;

//examinamos la página a mostrar y el inicio del registro a mostrar
$página = $_GET["pagina"];
if (!$página) {
    $inicio = 0;
    $página=1;
}
else {
    $inicio = ($página - 1) * $TAMANO_PAGINA;
}
```

Estoy definiendo el tamaño de la página. Luego procuro recibir la página por parámetro en la URL. Si no se recibió nada, se entiende que la página a mostrar es la primera, luego la variable \$inicio, que guarda el primer registro a mostrar (para indicarlo en la sentencia SQL en el apartado LIMIT), será cero. Si habíamos recibido algo como página, calculo el inicio con una simple multiplicación de la página a mostrar por el tamaño_ de página definido antes.

Es habitual en estas páginas de resultados informar un poco sobre la cantidad de registros encontrados y los datos de la página que estamos viendo. Estos datos se pueden obtener con unas sencillas operaciones.

```
//miro a ver el número total de campos que hay en la tabla con esa búsqueda
$ssql = "select * from pais " . $criterio;
$rs = mysql_query($ssql,$conn);
$num_total_registros = mysql_num_rows($rs);
```

```
//calculo el total de páginas
$total_paginas = ceil($num_total_registros / $TAMANO_PAGINA);

//pongo el número de registros total, el tamaño de página y la página que se muestra
echo "Número de registros encontrados: " . $num_total_registros . "<br>";
echo "Se muestran páginas de " . $TAMANO_PAGINA . " registros cada una<br>";
echo "Mostrando la página " . $pagina . " de " . $total_paginas . "<p>";
```

Nota: Este código podría mostrar una información como esta:

```
Número de registros encontrados: 256
Se muestran páginas de 10 registros cada una
Mostrando la página 19 de 26
```

Lo primero es hacer una búsqueda en la base de datos por el criterio que se esté utilizando para saber cuantos registros se obtienen en total sin la paginación (luego veremos de donde sale la variable \$criterio).

A continuación puedo calcular el número total de páginas de resultados que genera la búsqueda. La función ceil() redondea números en coma flotante o reales hacia arriba, así pues, devuelve el entero por arriba más próximo.

Las siguientes líneas, donde se utiliza echo, tienen como objeto mostrar los datos en la página.

Ahora veremos el código que realiza la búsqueda en la base de datos, extrayendo y mostrando solamente aquellos registros que corresponden con la página a mostrar.

```
//construyo la sentencia SQL
$sql = "select * from pais " . $criterio . " limit " . $inicio . "," . $TAMANO_PAGINA;
$rs = mysql_query($sql);
while ($fila = mysql_fetch_object($rs)){
    echo $fila->nombre_pais . "<br>";
}
//cerramos el conjunto de resultado y la conexión con la base de datos
mysql_free_result($rs);
mysql_close($conn);
```

Se construye la sentencia SQL para extraer los datos con el criterio, que veremos luego de donde sale, pero que en principio lo podemos tomar como una cadena vacía. También se utiliza LIMIT, como ya se indicó: poniendo los valores definidos antes como inicio y tamaño de página.

El resto es un recorrido típico por un conjunto de registros, en este caso los países de nuestra base de datos, donde se van mostrando todos los elementos desde el principio hasta el final. Finalizando este recorrido no vamos a realizar ninguna acción más con la base de datos, así que podemos cerrar el resultado de la búsqueda y la conexión con la base de datos.

Ahora podemos ver el código que muestra en la parte de abajo los numeritos de todas las páginas que genera la búsqueda, para que el visitante pueda seleccionar una página y moverse entre los resultados.

```
//muestro los distintos índices de las páginas, si es que hay varias páginas
if ($total_paginas > 1){
    for ($i=1;$i<=$total_paginas;$i++){
        if ($pagina == $i)
            //si muestro el índice de la página actual, no coloco enlace
            echo $pagina . " ";
    }
}
```



```
else
    //si el índice no corresponde con la página mostrada actualmente, coloco el enlace para ir a esa página
    echo "<a href='index.php?pagina=" . $i . "&criterio=" . $txt_criterio . "'>" . $i . "</a> ";
}
}
```

La primera línea comprueba si realmente hay varias páginas de resultados, pues, si no es así no tendría que mostrar nada. Si efectivamente hay varias páginas para mostrar se recorren todas y para cada una se muestra el índice.

El índice a mostrar puede que sea el de la página que se está visualizando en ese momento y en ese caso simplemente podríamos el numerito, pero no el enlace para ir a ese documento, pues es en el que estamos. En caso de que sea una página de resultados distinta, se muestra un enlace para moverse a dicha página, donde se incluye pasando por parámetro tanto el índice de la página que se desea ver como el criterio de la búsqueda que se estaba realizando.

Hasta aquí el código imprescindible para la paginación. Aunque aun vamos a ver alguna cosa más.

Código de búsqueda

Para hacer un taller un poco más completo y poder ofrecer una página de muestra con funcionalidades de búsqueda, hemos creado la posibilidad de añadir un criterio para encontrar tan sólo elementos relacionados con él. Después de todo, los códigos de paginación suelen utilizarse en situaciones en las que se están realizando búsquedas en la base de datos.

El criterio se podrá definir en una caja de texto y habrá un botón de buscar que llame a la misma página pero pasando el texto con las palabras a buscar en la base de datos.

```
<form action="index.php" method="get">
Criterio de búsqueda:
<input type="text" name="criterio" size="22" maxlength="150">
<input type="submit" value="Buscar">
</form>
```

Nos fijamos que el método por el que pasamos este formulario es GET. Esto es debido a que no queremos liar el código y como estamos pasando ya por GET el criterio en otros sitios, utilizamos el mismo método.

El formulario lo colocaremos debajo, pero habrá también un trozo de código que recogerá la información y la tratará para adaptarla a una sentencia de búsqueda en la base de datos. Este código lo colocaremos en la parte de arriba de la página.

```
//inicializo el criterio y recibo cualquier cadena que se desee buscar
$criterio = "";
if ($_GET["criterio"]!=""){
    $txt_criterio = $_GET["criterio"];
    $criterio = " where nombre_pais like '%" . $txt_criterio . "%'";
}
```

Se inicializa el criterio a una cadena vacía y luego se comprueba si se ha recibido algo por método GET en el campo criterio. Si era así se puede recoger el texto recibido y construir el criterio, que no es más que una cláusula WHERE donde se buscan elementos, en este caso países, cuyo nombre contenga por algún sitio las letras que se han recibido como texto del criterio.

Base de datos

Como se dijo, se está utilizando una base de datos MySQL. En el [Manual de Programación en PHP de DesarrolloWeb.com](#) se muestra la manera de trabajar con bases de datos.

En nuestro ejemplo nos faltan por indicar las sentencias para conectar con MySQL y seleccionar la base de datos a utilizar. Serían unas parecidas a estas.

```
//conecto con la base de datos
$conn = mysql_connect("servidor","usuario","password");
mysql_select_db("nombre_bbdd",$conn);
```

Conclusión

Para acabar, ponemos a vuestra disposición la descarga del [código de este ejercicio](#), donde se puede ver todo el ejemplo completo y comentado. Esperamos que podáis aplicarlo a vuestros desarrollos.

Referencia: A posteriori, hemos recibido [un código para paginación realizado por un lector](#), que nos lo ha mandado para complementar este artículo. No podemos ofrecer los créditos porque hemos perdido su comunicación, así que si alguien lo reivindica colocaremos la autoría gustosamente. Asimismo, disponemos de otro artículo que trata la paginación de resultados en PHP con un enfoque un poco distinto. El artículo viene con el script explicado y todo lo necesario para ponerlo en marcha: [Paginación PHP y MySQL. Ejemplo 2](#). Recomendamos también la lectura del artículo [Paginador, script para paginar resultados con PHP y MySQL](#). Es un texto que presenta un interesante script en español y gratuito para realizar la paginación de resultados cómodamente.

Artículo por *Miguel Angel Alvarez*

Paginación con PHP y MySQL. Ejemplo 2

La paginación de resultados con PHP es uno de los talleres más interesantes y prácticos que se pueden hacer con esta tecnología. Casi todos los sitios web que implementan algún tipo de búsqueda deben en algún momento utilizar la paginación de resultados para no arrojar una cantidad desmesurada de datos en una única página. Como se ha visto desde siempre en los buscadores tipo Google, es mucho mejor presentar los resultados en varias páginas distintas y colocar en la parte de abajo una lista de las páginas de resultados que la búsqueda ha encontrado.

Para realizar este ejercicio se ha utilizado una base de datos MySQL y programación utilizando la tecnología PHP.

Podemos [ver el resultado que vamos a conseguir con este artículo](#) ahora y así tendremos más facilidad de identificar las distintas partes del código que vamos a comentar.

Nota: La paginación de resultados ya se ha [visto en un artículo anterior de PHP](#). En este caso ofrecemos otro ejemplo realizado por otro programador que seguro que nos sirve para hacernos una idea más global sobre como afrontar este tipo de problemas.

El código de este ejercicio, junto con algunas instrucciones para ponerlo en marcha, se puede [descargar en este enlace](#).

Poner en marcha el ejemplo

En el propio código de la paginación se encuentran las instrucciones para ponerlo en marcha y explicaciones sobre el funcionamiento del script. En este texto se preproducen algunas de las notas ofrecidas para ponerlo en marcha.

- 1.- Copiar el fichero busqueda.php en cualquier directorio del servidor web
- 2.- Crear una base de datos llamado 'tpv' (si es diferente, deberíais cambiar el código)
- 3.- Cambiar los datos de conexión si son diferentes.
- 4.- Cargar la siguiente tabla en la base de datos:

```
CREATE TABLE comercios (  
  co_id varchar(10) NOT NULL default "",  
  co_nombre varchar(30) NOT NULL default "",  
  co_pais varchar(30) NOT NULL default "",  
  UNIQUE KEY co_id (co_id)  
 ) TYPE=MyISAM;
```

5.- Realizar los siguientes insert

```
INSERT INTO comercios VALUES ('ESGR000002', 'PRUEBA 2', 'ESPAÑA');  
INSERT INTO comercios VALUES ('ESGR000001', 'PRUEBA 1', 'ESPAÑA');  
INSERT INTO comercios VALUES ('516', 'JUAN PEREZ', 'MEXICO');  
INSERT INTO comercios VALUES ('984', 'ANTONIO RODRIGUEZ', 'MEXICO');  
INSERT INTO comercios VALUES ('996', 'INDALECIO TRAVIANNI', 'ARGENTINA');  
INSERT INTO comercios VALUES ('975', 'ABELARDO SAINZ', 'PERU');  
INSERT INTO comercios VALUES ('111', 'JOSE TOLTACA', 'PERU');  
INSERT INTO comercios VALUES ('332', 'RAIMUNDO ALONSO', 'ARGENTINA');  
INSERT INTO comercios VALUES ('123', 'JUN JUANES', 'BRASIL');  
INSERT INTO comercios VALUES ('585', 'JOAO PAMINHIO', 'BRASIL');  
INSERT INTO comercios VALUES ('23432432', 'JOAQUIN DIAZ', 'ESPAÑA');
```

Por supuesto estos datos pueden ser los que queráis, solo sirve de ejemplo.

Una vez realizados todos estos pasos, ejecutar el fichero busqueda php... y a probar... :D

Nota: Al ser este un pequeño ejemplo, en el formulario de búsqueda he dejado que busque el valor introducido en todos los campos.

El código del script se puede ver a continuación:

```
<?  
  mysql_connect("localhost","root","");  
?>  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">  
<html>  
<head>  
<title>ejemplo de paginación de resultados</title>  
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">  
<meta http-equiv="Pragma" content="no-cache" />  
<style type="text/css">  
<!--  
a.p:link {  
  color: #0066FF;  
  text-decoration: none;  
}  
a.p:visited {  
  color: #0066FF;
```

```
text-decoration: none;
}
a.p:active {
color: #0066FF;
text-decoration: none;
}
a.p:hover {
color: #0066FF;
text-decoration: underline;
}
a.ord:link {
color: #000000;
text-decoration: none;
}
a.ord:visited {
color: #000000;
text-decoration: none;
}
a.ord:active {
color: #000000;
text-decoration: none;
}
a.ord:hover {
color: #000000;
text-decoration: underline;
}
-->
</style>
</head>
<body bgcolor="#FFFFFF">
<script language="JavaScript">
function muestra(queCosa)
{
alert(queCosa);
}
</script>
<div align="center"><strong><font color="#000000" size="2" face="Verdana, Arial, Helvetica, sans-serif">Paginación
de Resultados de una consulta SQL (sobre MySQL)<br><br><p><a
href="http://www.pclandia.com">www.pclandia.com</a></p> </font></strong> </div>
<hr noshade style="color:CC6666;height:1px">
<br>
<?
//inicializo el criterio y recibo cualquier cadena que se desee buscar
$criterio = "";
$txt_criterio = "";
if ($_GET["criterio"]!=""){
$txt_criterio = $_GET["criterio"];
$criterio = " where co_id like '%" . $txt_criterio . "%' or co_nombre like '%" . $txt_criterio . "%' or co_pais like '%" .
$txt_criterio . "%'";
}

$sql="SELECT * FROM tpv.comercios ".$criterio;
$res=mysql_query($sql);
$numeroRegistros=mysql_num_rows($res);
if($numeroRegistros<=0)
{
echo "<div align='center'>";
echo "<font face='verdana' size='-2'>No se encontraron resultados</font>";
echo "</div>";
}else{
//////////elementos para el orden
if(!isset($orden))
{
```

```
$orden="co_id";
}
//////////fin elementos de orden

//////////calculo de elementos necesarios para paginacion
//tamaño de la pagina
$tamPag=5;

//pagina actual si no esta definida y limites
if(!isset($_GET["pagina"]))
{
    $pagina=1;
    $inicio=1;
    $final=$tamPag;
}else{
    $pagina = $_GET["pagina"];
}
//calculo del limite inferior
$limitInf=($pagina-1)*$tamPag;

//calculo del numero de paginas
$numPags=ceil($numeroRegistros/$tamPag);
if(!isset($pagina))
{
    $pagina=1;
    $inicio=1;
    $final=$tamPag;
}else{
    $seccionActual=intval(($pagina-1)/$tamPag);
    $inicio=($seccionActual*$tamPag)+1;

    if($pagina<$numPags)
    {
        $final=$inicio+$tamPag-1;
    }else{
        $final=$numPags;
    }

    if ($final>$numPags){
        $final=$numPags;
    }
}

//////////fin de dicho calculo

//////////creacion de la consulta con limites
$sql="SELECT * FROM tpv.comercios ".$criterio." ORDER BY ".$orden.",co_id ASC LIMIT ".$limitInf.", ".$tamPag;
$res=mysql_query($sql);

//////////fin consulta con limites
echo "<div align='center'>";
echo "<font face='verdana' size='-2'>encontrados ".$numeroRegistros." resultados<br>";
echo "ordenados por <b>".$orden."</b>";
if(isset($txt_criterio)){
    echo "<br>Valor filtro: <b>".$txt_criterio."</b>";
}
echo "</font></div>";
echo "<table align='center' width='80%' border='0' cellspacing='1' cellpadding='0'>";
echo "<tr><td colspan='3'><hr noshade></td></tr>";
echo "<th bgcolor='#CCCCCC'><a class='ord' href='".$_SERVER["PHP_SELF"]."?pagina = ".
    $pagina."&orden=co_id&criterio=".$txt_criterio."'>Código</a></th>";
echo "<th bgcolor='#CCCCCC'><a class='ord' href='".$_SERVER["PHP_SELF"]."?pagina = ".
    $pagina."&orden=co_nombre&criterio=".$txt_criterio."'>Nombre</a></th>";
echo "<th bgcolor='#CCCCCC'><a class='ord' href='".$_SERVER["PHP_SELF"]."?pagina = ".
    $pagina."&orden=co_pais&criterio=".$txt_criterio."'>País</a></th>";
```

```
while($registro=mysql_fetch_array($res))
{
?>
  <!-- tabla de resultados -->
  <tr bgcolor="#CC6666" onMouseOver="this.style.backgroundColor='#FF9900';this.style.cursor='hand';"
onMouseOut="this.style.backgroundColor='#CC6666'";" onClick="javascript:muestra('<? echo "[".
$registro["co_id"]."] ".$registro["co_nombre"]." - ".$registro["co_pais"]; ?>');">
  <td><font size="2" face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFCC"><b><? echo
$registro["co_id"]; ?></b></font></td>
  <td><font size="2" face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFCC"><b><? echo
$registro["co_nombre"]; ?></b></font></td>
  <td><font size="2" face="Verdana, Arial, Helvetica, sans-serif" color="#FFFFCC"><b><? echo
$registro["co_pais"]; ?></b></font></td>
  </tr>
  <!-- fin tabla resultados -->
?>
} //fin while
echo "</table>";
} //fin if
//////////a partir de aqui viene la paginacion
?>
  <br>
  <table border="0" cellspacing="0" cellpadding="0" align="center">
  <tr><td align="center" valign="top">
  <?
  if($pagina>1)
  {
  echo "<a class='p' href='".$_SERVER["PHP_SELF"]."?pagina=".(($pagina-1)."&orden=".$orden."&criterio="
$txt_criterio.">";
  echo "<font face='verdana' size='-2'>anterior</font>";
  echo "</a> ";
  }

  for($i=$inicio;$i<=$final;$i++)
  {
  if($i==$pagina)
  {
  echo "<font face='verdana' size='-2'><b>".$i."</b> </font>";
  }else{
  echo "<a class='p' href='".$_SERVER["PHP_SELF"]."?pagina=".$i."&orden=".$orden."&criterio="
$txt_criterio.">";
  echo "<font face='verdana' size='-2'>".$i."</font></a> ";
  }
  }
  if($pagina<$numPags)
  {
  echo " <a class='p' href='".$_SERVER["PHP_SELF"]."?pagina=".(($pagina+1)."&orden=".$orden."&criterio="
$txt_criterio.">";
  echo "<font face='verdana' size='-2'>siguiente</font></a>";
  }
  }
  ////////////fin de la paginacion
?>
  </td></tr>
  </table>
  <hr noshade style="color:CC6666;height:1px">
  <div align="center"><font face="verdana" size="-2"><a class="p" href="index.php">::Inicio::</a></font></div>

  <form action="busqueda.php" method="get">
  Criterio de búsqueda:
  <input type="text" name="criterio" size="22" maxlength="150">
  <input type="submit" value="Buscar">
  </form>

</body>
</html>
```

```
<?
mysql_close();
?>
```

Se puede [descargar aquí](#).

*Artículo por **Francisco J. Matias***

Biblioteca ADOdb para PHP

Esta nota técnica la envío a propósito de que ayer leí un taller de acceso a BD con PHPLIB, el cual me pareció muy interesante y creo necesario ampliar las ventajas de PHP utilizando la biblioteca ADODB.

La forma de como instalar los archivos y de las especificaciones, las cuales son muy muy sencillas, se las dejo a ustedes, pueden encontrar el manual y la descarga en: <http://php.weblogs.com/>

El asunto aquí es que adodb maneja objetos, al igual que PHPLIB, pero lo fantástico es que solo bastará que hagamos una sola vez nuestro código sin importar si nuestro DBMS cambia, así nos evitaremos el tener que mudar el código también y no haremos un sitio para cada manejador, pues el tipo de BD lo especificamos en una variable y tan solo hará falta cambiar esa variable por el nombre del actual DBMS.

Aquí les pondré tan solo un pequeño ejemplo de manejo de datos con ADODB.

```
<?
include('adodb.inc.php'); //Este es un archivo con los datos que necesita el programa para reconocer adodb

$conn = &ADONewConnection('mysql'); Aquí el tipo de BD
$conn->PConnect('usuarios'); Conexión con la BD

if (!$DB)
    print "No se realizó la conexión";
else{
    $query= "select * from usuario";
    $query.= "where nombre like '%$nombre%'";

    $datos = $DB->Execute("$query"); /*Ejecutamos el query*/

    $numFilas = $datos->RecordCount(); /* contamos el total de registros de resultado */

    for ($i=0;$i<=$numFilas;$i++){

        /* Comenzamos a extraer de la BD los registros */
        $nombre= $datos->fields["nombre"];
        $appat = $datos->fields["apellidoP"];

        print " $i. Nombre:$nombre $appat";
    }

    $datos->Close(); //opcional
    $conn->Close(); //opcional
}
?>
```

Bueno, ahí el pequeño ejemplo, ya les toca a ustedes investigar y aprovechar la biblioteca a como les acomode, saludos!

*Artículo por **Roberto Bárcenas***

Enviar un formulario por mail con PHP

El método para enviar un formulario con PHP resulta muy similar al [utilizado en ASP](#). Varía tan sólo la sintaxis utilizada y las líneas de código que realizan el envío del correo electrónico.

Para empezar, sería muy útil que [aprendamos a enviar correos electrónicos con PHP](#), para lo que tenemos un artículo en DesarrolloWeb.

Esquema de funcionamiento

En este caso nos vamos a apoyar en la variable `$HTTP_POST_VARS`, que debería contener el formulario, para saber si hemos recibido o no datos desde un formulario. Dicha variable la utilizamos en un enunciado `if (! $HTTP_POST_VARS)`, que si pasa por el caso positivo -no había nada en `$HTTP_POST_VARS`- significa que no se ha recibido nada desde un formulario. En ese caso, muestro el formulario de contacto.

En caso contrario -sí que había algo en `$HTTP_POST_VARS`- quiere decir que estamos recibiendo datos por un formulario y en ese caso, recogemos los datos y componemos el cuerpo del mensaje.

Veamos el código de la página para crear el formulario, recibirlo y componer el cuerpo del mensaje que se va a enviar. En el mismo código podremos encontrar también la llamada a la función que envía el correo.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
  <title>Mándanos tus comentarios</title>
</head>

<body bgcolor="#cccc66" text="#003300" link="#006060" vlink="#006060">
<?
if (! $HTTP_POST_VARS){
?>
<form action="envia_form_php.php" method=post>
Nombre: <input type=text name="nombre" size=16>
<br>
Email: <input type=text name=email size=16>
<br>
Comentarios: <textarea name=coment cols=32 rows=6></textarea>
<br>
<input type=submit value="Enviar">
</form>
<?
}else{
  //Estoy recibiendo el formulario, compongo el cuerpo
  $cuerpo = "Formulario enviado\n";
  $cuerpo .= "Nombre: " . $HTTP_POST_VARS["nombre"] . "\n";
  $cuerpo .= "Email: " . $HTTP_POST_VARS["email"] . "\n";
```



```
$cuerpo .= "Comentarios: " . $HTTP_POST_VARS["coment"] . "\n";

//mando el correo...
mail("admin@tudominio.com","Formulario recibido",$cuerpo);

//doy las gracias por el envío
echo "Gracias por rellenar el formulario. Se ha enviado correctamente.";
}
?>
</body>
</html>
```

*Artículo por **Miguel Angel Alvarez***

Cálculo de los días de un mes en PHP

A continuación vamos a ver una sencilla manera de hacer una función en PHP que realiza el cálculo de los días de un mes. Es la función UltimoDia() que hemos utilizado ya en alguna ocasión para el manual de [calendario en PHP](#).

Nota: La función UltimoDia() hace un cálculo de cuál es el último día de un mes. La hemos utilizado con anterioridad (una versión distinta de la presente) para realizar el [Manual del calendario en PHP](#).

No es necesario hacer un ciclo repetitivo para la función UltimoDia(), es un poco más simple de resolver.

Los meses 1,3,5,7,8,10,12 siempre tienen 31 días, los meses 4,6,9,11 siempre tienen 30 días, el único problema es el mes de febrero dependiendo del año puede tener 28 o 29 días, pero ese cálculo tampoco es difícil.

Aquí envío el código para la función UltimoDia(), que ojalá les sirva...

```
function UltimoDia($anho,$mes){
    if (((fmod($anho,4)==0) and (fmod($anho,100)!=0)) or (fmod($anho,400)==0)) {
        $dias_febrero = 29;
    } else {
        $dias_febrero = 28;
    }
    switch($mes) {
        case 01: return 31; break;
        case 02: return $dias_febrero; break;
        case 03: return 31; break;
        case 04: return 30; break;
        case 05: return 31; break;
        case 06: return 30; break;
        case 07: return 31; break;
        case 08: return 31; break;
        case 09: return 30; break;
        case 10: return 31; break;
        case 11: return 30; break;
        case 12: return 31; break;
    }
}
```

*Artículo por **Héctor A. Pinto F***

Problemas con las fechas en timestamp Unix de PHP

Existen en PHP una serie de funciones muy útiles para realizar cálculos de fechas, como por ejemplo, saber si una fecha es válida, obtener un dato concreto de una fecha, como el día, la hora o el día del mes o la semana.

En teoría, cualquier cálculo con fechas básico se puede realizar con las funciones que provee el lenguaje, que se pueden observar en la documentación de PHP, concretamente en la URL <http://www.php.net/manual/es/ref.datetime.php>

El problema del Timestamp Unix

Muchas de las funciones de PHP se basan en el Timestamp de Unix que es el número de segundos transcurridos desde las 00:00:00 del 1 de enero de 1970 GMT. Por ejemplo la utilísima función `date()`, que recibe un string con el formato que se desea para la fecha y un timestamp de Unix para introducir la fecha que se pretende formatear.

El problema que se encontrará con los cálculos de fecha es que el timestamp empieza a contar desde 1970. ¿Qué pasa con las fechas anteriores? Además, el timestamp, como estructura de datos, tiene un tamaño fijo, es decir, una capacidad limitada hasta el año 2038. En concreto soporta los siguientes intervalos de años:

Windows: desde 1970 hasta 2038
Unix: desde 1901 hasta 2038

Este artículo pretende dar a conocer un mecanismo para poder trabajar con fechas fuera de este intervalo.

Librería `adodb_date_time_library`

Existen varios métodos de solventar este problema, pero nosotros vamos a hablar de uno muy sencillo y fácil de utilizar. Se trata de la librería `adodb_date_time`, que se puede descargar y utilizar gratuitamente en nuestras aplicaciones.

Esta librería soporta fechas desde el año 100 D.C. hasta billones de años en el futuro.

Se puede descargar desde http://phplens.com/phpeverywhere/adodb_date_library

Para utilizarla, simplemente debemos sustituir algunas de las funciones típicas de fechas de PHP por las que implementa la librería.

`getdate()` reemplazar por `adodb_getdate()`
`date()` reemplazar por `adodb_date()`
`gmdate()` reemplazar por `adodb_gmdate()`
`mktime()` reemplazar por `adodb_mktime()`
`gmmktime()` reemplazar por `adodb_gmmktime()`

Nosotros ya hemos probado la librería con éxito en alguna creación nuestra, así que la recomendamos encarecidamente cuando supongamos que las fechas con las que vamos a trabajar se salen del intervalo del timestamp Unix.

Para ampliar esta información será imprescindible acceder a la página de inicio de la librería y

enterarnos de todos los detalles sobre su uso.

http://phplens.com/phpeverywhere/adodb_date_library

Artículo por Miguel Angel Alvarez

Listas de elementos con colores alternos en PHP

Vamos a ver como crear una lista de elementos en una tabla creada por filas con colores alternos. Es decir, vamos a mostrar una lista de elementos mostrados en un conjunto de filas que pueden tener uno de dos colores y que se van alternado, primero uno y luego el otro. Para entender mejor el ejercicio sería interesante [ver el efecto buscado en una página aparte](#).

Esta disposición en filas de distintos colores puede dar un aspecto un poco más completo a nuestro diseño y servir de guía visual para que la tabla se entienda con un golpe de vista y se conozca mejor cuales son los datos que pertenecen a cada fila.

Para conseguir los colores alternos vamos a utilizar una variable que llevará la cuenta del número de filas y si la fila es impar mostrar un color distinto que el de la fila par. Para deducir si una fila es par o impar podemos dividir el número de fila entre 2 y si el resto de esa división es igual a cero, es que el número es par y si es distinto de cero, entonces el número es impar.

El operador %

Para obtener el resto de una división utilizamos el operador %, llamado operador de módulo. Insistimos: % devuelve el resto de una división entre los dos valores que se apliquen a la operación. Por ejemplo $2 \% 2$ valdrá 0, y la operación $6 \% 5$ dará 1.

Bucle con filas de colores alternos

Veamos ahora el código de un bucle en el que se utilicen filas con colores alternos. Utilizaremos la [tabla del ejercicio del libro de visitas](#), relatado en un [Manual práctico](#) de DesarrolloWeb.com.

El recorrido será parecido a otros que hemos realizado por los elementos del resultado de una consulta SQL contra una base de datos. La única tarea a realizar que no hayamos visto con anterioridad en los artículos de PHP publicados en este sitio, consiste en que se lleva una cuenta del número de filas y por cada fila se comprueba si el módulo del número de fila y el número 2 es cero o no. Si es cero se colocará un color y si no vale cero se colocará otro color, con lo que se creará esa alternancia.

El código es el siguiente.

```
<?
//creo la sentencia sql para atacar a la base de datos.
$sql = "SELECT * FROM librovisitas_php";
$sql .= " ORDER BY id_librovisitas_php limit 10";

//ejecuto la sentencia para extraer un conjunto de resultados
$resultid = mysql_query($sql,$conn);
```

```
//coloco la cabecera de la tabla
?>
<table width=500 align=center>
<tr bgcolor="bbbbbb" align=center>
  <td><b>Nombre</b></td>
  <td><b>Email</b></td>
  <td><b>Valoracion</b></td>
</tr>
<?
//creo e inicializo la variable para contar el número de filas
$num_fila = 0;

//bucle para mostrar los resultados
while ($damefila=mysql_fetch_object($resultid)){
  echo "<tr ";
  if ($num_fila%2==0)
    echo "bgcolor=#dddddd"; //si el resto de la división es 0 pongo un color
  else
    echo "bgcolor=#dddddff"; //si el resto de la división NO es 0 pongo otro color
  echo ">";
?>
  <td><?echo $damefila->nombre;?></td>
  <td><?echo $damefila->email;?></td>
  <td><?echo $damefila->valoracion;?></td>
</tr>
<?
  //aumentamos en uno el número de filas
  $num_fila++;
} //cierro el while
?>
</table>
```

Se puede [ver el resultado de ejecutar este script en una página aparte](#).

Artículo por **Miguel Angel Alvarez**

Template Power

Una de las claves de un desarrollo en la web limpio y fácil de mantener es separar el código de la presentación. En esa línea de ideas, las tecnologías modernas, como XML o CSS, ya están pensadas para hacer posible la separación de forma y contenido, cosa que resulta impensable con HTML.

El caso que nos ocupa, Template Power, consiste en un módulo que viene a hacer posible la separación del código PHP y el código HTML de una manera fácil y rápida. Dicho de otra manera, Template Power permite escribir el código HTML en unos archivos llamados plantillas y el código PHP en páginas que no tendrían necesidad de incluir una sola etiqueta HTML, sino que solamente llamarían a las distintas plantillas necesarias para hacer la página.

El producto es gratuito para uso personal y se puede utilizar en versiones de PHP 4.0.1 o superior. Su página de inicio, donde se puede descargar y acceder al manual es <http://templatepower.codocad.com>.

Método de trabajo

Vamos a describir aquí el método de trabajo usando Template Power, aunque no deseamos alargarnos mucho, ya que el propio producto tiene un completo manual que podemos encontrar incluso en castellano.

Un primer paso sería crear una plantilla, que contendrá código HTML, junto con algunas marcas propias del sistema Power Template. Entre estas podremos encontrar variables, que permiten sustituir el nombre de una variable por un valor, y los bloques, que permiten incluir otras plantillas o una lista de elementos con distintos datos.

La sintaxis de una plantilla sería algo parecido a esto.

```
<div align=center>{nombrecategoria}</div>
<hr align="center">
<table width="300" cellpadding="2" cellspacing="2" align=center>
<!-- START BLOCK : productos-->
<tr><td>
<b><a href="ficha_producto.php?id={idproducto}">{nombreproducto}</a></b><br>
{descripcionproducto}
</td></tr>
<!-- END BLOCK : productos -->
</table>
```

En la plantilla del ejemplo podríamos estar visualizando una categoría de productos de una tienda virtual. En la categoría de productos habría un nombre y una lista de productos asociados a la categoría, que podrían ser uno o varios.

Las variables, serían las introducidas entre llaves, por ejemplo {nombrecategoria} y se podrían sustituir por un dato.

Los bloques estarían definidos por una línea de código como la que sigue:

```
<!-- START BLOCK : productos-->
```

Con su correspondiente línea de cierre de bloque, situada un poco más abajo. En este caso el bloque sirve para incluir una repetición de productos dentro de una categoría.

Como hemos podido comprobar, en las plantillas no se incluye ni una sola línea PHP. Ahora veremos cómo se programarían las páginas PHP para hacer uso de una plantilla como la que hemos visto.

```
<?
require "./clases/TemplatePower.inc.php";

$t = new TemplatePower("plantillas/cat_productos.tpl");
$t->prepare();

$t->assign(array(
nombrecategoria => 'Material de cocina',
));

$$sql = "select * from productos where nombrecategoria='Material de cocina'"
$productos = mysql_query($$sql);
while ($fila=mysql_fetch_object($productos)){
$t->newBlock("productos");
$t->assign(array(
nombreproducto => $fila->nombreproducto,
idproducto => $fila->idproducto,
descripcionproducto => $fila->decripproducto
```

```
));  
}  
$t->printToScreen();  
?>
```

Nota: Estos scripts están pensados únicamente para ilustrar el funcionamiento de Template Power. Han sido extractados de una aplicación que he realizado recientemente con este sistema, por lo que pueden estar incompletos o incorrectos sintacticamente.

El uso de plantillas desde el código PHP se realiza a través de una clase llamada `TemplatePower`, que implementa una serie de métodos útiles para realizar los trabajos necesarios con la plantilla de una manera rápida.

Lo primero es incluir el código de la clase, que se puede descargar desde el sitio de `TemplatePower` de manera gratuita. Más tarde ya estamos en condiciones de instanciar el objeto plantilla, con el constructor de la clase, indicando por parámetro el archivo desde donde se va a tomar el código HTML.

El método `prepare()` sirve para que se analice la plantilla. Después de ejecutarlo ya podremos asignar valores a las variables o bloques dentro de las plantillas.

Con el método `assign`, que recibe un array con los pares Nombre de variable => Valor de la variable, podemos asignar un valor a cada una de las variables de la plantilla.

