

41311 Laboratorio de Interfaces de Redes

Práctica 1 - Instalación de las aplicaciones

1. Objetivos

El objetivo de esta práctica es aprender a instalar y configurar un servidor con servicio Web, en concreto el de Apache. También instalaremos un servidor de base de datos MySQL, un módulo de PHP para Apache con soporte para MySQL y una páginas para la gestión del servidor MySQL por la web.

2. Introducción

En el Laboratorio de Telemática 1 emplearemos los ordenadores tlm14-tlm16, tlm24-tlm26, tlm34-tlm36, tlm44-tlm46, tlm54-tlm56, tlm64-tlm66, tlm74 y tlm75, es decir, cualquiera de las filas del lado izquierdo según se entra en el laboratorio (segunda planta del edificio de los pinos). Si bien las cuentas de la asignatura (lirxy) funcionan también en los ordenadores de la otra mitad del laboratorio nos restringiremos a los mencionados para una mejor utilización de los recursos del laboratorio. En el Laboratorio de Telemática 2 (primera planta del mismo edificio) emplearemos cualquier máquina del mismo.

Los programas que vamos a instalar en esta práctica son:

- Apache 2.0: El fichero a instalar es /opt3/lir/ficheros/httpd-2.0.51.tar.gz
- PHP 4.3.9: Fichero /opt3/lir/ficheros/php-4.3.9.tar.gz
- MySQL 4.0.21: Fichero /opt3/lir/ficheros/mysql-4.0.21.tar.gz
- phpMyAdmin 2.5.3: Fichero /opt3/lir/ficheros/phpMyAdmin-2.5.7-p11.tar.gz

Generalmente existen versiones pre-compiladas de estos programas para diferentes sistemas operativos y microprocesadores y una solución fácil y rápida es descargarse la adecuada e instalarla mediante el sistema que disponga el sistema operativo (.rpm, .pkg, .deb, etc). Sin embargo en esta práctica vamos a compilar los programas a partir de sus códigos fuente porque esto nos ofrece una mayor flexibilidad a la hora de escoger qué incluir en ellos de las numerosas opciones que ofrecen.

Para ahorrar tiempo hemos descargado ya unas de las últimas versiones de estos programas y se han dejado disponibles en los directorios mencionados. En cualquier caso, lo único que hay que hacer para obtenerlas es dirigirse a sus sitios web y descargarlas dado que son todas software abierto (<http://www.apache.org>, <http://www.php.net>, <http://www.mysql.com>, <http://www.phpmyadmin.net/>). Para que no haya problemas para seguir las instrucciones presentes en los guiones de prácticas por favor empleen los fichero ya descargados.

Lo primero que hay que hacer es descomprimir los ficheros en el directorio /tmp local de nuestra máquina (aunque vayamos a instalar en nuestra cuenta). Para ello pueden utilizar gzip y tar o simplemente tar. Si no sabe emplear estas herramientas consulte las correspondientes páginas del manual en línea. Por ejemplo podemos hacer:

```
[1] %> cp /opt3/lir/ficheros/httpd-2.0.51.tar.gz /tmp  
[2] %> cd /tmp  
[3] %> gzip -d httpd-2.0.51.tar.gz
```

Para descomprimir (opción -d) el fichero httpd-2.0.51.tar.gz, con lo que nos quedará el fichero httpd-2.0.51.tar y ahora para sacar los ficheros del tarball basta con hacer:

```
[4] %> tar xvf httpd-2.0.51.tar
```

Donde la opción x significa *extraer*, la opción f es porque le vamos a dar un fichero de entrada y la opción v hace que nos vaya diciendo cómo evoluciona el programa extrayendo los ficheros.

Con una versión suficientemente moderna del programa tar se pueden hacer ambos pasos en uno añadiendo la opción z al tar, es decir, podría haberse ejecutado:

```
[3] %> tar xzvf httpd-2.0.51.tar.gz
```

Llegado este punto tendremos un directorio llamado httpd-2.0.51 donde se encuentran las fuentes del programa Apache. Parte de las explicaciones sobre cómo instalar Apache están extraídas de la documentación del mismo (versión 2.0) que se puede encontrar en <http://httpd.apache.org/docs-2.0/>.

