

# **CAPÍTULO 13**

PROGRAMACIÓN DE DOCUMENTOS WEB  
UTILIZANDO LENGUAJES DE SCRIPT DE  
SERVIDOR: PHP

# OBJETIVOS

- Conocer la clasificación de los lenguajes de programación.
- Presentar algunas de las herramientas para la edición de código.
- Estudiar el lenguaje de programación web orientado al servidor más conocido y utilizado en la actualidad: PHP.
- Realizar ejemplos y prácticas sobre dicho lenguaje.

# ÍNDICE

- 13.1 CLASIFICACIÓN
- 13.2 HERRAMIENTAS DE EDICIÓN DE CÓDIGO
- 13.3 PHP
- 13.4 INTEGRACIÓN CON LOS LENGUAJES DE MARCAS. SINTAXIS
- 13.5 COMENTARIOS
- 13.6 ELEMENTOS DEL LENGUAJE ESTRUCTURADO: TIPOS DE DATOS, VARIABLES, OPERADORES, ESTRUCTURAS DE CONTROL, SUBPROGRAMAS...
- 13.7 OPERADORES ARITMÉTICOS
- 13.8 OPERADORES DE ASIGNACIÓN
- 13.9 OPERADORES DE COMPARACIÓN

# ÍNDICE

- 13.10 OPERADORES DE INCREMENTO/DECREMENTO
- 13.11 OPERADORES LÓGICOS
- 13.12 OPERADORES DE CADENAS
- 13.13 NÚMEROS ALEATORIOS
- 13.14 IF
- 13.15 ELSE
- 13.16 ELSEIF
- 13.17 WHILE
- 13.18 FOR
- 13.19 SWITCH
- 13.20 FUNCIONES INTEGRADAS Y DE USUARIO

# ÍNDICE

13.21 INCLUDE

13.22 REQUIRE

13.23 INCLUDE\_ONCE() Y REQUIRE\_ONCE()

13.24 ELEMENTOS DE ORIENTACIÓN A OBJETO

13.25 GESTIÓN DE ERRORES

13.16 EXCEPCIONES EN PHP

13.27 FORMULARIOS. PROCESAMIENTO DE DATOS RECIBIDOS  
DESDE EL CLIENTE. MÉTODOS DE ENVÍO DE DATOS DESDE EL  
CLIENTE AL SERVIDOR

13.28 USO DE LA FUNCION ISSET

# ÍNDICE

13.29 SESIONES

13.30 PROBLEMAS DE AUTENTICACIÓN DE USUARIOS

13.31 MECANISMOS PARA MANTENER EL ESTADO ENTRE CONEXIONES

# **13.1 CLASIFICACIÓN**

- Un lenguaje de programación es un lenguaje inventado para controlar una máquina, (normalmente, un ordenador). Hay muchísimos, de toda clase de tipos y características, inventados para facilitar el abordaje de distintos problemas, el mantenimiento del software, su reutilización, mejorar la productividad, etc.



## **13.2 HERRAMIENTAS DE EDICIÓN DE CÓDIGO**

## 13.2.1 NOTEPAD ++



## 13.2.2 DREAMWEAVER



The screenshot shows the Adobe Dreamweaver CS6 product page. At the top left is the Adobe logo. At the top right is the text "Asistencia". Below the logo, the text "Adobe Creative Suite family /" is followed by "Adobe Dreamweaver CS6" in a large font. A horizontal navigation bar contains the following links: "Resumen", "Funciones", "Especificaciones técnicas", "FAQ", "Muestras", and "Guía de compra" (with a small grid icon). The main content area features a large image with a green digital background. Overlaid on this image is a white box containing the text "Diseño a la vanguardia de la Web", "Crea sitios web más atractivos para más plataformas y dispositivos.", and two links: "Ver el vídeo >" and "Novedades en CS6 >". On the right side of the page, there is a vertical sidebar with a list of links, including "P", "p", "C", "P", "P", "P", "e", "C", "n", "P", "y", "C", "B", "C".

