PHP y MySQL

Proyecto Universidad Empresa - PUE

Agustín Calderón

agustincl@gmail.com

PHP y MySQL

Tema 0: Arquitectura y servicios Internet

Agustín F. Calderón M.
agustincl@gmail.com

Arquitectura y servicios Internet

1. Arquitectura de Internet (TCP/IP)
2. Servicios de Internet
3. Protocolo HTTP/HTTPS

Agustín F. Calderón M.
agustincl@gmail.com

Arquitectura de Internet (TCP/IP)

- **Internet Engineering Task Force (IETF)**
  - La IETF es mundialmente conocida por ser la entidad que regula las propuestas y los estándares de Internet, conocidos como RFC.

- **Internet Corporation for Assigned Names and Numbers (ICANN)**
  - Es la responsable de asignar las direcciones del protocolo IP, de los identificadores de protocolo, de las funciones de gestión del sistema de dominio y de la administración del sistema de servidores raíz.

Agustín F. Calderón M.
agustincl@gmail.com

Arquitectura de Internet (TCP/IP)

Asia Pacific - Red

Europe/Middle East/Central Asia/Africa - Green

North America - Blue

Latin American and Caribbean - Yellow

RFC1918 IP Addresses - Cyan

Unknown - White

Date: Nov 22 2003

opte.org

Agustín F. Calderón M.
agustincl@gmail.com

Servicios de Internet

- **Email**
  - Utiliza el protocolo SMTP ("Simple Mail Transfer Protocol"), para la recepción y envío.
- **TELNET**
  - Se utiliza para conectar a equipos remotos mediante la Red emulando un terminal del equipo al que se realiza la conexión.
- **FTP**
  - ("File Transfer Protocol"), se usa para enviar o recibir ficheros (de cualquier tipo) entre dos equipos conectados a la red.
- **DNS**
  - ("Domain Name Service") es usado por otros, como TELNET, FTP, WWW, etc. para conseguir las direcciones IP (numéricas) de las máquinas remotas a partir de los nombres de dominio.

Agustín F. Calderón M.
agustincl@gmail.com

Servicios de Internet 2

- **Gopher**
  - Un servicio de información basado en servidores y que sirve de interfaz para otros servicios de información.
- **WAIS**
  - ("Wide Area Information Service"). Se trata de otro servicio de información basado en bases de datos de ficheros que permiten su rápida localización.
- **Finger**
  - Un servicio de identificación de usuarios.
- **WWW**
  - La Web, WWW, W3. Un servicio basado en HTTP (Hyper Text Transfer Protocol), el último y más popular que está fagocitando a muchos de los anteriores [2].

Agustín F. Calderón M.
agustincl@gmail.com

Servicios de Internet 3

- **WWW**
  - La Web, WWW, W3. Un servicio basado en HTTP (Hyper Text Transfer Protocol), el último y más popular que está fagocitando a muchos de los anteriores [2].
- **NFS**
  - ("Network File System"). Un sistema que permite a equipos físicamente distantes, compartir discos y directorios mediante la técnica denominada RPC ("Remote Procedure Call").
- **NIS**
  - ("Network Information Services"). También basados en RPC, permite que varios sistemas puedan compartir una misma base de datos situada en remoto.
- **R**
  - Tales como rlogin, rsh y otros. Utilizan la idea de acuerdos entre sistemas (hosts trusting), que permite ejecutar comandos y otras órdenes en equipos remotos sin requerir un password.

Agustín F. Calderón M.
agustincl@gmail.com

Protocolo HTTP/HTTPS

- **HTTP**
  - ("Hyper-Text Transfer Protocol") (HTTP 1.2) es un protocolo de Capa de Aplicación para distribuir sistemas de información hipermedia.
  - HTTP es un protocolo de consulta-respuesta, estándar para aplicaciones cliente/servidor.

Agustín F. Calderón M.
agustincl@gmail.com

Protocolo HTTP/HTTPS 2

- **URI**
  - Los recursos a ser accedidos via HTTP son identificados usando URIs (“Uniform Resource Identifiers”)
- **URL**
  - Más específicamente, URLs (“Uniform Resource Locators”) usan los esquemas HTTP/HTTPS.
- **Métodos**
  - HTTP define 8 métodos: HEAD, GET, POST, PUT, DELETE, TRACE, OPTIONS, CONNECT.
- **Respuesta**
  - 1xx Mensajes, 2xx Operación exitosa, 3xx Redirección, 4xx Error por parte del cliente, 5xx Error del servidor.
  - 200 OK, 404 No encontrado, 500 Error interno.

Agustín F. Calderón M.
agustincl@gmail.com

PHP y MySQL

Tema 1: Introducción

Agustín F. Calderón M.
agustincl@gmail.com

Tema 1: Introducción

1. Introducción a PHP
2. Instalación de Apache
3. Instalación de PHP
4. Instalación de una distribución de Apache: XAMPP
5. Entornos de desarrollo para PHP
6. Recursos de PHP

Agustín F. Calderón M.
agustincl@gmail.com

Historia de php

1994

- PHP: personal home page.
- Era un CGI escrito en C por Rasmus Lerdorf.

1995

- Se libera php y se llama php 2.

1997

- Zeev Suraski y Andi Gutmans escriben el parser.
- PHP 3: Hypertext Preprocessor.
- Se libera en 1998

1999

- Zend Engine 1.0.
- Zend Technologies.

Agustín F. Calderón M.
agustincl@gmail.com

Historia de php 2

2000

- PHP 4 powered by Zend Engine 1.0.

2005

- PHP 5 powered by Zend Engine 2.0.

Agosto de 2008

- Php 4.4.9 se congela.

Noviembre 19 de 2009

- Php 5.3.1.

Agustín F. Calderón M.
agustincl@gmail.com

Benchmark

Mediciones de velocidad en PHP 5.

http://www.tiaon.com/wordpress/2007/04/10/fastest-loop-code-in-php5/

http://www.php.lt/benchmark/phpbench.php

Agustín F. Calderón M.
agustincl@gmail.com

Introducción a PHP

- **Lenguajes de *script***
  - PHP es un lenguaje de *script* del lado del servidor. Otros lenguajes similares son ASP, JSP o ColdFusion
  - Los scripts PHP están incrustados en los documentos HTML y el servidor los interpreta y ejecuta antes de servir las páginas al cliente
  - El cliente no ve el código PHP sino los resultados que produce

Agustín F. Calderón M.
agustincl@gmail.com

Introducción a PHP

- **¿Cómo funciona PHP? (1)**

<P>Hola, Ana</P>

Agustín F. Calderón M.
agustincl@gmail.com

Introducción a PHP

- **¿Cómo funciona PHP? (2)**

```
<?PHP
$nombre = "Ana";
print ("<P>Hola, $nombre</P>");
?>
```

<P>Hola, Ana</P>

Agustín F. Calderón M.
agustincl@gmail.com

Introducción a PHP

- **Breve historia de PHP**
  - Creado por Rasmus Lerdorf para uso personal en 1994
  - PHP = *PHP: Hypertext Preprocessor*
  - Versión actual: PHP 5
  - Es un módulo que se añade al servidor web y fue concebido inicialmente para Apache
- **¿Por qué PHP?**
  - Por sus ventajas: es potente, fácil de aprender, de libre distribución, permite el acceso a bases de datos y otras funcionalidades orientadas a la red
  - Dispone de abundante soporte en la Web

Agustín F. Calderón M.
agustincl@gmail.com

Introducción a PHP

- **Requisitos**
  - Servidor web Apache (www.apache.org)
  - con el módulo PHP (www.php.net)
  - y la base de datos MySQL (www.mysql.com) si se desea crear páginas dinámicas
- **Otras utilidades**
  - Herramientas para la gestión de MySQL, como PHPMyAdmin (www.phpmyadmin.net)
  - Editores de PHP, como DevPHP (www.sourceforge.net) o Eclipse (www.eclipse.org)
  - Manuales de PHP y MySQL

Agustín F. Calderón M.
agustincl@gmail.com

Instalación de Apache

- **Instalación y configuración de Apache**
  - Pasos:
    - Descargar
    - Instalar
    - Configurar

Agustín F. Calderón M.
agustincl@gmail.com

Instalación de Apache

- **Instalación y configuración de Apache. 1: descargar**
  - Conectar a www.apache.org
  - Seleccionar HTTP Server
  - Elegir la versión estable más reciente
    - Apache 2.2.8 (febrero 2008)
  - Descargar a una carpeta temporal

Agustín F. Calderón M.
agustincl@gmail.com

Instalación de Apache

- **Instalación y configuración de Apache. 2: instalar**
  - Ejecutar el archivo bajado y seguir las instrucciones
  - Introducir la información requerida:
    - Network Domain: localhost
    - Server name: localhost
    - Administrator's Email Address: webmaster@localhost
  - Instalar como servicio (no disponible en W9X)
  - Seleccionar instalación típica e instalar en la carpeta por defecto, c:\Archivos de Programa\Apache Group\
  - Se crea el grupo de programas Apache HTTP Server y la carpeta c:\Archivos de Programa\Apache Group\Apache2
  - Arrancar el servidor:
    - Inicio > Programas > Apache HTTP Server 2.2.8 > Control Apache Server > Start
  - Ejecutar el navegador y cargar http://localhost/
  - Si sale la página de inicio del servidor, la instalación habrá sido correcta

Agustín F. Calderón M.
agustincl@gmail.com

Instalación de Apache

- **Instalación y configuración de Apache. 3: configurar**
  - Archivo de configuración: httpd.conf, situado en c:\Archivos de Programa\Apache Group\Apache2\conf
  - Ejecutar Inicio > Programas > Apache HTTP Server 2.2.8 > Configure Apache Server > Edit the Apache httpd.conf Configuration File
  - Cambios:
    - DocumentRoot -> c:/htdocs (por defecto es c:/Archivos de Programa/Apache Group/Apache2/htdocs)
    - <Directory “c:/htdocs”>
    - Añadir index.php a DirectoryIndex
  - Crear carpeta c:\htdocs
  - Crear una página de inicio o copiar los ficheros de c:/Archivos de Programa/Apache Group/Apache2/htdocs
  - Reiniciar apache para leer la nueva configuración

Agustín F. Calderón M.
agustincl@gmail.com

Instalación de PHP

- **Instalación y configuración de PHP**
  - Pasos:
    - Descargar
    - Descomprimir
    - Configurar
    - Probar

