XAMP: PHP y MySQL

Emmanuel Ayala Mexicano

¿Qué es LAMP?

X	A	M	P
Sistema Operativo	Servidor Web	Administrador de Bases de Datos	Lenguaje de Programación
Windows, Mac, Solaris, Linux (Mayores distribuciones Red Hat | Debian)	Apache, Nginx, LightTPD, Mongrel, Thin, Webrick.	MySQL, Postgre, Firebird, Oracle, Sybase, SQLite3, XML.	Perl, Pyhon, PHP

Funciones generales.

X	A	M	P
Sistema Operativo	Servidor Web	Administrador de Bases de Datos	Lenguaje de Programación
Administrar usuarios, permisos en archivos, carpetas, hacer respaldos, instalar programas, monitorear recursos (RAM, DD, Procesador).	Enviar archivos (htdocs) a los usuarios, Cada carpeta corresponde a un proyecto. Tener hosts virtuales (múltiples dominios en un servidor, administración de subdominios).	Administrar usuarios, permisos en las BD. Almacenar información. Hacer respaldos de la información.	Conectar la GUI (usualmente visible en un navegador web) con la BD. Realizar tareas para procesar datos, archivos.

Archivos

X	A	M	P
Sistema Operativo	Servidor Web	Administrador de Bases de Datos	Lenguaje de Programación
Archivos de configuración de (A, M, P), configuración / instalación de software para la administración de el servidor (FTP, SSH, CRON), interfaz web, correo electrónico.	HTML + CSS + JS + Media (imagen, audio, video). * LP embedido (archivos PHP en este caso).	BD	Código para realizar funciones específicas (generalmente se encuentra embedido en archivos dentro de la carpeta del servidor web)*.

Servidor Web:
Abrir documento.html, .php, .txt en el navegador web directamente.
Pasar la carpeta (pasar-a-htdocs) a htdocs y abrir los archivos mediante el servidor web.

Faltante

Otra herramienta importante que no se menciona dentro del conjunto XAMPP es el correo electrónico.

Éste puede ser procesado mediante postfix / sendmail (incluso se puede tener una interfaz web) y, si se requiere descargar el correo a una computadora se puede utilizar dovecot.

Instalación en conjunto

Descargar algún paquete XAMPP (WAMP, MAMP, LAMP, XAMPP).

http://www.apachefriends.org/es/xampp.html

Instalación individual

OS X / Win

http://www.mysql.com http://www.php.net

Unix - OS X: http://httpd.apache.org/download.cgi

Windows: http://httpd.apache.org/docs/current/platform/windows.html

Debian - Manejador de Paquetes - apt / dpkg

apt-get install apache2 php5 libapache2-mod-php5 php5-gd php5-dom php5-mysql

Red Hat - Manejador de Paquetes - yum

yum install php httpd php-gd php-mysql

Diferencias entre las instalaciones.

La diferencia radica en el desempeño que van a tener los equipos. Si se instalan los programas de manera individual se pueden configurar en base a la arquitectura del servidor (i386 / x_64), los módulos a utilizar, y, por último, qué programas instalar o utilizar (diferente BD, diferente servidor web). Cabe mencionar que pueden funcionar todos dentro de un mismo equipo o pueden estar en diferentes equipos.

La ventaja de instalar todo como un paquete es tiempo a la hora de instalar y configurar (viene con opciones predeterminadas comunes que, aunque no sean las óptimas brindan buen desempeño).

La ventaja de utilizar todo en un mismo equipo es la configuración y administración (suele ser mucho más sencilla).

Consideraciones

Dependiendo de la instalación de los programas hay que tener en cuenta las siguientes consideraciones.

Revisar puertos.

En algunas instalaciones como MAMP para OS X se utilizan puertos distintos a los ‘bien conocidos’ para el servidor web y para el DBMS.

Revisar contraseña de MySQL.

Durante el proceso de instalación de algunas versiones (apt en linux) se asigna una contraseña para el usuario root de mysql, en otras viene sin contraseña (XAMPP).

Revisar puertos

http://localhost/

http://localhost:8888/

Revisar contraseña de MySQL.

Dependiendo de la instalación de XAMPP MySQL puede tener habilitada la contraseña de root o no.

usuario:contraseña

root:pwd_asignado

root:

Localizar dentro de la instalación de LAMP

htdocs * ⇒ Proyectos.

mysql * ⇒ data.

* Varía su ubicación dentro del sistema operativo dependiendo de la instalación / configuración.

Rutas relativas

Hay que tener como consideración el uso de rutas relativas. Es decir, si vamos a utilizar un archivo dentro de un documento, lo vamos a mandar llamar en relación a este. Y no en relación al sistema de archivos.

Las rutas relativas (generalmente) inician con: . (directorio actual), .. (directorio superior), /nombre (una carpeta llamada nombre desde el directorio actual), nombre (carpeta llamada nombre desde el directorio actual).