En el código PHP podremos hacer consultas a la base de datos para extraer los valores e las variables, como es el caso del ejemplo, en el que se extraen todos los productos de una categoría. Con un bucle `while`, que recorre todos los productos de la base de datos, podemos crear un bloque para cada uno de los productos obtenidos. Para ello está el método `newBlock()`, que recibe el nombre del bloque que se debe generar. Dentro de cada bloque podemos asignar las variables que contiene la plantilla dentro de dicho bloque, también con el método `assign()`.

Finalmente, cuando se han asignado todas las variables y bloques a la plantilla, se debe invocar el método `printToScreen()` para escribir el contenido generado por la plantilla y los valores dentro de la página web.

Otras posibilidades

La librería `Template Power` ha evolucionado bastante con el tiempo. Están por la versión 3.0, que soporta casi cualquier cosa que podamos necesitar en el trabajo de separar el código HTML del PHP.

Incluye soporte para plantillas anidadas, posibilidad de incluir códigos PHP, que se ejecutarían también. Asignación global de variables y asignación en una única sentencia de varias variables.

También podremos salvar en disco y utilizar una plantilla ya analizada, incluso guardar plantillas en la base de datos.

Conclusión

`Template Power` es un producto muy interesante para separar el código PHP del código HTML. Como es gratuito lo podemos utilizar o probar sin problemas ni desembolsos, aunque si deseamos utilizarlo para un sistema comercial estamos obligados a pagar una tasa, que es

realmente asequible (3 US\$).

En definitiva, es un sistema que puede subir mucho la calidad de nuestros códigos e incluso aumentar nuestra productividad. Para trabajar con Template Power necesitaremos acostumbrarnos un poco al mecanismo planteado y conocer bien las posibilidades del producto. Todo ello será muy fácil porque en la página de inicio de Template Power (<http://templatepower.codocad.com>) tenemos unos buenos ejemplos y un manual en varias lenguas.

Artículo por Miguel Angel Alvarez

Convertir fechas entre MySQL y castellano, en PHP

Las fechas son uno de esos típicos asuntos que pueden hacer que nos rompamos la cabeza a la hora de programar una página. Razón de ello es que tienen distintos formatos dependiendo del país, del lenguaje de programación o de la base de datos que estemos utilizando.

Cuando utilizamos la tecnología PHP solemos trabajar con la base de datos MySQL. En estos dos sistemas los formatos de fechas cambian sensiblemente, así que será muy interesante conocer una manera rápida de pasar de un formato de fecha a otro, dependiendo de dónde vamos a utilizar esa fecha. Pues, si trabajamos con MySQL deberemos expresar la fecha de una manera distinta a la que lo haríamos a la hora de mostrarla en la página para que la entienda fácilmente un lector hispano.

En muchos casos, debemos vérnoslas entre dos tipos de formatos distintos, aunque podría ser peor. Por ejemplo, si la página estuviese en varios idiomas, sería importante escribir correctamente las fechas en cada uno de los idiomas.

Dejando temas relacionados con el idioma aparte -concentrándonos tan sólo en el Español-, en nuestras páginas programadas en PHP y con base de datos MySQL, tendremos que trabajar con dos formatos. Por un lado tenemos las fechas en castellano, que tienen el formato dd/mm/aaaa y por otro lado tenemos el formato de MySQL, que tiene la sintaxis aaaa-mm-dd.

Lo más cómodo, tal como vemos nosotros este problema, es crear un par de funciones que conviertan las fechas de un formato a otro. Habrá una función que convertirá la fecha de MySQL a Castellano y otra que lo convierta de Castellano a MySQL.

```
////////////////////////////////////
//Convierte fecha de mysql a normal
////////////////////////////////////
function cambiaf_a_normal($fecha){
    ereg( "([0-9]{2,4})-([0-9]{1,2})-([0-9]{1,2})", $fecha, $mifecha);
    $lafecha=$mifecha[3]."/".$mifecha[2]."/".$mifecha[1];
    return $lafecha;
}

////////////////////////////////////
//Convierte fecha de normal a mysql
////////////////////////////////////

function cambiaf_a_mysql($fecha){
    ereg( "([0-9]{1,2})/([0-9]{1,2})/([0-9]{2,4})", $fecha, $mifecha);
```

```
$lafecha=$mifecha[3]."-".$mifecha[2]."-".$mifecha[1];  
return $lafecha;  
}
```

Las funciones utilizan expresiones regulares que no hemos visto todavía, así que no vamos a tratar de explicar cómo funcionan, sino que explicaremos cómo utilizarlas.

Mostrar en la página una fecha en castellano

Si tenemos una fecha en formato MySQL y deseamos colocarla en una página haremos algo como sigue.

Suponemos que la fecha está extrayéndose a través de una consulta a la base de datos y la tenemos en una variable llamada `$fila->fecha`. Además, colocamos la fecha en un campo de formulario.

```
<input type="text" name="fecha" value="<?echo cambiaf_a_normal($fila->fecha);?>">
```

Colocar en la base de datos una fecha en formato MySQL

Cuando el usuario nos manda una fecha, por ejemplo, a través de un formulario con un campo como el que acabamos de ver, lógicamente, escribirá la fecha en castellano. Pero nosotros deseamos guardarla en una base de datos en un formato distinto, así que habremos de convertirla.

Suponemos que tenemos la fecha en una variable llamada `$fecha` y que está en formato castellano. Además, queremos colocarla en una sentencia SQL que deseamos ejecutar en la base de datos para insertar un registro que contiene, entre otros datos, la fecha que el usuario ha escrito.

```
mysql_query ("insert into documento (titulo_documento, fecha_documento, cuerpo_documento) values  
('$titulo_documento', ' . cambiaf_a_mysql($fecha) . ', '$cuerpo_documento')");
```

Artículo por Miguel Angel Alvarez

Paginador PHP usando pear y templates

A continuación podemos ver un script para paginar datos utilizando pear y templates bastante fácil de entender. Lo he documentado todo lo que he podido en el propio código del script.

El uso del script es totalmente libre y gratuito. Espero que les sirva.

El script combina el uso de templates y la paginación de datos utilizando pear. Para que funcione deberás asegurarte de tener las siguientes librerías:

```
require_once('DB.php');  
require_once('DB/Pager.php');  
require_once('HTML/ITX.php');
```

Referencia: Tenemos más scripts para paginar resultados en PHP, englobados dentro del [Taller de PHP](#):

[Paginación de resultados con PHP y MySQL](#) [Paginación PHP y MySQL. Ejemplo 2.](#)

Para probar el script necesitas tener una carpeta Templates con el archivo template.html que esta en el archivo .zip que puedes [descargar en este enlace](#).

La estructura de la tabla para que corra este script es:

Nombre de la Tabla: "Invitaciones"

Campos:

ID_Invitacion
NombreInvitado
email

No se os olvide colocar el nombre de la base de datos donde dice "nombre_de_la_base".

Si encuentras mejoras al script, por favor infórmalos en diems2@yahoo.com.ar

Puedes [descargar el script aquí](#).

*Artículo por **Diego Villar***

Contador mejorado para páginas PHP

Dadas mis necesidades he tenido que modificar el script del [contador simple para páginas PHP](#), y me gustaría compartirlo con todos vosotros. La diferencia está en que ahora almacena tres datos en el archivo que hace de contador:

- 0) el número de mes de la última visita
- 1) el número de visitas del mes
- 2) el número de visitas totales

El script queda como sigue:

```
function interface_contador(){  
  
    $archivo = "contador.txt";  
    $info = array();  
  
    //comprobar si existe el archivo  
    if (file_exists($archivo)){  
        // abrir archivo de texto y introducir los datos en el array $info  
        $fp = fopen($archivo,"r");  
        $contador = fgets($fp, 26);  
        $info = explode(" ", $contador);  
        fclose($fp);  
  
        // poner nombre a cada dato  
        $mes_actual = date("m");  
        $mes_ultimo = $info[0];  
        $visitas_mes = $info[1];  
        $visitas_totales = $info[2];  
    }else{
```

```
// inicializar valores
$mes_actual = date("m");
$mes_ultimo = "0";
$visitas_mes = 0;
$visitas_totales = 0;
}

// incrementar las visitas del mes según si estamos en el mismo
// mes o no que el de la última visita, o ponerlas a cero
if ($mes_actual==$mes_ultimo){
    $visitas_mes++;
}else{
    $visitas_mes=1;
}
$visitas_totales++;

// reconstruir el array con los nuevos valores
$info[0] = $mes_actual;
$info[1] = $visitas_mes;
$info[2] = $visitas_totales;

// grabar los valores en el archivo de nuevo
$info_nueva = implode(" ", $info);
$fp = fopen($archivo, "w+");
fwrite($fp, $info_nueva, 26);
fclose($fp);

// devolver el array
return $info;
}
```

Este código devuelve un array cuando es llamado, con 3 elementos (el mes actual, visitas del mes, visitas totales) que cada uno puede utilizar como quiera.

Por cierto, me olvidaba decir que he añadido una cláusula condicional que verifica si existe el archivo \$contador en el directorio de la página, previamente a ser leído. Lo he hecho para evitar que diera error la primera vez que se ejecutaba el script. De esta forma, ya no hace falta preocuparse por poner en el servidor un archivo \$contador a cero antes de ejecutar el código por primera vez.

Se puede [ver el ejemplo en marcha en esta página](#). También se puede [descargar el código de la página del ejemplo](#), con la función y una sencilla muestra de cómo utilizarla.

*Artículo por **Sergio Rodriguez Rios***

Bucle para recibir todas las variables por POST en PHP

Vamos a ver una manera muy rápida de recibir todas las variables de un formulario, enviado por post, en el lenguaje PHP. Os aseguro que es una pequeña porción de código que os ahorrará escribir un montón de líneas de código.

Quién no se ha visto alguna vez en la tediosa tarea de recibir un montón de datos de un formulario, asignando una por una todas las variables en PHP? Eso se hacía con líneas como ésta:

```
$nombre = $_POST["nombre"];  
$edad = $_POST["edad"];  
$ciudad = $_POST["ciudad"];  
....
```

Si el formulario tuviera 10 elementos no sería muy pesado escribir las 10 líneas de código, pero si fueran 50 o 100 la cosa sería mucho menos agradable. El código que vamos a ver ahora nos solucionará la vida en esos casos.

```
foreach($_POST as $nombre_campo => $valor){  
    $asignacion = "\$" . $nombre_campo . "=" . $valor . " ";  
    eval($asignacion);  
}
```

Se realiza un bucle foreach que va recorriendo cada uno de los elementos del post. En cada iteración, se van accediendo a todos los elementos del post y se guarda en \$ nombre_campo el nombre del campo recibido por el formulario y en \$valor, el valor que se había introducido en el formulario.

Todo lo anterior se deduce de la primera línea. En las siguientes se compone en cada iteración, cada una de las asignaciones que deberíamos haber escrito manualmente. Es decir, en la variable asignación guardaremos una línea de código PHP que realiza la declaración de la variable de formulario dentro de PHP y su inicialización con el valor que se hubiera escrito.

En la siguiente línea, donde está la función eval(), se ejecuta la sentencia generada en el anterior paso. La función eval() de PHP ejecuta el contenido de una cadena de caracteres como si fuera una sentencia PHP. (Podemos ver la documentación de la función eval() en la página de PHP <http://es.php.net/manual/es/function.eval.php>)

Esperamos que os haya interesado este minúsculo, pero útil, código PHP.

Nota: Se puede consultar otro artículo que hace un recorrido similar, en ese caso para [declarar como variables todos los datos recibidos en una fila de un recordset](#). Así mismo tenemos un artículo que nos muestra cómo [recibir de forma segura los datos de un formulario](#).

Artículo por **Miguel Angel Alvarez**

Contador PHP con imágenes

Aquí les dejo el código PHP de un contador mejorado con la inclusión de imágenes para generar el número de visitas en lugar de utilizar texto como se venía realizando en el artículo [contador simple de páginas PHP](#).

Para poner en marcha el ejemplo sólo hay que crear los números del 0 al 9 en formato imagen. Es preferible que los números sean menores a estas propiedades: width="17" height="28"

Otra cosa, parte del código fue tomado de [este contador](#). Lo que he incluido yo es lo que hace que llame a las imágenes, que deben ser nombradas "0.gif", "1.gif", .., "9.gif".

```
<html>  
<head>
```

```
<title>Contador PHP con imágenes</title>
</head>

<body>
<?php
$archivo = "contador.txt";
$contador = 0;

$fp = fopen($archivo,"r");
$contador = fgets($fp, 26);
fclose($fp);

++$contador;

$fp = fopen($archivo,"w+");
fwrite($fp, $contador, 26);
fclose($fp);
?>

<table width="102" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
<?php

/* Arreglo de 0-9 nombre de los archivos gifs*/
$numero[0]="0.gif";
$numero[1]="1.gif";
$numero[2]="2.gif";
$numero[3]="3.gif";
$numero[4]="4.gif";
$numero[5]="5.gif";
$numero[6]="6.gif";
$numero[7]="7.gif";
$numero[8]="8.gif";
$numero[9]="9.gif";

/*Se crea variable para que contenga la longitud de la cadena*/
/*es a partir de ahí donde se sabe que mostrará el contador en GIFS*/

$longitud = strlen ($contador);

/* Bucles para mostrar los números*/
$hasta = 6-$longitud;

For ($celda = 1;$celda <= $hasta;$celda++)
{
    echo "<td width=\"17\" height=\"28\" valign=\"top\"><div align=\"center\"><img
src=\"\$numero[0]\"></div></td>";
}
$hasta = $longitud-1;
For ($celda = 0;$celda <= $hasta;$celda++)
{
    $num = substr ($contador, $celda, 1);
    echo "<td width=\"17\" height=\"28\" valign=\"top\"><div align=\"center\"><img
src=\"\$numero[$num]\"></div></td>";
}

?>

</tr>
</table>
</div>
</body>
</html>
```

Artículo por **Josue Gutiérrez Olivares**

Variables variables PHP

Escribo esta ayuda técnica sobre "variables variables" porque me parece un tema de no muy fácil comprensión pero de gran ayuda en cuanto a desarrollo con PHP, espero que a todos pueda ayudarnos a comprender un poco mejor si utilizo algún ejemplo práctico que en algunos momentos me ha ayudado a resolver ciertos problemas.

La explicación más sencilla y fácil de comprender se me ocurre sea la traducción al castellano de php.net, así que la pondré tal cual y después aplicaré la práctica procurando expresarme mas claramente.

Aquí comienza la explicación del sitio

A veces es conveniente tener nombres de variables variables. Dicho de otro modo, son nombres de variables que se pueden establecer y usar dinámicamente. Una variable normal se establece con una sentencia como:

```
<?php
$a = "hello";
?>
```

Una variable variable toma el valor de una variable y lo trata como el nombre de una variable. En el ejemplo anterior, hello, se puede usar como el nombre de una variable utilizando dos signos de dolar. p.ej.

```
<?php
$$a = "world";
?>
```

En este momento se han definido y almacenado dos variables en el árbol de símbolos de PHP: \$a, que contiene "hello", y \$hello, que contiene "world". Es más, esta sentencia:

```
<?php
echo "$a ${$a}";
?>
```

produce el mismo resultado que:

```
<?php
echo "$a $hello";
?>
```

Ambas producen el resultado: hello world.

Y aquí termina la explicación

El URL es <http://mx.php.net/manual/es/language.variables.variable.php>

Según creo, si hemos tenido algún contacto con la programación, habremos comprendido

bastante bien la explicación, pero ¿ y como nos puede llegar a servir? ¿ en que ocasiones me puede ayudar?. Pues en particular lo he contemplado en situaciones donde en un archivo tengo que poner varios checkbox o radio buttons y en el siguiente tengo que examinar una a una esas variables; y como no deseo realizarlo de esta forma, los termino metiendo en un for, pero si he metido los nombres de los checkbox de la siguiente manera...

```
<form name='form1' method='post' action='guarda_preferencias.php'>
<table width='700' border='0'>
<tr>
<td><span class='style4'>Para Empezar </span></td>
</tr>
<tr>
<td><blockquote class='style5'><strong>
<input type='checkbox' name='pref_1' value='bienvenida'>
  Bienvenida</strong></blockquote></td> </tr>
<tr>
<td><blockquote class='style5'><strong>
<input type='checkbox' name='pref_2' value='temario'>
  Temario </strong></blockquote></td>
</tr>
```

y así sucesivamente hasta pref_n; en mi archivo guarda_preferencias.php, después del for que empezaría la revisión de la transferencia de cada checkbox, existiría el ligero inconveniente de tener que revisar \$pref_\$i, lo cual es un tremendo problema porque \$pref_ no existe. Lo anterior yo lo he resuelto de la siguiente manera: hago uso de variables variables que me permitirán obtener el valor que realmente quiero, que es pref_1, pref_2....pref_n .

```
<?
for ($i=1;$i<=n;$i++){ //n es por supuesto el total de elementos checkbox
que hayamos creado
$nombre = "pref_$i";
$prefer = $$nombre;
if ($prefer != "") print "$nombre si trae algo, que es $prefer";
?>
```

Con lo cual \$nombre hago que sea pref_ más el valor de \$i, que irá aumentando de uno en uno cada vez que entre al for, y despues \$prefer me dará el valor real de pref_1, pref_2, pref_3 y así hasta n, pudiendo yo evaluar cada checkbox uno a uno sin problemas, en este caso yo solo estoy evaluando que no esté vacío e imprimo el nombre y el valor.

Espero que haya quedado un poquito mas claro para todos, saludos!!

*Artículo por **Roberto Bárcenas***

Cambio del tipo de las variables en PHP

PHP no requiere que indiquemos el tipo que va a contener una variable, sino que lo deduce del valor que asignemos a la variable. Asimismo, se encarga de actualizar automáticamente el tipo de la variable cada vez que le asignamos un nuevo valor.

Por ello, para cambiar el tipo de una variable simplemente le asignamos un valor con un nuevo tipo.

Nota: Se excluyen en este caso el cambio de variables a tipo Array porque la sintaxis puede resultar ambigua al expresar ese código, es decir, puede darse el caso de que una línea de código pueda significar dos cosas.

```
$a = "1";  
//$a es una cadena  
$a[0] = "f";  
//¿Estamos editando el índice de la cadena o forzando a array?
```

Forzado

En cualquier caso, podemos forzar una variable para que cambie de tipo con la función setType().

```
setType($variable,"nuevo_tipo");
```

la función setType() actualiza el tipo de \$variable a "nuevo_tipo" y devuelve un booleano indicando si hubo éxito o no en la conversión.

Entre "nuevo_tipo" tenemos:

- "integer"
- "double"
- "string"
- "array"
- "object"

También podemos hacer que una variable se comporte como un tipo determinado forzándola, de la misma manera a como se hace en el lenguaje C.

```
$variable = "23";  
$variable = (int) $variable;
```

Los forzados permitidos son:

- (int), (integer) - fuerza a entero (integer)
- (real), (double), (float) - fuerza a doble (double)
- (string) - fuerza a cadena (string)
- (array) - fuerza a array (array)
- (object) - fuerza a objeto (object)

Artículo por [Miguel Angel Alvarez](#)

Recibir el último ID de una inserción con PHP y MySQL

Creamos este artículo para responder una duda que nos han preguntado alguna vez en nuestro correo. Se trata sobre conseguir el último identificador de un registro insertado en la base de datos, programando con PHP y con la base de datos MySQL.

Realmente es una duda muy sencilla de solucionar, gracias al rico juego de funciones de PHP

para trabajar con bases de datos MySQL. Existe una función que nos devuelve directamente el identificador de la última inserción, utilizando la conexión a la base de datos que se le pase por parámetro, o la última conexión utilizada en caso de que no se le indique parámetro.

```
mysql_insert_id($connectid)
```

Ejemplo de utilización

Veamos un ejemplo de la utilización de esta función, en el que se realizan las acciones necesarias para insertar un elemento en la base de datos. En este ejemplo utilizaremos una base de datos llamada "test" y la tabla "cliente".

```
//conecto con la base de datos
$connectid = mysql_connect("localhost","root","");

//selecciono la base de datos a utilizar
mysql_select_db("test",$connectid);

//Sentencia de inserción del elemento
$sql = "INSERT INTO cliente (nombre_cliente, cif, direccion, email) VALUES ('xxx Nombre empresa', 'B3331113', 'C/
Corona 2', 'eepe@rell.com')";

//lo inserto en la base de datos
if (mysql_query($sql,$connectid)){

    //recibo el último id
    $ultimo_id = mysql_insert_id($connectid);
    echo $ultimo_id;
}else{
    echo "La inserción no se realizó";
}
```

Primero se conecta con MySQL y se selecciona la base de datos a utilizar. Luego se genera una sentencia SQL de inserción en una supuesta tabla de clientes. Si la ejecución de esta sentencia es correcta se recibe el último ID con la función insert_id(). Si la ejecución de la sentencia falla, muestra un mensaje de error.

Fijémonos que insert_id() recibe el identificador de la conexión, devuelto por mysql_connect() y no el identificador de resultado de la consulta, que devuelve mysql_query() cuando se ejecutan sentencias de selección en la base de datos.

*Artículo por **Miguel Angel Alvarez***

Variables globales de un recordset

Llegado el caso podría ser interesante que los datos devueltos en una consulta pasen a ser variables globales de PHP para una posterior manipulación más práctica.

Si de ahora en adelante surge la necesidad de realizar esto, no hay más que ejecutar el script siguiente:

```
function recibe_fila($fila){
```



```
foreach($fila as $nombre_campo => $valor){  
  
    if (gettype($nombre_campo)!="integer"){  
        $asignacion = "\$GLOBALS[\" . $nombre_campo . "\"]=";  
  
        if((is_integer($valor) || (is_float($valor))  
            $asignacion.= $valor . ",";";  
            else $asignacion.= "" . $valor . ",";";  
        eval($asignacion);  
        //echo $asignacion . "<br>";  
    }  
  
}
```

Recibe como parámetro toda la fila devuelta por la consulta. Ya dentro de la función, por cada atributo contenido en la fila, averigua el tipo de cada clave del array (ya que el bucle foreach siempre asigna 2 claves por cada una - una numérica, que es el índice del nombre del campo en la tabla, y la otra alfanumérica, que es el nombre del campo), y, por cada clave alfanumérica, averigua el tipo del valor a asignar a \$GLOBALS["atributo"] (si es numérico no va entre comillas, y si es una cadena sí las lleva). Lo último a realizar es pasarle a la función eval() la sentencia de asignación, para así interpretarla como sentencia PHP.

Artículo por Miguel Angel Alvarez

Formateo de una cadena

¿Cuántas veces has estado temeroso de que una cadena que vas a introducir en un campo de una tabla no tenga el formato correcto, y la has revisado, y vuelto a revisar... y luego descubres que ha dado un error al introducirla porque había caracteres no válidos?

Pues ya puedes respirar con alivio porque con el script que te proporcionamos tus dolores de cabeza tienen los días contados:

```
function str_asegurar($cadena){  
    //elimino etiquetas HTML y PHP  
    $cadena = strip_tags($cadena);  
    //elimino el caracter comilla, que puede estropear una sentencia  
    $cadena = str_replace("'", "", $cadena);  
  
    return $cadena;  
}
```

Después de haber ejecutado el script, sólo queda realizar la sentencia de inserción.

Artículo por Miguel Angel Alvarez

PHP isam, herramienta Web para Archivos Planos

Los Archivos Planos parecen olvidados desde hace tiempo, aunque en muchas instalaciones se siguen usando y se tienen que convertir para otras aplicaciones, p.e. una Base de Datos, para poder obtener resultados aceptables. Esta operación altera la estructura de los datos quedando fuera de nuestro alcance a no ser volvamos a hacer una conversión después de haberlos manipulado.

Para evitar estos trasvases y sin abandonar la estructura de nuestros Archivos Planos, tenemos un planteamiento que no es nuevo y es digno de tener en cuenta, se trata de conservar la estructura de los datos y montar aislados de ellos los recursos para su explotación, de esta forma conservamos la información en su formato original, y podemos acceder a ella sin depender de ningún software en concreto, es más, podemos hacer nuestros propios programas porque seguimos teniendo acceso a nuestros datos.

Cabe preguntarse si merece la pena invertir tiempo en este tema sobre todo aquellas instalaciones que entregan sus datos en Archivos Planos, p.e. al hacer Pedidos a un Proveedor que para la entrada a su sistema con MySQL o similar exige en el trasiego de datos un determinada estructura de Archivos Planos, con lo que tendremos que transformar nuestros datos para enviar Pedidos y otra vez pero a la inversa cuando recibimos Albaranes y Facturas del Proveedor en cuestión. Sería cómodo trabajar directamente con los Archivos Planos p.e. con una Aplicación que nos permitiese tener la Gestión de nuestros Clientes y a la vez comunicarnos sin ningún problema con nuestro Proveedor, teniendo un control sobre nuestros datos y software sin tanta diversidad de Herramientas.

La idea no es difícil de llevar a la práctica, podemos empezar creando una tabla que haga de MAPA para el manejo de Archivos con su Nombre, Campos, longitud y características, para lo que necesitamos unos programillas para mantenerla y unas funciones para manejar los Archivos según estas especificaciones, funciones que usaremos para reducir considerablemente nuestro programas evitando errores y tiempo de depuración.

Ahora necesitaremos presentar estos Archivos en la Web, para lo que ampliamos nuestro MAPA incluyendo Rejillas de presentación indicando su nombre y el del Archivo que vamos a ver con sus Campos y algún parámetro más, en base a esto haremos otra serie de funciones para explotar esta estructura, y siguiendo, podemos diseñar la presentación de Formularios de forma que invocando su nombre y número de registro podamos ver y modificar en pantalla los datos que queramos.

Con esto, disponemos de un artilugio que nos facilita la programación, pero nos queda algo tan importante como es el manejo en si de los datos, esto es, su ordenación y selección. Para esto echamos mano de ISAM estableciendo los Indices que necesitamos para nuestros Archivos Planos aislados de ellos en un Archivo aparte. El manejo de datos con ISAM es cómodo y rápido, y como el programa de ISAM lo hemos hecho a nuestro medida, tendrá además de los clásicos comandos de Búsqueda, Inserción, Eliminación, etc. otros más acordes a nuestras necesidades como la Selección de listas que nos devuelve los registros que cumplan determinadas condiciones, p.e. que tengan una parte inicial de la Clave igual al argumento de búsqueda como en las líneas de un Pedido que llevarán todas el Número de Pedido al que pertenecen.

Está claro que este trabajo PHPisam no aporta una Base de datos, pero las prestaciones obtenidas permiten trabajar con una soltura parecida, además el enfoque principal es la presentación de datos en la Web permitiendo la selección de datos y el envío a la Web en

Formularios de ficha y Rejillas de líneas de registros usando las funciones pertinentes, con lo que podemos hacer programas muy pequeños y de bastante utilidad como mi propio sitio desarrollado con esta herramienta, donde e-commerce y su Administrador constan de 20 programas y solo 2 exceden de 40 líneas de código.

*Artículo por **Serafin Vazquez***

Instalación de PhpAdsNew o OpenAds

PhpAdsNew es un servidor de banners de código abierto que tiene una potencia y posibilidades dignos de una aplicación comercial, pero que se distribuye gratuitamente. Este servidor de banners ahora se llama OpenAds. Le han cambiado el nombre, pero es exactamente el mismo producto con todas las posibilidades.

Es capaz de manejar distintos clientes, campañas y creatividades en distintos formatos, guardando toda la actividad del servidor en base de datos y proporcionando tanto al cliente como al administrador estadísticas detalladas de impresiones, clics, CTR, etc.

Instalación

Es un proceso extremadamente sencillo. Cometo a continuación los pasos necesarios para instalarlo en nuestro servidor.

Acceder a la página de PhpAdsNew, en la dirección: <http://phpadsnew.com>

Descargar la última versión disponible. Para ello hay un enlace en la parte izquierda de la página de inicio de PhpAdsNew que nos llevará a la zona de descargas.

En la página de soporte (Support) de PhpAdsNew se pueden descargar dos manuales en PDF muy completos para aprender a utilizar el sistema. "Administrador guide" es una ayuda para aprender a instalar y configurar PhpAdsNew y "User guide" es una guía para aprender a gestionar el servidor de banners, configurar campañas, banners, entender las estadísticas, etc.

Nota: PhpAdsNew tiene un instalador que puede hacer la mayor parte del trabajo automáticamente. Se deben seguir las instrucciones de instalación sin editar manualmente los datos de configuración, porque de no hacerlo así, puede dar lugar a errores futuros en el funcionamiento.

Según las instrucciones de instalación, se deben descomprimir los archivos en el servidor. Como eso no está permitido en la mayoría de los alojamientos, se tienen que descomprimir en el ordenador local y luego subir por FTP los archivos descomprimidos a algún directorio del servidor.

Para continuar, debemos modificar los atributos del archivo de configuración, llamado "config.inc.php", para permitir su escritura. Esto es necesario para que el sistema de instalación pueda guardar la configuración seleccionada en el archivo. Una vez terminado el proceso de instalación es importante acordarse de volver a protegerlo contra escritura para evitar posibles errores de seguridad.

Con un programa de FTP podemos asignar los atributos para permitir la escritura. El proceso puede variar de un programa a otro, pero será algo como pedir las propiedades del archivo y

habilitar el permiso de escritura.

Una vez llegado este punto se puede acceder a la página de nuestro servidor donde está el asistente de instalación de PhpAdsNew, que solicitará una serie de datos para configurar el programa.

<http://midominio.com/phpadsnew/admin/index.php>

Cuando procedemos con la ejecución del asistente, se realiza en un primer paso un chequeo de las características del servidor, para ver si todos los requerimientos están disponibles. Si este proceso falla nos indican qué debemos hacer para solucionarlo. Puede ser algo como un valor no permitido en una configuración de Apache o de PHP, o errores más críticos como no disponer de una versión de PHP 4.0.3 o superior.

El sistema también comprueba que todos los archivos subidos estén correctamente. Si algo de esto falla también nos mostrará un mensaje de advertencia para que lo solucionemos.

Si todo ha ido bien, tenemos acceso a la página donde configuraremos la cuenta de acceso del administrador. Nos requerirá sus datos, correo, contraseña, idioma, etc.

En la siguiente página nos solicitará los datos de conexión a la base de datos MySQL, como el nombre del servidor, usuario, clave y nombre de la base de datos. Estos datos los debemos conocer nosotros. En caso contrario, sería necesario preguntar al proveedor de hosting de nuestra web.

Una vez introducida la información, si todo ha ido bien, se accede a la página de confirmación y se finaliza el asistente. Con esto acaba el primer proceso de la instalación, aunque aun podemos hacer cosas para personalizar más aun el software.

Accediendo como administrador

Aparecemos entonces en la página de la administración del servidor de banners, donde encontramos un formulario para introducir el nombre de usuario y la contraseña que indicamos en el asistente de instalación. Como aun no hemos protegido contra escritura el archivo de configuración "config.inc.php", podemos definir algunas configuraciones adicionales de PhpAdsNew.

Ahora podemos configurar nuevos detalles en la base de datos, sistemas de invocación y envío de los banner, el sistema de estadísticas, configuración de los banner, del administrador, etc.

Acordarse de rectificar los permisos de escritura, para dejarlos como estaban antes, una vez terminada la configuración del sistema. Con ello evitaremos problemas de seguridad.

Tareas automáticas

PhpAdsNew necesita configurar una tarea automática para poder realizar también automáticamente las tareas de mantenimiento.

Crear esa tarea puede ser el paso más engorroso, porque necesitamos programarla en el servidor donde está alojada la página, y lo más seguro es que no lo permitan. Otra posibilidad es crearla en un ordenador en nuestra casa, llamando al servidor remoto, pero eso hace necesario tener el ordenador encendido siempre que se deba realizar la tarea. En el manual del

administrador en PDF del que hablaba antes se encuentran distintas posibilidades para programar esa tarea, explicadas paso a paso.

Nosotros hemos usado durante bastante tiempo este servidor de banners en DesarrolloWeb.com con éxito, dejando pasar por completo este paso. No se enviarán los reportes de las campañas automáticamente al cliente y posiblemente haya algún otro problema, aunque no resulta demasiado crítico.

No obstante, consultar este artículo que explica como [crear una tarea automática en Linux para OpenAds](#).

Conclusión

PhpAdsNew es un servidor de banners muy potente y versátil. Estas notas servirán para guiarse en la instalación del software. Podremos trabajar con el programa con bastante facilidad, si estamos familiarizados de alguna manera con los conceptos del "ad serving" o servidores de banners. Aunque no quepa duda, que para aprender o dominar todo el sistema, necesitaremos leer detenidamente los manuales que se proporcionan y echarle unas cuantas horas y pruebas. Lo recomendamos mucho.

Artículo por Miguel Angel Alvarez

Recomendar un sitio usando PHP

Este sistema permite al usuario enviar desde la web, un e-mail a otra persona, invitándolo a visitar el sitio.

Solo requiere un módulo PHP que se encargará de mostrar el formulario de recomendación, enviar el e-mail y devolver un acuse de envío.

recomendar.php

```
<!-- formulario de recomendación -->

<? if ($HTTP_GET_VARS["accion"] == "") {
?>

<form method="post" action="recomendar.php?accion=enviar" name="recomienda">
<b>Recomienda este sitio</b><br><br>
Tu Nombre: <input type="text" name="n_remitente" size="10"><br>
Tu E-mail: <input type="text" name="e_remitente" size="20"><br>
Nombre de tu amigo: <input type="text" name="n_destinatario" size="10"><br>
E-mail de tu amigo: <input type="text" name="e_destinatario" size="20"><br><br>
<input type="submit" value="Recomendar">
</form>

<!-- envío del formulario y acuse de envío o información de errores -->
<?
}

elseif ($HTTP_GET_VARS["accion"] == "enviar") {

// recojo las variables que vienen desde el formulario
$n_destinatario = $HTTP_POST_VARS["n_destinatario"];
```

```
$e_destinatario = $_HTTP_POST_VARS["e_destinatario"];
$n_remitente = $_HTTP_POST_VARS["n_remitente"];
$e_remitente = $_HTTP_POST_VARS["e_remitente"];

// si los campos no están vacíos
if ($n_destinatario != "" && $e_destinatario != "" && $n_remitente != "" && $e_remitente != "") {

//indica la url de tu sitio
$url = "http://www.tusitio.com";
//indica el nombre de tu sitio
$nombre_del_sitio = "Tu Sitio";
//indica el asunto del mensaje
$asunto = $n_remitente . " te recomienda un sitio";
//redacta el mensaje
$mensaje = "Hola " . $n_destinatario . " :<br>";
$mensaje .= $n_remitente . " te recomienda que visites <b>" . $nombre_del_sitio . "</b>.<br>";
$mensaje .= "Puedes verlo en <a href=\"" . $url . "\">" . $url . "</a><br><br>Saludos!";

//indica que el mail se envía en formato HTML

$encabezado = "From: ".$e_remitente."\nReply-To: ".$e_remitente."\n";
$encabezado .= "X-Mailer: PHP/" . phpversion() . "\n";
$encabezado .= "Mime-Version: 1.0\n";
$encabezado .= "Content-Type: text/html";

//envía el mensaje

mail($e_destinatario,$asunto,$mensaje,$encabezado);

//Informa al usuario que se ha enviado el mensaje

echo "<b>El mensaje ha sido enviado</b>.<br>Gracias por recomendarnos!";

}

//si existen campos vacíos, envía un mensaje de error

else {

echo "Por favor, es necesario que completes todos los campos.<br>";
echo "<a href='recomendar.php'>Pincha aquí</a> para corregir los campos.";

}

}
?>
```

Artículo por **Eugenia Bahit**

PhpIsam para Web

El manejo de los Archivos Planos (anteriormente descritos en el anterior [artículo](#)) en proyectos de cierta envergadura es un trabajo complejo y no permite un acceso rápido en Archivos con gran cantidad de registros, pero usando ISAM podemos solucionar este problema con una simple línea en nuestro programa para manipular el Archivo .INX que es donde se alojan los Índices del Archivo en cuestión, de esta forma podemos seguir usando nuestros Archivos Planos sin problemas, al tener aislada en el Archivo aparte .INX la maquinaria necesaria para el acceso a datos.