Agustín F. Calderón M.
agustincl@gmail.com

Instalación de PHP

- **Instalación y configuración de PHP. 1: descargar**
  - Conectar a www.php.net
  - Seleccionar *downloads*, *windows binaries*
  - Elegir la versión estable actual, PHP 5.2.5 (febrero 2008)
  - Hay varias posibilidades: seleccionar PHP 5.2.5 zip package
  - Elegir un *mirror* (España: rediris)
  - Descargar a una carpeta temporal

Agustín F. Calderón M.
agustincl@gmail.com

Instalación de PHP

- **Instalación y configuración de PHP. 2: descomprimir**
  - Extraer a la carpeta c:\
  - Se crea la carpeta c:\php-5.2.5-Win32. Renombrar a c:\php

Agustín F. Calderón M.
agustincl@gmail.com

Instalación de PHP

- **Instalación y configuración de PHP. 3: configurar**
  - Seguir las instrucciones del archivo install.txt
  - Copiar el archivo de configuración php.ini-recommended como php.ini en la carpeta del sistema (habitualmente c:\windows)
  - Editarlo y cambiar lo siguiente:
    - doc_root = c:\htdocs
    - extension_dir = c:\php\ext
  - Editar httpd.conf y añadir las líneas siguientes:
    - LoadModule php5_module c:/php/php5apache2.dll
    - AddType application/x-httpd-php .php
  - Reiniciar Apache

Agustín F. Calderón M.
agustincl@gmail.com

Instalación de PHP

- **Instalación y configuración de PHP. 4: probar**
  - Crear una página PHP de prueba y cargarla con el navegador
  - Ejemplo: prueba.php en c:\htdocs\
  - Ejecutar el navegador y teclear la URL http://localhost/prueba.php

```
<?PHP
phpinfo();
?>
```

prueba.php

Agustín F. Calderón M.
agustincl@gmail.com

Instalación de PHP

- **Ejercicio 1: configuración de PHP**
  - Comprobación del funcionamiento del módulo de PHP y de su configuración

Agustín F. Calderón M.
agustincl@gmail.com

Instalación de XAMPP

- **¿Qué es XAMPP?**
- **XAMPP** es una distribución de Apache que incluye MySQL, PHP y otras herramientas para el desarrollo de aplicaciones web, como phpMyAdmin
- XAMPP es gratuito y fácil de instalar: basta con descargar el archivo y extraerlo
- XAMPP es multiplataforma: existen versiones para Windows, Linux y Mac OS
- Precaución: la configuración por defecto de XAMPP no es segura y no es adecuada para un entorno de producción. El paquete incluye una herramienta para obtener una configuración más segura

Agustín F. Calderón M.
agustincl@gmail.com

Instalación de XAMPP

- **Instalación y configuración de XAMPP**
  - Pasos:
    - Descargar
    - Instalar
    - Probar

Agustín F. Calderón M.
agustincl@gmail.com

Instalación de XAMPP

- **Instalación y configuración de XAMPP. 1: descargar**
  - Conectarse a http://www.apachefriends.org/es/xampp.html
  - Seleccionar la plataforma adecuada
  - En el caso de Windows existen dos versiones: la normal (XAMPP) y la reducida (XAMPP Lite)
  - Para instalar la versión XAMPP Lite de Windows,
  - Seleccionar *Download* > XAMPP Lite
  - Elegir el fichero EXE autoextraíble
  - Guardar el archivo en disco

Agustín F. Calderón M.
agustincl@gmail.com

Instalación de XAMPP

- **Instalación y configuración de XAMPP. 2: instalar**
  - Ejecutar el archivo descargado. Extraer a la carpeta C:\Archivos de programa\xampplite
  - Ir a la carpeta C:\Archivos de programa\xampplite y ejecutar el programa setup_xampp.bat
  - Crear un acceso directo en el escritorio al programa xampp-control.exe situado en esa carpeta

Agustín F. Calderón M.
agustincl@gmail.com

Instalación de XAMPP

- **Instalación y configuración de XAMPP. 3: probar**
  - Ejecutar XAMPP haciendo doble clic sobre el icono xampp_control
  - Arrancar los módulos Apache y MySQL. Aparece el rótulo *Running* al lado de ambos
  - Pulsar el botón *Admin*. Aparecerá la página de inicio del servidor
  - Para parar XAMPP, pulsar el botón *Exit*

Agustín F. Calderón M.
agustincl@gmail.com

Instalación de XAMPP

- **Configuración segura de XAMPP**
  - Arrancar XAMPP y cargar la página de inicio
  - Seleccionar la opción *Chequeo de seguridad*
  - Pulsar el enlace recomendado para solucionar los problemas de seguridad
  - Establecer una contraseña para el administrador (*root*) de MySQL (por defecto está en blanco)
  - Crear un usuario con contraseña para proteger el acceso a la carpeta de XAMPP

Agustín F. Calderón M.
agustincl@gmail.com

Entornos de desarrollo para PHP

- **¿Cómo desarrollar un proyecto en PHP?**
  - Los ficheros PHP son ficheros de texto y se pueden crear con cualquier editor de texto, como el *WordPad* de Windows
  - Es mucho más conveniente utilizar **entornos de desarrollo** que permiten editar el código más cómodamente, y además proporcionan funciones como la detección y corrección de errores, visualización de las páginas en el navegador, ayuda sensible al contexto y gestión de todos los recursos asociados al proyecto
  - Algunos entornos de desarrollo:
    - Dev-PHP
    - Eclipse

Entornos de desarrollo para PHP

- **Instalación del editor Dev-PHP**
  - Descargar de http://sourceforge.net/projects/devphp
  - Ejecutar archivo descargado y seguir las instrucciones. Tomar los valores por defecto
  - Dev-PHP queda configurado como el editor predeterminado para los archivos PHP

NOTA

El editor Dev-PHP ocupa muy poco espacio y es muy simple de manejar. El entorno Eclipse (véase a continuación) ocupa bastante más espacio y consume más recursos del sistema, pero ofrece una potencia muy superior para el desarrollo de aplicaciones en PHP

Agustín F. Calderón M.
agustincl@gmail.com

Entornos de desarrollo para PHP

- **Entorno de programación Eclipse**
- **Eclipse** es una plataforma de desarrollo abierta creada inicialmente para construir aplicaciones Java
- Actualmente Eclipse soporta múltiples lenguajes y dispone de una enorme cantidad de componentes que cubren todas las fases del desarrollo de software
- El soporte de PHP lo proporciona el *plugin* **PDT** (*PHP Development Tool*)
- Para ejecutar Eclipse es preciso tener instalado el entorno de ejecución de Java (JRE), que se puede descargar de http://www.java.com/es/

Agustín F. Calderón M.
agustincl@gmail.com

Entornos de desarrollo para PHP

- **Instalación de Eclipse (Windows)**
  - Conectarse a http://www.eclipse.org
  - Seleccionar la pestaña *Downloads*
  - Elegir el paquete *Eclipse Classic – Windows*
  - Seleccionar un *mirror* y descargar el archivo a una carpeta temporal del disco
  - Descomprimir el archivo .zip en la carpeta C:\Archivos de programa. Se crea la carpeta Eclipse
  - Crear un acceso directo en el escritorio al fichero eclipse.exe situado en la carpeta C:\Archivos de programa\Eclipse

Agustín F. Calderón M.
agustincl@gmail.com

Entornos de desarrollo para PHP

- **Instalación del plugin PDT**
  - Ejecutar Eclipse
  - Seleccionar *Help* > *Software Updates* > *Find and Install*...
  - Marcar la casilla *Search for new features* y pulsar *Next*
  - Marcar la casilla *Europa Discovery Site*
  - Pulsar el botón *New Remote Site* y añadir:
    Name: PDT
    URL: http://download.eclipse.org/tools/pdt/updates
  - Pulsar *Finish*
  - En la ventana de resultados marcar la casilla PDT, expandir la línea de *Europa* y pulsar *Select Required*
  - Pulsar *Next*
  - Aceptar la licencia, pulsar *Next* y *Finish*

Agustín F. Calderón M.
agustincl@gmail.com

Recursos de PHP

- **Documentación**
  - Manuales en www.apache.org, www.php.net, www.mysql.com
- **Tutoriales, scripts, artículos**
  - www.phpbuilder.com
  - www.codewalkers.com
  - www.devshed.com
  - www.tutorialfind.com
  - www.melonfire.com

Agustín F. Calderón M.
agustincl@gmail.com

PHP y MySQL

Tema 2: Lenguaje PHP básico

José Mariano González Romano
mariano@lsi.us.es

Agustín F. Calderón M.
agustincl@gmail.com

Tema 2: Lenguaje PHP básico

1. Sintaxis básica
2. Tipos de datos
3. Variables
4. Constantes
5. Expresiones y operadores
6. Estructuras de control
7. Funciones
8. Tablas
9. Bibliotecas de funciones

Agustín F. Calderón M.
agustincl@gmail.com

Sintaxis básica

- PHP es sensible a las mayúsculas
- ¿Cómo se incrusta en la página web?