Si utilizamos rutas absolutas (muestran la ubicación del archivo dentro de todo el sistema de archivos). Lo más probable es que, cuando transportemos nuestro proyecto a otra computadora nos marque error (diciendo que no encuentra los archivos), aunque tengamos todos los archivos.

Las rutas absoultas generalmente empiezan con: C://, /Volumes/Nombre del DD, file://

PHP Básico

Lenguaje de scripts.
Se interpreta al momento de ejecución (no compilado).

PHP se escribe como una serie de instrucciones (comandos) o declaraciones, al terminar cada instrucción se indica con punto y coma (;).

Se pueden insertar comentarios dentro del código de PHP de la siguiente manera:

- Utilizar 2 diagonales antes del comentario (1 línea).
- Utilizar el signo de número antes del comentario (1 línea).
- Insertar el comentario dentro de /* */ (varias líneas).

PHP Código embedido

Se puede insertar código php en archivos que contengan etiquetas HTML (la extensión del archivo se renombra para que sea .php). O se puede tener un archivo meramente PHP si va a realizar ciertas funciones que no necesariamente se tienen que mostrar al usuario final (conexión a la BD por ejemplo).

```
<?php
   línea(s) de código en PHP.
?>
```

ó

```
<?
   línea(s) de código en PHP.
?>
```

La versión con php es más segura ya que no todos los servidores tienen configurada la abreviación (sin php).
Se pueden tener múltiples bloques de código dentro de un mismo archivo.

PHP echo

Una función de PHP que vamos a estar utilizando bastante es echo.
Lo que hace esta función es mostrar lo que pongamos después de la misma.

<?php echo 2 + 2 ?>
-> 4

<?php echo ‘¡Hola Mundo!’ ?>
-> ¡Hola Mundo!

Lo que vamos a hacer (un poco más adelante) es indicarle a PHP (mediante echo) que escriba código HTML para que, al enviarlo al navegador lo visualice como HTML. Código que escribe código.

<?php echo ‘<p>Esto se va a mostrar como un párrafo escrito en HTML.</p>’ ?>
-> Esto se va a mostrar como un párrafo escrito en HTML.

Abrir echo.php, escribir-html.php

PHP Variables

Como su nombre lo indica, las variables son contenedores los cuales pueden cambiar el valor que contienen durante la ejecución de un script.

Para crear una variable se tienen que considerar las siguientes reglas:

- Comienzan con el signo de dolar ($).
- El primer caracter (después del signo de dolar) no puede ser un número.
- Sólo se acepta guión bajo (no se acepta ningún otro caracter de puntuación o espacio en blanco).
- Son sensibles a mayúsculas y minúsculas.

PHP Variables

No espacios.

```
$nombre variable.
```

Hay una variable llamada nombre y marca error porque no reconoce la palabra variable.

Notación con guión bajo.

```
$nombre_variable;
```

Notación CamelCase.

```
$nombreVariable;
```

PHP Variables (2)

No es necesario inicializar variables en PHP (dar un valor inicial al momento de crearlas) sin embargo es una práctica buena. Ya que, las variables no inicializadas tienen un valor predeterminado de su tipo en función del contexto en el que se utilicen.

Para asignar un valor se utiliza el signo de igualdad (=).

Nota:

Existe diferencia entre asignar y comparar un valor. Para comparar un valor en PHP se utilizan dos signos de igualdad (==).

```
// La variable va a contener el valor 1.
// Asignación
$variable = 1;

// Compara si la variable es igual a 1.
// Comparación
$variable == 1;
```


Abrir el archivo variables.php

PHP números

En PHP podemos trabajar con números. Por el momento vamos a distinguir 2 tipos de números.
Enteros (….-2, -1, 0, 1, 2….) y flotantes también conocidos como decimales, dobles (….-2., -1.1, 0.5, 1.7, 2.9….)

PHP cuenta con operadores matemáticos para realizar operaciones básicas:

Operación	Operador	Ejemplo	Resultado
Suma	+	$x + $y	30
Resta	-	$x - $y	10
Multiplicación	*	$x * $y	200
División	/	$x / $y	2
Módulo División	%	$x % $z	2
Incrementar	++	$x++	21
Decrementar	--	$y--	9

Valores

$x = 20; $y = 10; $z = 4.5;

Módulo División

Regresa el residuo de la división:

26 % 5 = 1

26 % 27 = 26

10 % 2 = 0

Incremento / Decremento

Se le suma una unidad* al valor que le apliquemos el operador

$a = $a + 1;

$a = $a - 1;

*Se puede asignar otro valor para el operador (2, 3, 4).

PHP números (2)

Los cálculos en PHP siguen las mismas reglas que se utilizan en aritmética.