3. Instalación de Apache 2.0

A. Sobre los módulos

Apache es un servidor modular. Esto quiere decir que acepta añadirle funcionalidades a través de módulos sin necesidad de recompilar el servidor. Solo las funcionalidades básicas suelen estar incluidas en el programa del servidor. Durante la compilación del servidor se puede escoger qué módulos queremos que se compilen para emplear con el mismo. Además, estos módulos podrían incluirse dentro del propio ejecutable del servidor o, si el sistema operativo soporta cargar módulos de forma dinámica (módulos DSO), pueden ser compilados como ficheros separados .so para ser cargados cuando se requieran. Podemos especificar qué módulos cargar dinámicamente de los compilados mediante directrices en el fichero de configuración del Apache. Un módulo que siempre estará incluido en Apache será el mod_so que es precisamente el que se encarga de la carga de módulos de forma dinámica (más información en <http://httpd.apache.org/docs-2.0/en/dso.html>)

Podemos compilar los módulos bien cuando compilamos el programa `httpd` (el nombre del ejecutable del Apache) o los podemos compilar posteriormente uno a uno cuando decidamos añadirlos. En esta práctica vamos a compilar el servidor Apache con unos módulos básicos y con soporte para posteriormente cargar otros (es decir, con el módulo `mod_so`).

B. Configurar

Primero debemos seleccionar las opciones en concreto que queremos que se apliquen cuando compilemos el servidor. En esta etapa es cuando especificaremos los módulos a compilar. Esto se hace mediante el script `configure` que se encuentra en el directorio `httpd-2.0.47`. Este script acepta un gran número de opciones según cómo queramos compilar el Apache. Podemos ver estas opciones ejecutando:

```
[4]">%> cd httpd-2.0.51
[5]">%> ./configure --help
```

Echen una ojeada a las opciones disponibles. Puede que llegado al momento alguna nos sea útil.

Vamos a emplear ahora solo dos de ellas:

- `--enable-so` para incluir el módulo que permite cargar otros de forma dinámica
- `--prefix` para especificar dónde queremos que se instale el software del servidor

Así pues, podemos ejecutar algo como:

```
[6]">%> ./configure --enable-so --prefix="/opt3/lir/lirXY/www"
```

Donde `lirXY` debe sustituirse por su número de grupo de forma que especificamos un directorio de instalación dentro del HOME del grupo de prácticas. No hace falta que creamos el directorio `www`, el proceso de instalación lo creará por nosotros.

Cuestiones:

- Si no especificáramos el directorio donde debe instalarse, en dónde se instalaría Apache?
Puede usted instalar en ese directorio? Por qué?

C. Compilar

El script `configure`, si se ha completado con éxito, habrá preparado los ficheros `Makefile` necesarios para compilar el programa con las opciones que le hayamos indicado. Ahora solo debemos hacer:

```
[7]">%> make
```

Y esperar... (aproveche el tiempo para seguir leyendo)

D. Instalar

Solo queda instalar el software compilado, y no puede ser más simple que hacer:

```
[8]">%> make install
```

El resultado será una estructura de directorios bajo `/opt3/lir/lirXY/www` con el servidor Apache y otros programas y scripts asociados. Debería tener un aspecto como:

```
[9]">%> ls /opt3/lir/lirXY/www
bin    cgi-bin  error   icons   lib     man      modules
build  conf     htdocs  include  logs   manual
```

Y con esto ya hemos compilado el servidor de Web Apache.

Podemos comprobar los módulos que se han incluido en el ejecutable lanzándolo con la opción `-l`. El ejecutable `httpd` se encuentra en `www/bin`

```
[10]">%> /opt3/lir/lirXY/www/bin/httpd -l
Compiled-in modules:
 core.c
 mod_access.c
 mod_auth.c
 ...
```

A continuación borren el directorio con las fuentes del Apache (ocupa mucho espacio en disco) y el `.tar.gz`.