En todas las aplicaciones un elemento fundamental son las listas de registros como Catálogos, Pedidos de un Cliente, Material enviado en una fecha, etc., por lo que disponemos de Comandos con este enfoque, como Registros a partir de una Clave o entre dos Claves determinadas, o aquellos en que la parte izquierda de la Clave sea igual al valor indicado, además también podemos combinar los casos anteriores entre sí.

Estos Comandos devuelven a nuestro programa una lista en la que figuran todos los registros en el orden establecido en el Índice y que cumplen las condiciones solicitadas. Como la lista puede ser muy larga se habilitan dos parámetros para recibir solo un número de registros a partir de una posición, esto nos permite recorrer un gran Archivo y traer solamente las líneas de datos que componen la Página vamos a presentar.

En nuestro programa solo tendremos que hacer una lectura en disco por cada número de registro de la lista devuelta por el programa de ISAM, números que vienen separados con coma por lo que resulta muy fácil, veamos un ejemplo para PHP:

```
$nConsu = $cabecera; // Obtener el Número de Consulta.
$Longi = 300; // Longitud registro de Respuestas
$nLineas = 20; // Indica 20 líneas por Página.
$inicio += $nLineas; // Avance de Página.

$resul = exec("serafoX I/Respuestas/0/" . $nConsu . "/" . $inicio . "/" . $nLineas . "\");
$totRegis = strtok( $resul, "/");
echo "Total registros encontrados:" . $totRegis;
$nuReg = strtok( $resul, ",");
while($nuReg){
    fseek( $fich, $nuReg*$Longi );
    $Buf = fread( $fich, $Longi );

    SEPARACION EN CAMPOS DEL REGISTRO $Buf LEIDO
    Y CREACION DE HTML PARA PRESENTAR LA LINEA

    $nuReg = strtok( $resul, ",");
}
```

La primer línea supone que hemos leído una Consulta y tenemos su número que es \$nConsu. Hemos utilizado el Comando I/Respuestas/etc. que nos devuelve una string con un número que es el total de registros encontrados y un / slash de separación seguido de los números de Registro a presentar separados por coma, los cuales vamos seleccionando con strtok, leyendo y presentando en el subsiguiente bucle while.

Por supuesto también se dispone de los Comandos clásicos como la Creación del Archivo de Índices, Inserción de una Clave, Eliminar, Buscar una sola Clave, Primera, Siguiente, Anterior, Última.

Aprovechando la operativa descrita, se habilita la posibilidad de ejecutar varios Comandos en una sola llamada al programa de ISAM, para casos como en las modificaciones de Claves donde usaremos primero el de Eliminar todas las Claves del registro y a continuación el de Insertar los nuevos valores, otro caso, y este el más interesante, nos permite seleccionar varios Índices del mismo Archivo y obtener una lista con los que cumplen todas las condiciones, y con del orden de clasificación de cualquiera de los Índices implicados, p.e. todos los registros del mismo Cliente comprendidos entre dos Fechas, para lo que usaremos un Comando g/ARCHIVO/2/cliente para seleccionar los del Cliente, y otro con k/ARCHIVO/3/fch1/fch2 para Limitar las Fechas. Hemos supuesto el Índice=2 para Cliente y el Índice=3 para Fecha.

Para obtener estos resultados se hace una sola llamada a ISAM con los Comandos seguidos y separados por chr(198) el caracter ASCII Æ = 198. En el caso de listas es obligatorio un número que indica cuál de ellas impone el orden en la lista resultante, en caso contrario no se realiza el cruce de los diferentes Indices. e/ARCHIVO/etc.Æh/ARCHIVO/etc. para el primer ejemplo y g/ARCHIVO/2/clienteÆ0k/ARCHIVO/3/fch1/fch2 para el segundo, donde el cero que acompaña al separador indica que el orden de los datos resultante será el del comando cero, o sea el Indice 2, y si deseamos orden por Fecha que es el Indice 3 pondremos un 1 con el separador.

Artículo por *Serafin Vazquez*

Recoger datos de un Select Múltiple con PHP

Quizá un problema que se nos puede plantear durante el trabajo con PHP, es querer recoger los datos de un Select Múltiple. Recordamos que un select múltiple es un campo de formulario donde se pueden seleccionar varios elementos a la vez.

Nota: Para seleccionar varios elementos de un campo select múltiple, tenemos que utilizar el botón de "control" e ir pulsando cada uno de los elementos que deseamos seleccionar. Es importante que este punto quede claro para los visitantes de vuestras páginas, ya que es muy habitual que los usuarios no sepan manejar este tipo de campos.

Para ver mejor lo que deseamos hacer, es muy interesante [entrar la página que contiene el ejemplo en funcionamiento](#). Para continuar, echemos un vistazo al siguiente ejemplo...

```
<form action="trataformulario.php" method="POST">
Nombre: <input type="text" name="nombre"><br>
Apellidos: <input type="text" name="apellidos"><br>
Email: <input type="text" name="email"> <br>
Cerveza: <br>
<select multiple name="cerveza">
<option value="SanMiguel">San Miguel</option>
<option value="Mahou">Mahou</option>
<option value="Heineken">Heineken</option>
<option value="Carlsberg">Carlsberg</option>
<option value="Aguila">Aguila</option>
...
</select><br>
<input type="submit" value="Enviar datos!" >
</form>
```

Tenemos este formulario en nuestra página y queremos recoger los datos introducidos por el usuario. Para los campos "nombre", "apellidos" y "email" no tenemos ningún tipo de problema, recogemos los datos por el método habitual, es decir:

```
<?php
echo "Nombre: ". $_POST["nombre"];
echo "<br>Apellidos: ". $_POST["apellidos"];
echo "<br>E-mail: ". $_POST ["email"];
?>
```

Uno se podría preguntar, y porque no puedo poner "\$cerveza=\$_POST["cerveza"]"? el problema reside en que los option del select múltiple tienen todos el mismo nombre y en caso

de tener varios elementos seleccionados sólo obtendríamos el último elemento⁰. Y como resolver esto? Pues bien, la solución es bien sencilla. Basta con poner en el final del nombre del campo del select múltiple unos corchetes ([]), en nuestro caso sería "cerveza[]". De este modo PHP interpreta el campo en cuestión como un array con tantas posiciones como elementos haya seleccionados en el select múltiple. Veamos pues como quedaría nuestro ejemplo...

```
<form action="trataformulario.php" method="POST">
Nombre: <input type="text" name="nombre"><br>
Apellidos: <input type="text" name="apellidos"><br>
Email: <input type="text" name="email"> <br>
Cerveza: <br>
<select multiple name="cerveza[]">
<option value="SanMiguel">San Miguel</option>
<option value="Mahou">Mahou</option>
<option value="Heineken">Heineken</option>
<option value="Carlsberg">Carlsberg</option>
<option value="Aguila">Aguila</option>
...
</select><br>
<input type="submit" value="Enviar datos!" >
</form>
```

Y para recoger la información y tratarla actuaríamos del siguiente modo...

```
<?php
echo "Nombre: ". $_POST["nombre"];
echo "<br>Apellidos: ". $_POST["apellidos"];
echo "<br>E-mail: ". $_POST ["email"];
$cervezas=$_POST["cerveza"];
```

//recorremos el array de cervezas seleccionadas. No olvidarse q la primera posición de un array es la 0

```
for ($i=0;$i<count($cervezas);$i++)
{
echo "<br> Cerveza " . $i . ": " . $cervezas[$i];
}
?>
```

Nota: Al ponerle este tipo de nombres (terminando en corchete) a un campo de un formulario y si usamos funciones Javascript para validarlos, manejarlos, etc... debemos añadir a dicho campo, además del nombre, el atributo de HTML id y utilizar este último como identificador del campo, en vez del nombre que se haya colocado en el atributo name, puesto que daría problemas.

El código completo de este ejemplo podría ser el siguiente. Sería bueno fijarse que hemos colocado todo el código en un archivo PHP. Ese código tiene un formulario reentrante, es decir, la página se llama a si misma para procesar el formulario. Recordar que puede verse este [ejemplo en funcionamiento en una página aparte](#).

```
<?
if (!$_POST){
?>
<form action="select_multiple.php" method="POST">
Nombre: <input type="text" name="nombre"><br>
Apellidos: <input type="text" name="apellidos"><br>
Email: <input type="text" name="email"> <br>
Cerveza: <br>
<select multiple name="cerveza[]">
<option value="SanMiguel">San Miguel</option>
<option value="Mahou">Mahou</option>
<option value="Heineken">Heineken</option>
<option value="Carlsberg">Carlsberg</option>
```

```
<option value="Aguila">Aguila</option>
</select><br>
<input type="submit" value="Enviar datos!" >
</form>
<?
}else{

echo "Nombre: ". $_POST["nombre"];
echo "<br>Apellidos: ". $_POST["apellidos"];
echo "<br>E-mail: ". $_POST ["email"];
$cervezas=$_POST["cerveza"];

//recorremos el array de cervezas seleccionadas. No olvidarse q la primera posición de un array es la 0

for ($i=0;$i<count($cervezas);$i++)
{
echo "<br> Cerveza " . $i . ": " . $cervezas[$i];
}

}
?>
```

*Artículo por **José Lorenzo Escudero***

Función que escribe un campo select con valores de una base de datos

Durante el trabajo con formularios en PHP, en muchas ocasiones tenemos un campo select cuyos valores posibles se obtienen de una base de datos, por ejemplo, los países. En este taller vamos a mostrar una función que es capaz de escribir en un formulario un campo de select, obteniendo los valores de una tabla de la base de datos. Con esta función podemos olvidarnos de la complejidad de mostrar un campo select. Simplemente llamamos a la función, enviando los parámetros, y se encarga de mostrar el campo.

En el caso de los países, puede que sea una buena idea tener una tabla con la lista de países del mundo. En el formulario lo habitual es permitir escoger un país de la lista y enviar el código del país. De modo podemos estar seguros de que todos los usuarios han elegido un país de la lista, entre todos los posibles.

La tabla de países podría contener algo como esto:

id_pais	nombre_pais
1	España
2	México
3	Argentina
4	Colombia
5	Chile

... ..

Este es el código HTML de un posible formulario cuyo campo de países ha sido extraído de la anterior tabla:

```
<form>
Nombre: <input type=text name="nombre">
<br>
País:
<select name=id_pais>
  <option value=1>España
  <option value=2>México
  <option value=3>Argentina
  <option value=4>Colombia
  <option value=5>Chile ...
</select>
```

Función `saca_menu_desplegable()`

Ahora vamos a ver cómo se haría la función que recorre la tabla de la base de datos para mostrar el select del país. Para empezar vamos a ver una lista de los parámetros que acepta.

- **ssql**: sentencia SQL que permita extraer los datos de las posibles opciones del select. La sentencia SQL tendría una forma como esta "select id_pais, nombre_pais from pais". En la primera columna de los registros seleccionados con esta sentencia, se espera el value de los option (en este caso id_pais). En la segunda columna se espera el texto de los option.
- **valor**: En este parámetro se recibe el valor predeterminado que debe aparecer en el select. Este valor se compara con el primer campo de la sentencia SQL. (Si deseamos que el valor predeterminado fuera "España", deberíamos poner un 1 como valor)
- **nombre**: Es el nombre que se le da al campo select de formulario

El código de la función es el siguiente:

```
function saca_menu_desplegable($ssql,$valor,$nombre){
  echo "<select name='$nombre'>";
  $resultado=mysql_query($ssql);
  while ($fila=mysql_fetch_row($resultado)){
    if ($fila[0]==$valor){
      echo "<option selected value='$fila[0]'>$fila[1]";
    }
    else{
      echo "<option value='$fila[0]'>$fila[1]";
    }
  }
  echo "</select>";
}
```

Se escribe el campo de formulario select, indicando el nombre del campo que se ha recibido por parámetro. Luego se realiza una búsqueda en la base de datos, ejecutando la sentencia SQL recibida por parámetro.

Más tarde se hace un recorrido del conjunto de registros que se han encontrado en la base de datos y se escriben cada uno de los option. Para cada uno de los registros se comprueba si su identificador es igual al que se desea marcar como valor predeterminado. En caso de serlo, se escribe el option con el atributo selected.

Un detalle importante es que la función necesita tener una conexión abierta con la base de datos, pues en caso contrario no se podría ejecutar la sentencia SQL.

*Artículo por **Miguel Angel Alvarez***

Script en PHP que muestra un libro aleatorio a partir de un array de libros

Vamos a ver un pequeño script que muestra un recuadro en una página web con un libro, que se ofrece para la venta, seleccionado aleatoriamente entre un conjunto de posibilidades.

El script se realiza utilizando un conjunto de libros almacenado en un array. Podía haberse utilizado una base de datos para almacenar los distintos libros, pero hemos utilizado un array - o tabla- por dos razones. La primera, porque este código lo hemos sacado de una página que no hace conexiones con bases de datos, por lo que teníamos que utilizar un array y cargar los datos en el código de la propia página. La segunda, razón tiene un sentido didáctico, porque nos sirve para hacer un ejemplo con arrays de dos dimensiones en PHP.

El ejercicio es muy simple. Podemos separarlo en tres bloques. 1) La carga de los datos del array de libros, 2) la generación de un número aleatorio para acceder a un libro al azar entre todos los posibles del array o tabla, y 3) la muestra de un recuadro en la página que obtiene los datos del array de libros, para mostrar el libro aleatorio.

La carga de datos se puede ver a continuación. Primero vamos a ver la creación de 10 arrays con los datos de cada uno de los libros. En concreto, de cada libro se almacenan los datos de interés, como su nombre, dirección con la que enlazar los enlaces para la compra y a imagen que habría que mostrar de cada libro. Más tarde, incluiremos esos arrays como distintas casillas de otra tabla, generando un segundo array que tendrá dos dimensiones. La primera será cada uno de los libros y la segunda cada una de las propiedades de los libros.

```
//creamos un conjunto de arrays con todos los libros que se quieren rotar aleatoriamente
$libro1 = array("Apache Práctico","http://www.libreria.com/Apache-Practico-n29833i.htm","29833.jpg");
$libro2 = array("Fotografía Digital para Profesionales","http://www.libreria.com/Fotografia-digital-para-profesionales-
n37228i.htm","37228.jpg");
$libro3 = array("Java 2 Versión 1.4","http://www.libreria.com/Java-2-version-1-4-n10575i.htm","10575.jpg");
$libro4 = array("Hacker","http://www.libreria.com/Hacker-n10655i.htm","10655.jpg");
$libro5 = array("Música Digital","http://www.libreria.com/Musica-Digital-n10313i.htm","10313.jpg");
$libro6 = array("El Arte de los Juegos 3D","http://www.libreria.com/El-arte-de-los-juegos-3D-
n10423i.htm","10423.jpg");
$libro7 = array("Wireless. Los mejores trucos","http://www.libreria.com/Wireless-Los-mejores-trucos-
n27052i.htm","27052.jpg");
$libro8 = array("Hardware y componentes","http://www.libreria.com/Hardware-y-componentes-Edicion-2004-
n10622i.htm","10622.jpg");
$libro9 = array("Consigue las Mejores Fotos con Tu Cámara Digital","http://www.libreria.com/Consigue-las-mejores-
fotos-con-tu-camara-digital-n10310i.htm","10310.jpg");
$libro10 = array("DVD, DIVX y otros Formatos de Vídeo Digital","http://www.libreria.com/DVD-DivX-y-otros-formatos-
de-Video-Digital-n10506i.htm","10506.jpg");

//creamos un array con todos los libros que deseamos que roten
$libros = array($libro1, $libro2, $libro3, $libro4, $libro5, $libro6, $libro7, $libro8, $libro9, $libro10);
//el resultado de crear este segundo array (que es un array de arrays) es un nuevo array de dos dimensiones.
```

El segundo bloque corresponde con la creación de un número aleatorio para acceder a un libro al azar de entre todos los posibles del array. Para ello debemos obtener un número que se encuentre entre el cero y nueve. Nueve es el número de libros posibles, que corresponde con las posiciones del array \$libros menos uno.

```
//generamos un número aleatorio entre cero y el número de posiciones del array menos uno (los arrays en php
```

```
empiezan en la posición cero)
srand((double)microtime()*1000000);
$valor_aleatorio = rand(0,count($libros)-1);
```

Para acabar, tenemos un código HTML que muestra un recuadro con un libro. Los datos del libro que mostrar se obtienen del array, accediendo por el índice obtenido aleatoriamente. Dada la variable aleatoria que tenemos, que se llama \$valor_aleatorio, accederíamos al libro aleatorio con este código:

```
$libros[$valor_aleatorio]
```

Como el libro es a su vez otro array, para acceder a cada una de las posiciones utilizaríamos un índice. Para acceder al título del libro, tenemos que utilizar el índice primero:

```
$libros[$valor_aleatorio][0]
```

El código del recuadro quedaría de esta manera.

```
<table width="250" cellspacing="0" cellpadding="1" border="0" style="border-width: 1px; border-color: 808050; border-style: solid;">
<tr>
<td class=titcol><b>LIBRO RECOMENDADO</b></td>
</tr>
<tr>
<td class=fuente8 style="line-height:17px;"><table width=100% border=0 cellpadding=0 cellspacing=0 align=center><tr>
<td WIDTH=1><a href="<?echo $libros[$valor_aleatorio][1];?>" target="_blank"></a></td>
<td width="5"></td><td valign=top class=fuente8><b><a href="<?echo $libros[$valor_aleatorio][1];?>" target="_blank"><?echo $libros[$valor_aleatorio][0];?></a></b><p class=fuente8><a href="<?echo $libros[$valor_aleatorio][1];?>" target="_blank">Compra este libro</a> en nuestra libreria a domicilio.</div></tr>
</table>
</td></tr>
</table>
```

Este ejemplo se puede ver en funcionamiento en la portada de DesarrolloWeb.com. Sabemos que es un código un poco específico, pero esperamos que sirva como ejemplo de uso de los arrays en PHP y en concreto de los arrays de dos dimensiones.

*Artículo por **Miguel Angel Alvarez***

Paginación con PHP y Sybase

Aquí envío una adaptación del script de paginación, que hice para el motor de base de datos Sybase.

Referencia: La paginación la hemos visto y explicado con detalle en PHP, para la base de datos MySQL.
[Paginación de resultados con PHP y MySQL](#)
[Paginacion con PHP y MySQL. Ejemplo 2](#)
[Paginador PHP usando pear y templates](#)

Espero sea de utilidad para todos. El código es el siguiente. Está ligeramente comentado para que sirva de mayor utilidad.

```
<?php
$conn = sybase_connect("lapsededatos", "elsuario", "");

$TAMANO_PAGINA = 20;

$pagina = $_GET["pagina"];
if (!$pagina) {
    $inicio = 0;
    $pagina=1;
}
else {
    $inicio = ($pagina - 1) * $TAMANO_PAGINA;
}
//Importante hace el query ordenado y sin limit
$sql = "select * from data_user ORDER by nombre";
$rs = sybase_query($sql,$conn);
$num_total_registros = sybase_num_rows($rs);

$total_paginas = ceil($num_total_registros / $TAMANO_PAGINA);

echo "Número de registros encontrados: " . $num_total_registros . "<br>";
echo "Se muestran páginas de " . $TAMANO_PAGINA . " registros cada una<br>";
echo "Mostrando la página " . $pagina . " de " . $total_paginas . "<p>";

$rs = sybase_query($sql,$conn);

$arreglo[] = "";
while ($fila = sybase_fetch_object($rs)){
    $arreglo[]=$fila->nombre;
}

sybase_free_result($rs);
sybase_close($conn);

$contarray=1;

//Aqui, paso los datos extraidos en el query a un arreglo
//si son más campos como nombre, apellido...
//se puede trampiar así: $arreglo[]=$fila->nombre, $fila->numero_tlf";

foreach ($arreglo as $idarray=> $nombre){
    //aqui se hace la verificación que funciona como el limit

    if ($idarray > $inicio && $contarray <= $TAMANO_PAGINA){
        echo "$idarray $nombre<br>";
        $contarray++;
    }
}
if ($total_paginas > 1){
    for ($i=1;$i<=$total_paginas;$i++){
        if ($pagina == $i){
            echo $pagina . " ";
        }else{
            echo "<a href='dbd2.php?pagina=" . $i . "'>" . $i . "</a> ";
        }
    }
}
//Por José pino//
?>
```

*Artículo por **José Pino***

Comprimir página PHP

Para aligerar el tiempo de carga de nuestras páginas generadas con PHP, podemos enviarlas al navegador comprimidas con GZip utilizando las funciones de control de salida, para ello, llamaremos a la función predefinida `ob_gzhandler` como tratante de la función `ob_start`, veamos un ejemplo:

```
<?
ob_start("ob_gzhandler");

// Contenido de la página, puede contener
// tanto HTML como PHP

ob_end_flush();
?>
```

Tener en cuenta que todo el contenido debe estar en el lugar indicado por lo que los primeros caracteres del documento deben ser `<?` y los dos últimos `?>` y no se debe añadir nada excepto donde se indica, si no vamos con cuidado recibiremos un error parecido al siguiente:

Warning: Cannot add header information...

Otra forma más completa todavía de compresión, consiste en aplicar la misma función, pero eliminando a su vez los espacios y saltos de línea de la fuente del documento, lo que no tendrá ningún efecto visual pero disminuirá el tiempo de descarga, veamos cómo hacerlo:

```
<?
ob_start();

// Contenido de la página, puede contener
// tanto HTML como PHP

$cntACmp =ob_get_contents();
ob_end_clean();
$cntACmp=str_replace("\n",' ',$cntACmp);
$cntACmp=ereg_replace('[:space:]+',' ',$cntACmp);
ob_start("ob_gzhandler");
echo $cntACmp;
ob_end_flush();
?>
```

Este método funciona igual que el anterior solo que antes de comprimir elimina los saltos de línea y espacios innecesarios, pero debemos tener en cuenta las mismas precauciones o no funcionará.

Este método de compresión lo he desarrollado para ProgramacionWeb, como podéis comprobar al ver el código fuente de esta misma página, todo el código aparece sin saltos de línea ni tabulado por una simple cuestión de compresión.

*Artículo por **Eloi de San Martin Lagranje***

Creación de un log de errores

Un log de errores, nos permitirá controlar cuando se ha producido un error para corregirlo y evitar que se repita en el futuro.

Para crear un log, abriremos el archivo en modo 'a' (escritura al final) y escribiremos el error indicando la fecha, para simplificar el trabajo lo podemos incluir todo en una función:

```
<?php
function error($numero,$texto){
$ddf = fopen('error.log','a');
fwrite($ddf,"[" .date("r")."] Error $numero: $texto\r\n");
fclose($ddf);
}
?>
```

Una vez declarada la función, tan solo tendremos que llamarla de la siguiente manera cuando se produzca un error para que se guarde en error.log:

```
<?php
// Si no existe la cookie sesion
if(!isset($_COOKIE['sesion'])){
// Guardamos un error
error('001','No existe la cookie de sesion');
}
?>
```

De esta manera, cada vez que un usuario entra a esta página sin la cookie sesion, se almacena una nueva línea en el fichero indicando:

```
[fecha] Error 001: No existe la cookie de sesion
```

Vamos a ver ahora como podemos mejorar esto de manera que además de poder grabar los errores que nosotros definamos en nuestro sitio, nos almacene los errores producidos durante la ejecución del script php.

Esto lo conseguiremos indicando al interprete Zend que llame a la función error() cada vez que el código PHP contenga un error con la función set_error_handler:

```
<?php
set_error_handler('error');
?>
```

Entonces, el código completo nos queda de la siguiente manera:

```
<?php
function error($numero,$texto){
$ddf = fopen('error.log','a');
fwrite($ddf,"[" .date("r")."] Error $numero:$texto\r\n");
fclose($ddf);
}
set_error_handler('error');
?>
```


Y de esta manera damos por finalizado nuestro script para generar un log de errores personales y de PHP.

*Artículo por **Eloi de San Martin Lagranje***

Paginator, script para paginación con PHP y MySQL

Paginator es un script de paginación desarrollado en PHP para dividir resultados de consultas extensas a una base de datos MySQL, en grupos de "n" registros por página. Genera, además, una "barra de navegación" que contiene los enlaces a las diferentes páginas (<<anterior 1 2 3 4 siguiente>>).

Nota: Como ejemplo de paginación podemos ver la página de google, donde se dice que hay 13,235 resultados, pero aparecen divididos en varias páginas de 20 resultados cada una.

La característica principal de este script es su fácil utilización, ya que la forma de paginar es muy conocida y utilizada. Por ello, no es necesario entenderlo ni editarlo, sino simplemente incluirlo después de definir al menos una variable. No obstante, todas las líneas están comentadas debidamente, para que sea sencillo seguir la secuencia y saber qué hace el script en cada línea. Y lo mejor de todo es que está completamente en español.

Utilización del script

La utilización es extremadamente sencilla. Debemos Conectar con la Base de datos, definir una sentencia sql válida para MySQL y almacenarla en la variable \$_pagi_sql. Esta sentencia SQL no debe contener la cláusula "LIMIT", pues será agregada automáticamente por el script.

La definición de esta variable es obligatoria. Se pueden crear otras variables para personalizar el sistema de paginación en detalles como el número de resultados por página o el número de enlaces a otras páginas de resultados que aparecerán.

Luego, se incluye el script de paginación, y se generan otras variables que podemos utilizar para hacer la página de resultados. \$_pagi_result es el id de resultado de la consulta, que podemos utilizar con alguna función tipo mysql_fetch_array(). \$_pagi_navegación contiene la barra de navegación con enlaces a las distintas páginas y \$_pagi_info almacena la información de registros visualizados y número de páginas, algo así como "desde el 16 hasta el 30 de un total de 123".

Vamos a ver un código sencillo que utiliza Paginator para gestionar la paginación de resultados.

```
//Conexión a la base de datos
$con = mysql_connect("localhost","tu_username","tu_password") or die (mysql_error());
mysql_select_db("tu_base",$con) or die (mysql_error());

//Sentencia sql (sin limit)
$_pagi_sql = "SELECT * FROM clientes WHERE sexo='m' ORDER BY edad";

//cantidad de resultados por página (opcional, por defecto 20)
$_pagi_cuantos = 10;
```

```
//Incluimos el script de paginación. Éste ya ejecuta la consulta automáticamente
include("paginator.inc.php");

//Leemos y escribimos los registros de la página actual
while($row = mysql_fetch_array($_pagi_result)){
    echo $row['nombre']."<br />";
}

//Incluimos la barra de navegación
echo"<p>".$_pagi_navegacion."</p>";
```

Como podemos ver, no hay mayor trabajo adicional, respecto al que se realizaría en un código sin paginar. Cabe aclarar que el nombre de las variables que utiliza el script (internas y externas) empiezan todas por el prefijo `$_pagi_` para evitar conflictos por coincidencias con los nombres de otras variables que se estén utilizando.

Para acceder a una descripción detallada de Paginator y descargar el script de manera gratuita, se puede su página de inicio: <http://jpinedo.webcindario.com/scripts/paginator/>

Referencia: Si quieres aprender a paginar resultados por tu cuenta, en DesarrolloWeb.com hemos publicado un artículo interesante que te puede venir bien: [Paginación de resultados con PHP y MySQL](#).

*Artículo por **Jorge Pinedo***

Ejecutar scripts PHP automáticamente desde el cron con CURL

Frecuentemente se nos presenta la necesidad de tener que programar alguna tarea diaria, semanal etc...como un envío masivo de email, o unas consultas pesadas a nuestra base de datos, y hacerlo además en las horas de menor actividad de nuestro servidor.

Para muchos de nosotros resulta fácil programar un script con php que realice estas tareas, pero la mayoría de los servidores tienen solo el módulo de php instalado lo cual no nos permite ejecutar estos scripts desde la línea de comandos directamente.

La solución a este problema consiste en utilizar CURL. Este paquete viene incorporado en la mayoría de las distribuciones Linux, y en caso contrario, puede ser descargado desde esta dirección <http://curl.haxx.se>.

CURL es una herramienta para transferencia de ficheros con sintaxis URL, que envía la salida por defecto a la pantalla. Podemos hacer la siguiente prueba

```
# curl http://www.google.es
```

También podemos reenviar la salida y guardarla en un fichero

```
# curl http://www.google.es > google.htm
```

CURL soporta diferentes protocolos FTP, FTPS, HTTP, HTTPS, GOPHER, TELNET, DICT, FILE, LDAP y algunos mas que podéis consultar en su web. Además se instala una librería con un conjunto de funciones que pueden ser usadas desde php, pero esto es otra historia.

Siguiendo con nuestro caso, podemos ejecutar ahora un script php desde la línea de comandos, simplemente escribiendo esto:

```
# curl http://localhost/miscript.php
```

Supongamos que vamos a ejecutar un script que envía los email de nuestro boletín de correo llamado boletin.php pero este se encuentra alojado en una carpeta protegida de nuestro servidor web. En este caso ejecutamos el script pasándole el usuario y contraseña para acceder al directorio protegido.

```
#curl -u username:password http://localhost/protegido/boletin.php
```

Para programar en el cron de nuestro linux esta tarea y que sea ejecutada automáticamente de forma periódica debemos conocer la ruta completa donde esta instalado el curl que habitualmente suele ser en /usr/bin.

Ahora editaremos nuestro fichero de tareas de cron (por ejemplo con el comando crontab -e), y añadimos esta línea:

```
30 4 * * * /usr/bin/curl -u username:password http://localhost/protegido/boletin.php
```

Esto hará que todos los días a las 4:30 de la madrugada se ejecute el script boletin.php de manera automática. Si se desea conocer mas información sobre la programación de tareas consultar la ayuda (man crontab).

Artículo cedido por [ADR Formación](#)

*Artículo por **Alfredo Anaya***

Uso del FTP con PHP

Constantemente nos encontramos bajando archivos de Internet: Un programilla que alguien nos recomienda, la foto de la modelo de moda o los clásicos MP3's . Sin darnos cuenta de uno u otro modo hacemos uso del protocolo FTP (File Transfer Protocol) para bajar archivos desde un Servidor. Descargamos más de los que subimos... en otras palabras "Tomamos más de los que damos". Este artículo pretende explicarle el cómo subir -o permitir que sus usuarios suban (aporten) - archivos al Servidor usando las funciones FTP incluidas en el PHP. Recuerden el viejo dicho que cita: "Mas vale dar que recibir".

¿QUE ES EL FTP?

Siglas de File Transfer Protocol o Protocolo de Transferencia de Archivos. Como su propio nombre lo indica, es un protocolo (perteneciente a TCP/IP) que se encarga de la transferencia de archivos entre computadoras conectadas en red. Al basarse en la arquitectura Cliente / Servidor, el FTP hace uso de dos componentes básicos:

- **Un cliente FTP.** El cual se encarga de conectarse a un servidor FTP para descargar o subir archivos.
- **Un servidor FTP.** Se encarga de procesar las peticiones de los clientes FTP, permitiéndoles descargar o subir archivos desde él.

Para conectarnos a un servidor FTP, y así poder realizar consultas en él, necesitaremos los siguientes datos:

- **Nombre del Servidor.** Es la IP o Nombre del Servidor FTP al cual nos hemos de conectar, tal como: 65.134.10.5 o ftp.billysite.net
- **Puerto.** Número del puerto del servidor. Por defecto es 21.
- **Cuenta de Usuario.** Es el nombre de la cuenta de usuario que se nos ha asignado. Hay que asegurarse que cuenta con los permisos necesarios para subir o bajar archivos. De no tener una cuenta de usuario se puede acceder como usuario anónimo utilizando el nombre de usuario *anonymous*.
- **Clave de acceso.** Es nuestra contraseña de cuenta de usuario. De acceder como usuario anónimo colocaremos como clave nuestro correo-e a manera de cortesía.

Una vez conectados al servidor FTP podremos hacer uso de su comandos para realizar las tareas que mejor creamos conveniente. Este artículo no pretende esbozar en gran medida el tema de FTP, al no ser éste el objetivo fundamental del mismo. Para más información sobre este aspecto le recomiendo revisar el artículo de Michael Calore: "El ABC de la transferencia de archivos por Internet", disponible en el web site de [WebMonkey](http://www.webmonkey.com).

FUNCIONES FTP EN PHP.

PHP hace uso de funciones FTP para acceder aun servidor web, a manera de cliente. A continuación mostraremos las funciones básicas a usar en el script, así como una breve descripción de las misma. Si quiere mayor detalle de estas y otras funciones FTP le aconsejo consultar la documentación oficial del PHP, disponible en: <http://www.php.net/docs.php>.