Precedencia	Grupo	Operador	Regla
Mas alta	Paréntesis	( )	Las operaciones contenidas dentro de los paréntesis son evaluadas primero. Si estas expresiones se encuentran anidadas, las más internas son evaluadas primero.
Siguiente	Multiplicación y División	* / %	Estos operadores son evaluados a continuación. Si la expresión contiene 2 o más operadores, éstos serán evaluados de izquierda a derecha
Mas baja	Suma y Resta	+ -	Estos son los operadores que se van a evaluar al final de la expresión. Si una expresión contiene 2 o más operadores, éstos serán evaluados de izquierda a derecha.

PHP números (3)

PHP ofrece una manera rápida de realizar cálculos en una variable y asignar el resultado a la misma variable a través de operadores de cálculo y asignación combinados.

Operador	Ejemplo	Equivalente a
+=	$a += $b	$a = $a + $b
-=	$a -= $b	$a = $a - $b
*=	$a *= $b	$a = $a * $b
/=	$a /= $b	$a = $a / $b
%=	$a %= $b	$a = $a % $b

numeros-letras numeros.php

PHP letras

PHP cuenta con el manejo de cadenas de texto (String).
Por el momento vamos a trabajar con ellas de manera muy superficial.
Una cadena de texto puede estar entre comillas sencillas (apóstrofe) o comillas dobles.

```
<?php
    echo ‘soy una cadena de texto’;
    echo “yo también soy una cadena de texto”;
?>
```

La diferencia radica en que, si se utilizan comillas sencillas PHP envía la cadena de texto sin interpretar el contenido. Si se utilizan comillas (dobles) PHP trata de interpretar el valor de las variables y otros caracteres (de escape).

```
<?php
   $variable = 4;
    echo ‘El valor de 2 + 2 es $variable’;
?>
-> El valor de 2 + 2 es $variable
<?php
   $variable = 4;
    echo “El valor de 2 + 2 es $variable”;
?>
-> El valor de 2 + 2 es 4
```

PHP letras (2)

Así como en como con los operadores aritméticos, PHP nos permite combinar y asignar el resultado a la misma variable, también nos lo permite con el texto. Supongamos que tenemos la variable $a con el valor Hola.

Operador	Ejemplo	Equivalente a
.=	$a .= “Mundo”	$a = $a . “Mundo”

numeros-letras letras.php

PHP Comparaciones

La mayoría de decisiones están basadas en la comparación de al menos dos valores. Para realizar esto se utilizan los operadores de comparación.

Símbolo	Nombre	Uso
==	Igualdad	Devuelve true si los valores son iguales; de lo contrario, devuelve false.
!=	Desigualdad	Devuelve true si los valores son diferentes; de lo contrario, devuelve false.
===	Idéntico	Determina si los dos valores son idénticos. Para ser considerado idénticas, no sólo debe tienen el mismo valor, pero también ser del mismo tipo de datos (por ejemplo, tanto en números de punto flotante).
!==	No idéntico	Determina si los valores no son idénticas (de acuerdo con los mismos criterios que el operador anterior).
>	Mayor que	Determina si el valor en el lado izquierdo es mayor que el valor del lado derecho
>=	Mayor que o igual a	Determina si el valor en el lado izquiero es mayor o igual que el valor del lado derecho
<	Menor que	Determina si el valor en el lado izquiero es menor que el valor del lado derecho.
=<	Menor que o igual a	Determina si el valor en el lado izquierdo es menor o igual al valor del lado derecho.

PHP Boolean

Al realizar una comparación podemos obtener un valor de verdad (verdadero o falso), en PHP se puede asignar un valor de verdad a una variable de manera explícita (se conoce como boolean).

Hay ocasiones en que comparar 2 valores no es suficiente. PHP permite establecer una serie de condiciones utilizando operadores lógicos para especificar si todas o algunas deben de ser cumplidas.

Los operadores lógicos son:

Símbolo	Nombre	Uso
&&	Y Lógico (Conjunción)	Evalúa si ambas condiciones son verdaderas (true).
||	O Lógico (Disjunción)	Evalúa si al menos una condición es verdadera (regresa true). Si ambas son falsas regresa false.
!	Negación	Cambia el valor de verdad que contenga una variable o expresión.

PHP Tablas de verdad

$a	$b	conjunción $a && $b	disjunción $a || $b
true	true	true	true
false	true	false	true
false	true	false	true
false	false	false	false

NOTA:
Se pueden agrupar varias conjunciones / disjunciones. Para estar seguros del orden de comparación se recomienda la agrupación con paréntesis.

PHP Condicionales

if: Analiza el valor de verdad de una expresión, si resulta ser verdadero ejecuta el código que se encuentra dentro de las llaves, sino continúa con el flujo normal del código.

else: Hay veces en que se requiere ejecutar ciertas líneas de código si no se cumple con una condición y, para esto se utiliza else. Se utiliza como extensión de if ya que el código que contiene entre llaves se ejecuta si la condición de if no se cumple.

elseif: Combinación de else e if. Al igual que else se utiliza como extensión de if pero su diferencia radica en que antes de las llaves que contienen el código a ejecutar, contiene otra expresión para compararse.

switch: Esta declaración es una alternativa a if...else. La diferencia radica en que al utilizar switch se espera comparar más valores que al utilizar if.