4. Instalación de MySQL 4.0

Vamos a ver que la instalación de MySQL sigue unos pasos muy similares y tampoco entraña ninguna complicación (http://www.mysql.com/doc/en/Installing_source.html).

Descomprima el fichero con las fuentes de MySQL de forma análoga a como descomprimió el de Apache. Con eso debería tener un directorio `mysql-4.0.21` con las fuentes.

A. Configurar

MySQL dispone también de un script `configure`. Primero empleamos la opción `--help` para familiarizarnos con las opciones disponibles.

Vamos a emplear una configuración sin embargo muy sencilla:

```
[11] $> cd /tmp/mysql-4.0.21
[12] $> ./configure --prefix=/opt3/lir/lirXY/mysql
```

Donde simplemente indicamos el directorio donde queremos que se instale el software y el resto de las opciones son las que estén por defecto.

B. Compilar

No tiene más complicación que hacer:

```
[13] $> make
```

Aprovechen para leer el resto del guión mientras compila.

C. Instalar

Finalmente:

```
[14] $> make install
```

El directorio /lir/mysql contendrá algo como:

```
[15] $> ls /opt3/lir/lirXY/mysql
bin include info lib libexec man mysql-test share sql-bench
```

A continuación borren el directorio con las fuentes de MySQL (ocupa mucho espacio en disco) y el .tar.gz.

5. Instalacion de PHP 4.3

Vamos a compilar un módulo de PHP para Apache, con soporte para conectarse a un servidor de base de datos MySQL. Podemos encontrar la documentacion sobre PHP en <http://www.php.net/docs.php>.

A. Configurar

De nuevo nos encontramos con un configure. Las opciones que vamos a utilizar son las siguientes:

- `--prefix="directorio donde instalar"`
- `--with-apxs2="path al ejecutable apxs"` Aquí debemos indicar el directorio donde está el programa apxs2. Añste es un programa que instala Apache en su directorio bin. El programa sirve para compilar ficheros fuente en C con las opciones correctas para que el resultado sea un modulo que Apache pueda emplear. Así que PHP necesita saber dónde está este programa para compilar el modulo de PHP que va a crear para Apache. Dado que estamos compilando para Apache 2.x empleamos `--with-apxs2` (para más información en <http://httpd.apache.org/docs-2.0/programs/apxs.html>)
- `--with-mysql="directorio base de mysql"` Aquí le vamos a indicar dónde tenemos instalado MySQL para que utilice la librería que hemos compilado con anterioridad.

En nuestro caso vamos a ejecutar:

```
[16] $> cd /tmp/php-4.3.9
[17] $> ./configure --prefix=/opt3/lir/lirXY/php --with-apxs2=/opt3/lir/lirXY/www/bin/apxs --with-mysql=/opt3/lir/lirXY/mysql
```

Cuestiones:

Si quisiéramos compilar el módulo para Apache 1.x qué cambiaría en las opciones para el configure?

B. Compilar

Ahora solo tenemos que compilar:

```
[18] $> make
```

C. Instalar

Y finalmente instalar:

```
[19] $> make install
```

De nuevo borre las fuentes para ahorrar espacio

6. Probando el Apache

A continuación lo primero que vamos a hacer es comprobar que la instalación de Apache funciona correctamente. Para ello vamos a lanzarlo y solicitarle alguna página HTML de prueba.

Si lanzáramos el Apache ahora mismo éste emplearía la configuración por defecto que incluye emplear el puerto 80 para ejercer sus funciones de servidor Web en el puerto reservado a tal efecto. En máquinas con sistema operativo de tipo UNIX, el rango de puertos bien conocidos (well-known ports) está controlado, de forma que sólo el administrador (root) puede lanzar programas que actúen como servidor sobre uno de esos puertos. Así que si deseásemos lanzar Apache sobre el puerto 80 necesitaríamos privilegios de superusuario.