Función	Sintaxis	Descripción
ftp_connect	int ftp_connect (string host [, int port]) host: Nombre o IP de Servidor FTP. port: Puerto, por defecto 21.	Establece una conexión FTP al host especificado.
ftp_login	int ftp_login(int ftp_stream, string username, string password) ftp_stream: Manejador FTP obtenido con ftp_connect. username: Nombre de usuario. password: contraseña de usuario.	Comienza la sesion en una conexión FTP.
ftp_pasv	int ftp_pasv (int ftp_stream, int pasv) ftp_stream: Manejador FTP obtenido con ftp_connect. pasv: Si es TRUE activa el modo pasivo, si es FALSE lo desactiva.	Activa o desactiva el modo pasivo. En modo pasivo, las conexiones de datos son iniciadas por el cliente, en lugar de ser iniciadas por el servidor.
ftp_pwd	int ftp_pwd (int ftp_stream) ftp_stream: Manejador FTP obtenido con ftp_connect.	Devuelve el nombre del directorio actual.
ftp_put	int ftp_put (int ftp_stream, string remote_file, string local_file, int mode) ftp_stream: Manejador FTP obtenido con ftp_connect. remote_file: Nombre con el cual se guardará el archivo en el Servidor FTP. local_file: Archivo local que se encuentra en la	Sube un fichero al Servidor FTP.

	máquina cliente. mode: Modo de transferencia, puede ser FTP_ASCII o FTP_BINARY.	
ftp_nlist	int ftp_nlist (int ftp_stream, string directory) ftp_stream: Manejador FTP obtenido con ftp_connect. directory: Ruta del directorio a listar.	Devuelve una lista de ficheros del directorio dado.
ftp_size	int ftp_size (int ftp_stream, string remote_file) ftp_stream: Manejador FTP obtenido con ftp_connect. remote_file: Nombre del archivo en el Servidor FTP.	Devuelve el tamaño del fichero especificado. No todos los servidores soportan esta característica.
ftp_mdtm	int ftp_mdtm (int ftp_stream, string remote_file) ftp_stream: Manejador FTP obtenido con ftp_connect. remote_file: Nombre del archivo en el Servidor FTP.	Devuelve la fecha de última modificación del fichero especificado. No todos los servidores soportan esta característica
ftp_quit	int ftp_quit (int ftp_stream) ftp_stream: Manejador FTP obtenido con ftp_connect.	Cierra una conexión FTP

Nota: Debe asegurarse que se encuentren habilitadas las funciones ftp en la configuración de la version de PHP que posee y de tener los permisos necesarios en su cuenta FTP para subir y bajar archivos.

CODIGO FUENTE.

/inc/ftpfunc.php.

Script que contendrá las constantes y funciones a usar en **index.php**. En este script deberá modificar los valores de las constantes para ajustarlo a sus necesidades. La función **ConectarFTP** le permitirá conectarse al Servidor FTP; la función **SubirArchivo** tiene la tarea de subir un archivo al Servidor; y finalmente, la función **ObtenerRuta** le otorgará la ruta del directorio actual en el cual está trabajando el Servidor.

```
<?
# FUNCIONES FTP
# CONSTANTES
# Cambie estos datos por los de su Servidor FTP
define("SERVER","localhost"); //IP o Nombre del Servidor
define("PORT",21); //Puerto
define("USER","willy"); //Nombre de Usuario
define("PASSWORD","12345"); //Contraseña de acceso
define("PASV",true); //Activa modo pasivo

# FUNCIONES
function ConectarFTP(){
//Permite conectarse al Servidor FTP
$id_ftp=ftp_connect(SERVER,PORT); //Obtiene un manejador del Servidor FTP
ftp_login($id_ftp,USER,PASSWORD); //Se logea al Servidor FTP
ftp_pasv($id_ftp,MODODO); //Establece el modo de conexión
return $id_ftp; //Devuelve el manejador a la función
}
function SubirArchivo($archivo_local,$archivo_remoto){
//Sube archivo de la maquina Cliente al Servidor (Comando PUT)
$id_ftp=ConectarFTP(); //Obtiene un manejador y se conecta al Servidor FTP
ftp_put($id_ftp,$archivo_remoto,$archivo_local,FTP_BINARY);
//Sube un archivo al Servidor FTP en modo Binario
```

```

ftp_quit($id_ftp); //Cierra la conexion FTP
}
function ObtenerRuta(){
//Obriene ruta del directorio del Servidor FTP (Comando PWD)
$id_ftp=ConectarFTP(); //Obtiene un manejador y se conecta al Servidor FTP
$Directorio=ftp_pwd($id_ftp); //Devuelve ruta actual p.e. "/home/willy"
ftp_quit($id_ftp); //Cierra la conexion FTP
return $Directorio; //Devuelve la ruta a la función
}
?>
    
```

index.php.

Script que contiene un formulario (**form_ftp**) que nos permite buscar un archivo y subirlo al Servidor FTP, además nos muestra una lista de los directorios y archivos del mismo.



```

<?php echo "<?xml version="1.0" encoding="iso-8859-1"?." ">"; ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>...Funciones FTP...</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
</head>
<body>
<p align="center"><font size="5" face="Verdana, Tahoma, Arial"><strong><em>
Funciones FTP
</em></strong></font></p>
<p><font face="Verdana, Tahoma, Arial">
<?
include('inc/ftfunc.php'); //Incluye el archivo de funciones
if(!empty($_POST["archivo"])){ //Comprueba si la variable "archivo" se ha definido
SubirArchivo($_POST["archivo"],basename($_POST["archivo"]));
//basename obtiene el nombre de archivo sin la ruta
unset($_POST["archivo"]); //Destruye la variable "archivo"
}
?>
<strong><font color="#000000" size="3">Subir Archivo</font></strong></font></p>
<hr />
<!--Formulario para elegir el archivo a subir -->
<form action="" method="post" name="form_ftp" id="form_ftp">
<p><font size="2" face="Verdana, Tahoma, Arial"> Elegir archivo :
    
```

```
<input name="archivo" type="file" id="archivo" />
<input name="Submit" type="submit" value="Subir Archivo" />
</font><font size="2" face="Verdana, Tahoma, Arial"> </font> </p>
</form>
<hr />
<p><font face="Verdana, Tahoma, Arial"><strong><font color="#000000" size="3">
Lista de Archivos
</font></strong></font></p>
<table width="69%" border="1" cellspacing="0" cellpadding="0">
<tr>
<td width="48%"><div align="center"><font size="2" face="Verdana, Tahoma,
Arial"><strong>Nombre</strong></font></div></td>
<td width="22%"><div align="center"><font size="2" face="Verdana, Tahoma,
Arial"><strong>Tamaño</strong></font></div></td>
<td width="30%"><div align="center"><font size="2" face="Verdana, Tahoma, Arial"><strong>Fecha de
Modificación</strong></font></div></td>
</tr>
<?
$id_ftp=ConectarFTP(); //Obtiene un manejador y se conecta al Servidor FTP
$ruta=ObtenerRuta(); //Obtiene la ruta actual en el Servidor FTP
echo "<b>El directorio actual es: </b> ". $ruta;
$lista=ftp_nlist($id_ftp,$ruta); //Devuelve un array con los nombres de ficheros
$lista=array_reverse($lista); //Invierte orden del array (ordena array)
while ($item=array_pop($lista)) //Se leen todos los ficheros y directorios del directorio
{
$tamano=number_format((ftp_size($id_ftp,$item))/1024,2)." Kb";
//Obtiene tamaño de archivo y lo pasa a KB
if($tamano=="-0.00 Kb") // Si es -0.00 Kb se refiere a un directorio
{
$item="<i>". $item."</i>";
$tamano="&nbsp;";
$fecha="&nbsp;";
} else{
$fecha=date("d/m/y h:i:s", ftp_mdtm($id_ftp,$item));
//Filetime obtiene la fecha de modificación del fichero; y date le da el formato de salida
}
?>

<tr>
<td><font size="2" face="Verdana, Tahoma, Arial"><? echo $item ?></font></td>
<td align="right"><font size="2" face="Verdana, Tahoma, Arial"><? echo $tamano ?></font></td>
<td align="right"><font size="2" face="Verdana, Tahoma, Arial"><? echo $fecha ?></font></td>
</tr>
<? } ?>
</table>
</body>
</html>
```

Bueno, espero que éste aporte a al Comunidad Web les haya sido de utilidad, cualquier duda o sugerencia no duden en hacermesla llegar. Saludos.

Artículo por **William Wong Garay**

Portal PHP Nuke

Para la creación de un portal PHP Nuke es necesario saber antes, con qué nos vamos a encontrar. El sistema de gestión de portales PHP Nuke se ejecuta bajo un servidor Apache, utiliza el lenguaje de programación PHP y necesita de una base de datos MySQL. Estas tres aplicaciones se instalarían por separado y después habría que configurarlas para que

funcionaran entre sí. Pero aquí vamos a utilizar un programa que nos instala a la vez estas tres aplicaciones, se lo pueden descargar de cualquiera de estos [enlaces](#).



Instalación de Apache, PHP y MySQL

Ahora siga estas instrucciones para instalar este programa, que coordina Apache, PHP y MySQL:

1. Pulse en "next" hasta que le pida que introduzca el "server name" y "administrator's email adress".
2. En "server name" ponga "localhost". En "administrator's email adress" ponga su email. Y el puerto no lo modifique (80 por defecto).
3. Pulse en "next". Ahora le pedirá que introduzca el "user name" y el "password". Ponga el que usted quiera, esto sirve para acceder a nuestra base de datos.
4. Pulse en "next". Ahora acabará la instalación.
5. Pulse en "Close". Le aparecerá un ventana de ms-2, puede cerrarla.

Para comprobar su correcto funcionamiento abra su navegador (Internet Explorer, Mozilla...) e introduzca esta dirección (su dirección local) "http://localhost" . Si aparece una página donde pone "The AppServ Open Project..." es correcto todo.

Ahora el programa ha creado en su disco duro una carpeta llamada AppServ, la cual a su vez contiene: "Apache", "PHP", "MySQL" y "www". La carpeta "www" es donde puede poner cualquier archivo php para que sea ejecutado.

Normalmente todo lo anterior se realiza, para preparar la web antes de subirla al servidor, pero si usted va a trabajar directamente en el servidor web, puede seguir leyendo en el siguiente punto. (Asegúrese de que su servidor corre bajo Apache, PHP y tiene base de datos MySQL: un ejemplo de este tipo de servidor es www.tripod.lycos.es).

Instalación de un Portal PHP Nuke

Primero ha de descargarse los archivos que lo componen: puede descargárselo de este [enlace](#) directamente. Una vez descargado, extraiga (está en zip) los archivos a la anteriormente mencionada carpeta "www" dentro de "AppServ" en su disco duro. Ahora explicaré la

configuración de los archivos:

1. Abra su navegador web e introduzca esta dirección "http://localhost/phpMyAdmin".
2. Corrija los errores que aparecen en rojo, para ello, copie los mensajes de error en el "Google" y rápidamente dará con la solución. Aquí le explico los más usuales. El error que siempre sale al principio es el de la directiva \$cfg. Para solucionarlo abra con el bloc de notas el archivo siguiente :
su_disco_duro/AppServ/www/phpMyAdmin/congif.inc.php
3. Busque (Ctrl+B) la siguiente línea: \$cfg['PmaAbsoluteUri'] = ''; e introduzca esto entre las comillas http://localhost/phpMyAdmin de tal manera que quede así
\$cfg['PmaAbsoluteUri'] = 'http://localhost/phpMyAdmin'; . Ya está solucionado el error. Ahora queda otro, el que dice que el root no tiene contraseña.
4. Vuelva a la página "http://localhost/phpMyAdmin" (observe que si actualiza la página ya no sale el primer error), pulse en privilegios y luego pulse en editar (a la derecha de las filas root y localhost) . Ahora en el punto "Cambio de contraseña" pulse en la opción "contraseña" e introduzca una contraseña, reescribala y pulse en "continúe".
5. Le saldrá un error, la pantalla aparece medio en blanco, debemos configurar otro archivo más. Abra de nuevo (con el bloc de notas) el archivo "congif.inc.php" y busque (Ctrl+B) esto: \$cfg['Servers'][\$i]['host'], le llevará a la sección donde especificar los datos por los que nos daba error. Entre las comillas de la fila password introduzca el password que ha introducido antes en los privilegios del phpMyAdmin. Guarde y cierre el archivo.
6. Vaya a la página del phpMyAdmin (puede comprobar que todo funciona correctamente, sin errores), escriba nuke (por ejemplo) en el campo "crear una nueva base de datos" y pulse "crear".
7. Abra con el bloc de notas el archivo "su_disco_duro/AppServ/www/sql/nuke.sql" y seleccione todo, cópielo (Ctrl+C).
8. Abra el phpMyAdmin, a la izquierda, seleccione la base de datos creada antes (nuke), pulse en la pestaña superior "SQL", pegue (Ctrl+V) el texto antes copiado y pulse "continúe".
9. Abra con el bloc de notas el archivo "su_disco_duro/AppServ/www/html/config.php"
10. Rellene los siguientes datos: en el "dbpass", a la derecha, entre las comillas, ponga el password que puso inicialmente en el phpMyAdmin y en "dbname" ponga 'nuke' (si está ejecutando el php-nuke online en el "host" ponga su servidor). Guarde y cierre el archivo.
11. En su navegador ponga "http://localhost/html/admin.php" y si aparece un portal PHP Nuke, lo ha hecho todo correctamente.
12. Configure su superusuario, ponga el usuario, password, email... que quiera. Aquí lo único que puede que no sepa es la página de inicio, debe poner "http://localhost".

Ya tiene su propio portal PHP Nuke, lo puede configurar como usted quiera. Disfrútelo.

*Artículo por **Raúl Avilés***

Instalación de One or Zero PHP Helpdesk

One or Zero Helpdesk and Task Management System es un programa de código abierto muy potente y ligero. Está desarrollado en PHP y utiliza la base de datos MySQL y Sqlite. Es rápido, personalizable y se puede ejecutar en cualquier plataforma. Se trata de un producto gratuito, tanto para uso personal como comercial.

Este artículo cubre la versión 1.4 del programa. Asumo que el servicio técnico tiene acceso telnet (o ssh) al servidor y que tiene conocimientos básicos de Linux.

Aun así intentaré dar explicaciones alternativas por si la instalación se realiza sin conexión ssh/telnet.

Dónde obtenerlo y qué contiene

Puedes descargar la última versión de <http://www.oneorzero.com/index.php?page=downloads>

A pesar de que en la página oficial hay links para descargar un kit de idioma, este no es necesario, ya que viene incluido en el documento .zip

Los contenidos de esta versión se organizan en carpetas:

- admin: Páginas de administración. Con él podemos instalar muy fácilmente el programa
- common: Contiene archivos comunes, entre ellos "config.php"
- images: Obviamente, las imágenes del sitio
- kbase: Archivos de configuración
- supporter: Páginas del técnico que dará soporte a los usuarios
- lang: Archivos de lenguaje, entre ellos el de castellano
- docs: Archivos txt en inglés con el agreement, instrucciones de instalación...etc

Requerimientos mínimos

- Sistema operativo Windows o Linux.
- Apache
- PHP 4.0 (o versión superior)
- MySQL
- SSL (para transacciones seguras)
- Software de foros (opcional) Se recomienda XMB Forum (<http://www.xmbforum.com/>)

Nota: Yo lo he probado en un php 4.1.2 y funciona estupendamente.

Instalación del software

Primeros pasos:

Entra al servidor remoto

Accede al directorio del usuario que ha solicitado la instalación

Crea una carpeta "helpdesk" (mkdir helpdesk)

Entra en la carpeta helpdesk (cd helpdesk)

Sube el archivo "ozhv14f.zip" a la carpeta usando FTP.

Descomprime el archivo (unzip ozhv14f.zip) -> En caso que la distro no tenga el paquete unzip instalado, siempre puedes descomprimirlo en un pc con Windows y subir la carpeta.

Elimiando lo que no queremos:

Borra el archivo comprimido (rm -rf ozhv14f.zip)

Permisos:

Para que el usuario pueda, en un futuro administrar el helpdesk, no olvides cambiar de propietario la carpeta phpnuke (chown usuario:grupousuario ./phpnuke)

Configuración: Edita el archivo "common/config.php"

\$db_host Nombre del host donde está la base de datos --> Dejar en "localhost"
\$db_user Usuario (del hosting, que coincide con el de la bbdd)
\$db_pwd Contraseña (del hosting, que coincide con la de la bbdd)
\$db_name Nombre de la bbdd

Los demás campos no es necesario tocarlos.

No metais la versión de php en un campo que viene un poco más abajo, sólo es por si usais una versión mayor o igual a la 4.2.0

Añadir las tablas de la base de datos y configurar el administrador

Esto también se lo podemos dejar al cliente, pero ocasionará problemas con los menos experimentados.

Entramos via web a <http://dominiodelcliente.com/helpdesk/install.php>

Solo hace falta rellenar los campos que pide (login, passwd, etc), y como veis la instalación es muy sencilla. El solito genera las tablas de la bbdd.

Después ve al panel de control y cambia el "default lenguaje" por "Spanish". Aunque no veas el cambio en ese momento, sal y entra, y verás que ya sale en castellano.

Notas finales

Ya hemos terminado la instalación. Las páginas donde ha quedado la aplicación son <http://www.dominio.com/helpdesk> para el Helpdesk.

<http://www.dominio.com/helpdesk/admin/control.php> para la administración.

Más información y descargas en <http://www.oneorzero.com/>.

*Artículo por **Angel Deblás***

Generando feed RSS de noticias en PHP

Muchas veces nos hemos preguntado como generar feed RSS para nuestros blogs o sistema de noticias. Al igual que muchos, yo pensaba que era una tarea más que dura, pero con observación y un poco de ingenio todo se puede. A continuación explicaré una de las formas de generar RSS para nuestros sistemas. Necesitaremos que nuestras noticias se recojan de una base de datos.

El XML a veces es complejo de entender, para mi aún lo es, pero como dije: Con ingenio todo es posible. Lo primero es saber: ¿Qué etiquetas ocupan las distintas versiones de RSS?, pues fácil, nos vamos a algún sitio que genere dichos RSS, por ejemplo, un Blog y miramos el código fuente de dicho RSS. Ya tenemos aproximadamente el 90% del "trabajo" hecho, solo falta el código de nuestro documento en PHP. Hay que tener claro que el XML permite que creamos nuestras propias etiquetas, las que veremos a continuación, podemos tomarlas como standar

para los feeds.

Para el ejemplo se utilizará RSS 2.0:

```
<?
header('Content-Type: text/xml'); //Indicamos al navegador que es un documento en XML
//Versión y juego de caracteres de nuestro documento
echo <?xml version="1.0" encoding="iso-8859-1"?>;
//Aquí la conexión o archivo de conexión a la base de datos
//Hacemos la consulta y la ordenamos por post para mostrar siempre el último
$resultado=mysql_query("select * from noticias order by id_post Desc",$link);
$row=mysql_fetch_array($resultado);
//"Cortaremos" el artículo en 300 caracteres para nuestra descripción
$descripcion=substr($row[articulo],0,300)."...";
// Y generamos nuestro documento
echo '<rss version="2.0">
<channel>
  <title>Nombre de nuestro blog o web</title>
  <link>http://www.miurl.com/</link>
  <language>es-CL</language>
  <description>Descripción de nuestro blog o web</description>
  <generator>Autor del RSS</generator>
  <item>
<title>'.$row[titulo].'</title>
<link>http://www.miurl.com/noticias.php?id='.$row[id_post].'</link>
<comments>http://www.miurl.com/comentarios.php?id='.$row[id_post].'
</comments>
<pubDate>'.$row[fecha].'</pubDate>
<category>'.$row[categoria].'</category>
<guid>http://www.miurl.com/comentarios.php?id='.$row[id_post].'</guid>
<description><![CDATA['.$descripcion.']]></description>
<content:encoded><![CDATA['.$row[articulo].']]></content:encoded>
</item></channel></rss>';
?>
```

Podemos ver que la estructura del documento es sencilla y hasta repetitiva. Básicamente podemos dividirlo en dos: `<channel>` e `<item>` en donde `<channel>` contiene a `<item>`. El primer bloque antes de `<item>` se llena con los datos de nuestro blog o web, luego bajo la etiqueta mencionada van los datos del post o noticia a mostrar

Una práctica sana es utilizar la etiqueta `<content:encoded>`, para proporcionar la información que contiene nuestro artículo completo en vez de `<description>` ¿por qué?, porque los sistemas de recopilación de feeds o lectura de estos, leen la etiqueta `<description>` para darle al lector un resumen de lo que trata el post. Lo menciono, porque hay sistemas de blog que usan `<description>` para generar la información del artículo completo y no dan la opción de ver si queremos o no seguir leyendo lo que nos ofrecen, aparte de ocupar espacio. Para estas dos etiquetas usamos la sección XML `<![CDATA[]]>` (character data), básicamente para que el XML no interprete ciertos caracteres como propios de su lenguaje y trate de ejecutarlos.

Por último decirles, que deben cerrar bien todas las etiquetas del documento o noticia a la que se le hará feed, o dará un hermoso error de parseo.

En cuanto al script, se le pueden dar más opciones (aunque esto es lo básico), como por ejemplo: Hacer un while luego de `<item>` y mostrar todas las noticias, o la cantidad que quieras, usando `limit` en la llamada a la base.

*Artículo por **Alejandro Suazo***

Editor PHP Maguma Studio

Introducción:

El principal problema que tenemos con php es que no podemos ver paso a paso lo que esta haciendo el código con lo que se nos hace mucho más difícil descubrir los fallos que nos puedan surgir.

Una alternativa a esto es instalarnos un programa que nos permita realizar esto. Entre otros podemos destacar el PHP debugger (permite depurar programas escritos en php), que es Freeware y esta disponible en <http://www.nusphere.com>.

También tenemos el Maguma Studio Light for PHP, que permite escribir y depurar scripts de PHP y también es Freeware y lo tenemos disponible en <http://www.maguma.com>.

Instalación y configuración del Maguma:

Una vez descargado e instalado nos vamos a preferencias en View>Preferences o en Edit>Preferences.

Aquí le debemos indicar el intérprete de php. Es decir le debemos decir donde se encuentra el archivo php.exe. Este debe estar en la carpeta donde hemos descomprimido el php, lo buscamos y ponemos toda la ruta.

Será una página como esta:



Debajo de esta casilla en Server Setting le debemos poner las opciones del servidor web utilizado. Tal como:



Ahora debemos instalar y configurar el DBG, es decir el debugger.

Lo descomprimos y se habrá creado un directorio cuyo nombre indica la versión utilizada por el Maguma. Será algo como C:\dbg-2.11.3-Win32. Lo abrimos y copiamos el archivo dll del subdirectorio Server del DBG al subdirectorio extensions del PHP utilizado y lo renombramos:

```
Php_dbg.dll-4.2.3>php_dbg.dll.
```

Es decir le quitamos los números del final y solo dejamos la extensión dll.

Ahora para que el debugger nos funcione en el Maguma deberemos modificar algunas partes más.

- Buscamos el archivo php.ini (que estará en la carpeta Windows) y cambiamos la opción extensión_dir por:

```
Extensión_dir=c:\php-4.2.1-win32\extension\
```

- En la sección Windows Extensión, añadimos:

```
Extensión=php_dbg.dll
```

- Al final del archivo php.ini añadiremos:

```
[Debugger]
debugger.enable_session_cookie=off
debugger.enabled = on
debugger.enabled = on
debugger.fail_silently = on
debugger.ignore_nops=off
debugger.JIT_enabled = off
debugger.JIT_host= clienthost
debugger.JIT_level = 3
debugger.JIT_port = 7869
debugger.profiler_enabled = trae
debugger.sessioni_nocache = on
debugger.timeout_seconds=300
```

Una vez echo esto configuraremos el DBG, es decir; lo único que nos queda es arrancarlo para

que funcione. Para ello nos vamos a la carpeta C:\dbg-2.11.3-Win32 la abrimos y abrimos otra carpeta de dentro llamada "client". Aquí vemos unos cuantos archivos, el único que nos interesa es el install.bat. Lo ejecutamos y comprobamos que nos aparece una especie de antena parabólica al lado del reloj de Windows (parte derecha inferior).

Ahora abriremos el Maguma y volvemos a abrir la opción de preferencias que se encontraba en Edit>Preferences o en View>Preferenes y en la opción PHP-Parsing la ponemos como indica la figura



Nota: Podemos ponerlo como queramos, la diferencia es la velocidad que tendrá a la hora de ejecutar los scripts. Si lo ponemos como indica el dibujo será más rápido porque llamara directamente al intérprete de PHP y seleccionando la otra opción mostrara exactamente como saldrá la pagina.

Las dos funciones de debajo puede que no estén (según versiones) esto solo sirve para verlo desde nuestro ordenador o desde el servidor al que subimos la pagina. También es opcional y funciona de las dos formas (siempre que este configurado un servidor de Internet en el caso de que lo queramos mostrar externamente), pero si simplemente es para hacer pruebas lo podemos dejar como indica el dibujo.

Ahora podemos abrir una pagina en el Maguma y comprobar que funciona el editor. Para ello le daremos a:



y se arrancara el debugger.

Los siguientes botones son los que controlan el debugger. Son muy básicos:



1. arranca el debugger.
2. línea a línea.
3. de bloque en bloque.

4. hasta el final del script del php.
5. para el debugger.

*Artículo por **Agustin Jareño**.*

Mostrar Contenido de acuerdo al pais en php

Muchas veces, y sobre todo los webmaster internacionales, nos hemos encontrado con la necesidad de mostrar un contenido de acuerdo al pais y los idiomas.

En este Taller de PHP vamos a ver un sencillo código que permitirá conocer el país del usuario y mostrar una página distinta para cada caso.

```
<?php
// cambia pagina.php por el archivo correcto de acuerdo al pais

if (isset($pais) && ($pais <> "")) {
$dgo_pais = substr($pais,0,5);
} else {
$dgo_pais = substr($_SERVER["HTTP_ACCEPT_LANGUAGE"],0,5);
}
switch ($dgo_pais) {

case "es-mx":
//si el pais es mexico
include("pagina.php");
break;

case "es-ar":
//si el pais es argentina
include("pagina.php");
break;

case "es-cl":
//si el pais es chile
include("pagina.php");
break;

case "es-ve":
//si el pais es venezuela
include("pagina.php");
break;

case "pt-br":
//si el pais es brasil
include("pagina.php");
break;

case "es-co":
//si el pais es colombia
include("pagina.php");
break;

case "es-ec":
//si el pais es ecuador
```



```
include("pagina.php");
break;

case "es-uy":

//si el pais es uruguay
include("pagina.php");
break;

default:
//si es algun otro pais
include("pagina.php");
break;
}
?>
```

Recuerda cambiar pagina.php por la pagina de cada pais, por ejemplo, pagina-mexico.php. Se puede usar este script como pagina principal, para que sea distinta en función del país.

*Artículo por **Aaron Gomez Perez***

Blasten blt-SEARCH 1.0.5

En este artículo vamos a presentar Blasten blt-SEARCH, un sistema de búsqueda e indexación de contenidos escrito en PHP y que utiliza base de datos MySQL. Se trata de un producto gratuito y con interesantes posibilidades, que se muestra muy útil para proveer de herramientas búsqueda dentro de nuestro sitio web, o de otras páginas.

Una de las partes fundamentales de blt-SEARCH es un sistema escaneador de enlaces, que pondera los resultados por importancia y que permite ordenar al instante todos los enlaces encontrados en la búsqueda. Para ello calcula la similitud entre el criterio buscado y los contenidos indexados, basándose en la importancia de la página y consigue asignar una posición adecuada, acorde a la cantidad de enlaces referidos durante el proceso de indexación.

A la hora de indexar los contenidos de los sitios web, es capaz de excluir ciertas URL que superen una cantidad dada de parámetros enviados a través de HTTP-GET. Si así lo deseamos, también puede eliminar parámetros sin necesidad de suprimir la exploración a esa página. Lo que se traduce en una gran ventaja para sitios generados dinámicamente a través de cualquier tecnología. Asimismo puede iniciar conexiones a páginas de acceso restringido con autenticación HTTP.

Tiene también capacidad para denegar el seguimiento de enlaces a determinados dominios indicados por usted, o denegar todos.

Dispone de otras características adicionales. Entre las que cabría destacar:

- Una biblioteca de funciones, que permiten obtener información confidencial de cada documento indexado, que son utilizadas por blt-SEARCH para generar resultados.
- La posibilidad de obtener resultados aún mucho más específicos y avanzados, buscado en URLs definidas por el usuario.
- Arroja sugerencias totalmente exactas y efectivas cuando no sea encontrado con exactitud un criterio en la base de datos

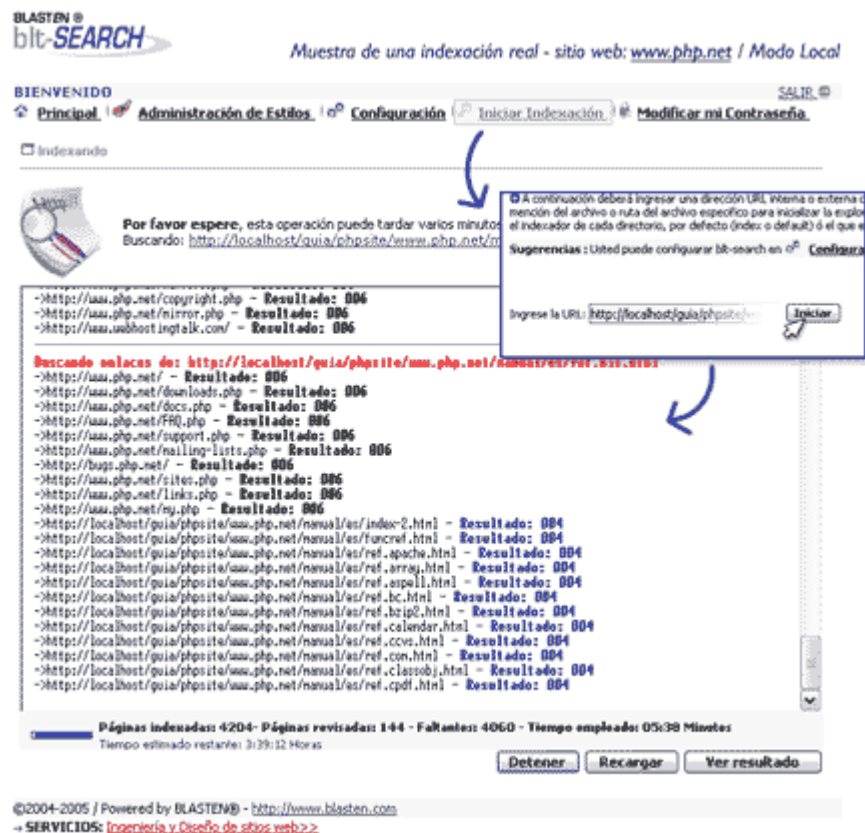
- Se pueden definir una serie de palabras comunes para excluirlas en las búsquedas, lo que mejora su efectividad
- Es capaz de leer todos los metas tags y robots.txt de las páginas a indexar y trabajar acorde a lo que ellos indiquen.
- Indexa también la totalidad de los marcos flotantes interpretados a través de las etiquetas <frame> e <iframe> respectivamente.
- Puede mostrar un mapa de todo el sitio web explorado y los elementos que lo constituyen.
- Es posible indexar sitios personales individuales o múltiples.
- No solo indexa páginas web, sino también imágenes en formatos (Jpeg, Gif, Png y Wbmp), de las que extrae información como el tamaño de la imagen, dimensiones, formato, Url, etc.

A la hora de mostrar los resultados, es posible configurar una serie de opciones y parámetros, para adaptarlos a las necesidades de cada sitio web.

Conclusión

No cabe duda que los sistemas de búsqueda son una parte importante dentro de sitios web grandes. blt-SEARCH nos facilita la tarea de encontrar contenidos de forma rápida y precisa, y eso es muy de agradecer. Este sistema es gratuito y adaptable a muchos casos distintos, por lo que seguro que puede resultar muy útil a los administradores de sitios web.

Se puede obtener más información en la [página del producto](#) o en su [área de documentación](#).



BLASTEN®
blt-SEARCH

Muestra de una indexación real - sitio web: www.php.net / Modo Local

BIENVENIDO

Principal | Administración de Estilos | Configuración | Iniciar Indexación | Modificar mi Contraseña

Indizando

Por favor espere, esta operación puede tardar varios minutos.
Buscando: <http://localhost/guia/phpsite/www.php.net/>

A continuación deberá ingresar una dirección URL interna o externa o mención del archivo o ruta del archivo específico para inicar la exploración del indexador de cada directorio, por defecto (index o default) o el que usted desee.
Sugerencias: Usted puede configurar blt-search en Configuración

Ingrese la URL: Iniciar

Buscando enlaces de: <http://localhost/guia/phpsite/www.php.net/administracion/estilos.css>

- ><http://www.php.net/copyright.php> - Resultado: 006
- ><http://www.php.net/mirror.php> - Resultado: 006
- ><http://www.usdhostingtalk.com/> - Resultado: 006
- ><http://www.php.net/> - Resultado: 006
- ><http://www.php.net/downloads.php> - Resultado: 006
- ><http://www.php.net/docs.php> - Resultado: 006
- ><http://www.php.net/FRO.php> - Resultado: 006
- ><http://www.php.net/support.php> - Resultado: 006
- ><http://www.php.net/mailling-lists.php> - Resultado: 006
- ><http://bugs.php.net/> - Resultado: 006
- ><http://www.php.net/sites.php> - Resultado: 006
- ><http://www.php.net/links.php> - Resultado: 006
- ><http://www.php.net/ny.php> - Resultado: 006
- ><http://localhost/guia/phpsite/www.php.net/manual/es/index-2.html> - Resultado: 004
- ><http://localhost/guia/phpsite/www.php.net/manual/es/funcionf.html> - Resultado: 004
- ><http://localhost/guia/phpsite/www.php.net/manual/es/ref.apache.html> - Resultado: 004
- ><http://localhost/guia/phpsite/www.php.net/manual/es/ref.array.html> - Resultado: 004
- ><http://localhost/guia/phpsite/www.php.net/manual/es/ref.spoll.html> - Resultado: 004
- ><http://localhost/guia/phpsite/www.php.net/manual/es/ref.bc.html> - Resultado: 004
- ><http://localhost/guia/phpsite/www.php.net/manual/es/ref.bzip2.html> - Resultado: 004
- ><http://localhost/guia/phpsite/www.php.net/manual/es/ref.calendar.html> - Resultado: 004
- ><http://localhost/guia/phpsite/www.php.net/manual/es/ref.cvs.html> - Resultado: 004
- ><http://localhost/guia/phpsite/www.php.net/manual/es/ref.com.html> - Resultado: 004
- ><http://localhost/guia/phpsite/www.php.net/manual/es/ref.classobj.html> - Resultado: 004
- ><http://localhost/guia/phpsite/www.php.net/manual/es/ref.cpdf.html> - Resultado: 004

Páginas indexadas: 4204 - Páginas revisadas: 144 - Fallos: 4060 - Tiempo empleado: 05:38 Minutos
Tiempo estimado restante: 3:29:12 Horas

Detener Recargar Ver resultado

©2004-2005 / Powered by BLASTEN® - <http://www.blasten.com>
- SERVICIOS: [Ingeniería y Diseño de sitios web](#) >>

Imagen de una de las pantallas del sistema

Artículo por **Emmanuel García De Caro**

Clases y Objetos en PHP

Una clase es una colección de variables y de funciones que acceden a esas variables:

```
class Cart {
    var $items; // Items en nuestro carro de la compra

    // Añadir $num artículos de tipo $artnr al carro

    function add_item ($artnr, $num) {
        $this->items[$artnr] += $num;
    }

    // Sacar $num artículos del tipo $artnr del carro

    function remove_item ($artnr, $num) {
        if ($this->items[$artnr] > $num) {
            $this->items[$artnr] -= $num;
            return true;
        } else {
            return false;
        }
    }
}
```

El ejemplo define una clase llamada Cart que consiste en un array asociativo de artículos en el carro y dos funciones para meter y sacar ítems del carro

Las clases son tipos, es decir, son plantillas para variables. Tienes que crear una variable del tipo deseado con el operador new.