PHP Ciclos

Un ciclo se refiere a que una sección de código va a ser repetida hasta que se cumpla con cierta condición. PHP cuenta con 4 maneras distintas de realizar un ciclo: while, do...while, for, foreach.

while es la manera más simple de realizar un ciclo. La estructura es la siguiente:

```
while (condición es verdadera) {
    código repetido;
}
```

do...while es similar a while. Su estructura varía un poco:

```
do {
  código repetido;
} while (condición es verdadera);
```

La diferencia entre estos dos ciclos es que while evalúa la condición antes de ingresar al ciclo mientras que en el ciclo do..while la condición se evalúa después. Por lo tanto en do...while el código interno se ejecuta al menos una vez incluso si la condición no se cumple.

Una desventaja con este tipo de ciclos radica en que, dentro del ciclo debemos de tener código que modifique la condición para que sea verdadera. Si no contamos con ello entraríamos a un ciclo sin fin, el cuál hace que se congele la computadora.

PHP Ciclos (2)

for es una de las maneras más utilizadas (al menos por mi) para realizar ciclos. Su estructura es la siguiente:

```
for (inicializar cuenta; evaluar; incrementar cuenta) {
  código a ejecutarse;
}
```

Al utilizar for es menos probable que se genere un ciclo infinito ya que todas las condiciones (y la comparación) se encuentran en la primer línea.

foreach es el último tipo de ciclo y se utiliza exclusivamente en arreglos. Existen 2 formas en las cuales se puede realizar este ciclo, dependiendo si el arreglo es indizado o asociativo.

Arreglo indizado:

```
foreach(nombre_del_arreglo as variable_temporal) {
    código a ejecutarse;
}
```

Arreglo asociativo:

```
foreach(nombre_del_arreglo as nombre_clave => valor) {
    código a ejecutarse;
}
```

PHP Ciclos (3)

Para lograr que un ciclo llegue a un fin prematuramente (cuando cierta condición se cumpla) hay que insertar la palabra break dentro de una declaración condicional. Tan pronto como el script se encuentre con esta palabra, se sale del ciclo.

Para saltar una iteración del ciclo cuando se cumpla una condición determinada hay que utilizar la palabra continue. En lugar de salir vuelve a la parte superior del ciclo y ejecuta la siguiente iteración.

PHP (re)utilizar archivos

require: Manda llamar el archivo especificado y, si no lo encuentra no continúa con la ejecución del código.

include: Manda llamar el archivo especificado y, si no lo encuentra manda una advertencia e intenta continuar con la ejecución del código.

require_once: Mismo comportamiento que require pero se asegura de mandar llamar el archivo solo una vez para evitar conflicto de nombre de funciones duplicado.

include_once: Mismo comportamiento que include pero se asegura de mandar llamar el archivo solo una vez para evitar conflicto de nombre de funciones duplicado.

Se pueden llamar archivos de cualquier tipo (html, php, js, css).

Ej. require-include
documento-require.php, documento-include.php, documento-require-valido.php, documento-include-valido.php

Formularios

Elementos Básicos

Enviar información que no pueda ser manipulada por el usuario.
input type="hidden"

El usuario decide la información que quiere ingresar.
input type="text"
textarea rows="#" cols="#"

El se muesta un campo de texto pero el texto es reemplazado por otro caracter.
input type="password"

Se habilita un campo de texto con un botón para buscar un archivo dentro de la computadora del usuario.
input type='file'

Elementos Excluyentes

El usuario puede elegir solo una entre varias opciones.

radio - Generalmente utilizado cuando las opciones a elegir son pocas (2 o 3), para que el usuario tenga una visualización rápida de las mismas.

select - Generalmente utilizado cuando las opciones a elegir son más de 3.

Formularios

Elementos Incluyentes

El usuario puede elegir varias opciones

Checkbox - Casi como la única opción existente. Se utiliza sin importar cuántas opciones existan para que el usuario elija alguna o varias.

Select con atributo “multiple” (casi no utilizado) - Este elemento se muestra como una caja de texto con las opciones en cada renglón. Para poder seleccionarlas es necesario dar clic y utilizar CTRL (Win / Linux) CMD (OS X).

Acciones

Son los botones que el usuario va a utilizar para llevar a cabo una acción, aunque pueden ser utilizados indistintamente hay que hacer notar la diferencia.

input type='submit' - Se utiliza en los formularios y la acción que va a realizar es la que se encuentra especificada como valor del atributo action del formulario (etiqueta de apertura form).

input type='button' - Aunque se visualiza igual que submit, este botón no realiza acción alguna (se le tiene que asignar vía JS).

input type='image' - Se utiliza cuando se quiere reemplazar la visualización del botón con una imagen. Similar a submit porque la acción que realiza es la que se encuentra especificada dentro de la etiqueta de apertura del formulario.