  <?PHP ... ?>

  recomendado, siempre disponible

  <?= expresión ?>

  equivale a <? echo expresión ?>
- Las instrucciones se separan con un ; como en C. La marca final ?> implica un ;
- Comentarios: como en C, /* … */ (varias líneas) y // (una línea)

```
/* Comentario de
varias líneas */
print “hola”; // Comentario de una línea
```

Agustín F. Calderón M.
agustincl@gmail.com

Sintaxis básica

- Para imprimir: **echo** y **print**

  echo: muestra una o más cadenas

  echo cadena1 [, cadena2…];

  echo “Hola mundo”;

  echo “Hola “, “mundo”;

  print: muestra una cadena

  print cadena;

  print “Hola mundo”;

  print “Hola “ . “mundo”;

Agustín F. Calderón M.
agustincl@gmail.com

Sintaxis básica

- Ejemplo:

```
<HTML>
<HEAD>
<TITLE>Mi primer programa en PHP</TITLE>
</HEAD>
<BODY>
<?PHP
print (“<P>Hola mundo</P>”);
?>
</BODY>
</HTML>
```

Agustín F. Calderón M.
agustincl@gmail.com

Sintaxis básica

- Uso de \n para generar código HTML legible
- a) Sin \n

Código PHP

```
print (“<P>Párrafo 1</P>”);
print (“<P>Párrafo 2</P>”);
```

Código HTML

```
<P>Párrafo 1</P><P>Párrafo 2</P>
```

Salida

```
Párrafo 1

Párrafo 2
```

Agustín F. Calderón M.
agustincl@gmail.com

Sintaxis básica

- Uso de \n para generar código HTML legible
- b) Con \n

Código PHP

```
print (“<P>Párrafo 1</P>\n”);
print (“<P>Párrafo 2</P>\n”);
```

Código HTML

```
<P>Párrafo 1</P>
<P>Párrafo 2</P>
```

Salida

```
Párrafo 1

Párrafo 2
```

Agustín F. Calderón M.
agustincl@gmail.com

Sintaxis básica

- **Ejercicio 1: programa que muestra un mensaje**
  - Ilustra cómo incrustar código PHP en un documento HTML y cómo imprimir desde PHP

Agustín F. Calderón M.
agustincl@gmail.com

Sintaxis básica

- Inclusión de ficheros externos:
  - **include**()
  - **require**()
- Ambos incluyen y evalúan el fichero especificado
- Diferencia: en caso de error include() produce un warning y require() un error fatal
- Se usará require() si al producirse un error debe interrumpirse la carga de la página
- Ejemplo:

Sintaxis básica

```
<HTML>
 <HEAD>
 <TITLE>Título</TITLE>

<?PHP
 // Incluir bibliotecas de funciones
 require ("conecta.php");
 require ("fecha.php");
 require ("cadena.php");
 require ("globals.php");

?>
 </HEAD>
 <BODY>

<?PHP
 include ("cabecera.html");

?>

// Código HTML + PHP

. . .

<?PHP
 include ("pie.html");

?>

</BODY>

</HTML>
```

Agustín F. Calderón M.
agustincl@gmail.com

Tipos de datos

- PHP soporta 8 **tipos de datos primitivos**:
  - Tipos escalares: boolean, integer, double, string
  - Tipos compuestos: array, object
  - Tipos especiales: resource, NULL
- El tipo de una variable no se suele especificar. Se decide en tiempo de ejecución en función del contexto y puede variar
- Funciones de interés:
  - La función gettype() devuelve el tipo de una variable
  - Las funciones is_*type* comprueban si una variable es de un tipo dado:

    is_array(), is_bool(), is_float(), is_integer(), is_null(), is_numeric(), is_object(), is_resource(), is_scalar(),

    is_string()
  - La función var_dump() muestra el tipo y el valor de una variable. Es especialmente interesante con los arrays

Agustín G. Calderón M.
agustincl@gmail.com

Tipos de datos

- Tipo **integer** (números enteros)
  - 27, -5, 0
- Tipo **double** (números reales)
  - 1.234, -5.33
- Tipo **boolean** (lógico)
  - Valores: *true*, *false* (insensibles a las mayúsculas)
  - El 0 y la cadena vacía tienen valor *false*

Agustín F. Calderón M.
agustincl@gmail.com

Tipos de datos

- Tipo string:
  - Las cadenas se encierran entre comillas simples o dobles:
    - ‘simples’: admite los caracteres de escape \’ (comilla simple) y \\ (barra). Las variables **NO** se expanden
    - “dobles”: admite más caracteres de escape, como \n, \r, \t, \\, \$, \”. Los nombres de variables **SÍ** se expanden
    - Ejemplos:

```
$a = 9;
print ‘a vale $a\n’;
// muestra a vale $a\n
print “a vale $a\n”;
// muestra a vale 9 y avanza una línea
print “<IMG SRC=‘logo.gif’>”;
// muestra <IMG SRC=‘logo.gif’>
print “<IMG SRC=\”logo.gif\”>”;
// muestra <IMG SRC=“logo.gif”>
```

Agustín F. Calderón M.
agustincl@gmail.com

  - Acceso a un carácter de la cadena:

Variables

- Las variables siempre van precedidas de un $
- El nombre es sensible a las mayúsculas
- Comienzan por letra o subrayado, seguido de letras, números o subrayado
- Variables predefinidas:

  $GLOBALS, $_SERVER, $_GET, $_POST, $_COOKIES, $_FILES,

  $_ENV, $_REQUEST, $_SESSION
- Ámbito: globales al fichero (excepto funciones) o locales a una función
- Ejemplo:

```
$valor = 5;

print “El valor es: “ . $valor . “\n”;

print “El valor es: $valor\n”; // ojo: comillas dobles
```

Resultado:

El valor es: 5

Agustín F. Calderón M.
agustincl@gmail.com

Variables

- Variables variables
  - Se pueden crear nombres de variables dinámicamente
  - La variable variable toma su nombre del valor de otra variable previamente declarada
  - Ejemplo:

```
$a = "hola";

$$a = "mundo";


print "$a $hola\n";

print "$a ${$a}";
```

Resultado:

```
hola mundo

hola mundo
```

Agustín F. Calderón M.
agustincl@gmail.com

Variables

- Ejemplo de variables variables: página internacionalizada (1)

```
<?PHP
$mensaje_es="Hola";
$mensaje_en="Hello";
$idioma = "es";
$mensaje = "mensaje_" . $idioma;
print $$mensaje;
?>
```

Agustín F. Calderón M.
agustincl@gmail.com

Variables

- Ejemplo de variables variables: página internacionalizada (2)

```
<?PHP
$mensaje_es="Hola";
$mensaje_en="Hello";
$idioma = "en";
$mensaje = "mensaje_" . $idioma;
print $$mensaje;
?>
```

Agustín F. Calderón M.
agustincl@gmail.com

Constantes

- Definición de constantes:

  define (“CONSTANTE”, “hola”);

  print CONSTANTE;

- No llevan $ delante
- Sólo se pueden definir constantes de los tipos escalares (boolean, integer, double, string)

Agustín F. Calderón M.
agustincl@gmail.com

Expresiones y operadores

- Operadores aritméticos:
  +, -, *, /, %, ++, --
- Operador de asignación:
  =
  operadores combinados: .=, +=, etc
  $a = 3; $a += 5; □ a vale 8
  $b = “hola ”; $b .= “mundo”; □ b vale “hola mundo”
  □ Equivale a $b = $b . “mundo”;
- Operadores de comparación:
  ===, ==, !=, <, >, <=, >= y otros
- Operador de control de error: @. Antepuesto a una expresión, evita cualquier mensaje de error que pueda ser generado por la expresión
- Operadores lógicos:
  and (&&), or (||), !, xor
  and/&& y or/|| tienen diferentes prioridades
- Operadores de cadena:
  concatenación: . (punto)
  asignación con concatenación: .=

Agustín F. Calderón M.
agustincl@gmail.com

Expresiones y operadores

- Precedencia de operadores (de mayor a menor):

++, --

*, /, %

+,-

<, <=, >, >=

==, !=

&&

||

and

or

Agustín F. Calderón M.
agustincl@gmail.com

Estructuras de control

- Estructuras selectivas:
  - if-else
  - switch
- Estructuras repetitivas:
  - while
  - for
  - foreach

Agustín F. Calderón M.
agustincl@gmail.com

Estructuras de control

- Estructura selectiva **if-else**

```
if (condición)
sentencia

if (condición)
sentencia 1
else
sentencia 2
```

```
if (condición1)
sentencia 1
else if (condición2)
sentencia 2
...
else if (condición n)
sentencia n
else
sentencia n+1
```

- Mismo comportamiento que en C
- Las sentencias compuestas se encierran entre llaves
- elseif puede ir todo junto

Agustín F. Calderón M.
agustincl@gmail.com

Estructuras de control

- Ejemplo de estructura selectiva if-else:

```
<?PHP
if ($sexo == ‘M’)
$saludo = "Bienvenida, ";
else
$saludo = "Bienvenido, ";
$saludo = $saludo . $nombre;
print ($saludo);
?>
```

Agustín F. Calderón M.
agustincl@gmail.com

Estructuras de control

- Estructura selectiva **switch**

```
switch (expresión)
{
case valor_1:
sentencia 1
break;
case valor_2:
sentencia 2
break;
...
case valor_n:
sentencia n
break;
default
sentencia n+1
}
```

- Mismo comportamiento que en C, sólo que la expresión del case puede ser integer, float o string

Agustín F. Calderón M.
agustincl@gmail.com

Estructuras de control

- Ejemplo de estructura selectiva switch:

```
switch ($extension)
{
case ("PDF"):
    $tipo = "Documento Adobe PDF";
break;
case ("TXT"):
    $tipo = "Documento de texto";
break;
case ("HTML"):
case ("HTM"):
    $tipo = "Documento HTML";
break;
default:
    $tipo = "Archivo " . $extension;
}
print ($tipo);
```

Agustín F. Calderón M.
agustincl@gmail.com

Estructuras de control

- Estructura repetitiva **while**

  while (condición)

  sentencia

- Mismo comportamiento que en C

Agustín F. Calderón M.
agustincl@gmail.com

Estructuras de control

- Ejemplo de estructura repetitiva while:

```
<?PHP
print ("<UL>\n");
$i=1;
while ($i <= 5)
{
    print ("<LI>Elemento $i</LI>\n");
    $i++;
}
print ("</UL>\n");
?>
```

Agustín F. Calderón M.
agustincl@gmail.com

Estructuras de control

- Estructura repetitiva **for**

```
for (inicialización; condición; incremento)
sentencia
```

- Mismo comportamiento que en C

Agustín F. Calderón M.
agustincl@gmail.com

Estructuras de control

- Ejemplo de estructura repetitiva for:

```
<?PHP
print ("<UL>\n");
for ($i=1; $i<=5; $i++)
      print ("<LI>Elemento $i</LI>\n");
print ("</UL>\n");
?>
```

Agustín F. Calderón M.
agustincl@gmail.com

Estructuras de control

- **Ejercicio 2: programa que calcula una tabla de multiplicar**
  - Ilustra cómo manejar variables y cómo usar bucles

Agustín F. Calderón M.
agustincl@gmail.com

Funciones

- Ejemplo:

```
function suma ($x, $y)
{
$s = $x + $y;
return $s;
}


$a=1;
$b=2;
$c=suma ($a, $b);
print $c;
```

Agustín F. Calderón M.
agustincl@gmail.com

Funciones

- Por defecto los parámetros se pasan por valor
- Paso por referencia:

```
function incrementa (&$a)
{
$a = $a + 1;
}

$a=1;
incrementa ($a);
print $a; // Muestra un 2
```

Agustín F. Calderón M.
agustincl@gmail.com

Funciones

- Argumentos por defecto

```
function muestranombre ($titulo = "Sr.")
{
print "Estimado $titulo:\n";
}
muestranombre ();
muestranombre ("Prof.");
```

- Salida:

```
Estimado Sr.:
Estimado Prof.:
```

Agustín F. Calderón M.
agustincl@gmail.com

Funciones

- Los argumentos con valores por defecto deben ser siempre los últimos:

```
function muestranombre ($nombre, $titulo= "Sr.")
{
print "Estimado $titulo $nombre:\n";
}
muestranombre (“Fernández”);
muestranombre (“Fernández”, "Prof.");
```

- Salida:

```
Estimado Sr. Fernández:
Estimado Prof. Fernández:
```

Agustín F. Calderón M.
agustincl@gmail.com

Tablas

- Sintaxis:

  array ([clave =>] valor, ...)