Este ejemplo crea un objeto \$cart de clase Cart. La función add_item() de ese objeto se llama para añadir un ítem del artículo número 10 al carro.

Las Clases pueden ser extensiones de otras clases. Las clases extendidas o derivadas tienen todas las variables y funciones de la clase base y lo que les añadas al extender la definición. La herencia múltiple no está soportada.

```
class Named_Cart extends Cart {
    var $owner;

    function set_owner ($name) {
        $this->owner = $name;
    }
}
```

Ese ejemplo define una clase Named_Cart (carro con nombre o dueño) que tiene todas las variables y funciones de Cart, y además añade la variable \$owner y una función adicional set_owner(). Un carro con nombre se crea de la forma habitual y, una vez hecho, puedes acceder al propietario del carro. En los carros con nombre también puedes acceder a las funciones normales del carro: +

```
$ncart = new Named_Cart; // Creamos un carro con nombre
```

```
$ncart->set_owner ("kris"); // Nombramos el carro
print $ncart->owner; // Imprimimos el nombre del propietario
$ncart->add_item ("10", 1); // Funcionalidad heredada de Cart
```

Entre funciones de una clase, la variable `$this` hace referencia al propio objeto. Tienes que usar `$this->loquesea` para acceder a una variable o función llamada `loquesea` del objeto actual.

Los constructores son funciones de una clase que se llaman automáticamente al crear una nueva instancia (objeto) de una clase. Una función se convierte en constructor cuando tiene el mismo nombre que la clase.

```
class Auto_Cart extends Cart {
    function Auto_Cart () {
        $this->add_item ("10", 1);
    }
}
```

Este ejemplo define una clase `Auto_Cart` que es un `Cart` junto con un constructor que inicializa el carro con un item del tipo de artículo "10" cada vez que se crea un nuevo `Auto_Cart` con "new". Los constructores también pueden recibir parámetros y estos parámetros pueden ser opcionales, lo que los hace más útiles.

```
class Constructor_Cart extends Cart {
    function Constructor_Cart ($item = "10", $num = 1) {
        $this->add_item ($item, $num);
    }
}
```

```
// Compramos las mismas cosas aburridas de siempre
```

```
$default_cart = new Constructor_Cart;
```

```
// Compramos las cosas interesantes
```

```
$different_cart = new Constructor_Cart ("20", 17);
```

Comentario: En nuestro manual de PHP contamos con detalle el trabajo con [objetos y clases](#). Por otra parte, también se comenta en el manual de [PHP 5](#).

*Artículo por **Emmanuel García De Caro***

Recuperando la calidad de una imagen JPEG

Buscando en php.net he conseguido una función que permite regenerar prácticamente una imagen con su color real, una de las principales deficiencias cuando usamos sistemas de Thumbnail.

Esta función es: **`imagecreatetruecolor($ancho, $alto)`**; El resto del proceso ya lo conocemos.

```
header("Content-type: image/jpeg");
$src_img= @imagecreatefromjpeg('LA IMAGEN.JPG')
```

```
$dst_img = @imagecreatetruecolor($ANCHO,$ALTO);  
  
@imagecopyresized($dst_img, $src_img, 0,0,0,0, $ANCHO, $ALTO, ImageSX($src_img), ImageSY($src_img));  
  
@imagejpeg($dst_img);  
@imagedestroy($dst_img);
```

Explicación:

En primer lugar enviamos una cabecera pura con el tipo de documento:

header();

Guardamos en la variable `$src_img` una nueva imagen creada de tipo JPEG a partir de LA IMAGEN.JPG que será la imagen que redimensionaremos o la editaremos.

Luego en la variable `$dest_img`, Creamos una imagen nueva con color REAL, esta será la que utilizaremos para mostrarla, nótese que no esta definidas las variables `$ANCHO`, `$ALTO`. Estas variables guardan el nuevo tamaño de la imagen que obviamente sera inferior a la original, por lo cual pueden ayudarse con la función `getsizeimage()`; que bueno devuelve un vector con los píxeles de X y Y , o en su efecto `ImageSX()` y `ImageSY()` para obtener en base a esos parámetros el nuevo tamaño uniforme y acorde con nuestra galería de imágenes.

`Imagecopyresized()`, bueno el nombre lo dice, copia solo partes de una imagen REDIMENSIONADA .

Luego mostramos la imagen con `imagejpeg()`; si queremos que se guarde en el directorio con sus respectivos permisos, agregaremos un nuevo parámetro:

```
@imagejpeg($dst_img,'NUEVAIMAGEN.JPG');
```

Destruimos la imagen para ahorrar memoria utilizada por `imagecreatefromjpeg()` y los procesos subsiguientes;

Y bueno, como ya sabemos, la @ (arroba) al comienzo de cada función sirve para evitar que se imprima el error en pantalla. Si es de su preferencia pueden quitarlas.

*Artículo por **Emmanuel García De Caro***

Comprobar si existe el dominio del correo

En Php existen varias funciones que nos permite obtener la veracidad de direcciones URL, DNS o IPS.

Una de estas funciones, **checkdnsrr()**, comprueba registros correspondientes a nombres de maquinas o direcciones IP.

Algo que puede resultar muy útil si queremos comprobar, si existe el dominio del correo aportado por el usuario, Sin embargo solo comprueba el dominio, no si la cuenta de correo existe.

Su sintaxis es la siguiente:

checkdnsrr (string host [, string type])

Busca en DNS entradas del tipo type correspondientes a host. Devuelve verdadero(TRUE) si encuentra algún registro; devuelve false(FALSO) si no encuentra ninguno o sucedió algún error.

type puede ser: A, MX, NS, SOA, PTR, CNAME, o ANY. Por defecto es MX.

El código es el siguiente:

```
<?? <?php
function validar_email($email){
    $exp = "^([a-z\0-9]+([._-][a-z\0-9]+)*@[a-z0-9]+([._-][a-z0-9]+)+)$";
    if(eregi($exp,$email)){
        if(checkdnsrr(array_pop(explode("@",$email)),"MX")){
            return true;
        }else{
            return false;
        }
    }else{
        return false;
    }
}
// para aplicarla
if (validar_email($_GET[correo])){
// coloco las instrucciones
echo 'Es correcto el correo';
}else{
echo 'Es falso el correo';
}
?>
```

Explicación:

Lo primero es definir nuestra función, que tendrá como nombre validar_email(). Utilizaremos Expresiones regulares para buscar las coincidencias y similitudes entre la expresión (\$exp) y el valor de (\$email).

Ver: <http://www.php.net/eregi>

Si esta condición evalúa TRUE entonces quiere decir que efectivamente \$email, Comienza por cualquier letra que vaya de [a-z] , [0-9], puede contener un guión, punto u otro, seguido de una @(arroba como obligatorio) y luego se repetiría el proceso para el nombre del dominio, y finalmente un (.) PUNTO como obligatorio, donde continuaría la extensión del dominio como final de la cadena.

Ahora ejecutamos la función **checkdnsrr()**, Como ya dije anteriormente siempre y cuando eregi, tome el valor TRUE, Con la función explode() dividimos el valor de \$email en un ARRAY o MATRIZ, tomando como punto de separación la @ ARROBA. Recordemos que ahora tenemos un Array donde [0] es el usuario del correo y [1] es el domino del correo. Para luego obtener este ultimo valor [1] de la matriz con array_pop().

El siguiente parámetro corresponde a buscar las entradas de tipo MX que es el default de esta función.

Acorde a todas estas condiciones retornamos Booleanos según sea el caso:

```
return( TRUE/ FALSE );
```

Un ejemplo:

Si alguien escribe: nosess@hotmail.com, nuestra función nos devolverá TRUE, recordemos que esta función nos comprueba si existe el host : >>> Hotmail.com , más no el nombre de usuario nosse...

Artículo por Emmanuel García De Caro

Tutorial de BBCode

Introducción

BBCode es una implementación especial del HTML. Nos va a servir para simplificar código HTML de modo que nuestros usuarios puedan utilizarlo en al enviar noticias a nuestra web, dejar sus mensajes en nuestro foro, etc. Lo usaremos en todo tipo de formularios en los que queramos que el usuario pueda insertar enlaces, imágenes, tablas, imoticonos, etc. de forma sencilla y sin necesidad de tener conocimientos sobre HTML.

El BBCode se utiliza de una forma similar al lenguaje HTML: Se utilizan etiquetas entre corchetes de la siguiente manera: Usaremos [etiqueta] para abrir la etiqueta y [/etiqueta] para cerrarla.

Muchos conocidos sistemas de foros y plataformas de gestión de contenidos web utilizan BBCode. Por ejemplo [phpBB](#) ó [SMF](#)

Vamos a ver el esquema de envío de texto + BBCode para incluir una imagen. El usuario escribiría en el textarea de un formulario:

```
[imagen]http://www.mi_servidor.com/mi_imagen.jpg[/imagen]
```

Y al enviarlo reemplazaríamos mediante una función en PHP las etiquetas [imagen] e [/imagen] por código HTML, quedando de la siguiente manera:

```

```

Esto nos permite jugar como queramos con las etiquetas. Si usamos una base de datos podemos guardar los mensajes enviados en modo texto y BBCode tal como han sido redactados y para mostrarlos utilizar una función PHP para reemplazar el código BBCode por el código HTML correspondiente.

Creando nuestros BBcodes con PHP

Definiremos las etiquetas BBCode que nosotros queramos con el código HTML que les corresponda y crearemos una función con PHP para sustituir los BBcodes escritos por el usuario por código HTML. Utilizaremos la función STR_REPLACE() de PHP para reemplazar las etiquetas BBcode por HTML.

Primero asociaremos código HTML a etiquetas elementales de BBCode mediante un array. Lógicamente podéis crear vuestras propias etiquetas con el código HTML que queráis:

```
<?php

$bb_code = array(
// emoticonos: debéis apuntar a vuestras imágenes en el código HTML
':)' => '',
':(' => '',
':D' => '',

// letra negrita
'[b]' => '<span style="font-weight:bold">',
'[b]' => '</span>',

// letra cursiva
'[i]' => '<span style="font-style:italic">',
'[i]' => '</span>',

// letra subrayada
'[u]' => '<span style="text-decoration:underline">',
'[u]' => '</span>',

// salto de línea
'[salto]' => '<br><br>',

// imagenes
'[imagen]' => '<img src="",
'[imagen]' => "" />'

// recordad que después del último elemento no hay coma
);
?>
```

Ahora utilizaremos una función para reemplazar las etiquetas BBCode por código HTML:

```
<?php
function reemplazar( $codigo )
{
$search = array_keys( $GLOBALS['bb_code'] );
$codigo = str_replace( $search, $GLOBALS['bb_code'], $codigo );
return $codigo;
}
?>
```

En este ejemplo \$codigo es la variable en la que pasaremos a la función con el texto+BBCode que haya escrito el usuario.

Ahora vamos a ver el código completo listo para ser probado que incluye los códigos anteriores con un texto+BBCode de ejemplo:

```
<?php

// definimos nuestras etiquetas
$bb_code = array(
// emoticonos: debéis apuntar a vuestras imágenes en el código HTML
':)' => '',
':(' => '',
':D' => '',

// letra negrita
'[b]' => '<span style="font-weight:bold">',
'[/b]' => '</span>',

// letra cursiva
'[i]' => '<span style="font-style:italic">', < '[/i]' => '</span>',

// letra subrayada
'[u]' => '<span style="text-decoration:underline">',
'[/u]' => '</span>',

// salto de línea
'[salto]' => '<br><br>',

// imagenes
'[imagen]' => '<img src="",
'[/imagen]' => "" />'

// recordad que después del último elemento no hay coma
);

// Definimos nuestro mensaje de prueba
$mensaje = 'Esta palabra está en [b]negrita[/b] y [u]esto está subrayado.[/u] '
.'[salto]Este es el logo de desarrolloweb: [salto]'
.'[imagen]http://www.desarrolloweb.com/imagenes/desarrollo_web.gif'
.'[/imagen]';

// Reemplazamos el BBCode por código HTML y lo mostramos en la página
echo '<p>'.reemplazar( $mensaje ).</p>';

// Incluimos la función para reemplazar el código
function reemplazar( $codigo )
{
$search = array_keys( $GLOBALS['bb_code'] );
$codigo = str_replace( $search, $GLOBALS['bb_code'], $codigo );
return $codigo;
}
?>
```

Para trabajar con formularios sólo tenemos que reemplazar \$mensaje por el valor enviado a través del formulario. Por ejemplo pondríamos un campo textarea llamado "mensaje" en el formulario que apuntara a nuestro script PHP y en éste cambiaríamos el valor de \$mensaje por \$HTTP_POST_VARS['mensaje']

Espero que este tutorial haya sido de utilidad para más de uno.

*Artículo por **Antonio Melé***

Mambo, sistema gestor de contenidos

Mambo Open Source es un CMS (Content Manager Service) profesional muy fácil de usar, tanto para páginas sencillas como aplicaciones más complejas. Tiene incluidas características como editores de contenido WYSIWYG, noticias, banners, administración de enlaces, estadísticas, archivo de contenidos, contenido de bases de datos, 20 idiomas, módulos, componentes, etc.

A destacar que Mambo es software libre, con licencia GNU/GPL, es decir que podemos utilizarlo para todos nuestros desarrollos sin límite alguno. Muy importante, sobre todo si estamos empezando un proyecto y no podemos permitirnos un gasto grande en un desarrollo de una aplicación a medida. Este gestor de contenidos, gracias a su facilidad de uso, nos va a permitir, con unos mínimos conocimientos, la creación de nuestra web corporativa de una manera fácil e intuitiva.

No es el típico constructor de portales. Está más orientado a la creación de webs personales y corporativas, su fortaleza está en un núcleo muy estable antes que en la cantidad de paquetes que se puedan añadir para dar más funcionalidad. Es decir, si preferimos estabilidad, escoge Mambo. Ello no quiere decir que no se puedan utilizar añadidos, sino que el equipo de desarrollo se orienta más a la estabilidad y los add-on, suelen correr por cuenta de terceras personas, que desinteresadamente colaboran con el proyecto para mejorar la funcionalidad sobre una base muy estable.

Funcionalidades soportadas

Algunas de las características de este sistema son las siguientes:

- Nivel básico de aprobación de contenidos para usuarios certificados.
- Ayuda Online.
- Mecanismo caching para mejorar el funcionamiento de sitios saturados.
- Gerente de basura.
- Dirección publicitaria (banners, etc.).
- Medios de comunicación (imágenes, documentos) upload y dirección.
- Planificación de demostración de contenidos.
- Distribución de contenidos mediante RSS.
- Búsqueda de URL'S.
- Internacionalización (traducción de interfaz).
- Macro-lenguaje.
- Sistema avanzado de administración del sistema.
- Sistema de plantilla simple pero poderoso (escrito sobretodo en HTML)
- Estadística de visitantes.
- Múltiple redactor de contenido.
- Múltiples paquetes añadidos y plantillas (package/addon/template).
- Autenticación LDAP.
- Perfiles de usuario ampliados.
- Foros, Galerías media, Gerentes de documento, Plantillas, Calendarios de eventos, etc.

Requerimientos

Respecto a los requisitos necesarios, decir que Mambo trabaja sobre linux, freeBSD, Mac OS X, Windows NT, 2000 y XP. Se recomienda un servidor Apache, incluso para Windows, aunque sobre IIS también podría funcionar.

Por supuesto, necesitaremos que Apache sirva páginas en PHP. Por lo menos debemos tener una versión de PHP 4.1.2 (o superior). También se requiere disponer de MySQL, aunque en

este caso no nos informan de la versión mínima requerida.

Para obtener más información, podemos visitar el propio sitio de Mambo:
www.mamboserver.com/.

También podemos encontrar información en algunas páginas web dedicadas a este sistema gestor de contenidos, como por ejemplo:

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

<http://www.mambohispano.org/> (Esta página está en castellano, pero se encuentra fuera de servicio en el momento de escribir el artículo)

Plantillas para Mambo

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

<http://www.mamplates.net/>

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

<http://www.peekmambo.com>

*Artículo por **Diego Pinilla***

Función en PHP para el cálculo de minutos transcurridos en dos horas dadas

Se trata de una función que nos permite obtener El total de Minutos u Horas transcurridas entre dos Tiempos de un mismo día.

Algo como:

Hora 1= 13:56

Hora 2=18:07

La función Imprime: Han transcurrido 04:11 Horas.

El proceso es muy sencillo, veamos el código fuente:

```
<? function calcular_tiempo_trasnc($hora1,$hora2){
    $separar[1]=explode(':', $hora1);
    $separar[2]=explode(':', $hora2);

    $total_minutos_transcurridos[1] = ($separar[1][0]*60)+$separar[1][1];
    $total_minutos_transcurridos[2] = ($separar[2][0]*60)+$separar[2][1];
    $total_minutos_transcurridos = $total_minutos_transcurridos[1]-$total_minutos_transcurridos[2];
    if($total_minutos_transcurridos<=59) return($total_minutos_transcurridos.' Minutos');
    elseif($total_minutos_transcurridos>59){
        $HORA_TRANSCURRIDA = round($total_minutos_transcurridos/60);
        if($HORA_TRANSCURRIDA<=9) $HORA_TRANSCURRIDA='0'.$HORA_TRANSCURRIDA;
        $MINUITOS_TRANSCURRIDOS = $total_minutos_transcurridos%60;
        if($MINUITOS_TRANSCURRIDOS<=9) $MINUITOS_TRANSCURRIDOS='0'.$MINUITOS_TRANSCURRIDOS;
        return ($HORA_TRANSCURRIDA.':'.$MINUITOS_TRANSCURRIDOS.' Horas');
    }
}
//llamamos la función e imprimimos
echo calcular_tiempo_trasnc(date('H:i'),'16:12');
?>
```

Explicando:

Primero definiremos una función para no repetir el mismo código en todas las partes donde lo utilizaremos, De nombre `calcular_tiempo_trasnc($hora1,$hora2)`;

Donde: `$hora1` es la hora mayor desde donde se calculara; en el ejemplo anterior seria 18:07, Y `$hora2` es el punto desde donde se calculara las horas y minutos transcurridos.

Cabe destacar que ambas horas deben estar en el siguiente formato: (H: i) .
Como ya sabemos "H" es la hora de 00 a 23 e "i", Los minutos de 00 a 59 parámetros definidos para la función `date()`;

Luego asignamos valor a las matrices `$separar[1]` y `$separar[2]` respectivamente con el resultado devuelto por la función `explode()`; encargada de separar una cadena a partir de un valor, en este caso será : (dos puntos) devolviendo una nueva matriz con dos partes, donde:

`$separar[1][0]` <= Es la hora de `$hora1`. Si el valor de `$hora1` es igual a (11:15), entonces `$separar[1][0]` es igual a "11";
Y `$separar[1][1]` <= Los minutos de `$hora 1`, igual a 15.

Donde el primer [1], es el Key o la clave de `$separar`, y correspondiente a la `$hora1`, si fuese 2, seria la `$hora2`. Y hasta allí porque solo están definidos 2 parámetros.

Pasaremos los valores de `$separar[1][0]` y `$separar[2][0]` respectivamente de horas a minutos (Multiplicando ese valor por 60) que son los minutos de una hora.

```
$separar[1][0]*60
```

Para Luego sumarle los minutos devueltos por `$separar[1][1]` y `$separar[2][1]` respectivamente, De este modo obtenemos el valor global de los minutos transcurridos en ambas horas. Los resultados los agruparemos en `$total_minutos_transcurridos[1]` y `$total_minutos_transcurridos[2]`.

El paso siguiente es asignarle el valor de la resta entre `$total_minutos_transcurridos[1]` y `$total_minutos_transcurridos[2]` a la variable `$total_minutos_transcurridos`.

Ya tenemos la diferencia en Minutos de ambos cortes de horas. Ahora debemos darle un formato legible y entendible.

Iniciamos una condición donde decimos: Si `$total_minutos_transcurridos` es menor o igual a 59(`if($total_minutos_transcurridos<=59)`) quiere decir que aun no ha transcurrido 1 hora , por lo tanto retornamos (`return($total_minutos_transcurridos.' Minutos')`), que nos es más que el mensaje que devolverá toda la función cuando se llame.

De lo contrario si (`elseif($total_minutos_transcurridos>59)`) `$total_minutos_transcurridos` es mayor que 59, entonces si ha transcurrido al menos 1 hora y ejecutamos las siguientes instrucciones:

Llevaremos los minutos a horas, dividiéndolo por 60 (minutos de una hora) y redondeamos el valor con `round()`; Ya tenemos la Hora, pero nos falta los minutos.

Para los minutos (`$MINUTOS_TRANSCURRIDOS`), obtendremos el residuo de la división entre `$total_minutos_transcurridos` y 60 (minutos de una hora).

```
$MINUTOS_TRANSCURRIDOS=$total_minutos_transcurridos%60
```

Ya tenemos los valores deseados.

Ahora indicaremos si el valor de \$HORA_TRANSCURRIDA ó \$MINUTOS_TRANSCURRIDOS es menor o igual que 9, para colocarle un cero a la izquierda. Por ejemplo: si \$MINUTOS_TRANSCURRIDOS es igual a 9 entonces la condición evalúa TRUE y le coloca el 0 al comienzo, devolviendo 09.

El mismo proceso lo repetimos con \$HORA_TRANSCURRIDA.

Finalmente retornamos el resultado:

```
return ($HORA_TRANSCURRIDA.':'.$MINUTOS_TRANSCURRIDOS.' Horas');
```

Cabe destacar que pueden adaptarle unas instrucciones para que cuando Los minutos sean igual a 1 diga (Minuto) y no (Minutos), igual que con las horas.

Para llamar la función: `calcular_tiempo_trasnc(date('H:i'),'16:12');`

Donde `date('H:i')`, es la hora actual del servidor y `16:12` desde donde se comienza a contar.

*Artículo por **Emmanuel García De Caro***

Contenidos en base de datos con PHP

En esta ocasión tengo la oportunidad de explicar algo extremadamente sencillo en PHP, como lo es colocar nuestros contenidos en una base de datos y llamarlos desde una URL como esta: `contenidos.php?id=6565`.

De esta manera, nosotros simplemente vamos a crear una página web, que recibe por parámetro el contenido que se debe mostrar, en lugar de tener que realizar una página para cada uno de los contenidos del sitio.

Nota: Cabe destacar el inconveniente que hasta cierto punto se presenta con este tipo de URL con los principales robots recopiladores de documentos html en Internet. Pues los buscadores como Google no tratan muy bien a las URLs que reciben datos por parámetro en la propia URL.

Crearemos una tabla de datos:

```
<? CREATE TABLE `contenidos` (  
  `id` int(11) NOT NULL auto_increment,  
  `titulo` varchar(50) NOT NULL default "",  
  `fecha` date NOT NULL default '0000-00-00',  
  `descripcion` text NOT NULL,  
  PRIMARY KEY (`id`)  
)TYPE=MyISAM AUTO_INCREMENT=1 ;
```

Luego el código Fuente:

```
<?PHP  
#####CONEXIÓN A MYSQL  
@mysql_connect('localhost','user','pass')or die ('Ha fallado la conexión: '.mysql_error());  
@mysql_select_db('base_de_datos')or die ('Error al seleccionar la BD: '.mysql_error());  
///end conexión  
  
$ssql_=mysql_query("SELECT * FROM contenidos WHERE id='".$$_GET[id]."'")or die(mysql_error());  
if(mysql_num_rows($ssql_)==0) die ('Error: intentas acceder a un artículo o contenido inexistente');  
$obj=mysql_fetch_object($ssql_);
```

```
echo 'Titulo: <strong>'.$obj->titulo.'</strong> Fecha: '.$obj->fecha.'<br> Descripción: '.nl2br(strip_tags($obj->descripcion)); ?>
```

Explicando:

Primero que nada como de costumbre, creamos una conexión al servidor Mysql para que tenga efecto en las líneas posteriores del documento Php.

Obviamente debemos modificar los datos (localhost, user, pass y 'base_de_datos' por los existentes.

Luego enviamos una petición al servidor Mysql con la función mysql_query, con una consulta que traducida al español seria algo como:

SELECCIONAR TODOS LOS CAMPOS DE contenido DONDE id SEA IGUAL A \$_GET[id]

Una vez generada la respuesta, Con la función mysql_num_rows(), extraemos el total de registros que debe ser siempre 1, y utilizamos este valor como elemento indispensable para iniciar una condición elemental que generara un error con la función die(); siempre y cuando no haya ningún registro.

```
if(mysql_num_rows($ssql)==0).
```

La función die(); se encarga de imprimir un mensaje, pero además detiene hasta ese punto la ejecución del script, por lo cual es innecesario utilizar ("else" De lo contrario a si (if)) Condicional.

Agrupamos en la variable \$obj, la fila devuelta por la consulta mysql_fetch_object () "COMO OBJETO", si queremos hacerlo como array debemos utilizar mysql_fetch_array(), en vez de mysql_fetch_object(), la diferencia radica esencialmente en el tipo de valor (Objeto / Matriz) siendo el resultado igual.

Para imprimir nuestros datos utilizáramos:

Con mysql_fetch_object = \$obj->nombre_del_campo , similar a como llamamos las variables o funciones de una clase.

Con mysql_fetch_array() = \$obj[nombre_del_campo]. Finalmente imprimimos el resultado, con los valores extraídos en procesos anteriores, utilizando la función strip_tags(); para extraer todas la entidades de html si las tuviera, y nl2br(); para agregar un salto de linea al final de la misma.

Ya quedaría de parte de cada quien, crear el diseño de su página dinámica ó diagramar de una manera más exacta y personalizada la tabla de datos con los posteriores elementos definidos en el script.

*Artículo por **Emmanuel García De Caro***

Creación de gráficas en PHP con JpGraph

Una tarea, a la que más tarde o más temprano vamos a tener que enfrentarnos a lo largo de nuestra andadura como profesionales del web, es la creación de gráficas a partir de datos. Cuando hablo de gráficas me refiero a todo tipo de imágenes que sirvan para representar datos, como gráficas de barras, de líneas de progreso, de tarta, etc.

Obviamente, la creación de gráficas no es un tema trivial, sino que requerirá de una gran dosis de dedicación y esfuerzo. Las gráficas, que generalmente se muestran en imágenes, pueden ser de muchos tipos distintos y sólo el hecho de tratar de dibujar en una imagen líneas, barras o incluso tartas en tres dimensiones, puede ser sobradamente complicado.

Sin embargo, existen sistemas como JpGraph, que nos pueden facilitar la tarea de una manera muy interesante, pues ofrecen una serie de mecanismos para la generación de las imágenes con las gráficas, de modo que nosotros sólo tenemos que centrarnos en cargar los datos a representar y escoger el tipo de gráfica que deseamos visualizar.

Qué es JpGraph

Es una librería que incluye una serie de clases -código orientado a objetos- que sirven para crear imágenes con todo tipo de gráficas, dinámicamente desde páginas PHP.

El sistema está muy depurado y soporta multitud de funcionalidades, por lo que seguramente encontraremos solución a casi cualquier necesidad en el ámbito de creación de gráficas. Además, la mayoría de las configuraciones de las gráficas vienen con opciones por defecto, así que resulta bastante sencillo obtener resultados rápidamente.

Algunas de las características del sistema son:

- Reducido peso en bytes de las imágenes resultado. Habitualmente unas pocas KB.
- Soporte a las librerías GD1 o GD2.
- Uso de la Interpolación matemática para obtener curvas a partir unos pocos valores.
- Diversos tipos de gráficas 2D o 3D, como de puntos, líneas, tartas, barras, cajas...
- Escalas flexibles tanto en el eje X como el Y, que se ajustan al juego de datos que se tenga que representar.
- Soporte para generar gráficas con varios juegos de valores a la vez.
- Configurable con distintos tipos de colores, leyendas, tipografías, imágenes de fondo, etc.

Cómo usar JpGraph

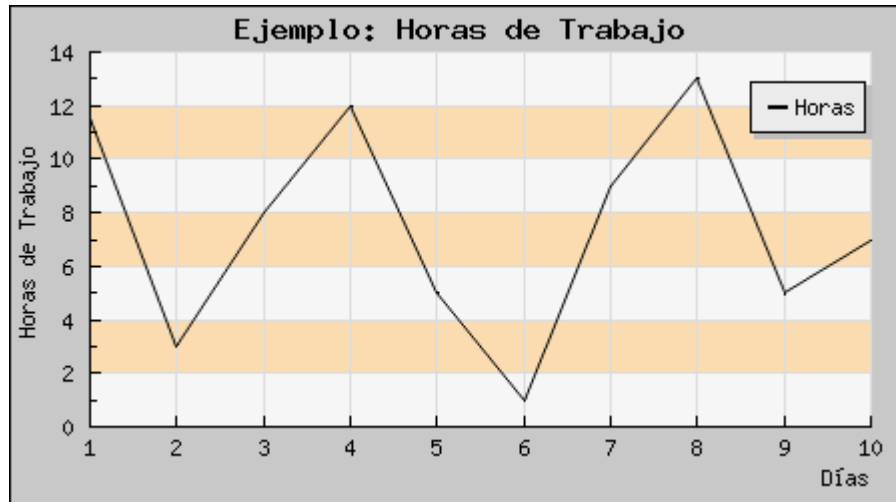
Este juego de librerías dispone de una extensa documentación y tutoriales para aprender a manejarlo. En la documentación se encuentran además numerosos ejemplos de su uso, desde los que podemos partir para solucionar nuestras necesidades.

El modo de trabajo para usar esta librería es muy simple, se trata de crear una imagen con la etiqueta `` de HTML, en cuyo atributo `src` colocaremos la ruta hacia el script PHP que se encargará de generar la gráfica.

En el archivo PHP que generará la gráfica tendremos que incluir las librerías apropiadas para el tipo de gráfica que deseemos realizar, también habrá que instanciar el objeto JpGraph correspondiente, cargar los datos a visualizar y llamar a los métodos adecuados para mostrar la imagen. Un mecanismo bastante sencillo que veremos en un par de ejemplos a

continuación.

Ejemplo 1: una gráfica de línea.



En este ejemplo vamos a crear una gráfica lineal en la que mostraremos las horas de trabajo de una persona a lo largo de 10 días.

La generación de la gráfica de este ejemplo la hacemos en un archivo que hemos llamado `grafico_linea.php`, por lo tanto, la llamada a este archivo dentro de una imagen será la siguiente:

`` El código PHP del archivo `grafico_linea.php` es el siguiente:

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

// Some data
$ydata = array(11.5,3,8,12,5,1,9,13,5,7);

// Create the graph. These two calls are always required
$graph = new Graph(450,250,"auto");
$graph->SetScale("textlin");
$graph->img->SetAntiAliasing();
$graph->xgrid->Show();

// Create the linear plot
$lineplot=new LinePlot($ydata);
$lineplot->SetColor("black");
$lineplot->SetWeight(2);
$lineplot->SetLegend("Horas");

// Setup margin and titles
$graph->img->SetMargin(40,20,20,40);
$graph->title->Set("Ejemplo: Horas de Trabajo");
$graph->xaxis->title->Set("Días");
$graph->yaxis->title->Set("Horas de Trabajo");
$graph->ygrid->SetFill(true,'#EFEFEF@0.5','#F9BB64@0.5');
//$graph->SetShadow();

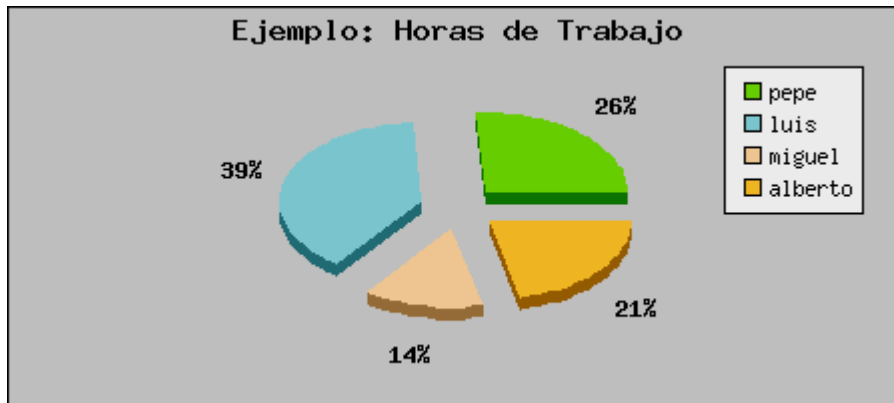
// Add the plot to the graph
```



```
$graph->Add($lineplot);
```

```
// Display the graph
$graph->Stroke();
?>
```

Ejemplo 2: una gráfica de tarta en 3D



Por otra parte, vamos a realizar un ejemplo de una gráfica de tarta, en la que aparecen las horas realizadas por cada uno de los empleados y el porcentaje respecto a las totales. En este caso, la tarta va a presentarse en un dibujo en 3 dimensiones.

El archivo donde se genera la gráfica se llama grafico_tarta.php. Lo llamaríamos dentro de una imagen con este código HTML.