Formularios

Elementos de Agrupación

Son elementos que tienen la función de delimitar visualmente algunos campos de un formulario y pueden contener una etiqueta que indica lo que se está representando.

fieldset: Se encarga de agrupar los campos dentro de un recuadro.
legend: Se utiliza como título o descripción del conjunto de campos (va dentro de fieldset).

POST / GET

POST

Se utiliza para enviar información al servidor. Generalmente se utiliza para realizar acciones que modifican la BD (insertar, actualizar, eliminar). Los parámetros no son visibles en el URL. No se puede agregar a favoritos (bookmark).

GET

Se utiliza para obtener información del servidor. Se utiliza para realizar consultas. Los parámetros son visibles en el URL. Se puede agregar a favoritos (bookmark).

Conexión a Base de Datos (MySQL)

mysql_connect
mysql_select_db
mysql_close

php-mysql: conexion.php

Consulta a Base de Datos (MySQL)

mysql_query

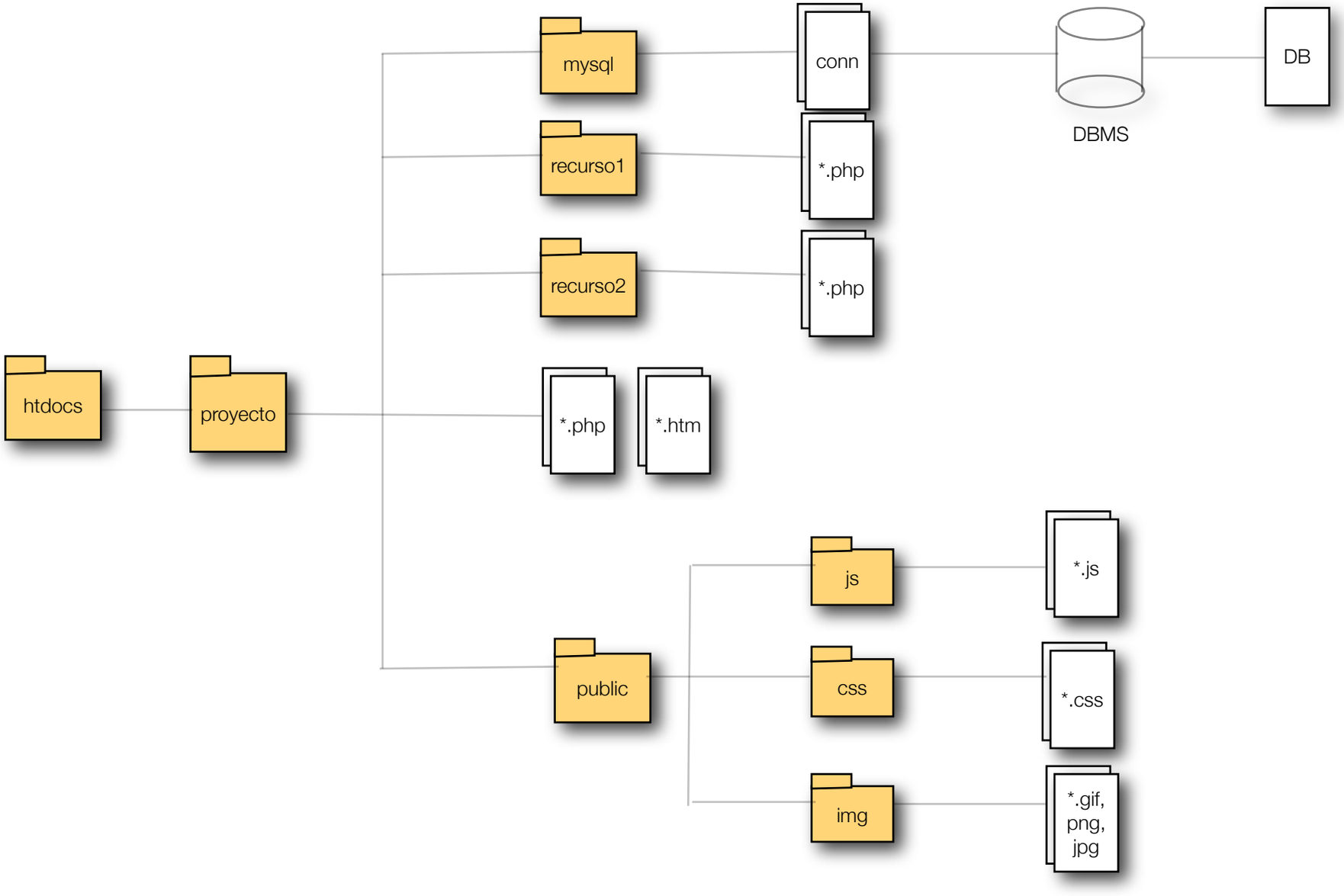
mysql_fetch_row
mysql_fetch_object
mysql_fetch_array
mysql_result

mysql_affected_rows
mysql_num_rows

Errores MySQL

die

Organización Proyecto



PHP / MySQL - Recursos

PHP	MYSQL	Método
index	SELECT	GET
show	SELECT	GET
new (formulario*)	------	GET
create	INSERT	POST
edit (formulario*)	------	GET
update	UPDATE	POST
destroy	DELETE	POST