## 13.2.3 ECLIPSE

The screenshot shows the Eclipse website homepage. At the top, there is a banner for "AGILE ALM CONNECT" in Reston, Virginia, from March 26th to 29th, featuring a woman's profile. To the right of the banner is a purple box with the text "AGILEALM 2012 FEATURED SESSIONS" and "ADAPTIVE LEADERSHIP FOR CONTINUOUS DELIVERY". Further right, there is a section titled "Visit other Eclipse Sites" with icons for mp, a microphone, a magnifying glass, a speech bubble, and a server. Below the banner is a navigation bar with links: Home, Downloads, Users, Members, Committers, Resources, Projects, and About Us. To the right of the navigation bar is a "Google™ Custom Search" box. The main content area is titled "Featured Eclipse Project" and features a blue box on the left with the text: "The **Virgo Web Server** from EclipseRT is a completely module-based Java application server that is designed to run enterprise Java applications and Spring-powered applications with a high degree of flexibility and reliability." Below this text is a "read more" button. To the right of the text is a large image of the "VIRGO from eclipseRT" logo. On the far right, there is a yellow box with the text "Get Started now... Download Eclipse". Below this, there are two columns of links: "» Plugins" and "» Documentation" on the left, and "» Co" and "» Re" on the right.

**AGILE ALM CONNECT**  
Reston, Virginia - March 26<sup>th</sup> - 29<sup>th</sup>

**AGILEALM 2012 FEATURED SESSIONS**  
**ADAPTIVE LEADERSHIP FOR CONTINUOUS DELIVERY**

Visit other Eclipse Sites

Home Downloads Users Members Committers Resources Projects About Us

Google™ Custom Search

### Featured Eclipse Project

The **Virgo Web Server** from EclipseRT is a completely module-based Java application server that is designed to run enterprise Java applications and Spring-powered applications with a high degree of flexibility and reliability.

[read more](#)