Dado que no disponemos de privilegios de superusuario lo que vamos a hacer es configurar el Apache para que emplee otro puerto como servidor, uno fuera del rango de reservados y que no se esté empleando. Para ello tenemos que realizar una simple modificación en el fichero de configuración del Apache.

Cuestiones:

Busque alguna opción de compilación de Apache que haga que emplee otro puerto

A. httpd.conf

A lo largo de las diferentes versiones de Apache los ficheros de configuración han cambiado un poco su ubicación y la forma de repartir las directrices de configuración entre un fichero o varios. Se pueden crear varios ficheros de configuración para Apache y luego incluirlos en el principal (como con un #include de C). Incluso podemos especificar el fichero de configuración que debe emplear cuando lo lanzamos (con la opción -f). Por simplicidad vamos a emplear un solo fichero de configuración, el que emplea por defecto, llamado `httpd.conf` y que el programa busca al arrancar en el directorio `/opt3/lir/lirXY/www/conf/`

i) Sintaxis del fichero de configuración

El fichero de configuración es texto simple. Contiene directrices que entiende Apache o alguno de los módulos que tenga instalados. Se emplea una directriz por línea y podemos incluir líneas de comentario comenzándolas con el carácter "#".

Se puede comprobar si se ha escrito el fichero sin errores de sintaxis ejecutando `apachectl configtest`. Este programa se encuentra en el directorio `/opt3/lir/lirXY/www/bin/`

ii) Directriz Listen

Con la directriz `Listen` podemos especificar el puerto que va a emplear el servidor. A partir de Apache 2.0 ésta es una directriz obligatoria. Lo único que hemos de hacer es buscarla en el fichero de configuración y cambiar el valor de puerto por otro fuera del rango de reservados. Por ejemplo lo podemos cambiar por:

`Listen 3456`

Cuestiones:

Cuál es el valor máximo de puerto que se podría especificar?

Podríamos tener dos programas `httpd` ejecutándose al mismo tiempo en la misma máquina (la cual tiene una sola dirección IP)? Cómo?

B. Lanzando Apache

Podemos lanzar el servidor Apache simplemente ejecutando el programa `httpd`. Sin embargo, es mucho más cómodo emplear un script preparado para lanzarlo correctamente. Este script es el `apachectl` que se encuentra en el directorio `/opt3/lir/lirXY/www/bin/`. El script lanzará el servidor, el cual crea varios procesos para atender a las peticiones de los clientes.

Lanzamos Apache:

```
[20] %> /opt3/lir/lirXY/www/bin/apachectl start
```

Podemos ver los procesos que quedan corriendo en la máquina como demonios:

```
[21] %> ps ax | grep httpd
```

```
29635 ? S 0:00 /opt3/lir/lirXY/www/bin/httpd -k start
29636 ? S 0:00 /opt3/lir/lirXY/www/bin/httpd -k start
29637 ? S 0:00 /opt3/lir/lirXY/www/bin/httpd -k start
29638 ? S 0:00 /opt3/lir/lirXY/www/bin/httpd -k start
29639 ? S 0:00 /opt3/lir/lirXY/www/bin/httpd -k start
29640 ? S 0:00 /opt3/lir/lirXY/www/bin/httpd -k start
29641 ? S 0:00 /opt3/lir/lirXY/www/bin/httpd -k start
```

Donde los primeros procesos son el servidor web que ya tenía corriendo la máquina y los restantes son el que acabamos de lanzar.

C. Probando

Ahora lo si solicitamos una página web a la máquina en la que nos encontramos debería mostrarnos la página por defecto que trae el servidor. Para ello escriba en un navegador la siguiente dirección: <http://localhost:3456>

Cuestiones:

Pruebe a solicitar la página por defecto desde otra máquina del laboratorio. Recuerde que `localhost` hace referencia a la máquina local en la que se está ejecutando el programa, en este caso el navegador, así que no puede emplear el URI anterior.