NOTA:

Algunas veces se tienen que realizar consultas en los formularios.

Ya sea para llenar los datos de radio, select, check.

Si se trata de una actualización se utiliza para que el usuario tenga como referencia los valores anteriores.

MySQL vía consola - conexión

Conexión / Desconexión

Vía TCP / IP

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

Vía Socket (generalmente usado en LAMP - localhost)

```
mysql --socket=/var/mysql/mysql.sock -u root
mysql --socket=/var/mysql/mysql.sock -u root -proot
```

MySQL vía consola - prompt

Prompt de MySQL
Interacción con el Servidor MySQL vía comandos.

mysql> SELECT VERSION( );
Muestra la versión de MySQL que se está utilizando.

mysql> SELECT USER( );
Muestra el usuario con el que se inició sesión.

mysql> exit;
Sale de la consola de MYSQL al Sistema Operativo.

MySQL vía consola - prompt (2)

mysql> Listo para un comando nuevo.

-> Espera la siguiente línea, de un comando de varias líneas.

'> Espera la siguiente línea, indica terminar una cadena que inicia con un apóstrofe (').

"> Espera la siguiente línea , indica terminar una cadena que inicia con comillas (").

MySQL vía consola - comandos

Un comando normalmente consiste en una sentencia SQL seguida por un punto y coma.

Cuando se emite un comando, mysql lo envía al servidor para su ejecución y muestra los resultados, luego imprime otro prompt para indicar que está listo para otro comando.

Mysql imprime los resultados de consulta como una tabla (filas y columnas). La primera fila contiene las etiquetas para las columnas. Las filas siguientes son los resultados de la consulta.

Mysql muestra cuántas filas fueron devueltas y cuánto tiempo llevó la consulta a ejecutar, que le da una idea aproximada del rendimiento del servidor.

MySQL administración de usuarios

MySQL viene con un usuario predeterminado que es root. Este usuario puede realizar cualquier acción sobre cualquier base de datos que se encuentre en el sistema.

Para el ambiente de desarrollo es fácil utilizar este usuario (nos evitamos la creación de un usuario y asignación de permisos) pero cuando pasamos a producción es conveniente crear otros usuarios (generalmente un usuario asociado a una BD).

MySQL administración de usuarios - seguridad

MySQL viene con un modelo se seguridad, capaz de controlar prácticamente cada acción que realicen los usuarios. El rango abarca desde los comandos que pueda ejecutar cada usuario hasta el número de consultas que puede realizar en una hora. Este modelo trabaja en una secuencia de 2 pasos:

Paso 1 - Autenticación	Se examina el servidor, nombre de usuario y contraseña. Si concuerda con la tabla de privilegios de MySQL, el usuario es autorizado. De otra forma se niega el intento de conexión.
Paso 2 - Autorización	Una vez autenticado, el comando enviado por el usuario se examina y se compara con los privilegios definidos para el usuario. Si el usuario tiene suficientes privilegios, el comando se ejecuta. De otra forma se niega la ejecución del comando.

MySQL administración de usuarios - crear

La manera más fácil de crear una cuenta de usuario es con GRANT.

```
mysql> GRANT privilegio1, privilegio2, privilegioN ON mi_base_de_datos.*
    -> TO 'usuario'@'host' IDENTIFIED BY 'password';

mysql> GRANT SELECT, INSERT, UPDATE ON proyecto.*
    -> TO 'eamexicano'@'localhost' IDENTIFIED BY 'pwd_secreto';
```

Para permitir todos los privilegios en todas las bases de datos, utilizar *.* en lugar del nombre de la BD y ALL PRIVILEGES en lugar de los privilegios.

Asegurarse de incluir la cláusula IDENTIFIED BY cuando se crean los nuevos usuarios. Si no se incluye esta información se va a crear el usuario sin contraseña.