- La clave es una cadena o un entero no negativo. El valor puede ser de cualquier tipo válido en PHP, incluyendo otro array
- Ejemplos:

  $color = array (‘rojo’=>101, ‘verde’=>51, ‘azul’=>255);

  $medidas = array (10, 25, 15);

- Acceso:

  $color[‘rojo’] // No olvidar las comillas

  $medidas[0]

- El primer elemento es el 0

Agustín F. Calderón M.
agustincl@gmail.com

Tablas

- La estructura de control **foreach** permite iterar sobre arrays
- Sintaxis:

```
foreach (expresión_array as $valor)
 sentencia

foreach (expresión_array as $clave => $valor)
 sentencia
```

- Ejemplos:

```
foreach ($color as $valor)

print “Valor: $valor<BR>\n”;

foreach ($color as $clave => $valor)

print “Clave: $clave; Valor: $valor<BR>\n”;
```

- Salida:

```
Valor: 101

Valor: 51

Valor: 255

Clave: rojo; Valor: 101

Clave: verde; Valor: 51

Clave: azul; Valor: 255
```

Agustín F. Calderón M.
agustincl@gmail.com

Bibliotecas de funciones

- Existen muchas bibliotecas de funciones en PHP
- Algunos ejemplos:
  - Funciones de manipulación de cadenas
  - Funciones de fecha y hora
  - Funciones de arrays
  - Funciones de ficheros
  - Funciones matemáticas
  - Funciones de bases de datos
  - Funciones de red
- Algunas bibliotecas requieren la instalación de componentes adicionales
- Todas las funciones de biblioteca están comentadas en la documentación de PHP

Agustín F. Calderón M.
agustincl@gmail.com

Bibliotecas de funciones

- Funciones de manipulación de cadenas
  - explode()
    - Divide una cadena en subcadenas
    - array **explode** (string separator, string string [, int limit])
  - rtrim(), ltrim(), trim()
    - Eliminan caracteres a la derecha, a la izquierda o por ambos lados de una cadena
    - string **rtrim** (string str [, string charlist])
  - strstr()
    - Busca la primera ocurrencia de una subcadena
  - strtolower() / strtoupper()
    - Convierte una cadena a minúscula / mayúscula
  - strcmp() / strcasecmp()
    - Compara dos cadenas con/sin distinción de mayúsculas
  - strlen()

Agustín F. Calderón M.
agustincl@gmail.com

Bibliotecas de funciones

- Funciones de fecha y hora
  - date()
    - Formatea una fecha según un formato dado
    - Ejemplo:

```
$fecha = date ("j/n/Y H:i");
print ("$fecha");
```

Resultado:

26/9/2005 17:36

  - strtotime()
    - Convierte una fecha en un *timestamp* de UNIX
    - Ejemplo:

```
$fecha = date ("j/n/Y", strtotime("5 april 2001"));
print ("$fecha");
```

Resultado:

5/4/2001

Agustín F. Calderón M.
agustincl@gmail.com

Bibliotecas de funciones

- Funciones de arrays
  - array_count_values()
    - Calcula la frecuencia de cada uno de los elementos de un array
  - array_search()
    - Busca un elemento en un array
  - count()
    - Cuenta los elementos de un array
  - sort(), rsort()
    - Ordena y reindexa un array (r=decreciente)
  - ksort(), krsort()
    - Ordena por claves un array (r=decreciente)

Agustín F. Calderón M.
agustincl@gmail.com

Bibliotecas de funciones

- **Ejercicio 3: programa que muestra la fecha actual**
  - Ilustra cómo usar comentarios, tablas y funciones (propias y de biblioteca). También cómo usar el manual de PHP

Agustín F. Calderón M.
agustincl@gmail.com

PHP y MySQL

Tema 3: Formularios

José Mariano González Romano
mariano@lsi.us.es

Agustín F. Calderón M.
agustincl@gmail.com

Tema 3: Formularios

1. Acceso a formularios HTML desde PHP
2. El formulario de PHP
3. Subida de ficheros al servidor
4. Validación de los datos de un formulario

Agustín F. Calderón M.
agustincl@gmail.com

Acceso a formularios desde PHP

- Desde PHP se puede acceder fácilmente a los datos introducidos desde un formulario HTML
- Veámoslo con un ejemplo simple

Agustín F. Calderón M.
agustincl@gmail.com

Acceso a formularios desde PHP

desde PHP

- Fichero uno.php

```
<HTML>
<BODY>
<FORM ACTION=”dos.php” METHOD=”POST”>
Edad: <INPUT TYPE=”text” NAME=”edad”>
<INPUT TYPE=”submit” VALUE=”aceptar”>
</FORM>
</BODY>
</HTML>
```

- Fichero dos.php

```
<HTML>
<BODY>
<?PHP
print (“La edad es: $edad”);
?>
</BODY>
</HTML>
```

Agustín F. Calderón M.
agustincl@gmail.com

Acceso a formularios desde PHP

Agustín F. Calderón M.
agustincl@gmail.com

Acceso a formularios desde PHP

- A partir de PHP 4.2.0, el valor por defecto de la directiva de PHP **register_globals** es off
- Esto tiene una gran importancia sobre los formularios, ya que no es posible acceder a las variables enviadas de la manera anterior (como variables globales). En su lugar hay que utilizar la variable predefinida de PHP **$_REQUEST**, escribiendo $_REQUEST['edad'] en lugar de $edad
- Se puede poner register_globals = on en el fichero de configuración php.ini, pero no es recomendable por motivos de seguridad. Una alternativa que permite hacer mínimos cambios en el código ya existente es la siguiente:

```
$edad = $_REQUEST['edad'];
```

Agustín F. Calderón M.
agustincl@gmail.com

Acceso a formularios desde PHP

desde PHP

- Fichero uno.php

```
<HTML>
<BODY>
<FORM ACTION=”dos.php” METHOD=”POST”>
Edad: <INPUT TYPE=”text” NAME=”edad”>
<INPUT TYPE=”submit” VALUE=”aceptar”>
</FORM>
</BODY>
</HTML>
```

- Fichero dos.php

```
<HTML>
<BODY>
<?PHP
$edad = $_REQUEST[‘edad’];
print (“La edad es: $edad”);
?>
</BODY>
</HTML>
```

Agustín F. Calderón M.
agustincl@gmail.com

Acceso a formularios desde PHP

- **Ejercicio 1: formulario simple**
  - Ilustra cómo acceder a los valores introducidos desde un formulario HTML

Agustín F. Calderón M.
agustincl@gmail.com

Acceso a formularios desde PHP

desde PHP

- Acceso a los diferentes tipos de elementos de entrada de formulario
  - Elementos de tipo INPUT
    - TEXT
    - RADIO
    - CHECKBOX
    - BUTTON
    - FILE
    - HIDDEN
    - PASSWORD
    - SUBMIT
  - Elemento SELECT
    - Simple / múltiple
  - Elemento TEXTAREA

Agustín F. Calderón M.
agustincl@gmail.com

Acceso a formularios desde PHP

- TEXT

Introduzca la cadena a buscar:
<INPUT TYPE="text" NAME="cadena" VALUE="valor por defecto" SIZE="20">

```
<?PHP
$cadena = $_REQUEST['cadena'];
print ($cadena);
?>
```

Agustín F. Calderón M.
agustincl@gmail.com

Acceso a formularios desde PHP

- RADIO

```
Sexo:
<INPUT TYPE="radio" NAME=“sexo" VALUE=“M“ CHECKED>Mujer
<INPUT TYPE="radio" NAME=“sexo" VALUE=“H">Hombre
```

```
<?PHP
$sexo = $_REQUEST[‘sexo’];
print ($sexo);
?>
```

Acceso a formularios desde PHP

```
<INPUT TYPE="checkbox" NAME="extras[]" VALUE="garaje" CHECKED>Garaje
<INPUT TYPE="checkbox" NAME="extras[]" VALUE="piscina">Piscina
<INPUT TYPE="checkbox" NAME="extras[]" VALUE="jardin">Jardín
```

- CHECKBOX

```
<?PHP
$extras = $_REQUEST['extras'];
foreach ($extras as $extra)
print ("$extra<BR>\n");
?>
```

Acceso a formularios desde PHP

- BUTTON

```
<INPUT TYPE="button" NAME="actualizar" VALUE="Actualizar datos">
```

```
<?PHP
$actualizar = $_REQUEST['actualizar'];
if ($actualizar)
print ("Se han actualizado los datos");
?>
```

Acceso a formularios desde PHP

- FILE

```
<FORM ACTION="procesa.php" METHOD="post“
ENCTYPE="multipart/form-data">
<INPUT TYPE="file" NAME="fichero">
</FORM>
```

Acceso a formularios desde PHP

- HIDDEN

```
<?PHP
print("<INPUT TYPE='hidden' NAME='username' VALUE='$usuario'>\n");
?>
```

```
<?PHP
$username = $_REQUEST['username'];
print ($username);
?>
```

Acceso a formularios desde PHP

Contraseña: <INPUT TYPE="password" NAME="clave">

- PASSWORD

```
<?PHP
$clave = $_REQUEST['clave'];
print ($clave);
?>
```

Acceso a formularios desde PHP

- SUBMIT

```
<INPUT TYPE="submit" NAME="enviar" VALUE="Enviar datos">
```

```
<?PHP
$enviar = $_REQUEST['enviar'];
if ($enviar)
print ("Se ha pulsado el botón de enviar");
?>
```

Acceso a formularios desde PHP

- SELECT simple

```
Color:
<SELECT NAME=“color">
<OPTION VALUE=“rojo" SELECTED>Rojo
<OPTION VALUE=“verde">Verde
<OPTION VALUE=“azul">Azul
</SELECT>
```

```
<?PHP
$color = $_REQUEST[‘color’];
print (
?
```