Si tiene configurado un proxy en el navegador Web le recomiendo que le indique al navegador que no use el proxy para las conexiones con máquinas dentro del dominio `net.tlm.unavarra.es`, para ello, en la sección de preferencias avanzadas de un Netscape o Mozilla, en la subsección de Proxies añada a "No Proxy for" algo como `.net.tlm.unavarra.es` para que cuando intente acceder a una máquina en una máquina cuyo nombre de dominio termine así no le haga la solicitud al proxy. Recuerde que los nombre de las máquinas del laboratorio siguen la forma `t1mXY.net.tlm.unavarra.es`

7. Probando el módulo de PHP

A continuación vamos a comprobar que hemos añadido correctamente el módulo de PHP a Apache.

A. Configuración necesaria para Apache

Hay dos directrices necesarias en el fichero de configuración de Apache `httpd.conf` para que emplee el módulo PHP y reconozca las páginas PHP.

En primer lugar debemos decirle que cargue el módulo de PHP. Cargar módulos se realiza con la directriz `LoadModule`. Esta directriz es procesada por el módulo `mod_so`. Su sintaxis es la siguiente:

```
LoadModule nombre_del_modulo fichero_del_modulo
```

En nuestro caso debemos tener la siguiente línea en el fichero de configuración:

```
LoadModule php4_module modules/libphp4.so
```

Como podemos ver el último argumento incluye el path para llegar al fichero desde el directorio donde se instaló el Apache.

Lo segundo que debemos modificar en la configuración del Apache es indicarle cómo reconocer los ficheros que contienen código PHP. Lo normal para esto es emplear la extensión del fichero de forma que cuando el servidor vaya a servir un fichero con esa extensión lo reconozca como un fichero con código PHP y lo procese a través del módulo de PHP. La extensión típica es `.php` aunque ha habido otras como `.php3`. La forma de configurar esto es indicándole al servidor un nuevo tipo MIME asociado a la extensión que hayamos escogido. El tipo MIME para los ficheros PHP debe ser: `application/x-httpd-php` y la forma que tenemos de declararlos en el fichero de configuración es mediante la directriz `AddType`.

La sintaxis de `AddType` es:

```
AddType MIME_type extension
```

Donde podemos poner varias extensiones que asociar al mismo tipo MIME. La línea que debemos incluir será simplemente:

```
AddType application/x-httpd-php .php
```

B. Página PHP de prueba

Creamos un fichero llamado `prueba.php` en el directorio `/opt3/lir/lirXY/www/htdocs` con el siguiente código:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body>
    Si despues de esta linea no ves nada es que PHP no funciona<br>
    <?php
      echo "Vaya, pues funciona!";
    ?>
  </body>
</html>
```

Y solicitamos esa página desde un navegador poniendo el siguiente URI: <http://localhost:3456/prueba.php>
El resultado debe ser el siguiente, si no tenemos algún error:

```
Si despues de esta linea no ves nada es que PHP no funciona
Vaya, pues funciona!
```

8. Probando la instalación de MySQL

Ahora vamos a comprobar que hemos instalado correctamente MySQL y para ello vamos a arrancar el servidor de base de datos que incluye.

A. Pasos previos

Antes de lanzar por primera vez el servidor de base de datos debemos ejecutar un script que termina la configuración necesaria. Este script creará la base de datos en la que se guardan los permisos de acceso que damos a las diferentes bases de datos que se creen, así como información de los usuarios. También se crea una base de datos para pruebas y un usuario inicial (`root`). Con la opción `--datadir` le podemos decir dónde queremos tener las bases de datos (las ponemos en nuestro directorio) y con la opción `--socket` indicamos dónde se creará el socket UNIX para la comunicación local con la base de datos (para cuando no se hace por la red sino en la misma máquina), que también colocamos en el directorio del usuario.