MySQL administración de usuarios - eliminar

Para eliminar un usuario completamente del sistema, quitándo todos los privilegios y borrando su cuenta de acceso se utiliza el comando DROP USER.

```
mysql> DROP USER ‘usuario’@’host’;
```

MySQL administración de usuarios - privilegios (+)

Para conceder privilegios adicionales a un usuario también se utiliza el comando GRANT como se utilizó para la creación del usuario. MySQL reconocerá que existe el usuario y solo modificará los privilegios.

```
mysql> GRANT privilegio1, privilegio2, privilegioN ON mi_base_de_datos.*
    -> TO 'usuario'@'host';

mysql> GRANT DELETE ON proyecto.* TO 'eamexicano'@'localhost';
```

MySQL administración de usuarios - privilegios (-)

Para remover privilegios a un usuario también se utiliza el comando REVOKE.

```
mysql> REVOKE privilegio1, privilegio2, privilegioN ON mi_base_de_datos.*
    -> FROM ‘usuario’@’host’;

mysql> REVOKE DELETE, UPDATE FROM ‘eamexicano’@’localhost’;
```

MySQL administración de usuarios - privilegios

Los mismos comandos GRANT y REVOKE pueden ser utilizados para administrar los privilegios de los usuarios a nivel tabla y columna.

```
mysql> GRANT INSERT ON proyecto.usuarios TO
    -> ‘eamexicano’@’localhost’;

mysql> GRANT INSERT (estatus), SELECT (estatus)  ON proyecto.usuarios
    -> TO ‘eamexicano’@’localhost’;
```

MySQL administración de usuarios - renombrar

Para renombrar un usuario existente se utiliza el comando RENAME USER

```
mysql> RENAME USER ‘usuario’@’host’ TO ‘usuario_uno’@’host’;

mysql> RENAME USER ‘eamexicano’@’localhost’ TO ‘eam’@’localhost’;
```

MySQL BD - Mostrar

Para conocer las bases de datos existentes en el sistema se utiliza el comando SHOW DATABASES.

mysql> SHOW DATABASES;

MySQL BD - Crear

Para crear una base de datos desde la consola de mysql se usa el comando CREATE DATABASE:

```
mysql> CREATE DATABASE mi_base_de_datos;
```

También se puede crear una base de datos sin iniciar sesión en el servidor utilizando el cliente mysqladmin:

```
$> mysqladmin -u root -p create mi_base_de_datos;
```

MySQL BD - Utilizar

Para utilizar una base de datos desde la consola de mysql se usa el comando USE.

```
mysql> USE mi_base_de_datos;
```

Se puede preseleccionar la base de datos a utilizar desde el momento de iniciar de sesión:

```
$> mysql -u root -p mi_base_de_datos;
```

MySQL Tablas - Mostrar

Para conocer las tablas que existen en una base de datos se utiliza el comando SHOW TABLES.

mysql> SHOW TABLES FROM base_de_datos;

Si actualmente se está trabajando en una base de datos y se quieren conocer las tablas de la misma base de datos solo se necesita hacer SHOW TABLES.

mysql> SHOW TABLES;

MySQL Tablas - Crear

Para crear tabla, hay que utilizar el comando CREATE TABLE junto con el nombre de la tabla y la definición de sus columnas:

```
mysql> CREATE TABLE nombre_de_la_tabla (
columna1 definición, columna2 definición,columnaN definición
);
```

```
mysql> CREATE TABLE usuarios (
id INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY,
nombre VARCHAR(255) NOT NULL,
email VARCHAR(255) NOT NULL
);
```

MySQL Tablas - Mostrar estructura

Para mostrar la estructura de una tabla se utiliza el comando DESCRIBE junto con el nombre de la tabla:

```
mysql> DESCRIBE nombre_de_la_tabla;
```

```
mysql> DESCRIBE usuarios;
```

MySQL Tablas - Modificar estructura

Se pueden agregar, eliminar y modificar las columnas de una tabla utilizando el comando ALTER TABLE.

Para agregar una columna a una tabla existente.

```
mysql> ALTER TABLE nombre_de_la_tabla ADD COLUMN
    -> nombre_columna tipo_columna atributos_tipo_columna;
```

```
mysql> ALTER TABLE usuarios ADD COLUMN telefono VARCHAR(20)
    -> NULL;
```

Para agregar la columna en una posición específica se utiliza la sentencia AFTER;

```
mysql> ALTER TABLE usuarios ADD COLUMN telefono VARCHAR(20)
    -> NOT NULL AFTER nombre;
```

MySQL Tablas - Modificar estructura (2)

Para modificar una columna se utiliza CHANGE COLUMN.

```
mysql> ALTER TABLE nombre_de_la_tabla CHANGE COLUMN
    -> nombre_columna tipo_columna atributo_tipo_columna;
```

```
mysql> ALTER TABLE usuarios CHANGE COLUMN telefono VARCHAR(30) NOT
NULL;
```

MySQL Tablas - Modificar estructura (2)

Para modificar una columna se utiliza CHANGE COLUMN.

```
mysql> ALTER TABLE nombre_de_la_tabla CHANGE COLUMN
    -> nombre_columna tipo_columna atributo_tipo_columna;
```

```
mysql> ALTER TABLE usuarios CHANGE COLUMN telefono VARCHAR(30) NOT
NULL;
```