Acceso a formularios desde PHP

- SELECT múltiple

```
Idiomas:
<SELECT MULTIPLE SIZE="3" NAME="idiomas[]">
<OPTION VALUE="ingles" SELECTED>Inglés
<OPTION VALUE="frances">Francés
<OPTION VALUE="aleman">Alemán
<OPTION VALUE="holandes">Holandés
</SELECT>
```

```
<?PHP
$idiomas = $_REQUEST['idio
foreach ($idiomas as $idion
print ("$idioma<BR>\n");
?>
```

Acceso a formularios desde PHP

- TEXTAREA

```
Comentario:
<TEXTAREA COLS=“50" ROWS=“4" NAME="comentario">
Este libro me parece ...
</TEXTAREA>
```

```
<?PHP
$comentario = $_REQUEST[‘comentario’];
print ($comentario);
```

Acceso a formularios desde PHP

- **Ejercicio 2: programa que muestra los datos introducidos desde un formulario**
  - Ilustra cómo acceder a los valores introducidos desde todos los tipos de elementos de entrada de un formulario, con excepción de los tipos BUTTON y FILE, que se tratan en ejercicios posteriores

Agustín F. Calderón M.
agustincl@gmail.com

El formulario de PHP

- La forma habitual de trabajar con formularios en PHP es utilizar un único programa que procese el formulario o lo muestre según haya sido o no enviado, respectivamente
- Ventajas:
  - Disminuye el número de ficheros
  - Permite validar los datos del formulario en el propio formulario
- Procedimiento:

si se ha enviado el formulario:

Procesar formulario

si no:

Mostrar formulario

fsi

Agustín F. Calderón M.
agustincl@gmail.com

El formulario de PHP

- Esquema de funcionamiento:

si se ha enviado el formulario:
Procesar formulario

si no:
Mostrar formulario
fsi

- La 1ª vez que se carga la página se muestra el formulario (a)
- La 2ª vez se procesa el formulario (b)

Agustín F. Calderón M.
agustincl@gmail.com

El formulario de PHP

- Para saber si se ha enviado el formulario se acude a la variable correspondiente al botón de envío. Si este botón aparece de la siguiente forma en el formulario HTML:

```
<INPUT TYPE="SUBMIT" NAME="enviar“ VALUE="procesar">
```

entonces la condición anterior se transforma en:

```
if (isset($enviar))
```

o bien

```
if ($enviar == “procesar”)
```

Agustín F. Calderón M.
agustincl@gmail.com

El formulario de PHP

- **Ejercicio 3: formulario de PHP**
  - Ilustra cómo crear páginas multipropósito que sirven tanto para mostrar un formulario como para procesarlo

Agustín F. Calderón M.
agustincl@gmail.com

Subida de ficheros al servidor

- Para subir un fichero al servidor se utiliza el elemento de entrada FILE
- Hay que tener en cuenta una serie de consideraciones importantes:
  - El elemento FORM debe tener el atributo ENCTYPE="multipart/form-data“
  - El fichero tiene un límite en cuanto a su tamaño. Este límite se fija de dos formas diferentes:
    - En el fichero de configuración php.ini
    - En el propio formulario

Agustín F. Calderón M.
agustincl@gmail.com

Subida de ficheros al servidor

php.ini

```
;;;;;;;;;;;;;;;;;;
; File Uploads ;
;;;;;;;;;;;;;;;;;;
; Whether to allow HTTP file uploads.
file_uploads = On


; Temporary directory for HTTP uploaded files (will use
; system default if not specified).
;upload_tmp_dir =


; Maximum allowed size for uploaded files.
upload_max_filesize = 2M
```

formulario

```
<INPUT TYPE=”HIDDEN” NAME=”MAX_FILE_SIZE” VALUE='102400'>
<INPUT TYPE=”FILE” NAME="fichero">
```

Agustín F. Calderón M.
agustincl@gmail.com

Subida de ficheros al servidor

- Consideraciones (cont)
  - Debe darse al fichero un nombre que evite coincidencias con ficheros ya subidos. Por ello, y como norma general, debe **descartarse el nombre original** del fichero y crear uno nuevo que sea único
  - El fichero subido se almacena en un directorio temporal y hemos de moverlo al directorio de destino usando la función move_upload_file()
- Procedimiento:

si se ha subido correctamente el fichero:

Asignar un nombre al fichero

Mover el fichero a su ubicación definitiva

si no:

Mostrar un mensaje de error

fsi

Agustín F. Calderón M.
agustincl@gmail.com

Subida de ficheros al servidor

HTML

```
<INPUT TYPE="HIDDEN" NAME="MAX_FILE_SIZE" VALUE="102400">
<INPUT TYPE="FILE" SIZE="44" NAME="imagen">
```

- La variable $_FILES contiene toda la información del fichero subido:
  - $_FILES['imagen']['name']
    - Nombre original del fichero en la máquina cliente
  - $_FILES['imagen']['type']
    - Tipo mime del fichero. Por ejemplo, "image/gif"
  - $_FILES['imagen']['size']
    - Tamaño en bytes del fichero subido
  - $_FILES['imagen']['tmp_name']
    - Nombre del fichero temporal en el que se almacena el fichero subido en el servidor
  - $_FILES['imagen']['error']
    - Código de error asociado al fichero subido

Agustín F. Calderón M.
agustincl@gmail.com

Subida de ficheros al servidor

PHP

```
if (is_uploaded_file ($_FILES['imagen']['tmp_name']))
{
$nombreDirectorio = "img/";
$idUnico = time();
$nombreFichero = $idUnico . "-" . $_FILES['imagen']['name'];


move_uploaded_file ($_FILES['imagen']['tmp_name'],
$nombreDirectorio . $nombreFichero);
}
else
print ("No se ha podido subir el fichero\n");
```

Agustín F. Calderón M.
agustincl@gmail.com

Subida de ficheros al servidor

PHP

```
if (is_uploaded_file ($_FILES['imagen']['tmp_name']))
{
$nombreDirectorio = "img/";
$nombreFichero = $_FILES['imagen']['name'];


$nombreCompleto = $nombreDirectorio . $nombreFichero;
if (is_file($nombreCompleto))
{
$idUnico = time();
$nombreFichero = $idUnico . "-" . $nombreFichero;
}

move_uploaded_file ($_FILES['imagen']['tmp_name'],
$nombreDirectorio . $nombreFichero);
}
else
print ("No se ha podido subir el fichero\n");
```

Agustín F. Calderón M.
agustincl@gmail.com

Subida de ficheros al servidor

- **Ejercicio 4: subida de un fichero al servidor**
  - Ilustra cómo subir ficheros a un servidor, cómo controlar su tamaño, cómo crear un nombre único para el fichero y cómo almacenarlo en el lugar deseado

Agustín F. Calderón M.
agustincl@gmail.com

Validación de formularios

- Toda la información proveniente de un formulario debe considerarse por norma como contaminada, y hay que validarla antes de darla por buena y procesarla
- Lo más eficiente es mostrar los errores sobre el propio formulario para facilitar su corrección. Procedimiento:

si se ha enviado el formulario:

si hay errores:

Mostrar formulario con errores

si no:

Procesar formulario

fsi

si no:

Mostrar formulario

Agustín F. Calderón M.
agustincl@gmail.com

Validación de formularios

- Este procedimiento se puede resumir para que sólo haya que mostrar una vez el formulario, bien con los valores por defecto o con los valores introducidos, y con los errores en su caso:

si se ha enviado el formulario:

validar datos

fsi

si se ha enviado el formulario y no hay errores:

Procesar formulario

si no:

Mostrar formulario con valores por defecto o ya enviados

fsi

Agustín F. Calderón M.
agustincl@gmail.com

Validación de formularios

- Esquema de funcionamiento:

si se ha enviado el formulario:
validar datos
fsi

si se ha enviado y no hay errores:
Procesar formulario

si no:
Mostrar formulario
fsi

- La 1ª vez que se carga la página se muestra el formulario (a)
- La 2ª y sucesivas veces se validan los datos
- Si hay errores, se muestra de nuevo el formulario con los errores (b)
- Si no hay, se procesa el formulario (c)

Agustín F. Calderón M.
agustincl@gmail.com

Validación de formularios

- **Ejercicio 5: validación de los datos introducidos en un formulario**
  - Ilustra cómo realizar una validación de los datos introducidos en un formulario en el propio formulario

Agustín F. Calderón M.
agustincl@gmail.com

Desarrollo de sitios web con PHP y MySQL

Tema 4: Acceso a bases de datos MySQL en PHP

José Mariano González Romano
mariano@lsi.us.es

SOME RIGHTS RESERVED

Agustín F. Calderón M.
agustincl@gmail.com

Tema 4: Acceso a bases de datos MySQL en PHP

1. Bases de datos en la Web
2. Instalación y configuración de MySQL
3. MySQL
4. Herramientas de administración: phpMyAdmin
5. Lenguaje SQL
6. Funciones de PHP para el acceso a bases de datos MySQL
7. Ejercicios
8. Consulta avanzada de tablas

Agustín F. Calderón M.
agustincl@gmail.com

Bases de datos en la Web

- Las bases de datos permiten almacenar de una forma estructurada y eficiente toda la información de un sitio web
- Ventajas
- Proporcionar información actualizada
- Facilitar la realización de búsquedas
- Disminuir los costes de mantenimiento
- Implementar sistemas de control de acceso
- Almacenar preferencias de los usuarios

Agustín F. Calderón M.
agustincl@gmail.com

Bases de datos en la Web

- Esquema básico de un sitio web soportado por bases de datos:

Instalación y config. de MySQL

- Pasos:
  - Descargar
  - Descomprimir e instalar
  - Configurar
  - Arrancar
  - Conectar con el servidor
  - Instalar la extensión para MySQL de PHP

A n M.
agustincl@gmail.com

Instalación y config. de MySQL

- **Instalación y configuración de MySQL. 1: descargar**
  - Conectarse a la página web de MySQL, www.mysql.com
  - Seleccionar la pestaña *Developer Zone* y dentro de ella la sección *Downloads*
  - Elegir la versión estable más reciente de *MySQL Community Server*. A fecha de febrero de 2008 es la 5.0
  - Seleccionar la plataforma Windows y descargar el archivo comprimido a una carpeta temporal