```
[22] $> cd /opt3/lir/lirXY/mysql/bin
[23] $> ./mysql_install_db --datadir=/opt3/lir/lirXY/mysql/var --socket=/opt3/lir/lirXY/socket.sock
```

El programa servidor de la base de datos se llama `mysqld` (la 'd' viene de 'daemon' o 'demonio', igual que en 'httpd'). Hay scripts para lanzarlo pero en estas prácticas vamos a comenzar lanzándolo directamente con el ejecutable:

```
[24] $> ../libexec/mysqld --socket=/opt3/lir/lirXY/mysql/socket.sock --datadir=/opt3/lir/lirXY/mysql/var &
```

Ahora podemos comprobar que el servidor está corriendo correctamente empleando el programa `mysqladmin` que permite realizar algunas tareas de gestión.

```
[25]">$> ./mysql/bin/mysqladmin --socket=/opt3/lir/lirXY/mysql/socket.sock version
```

Deberíamos obtener información sobre el servidor. Como por ejemplo:

```
bin/mysqladmin Ver 8.40 Distrib 4.0.21, for pc-linux on i686
Copyright (C) 2000 MySQL AB & MySQL Finland AB & TCX DataKonsult AB
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL license
```

```
Server version      4.0.21
Protocol version   10
Connection          Localhost via UNIX socket
UNIX socket        /opt3/lir/lirXY/mysql/socket.sock
Uptime:             41 sec
```

```
Threads: 1 Questions: 1 Slow queries: 0 Opens: 6 Flush tables: 1 Open tables: 0 Queries per second avg: 0.024
```

El servidor de la base de datos controla los accesos a las mismas mediante un sistema de usuarios y privilegios. Inicialmente existe un usuario llamado `root` que tiene todos los privilegios posibles y carece de password. No se deben confundir los usuarios de la base de datos con los usuarios de una máquina Unix. Ambos conjuntos de usuarios son independientes. Se emplea el nombre `root` para ese superusuario por la tradición existente en máquinas Unix pero el usuario `root` de nuestro servidor de base de datos es independiente del usuario `root` de la máquina.

Podemos cambiar la password de este usuario empleando el siguiente comando:

```
[26]">$> ./mysqladmin --socket=/opt3/lir/lirXY/mysql/socket.sock -u root password nuevapassword
```

Cuestiones:

Averigüe cómo detener el servidor de base de datos con la opción `shutdown` del comando `mysqladmin`

La instalación de MySQL ofrece un cliente para realizar comandos sobre la base de datos. Este cliente es un programa para la shell llamado `mysql`:

```
[27]">$> ./mysql --socket=/opt3/lir/lirXY/mysql/socket.sock -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8 to server version: 4.0.21

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Como se puede ver empleamos la opción `-u` para especificar el usuario con el que nos queremos conectar a la base de datos y con la opción `-p` hacemos que nos solicite la password del usuario.

Para salir del programa basta con emplear el comando `QUIT`:

```
mysql> quit
Bye
```

Vamos a comprobar la correcta instalación realizando algunas consultas básicas al servidor.

Primero solicitamos una lista de las bases de datos existentes. Si estamos empleando el usuario `root` tendremos acceso a ver todas las que hay que al instalar son:

```
[28]">$> ./mysql --socket=/opt3/lir/lirXY/mysql/socket.sock -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 13 to server version: 4.0.21

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| mysql    |
| test     |
+-----+
2 rows in set (0.00 sec)
```

```
mysql> quit
Bye
```

Pero lo normal es que por defecto el servidor nos deje conectarnos empleando el usuario anónimo (`anonymous`) solo que en ese caso no tendremos acceso a todas las bases de datos:

```
[29]">$> ./mysql --socket=/opt3/lir/lirXY/mysql/socket.sock
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 14 to server version: 4.0.21
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| test     |
+-----+
```

```
+-----+
1 row in set (0.00 sec)
```

```
mysql> quit
Bye
```

Las bases de datos `mysql` y `test` fueron creadas por el script `mysql_install_db`. La primera contiene la información referente a los usuarios y sus privilegios de acceso al servidor y por eso solo es visible para el superusuari. La segunda (`test`) es una base de datos creada para realizar pruebas.

