

# Apache, PHP y MySQL

Alvaro Marín Illera [split77@terra.es](mailto:split77@terra.es)

*Instalación básica del servidor web Apache, con el módulo de PHP e interacción con el servidor MySQL.*

## Introducción

Apache es uno de los servidores web más utilizados actualmente en Internet según los datos de Netcraft. En el momento en el que estoy escribiendo este How-to, se acaba de liberar la versión 2.0.35, con importantes mejoras como el soporte de threads para sistemas multiprocesador.

Otra de las características importantes de Apache, es que tiene una licencia que aunque no es la GPL, permite ver el código fuente, al contrario de otros productos que se jactan de seguros y que curiosamente, son los que más incidencias tienen.

Vamos a usar también PHP, acrónimo de PHP Hipertext Preprocessor (típica recursividad como la de GNU :)), para la programación del lado del servidor. Estos lenguajes, funcionan de modo que cuando un cliente envía una petición de GET al servidor de una determinada página PHP, éste antes de entregarla, interpreta el código PHP (acotado entre las etiquetas `<?php>`) y envía la página html resultante.

Una de las ventajas que tiene PHP es su potente API para poder conectar con servidores de Bases de datos como PostgreSQL y MySQL. En este caso, vamos a usar este último, ya que es ideal para bases de datos de pequeño y mediano tamaño por su rapidez. Haremos una Base de Datos y lanzaremos consultas SQL contra ella mediante PHP.

## Instalación y configuración

Cómo no, vamos a implementar todo esto bajo GNU/Linux Debian Woody O:) , pero también se puede realizar bajo otras distribuciones fácilmente.

Instalamos primeramente los paquetes necesarios para apache (v1.3.23-1):

```
apt-get install apache-common apache
```

A continuación vamos con el MySQL(v3.23.49-7):

```
apt-get install mysql-common mysql-client mysql-server
```

Y por último el soporte para PHP (v4.1.2-3.1):

```
apt-get install php4 php4-cgi php4-mysql
```

Los archivos de configuración de Apache quedan en el directorio `/etc/apache/` , donde podemos encontrar el `httpd.conf`. No hay que tocar nada para que el servidor funcione, tan solo descomentamos las líneas que se refieren a php:

```
LoadModule php4_module /usr/lib/apache/1.3/libphp4.so
AddType application/x-httpd-php .php
AddType application/x-httpd-php-source .phps
```

Volvemos a reiniciar el apache con `/etc/init.d/apache restart`

Ahora ya podríamos acceder a cualquier documento que esté bajo la el directorio raíz del servidor web, que por defecto es `/var/www` , poniendo en nuestro navegador `http://localhost/...`

Para hacer funcionar el servidor MySQL, lo hacemos con `/etc/init.d/mysql start`. El archivo de configuración es `/etc/mysql/my.cnf`. Tan solo señalar que para que se pueda acceder desde una máquina distinta al prompt de MySQL, debemos de comentar la opción `skip-networking`, ya que por defecto, por motivos de seguridad no se abre el puerto 3306 correspondiente y está deshabilitado el acceso por red.

Para conectarnos a MySQL desde el equipo local, basta con hacer:

```
mysql -h localhost -u root
```