Instalación y config. de MySQL

- **Instalación y configuración de MySQL. 2: descomprimir e instalar**
  - Descomprimir el archivo descargado en una carpeta temporal
  - Ejecutar el archivo setup.exe y seguir las indicaciones:
    - Seleccionar instalación típica
    - Aceptar la carpeta de instalación por defecto
    - Saltar el formulario de registro
  - Pulsar el botón *Finish* para terminar la instalación y pasar a la configuración del servidor

A n M.
agustincl@gmail.com

Instalación y config. de MySQL

- **Instalación y configuración de MySQL. 3: configurar**
  - Al finalizar la instalación se ejecuta el asistente para la configuración del servidor. Seguir sus indicaciones:
    - Elegir la configuración estándar
    - Seleccionar la instalación como servicio Windows y Marcar la casilla para lanzar el servidor automáticamente al arrancar el sistema
    - Establecer una contraseña para el administrador (root)
    - Pulsar el botón *Execute* para realizar la configuración
    - Pulsar el botón *Finish* para finalizar el asistente

A n M.
agustincl@gmail.com

Instalación y config. de MySQL

- **Instalación y configuración de MySQL. 4: arrancar**
  - Si se indicó el arranque automático en la configuración, el servidor se inicia de forma automática al arrancar el sistema
  - En caso contrario hay que iniciarlo manualmente con Inicio > Programas > MySQL > MySQL Server 5.0 > MySQL Server Instance Config Wizard

A n M.
agustincl@gmail.com

Instalación y config. de MySQL

- **Instalación y configuración de MySQL. 5: conectar**
  - Formas de establecer la conexión con el servidor:
    - Desde la línea de órdenes con Inicio > Programas > MySQL > MySQL Server 5.0 > MySQL Command Line Client

```
Enter password: ********

mysql>

...

mysql> exit
```

    - Mediante alguna herramienta que proporcione una inter[illegible]o phpMyAdmin
    - Desde una página web mediante la interfaz que proporciona MySQL. Es lo

A[illegible]n M.
agustincl@gmail.com

Instalación y config. de MySQL

- **Instalación y configuración de MySQL. 5: conectar**
  - Para conectar con el servidor hay que crear antes una cuenta de usuario y asignarle los correspondientes permisos de acceso. En general para una base de datos es conveniente definir al menos dos usuarios:
    - Un usuario anónimo que tenga permisos de lectura sobre las tablas que se estime adecuado
    - Un usuario administrador que tenga permisos para insertar, modificar o eliminar elementos de las tablas de la base de datos
  - En cada conexión hay que indicar el nombre del usuario, su contraseña y la máquina desde la que se realiza la conexión (localhost si es la propia máquina donde reside el servidor, que es lo habitual en el acceso desde la Web)

A M.
agustincl@gmail.com

Instalación y config. de MySQL

- **Instalación y configuración de MySQL. 6: MySQL-PHP**
  - En PHP 5 no se instala por defecto la extensión para el acceso a bases de datos MySQL y hay que hacerlo manualmente
  - Se hace de la siguiente manera:
    - Editar el fichero php.ini y habilitar la extensión:

```
extension=php_mysql.dll
```

    - Añadir la ruta c:\php al PATH del sistema (ver las instrucciones para hacerlo en el fichero install.txt que hay en la carpeta c:

A n M.
agustincl@gmail.com

MySQL

- **Características de MySQL**
  - Modelo relacional, multiusuario
- **Tipos de datos**
  - Numéricos
    - tinyint, smallint, mediumint, int, integer, bigint
    - decimal, float, numeric
  - Fecha y hora
    - date, time, datetime, year, timestamp
  - Cadena
    - char, varchar
    - tinytext, text, mediumtext, longtext
    - tinyblob, blob, mediumblob, longblob
    - enum, set

Agustín F. Calderón M.
agustincl@gmail.com

MySQL

- **Operadores**
  - Aritméticos
    - +, -, *, /
  - Comparación
    - =, !=, <=, <, >=, >, IS NULL, IS NOT NULL
  - Lógicos
    - not (!), and (&&), or (||), xor
- **Funciones**
  - Funciones de cadena
  - Funciones de comparación de cadenas
  - Funciones numéricas
  - Funciones de fecha y hora

Agustín F. Calderón M.
agustincl@gmail.com

Herramientas de administración: phpMyAdmin

- **phpMyAdmin** es una herramienta para la administración del servidor de bases de datos MySQL
- Dispone de una interfaz gráfica y es de libre distribución
- Permite realizar todo tipo de operaciones sobre bases de datos:
  - crear, borrar y modificar tablas
  - consultar, insertar, modificar y eliminar datos
  - definir usuarios y asignar permisos
  - realizar copias de seguridad
  - etc
- Está escrita en php y se ejecuta desde el navegador
- Si está instalada en la carpeta phpmyadmin, se ejecuta escribiendo en la barra de direcciones del navegador la url

http://localhost/phpmyadmin/

Agustín F. Calderón M.
agustinfc@gmail.com

phpMyAdmin

lindavista ejecutándose en localhost - phpMyAdmin 2.3.2 - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

Atrás Búsqueda Favoritos Multimedia Vínculos

Página de inicio

lindavista (4)

lindavista
- noticias
- usuarios
- viviendas
- votos

Base de datos *lindavista* ejecutándose en *localhost*

Estructura | SQL | Exportar | Buscar | Consulta de ejemplo | Eliminar

	Tabla	Acción						Campos	Tipo	Ta
☐	noticias	Examinar	Seleccionar	Insertar	Propiedades	Eliminar	Vaciar	5	MyISAM	2
☐	usuarios	Examinar	Seleccionar	Insertar	Propiedades	Eliminar	Vaciar	1	MyISAM	2
☐	viviendas	Examinar	Seleccionar	Insertar	Propiedades	Eliminar	Vaciar	8	MyISAM	3
☐	votos	Examinar	Seleccionar	Insertar	Propiedades	Eliminar	Vaciar	1	MyISAM	2
	4 tabla(s)	Tamaño de las tablas						15	--	9

Revisar todos/as / Desmarcar todos Con marca:

- Vista de impresión
- Crear nueva tabla en base de datos lindavista :

Nombre :

Campos : Continúe

Intranet local

phpMyAdmin

- Pasos para su instalación:
  - Descargar
  - Descomprimir
  - Configurar
  - Ejecutar

Agustín F. Calderón M.
agustincl@gmail.com

phpMyAdmin

- **Instalación y configuración de phpMyAdmin. 1: descargar**
  - Conectarse a la dirección http://www.phpmyadmin.net
  - Seleccionar *Downloads*
  - Seleccionar la versión más reciente. A fecha de febrero de 2008 es la 2.11.4

phpMyAdmin

- **Instalación y configuración de phpMyAdmin. 2: descomprimir**
  - Descomprimir debajo de la carpeta raíz de la web, c:\htdocs
  - Cambiar el nombre de la carpeta creada a phpmyadmin

phpMyAdmin

- **Instalación y configuración de phpMyAdmin. 3: configurar**
  - El fichero de configuración se llama config.inc.php
  - Este fichero no existe sino que hay que crearlo. Para ello se hace una copia del fichero config.default.php, situado en la carpeta donde se haya instalado phpMyAdmin, y se le pone como nombre config.inc.php
  - Seguidamente se modifica el fichero config.inc.php
  - Configuración típica para un servidor local:

```
...
$cfg['Servers'][$i]['host'] = 'localhost'; //MySQL hostname
$cfg['Servers'][$i]['user'] = 'root'; //MySQL user
$cfg['Servers'][$i]['password'] = 'clave'; //MySQL password
...
```

Ag

siendo ‘clave’ la contraseña asignada al administrador (root) de MySQL

phpMyAdmin

- **Instalación y configuración de phpMyAdmin. 4: ejecutar**
  - Ejecutar Apache
  - Abrir el navegador y teclear la url http://localhost/phpmyadmin/index.php

Lenguaje SQL

- SQL (*Structured Query Language*) es el lenguaje que se utiliza para comunicarse con la base de datos
- Procedimient[illegible]unicación con la [illegible] datos:

Agustín F. Calderón M.
agustincl@gmail.com

Lenguaje SQL

- Las instrucciones más habituales son SELECT, INSERT, UPDATE, DELETE
- Veamos su sintaxis básica y algunos ejemplos de uso
- Para ello utilizaremos una tabla **noticias** con cinco campos: un identificador único de la noticia, el título de la noticia, el texto de la noticia, la categoría de la noticia y la fecha de publicación de la noticia

Agustín F. Calderón M.
agustincl@gmail.com

Lenguaje SQL

- SELECT

Sintaxis:

SELECT expresión FROM tabla
[WHERE condición]
[ORDER BY {unsigned_integer | col_name | formula} [ASC | DESC] ,...]
[LIMIT [offset,] row_count | row_count OFFSET offset]

Ejemplo:

SELECT * from noticias WHERE fecha=CURDATE() LIMIT 10 ORDER BY fecha DESC

Obtiene las noticias del día con un tope máximo de 10, ordenadas de la más reciente a la más antigua

Agustín F. Calderón M.
agustincl@gmail.com

Lenguaje SQL

- INSERT

Sintaxis:

INSERT [INTO] nombre_tabla [(nombre_columna,...)]
VALUES ((expresión | DEFAULT),...), (...),...
INSERT [INTO] nombre_tabla
SET nombre_columna=(expresión | DEFAULT), ...