Ahora vamos a conectarnos al servidor y a crear una nueva tabla dentro de la base de datos de pruebas:

```
[30] $> ./mysql --socket=/opt3/lir/lirXY/mysql/socket.sock
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 18 to server version: 4.0.21

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use test;
Database changed
mysql> create table precios (nombre varchar(50), valor double(6,2));
Query OK, 0 rows affected (0.00 sec)

mysql> describe precios;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| nombre | varchar(50) | YES  |     | NULL    |       |
| valor  | double(6,2)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> insert into precios values ("El taxi", 10000);
Query OK, 1 row affected (0.00 sec)

mysql> insert into precios values ("Un duro", .03);
Query OK, 1 row affected (0.01 sec)

mysql> select * from precios;
+-----+-----+
| nombre | valor   |
+-----+-----+
| El taxi | 10000.00 |
| Un duro |      0.03 |
+-----+-----+
2 rows in set (0.00 sec)

mysql> quit
Bye
```

Con el comando `use` hemos indicado la base de datos con la que queremos trabajar. El comando `create` sirve para crear una nueva tabla en la base de datos, en este caso el nombre de la tabla es `precios` y tiene dos columnas: una cadena de 50 caracteres de nombre `nombre` y un número flotante con presentación de 6 cifras enteras y 2 decimales con nombre `valor`. Con el comando `describe` hemos podido ver las columnas de la base de datos que coinciden con lo deseado. A continuación hemos insertado (comando `insert` por si alguien se lo preguntaba) dos nuevas filas en la tabla: el precio del taxi y el precio de un duro. Finalmente hemos realizado una petición al servidor con `SELECT` para que nos mostrase todas las entradas de la tabla `precios` recién creada.

Finalmente, vamos a crear un usuario al que se le permitirá acceder a las tablas de esa base de datos desde cualquier máquina en la red (en este punto vamos a ser bastante permisivos, en prácticas futuras pueden estudiar los privilegios de usuarios y restringir un poco el acceso si lo desean). También vamos a darle permisos al usuario `root` del servidor de base datos para conectarse mediante TCP siempre que sea desde la máquina local:

```
[28] $> ./mysql --socket=/opt3/lir/lirXY/mysql/socket.sock -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 13 to server version: 4.0.21

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> grant all on test.* to lir@'%' identified by 'laclave';
Query OK, 0 rows affected (0.01 sec)

mysql> grant all on *.* to root@'localhost.localdomain' identified by 'laclavepararoot';
Query OK, 0 rows affected (0.01 sec)

mysql> quit
Bye
```

9. Probando el soporte de MySQL desde PHP

Vamos a comprobar que somos capaces de acceder a la base de datos desde scripts PHP. Para ello crearemos un simple script que se conecte a la base de datos y solicite el contenido de la tabla `precios` de forma similar a como hicimos anteriormente con el cliente `mysql` solo que ahora mostrando nosotros el contenido con una tabla HTML.

Creamos un fichero llamado `pruebasql.php` en el directorio `/lir/www/htdocs/` con el siguiente código:

```

<html>
<html>
<head>
<title>Probando acceso a MySQL</title>
</head>
<body>
Si despues de esta linea no ves nada es que PHP no funciona<br>
<?php
/* Conectarse empleando un usuario en concreto, como el usuario no existe en
la base de datos estamos empleando el usuario anonimo. */
$link = mysql_connect("127.0.0.1", "lir", "laclave")
or die("Fallo al conectar con la base de datos");
print "Conectado<br>";
/* Seleccionar la base de datos a emplear */
mysql_select_db("test") or die("Error al seleccionar la base de datos");
print "Base de datos <tt>test</tt> seleccionada<br>";

/* Realizar una busqueda */
$query = "SELECT * FROM precios";
$result = mysql_query($query) or die("Query failed");

/* Sacar el contenido como una tabla HTML */
print "<table border=1>\n";
while ($line = mysql_fetch_array($result, MYSQL_ASSOC))
{
print "\t<tr>\n";
foreach ($line as $col_value)
{
print "\t\t<td>$col_value</td>\n";
}
print "\t</tr>\n";
}
print "</table>\n";
?>
</body>
</html>