Nada más instalarse, se crea un usuario llamado root sin password. Esto es conveniente cambiarlo en la Base de Datos de user del servidor. Lo conseguimos de la siguiente forma:

```
mysql -h localhost -u root#A continuación nos sale el prompt de MySQL:
mysql>
```

```
#Mostramos todas las Bases de Datos exitentes(nótese ";" para todos los comandos y sentencias):
```

```
mysql> show databases;
```

```
+-----+
| Database |
+-----+
| ghosteca |
| mail     |
| mysql    |
| test     |
+-----+
```

```
4 rows in set (0.02 sec)
```

```
#Indicamos la BD con la que vamos a trabajar a partir de ahora,
```

```
#para hacer las selects, inserts y todas las sentencias SQL.
```

```
#Si posteriormente queremos saber cuál elegimos, hacemos select database();
```

```
#para verlo.
```

```
mysql> use mysql;
```

```
Reading table information for completion of table and column names
```

```
You can turn off this feature to get a quicker startup with -A
```

```
Database changed
```

```
#Mostramos las Tablas que hay en la BD:
```

```
mysql> show tables;
```

```
+-----+
| Tables_in_mysql |
+-----+
| columns_priv    |
| db              |
| func            |
| host            |
| tables_priv     |
| user            |
+-----+
```

```
6 rows in set (0.00 sec)
```

```
mysql>
```

```
#Si hacemos ahora un "select * from user;" no saldrá toda la tabla
```

```
#de usuarios aceptados en el sistema. El campo password, puede estar cifrado,
```

```
#usando la función password() o bien, en texto plano.
```

```
#Vamos a introducir un usuario de username prueba y con password prueba también
```

```

#Para que en la tabla la contraseña salga cifrada, lo hacemos con password().

mysql> insert into user (Host,User>Password) values ('localhost','phpuser',password('phpuser'));
Query OK, 1 row affected (0.05 sec)

#Si hacemos una select de la tabla de nuevo, veremos ya el nuevo usuario
#pero podremos usarlo hasta que no reiniciemos el servicio.

#Vamos a cambiar ahora la password de root:
mysql>update user SET Password=password('aupa_elPuNk77') where User='root';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

#Creamos una BD con la que trabajaremos a continuación:
mysql>create database BDphp;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

#Damos permisos totales al usuario sobre las tablas de BDphp y actualizamos:
mysql>GRANT ALL ON BDphp.* TO phpuser@localhost IDENTIFIED BY "phpuser";
mysql>flush privileges;
mysql>quit;

```

Ya tenemos entonces el usuario phpuser creado para poder manejar nuestra base de datos. A partir de ahora por tanto, nos conectaremos a mysql desde él de la siguiente forma:

```
mysql -h localhost -u phpuser -p
```

Con lo que nos pedirá la contraseña, que en este caso es "phpuser" (lo siento, no tengo compromiso alguno con la seguridad :P ).

Pues vamos a crear una tabla de ejemplo para poder usarla luego con PHP

```

mysql -h localhost -u phpuser -p
mysql>use BDphp;
mysql> create table TPersona(
    -> DNI char(10) NOT NULL,
    -> Nombre char(20) NOT NULL,
    -> Apellidos char(60) NOT NULL,
    -> Mail char(60) NOT NULL,
    -> PRIMARY KEY (DNI));
Query OK, 0 rows affected (0.01 sec)

mysql> show tables;
+-----+
| Tables_in_BDphp |
+-----+
| TPersona        |
+-----+
1 row in set (0.00 sec)

mysql>quit;
Bye

```

Algún otro comando interesante y a tener en cuenta para lanzar desde la shell:

```
#Creando una copia de seguridad de la base de datos
mysqldump -p BDphp > backup.sql

#Recuperar la copia de seguridad anteriormente realizada
mysql -u root BDphp -p < backup.sql

#Consultas directas:
mysql -e "select * from TPersona" BDphp -p

#Mostrar las Bases de datos de MySQL:
mysqlshow

#Mostrar tablas de una BD:
mysqlshow BDphp

#Mostrar el contenido de una determinada tabla:
mysqlshow BDphp TPersona
```

Ya tenemos la tabla BDphp creada en la Base de Datos para nuestras pruebas. Mediante PHP accederemos a ella para lanzar consultas del tipo insertar datos, borrar, listar...

## Programando con PHP

Una vez llegados hasta aquí podemos empezar a usar PHP. Para poder llegar a manejar completamente este lenguaje, hay bastantes manuales por Internet como el siguiente:

<http://www.php.net/manual/es/>

el cuál es muy completo. No voy a explicar todo el tema de variables, estructuras, funciones... porque vienen perfectamente explicadas en el tutorial anteriormente citado y sería una pérdida de tiempo.