Ejemplo:

INSERT INTO noticias (id, titulo, texto, categoria, fecha) VALUES (37, “Nueva promoción en Nervión”, “145 viviendas de lujo en urbanización ajardinada situadas en un entorno privilegiado”, “promociones”, CURDATE())

Inserta una noticia con los valores indicados

Agustín F. Calderón M.
agustincl@gmail.com

Lenguaje SQL

- UPDATE

Sintaxis:

UPDATE nombre_tabla
SET nombre_columna1=expr1 [, nombre_columna2=expr2 ...]
[WHERE condición]
[ORDER BY ...]
[LIMIT row_count]

Ejemplo:

UPDATE noticias SET categoria = “ofertas” WHERE id=37

Modifica la categoría de la noticia con id=37 de la tabla

Agustín F. Calderón M.
agustincl@gmail.com

Lenguaje SQL

- DELETE

Sintaxis:

DELETE FROM nombre_tabla
[WHERE condición]
[ORDER BY ...]
[LIMIT row_count]

Ejemplo:

DELETE FROM noticias WHERE fecha < CURDATE()-10

Borra las noticias con más de 10 días de antigüedad

Agustín F. Calderón M.
agustincl@gmail.com

Funciones de PHP para el acceso a bases de datos MySQL

- Los pasos para acceder desde PHP a una base de datos son los siguientes:
  - Conectar con el servidor de bases de datos
  - Seleccionar una base de datos
  - Enviar la instrucción SQL a la base de datos
  - Obtener y procesar los resultados
  - Cerrar la conexión con el servidor de bases de datos

Agustín F. Calderón M.
agustincl@gmail.com

Acceso a bases de datos MySQL

- Las funciones concretas de MySQL que realizan estas operaciones son:
  - Conectar con el servidor de bases de datos:
    - **mysql_connect()**
  - Seleccionar una base de datos:
    - **mysql_select_db()**
  - Enviar la instrucción SQL a la base de datos:
    - **mysql_query()**
  - Obtener y procesar los resultados:
    - **mysql_num_rows() y mysql_fetch_array()**
  - Cerrar la conexión con el servidor de bases de datos:
    - **mysql_close()**

Agustín F. Calderón M.
agustincl@gmail.com

Acceso a bases de datos MySQL

- Conectar con el servidor de bases de datos: **mysql_connect()**
  - Devuelve un identificador de la conexión en caso de éxito y false en caso contrario

- Sintaxis:

```
$conexion = mysql_connect (servidor, username, password);
```

- Ejemplo:

```
$conexion = mysql_connect ("localhost", "cursophp", "")
or die ("No se puede conectar con el servidor");
```

```
$conexion = mysql_connect ("localhost", "cursophp-ad", "php.hph")
or die ("No se puede conectar con el servidor");
```

Agustín F. Calderón M.
agustincl@gmail.com

Acceso a bases de datos MySQL

- Seleccionar una base de datos: **mysql_select_db()**
  - Devuelve true en caso de éxito y false en caso contrario
- Sintaxis:

```
mysql_select_db (database);
```

- Ejemplo:

```
mysql_select_db (“lindavista”)
or die (“No se puede seleccionar la base de datos”);
```

Agustín F. Calderón M.
agustincl@gmail.com

Acceso a bases de datos MySQL

- Enviar la instrucción SQL a la base de datos: **mysql_query()**
  - Devuelve un identificador o true (dependiendo de la instrucción) si la instrucción se ejecuta correctamente y false en caso contrario
- Sintaxis:

```
$consulta = mysql_query (instrucción, $conexion);
```

- Ejemplo:

```
$consulta = mysql_query (“select * from noticias”, $conexion)
or die (“Fallo en la consulta”);
```

Agustín F. Calderón M.
agustincl@gmail.com

Acceso a bases de datos MySQL

- Obtener y procesar los resultados: **mysql_num_rows(), mysql_fetch_array()**
  - En el caso de que la instrucción enviada produzca unos resultados, mysql_query() devuelve las filas de la tabla afectadas por la instrucción
  - mysql_num_rows() devuelve el número de filas afectadas
  - Para obtener las distintas filas del resultado se utiliza la función mysql_fetch_array(), que obtiene una fila del resultado en un array asociativo cada vez que se invoca
- Sintaxis:

```
$nfilas = mysql_num_rows ($consulta);
$fila = mysql_fetch_array ($consulta);
```

Agustín F. Calderón M.
agustincl@gmail.com

Acceso a bases de datos MySQL

- Ejemplo:

noticias				
1	Título 1	Texto 1	ofertas	05/02/2004
2	Título 2	Texto 2	promociones	05/02/2004
3	Título 3	Texto 3	promociones	04/02/2004
4	Título 4	Texto 4	costas	01/02/2004
5	Título 5	Texto 5	promociones	31/01/2004

Instrucción:

select * from noticias where categoria=“promociones”

Agustín F. Calderón M.
agustincl@gmail.com

Acceso a bases de datos MySQL

- Ejemplo:

noticias				
1	Título 1	Texto 1	ofertas	05/02/2004
2	Título 2	Texto 2	promociones	05/02/2004
3	Título 3	Texto 3	promociones	04/02/2004
4	Título 4	Texto 4	costas	01/02/2004
5	Título 5	Texto 5	promociones	31/01/2004

$consulta

$nfilas=3

Instrucción:

select * from noticias where categoria="promociones"

Agustín F. Calderón M.
agustincl@gmail.com

Acceso a bases de datos MySQL

- Obtención de las filas:

```
$nfilas = mysql_num_rows ($consulta);
if ($nfilas > 0)
{
for ($i=0; $i<$nfilas; $i++)
{
$fila = mysql_fetch_array ($consulta);
procesar fila i-ésima de los resultados
}
}
```

Agustín F. Calderón M.
agustincl@gmail.com

Acceso a bases de datos MySQL

- Obtener los resultados: **mysql_num_rows(), mysql_fetch_array()**
  - Para acceder a un campo determinado de una fila se usa la siguiente sintaxis:

```
$fila["nombre_campo"] // por ser un array asociativo

$fila[$i] // $i=índice del campo desde 0
```

- Ejemplo:

```
for ($i=0; $i<$nfilas; $i++)
{
$fila = mysql_fetch_array ($consulta);
print "Título: " . $fila["titulo"];
print "Fecha: " . $fila["fecha"];
}
```

Agustín F. Calderón M.
agustincl@gmail.com

Acceso a bases de datos MySQL

- Cerrar la conexión con el servidor de bases de datos: **mysql_close()**
- Sintaxis:

```
mysql_close ($conexion);
```

- Ejemplo

```
mysql_close ($conexion);
```

Agustín F. Calderón M.
agustincl@gmail.com

Ejercicios

- **Ejercicio 1: consulta de noticias**
  - Ejercicio simple de acceso a una tabla de una base de datos. Ilustra cómo conectar con una base de datos, enviar una consulta, recuperar los resultados y mostrarlos en pantalla.
  - Pasos previos:
    1. Crear la base de datos lindavista
    2. Crear la tabla noticias en la base de datos lindavista
    3. Crear el usuario ‘cursophp’ (sin contraseña)
    4. Dar al usuario ‘cursophp’ permiso SELECT sobre la tabla noticias de la base de datos lindavista

Agustín F. Calderón M.
agustincl@gmail.com

Ejercicios

- **Ejercicio 2: encuesta**
  - Ilustra cómo actualizar una tabla y cómo generar gráficos simples
  - Pasos previos:
    1. Crear la tabla votos en la base de datos lindavista
    2. Dar al usuario ‘cursophp’ permisos SELECT y UPDATE sobre la tabla votos de la base de datos lindavista

Agustín F. Calderón M.
agustincl@gmail.com

Ejercicios

- **Ejercicio 3: inserción de noticias**
  - Se creará un formulario para insertar las noticias en la base de datos
  - Pasos previos:
    1. Crear el usuario cursophp-ad (contraseña: php.hph)
    2. Dar al usuario ‘cursophp-ad’ permisos SELECT e INSERT sobre la tabla noticias de la base de datos lindavista

Agustín F. Calderón M.
agustincl@gmail.com

Ejercicios

- **Ejercicio 4: eliminación de noticias**
  - Se creará un formulario para eliminar noticias de la base de datos
  - Primero se seleccionarán las noticias a eliminar de entre todas las existentes y a continuación se eliminarán
  - Paso previo:
    1. Dar al usuario ‘cursophp-ad’ permiso DELETE sobre la tabla noticias de la base de datos lindavista

Agustín F. Calderón M.
agustincl@gmail.com

Consulta avanzada de tablas

- Objetivo:
  - mostrar los resultados de la consulta **divididos en bloques de un número determinado de elementos** (por ejemplo, de 5 en 5)
- Requisitos:
  - Recuperar un número limitado de elementos de la tabla
  - Implementar un mecanismo de navegación que permita avanzar al siguiente o volver al anterior bloque de elementos

Agustín F. Calderón M.
agustincl@gmail.com

Consulta avanzada de tablas

- Para recuperar un número fijo de elementos de una tabla se utiliza la opción **LIMIT** de la orden SELECT. Así, por ejemplo,

```
SELECT * from noticias LIMIT 0, 5
```

recupera los 5 primeros elementos de la tabla. Y en general,

```
SELECT * from noticias LIMIT $comienzo, $num
```

recupera $num elementos a partir de la posición $comienzo

Agustín F. Calderón M.
agustincl@gmail.com

Consulta avanzada de tablas

- La variable $num tendrá un valor constante (en este caso 5), mientras que la variable $comienzo se incrementará o decrementará en 5 unidades al pasar a la página siguiente o anterior
- Para ello se pasará la variable como parámetro en el enlace asociado al botón correspondiente
- Por ejemplo, el código para el botón siguiente será:

```
"<A HREF='" . $_SERVER['PHP_SELF'] . "?comienzo=" . ($comienzo + $num) .
"'>Siguiente</A>"
```

- Habrá que comprobar previamente que el nuevo valor de comienzo es válido, es decir, que se encuentra dentro de los límites de la tabla devuelta por la consulta

Agustín Fernández M.
agustincl@gmail.com

Consulta avanzada de tablas

- **Ejercicio 5: consulta de noticias con paginado de resultados**
  - Ejercicio que ilustra cómo dividir los resultados de una consulta en varias páginas y permitir desplazarse entre ellas

Agustín F. Calderón M.
agustincl@gmail.com

Consulta avanzada de tablas

- Objetivo:
  - Mostrar los resultados de una consulta de manera que se puedan **filtrar en función del valor de una determinada columna** de la tabla
- Requisitos:
  - Recuperar de una tabla los elementos que cumplan una determinada condición
  - Permitir seleccionar un valor de entre los valores posibles de una columna

Agustín F. Calderón M.
agustincl@gmail.com

Consulta avanzada de tablas

- Para recuperar los elementos de una tabla que cumplen una condición se utiliza la opción **WHERE** de la orden SELECT. Por ejemplo,

```
SELECT * from noticias WHERE categoria='ofertas'
```

recupera las noticias cuya categoría tiene el valor “ofertas”. Y en general,

```
SELECT * from noticias WHERE categoria ='$categoria'
```

recupera las noticias cuya categoría tiene el valor dado por la variable $categoria

Agustín F. Calderón M.
agustincl@gmail.com

Consulta avanzada de tablas

- **Ejercicio 6: consulta de noticias con filtrado de resultados**
  - Ejercicio que ilustra cómo mostrar los resultados de una consulta filtrados en función del valor de una columna de tipo enumerado con actualización manual al pulsar un botón