```

```

El código PHP del archivo grafico_tarta.php será el siguiente:

```
<?php
include ("jpgraph/jpgraph.php");
include ("jpgraph/jpgraph_pie.php");
include ("jpgraph/jpgraph_pie3d.php");

$data = array(40,60,21,33);

$graph = new PieGraph(450,200,"auto");
$graph->img->SetAntiAliasing();
$graph->SetMarginColor('gray');
//$graph->SetShadow();

// Setup margin and titles
$graph->title->Set("Ejemplo: Horas de Trabajo");

$p1 = new PiePlot3D($data);
$p1->SetSize(0.35);
$p1->SetCenter(0.5);

// Setup slice labels and move them into the plot
$p1->value->SetFont(FF_FONT1,FS_BOLD);
$p1->value->SetColor("black");
$p1->SetLabelPos(0.2);

$nombres=array("pepe","luis","miguel","alberto");
$p1->SetLegends($nombres);
```

```
// Explode all slices
$p1->ExplodeAll();

$graph->Add($p1);
$graph->Stroke();
?>
```

Conclusión

JpGraph es una herramienta muy potente para la generación de gráficos en nuestra página web y gracias a su uso nos damos cuenta de sus muchas bondades:

- Es una librería gratuita (para uso no comercial), fácil de instalar y de fácil manejo.
- Incluye una completa documentación con multitud de ejemplos de los distintos gráficos que se pueden generar.
- Además de generar muchos tipos de gráficos, permite 'customizar' casi todo lo que se ve, resultando esto muy útil para integrar perfectamente el gráfico en el diseño de nuestra web.
- La forma de integrar el gráfico es muy sencilla: únicamente es necesario incluir una imagen (etiqueta) cuyo src sea el script PHP que generará nuestro gráfico (ver ejemplos).

Pocos defectos hemos encontrado en la librería, tan solo se podría mejorar lo siguiente:

- Los mensajes de error son algo escasos, y la mayoría de las veces cuando algo falla no obtenemos ninguna explicación.
- Se echa de menos algo de definición en los gráficos, sobre todo en los gráficos de sectores.

*Artículo por **Pablo González***

Valoraciones de los usuarios en PHP

El proceso es similar o mejor dicho el mismo empleado para sacar cualquier promedio, por ejemplo (El promedio de notas de un estudiante).

PTOS ACUMULADOS / TOTAL DE ACUMULACIONES

Para ello utilizaremos una tabla de datos:

```
<? CREATE TABLE `valoraciones` (
  `id_noticia` int(9) NOT NULL default '0',
  `num_votos` int(20) NOT NULL default '0',
  `ptos` int(20) NOT NULL default '0'
) TYPE=MyISAM;
?>
```

Cabe destacar que no es necesario utilizar una tabla de datos individual, siempre y cuando se cuente con una tabla general de artículos o contenidos y obviamente se dispongan al menos de los dos últimos campos de esta tabla, para el funcionamiento del script.

Este sistema trabaja con 1 solo registro por artículo o contenido, donde el campo id_noticia, se

refiere al identificador de nuestra noticia o contenido, num_votos; el total de votos realizados entorno a ese contenido y ptos la suma de todos los votos:

Por ejemplo:

id_noticia= 00001 - nuestra primera noticia
 num_votos= 3 - solo tres votos en esta noticia.
 Ptos= 15 - es probable que los tres hayan votado por 5 puntos.

Entonces lógicamente la valoración media es de 5 ptos.

Código Fuente:

```
<?
#####CONEXIÓN A MYSQL
@mysql_connect('localhost','usuario','pass')or die ('Ha fallado la conexión: '.mysql_error());
@mysql_select_db('base_de_datos')or die ('Error al seleccionar la BD: '.mysql_error());
////end conexión
$SSQL=mysql_query("SELECT * FROM valoraciones WHERE id_noticia='".$_GET[id]."'")or die(mysql_error());
$array_d=mysql_fetch_array($SSQL_);
$valoracion_total=@round($array_d[ptos]/$array_d[num_votos],2);

if(isset($_POST[valor])){
    if(mysql_num_rows($SSQL_)==0){
        @mysql_query("INSERT INTO valoraciones VALUES('".$_GET[id]."', '1', '".$_POST[valor]."'")or die ('ERROR AL
INSERTAR REGISTRO: '.mysql_error());
    }else{
        @mysql_query("UPDATE valoraciones SET num_votos=num_votos+1,ptos=ptos+'".$_POST[valor]."' WHERE
id_noticia='".$_GET[id]."'")or die ('ERROR AL MODIFICAR REGISTRO: '.mysql_error());
    }
    header('Location:'.$REQUEST_URI);
    exit;
}

echo '<strong>Valoración Media : '.$valoracion_total.</strong> <div style="background-color:#EFEFEF;
width:50px"><img width="'.($valoracion_total*5).'" height="6" style="background-color: #000099"></div>
<br>Total de votos: '.$array_d[num_votos];

?><hr>
<form action="<? echo $REQUEST_URI;?>" method="post">
Nueva valoración:
<select name="valor" id="valor">
<? for ($i=1; $i<=10 ; $i++) echo '<option value="'. $i.'">'. $i.'</option>';?>
</select>
<input type="submit" value="Votar por este artículo">
</form>
```

Explicando:

Primero que nada, como de costumbre creamos la conexión al servidor de mysql; donde debemos aportar nuestros datos de acceso reales.

Seleccionamos la base de datos sobre la cual trabajaremos.

Y generamos una petición al servidor Mysql mysql_query(); donde solicitamos los registros almacenados previamente por cada contenido o artículo.

Le asignamos a la variable \$array_d, la matriz devuelta por mysql_fetch_array(), para luego agrupar en la variable \$valoracion_total; el valor devuelto por la división entre \$array_d[ptos]

y `$array_d[num_votos]`, redondeamos el valor con 2 decimales `round()`, le colocamos el `@` al comienzo de la función para evitar que se muestren posibles errores por si acaso la división es entre 0 (cero).

Finalmente iniciamos una condición para comprobar si (`$_POST[valor]`) esta definida, es decir si se esta procesando el formulario para insertar nuevas valoraciones. Si la condición evalúa TRUE entonces ejecutamos una serie de instrucciones, siendo la primera una nueva condición:

```
if(mysql_num_rows($SSQL_)==0):
```

Para comprobar si aún no hay ningún registro o valoración e insertar uno nuevo.
`@mysql_query("INSERT INTO...`

De lo contrario (`else`) quiere decir en esta ocasión que ya ha votado al menos 1 persona y por ende no podemos hacer un nuevo registro, si no más bien una modificación del ya existente.

```
@mysql_query("UPDATE valoraciones...
```

Posteriormente sin importar cual de las dos rutinas halla seleccionado el script, realiza una redirección `header(Location: ...) a $_REQUEST_URI`, que es exactamente la misma URL , para que el usuario vea los datos actualizados. Es decir su voto agregado a la lista.

Finalmente imprimimos el resultado, con un grafico y agregamos el formulario que se utilizara para una próxima votación.

Ya quedaría de parte de ustedes modificarlo un poco (si desean) para que el usuario pueda votar solo 1 vez.

*Artículo por **Emmanuel García De Caro***

Rastrear a los robots de buscadores

En esta oportunidad les voy a comentar sobre los buscadores y como poder hacer un rastreo de su comportamiento o interactividad con nuestra web con las funciones PHP `$HTTP_USER_AGENT` y `$QUERY_STRING`

Lo primero que vamos hacer es hacernos una pregunta ¿a cuál buscador voy a rastrear? Una ves respondido esto en este caso Google.com.

Veamos el siguiente código php que debe usarse en todas las páginas de nuestra web que queremos rastrear al robot (lo ideal es crear una página con este código y agregar la página creada a las demás páginas con `include`, esto hace mas fácil los cambios a este script mas adelante):

```
<?php
if(ereg("google",$HTTP_USER_AGENT))
{
    if ($QUERY_STRING != "")
    {$url = "http://".$SERVER_NAME.$PHP_SELF.'?'.$QUERY_STRING;}
    else
    {$url = "http://".$SERVER_NAME.$PHP_SELF;}
}
```

```
$today = date("F j, Y, g:i a");  
mail("webmaster@jjcreaciones.com", "Se ha detectado un robot de Google en http://$SERVER_NAME",  
"$today - Google ha indexado la página $url.\n...: OTRA PÁGINA MAS :) :..");  
}  
?>
```

Explico el código

Este script reconoce al robot de google y la página en la cual realizo el escáneo, y envía un mail al webmaster de la página con los datos fecha y hora del suceso y nos da un breve mensaje de aliento (Un toque de estímulo no viene mal cierto), claro también se puede almacenar en una base de datos etc. Es cuestión de inventar ya les di la base inventen...

Ahora bien para los demás buscadores solo basta con cambiar: ("google", \$HTTP_USER_AGENT) por el nombre respectivo al robot, ejemplo:

- Google: ("google", \$HTTP_USER_AGENT)
- Yahoo: ("slurp", \$HTTP_USER_AGENT)
- Altavistar: ("scooter", \$HTTP_USER_AGENT)

Espero que les sirva.

*Artículo por **Jorge De Freitas***

Generar documentos de Word RTF con php

En muchos desarrollos web surge la necesidad de crear documentos que proceden de la combinación de un documento origen con los datos almacenados en la Intranet/Extranet de la empresa, algo parecido a lo que se consigue con la combinación de correspondencia de Word.

Ejemplos prácticos pueden ser los siguientes:

Generación de etiquetas a partir de una base de datos de clientes o artículos
Certificados o diplomas para los alumnos de un curso
Listados de participantes en un evento.

El control de la impresión desde un documento HTML es algo bastante complejo, por lo que generar un documento DOC, o mejor dicho RTF, nos permitirá resolver nuestro problema de una manera muy efectiva.

En este artículo desarrollaremos un ejemplo práctico, el de obtener un certificado de asistencia para cada uno de los alumnos que asisten a un curso.

La estrategia:

Nuestra estrategia se fundamenta en los siguientes pasos:

Partimos del documento en formato de Word que queremos obtener.

Convertimos el formato .DOC en .RTF. (Los archivos .RTF son de texto legible)

Introducimos en el documento .RTF unos códigos manuales que serán sustituidos por los datos procedentes de la base de datos.

Creamos una aplicación en PHP que lea el archivo .RTF y sustituya los códigos introducidos por los datos buscados.

Primer paso, Preparar la plantilla:

Para nuestro ejemplo partimos de un documento .DOC que certifica que un alumno ha realizado un curso. En la siguiente imagen vemos un certificado tipo.

CARMEN PINTO DE MENA
CANINO DEL PARRAL, 1 - 6º IZDA.
16002 - LOGROÑO
(LA RIOJA)

JOSE LUIS DEL RINCÓN, Responsable del Departamento de Prevención de Riesgos Laborales de la Fundación Laboral de la Construcción de La Rioja,

C E R T I F I C A

Que D./DÑA. CARMEN PINTO DE MENA, con D.N.I./N.I.E. número 17898754E, ha asistido con aprovechamiento al curso

PRESTO INICIACION 8.6. APLIC. INF. MED. Y PRES.

impartido por la Fundación Laboral de la Construcción de La Rioja, subvencionado por la Consejería de Hacienda y Empleo del Gobierno de La Rioja, y desarrollado entre los días 26-1-2004 y 24-2-2004, con una duración de 55.00 horas lectivas, con los contenidos relacionados al dorso,

Y para que conste y en cumplimiento a lo establecido en la Ley de Prevención de Riesgos Laborales, y agradeciendo la confianza depositada en esta Fundación Laboral de la Construcción, firmo y sello la presente en Logroño a 25 de Enero de 2005.

Será preciso crear un certificado para cada uno de los alumnos que ha realizado un curso, los datos que varía son los del alumno y los del curso realizado.

Una vez abierto el documento con Word, lo guardamos en formato RTF y sustituimos cada dato variable por unos códigos fácilmente reconocibles. Ojo, el formato RTF es muy puñetero, por lo que será preciso escribir los códigos sin confundirse para que no se introduzcan caracteres raros.

```
##*NOMBRE*#
##*DIRECCION*#
##*CP*# - ##*LOCALIDAD*#
(##*PROVINCIA*#)
```

JOSE LUIS DEL RINCÓN, Responsable del Departamento de Prevención de Riesgos Laborales de la Fundación Laboral de la Construcción de La Rioja,

C E R T I F I C A

Que D./DÑA. ##*NOMBRE*#, con D.N.I./N.I.E. número ##*DNI*#, ha asistido con aprovechamiento al curso

##*CURSO*#

impartido por la Fundación Laboral de la Construcción de La Rioja, subvencionado por la Consejería de Hacienda y Empleo del Gobierno de La Rioja, y desarrollado entre los días ##*FINICIO*# y ##*FFINAL*# con una duración de ##*HORAS*# horas lectivas, con los contenidos relacionados al dorso,

Y para que conste y en cumplimiento a lo establecido en la Ley de Prevención de Riesgos Laborales, y agradeciendo la confianza depositada en esta Fundación Laboral de la Construcción, firmo y sello la presente en Logroño a ##*AHORA*#.

En el caso del ejemplo utilizamos la # y el * de tal manera que por ejemplo, sustituimos el nombre del alumno por ##*NOMBRE*#

Guardaremos el documento con el nombre de **plantilla.rtf**

Segundo paso, El programa PHP:

Crearemos dos funciones auxiliares, la primera se llama leaf().

```
3 //Función que leer un archivo de texto y lo mete en una cadena
4 function leaf ($fichero) {
5     $texto = file($fichero);
6     $tamleaf = sizeof($texto);
7     for ($n=0;$n<$tamleaf;$n++){ $todo= $todo.$texto[$n];}
8     return $todo;
9 }
```

Esta función como se puede ver, recibe como argumento el nombre del fichero RTF, lo abre y almacena el contenido en la variable de texto \$todo. Esta función por tanto devuelve el contenido de un fichero.

La siguiente función, rtf(), es la que crea un RTF a partir de una plantilla y sustituyendo los códigos por los datos de la Intranet.

```
11 //Función que genera un rtf
12 function rtf($sql, $plantilla, $fsalida, $matequivalencias){
13 $pre=time();
14 $fsalida="/teleusers/certificados/".$pre.$fsalida;
15 mysql_connect("localhost","usuario","contraseña");
16 //Paso n°1.- Leo la plantilla rtf
17 $txtplantilla = leef($plantilla);
18 //Paso n°2.- Saca cabecera, el cuerpo y el final.
19 $matriz=explode("sectd",$txtplantilla);
20 $cabecera=$matriz[0]."sectd";
21 $inicio=strlen($cabecera);
22 $final=strrpos($txtplantilla,"}");
23 $largo=$final-$inicio;
24 $cuerpo=substr($txtplantilla,$inicio,$largo);
25 //Paso n°3.- Escribo el fichero
26 $punt = fopen($fsalida , "w");
27 fputs($punt,$cabecera);
28 $result = mysql("base_datos", $sql);
29 While($row=mysql_fetch_object($result)){
30     $despues=$cuerpo;
31     foreach ($matequivalencias as $dato) {
32         $datosql=$row->$dato[1];
33         $datosql = stripslashes ($datosql);
34         $datortf=$dato[0];
35         $despues=str_replace($datortf,$datosql,$despues);
36     }
37     fputs($punt,$despues);
38     $saltopag="\par \page \par";
39     fputs($punt,$saltopag);
40 }
41 fputs($punt,"}");
42 fclose ($punt);
43 return $fsalida;
44 }
```

Esta función recibe cuatro argumentos:

\$sql: la sentencia sql que devuelve los datos que se necesitan

\$plantilla: la plantilla RTF con los códigos

\$fsalida: el nombre del fichero resultado

\$matequivalencias: una matriz con las equivalencias entre los datos procedentes de la sentencia sql y los campos del certificado.

Vamos a comentar línea por línea el programa:

Las líneas 13 y 14 determinan el nombre y la ruta del fichero de salida, utilizamos la función time() para asegurarnos de que no se producirán dos ficheros con el mismo nombre.

En la línea 17 se almacena el contenido de la plantilla en la variable \$txtplantilla

En las líneas de la 19 a la 24 se extrae el cuerpo del documento RTF, el cuerpo del documento es lo que se repite, para extraerlo primero obtenemos la cabecera del documento, la cabecera queda determinada por la etiqueta rtf sectd. Un documento rtf termina siempre en }, con esa información extraemos el cuerpo.

En el paso 3, a partir de la línea 26 vamos escribiendo el archivo resultado.

En la línea 27 escribimos la cabecera del documento

En la línea 28 ejecutamos la consulta y en la 29 entramos en un bucle, en cada ciclo obtendremos los datos de un alumno y crearemos un certificado.

En la línea 30 creamos una copia del cuerpo del documento llamada \$despues, sobre esa copia realizaremos los cambios.

En la línea 31 entramos en un bucle que recorre la matriz de equivalencias, la línea 32 obtiene de la base de datos el dato, la línea 34 obtiene de la matriz el código a sustituir en la plantilla y la línea 35 sustituye el dato por el código.

Una vez sustituidos todos los campos la línea 37 escribe el certificado del alumno y las líneas 38 y 39 generan un salto de página para volver a escribir el siguiente certificado.

Una vez que hemos salido del While que va extrayendo los datos, la línea 41 cierra el documento rtf y la 42 la escritura del fichero.

Tercer paso, Llamada a la función RTF:

Una vez hemos creado estas dos funciones ya podemos llamar a la función rtf() y generar el documento.

```
48 $plantilla = "plantilla.rtf";
49 $sql = "SELECT SITUACION.CODSOLICITUD, SITUACION.CODALUMNO, CONCAT(
ALUMNOS.NOMALUMNO, ' ', ALUMNOS.AP1ALUMNO, ' ', ALUMNOS.AP2ALUMNO ) AS
NONCOMPLETO, ALUMNOS.DIRALUMNO, ALUMNOS.LOCALUMNO, ALUMNOS.CPSALUMNO,
ALUMNOS.PROALUMNO, CURSOS.NOMBRE, ALUMNOS.NIFALUMNO, CURSOS.NUMHORAS,
DATE_FORMAT(CURSOS.FINICIO,'%e-%c-%Y') AS FINICIO,
DATE_FORMAT(CURSOS.FFIN,'%e-%c-%Y') AS FFIN, DATE_FORMAT(CURRENT_DATE,
CONCAT('%e de ', ELT(MONTH(CURRENT_DATE),
'Enero','Febrero','Marzo','Abril','Mayo','Junio','Julio','Agosto','Septiembr
e','Octubre','Noviembre','Diciembre'), ' de %Y')) AS AHORA FROM SITUACION
LEFT JOIN ALUMNOS ON SITUACION.CODALUMNO = ALUMNOS.NIFALUMNO LEFT JOIN
CURSOS ON CURSOS.CODCURSO = SITUACION.CODCURSO WHERE SITUACION.CODCURSO
='%codcurso' AND SITUACION.ESTADO='APROBADO' ORDER BY SITUACION.ESTADO,
NONCOMPLETO";
50 $equivalencias[0][0]="*NOMBRE*";
51 $equivalencias[0][1]="NONCOMPLETO";
52 $equivalencias[1][0]="*DIRECCION*";
53 $equivalencias[1][1]="DIRALUMNO";
54 $equivalencias[2][0]="*CP*";
55 $equivalencias[2][1]="CPSALUMNO";
56 $equivalencias[3][0]="*LOCALIDAD*";
57 $equivalencias[3][1]="LOCALUMNO";
58 $equivalencias[4][0]="*PROVINCIA*";
59 $equivalencias[4][1]="PROALUMNO";
60 $equivalencias[5][0]="*CURSO*";
61 $equivalencias[5][1]="NOMBRE";
62 $equivalencias[6][0]="*DNI*";
63 $equivalencias[6][1]="NIFALUMNO";
64 $equivalencias[7][0]="*FINICIO*";
65 $equivalencias[7][1]="FINICIO";
66 $equivalencias[8][0]="*FFINAL*";
67 $equivalencias[8][1]="FFIN";
68 $equivalencias[9][0]="*HORAS*";
69 $equivalencias[9][1]="NUMHORAS";
70 $equivalencias[10][0]="*AHORA*";
71 $equivalencias[10][1]="AHORA";
72 $salida = rtf($sql, $plantilla, "certificado.rtf", $equivalencias);
73 $salida = "<A href='".$salida.'">Obtener</a>";
74 echo "<p>$salida</p>";
```

Comentamos las líneas:

En la línea 48 especificamos cual es la plantilla.

En la línea 49 establecemos la sentencia SQL que extrae los datos, en el caso del ejemplo es un poco compleja pues extrae los datos de varias tablas.

En las líneas desde la 50 a la 71 creamos la matriz de equivalencias, se trata de una matriz de dos dimensiones que relaciona el código que se encontrará en el archivo RTF con el campo de la sentencia SQL.

La línea 72 llama a la función `rtf()`

Y por fin las líneas 73 y 74 generan un enlace al archivo resultado.

Mejoras y comentarios.

Todo en esta vida es susceptible de ser mejorado, en este caso me queda pendiente el generar el archivo de salida directamente al navegador, con el objeto de no tenerlo que escribir en el disco duro.

Además tenemos desarrollados ejemplos de listados y de etiquetas que podemos documentar en posteriores artículos.

Un aspecto importante para seguir progresando en este tema es conocer la sintaxis de los documentos rtf, se trata de un trabajo un poco durillo que también podemos abordar en posteriores artículos.

*Artículo por **Alfredo Anaya***

Redondear decimales en PHP

Entre las funciones matemáticas de PHP se encuentra una que nos permite redondear un float (número en coma flotante o número con decimales) al valor entero más próximo. Se trata de la función `round()`. Para explicar su uso, lo mejor es verlo con un par de ejemplos:

```
round(0.6) devolvería el valor entero 1.  
round(7.3) devolvería el valor entero 7.
```

Pero en muchas ocasiones vamos a necesitar redondear un número en coma flotante de manera que conservemos algunos decimales. Por ejemplo, si estamos haciendo operaciones con euros o dólares, lo lógico es que conservemos dos decimales en el redondeo. Por extraño que parezca, PHP no incluye una función de redondeo que permita redondear conservando decimales y es por ello que vamos a comentar en este artículo una manera de conseguirlo.

Redondeo con dos decimales

Vamos a utilizar un pequeño truco para redondear con dos decimales, ayudándonos de la función `round()` de PHP. Se trata de hacer esta sencilla operación matemática.

Dado un valor, primero vamos a multiplicarlo por 100, lo vamos a redondear y finalmente, el

resultado lo vamos a dividir otra vez por 100. El caso, en un ejemplo, es este:

Para el valor 2.7931. Primero lo multiplicamos por 100. Tenemos el valor 279.31. Ahora hacemos el redondeo, puesto que los 0.31 queremos deshacernos de ellos. Haríamos un `round(279.31)` que devuelve como resultado 279. Por último, a este valor lo dividimos por 100, para recuperar los decimales, lo que nos da 2.79, que es el valor que deseábamos obtener.

La fórmula que sacamos sería la siguiente:

```
round(valor_float * 100) / 100
```

Si queremos, podemos utilizar una función como esta para el redondeo hasta obtener dos cifras decimales:

```
function redondear_dos_decimal($valor) {  
    $float_redondeado=round($valor * 100) / 100;  
    return $float_redondeado;  
}
```

Para un juego de valores como este:

```
redondear_dos_decimal(3.56666)  
redondear_dos_decimal(3.008)  
redondear_dos_decimal(3.66)  
redondear_dos_decimal(3.199)  
redondear_dos_decimal(3.9999)
```

Nos daría el siguiente resultado:

```
3.57  
3.01  
3.66  
3.2  
4
```

Redondear para un número de decimales cualquiera

Siguiendo las pautas marcadas en el ejemplo anterior, vamos a crear una función para redondear un float a cualquier número de decimales deseado.

Se trata de aplicar la misma lógica anterior. Para redondear a dos decimales hemos visto que había que multiplicar por 100, redondear y luego dividir otra vez por 100. Si queremos redondear con un solo decimal, tendremos que multiplicar por 10, redondear y luego dividir otra vez por 10.

Vamos a realizar un ejemplo de redondeo a tres decimales. Tenemos el valor 0.65195. Como queremos conservar 3 decimales, debemos multiplicar por 1000, con lo que obtenemos 651.95. Ahora hacemos el redondeo con la función `round()`, que nos deja la cifra sin decimales en 652 (se redondea al entero más próximo). Finalmente, debemos volver a dividir por 1000 el valor, con lo que tenemos 0.652.

La regla entonces es que tenemos que multiplicar y luego dividir, por la potencia de 10 con el número de decimales que deseamos conservar. Para obtener potencias en PHP tenemos la

función `pow()`, que recibe la base y exponente de la potencia buscada.

Por ejemplo, `pow(10,3)` nos devuelve 1000 y `pow(10,1)` nos devuelve 10.

Con esta función en PHP podremos obtener el redondeo a los decimales que se desee.

```
function redondeado ($numero, $decimales) {  
    $factor = pow(10, $decimales);  
    return (round($numero*$factor)/$factor); }
```

Si la llamamos con un juego de valores como estos:

```
redondeado (0.00211,3)  
redondeado (8.56,2)  
redondeado (7.26,1)  
redondeado (211,3)  
redondeado (2.009,2)
```

Obtendremos los siguientes resultados:

```
0.002  
8.56  
7.3  
211  
2.01
```

Esperamos que este par de funciones os resulten útiles. Hay que agradecer a Héctor Alonso la segunda de las funciones mostradas en este artículo.

Referencias: También recomendamos la lectura de la FAQ [Formatear decimales en PHP](#), donde se explica con detalle la función de PHP `number_format()`, que aparte de redondear decimales, permite formatear el número para que muestre el separador que deseemos en los millares y en los decimales. Si trabajamos con ASP, tenemos otro artículo para aprender a [redondear decimales en ASP](#).

*Artículo por **Pablo González***

Expresiones regulares

Las expresiones regulares son una serie de caracteres que forman un patrón, normalmente representativo de otro grupo de caracteres mayor, de tal forma que podemos comparar el patrón con otro conjunto de caracteres para ver las coincidencias.

Las expresiones regulares están disponibles en casi cualquier lenguaje de programación, pero aunque su sintaxis es relativamente uniforme, cada lenguaje usa su propio dialecto.

Si es la primera vez que te acercas al concepto de expresiones regulares (regex para abreviar) te animará saber que seguro que ya las has usado, aún sin saberlo, al menos en su vertiente más básica. Por ejemplo, cuando en una ventana DOS ejecutamos `dir *.*` para obtener un listado de todos los archivos de un directorio, estamos utilizando el concepto de expresiones regulares, donde el patrón `*` coincide con cualquier cadena de caracteres.

Unos ejemplos simplificados:

```
<? am // este es nuestro patrón. Si lo comparamos con:  
am // coincide  
panorama // coincide  
ambicion // coincide  
campamento // coincide  
mano // no coincide  

```

Se trata sencillamente de ir cotejando un patrón (pattern) -que en este ejemplo es la secuencia de letras 'am'- con una cadena (subject) y ver si dentro de ella existe la misma secuencia. Si existe, decimos que hemos encontrado una coincidencia (match, en inglés).

Otro ejemplo:

patrón: el

el ala aleve del leve abanico

Hasta ahora los ejemplos han sido sencillos, ya que los patrones usados eran literales, es decir que solo encontramos coincidencias cuando hay una ocurrencia exacta.

Si sabemos de antemano la cadena exacta a buscar, no es necesario quebrarse con un patrón complicado, podemos usar como patrón la exacta cadena que buscamos, y esa y no otra será la que de coincidencia. Así, si en una lista de nombres buscamos los datos del usuario pepe podemos usar pepe como patrón. Pero si además de pepe nos interesa encontrar ocurrencias de pepa y pepito los literales no son suficientes.

El poder de las expresiones regulares radica precisamente en la flexibilidad de los patrones, que pueden ser confrontados con cualquier palabra o cadena de texto que tenga una estructura conocida.

De hecho normalmente no es necesario usar funciones de expresiones regulares si vamos a usar patrones literales. Existen otras funciones (las funciones de cadena) que trabajan más eficaz y rápidamente con literales.

Caracteres y meta caracteres

Nuestro patrón puede estar formado por un conjunto de caracteres (un grupo de letras, números o signos) o por meta caracteres que representan otros caracteres, o permiten una búsqueda contextual.

Los meta-caracteres reciben este nombre porque no se representan a ellos mismos, sino que son interpretados de una manera especial.

He aquí la lista de meta caracteres más usados:

. * ? + [] () { } ^ \$ | \

Iremos viendo su utilización, agrupándolos según su finalidad.

Meta caracteres de posicionamiento, o anclas

Los signos ^ y \$ sirven para indicar donde debe estar situado nuestro patrón dentro de la cadena para considerar que existe una coincidencia.

Cuando usamos el signo ^ queremos decir que el patrón debe aparecer al principio de la cadena de caracteres comparada. Cuando usamos el signo \$ estamos indicando que el patrón debe aparecer al final del conjunto de caracteres. O mas exactamente, antes de un caracter de nueva linea Asi:

```
<?
```

```
^am // nuestro patrón  
am // coincide  
cama // no coincide  
ambidiestro // coincide  
Pam // no coincide  
caramba // no coincide
```

```
am$  
am // coincide  
salam // coincide  
ambar // no coincide  
Pam // coincide
```

```
^am$  
am // coincide  
salam // no coincide  
ambar // no coincide
```

```
?>
```

o como en el ejemplo anterior:

patrón: ^el

el ala aleve del leve abanico

Las expresiones regulares que usan anclas solo devolveran una ocurrencia, ya que por ejemplo, solo puede existir una secuencia el al comienzo de la cadena.

patrón: el\$

el ala aleve del leve abanico

Y aqui no encontramos ninguna, ya que en la cadena a comparar (la linea en este caso) el patrón "el" no está situado al final.

Para mostrar una coincidencia en este ejemplo, tendríamos que buscar "co":

patrón: co\$

con el ala aleve del leve abanico

Hemos comenzado por unos metacaracteres especiales, ya que ^ \$ no representan otros caracteres, sino posiciones en una cadena. Por eso, se conocen tambien como anchors o anclas.

Escapando caracteres

Puede suceder que necesitemos incluir en nuestro patrón algun metacaracter como signo literal, es decir, por si mismo y no por lo que representa. Para indicar esta finalidad usaremos un carácter de escape, la barra invertida \.

Así, un patrón definido como 12\\$ no coincide con una cadena terminada en 12, y sí con 12\$:

patrón: 100\$

el ala aleve del leve abanico cuesta 100\$

patrón: 100\\$

el ala aleve del leve abanico cuesta 100\$

Fijate en los ejemplos anteriores. En el primero, no hay coincidencia, porque se interpreta "busca una secuencia consistente en el número 100 al final de la cadena", y la cadena no termina en 100, sino en 100\$.

Para especificar que buscamos la cadena 100\$, debemos escapar el signo \$

Como regla general, la barra invertida \ convierte en normales caracteres especiales, y hace especiales caracteres normales.

El punto . como metacaracter

Si un metacaracter es un caracter que puede representar a otros, entonces el punto es el metacaracter por excelencia. Un punto en el patrón representa cualquier caracter excepto nueva línea.

Y como acabamos de ver, si lo que queremos buscar en la cadena es precisamente un punto, deberemos escaparlos: \.

patrón: '\.'

el ala aleve del leve abanico

Observa en el ejemplo anterior como el patrón es cualquier caracter (incluido el de espacio en blanco) seguido de una l.

Metacaracteres cuantificadores

Los metacaracteres que hemos visto ahora nos informan si nuestro patrón coincide con la cadena a comparar. Pero ¿y si queremos comparar con nuestra cadena un patrón que puede estar una o mas veces, o puede no estar? Para esto usamos un tipo especial de meta caracteres: los multiplicadores.

Estos metacaracteres que se aplican al caracter o grupo de caracteres que les preceden indican en que número deben encontrarse presentes en la cadena para que haya una ocurrencia.

Por ello se llaman cuantificadores o multiplicadores. Los mas usados son * ? +

<?

```
* // coincide si el caracter (o grupo de caracteres) que le
// precede esta presente 0 o mas veces
// ab* coincide con "a", "ab", "abbb", etc.
//ejemplo:
cant*a // coincide con canta, cana, cantttta
```

```
? // coincide si el carácter (o grupo de caracteres) que precede
// esta presente 0 o 1 vez
// ab? coincide con "a", "ab", no coincide con "abb"
// ejemplo:
cant?a // coincide con canta y cana
d?el // coincide con del y el
(ala)?cena // coincide con cena y alacena
```

```
+ // coincide si el carácter (o grupo) que le precede está
// presente al menos 1 o mas veces.
// ab+ coincide con "ab", "abbb", etc. No coincide con "a"
//ejemplo:
cant+a // coincide con canta, canttttta, NO coincide con
// cana ?>
```

patrón: 'a*le'

el ala aleve del leve abanico

patrón: ' ?le'
el ala aleve del leve abanico

patrón: ' +le'
el ala aleve del leve abanico

Ademas de estos cuantificadores sencillos tambien podemos especificar el numero de veces máximo y mínimo que debe darse para que haya una ocurrencia:

patrón: (.a){2}
el ala aleve del leve abanico

```
<?
{x,y} // coincide si la letra (o grupo) que le precede esta presente
// un mínimo "x" veces y como máximo "y" veces
// "ab{2}" coincide con "abb": exactamente dos ocurrencias de "b"
// "ab{2,}" coincide con "abb", "abbbb" ... Como mínimo dos
// ocurrencias de b, máximo indefinido
// "ab{3,5}" coincide con "abb", "abbbb", o "abbbbb": Como mínimo
// dos ocurrencias, como máximo 5
```

```
a{2,3} // coincide con casaa, casaaa
```

```
a{2, } // coincide con cualquier palabra que tenga al
// menos dos "a" o mas: casaa o casaaaaa, no con casa
```

```
a{0,3} // coincide con cualquier palabra que tenga 3 o
// menos letras "a".
// NOTA: puedes dejar sin especificar el valor máximo. NO
// puedes dejar el valor inicial vacío
```

```
a{5} // exactamente 5 letras "a"
?>
```

Por tanto, los cuantificadores * + ? pueden también ser expresados así:

* equivale a {0,} (0 o mas veces)

+ equivale a {1,} (1 o mas veces)

? equivale a {0,1} (0 o 1 vez)

Este artículo continúa en el [manual de expresiones regulares](#).

Artículo por irv.

Envío de boletines a emails de una base de datos MySQL

Ahora está muy de moda el envío de boletines para informar de las novedades en las webs. El único problema es que para hacerlo, la mayoría de webmasters tienen que depender de alguna empresa (o servidor) que les gestione la base de datos y les envíe los emails cuando lo requieran.

En este artículo explicaré paso a paso como crear la tabla (en la base de datos MySQL) necesaria para el almacenamiento de los emails y posteriormente, cómo enviar un boletín a todos ellos, con el aliciente de que no aparezcan todos los destinatarios en el campo "Para:"




































del mensaje.

Creación de tabla en la base de datos

Mediante su gestor de la base de datos MySQL (seguramente "phpMyAdmin") debe crear una nueva tabla en su base de datos. Aquí le facilitamos el archivo de texto para ello:

```
CREATE TABLE usuarios (
id bigint(7) NOT NULL auto_increment,
nick varchar(100) NOT NULL,
passowrd varchar(100) NOT NULL,
nombre varchar(255) default NULL,
email varchar(100) default NULL,
KEY id (id)
)
```

Y quedará así en su phpMyAdmin:

| Campo | Tipo | Atributos | Nulo | Predeterminado | Extra | Acción |
|--------------------------|----------|--------------|------|----------------|----------------|---|
| <input type="checkbox"/> | id | bigint(7) | No | | auto_increment |        |
| <input type="checkbox"/> | nick | varchar(100) | No | | |        |
| <input type="checkbox"/> | passowrd | varchar(100) | No | | |        |
| <input type="checkbox"/> | nombre | varchar(255) | Si | NULL | |        |
| <input type="checkbox"/> | email | varchar(100) | Si | NULL | |        |

Tras esto usted ya tiene la tabla necesaria para almacenar los datos dentro de una base de datos MySQL. Podría explicar como introducir los datos en esta tabla mediante formularios de registro... pero eso ya lo dejo a gusto del webmaster.

Envío del boletín

Para enviar el boletín solo debe cambiar los datos del siguiente archivo php por los de su base de datos y tendrá listo su archivo. Para enviar el boletín solo tiene que ejecutar este archivo en su servidor (cargarlo: por ejemplo, entrar en <http://usuarios.lycos.es/tunombredeusuario/prueba.php>):

```
<?php
$mi_conexion=mysql_connect('servidor','nombre_de_usuario','contraseña') or die('No se pudo conectar con la base de datos!');

/*
Servidor: aquí debe poner la dirección de su servidor, en la mayoría de las compañías es 'localhost'
Nombre_de_usuario: debe poner su nombre de usuario en el servidor.
Contraseña: su contraseña.
*/

mysql_select_db('base_de_datos',$mi_conexion);

/*
Base_de_datos: nombre de su base de datos.
*/

$query_Tabla = "SELECT email FROM usuarios ORDER BY id ASC";
$Tabla = mysql_query($query_Tabla, $mi_conexion) or die(mysql_error());

/*
Email: es el nombre del campo que requerimos, si ha seguido nuestras instrucciones creando nuestra misma tabla en la base de datos no debe cambiar esto.
Usuarios: es el nombre de la tabla donde está el campo 'email', si ha seguido nuestras instrucciones creando nuestra
```

misma tabla en base de datos no debe cambiar esto.