Dicho esto, vamos a hacer nuestra primera página PHP. El código que queramos que sea interpretado por el servidor, lo pondremos entre las marcas `<php y ?>` o `<? y ?>`, para que sepa diferenciarlo de las restantes etiquetas HTML. Vamos a poner un ejemplo, que lo guardaremos como `ej.php` en la raíz del servidor web, es decir en `/var/www`:

```
<html>
<body>
<?php
    phpinfo();
?>
</body>
</html>
```

Ahora solo bastaría con arrancar `apache /etc/init.d/apache start` si no lo estaba ya y con nuestro navegador, ir a `http://localhost/ej.php`.

Nos debe de salir una página bastante chula con toda la información sobre PHP en la máquina.

Vamos a dejarnos de rodeos y conectar con MySQL y más concretamente, con la BD anteriormente creada. Bastaría con poner lo siguiente:

```

// Conectamos con MySQL llamando a la función mysql_connecta y pasándola como
// parámetros la máquina donde está el servidor, el usuario para conectar y su pass.
$connect = mysql_connect("localhost", "phpuser","phpuser");
// Como hemos dicho anteriormente, MySQL necesita saber con qué BD vamos a trabajar
// y vamos a lanzar las consultas. Estos lo indicamos de la siguiente forma:
mysql_select_db("BDphp", $connect);
// Lanzamos ya la primera consulta, que nos devolverá un cierto número de filas
// que dejará en la variable $resultado
$resultado=mysql_query("select * from TPersona", $connect);

```

Una vez contenida en la variable \$resultado, el resultado de la consulta SQL contra la tabla de la BD seleccionada, tenemos varias maneras de "coger" los datos de ella. Una de las más sencillas es la siguiente; mediante un bucle while, vamos cogiendo todas las filas hasta que no haya más y se acabe el bucle. Directamente, podemos ordenar los datos creando una tabla HTML. Esta es su estructura:

```

// Mediante echos, mandamos el código html al cliente
echo "<table border = '1'> \n";
echo "<tr><td>DNI</td><td>Nombre</td><td>Apellidos</td><td>Mail</td></tr>";

// Mientras sigamos obteniendo filas de la variable $resultado...
while ($fila = mysql_fetch_row($resultado))
{
    // Mandamos una fila al cliente sustituyendo cada celda por el contenido
    // de la fila obtenida de la variable $resultado.
    // Los valores se consiguen de la forma $variable[n], donde n es el nº de
    // columna de la Tabla o puede ser tb de la forma $variable["Nombre_Columna"]
    echo "<tr>";
    echo "<td>".$fila[0]."</td>";
    echo "<td>".$fila[1]."</td>";
    echo "<td>".$fila[2]."</td>";
    echo "<td>".$fila[3]."</td>";
    echo "</tr>";
}
echo "</table>";

// Finalmente cerramos la conexión
mysql_close($connect);

```

De esta forma, ya tenemos una tabla con todos los valores de TPersona.

A partir de aquí, se trabaja todo el rato de forma parecida. Es decir, todas las consultas se realizan de forma exáctamente igual a la Select del ejemplo, solamente que si por ejemplo, hacemos una "insert", no se tendrá resultado alguno del mysql\_query. Comentar que hay muchas más funciones para el manejo de BDs pero simplemente, no me iba a poner a explicarlas todas. Éstas son las que más uso y por lo tanto, las que he comentado.

## Conclusión

Como habeis podido comprobar, esto ha sido una ínfima parte de todo lo referente a PHP y MySQL. Por tanto, para adentrarse más en todo esto, recomiendo el link que he citado antes : <http://www.php.net/manual/es/>.

---

Este documento ha sido escrito por un miembro de e-GHOST, y su contenido es libre de ser reproducido en otros medios bajo las condiciones de la Licencia FDL (GNU Free Documentation License).