Para eliminar una columna se utiliza DROP COLUMN.

```
mysql> ALTER TABLE nombre_de_la_tabla DROP COLUMN
    -> nombre_columna;
```

```
mysql> ALTER TABLE usuarios DROP COLUMN telefono;
```

MySQL Tablas - Renombrar

Para renombrar una tabla se utiliza el comando ALTER TABLE junto con la sentencia RENAME:

mysql> ALTER TABLE nombre_de_la_tabla RENAME nuevo_nombre;

mysql> ALTER TABLE usuarios RENAME usuario;

MySQL Tablas - Eliminar

Para eliminar una tabla se utiliza el comando DROP TABLE junto con el nombre de la tabla:

mysql> DROP TABLE nombre_de_la_tabla;

mysql> DROP TABLE usuario;

MySQL Datos - Operaciones básicas

Acción	MySQL	Descripción
Create	INSERT	Agregar información.
Read	SELECT	Consultar información.
Update	UPDATE	Actualizar información
Delete	DELETE	Eliminar información.

MySQL Datos - Insert - Crear

Para insertar un registro nuevo se tiene que especificar el valor asignado a todas las columnas de la tabla.

```
mysql> INSERT INTO tabla VALUES (‘valor1’,‘valor2’,‘valorN’);
```

```
mysql> INSERT INTO usuarios (‘1’, ‘Emmanuel’, ‘12-34-56-78’);
```

Si no se tienen todos los valores se pueden seleccionar las columnas con las cuales se va a generar el registro.

```
mysql>  INSERT INTO tabla (columna1, columna3) VALUES (‘valor1’,‘valor2’);
```

```
mysql>  INSERT INTO usuarios (nombre, telefono)
    ->  VALUES (‘Emmanuel’, ‘12-34-56-78’);
```

MySQL Datos - Select - Leer

mysql> SELECT (columna1, columna2) FROM tabla;

mysql> SELECT (nombre, correo) from usuarios;

Si se quiere obtener todo el registro (todas las columnas) se puede usar el comodín *.

mysql> SELECT * FROM tabla;

mysql> SELECT * from usuarios;

MySQL Datos - Update - Actualizar

```
mysql> UPDATE tabla  SET columna1 = “X”, columna2 = “Y”
    ->  WHERE columna3 = “Z”;

mysql> UPDATE usuarios SET telefono = “5591918327”
    -> WHERE nombre = “Emmanuel”;
```

MySQL Datos - Delete - Eliminar

mysql> DELETE FROM tabla WHERE columna3 = “Z”;

mysql> DELETE FROM usuarios WHERE nombre = “Emmanuel”;

MySQL Datos - Comparar

La cláusula WHERE se utiliza para refinar las búsquedas, al limitar los resultados de búsqueda a los registros que cumplan con ciertas condiciones. Las condiciones están dadas en forma de una o varias condiciones.

condición = atributo operador valor

WHERE (condición)

WHERE (condición operador_lógico condición)

WHERE (id < 10)

WHERE (id < 10 AND name = ‘Emmanuel’)

AND, OR, NOT….funcionan como && || ! en PHP

MySQL Datos - Ordenar

Al obtener resultados de una consulta los podemos ordenar por una o varias columnas en el orden que designemos:

ORDER BY id ASC
ORDER BY id DESC

Primero ordena los resultados por id y , dentro de los resultados que pertenecen a la misma id ordena los resultados por nombre.

ORDER BY id, nombre ASC
ORDER BY id, nombre DESC

MySQL Datos - NULL

NULL es la representación para valores no conocidos.

Generalmente al trabajar con MySQL, establece que, si al crear un registro no se asigna un valor NULL es el valor predeterminado.

Al crear una columna se puede definir que no acepte valores nulos utilizando la sentencia NOT NULL.

También se puede definir un valor predeterminado por si no se asigna al momento de crear un registro utilizando la cláusula DEFAULT.

MySQL Datos - Búsqueda en texto

Para realizar búsquedas en texto

% cualquier número de caracteres.

WHERE columna LIKE ‘%variable%’
WHERE columna NOT LIKE ‘%variable%’

WHERE nombre LIKE ‘E%’ // nombre que comience con E
WHERE nombre LIKE ‘%E’ // nombre que termine con E
WHERE nombre LIKE ‘%E%’ // nombre que contenga con E

WHERE nombre NOT LIKE ‘E%’ // nombre que NO comience con E
WHERE nombre NOT LIKE ‘%E’ // nombre que NO termine con E
WHERE nombre NOT LIKE ‘%E%’ // nombre NO que contenga con E

MySQL Datos - Faltantes

Funciones
Respaldos
Software de Administración (phpMyAdmin, SequelPro, etc)
Batch
Teoría - Relaciones (sin teoría relacional)
Ejercicios MySQL