```

Y solicitamos esa página desde un navegador poniendo el siguiente URI: <http://localhost:3456/pruebasql.php>
El resultado debe ser el siguiente, si no tenemos algún error:

Si despues de esta linea no ves nada es que PHP no funciona

Conectado

Base de datos test seleccionada

El taxi	10000.00
Un duro	0.03

Checkpoint

Muestre al responsable de prácticas que todo le funciona correctamente

10. Instalando phpMyAdmin

Ahora instalaremos *phpMyAdmin* que es un conjunto de páginas HTML (en realidad scripts PHP) que permiten gestionar una base de datos MySQL directamente por la web.

Si bien generalmente aprenderemos a realizar la gestión de nuestro servidor de base de datos a través del cliente *mysql* una vez que tengamos los conceptos claros resultará más cómodo realizar algunas tareas con un interfaz como este.

A. Instalando

Lo primero es descomprimir el tarball:

```
[31] %> cd ~
[32] %> tar xzvf /opt3/lir/ficheros/phpMyAdmin-2.5.7-p11.tar.gz
```

A continuación moveremos el directorio al interior del directorio principal desde donde nuestro servidor Web sirve páginas (lo que veremos más adelante que se llama el *DocumentRoot*) y de paso le cambiamos el nombre a algo más cómodo:

```
[33] %> mv phpMyAdmin-2.5.7-p11 /opt3/lir/lirXY/www/htdocs/sqladmin
```

B. Configurando

Y ahora editamos el fichero de configuración de *phpMyAdmin* para que funcione con nuestro servidor. Dicho fichero es */opt3/lir/lirXY/www/htdocs/sqladmin/config.inc.php*. Debemos modificar al menos tres líneas. Primero en la línea que pone:

```
$cfg['PmaAbsoluteUri'] = '';
```

debemos introducir el URI de esta instalación de *phpMyAdmin*. En nuestro caso:

```
$cfg['PmaAbsoluteUri'] = 'http://localhost:3456/sqladmin';
```

También tenemos que especificar el usuario con el que se conectarán estas páginas web a la base de datos. Como queremos realizar gestión de toda la base de datos tenemos que cambiar:

```
$cfg['Servers'][$i]['user']      = 'root';      // MySQL user  
$cfg['Servers'][$i]['password']  = '';
```

por:

```
$cfg['Servers'][$i]['user']      = 'root';      // MySQL user  
$cfg['Servers'][$i]['password']  = 'lapasswordquelehayamospuesto';      // MySQL password (only needed)
```

Y finalmente cambiamos el modo de acceso para que se acceda mediante TCP en vez del socket UNIX. Para ello cambien:

```
$cfg['Servers'][$i]['host']      = 'localhost';      // MySQL hostname or IP address
```

por

```
$cfg['Servers'][$i]['host']      = '127.0.0.1';      // MySQL hostname or IP address
```

C. Probando

Ahora podemos probar estas páginas accediendo al índice de las mismas <http://localhost:3456/sqladmin/index.php>

Cuestiones:

Pruebe a realizar la misma consulta que hizo antes desde mysql y desde la página php de prueba pero utilizando este interfaz.
Indague un poco sobre cómo crear bases de datos, tablas, insertar datos, etc. con este interfaz.

11. Resumen y conclusiones

En esta práctica hemos instalado un servidor Apache con soporte de PHP, un servidor de bases de datos MySQL y hemos comprobado el correcto funcionamiento de todo.

La instalación ha sido a partir del código fuente de cada uno de los programas y se ha realizado en directorio home del usuario.

Se han utilizado ejemplos simples de HTML, PHP y SQL a sabiendas de que se profundizará en el uso de todos ellos en las siguientes prácticas