```
*/

//elaboramos cadena de emails
$losemails="";
while ($row_Tabla=mysql_fetch_assoc($Tabla)) {
    $losemails.=( $row_Tabla['email'].", ");
}

$largo=strlen($losemails);
if ($largo>2)
{
    //quitamos ultimos ", "
    $losemails=substr($losemails,0,$largo-2);
}
else
{
    echo "No hay destinatarios!";
    die();
};

// se definen los argumentos de mail( ):
$asunto='Prueba de email desde PHP';
$mensaje='<html>
<head>
    <title>Titulo de la Pagina</title>
</head>
<body>
    <p>Esta es una prueba de emails en formato HTML</p>
    Precios de nuestros productos:
    <table>
        <tr>
            <th>PRODUCTO<th>PRECIO<th>PROMOCION</th>
        </tr>
        <tr>
            <td>Refresco Grande</td><td>25,50</td><td>23,00</td>
        </tr>
        <tr>
            <td>Helado Napolitano</td><td>27,50</td><td>25,00</td>
        </tr>
        <tr>
            <td>Patatas</td><td>18,50</td><td>15,00</td>
        </tr>
    </table>
</body>
</html>';

/*
Aquí debe poner su email en formato HTML
*/

$envia='enviante';
$remite='email_remitente';

/*
Enviante: Nombre del enviante
Email_remitente: email que desea mostrar como remitente.
*/

/// Envío del email:

mail(null, $asunto, $mensaje, "MIME-Version: 1.0
Content-type: text/html; charset=iso-8859-1
From: $envia <$remite>
Bcc: $losemails" . "\r\n") or die("Error al Enviar el Email");
```

```
echo "Mensaje Enviado con Éxito!"; //
```

```
mysql_free_result($Tabla);
mysql_close($mi_conexion);
```

?> Ya tenemos nuestro sistema de envío de boletines, no tendremos que depender de nadie.

El único problema que presenta este sistema es que al no aparecer nada en el campo "Para:" del email, algunos gestores de email lo colocan como "correo no deseado". Acepto contribuciones de otros lectores para mejorarlo.

*Artículo por **Raúl Avilés***

Agregar caritas al sistema

Agreguemos expresiones o caritas a nuestro sistema, (foro, tagboard, chat, comentarios ósea cualquier script que queramos que tenga una(s) carita(s) o expresión(es).

Empecemos:

Vamos a usar la función `str_replace()` de PHP: esta función sustituye tal por tal2

ejemplo:`str_replace("carita_loca", "")` ósea cambiamos cualquier aparición del texto "carita_loca" por una imagen de la carita loca. Solo es un ejemplo.

Veamos como funciona:

Digamos que tenemos un sistema sea mysql o puro .txt, es indiferente lo importante es cambiar las expresiones de texto por imágenes, cierto. Bien esta es la función:

```
<?php
function expresiones($mensaje) {
    $mensaje = str_replace(":", "<img src= feliz.gif>", $mensaje);
    $mensaje = str_replace(":", "<img src=triste.gif>", $mensaje);
    $mensaje = str_replace(":", "<img src=lengua.gif>", $mensaje);
    return $mensaje;
}
?>
```

Así sola esta función no hace nada:

Para que muestre las caritas filtramos el resultado de la variable `$mensaje` así : `$mensaje = expresiones($mensaje);` Esto le dice al sistema que si dentro de la variable `$mensaje` encuentra (ejemplo: :), :P, :(,) lo cambie por la imagen correspondiente.

Hasta el momento cuando el usuario coloque en nuestro campo `name="mensaje": * hola :) *` cambiara el :) por la carita correspondiente

Pero vamos mas lejos:

Recordamos que "`$mensaje`" es el nombre del campo donde pondremos el texto que el sistema cambiara por las caritas y el texto grande o idea o lo que sea que estemos escribiendo. Entonces tenemos el ejemplo de un formulario:

```
<form name="mi_formulario" action="procesar.php" method="post">
Nombre: <input type="text" name="nombre" size="26">
Mensaje <textarea name="mensaje" cols="58" rows="14" wrap="virtual"><?echo $mensaje;?></textarea>
<input type="submit" name="Submit" value="Publicar">
</form>
```

Ahora nosotros queremos que al dar un clic en un botón o carita salga :) en el campo name="mensaje" tendremos que hacer esto:

Antes del formulario ponemos esto:

```
<script language="JavaScript">
<!--
function caritas(Which){
document.mi_formulario.mensaje.value = document.mi_formulario.mensaje.value + Which;
}
// -->
</script>
```

luego el formulario, ojo con esto mi_formulario es el nombre del formulario y mensaje es el campo donde va a ir las expresiones en texto de las caritas.

Este es el links que va hacer aparecer las caritas en el textarea:

```
<a href="javascript:caritas('(')"><img src=triste.gif></a>
```

Ahora les toca a ustedes inventar.

*Artículo por **Jorge De Freitas***

Htaccess y páginas dinámicas

Una de las dificultades que se presenta cuando utilizamos URL dinámicas, es decir que envías valores a través del método GET; Radica en el momento cuando los robots que navegan por toda la red recopilando información de diferentes índole, ingresan a nuestro sitio Web y se consiguen con parámetros que ellos no admiten en el mayor de los casos.

Por ejemplo tenemos esta url: <http://localhost/index.php?cat=4&subcat=45&id=4566>
En este caso la mayoría de los indexadores no admitirían más de 1 solo parámetro y harían caso omiso a los caracteres: (& ? = #) siguientes. Entonces ya dependería de nuestro PageRank si esa url es indexada a los principales motores de búsqueda.

Sin embargo existe una forma que "DIFRAZA" la url, por decir alguna palabra, sin modificar la funcionalidad de nuestra página; editando el archivo .htaccess que debe ubicarse al comienzo de cada directorio.

Debemos agregar las siguientes líneas:

```
<? RewriteEngine on
RewriteRule ^.*$ index.php
?>
```

Donde index.php, es nuestro archivo principal del directorio a mostrar.

Ahora volviendo al ejemplo anterior, ya podríamos llamar la url de la siguiente forma:
<http://localhost/4/45/4566>

Ya no tendremos ningún problema con los motores de búsqueda. Ahora nos queda una tarea sumamente sencilla.

¿Como extraer esos valores que se almacenan en esa url?

Ciertamente `$REQUEST_URI`, nos devuelve todo el fragmento o path físico de la url actual. Lo siguiente seria utilizar una función que nos permita dividir esa url; utilizando la / (Barra) como punto de referencia, pudiese ser con `explode()` que nos devuelve una matriz con las cada una de las partes.

Código Fuente:

```
<? PHP
$aux = substr( $_SERVER['REQUEST_URI'], strlen('/') );
if( substr( $aux, -1) == '/') {
    $aux=substr( $aux, 0, -1);
}
$urlARRAY___ =explode( '/', $aux);

//////////VALORES A UTILIZAR
$CATEGORIA=$urlARRAY___[0];
$SUBCATEGORIA=$urlARRAY___[1];
$ID=$urlARRAY___[2];
?>
```

Hemos publicado un [Manual de .htaccess para la creación de URL amigable](#) donde podrás ampliar la información de configuración de htaccess.

Artículo por [Emmanuel García De Caro](#)

Programar un buscador con PHP y MySQL.

Supongo al lector familiarizado con la programación en php y la administración básica de Mysql.

Como punto de partida, la información a buscar la tenemos que tener almacenada en una tabla de nuestra base de datos. Pongamos como ejemplo una tabla con artículos técnicos que llamaremos ARTICULOS. Esta tabla ARTICULOS tendrá como mínimo dos campos llamados TITULO y DESARROLLO, que almacenarán el título del artículo y su contenido respectivamente.

	Campo	Tipo
<input type="checkbox"/>	REFERENCIA	int(11)
<input type="checkbox"/>	TITULO	varchar(100)
<input type="checkbox"/>	DESARROLLO	text
<input type="checkbox"/>	AUTOR	varchar(50)

Los tipos de los campos a buscar deberán ser CHAR, VARCHAR o TEXT con sus variantes.

Ahora diseñaremos una página con un formulario y una caja de texto para escribir la cadena de consulta "busqueda" y llamar al script de búsqueda "buscar.php". Ejemplo:

```
<FORM METHOD=POST ACTION="buscar.php">
Buscar: <INPUT TYPE="text" NAME="busqueda">
</FORM>
```

En el fichero buscar.php es donde está el click de la cuestión. Hay que hacer la consulta que busque las palabras y devuelva como resultado los registros coincidentes. Hasta aquí parece fácil y podríamos resolverlo así:

```
SELECT * FROM ARTICULOS WHERE DESARROLLO LIKE '%$busqueda%' OR TITULO LIKE '%$busqueda%'
```

Pues esta consulta nos mostrará todos los artículos que en su título o en su contenido aparezca la frase de búsqueda tal y como nosotros la introducimos. Esto es muy limitado ya que un cambio en el orden de las palabras o un artículo de separación dará al traste con nuestra búsqueda no mostrando los resultados deseados.

Si bien podríamos depurar y mejorar la sintaxis de nuestra búsqueda utilizando el operador LIKE las búsquedas resultarían muy lentas y no tendrían el resultado esperado.

La solución más eficiente es utilizar los índices FULLTEXT específicamente indicados para estos menesteres. Pues bien esto implica ir a nuestra base de datos, y crear un índice FULLTEXT con todos los campos que deseamos incluir en nuestra búsqueda, que en nuestro caso son TITULO y DESARROLLO. Para ello basta con el phpmyadmin o bien escribir la instrucción directamente:

```
ALTER TABLE ARTICULOS ADD FULLTEXT(TITULO, DESARROLLO);
```

Índices : [\[Documentación\]](#)

Nombre de la clave	Tipo	Cardinalidad	Acción	Campo
PRIMARY	PRIMARY	0	 	REFERENCIA
PUBDATE	INDEX	Ninguna	 	PUBDATE
CATEGORIA	INDEX	Ninguna	 	CATEGORIA
BUSQUEDA	FULLTEXT	Ninguna	 	TITULO DESARROLLO 1

Una vez creado el índice la instrucción SQL para buscar será:

```
SELECT * FROM ARTICULOS WHERE MATCH(TITULO, DESARROLLO) AGAINST ('$busqueda')
```

Esta línea utiliza la función MATCH ... AGAINST ... que encuentra el texto buscado, usando consultas en lenguaje natural parecido a como lo hacen los motores de búsqueda. Además, se calcula internamente una puntuación en función de como aparecen los términos buscados dentro de nuestro artículo.

Perfeccionando la búsqueda:

```
SELECT * , MATCH (TITULO,DESARROLLO) AGAINST ('$busqueda') AS puntuacion FROM ARTICULOS WHERE MATCH (TITULO, DESARROLLO) AGAINST ('$busqueda') ORDER BY puntuacion DESC LIMIT 50
```

Esta línea devuelve los 50 primeros resultados encontrados ordenados de mas a menos

puntuación. El valor de la puntuación es un número decimal comprendido entre 0 y 1 por cada ocurrencia del patrón de búsqueda, que se irá sumando si ese patrón es encontrado en varias ocasiones. De cualquier forma esta puntuación es un algoritmo interno de la base de datos.

Algunos problemas. Las búsquedas realizadas con MATCH ...AGAINST en ocasiones fallan cuando el término a buscar contiene una sola palabra. Por el contrario son unas búsquedas rapidísimas que producen mejores resultados que otros métodos más rudimentarios como el LIKE cuando se trata de varias palabras o frases.

La solución que le he dado a ese problema de las búsquedas con MATCH...AGAINST ha sido chequear el número de palabras a buscar, utilizando una búsqueda simple con LIKE en el caso de una sola palabra, y el método MATCH...AGAINST en el caso de varias. Ejemplo del fichero buscar.php

```
<?php
//cadena de conexión
mysql_connect("host","usuario","password");
//DEBO PREPARAR LOS TEXTOS QUE VOY A BUSCAR si la cadena existe
if ($busqueda<>""){
    //CUENTA EL NUMERO DE PALABRAS
    $trozos=explode(" ",$busqueda);
    $numero=count($trozos);
    if ($numero==1) {
        //SI SOLO HAY UNA PALABRA DE BUSQUEDA SE ESTABLECE UNA INSTRUCCION CON LIKE
        $cadbusca="SELECT REFERENCIA, TITULO FROM ARTICULOS WHERE VISIBLE =1 AND DESARROLLO LIKE '%$busqueda%' OR TITULO LIKE '%$busqueda%' LIMIT 50";
    } elseif ($numero>1) {
        //SI HAY UNA FRASE SE UTILIZA EL ALGORITMO DE BUSQUEDA AVANZADO DE MATCH AGAINST
        //busqueda de frases con mas de una palabra y un algoritmo especializado
        $cadbusca="SELECT REFERENCIA, TITULO , MATCH ( TITULO, DESARROLLO ) AGAINST ( '$busqueda' ) AS Score
FROM ARTICULOS WHERE MATCH ( TITULO, DESARROLLO ) AGAINST ( '$busqueda' ) ORDER BY Score DESC LIMIT
50";
    }
}
$result=mysql("teleformacion", $cadbusca);
While($row=mysql_fetch_object($result))
{
    //Mostramos los titulos de los articulos o lo que deseemos...
    $referencia=$row->REFERENCIA;
    $titulo=$row->TITULO;
    echo $referencia." - ".$titulo."<br>";
}
?>
```

Enlaces para completar esta información:

http://www.fabio.com.ar/verpost.php?id_noticia=959

http://dev.mysql.com/doc/mysql/en/Fulltext_Search.html

*Artículo por **Alfredo Anaya***

Códigos y valores de colores RGB

Hola amigos, haciendo una de mis aplicaciones me surgió una necesidad ¿Como obtener el código de un color partiendo de los valores (ROJO, VERDE, AZUL), y también lo contrario, ¿ Como obtener de un código RGB los valores (ROJO, VERDE, AZUL)?.

Por ejemplo:

Caso 1

Si tengo: Rojo: 20, Verde: 74 y Azul: 216; como hacer para imprimir #144AD8, que es su código Hexadecimal.

Caso 2

Si tengo #D8B014, como hacer para imprimir: Rojo: 216, Verde: 176, Azul: 20

Haciendo varias pruebas con la paleta de colores, desarrolle una función que soluciona este problema.

¿Cual es la utilidad?

Por ejemplo si queremos obtener los matices o tonos de un color específico, utilizando la función y unos bucles (con condiciones obviamente) esta todo solucionado.

Así que a partir de unos 3 colores, podemos crear todo un estilo (CSS) para nuestra página web, utilizando todos los tonos instruidos por nosotros.

También si generamos graficas dinámicamente por php.

Teoría

Recordemos que el código de un color RGB se constituye de 6 CARACTERES (Numéricos de 0 a 9 ó letras de (A) a (F), y el esquema es el siguiente: (RR-VV-AA)

RR= Rojo
VV= Verde
AA= Azul

De donde precisamente viene el termino (RGB) en Ingles claro esta.

¿Que quiere decir esto?

Los códigos de colores son Hexadecimales; es decir con base 16 entonces:

0= 1
1= 1
2= 2
3= 3
4= 4
5= 5
6= 6
7= 7
8= 8
9= 9
A= 10
B= 11
C= 12

D= 13
 E= 14
 F= 15

LA FORMULA SERIA LA SIGUIENTE:

Sí rojo vale FF (máximo valor)

Entonces la primera "F" Valeria 15×16 donde 16 es constante y 15 el valor de su posición con respecto a 0 (El máximo)

El resultado seria: 240, luego le sumamos la posición con respecto a 0 del segundo carácter (NUEVAMENTE F, por tanto el valor seria 15)

$240 + 15 = 255$ (color máximo)

El proceso se repetiría en cada uno de los pares (Rojo – Verde – Azul)

De acuerdo con esto, entonces los valores del color: #4AF121 serían:

Rojo: (4A) $4 \times 16 + 4 = 68$. El valor en la paleta es equivalente a 68

Verde: (F1) $15 \times 16 + 1 = 241$. El valor en la paleta es equivalente a 241

Azul: (21) $2 \times 16 + 1 = 33$. El valor en la paleta es equivalente a 33

Basándonos en la teoría anterior, podemos desarrollar la función:

Código Fuente:

```
<?
function color_formt($COLOR,$tipo){
for ($i=0; $i<=9; $i++){
    $codigo_part[$i]=$i;
    $fac_mult[$i]=$i; }
for ($is='a'; $is<='f'; $is++){
    $codigo_part[$is]=$is;
    $fac_mult[$is]=$i;
    $i++;
}

if($tipo==1){
    $COLOR=strtolower($COLOR);
    /* de código a valores FORMATO VALIDO : #RRVVAA */
    $COLOR= ereg_replace('#','',$COLOR);
    $col[rojo][1]= substr($COLOR,0,1);
    $col[rojo][2]= substr($COLOR,1,1);
    $col[verde][1]= substr($COLOR,2,1);
    $col[verde][2]= substr($COLOR,3,1);
    $col[azul][1]= substr($COLOR,4,1);
    $col[azul][2]= substr($COLOR,5,1);
    $color_[rojo]=($fac_mult[$col[rojo][1]]*16)+$fac_mult[$col[rojo][2]];
    $color_[verde]=($fac_mult[$col[verde][1]]*16)+$fac_mult[$col[verde][2]];
    $color_[azul]=($fac_mult[$col[azul][1]]*16)+$fac_mult[$col[azul][2]];
}
elseif($tipo==2){ /* de valores a código FORMATO VALIDO : RRR|VVV|AAA */
    $rojo=$COLOR[0];
    $verde=$COLOR[1];
    $azul=$COLOR[2];
    $color_=strtoupper('#'.$codigo_part[floor($rojo/16)].$codigo_part[$rojo%16]. /*rojo */
    $codigo_part[floor($verde/16)].$codigo_part[$verde%16]. /*verde*/
    $codigo_part[floor($azul/16)].$codigo_part[$azul%16]); /*azul*/
}
}
```

```

else{
    $color_='N/a';
}
return $color_;
}

/*APLICAMOS LA FUNCIÓN de las dos formas*/

echo '<strong>de código a valores ( Rojo, Verde, Azul)</strong><HR>';
/* uso color_fmt('#RRVVAA',1)*/
foreach(color_fmt('#4AF121',1) as $color=>$valor) {
    $VALOR[$color]=$valor;
    echo $color.'->'. $valor.<BR>';
}
echo '<strong>de valores a código #RRVVAA </strong><HR>';

/* uso color_fmt(array(ROJO,AZUL,VERDE),2)*/
echo color_fmt(array($VALOR[rojo],$VALOR[verde],$VALOR[azul]),2);

// OBTENER LOS TONOS DE UN COLOR
echo '<HR><strong> Obtener los tonos de un color </strong>';

for ($i=1; $i<=255 ; $i++){
    echo '<div style="background-color:'.color_fmt(array($i,$i,$i),2).'; width:50px"><img width="1"
height="1"></div>';
}
?>
    
```

Explicando:

Definimos una función bajo el nombre de `color_fmt(ARG 1, ARG 2)`

Donde ARG 1: es el código del color o los datos del color, tomando como punto de referencia ARG 2, cuyo valor puede ser 1 ó 2.

Sí ARG 2 es 1, entonces transformaríamos de código a valores) y ARG 1 debe ser un código de color válido

Sí ARG 2 es 2, entonces transformaríamos de valores a códigos) y ARG 1 debe estar conformado por los 3 parámetros que constituyen el color, asociados a una matriz.

Posteriormente realizaremos varias iteraciones con el bñcle FOR, a fin de recolectar los números de 0 a 9 y las letras de (A) á (F) y asociarlos a una matriz escalar, para su posterior utilización en las condiciones subsiguientes.

SI `$tipo==1`

Primero que nada pasamos a minúscula el color, para no tener problemas a posterior con la función de cadenas `strtolower()`;

Reemplazamos el valor (#) del código de color por nada (Si lo contiene) mediante la función `ereg_replace()`, de expresiones regulares.

Luego extraeremos carácter por separado de `$codigo`, utilizando la función de cadenas `substr()`;

Aplicamos la formula para obtener cada valor y los almacenamos en un array asociado `$color_` ;

SI \$tipo==2

Realizamos el proceso contrario a lo anterior, antes multiplicábamos ahora dividiremos y redondeamos el valor hacia abajo con la función matemática floor(), esto para el primer caracteres del conjunto, con el segundo obtenemos el residuo de la división entre el nombre del color y 16 , repetimos el mismo proceso los 3 grupos. Pasamos todo el resultado a mayúscula con strtoupper()

De lo contrario a si \$tipo==1 ó \$tipo==2

Asignamos de valor a la variable \$color_='N/a';

Finalmente sin importar la rutina retornamos la variable \$color_.

Y aplicamos la función.

Nota: Como el objetivo en blasten.com siempre es y será propiciar la creación de funciones donde muchas de las cuales ya existen desde hace años en php, con la finalidad de poner en práctica una serie de conocimientos de gran envergadura y desarrollar destrezas como la lógica de programación tan importante para la construcción de cualquier algoritmo.

¿Porque digo lo anterior?

Porque pudiésemos resumir el código anterior a mínimo 3 líneas con un par de funciones ya existentes en php.

Una de ellas: La función de cadenas sscanf(), que nos permite analizar y Dividir ó fraccionar (Trocear) una cadena antes de arrojar su salida al navegador a partir de un formato específico.

Es decir pudiésemos obtener el valor hexadecimal de un numero sin necesidad de aplicar esa formula y realizar ese par de bucles.

Por ejemplo:

```
<?
scanf('33FFCC', "%2x%2x%2x", $rojo, $verde, $azul);
echo 'Rojo: '.$rojo;
echo '<br> Verde: '.$verde;
echo '<br> Azul: '.$azul;
?>
```

Produciría la salida:

```
Rojo: 51
Verde: 255
Azul: 204
```

¿Que tal ? y si ha eso le agregamos:

```
<?
$hexadecimales='0123456789ABCDEF';
echo $hexadecimales[15];
?>
```

Algo sumamente básico en php, Entonces entenderíamos que "ilógico fue realizar los bucles y tantas otras cosas" que aumenta a " nivel micro" en ese caso el procesamiento del script ósea

en generar un resultado o salida al navegador.

Artículo por **Emmanuel García De Caro**

Test de velocidad de conexión a Internet

Estudiando rápidamente los sistemas que obtienen un reflejo aproximado de la conexión a Internet, medida en Kbps; se me ocurrió realizar algo similar en Php.

¿Cual es la utilidad?

Se puede determinar en cuanto tiempo aproximado se descargara un archivo de Internet. Por ejemplo, si estamos elaborando una página de descargas y queremos mostrarle al usuario en cuanto tiempo APROX, estará ese archivo en su PC.

En realidad algo extremadamente sencillo ya veremos más adelante porque.

Teoría Básica

Es importante resaltar que un carácter CUALQUIERA siempre tendrá un tamaño de 1 Byte, es decir 8 impulsos electrónicos (8 Bit), por lo cual una cadena de texto que contenga mil veinticuatro caracteres en total, tendrá una dimensión de 1024 Bytes lo que seria igual a 1 KB.

1 KB es igual 1024 Bytes

1 MB es igual a 1024 KB

1 GB es igual a 1024 MB

1 TB es igual a 1024 GB

El código fuente:
archivo : test.php

```
<?
set_time_limit(0);
$tiempo_micro[1]= microtime();
$q_espacios = explode(" ",$tiempo_micro[1]);
$tiempo_[1]= $q_espacios[1]+$q_espacios[0];

$contento=file_get_contents('http://localhost/cargar_bytes.php'); /* cambiar por path real (URL)
http://tudominio/cargar_bytes.php)*/
$tamano_KB= strlen($contenido)/1024;

$tiempo_micro[2] = microtime();
$q_espacios= explode(" ",$tiempo_micro[2]);
$tiempo_[2] =$q_espacios[1] + $q_espacios[0];
$tiempo_utilizado = number_format(($tiempo_[2] - $tiempo_[1]),3, ".", "");

$velocidad= round($tamano_KB/$tiempo_utilizado,2);
echo 'Su velocidad de conexión es de: '.$velocidad.' Kbps <br> Se enviarón: '.$tamano_KB.
' Kb, Tiempo utilizado: '.$tiempo_utilizado.' Segundos <hr>';

for ($i=10; $i>=1; $i--){
$val_Kb=$i*100;
if($velocidad>=800)$velocidad_=800;
```

```
else $velocidad_=$velocidad/2;

if($velocidad>=$val_Kb && !$col){
echo '<div style="background-color:#F0F0F0; width:500px; float:left"><img width=".'($velocidad_).' height="8"
style="background-color: #FF0000" border="0"></div><strong>Su conexión</strong><br>';
$col=true;
}
echo '<div style="background-color:#F0F0F0; width:500px; float:left"><img width=".'($val_Kb/2).' height="8"
style="background-color: #000099" border="0"></div>'. $val_Kb.' Kbps<br>';
}
?>
```

archivo : cargar_bytes.php

```
<?
print str_repeat("i",1024000);
?>
```

Explicando:

Creamos el archivo cargar_bytes.php, que contendrá únicamente un proceso aleatorio de repeticiones hasta que alcance el valor de 1024000, 1000 KB, utilizando la función str_repeat(); que funciona en algo similar a un bucle. Se repetirá 1024000, la letra i

Llamaremos el archivo test.php,

Primero que nada utilizaremos la función set_time_limit(), para indicar que no hay limite en el tiempo de ejecución del script, recordemos que por default viene en el php.ini 30 segundos, y si este tiempo se alcanza se produce un error fatal de tiempo de ejecución.

Obtenemos el tiempo en ese instante en microsegundos

Luego hacemos una apertura del archivo anterior y obtenemos el contenido, mediante la función file_get_contents(), cabe destacar que el parámetro que debemos utilizar debe seguir el esquema: protocol:// para que se tome como una URL y se extraiga el resultado del proceso repetitivo anterior, de lo contrario se asumirá que el archivo es local y se ejecutara el código fuente directamente: es decir debemos colocar;

file_get_contents('http://eldominio/cargar_bytes.php'), para que funcione correctamente. Pero no imprimiremos en pantalla el contenido, solo lo cargaremos. Agrupamos en la variable \$tamano_KB, la longitud de caracteres de \$contenido, y lo dividimos entre 1024; para pasarlo de BYTES A KB.

Luego de la apertura, volvemos a obtener el tiempo en microsegundos para verificar cuanto tiempo transcurrió durante la ejecución del paso anterior y agrupamos ese valor en la variable \$tiempo_utilizado, le damos formato con la función number_format();

Obtenemos la velocidad en KBPS, dividiendo el total de KB cargados , entre el tiempo empleado;

Por ejemplo si: 100 Kb, son cargados en 2 segundos, la velocidad en ese instante es de 50 Kbps, pero puede variar dependiendo de muchos factores.

Imprimimos el resultado, y generamos un atractivo esquema visual.

Artículo por Emmanuel García De Caro

De Segundos a Horas, Minutos y Segundos

Por ejemplo si tengo 28240 Segundos, como hacer que imprima: 7:50:40 Horas

El proceso es sin duda sencillo, veamos el código y luego la explicación:

Código Fuente:

```
<?
function segundos_tiempo($segundos){
$minutos=$segundos/60;
$horas=floor($minutos/60);
$minutos2=$minutos%60;
$segundos_2=$segundos%60%60%60;
if($minutos2<10)$minutos2='0'.$minutos2;
if($segundos_2<10)$segundos_2='0'.$segundos_2;

if($segundos<60){ /* segundos */
$resultado= round($segundos).' Segundos';
}elseif($segundos>60 && $segundos<3600){/* minutos */
$resultado= $minutos2.':'.$segundos_2.' Minutos';
}else{/* horas */
$resultado= $horas.':'.$minutos2.':'.$segundos_2.' Horas';
}
return $resultado;
}
$segundos=date('h')*60*60+(date('i')*60)+date('s');

echo 'Segundos: '.$segundos.' Resultado: '.segundos_tiempo($segundos);
?>
```

Explicando:

Definimos una función con el nombre de `segundos_tiempo(ARG1)`, donde el ARG1 es el valor que indicamos (segundos) cuando llamamos la función.

Realizaremos una serie de operaciones aritméticas, para luego condicionar el resultado. `$Minutos`, tal como se ve; sera igual a la división entre `$segundo` y 60.
Ejemplo: 120 segundos / 60 es igual a 2 minutos.

`$horas` es igual a la división de `$minutos` entre 60 , y lo redondeamos hacia abajo con la función matemática `floor()`

`$minutos2` será igual al residuo (resto) devuelto por la división entre `$minutos` y 60.

Y `$segundos_2` el residuo de la división entre `$segundos`, 60, 60, 60 sucesivamente.

Que no son más que los datos acumulados en los procesos anteriores.

Luego iniciamos una condición para verificar si el número es menor a 10 y colocarle un 0 (cero) al comienzo: Por ejemplo 9 => 09 , el proceso lo repetimos con los segundos y minutos, con la horas seria innecesario.

Creamos otra condición para indicar si se ha definido Horas, minutos ó segundos y retornar el

formato y estilo correcto.

Si \$segundos es menor que 60 (entonces solo hay segundos)
`if($segundos<60)`

De lo contrario si \$segundos es mayor que 60 y es menor que 3600 (Se han definido minutos y segundos)
`elseif($segundos>60 && $segundos<3600)`

De lo contrario a todas estas (se ha definido horas, minutos y segundos)
`else`

Finalmente retornamos el resultado
`return $resultado;`

Y aplicamos la función, con los segundos actuales del servidor para verificar que nos devuelve la hora exacta.

*Artículo por **Emmanuel García De Caro***

¿Qué es OsCommerce?

En el 2000 nació osCommerce, una aplicación web de código abierto (OpenSource) que te permite montar una tienda virtual en cuestión de minutos para vender en Internet. Consta de dos partes principalmente el front y el back-end, es decir, la parte que vemos todo el mundo, la tienda virtual en si, y la parte de administración, donde podrás mantener tu propia tienda virtual, actualizando productos, insertando nuevas ofertas, categorías, idiomas, monedas, consultar los pedidos, los clientes.. y sin coste ninguno por parte del vendedor y sin necesidad ninguna de saber programación.

Front o Página Principal



Back End o Administración de la Tienda



Uno de los beneficios de ser código abierto, es que está siendo diariamente actualizada por su comunidad, añadiendo contribuciones de todo tipo. Módulos de pago, de envío, contribuciones para el diseño, como templates, así como nuevas funcionalidades, noticias, lector de rss...

osCommerce es de las mejores soluciones de código abierto existentes para la creación de tiendas virtuales, además de ser gratuita y sencilla de administrar. Programada en lenguaje PHP trabaja sobre un servidor Apache y usa MySQL como servidor de base de datos.

Actualmente casi todas las empresas de Hosting ofrecen una solución con OsCommerce ya instalada, aunque es realmente fácil instalar en cuestión de 5 clicks con el "install" que lleva incorporado.

El único pero que le podemos poner es que hay pocas contribuciones en español, casi todas en el universal inglés, aunque tenemos un buen soporte en la web española <http://oscommerce.qadram.com>

[OsCommerce](#)
[Contribuciones, addons...](#)

Características principales de OsCommerce

La instalación básica de oscommerce, tiene unas características por defecto, que luego podemos modificar bien por administración o con ayuda de contribuciones, como nuevos módulos de pago (e-pagado), de envío (Seur), etc..

Estas son las características principales de la instalación de Oscommerce:

Generales

- Los pedidos, clientes y productos se almacenan en una base de datos de fácil consulta via administración-web.
- Los clientes podrán comprobar el histórico y el estado de sus pedidos una vez registrados
- Los clientes pueden cambiar sus datos de perfil de usuario desde su apartado cliente.
- Múltiples direcciones de envío por usuario, para regalos por ejemplo
- Búsqueda de productos.
- Posibilidad de permitir a los usuarios valorar los productos comprados, además de comentarlos.
- Posibilidad de implementar un servidor seguro (SSL).
- Puede mostrar el número de productos en cada una de las categorías.
- Lista global o por categoría de los productos más vendidos y más vistos.
- Fácil e intuitiva navegación por categorías.
- Plataforma multi-idomas, por defecto estarán disponibles el español, inglés y alemán.

Producto

- Relaciones dinámicas entre productos.
- Descripciones de productos basadas en HTML.
- Generación automática de productos especiales.
- Controla la posibilidad de mostrar o no en la tienda virtual los productos agotados.
- Posibilidad de ofrecer a los usuarios la suscripción a una newsletter de novedades.

Pagos

- Medios de pago offline (transferencias, cheques, ingresos, etc.).
- Muchos medios de pago online (E-Pagado, PayPal, TPV virtual, etc.).
- Posibilidad de deshabilitar algunos medios de pago según la zona geográfica del usuario.
- Posibilidad de añadir el IVA por zonas geográficas y por productos, ya que por ejemplo Nigeria no tienen IVA

Envíos

- Precios de envío por peso, destino y precio.
- Precios reales disponibles en tiempo real para algunos operadores (UPS, FedEx).
- Envío gratuito según importe del pedido y destino
- Posibilidad de deshabilitar determinados servicios de envío en función de zonas geográficas.

Nota: Más adelante publicaremos nuevos artículos en los que profundizaremos más sobre el OsCommerce.

*Artículo por **Miguelo***

¿Qué son las contribuciones de OsCommerce?

Son algunas de las preguntas que surgen cuando instalas una tienda Oscommerce, ya que por necesidad vamos a echar mano de alguna contribución para nuestra tienda on-line.

Hay contribuciones de todo tipo, que sirven para modificar y/o ampliar el código php original

para adaptar la tienda a nuestras necesidades. Por ejemplo, si necesitas además de los 3 idiomas que viene por defecto o necesitas alguno más como catalán... O si quieres cambiar el aspecto de las Infoboxes o quieres que la cabecera cambie aleatoriamente de imagen, o quieres grabar el trazado de los usuarios de tu web durante su visitas... son muchas las mejoras, ampliaciones o cambios que se le pueden hacer a tu tienda en Oscommerce.

Todas están en la comunidad de Oscommerce, ánimo a subir la tuya. Las contribuciones son de uso a tu propio riesgo, es decir, debes de hacer backups de todos tus archivos ya que no te garantizan que vayan del todo bien, o que junto a otras contribuciones puedan dar problemas, pruébalas antes de usarlas. Backups, backups, backups..

Todas las contribuciones... Además de los foros de OsCommerce que nos puede venir muy bien .. Foro OsCommerce.

Las contribuciones las dividimos en 12 categorías

- **Módulos de Envío**
¿Necesitas enviar cosas al extranjero? ¿Sólo a tu país? Me cobran más si envío a Palma de Mallorca que a Barcelona, ¿qué hago? Añade el módulo necesario, los hay incluso de UPS, Seur... ya creados, otros por tarifas de producto, o por cantidades, por Códigos Postales, o no cobrar si llega el pedido a un importe.. Adapta tu Oscommerce a tus necesidades.
- **Módulos de Pago**
Módulos para cobrar con tarjeta, o contra reembolso, o por transferencia bancaria, para cobrar en tu país o en otros, así como módulos para e-pagado, paypal y otras empresas de cobro por web..
- **Módulos de Pedidos**
Sirven para hacer descuentos a grupos, para hacer pedidos con un mínimo de euros, o para controlar si necesitas añadir IVA o otro impuesto
- **Módulos de Crédito**
Es el más amplio en cuanto a contribuciones, las hay de todo tipo, para administración, como control por niveles, o para la tienda, como agregar e-mails en los pedidos o aceptar condiciones de compra... muy interesantes.
- **Idiomas**
Traducciones de la tienda tanto el apartado de administración como la tienda en si, bulgaro, coreano, turco, ruso, que idiomas necesitas?
- **Imágenes**
Principalmente botones en idiomas que no vienen por defecto en nuestro Oscommerce, así como el tratamiento de las imágenes, como creación de Thumbnails de las fotos de tus productos, iconos de todo tipo..
- **Informes**
Contribuciones que te informan cuando un cliente se ha dado de alta, o de las ventas mensuales de la tienda, editar pedidos, clientes, informes del catálogo en PDF, etc..
- **Zonas**
Contribuciones que sirven para determinar las ciudades, capitales, y regiones de países en concreto.
- **Info Boxes**
Añade cajas o infoboxes en las columnas laterales con nuevas funcionalidades o modificaciones de estas, como el betsellers con imágenes o la caja de Login mejorada, lector de rss, publicidad, etc..
- **Plantillas o Templates**
Botones, iconos, y estructuras para modificar tu tienda fácilmente en aspecto gráfico. Los PSDS o archivos de Photoshop para modificar los botones de la tienda..

- **Mejoras o Features**

Las hay de todo tipo, para mejorar la página de producto, más fotos, más opciones de compra, mejoras en cuanto a funcionalidades de la tienda Oscommerce, nuevos menús, comprar sin necesidad de crear cuenta d usuario,etc..

- **Otros**

De todo tipo sin clasificar, como crear la página 404 o como poner Flash en el catálogo, ...

Normalmente las contribuciones te las descargas en formato .zip o .rar y suelen ser archivos que debes reemplazar por los originales, o código que debes añadir a tus archivos. De todas formas, haz backup de todos tus archivos, ya que pueden no funcionar correctamente todas, o podemos tener algún error al editar nuestros archivos.

*Artículo por **Miguelo***

Zonas horarias

Otras de las grandes necesidades originadas por la distancia entre diferentes países y en algunos casos ciudades es la hora.

Por ejemplo: ¿Si un usuario se encuentra en cualquier parte del mundo, como hacer para que cuando emita una opinión, esta aparezca con su hora local ó de su zona horaria?

Para solucionar este gran problema y tantos otros relacionados con este género utilizaremos las zonas horarias de las principales ciudades del mundo.

Utilizaremos dos aspectos cruciales o decisivos la hora con respecto al meridiano de la ciudad local utilizada como base para calcular la diferencia con respecto a la ciudad distante tomando como referencia la hora del Meridiano de Greenwich u hora 0 (CERO).

Veamos el código fuente y luego la explicación como de costumbre:

```
<?
if(isset($_POST[hora])){
    settype ($_POST[hora],"integer");
    settype ($_POST[min],"integer");
    settype ($_POST[seg],"integer");
    if($_POST[hora]<=0 OR $_POST[hora]>12){
        echo '<strong> Hora incorrecta </strong>';
    }elseif($_POST[min]<=0OR $_POST[min]>60){
        echo '<strong> Minutos incorrectos </strong>';
    }elseif($_POST[seg]<=0 OR $_POST[seg]>60){
        echo '<strong> Segundos Incorrectos </strong>';
    }else{
        $hor=$_POST[hora];
        if($_POST[tm]=='pm' && $_POST[hora]>=1 && $_POST[hora]<12){
            $_POST[hora]+=12;
        }
        $total_minutos_=floor($_POST[hora]*60)+$_POST[min];
        $dia="del mismo día";
        if($_POST[ciudad1]>=0){ // es positiva la zona h
            $gmt= (int) $total_minutos_-$POST[ciudad1];
        }else{ // es negativo
            $_POST[ciudad1]= (int) $_POST[ciudad1]*-1;
        }
    }
}
```

```
$gmt= $total_minutos_+$_POST[ciudad1];
}
if($gmt<0){
$dia=' del día anterior';
$gmt+=1440; //minutos de 1 dia
}
if($_POST[ciudad2]>0){
$tiempo2=$_POST[ciudad2]+$total_minutos_;
}else{
$tiempo2=$_POST[ciudad2]+$gmt;
if($tiempo2 > 1440)
{
$dia = "del dia siguiente";
$tiempo2-= 1440;
}
}
if ($tiempo2 < 0)
{
$dia = "del dia anterior";
$tiempo2 += 1440;
}
}
}
$hora_de_la_otra_ciudad=floor($tiempo2/60);

echo 'En la otra ciudad son: '.$hora_de_la_otra_ciudad.' : '.$_POST[min].' : '.$_POST[seg].' '.$dia;
}
} ?>

<form action="<? echo $PHP_SELF?>" method="post">Si en mi pc son las
<input name="hora" type="text" id="hora" value="<? echo $hor?>" size="2" maxlength="2">
:
<input name="min" type="text" id="min" value="<? echo $_POST["min"] ?>" size="2" maxlength="2">
:
<input name="seg" type="text" id="seg" value="<? echo $_POST["seg"] ?>" size="2" maxlength="2">
<select name="tm" id="tm">
<option value="am">am</option>
<option value="pm">pm</option>
</select>
: y estoy en :

<SELECT name=ciudad1 size=1 id="ciudad1">
<OPTION value=180>Addis Ababa</OPTION>
<OPTION value=570>Adelaida</OPTION>
<OPTION value=180>Aden</OPTION>
<OPTION value=-360>Aklavik</OPTION>
<OPTION value=60>Argel</OPTION>
<OPTION value=120>Amman</OPTION>
<OPTION value=60>Amsterdam</OPTION>
<OPTION value=720>Anadyr</OPTION>
<OPTION value=120>Ankara</OPTION>
<OPTION value=180>Antananarivo</OPTION>
<OPTION value=-240>Asunció</OPTION>
<OPTION value=120>Atenas</OPTION>
<OPTION value=-300>Atlanta</OPTION>
<OPTION value=-360>Austin</OPTION>
<OPTION value=0>Azores</OPTION>
<OPTION value=180>Baghdad</OPTION>
<OPTION value=420>Bangkok</OPTION>
<OPTION value=60>Barcelona</OPTION>
<OPTION value=180>Beirut</OPTION>
<OPTION value=60>Belgrado</OPTION>
<OPTION value=60>Berlín</OPTION>
<OPTION value=-240>Bermuda</OPTION>
<OPTION value=60>Berna</OPTION>
<OPTION value=120>Biel</OPTION>
<OPTION value=-300>Bogotá</OPTION>
```

```
<OPTION value=-300>Boston</OPTION>
<OPTION value=-180>Brasilia</OPTION>
<OPTION value=600>Brisbane</OPTION>
<OPTION value=60>Bruselas</OPTION>
<OPTION value=120>Bucarest</OPTION>
<OPTION value=60>Budapest</OPTION>
<OPTION value=-180>Buenos Aires</OPTION>
<OPTION value=330>Calcuta</OPTION>
<OPTION value=-240 selected>Caracas</OPTION>
<OPTION value=0>Casablanca</OPTION>
<OPTION value=-300>Chicago</OPTION>
<OPTION value=120>Ciudad del Cabo</OPTION>
<OPTION value=60>Copenhague</OPTION>
<OPTION value=-360>Dallas</OPTION>
<OPTION value=180>Dar es Salaam</OPTION>
<OPTION value=570>Darwin</OPTION>
<OPTION value=-420>Denver</OPTION>
<OPTION value=-300>Detroit</OPTION>
<OPTION value=360>Dhaka</OPTION>
<OPTION value=0>Dublín</OPTION>
<OPTION value=-420>Edmonton</OPTION>
<OPTION value=120>Estambul</OPTION>
<OPTION value=60>Estocolmo</OPTION>
<OPTION value=180>El Cairo</OPTION>
<OPTION value=60>Francfort</OPTION>
<OPTION value=60>Ginebra</OPTION>
<OPTION value=-360>Guatemala</OPTION>
<OPTION value=420>Hanoi</OPTION>
<OPTION value=120>Harare</OPTION>
<OPTION value=120>Helsinki</OPTION>
<OPTION value=420>Ho Chi Minh City</OPTION>
<OPTION value=480>Hong Kong</OPTION>
<OPTION value=600>Honolulu</OPTION>
<OPTION value=-360>Houston</OPTION>
<OPTION value=-300>Indianápolis</OPTION>
<OPTION value=300>Islamabad</OPTION>
<OPTION value=420>Jakarta</OPTION>
<OPTION value=120>Jerusalén</OPTION>
<OPTION value=120>Johanesburgo</OPTION>
<OPTION value=270>Kabúl</OPTION>
<OPTION value=720>Kamchatka</OPTION>
<OPTION value=300>Kathmandú</OPTION>
<OPTION value=120>Khartoum</OPTION>
<OPTION value=120>Kigali</OPTION>
<OPTION value=-300>Kingston</OPTION>
<OPTION value=480>Kuala Lumpur</OPTION>
<OPTION value=180>Kuwait</OPTION>
<OPTION value=-240>La Habana</OPTION>
<OPTION value=-240>La Paz</OPTION>
<OPTION value=0>Las Palmas</OPTION>
<OPTION value=60>Lagos</OPTION>
<OPTION value=-300>Lima</OPTION>
<OPTION value=0>Lisboa</OPTION>
<OPTION value=0>Londres</OPTION>
<OPTION value=-480>Los Ángeles</OPTION>
<OPTION value=60>Madrid</OPTION>
<OPTION value=-360>Managua</OPTION>
<OPTION value=480>Manila</OPTION>
<OPTION value=240>Masqat</OPTION>
<OPTION value=600>Melburne</OPTION>
<OPTION value=-360>Méjico D.F.</OPTION>
<OPTION value=300>Miami</OPTION>
<OPTION value=60>Milán</OPTION>
<OPTION value=-360>Minneápolis</OPTION>
<OPTION value=-180>Montevideo</OPTION>
```

```
<OPTION value=-300>Montreal</OPTION>
<OPTION value=180>Moscú</OPTION>
<OPTION value=330>Mumbai</OPTION>
<OPTION value=60>Múnich</OPTION>
<OPTION value=180>Nairobi</OPTION>
<OPTION value=60>Nápoles</OPTION>
<OPTION value=-300>Nassau</OPTION>
<OPTION value=330>Nueva Delhi</OPTION>
<OPTION value=-360>Nueva Orleans</OPTION>
<OPTION value=-300>Nueva York</OPTION>
<OPTION value=60>Oslo</OPTION>
<OPTION value=-240>Ottawa</OPTION>
<OPTION value=60>París</OPTION>
<OPTION value=480>Pequín</OPTION>
<OPTION value=480>Perth</OPTION>
<OPTION value=-420>Phoenix</OPTION>
<OPTION value=420>Phnom Penh</OPTION>
<OPTION value=60>Praga</OPTION>
<OPTION value=540>Pyongyang</OPTION>
<OPTION value=0>Reikiavik</OPTION>
<OPTION value=-180>Río de Janeiro</OPTION>
<OPTION value=180>Riyadh</OPTION>
<OPTION value=60>Roma</OPTION>
<OPTION value=-240>St. John's</OPTION>
<OPTION value=-360>St. Paul</OPTION>
<OPTION value=-480>San Francisco</OPTION>
<OPTION value=-240>San Juan</OPTION>
<OPTION value=-360>San Salvador</OPTION>
<OPTION value=-300>Santo Domingo</OPTION>
<OPTION value=-240>Santiago</OPTION>
<OPTION value=-180>Sao Paulo</OPTION>
<OPTION value=-480>Seattle</OPTION>
<OPTION value=540>Seúl</OPTION>
<OPTION value=480>Shanghai</OPTION>
<OPTION value=480>Singapur</OPTION>
<OPTION value=60>Sofía</OPTION>
<OPTION value=720>Suva</OPTION>
<OPTION value=600>Sydney</OPTION>
<OPTION value=480>Taipei</OPTION>
<OPTION value=120>Tallinn</OPTION>
<OPTION value=300>Tashkent</OPTION>
<OPTION value=-210>Teherán</OPTION>
<OPTION value=540>Tokio</OPTION>
<OPTION value=-300>Toronto</OPTION>
<OPTION value=60>Turín</OPTION>
<OPTION value=-300>Vancouver</OPTION>
<OPTION value=60>Varsovia</OPTION>
<OPTION value=60>Venecia</OPTION>
<OPTION value=60>Viena</OPTION>
<OPTION value=600>Vladivostok</OPTION>
<OPTION value=-300>Washington</OPTION>
<OPTION value=720>Wellington</OPTION>
<OPTION value=-360>Winnipeg</OPTION>
<OPTION value=390>Yangon</OPTION>
<OPTION value=60>Zagreb</OPTION>
<OPTION value=60>Zurich</OPTION>
</SELECT>
<br>
Entonces en

<SELECT name=ciudad2 size=1 id="ciudad2">
  <OPTION value=180>Addis Ababa</OPTION>
  <OPTION value=570>Adelaida</OPTION>
  <OPTION value=180>Aden</OPTION>
  <OPTION value=-360>Aklavik</OPTION>
```

<OPTION value=60>Argel</OPTION>
<OPTION value=120>Amman</OPTION>
<OPTION value=60>Amsterdam</OPTION>
<OPTION value=720>Anadyr</OPTION>
<OPTION value=120>Ankara</OPTION>
<OPTION value=180>Antananarivo</OPTION>
<OPTION value=-240>Asunció</OPTION>
<OPTION value=120>Atenas</OPTION>
<OPTION value=-300>Atlanta</OPTION>
<OPTION value=-360>Austin</OPTION>
<OPTION value=0>Azores</OPTION>
<OPTION value=180>Baghdad</OPTION>
<OPTION value=420>Bangkok</OPTION>
<OPTION value=60>Barcelona</OPTION>
<OPTION value=180>Beirut</OPTION>
<OPTION value=60>Belgrado</OPTION>
<OPTION value=60>Berlín</OPTION>
<OPTION value=-240>Bermuda</OPTION>
<OPTION value=60>Berna</OPTION>
<OPTION value=120>Biel</OPTION>
<OPTION value=-300>Bogotá</OPTION>
<OPTION value=-300>Boston</OPTION>
<OPTION value=-180>Brasilia</OPTION>
<OPTION value=600>Brisbane</OPTION>
<OPTION value=60>Bruselas</OPTION>
<OPTION value=120>Bucarest</OPTION>
<OPTION value=60>Budapest</OPTION>
<OPTION value=-180>Buenos Aires</OPTION>
<OPTION value=330>Calcuta</OPTION>
<OPTION value=-240>Caracas</OPTION>
<OPTION value=0>Casablanca</OPTION>
<OPTION value=-300>Chicago</OPTION>
<OPTION value=120>Ciudad del Cabo</OPTION>
<OPTION value=60>Copenhague</OPTION>
<OPTION value=-360>Dallas</OPTION>
<OPTION value=180>Dar es Salaam</OPTION>
<OPTION value=570>Darwin</OPTION>
<OPTION value=-420>Denver</OPTION>
<OPTION value=-300>Detroit</OPTION>
<OPTION value=360>Dhaka</OPTION>
<OPTION value=0>Dublín</OPTION>
<OPTION value=-420>Edmonton</OPTION>
<OPTION value=120>Estambul</OPTION>
<OPTION value=60>Estocolmo</OPTION>
<OPTION value=180>El Cairo</OPTION>
<OPTION value=60>Francfort</OPTION>
<OPTION value=60>Ginebra</OPTION>
<OPTION value=-360>Guatemala</OPTION>
<OPTION value=420>Hanoi</OPTION>
<OPTION value=120>Harare</OPTION>
<OPTION value=120>Helsinki</OPTION>
<OPTION value=420>Ho Chi Minh City</OPTION>
<OPTION value=480>Hong Kong</OPTION>
<OPTION value=600>Honolulu</OPTION>
<OPTION value=-360>Houston</OPTION>
<OPTION value=-300>Indianápolis</OPTION>
<OPTION value=300>Islamabad</OPTION>
<OPTION value=420>Jakarta</OPTION>
<OPTION value=120>Jerusalén</OPTION>
<OPTION value=120>Johanesburgo</OPTION>
<OPTION value=270>Kabúl</OPTION>
<OPTION value=720>Kamchatka</OPTION>
<OPTION value=300>Kathmandú</OPTION>
<OPTION value=120>Khartoum</OPTION>
<OPTION value=120>Kigali</OPTION>

```
<OPTION value=-300>Kingston</OPTION>
<OPTION value=480>Kuala Lumpur</OPTION>
<OPTION value=180>Kuwait</OPTION>
<OPTION value=-240>La Habana</OPTION>
<OPTION value=-240>La Paz</OPTION>
<OPTION value=0>Las Palmas</OPTION>
<OPTION value=60>Lagos</OPTION>
<OPTION value=-300>Lima</OPTION>
<OPTION value=0>Lisboa</OPTION>
<OPTION value=0>Londres</OPTION>
<OPTION value=-480>Los Ángeles</OPTION>
<OPTION value=60>Madrid</OPTION>
<OPTION value=-360>Managua</OPTION>
<OPTION value=480>Manila</OPTION>
<OPTION value=240>Masqat</OPTION>
<OPTION value=600>Melburne</OPTION>
<OPTION value=-360>Méjico D.F.</OPTION>
<OPTION value=300>Miami</OPTION>
<OPTION value=60>Milán</OPTION>
<OPTION value=-360>Minneapolis</OPTION>
<OPTION value=-180>Montevideo</OPTION>
<OPTION value=-300>Montreal</OPTION>
<OPTION value=180>Moscú</OPTION>
<OPTION value=330>Mumbai</OPTION>
<OPTION value=60>Múnich</OPTION>
<OPTION value=180>Nairobi</OPTION>
<OPTION value=60>Nápoles</OPTION>
<OPTION value=-300>Nassau</OPTION>
<OPTION value=330>Nueva Delhi</OPTION>
<OPTION value=-360>Nueva Orleans</OPTION>
<OPTION value=-300>Nueva York</OPTION>
<OPTION value=60>Oslo</OPTION>
<OPTION value=-240>Ottawa</OPTION>
<OPTION value=60>París</OPTION>
<OPTION value=480>Pequín</OPTION>
<OPTION value=480>Perth</OPTION>
<OPTION value=-420>Phoenix</OPTION>
<OPTION value=420>Phnom Penh</OPTION>
<OPTION value=60>Praga</OPTION>
<OPTION value=540>Pyongyang</OPTION>
<OPTION value=0>Reikiavik</OPTION>
<OPTION value=-180>Río de Janeiro</OPTION>
<OPTION value=180>Riyadh</OPTION>
<OPTION value=60>Roma</OPTION>
<OPTION value=-240>St. John's</OPTION>
<OPTION value=-360>St. Paul</OPTION>
<OPTION value=-480>San Francisco</OPTION>
<OPTION value=-240>San Juan</OPTION>
<OPTION value=-360>San Salvador</OPTION>
<OPTION value=-300>Santo Domingo</OPTION>
<OPTION value=-240>Santiago</OPTION>
<OPTION value=-180>Sao Paulo</OPTION>
<OPTION value=-480>Seattle</OPTION>
<OPTION value=540>Seúl</OPTION>
<OPTION value=480>Shanghai</OPTION>
<OPTION value=480>Singapur</OPTION>
<OPTION value=60>Sofia</OPTION>
<OPTION value=720>Suva</OPTION>
<OPTION value=600>Sydney</OPTION>
<OPTION value=480>Taipei</OPTION>
<OPTION value=120>Tallinn</OPTION>
<OPTION value=300>Tashkent</OPTION>
<OPTION value=-210>Teherán</OPTION>
<OPTION value=540>Tokio</OPTION>
<OPTION value=-300>Toronto</OPTION>
```



```
<OPTION value=60>Turín</OPTION>
<OPTION value=-300>Vancouver</OPTION>
<OPTION value=60>Varsovia</OPTION>
<OPTION value=60>Venecia</OPTION>
<OPTION value=60>Viena</OPTION>
<OPTION value=600>Vladivostok</OPTION>
<OPTION value=-300 selected>Washington</OPTION>
<OPTION value=720>Wellington</OPTION>
<OPTION value=-360>Winnipeg</OPTION>
<OPTION value=390>Yangon</OPTION>
<OPTION value=60>Zagreb</OPTION>
<OPTION value=60>Zurich</OPTION>
</SELECT>
son

<input type="submit" value="Ver hora"></form>
```

Explicación:

Primero que nada iniciaremos una condición para comprobar si `$_POST[hora]` se ha definido en algún momento a través de la función `isset()`; (solo tomara valor cuando se envíen datos a través de HTTP-POST)

Cambiaremos el tipo de dato de `$_POST[hora]`, `$_POST[min]` y `$_POST[segundo]` de String o Cadena a Integer o Entero a través de la función `settype()`:

Realizaremos una validación con una serie de condiciones para comprobar:

Sí El Valor es menor que 0 (cero) o mayor que 12 ó 60 según sea el caso, de ser contrarias todas estas situaciones iniciaremos el ensamblaje de una serie de instrucciones.

Condición: `if($_POST[tm]='pm' && $_POST[hora]>=1 && $_POST[hora]<12)`

Expresamos si la hora es PM pero además esa hora debe ser mayor o igual que 1 y menor a 12, pues si evalúa TRUE nos hará llevar la hora a en base 24 :

Por ejemplo: si se inserta 1 HORA PM debemos llevarlo a 13 horas, para ello sumamos `$_POST[hora]` más 12 horas transcurridas como constante.

Luego asociamos a la variable `$total_minutos` el valor devuelto por la multiplicación entre `$_POST[hora]` y 60 más los minutos "`$_POST[min]`" redondeado hacia abajo mediante la función matemática `floor()`, para llevar la expresión de Hora:Minutos a minutos totales.

Definimos el valor de la variable `$dia`, este valor se mantendrá siempre y cuando las circunstancias así lo toleran, pues puede ser cambiado o remplazado si alguna de las condiciones subsiguientes evalúa TRUE.

Sí `$_POST[ciudad1]` es mayor o igual a 0,

Recordemos que el valor de `$_POST[ciudad1]` será la hora con respecto al meridiano de la ciudad tomada como referencia, Ese entero puede ser negativo o positivo y esta condición evaluara TRUE solo si es positivo; para realizar una resta entre `$total_minutos` y el valor de `$_POST[ciudad1]` asociándolo a la variable `$gmt`.

De lo contrario a sí `$_POST[ciudad1]` es mayor o igual a 0,

Quiere decir que `$_POST[ciudad1]` es negativo y lo pasaremos a positivo, multiplicándolo por `-1`.

```
$_POST[ciudad1]*-1;
```

Y realizamos una suma entre `$_POST[ciudad1]` y `$total_minutos_`, lo contrario del caso anterior.

¿ Para que y con que sentido?

Debemos determinar si el valor resultante es positivo o negativo para constatar si la diferencia es del día anterior o del día siguiente.

Si es menor a 0 (negativo)

Cambiamos el valor anteriormente asignado a `$dia` por día anterior y le agregamos a `$gmt` "1440" minutos totales de un día.

Sí `$_POST[ciudad2]` es mayor que 0 , esta hacia el ESTE del meridiano

Sumamos `$tiempo2` y `$total_minutos_`, y lo asociamos a `$tiempo2`.

De lo contrario a Sí `$_POST[ciudad2]` es mayor que 0.

Realizamos un operación similar a la anterior pero esta vez le sumamos el valor de `$gmt`, definido en procesos anteriores.

Dentro de ese marco realizamos dos condiciones mas:

Si `$tiempo2` es mayor que el total de minutos de un día (1440), entonces es el día siguiente.

Si `$tiempo2` es menor que 0 entonces es del día anterior.

Finalmente sin importar la rutina empleada retornamos el resultado, transformando esos minutos resultantes en horas : para ello lo dividiremos entre 60 y redondeamos ese valor hacia abajo, asociamos el resultado a la variable: `$hora_de_la_otra_ciudad`

Y damos salida al navegador por medio de un echo.

*Artículo por **Emmanuel García De Caro***

Detectar país del visitante

Una de las grandes necesidades de todo desarrollador web radica al momento de conocer el país del visitante bien sea para redireccionarlos a módulos o páginas con particularidades o características propias de cada determinación, entre otras funciones que se pueden desarrollar en el entorno.

Ejemplo: Si tenemos una página de productos y catálogos (Comercio Web) y deseamos mostrare al visitante el precio del referido producto o artículo expresado monetariamente en su moneda local.

Entre otras infinidades de particularidades.

En realidad el proceso de la detección del país a través del nick se realiza a partir de la dirección IP devuelta por superglobales como \$_SERVER['REMOTE_ADDR'], tomando como soporte o base una serie criterios de posibilidades almacenados lógicamente en archivos de base de datos, para su posterior validación y deducción de sus cuatros componentes esenciales menores todos a 255.

En Php solo existen alguna funciones que permiten obtener información acerca de maquinas conectadas a una red especifica por medio de las Funciones de Red, pero no creamos que una función o un par de funciones nos harán todo el trabajo; por el contrario solo serán bases para todo el ensamblaje.

Ciertamente la empresa Maxmind, desarrolladora por excelencia de este tipo de herramientas con soporte para diferentes tecnologías "GeoIp", logro solucionar este gran problema con un 95% de efectividad con un conjunto de funciones y un archivo de bases de datos.

Para poder utilizar esta herramienta deberemos descargar la base de datos:

[GeoIP.dat.gz](#)

Biblioteca de vínculos y funciones:

[Geoip.inc](#)

Fianlemente realizaremos un test con las siguientes lineas:

```
<?
require("geoip.inc"); /*requerimos la biblioteca o liberia */
$abir_bd = geoip_open("GeoIP.dat",GEOIP_STANDARD); /* apertura y lectura del archivo utilizando la constante
GEOIP_STANDARD como forma de lectura que adquiere valor en geoip.inc */
/* imprimimos el nick del país */
echo geoip_country_name_by_addr($abir_bd, $_SERVER['REMOTE_ADDR']);
/* liberamos memoria cerrando el resorce devuelto por $abir_bd de la apertura*/
geoip_close($abir_bd);
?>
```

Desventajas y Sugerencias:

Seria un gravísimo error que incluyéramos estas lineas en todas nuestra páginas, recordemos que cada vez que esta se cargue llamara a si mismo el archivo geoip.inc y GeoIP.dat, este ultimo supera los 700 Kb de tamaño y puede considerablemente extender el tiempo de carga de la página.

Sugerencias:

Podemos almacenar el valor del nick o código del país en una session (\$_SESSION) preferiblemente, o en su defecto en COOKIES; para que de este modo solo se lea una vez ya que lógicamente el visitante no cambiara de IP a cada vez que refresque o navegue en una página.

*Artículo por **Emmanuel García De Caro***

Obligaciones empresariales con OsCommerce en España

La Protección de Datos de Carácter Personal es una materia que preocupa a todas las empresas, principalmente por las fuertes sanciones económicas que impone la Agencia Española de Protección de Datos en caso de incumplimiento.

El deber de las empresas en esta materia no sólo debe consistir en la adaptación a la normativa vigente, sino también en la aplicación práctica de los principios de protección de datos en el seno de la empresa, dentro de cada uno de los departamentos, involucrando a todo el personal... Las consecuencias del incumplimiento de la normativa no se quedan sólo en las sanciones de la Agencia, sino que van más allá y afectan a todos.

El cumplimiento de lo establecido en la LOPD (LO 15/1999 de Protección de Datos de Carácter Personal) y su normativa de desarrollo (RD 994/1999, Reglamento de Medidas de Seguridad de los Ficheros Automatizados que contengan datos de carácter personal) pasa no sólo por la legalización, legitimación y protección de los ficheros de datos de carácter personal, sino también por su aplicación práctica en el seno de las organizaciones, es decir, la incorporación a la dinámica de la empresa de los principios rectores de la Protección de Datos de Carácter Personal.

La incorporación a la dinámica de la empresa de los principios rectores de la Protección de Datos de Carácter Personal, adquiere una gran importancia desde el momento en que las consecuencias de su incumplimiento conllevan grandes responsabilidades tanto para la Organización como para el personal que trata o accede a los datos de carácter personal, es decir, las sanciones ya no sólo son administrativas y dirigidas a la Organización en sí, sino que además de ellas se pueden derivar responsabilidades civiles, penales y laborales.

- **Administrativas:** Sanciones contempladas en la LOPD y que son impuestas por la Agencia Española de Protección de Datos en el ejercicio de sus funciones, ya sea por inspección de oficio o abierta a instancia de parte. Estas sanciones son de carácter económico y su cuantía está entre 601,01 y 601.012,10 Euros.
- **Civiles:** Artículos del Código Civil relativos a la Responsabilidad Contractual y Extracontractual (arts. 1902 y 1903 CC). Así cuando determinado servicio es contratado a un tercero ajeno a la propia organización e implique un acceso a los ficheros de datos de carácter personal, deberá estar precedido del correspondiente contrato de acceso a datos en el que se limiten las facultades del tercero en cuanto al tratamiento de los datos de carácter personal, se especifiquen las medidas de seguridad que deberán ser implantadas o cumplidas por el tercero para la protección del fichero, y se determinen las responsabilidades derivadas del incumplimiento de la LOPD o de lo establecido en el contrato.
- **Penales:** El Código Penal tipifica los delitos contra la intimidad y concretamente el descubrimiento y revelación de secretos en los artículos 197 y siguientes. En este sentido, ha sido condenado recientemente un funcionario de la Seguridad Social, a cinco años y tres meses de prisión, una multa de 330.556 Euros y la inhabilitación especial de 11 años para empleo o cargo público, por vender datos personales de cotizantes a una empresa privada.
- **Laborales:** La fuga de datos, un tratamiento inadecuado de los ficheros de datos de carácter personal, un acceso no autorizado a los datos del fichero, una protección inadecuada de los ficheros, pueden venir derivadas de cualquier puesto laboral dentro del seno de la Organización. Cuando una cadena de fallos deriva en la imposición de una sanción a la Organización, es frecuente que además se deriven responsabilidades laborales.

En consecuencia, se hace absolutamente necesario el establecimiento de una serie de medidas adicionales para la correcta aplicación de la Ley de Protección de Datos en las empresas, tendentes principalmente a informar y formar al personal que trata los datos, independientemente de su cargo o función.

Como principales acciones adicionales encaminadas a informar y formar al personal sobre los principios del tratamiento de datos de carácter personal encontramos las siguientes:

1. Establecimiento de una Política de Tratamiento de Datos en la Organización. Esta Política de Tratamiento de Datos deberá entregarse a cada uno de los empleados que entre sus funciones esté el tratamiento de datos de carácter personal, quién deberá firmarla tras su conocimiento.
En esta Política se suele informar sobre el carácter y finalidad de la LOPD y su normativa de desarrollo, las obligaciones básicas impuestas, así como los diferentes principios, acciones y procedimientos a adoptar en la gestión diaria de la empresa. Igualmente, se puede informar al empleado sobre otras medidas o políticas de seguridad de los sistemas de información implementados por la Organización, así como de las responsabilidades que se pueden derivar de su incumplimiento.
2. Formación en materia de Protección de Datos. El establecimiento de seminarios, conferencias o jornadas de formación en materia de protección de datos en el seno de las Organizaciones es otra de las medidas que se pueden adoptar. Como principales ventajas de la formación encontramos: -Información y Formación para el empleado. -Formación personalizada para cada empresa. Se analizarán los casos concretos de cada organización y su problemática particular. -Mayor participación de los empleados. Podrán consultar y comentar las particularidades de cada puesto.

*Artículo por **Miguelo***