Agustín F. Calderón M.
agustincl@gmail.com

Consulta avanzada de tablas

- Objetivo:
  - Modificar el ejercicio anterior para que los resultados se actualicen de forma automática al seleccionar un nuevo valor para la columna
- Requisitos:
  - Obtener automáticamente los valores de una columna de tipo enumerado a partir de la tabla
  - Utilizar código JavaScript para detectar un cambio en la opción seleccionada de un elemento de tipo SELECT

Agustín F. Calderón M.
agustincl@gmail.com

Consulta avanzada de tablas

- Función JavaScript que actualiza una página en función de la opción seleccionada en el elemento SELECT de nombre ‘categoria’ del formulario de nombre ‘selecciona’:

```
<SCRIPT LANGUAGE='JavaScript'>
<!--
function actualizaPagina ()
{
i = document.forms.selecciona.categoria.selectedIndex;
categoria =
document.forms.selecciona.categoria.options[i].value;
window.location = ‘muestra_noticias.php?categoria=' +
categoria;
}
```

Agustín F. Calderón M.
agustincl@gmail.com

Consulta avanzada de tablas

- Para ejecutar la función JavaScript es preciso asociarla al elemento SELECT a través del evento ONCHANGE, que se activa cuando se modifica la opción seleccionada:

```
<SELECT NAME='categoria' ONCHANGE='actualizaPagina()'>
```

Agustín F. Calderón M.
agustincl@gmail.com

Consulta avanzada de tablas

- Es conveniente que la opción seleccionada aparezca marcada por defecto al actualizar la página
- Para ello es preciso generar las opciones del elemento SELECT de forma automática
- De esta manera se puede comparar cada una de las opciones con el valor proveniente del formulario y colocar el atributo SELECTED a la que corresponda

Agustín F. Calderón M.
agustincl@gmail.com

Consulta avanzada de tablas

- En el formulario del ejercicio anterior los valores de los campos de tipo enumerado (ENUM) están escritos directamente en el propio código. Así, para la categoría de noticia se tiene:

```
<SELECT NAME=“categoria">
<OPTION VALUE=“promociones">promociones
<OPTION VALUE=“ofertas" SELECTED>ofertas
<OPTION VALUE=“costas">costas
</SELECT>
```

- Este hecho no es muy deseable ya que si se modifican los valores de la categoría en la tabla hay que modificar también el código del formulario

Agustín F. Calderón M.
agustinc@gmail.com

Consulta avanzada de tablas

- Lo ideal es tomar los valores automáticamente de la tabla en lugar de escribirlos ‘a mano’ en el código. Para ello se utiliza la siguiente instrucción SQL:

SHOW columns FROM noticias LIKE ‘categoria‘

que devuelve una tabla con las propiedades del campo ‘categoria’. El elemento [1] de esta tabla contiene lo siguiente:

enum(“promociones",“ofertas",“costas")

- A partir de esta información podemos obtener los valores del tipo enumerado y generar las opciones del elemento SELECT. Sólo queda marcar con SELECTED la opción que estaba seleccionada

Agustín F. Calderón M.
agustinc@gmail.com

Consulta avanzada de tablas

- **Ejercicio 7: consulta de noticias con filtrado de resultados**
  - Ejercicio que ilustra cómo mostrar los resultados de una consulta filtrados en función del valor de una columna de tipo enumerado con actualización automática a través de JavaScript

Agustín F. Calderón M.
agustincl@gmail.com

Desarrollo de sitios web con PHP y MySQL

Tema 5: Sesiones

José Mariano González Romano
mariano@lsi.us.es

Agustín F. Calderón M.
agustincl@gmail.com

Tema 5: Sesiones

1. Introducción
2. Manejo de sesiones
3. Autenticación de usuarios

Agustín F. Calderón M.
agustincl@gmail.com

Introducción

- A veces es necesario mantener el estado de una conexión entre distintas páginas o entre distintas visitas a un mismo sitio
  - Ejemplos: aplicaciones personalizadas, carrito de la compra, control de acceso
- HTTP es un **protocolo sin estado**: cada conexión entre el cliente y el servidor es independiente de las demás
- Para mantener el estado entre diferentes conexiones hay que establecer lo que se conoce como una **sesión**
- Las sesiones permiten disponer de unas variables con valores persistentes durante toda la conexión del usuario. Estas variables pueden almacenarse en el cliente mediante *cookies* o en el servidor
- PHP dispone de una biblioteca de funciones para la gestión de sesiones

Manejo de sesiones

- Funciones de PHP para el manejo de sesiones (register_globals On)
  - **session_start** ()
    - inicializa una sesión y le asigna un identificador de sesión único. Si la sesión ya está iniciada, carga todas las variables de sesión
  - **session_register** (*variable*)
    - registra una variable de sesión
  - **session_unregister** (*variable*)
    - elimina una variable de sesión
  - **session_is_registered** (*variable*)
    - comprueba si una variable está registrada. Devuelve *true* en caso afirmativo y *false* en caso contrario
  - **session_destroy** ()

Agustín F. Calderón M.
agustincl@gmail.com

Manejo de sesiones

- Funciones de PHP para el manejo de sesiones (register_globals Off)
  - **session_start** ()
    - inicializa una sesión y le asigna un identificador de sesión único. Si la sesión ya está iniciada, carga todas las variables de sesión
  - $_SESSION[‘nombre’] = valor;
    - registra una variable de sesión
  - unset ($_SESSION[‘nombre’]);
    - elimina una variable de sesión
  - if (isset($_SESSION[‘nombre’]))
    - comprueba si una variable está registrada. Devuelve *true* en caso afirmativo y *false* en caso contrario
  - **session_destroy** ()

Agustín F. Calderón M.
agustincl@gmail.com

Manejo de sesiones

- El manejo de las sesiones se realiza de la siguiente forma:
  - Todas las páginas deben realizar una llamada a session_start() para cargar las variables de la sesión
  - Esta llamada debe estar colocada antes de cualquier código HTML
  - Conviene llamar a session_destroy() para cerrar la sesión

Agustín F. Calderón M.
agustincl@gmail.com

Manejo de sesiones

- **Ejercicio 1: ejemplo simple de sesiones**
  - Ilustra cómo registrar variables, acceder a ellas y posteriormente eliminarlas

Agustín F. Calderón M.
agustincl@gmail.com

Autenticación de usuarios

- Una cuestión frecuente en un sitio web es controlar el acceso de los usuarios a una zona determinada del mismo
- La autenticación de usuarios puede realizarse en el propio servidor web. Así, en Apache los ficheros .htaccess permiten limitar el acceso a un determinado recurso del servidor
- Una alternativa más compleja pero más flexible es utilizar PHP junto con una base de datos para controlar el acceso de los usuarios. Para ello se utilizan las sesiones

Agustín F. Calderón M.
agustincl@gmail.com

Autenticación de usuarios

- Esquema de una página que utiliza sesiones para autenticar usuarios:

```
<?PHP
session_start ();
?>
<HTML LANG="es">
<HEAD> ... </HEAD>
<BODY>
<?PHP
if (isset($_SESSION["usuario_valido"]))
// Código para usuarios autorizados
else
// Mensaje de acceso no autorizado
?>
</BODY>
</HTML>
```

Agustín F. Calderón M.
agustincl@gmail.com

Autenticación de usuarios

- **Ejercicio 2: autenticación de usuarios**
  - Para nuestro sistema de gestión de noticias se va a restringir el acceso a las operaciones a unos usuarios identificados por un nombre y una contraseña
  - La información de los usuarios autorizados se almacenará en una tabla de la base de datos
  - Las contraseñas de los usuarios se almacenarán en forma encriptada
  - Esquema:

Agustín F. Calderón M.
agustincl@gmail.com

Autenticación de usuarios

Agustín F. Calderón M.
agustincl@gmail.com

Autenticación de usuarios

erón M.
l.com

Desarrollo de sitios web con PHP y MySQL

Tema 6: Creación de imágenes

José Mariano González Romano
mariano@lsi.us.es

Agustín F. Calderón M.
agustincl@gmail.com

Creación de imágenes

1. Introducción
2. Imágenes dinámicas en PHP
3. Funciones de biblioteca para la creación de imágenes
4. Ejercicios

Agustín F. Calderón M.
agustincl@gmail.com

Introducción

- PHP dispone de una biblioteca de funciones para la generación dinámica de imágenes
- Estas funciones son particularmente interesantes para generar gráficos estadísticos actualizados al instante
- Formatos: GIF, PNG, JPEG
- Configuración:
  - Requiere el uso de la librería php_gd2.dll (Windows) o gd.so (Unix, Linux)
  - Debe activarse la extensión correspondiente en el fichero php.ini:

```
extension = php_gd2.dll

extension = gd.so
```

Agustín F. Calderón M.
agustincl@gmail.com

Imágenes dinámicas en PHP

imagen.php

```
<?PHP
header (“Content-type: image/png”);

$imagen = imagecreate (300, 200); // ancho x alto

// instrucciones gráficas

imagepng ($imagen);
imagedestroy ($imagen);
?>
```

llamada

```
<IMG SRC=“imagen.php”>
```

Agustín F. Calderón M.
agustincl@gmail.com

Imágenes dinámicas en PHP

(ancho,alto)

Agustín F. Calderón M.
agustincl@gmail.com

Funciones de biblioteca

- Colores

  $color = imagecolorallocate ($imagen, R, G, B)

- Formas geométricas

  imagerectangle ($imagen, x0, y0, x1, y1, $color)

  imagefilledrectangle

  imagearc ($imagen, xc, yc, ancho, alto, gi, gf, $color)

  imagefilledarc

Agustín F. Calderón M.
agustincl@gmail.com

Funciones de biblioteca

- Fondos

  $imagen = imagecreatefrompng (“imagen.png”)

  $imagen = imagecreatefromjpeg (“imagen.jpg”)

- Textos

  imagestring ($imagen, tamaño, x, y, $texto, $color)

- Guardar imágenes

  imagepng ($imagen, “imagen.png”);

  imagejpeg ($imagen, “imagen.jpg”);

Agustín F. Calderón M.
agustincl@gmail.com

Ejercicios

- **Ejercicio 1: creación de imágenes simples**
  - Ilustra cómo crear un gráfico de tarta

Agustín F. Calderón M.
agustincl@gmail.com

Ejercicios

- **Ejercicio 2: encuesta**
  - Ilustra cómo utilizar un gráfico de tarta para mostrar los resultados de una encuesta

Agustín F. Calderón M.
agustincl@gmail.com