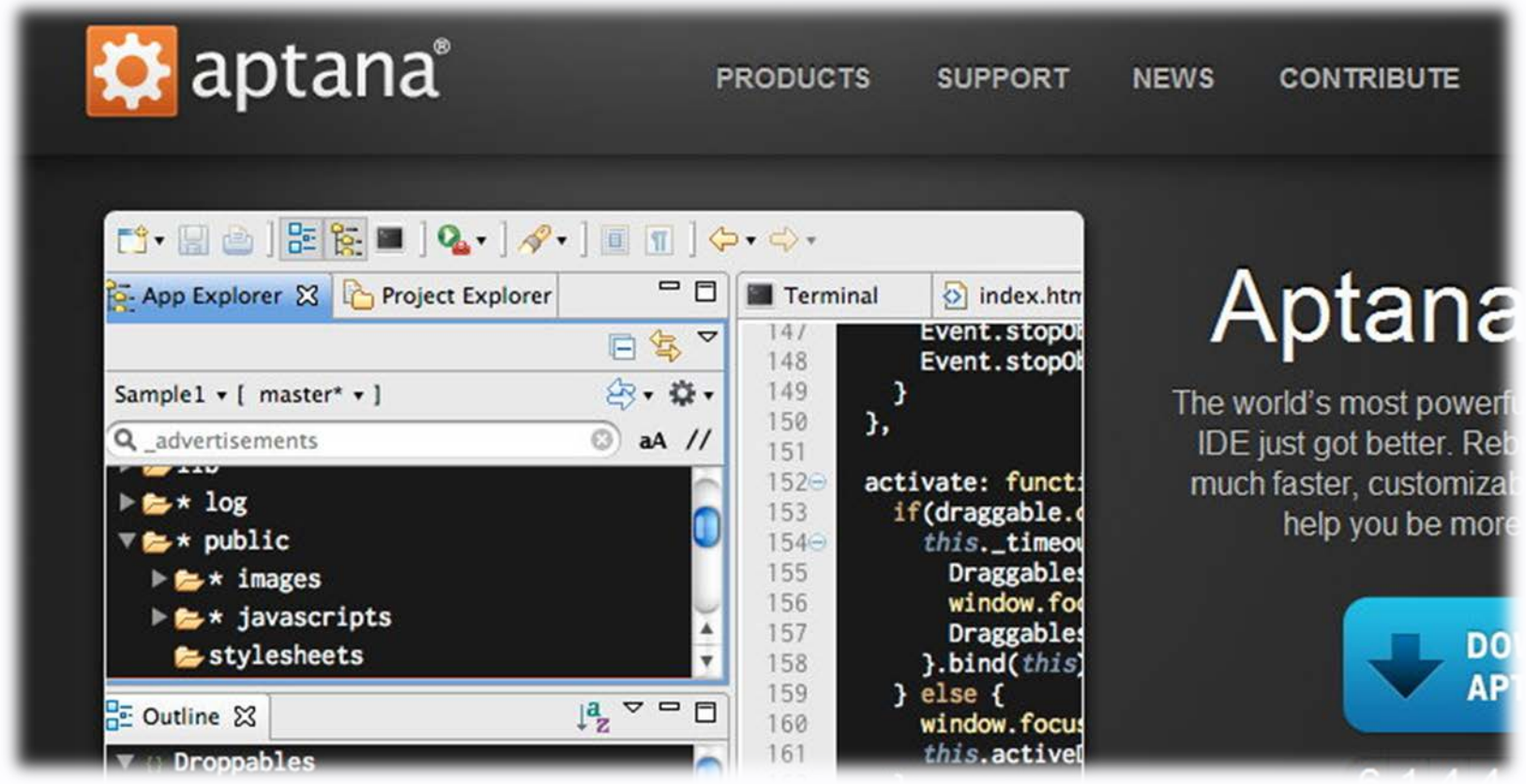
**VIRGO**  
from eclipseRT

Get Started now...  
**Download Eclipse**

» Plugins » Co  
» Documentation » Re



## 13.2.4 APTANA STUDIO



## 13.3 PHP

- PHP, acrónimo de "*PHP Hypertext Preprocessor*", es un lenguaje interpretado de alto nivel embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl, con solamente un par de características PHP específicas. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil.

# **13.4 INTEGRACIÓN CON LOS LENGUAJES DE MARCAS. SINTAXIS**



```
<html>
<head>
<title>EJEMPLO</title>
</head>
<body>
</html>
<?php
echo "!Hola, Soy un script PHP!";
?>
<P>Una parte en HTML
<?
echo "esta es una segunda parte en PHP";
?>
</body>
</html>
```

Hola, Soy un script PHP!

Una parte en HTML esta es una segunda parte en PHP

## **13.5 COMENTARIOS**

```
<?
echo 'Esto es una prueba';
// Esto es un comentario estilo c++ de una sola
línea
/* Esto es un comentario multilínea
!Si! otra línea de comentarios */
echo 'Esto es otro test';
echo 'Un test final';
# Comentario estilo consola de una sola línea
?>
```

# **13.6 ELEMENTOS DEL LENGUAJE ESTRUCTURADO: TIPOS DE DATOS, VARIABLES, OPERADORES, ESTRUCTURAS DE CONTROL, SUBPROGRAMAS...**

# ENTEROS

```
$a = 1234; # número decimal
```

```
$a = -123; # un número negativo
```

```
$a = 0123; /* número octal (equivalente al 83 decimal) */
```

```
$a = 0x12; /* número hexadecimal (equivalente al 18 decimal) */
```

# NÚMEROS EN PUNTO FLOTANTE

`$a = 1.234;`

`$a = 1.2e3;`

# CADENAS

Secuencia	Significado
\n	Nueva línea.
\r	Retorno de carro.
\t	Tabulación.
\\	Barra invertida.
\s	Signo del dólar.
\"	Comillas dobles.



# ARRAYS

```
$a[0] = "abc";  
$a[1] = "def";  
$b["foo"] = 13;
```

```
$a[] = "hola"; // $a[2] == "hola"  
$a[] = "mundo"; // $a[3] == "mundo"
```

```
$a[1] = $f; # ejemplos de una sola dimensión  
$a["foo"] = $f;  
$a[1][0] = $f; # bidimensional  
$a["foo"][2] = $f;  
$a[3]["bar"] = $f;  
$a["foo"][4]["bar"][0] = $f; # tetradimensional
```

## 13.6.2 VARIABLES

```
$var = "Bob";  
$Var = "Joe";  
echo "$var, $Var"; // produce la salida "Bob, Joe"
```

# **13.7      OPERADORES ARITMÉTICOS**



## 13.7 OPERADORES ARITMÉTICOS

Operación	Sintaxis	\$a	\$b	Resultado
Suma	$\$a + \$b$	12	7,3	19,3
Diferencia	$\$a - \$b$	12	7,3	4,7
Producto	$\$a * \$b$	12	7,3	87,6
Cociente	$\$a / \$b$	12	7,3	1,64383561644
Cociente entero	$(\text{int})\$a / \$b$	12	7,3	1
Resto división	$\$a \% \$b$	12	5	2
Potencia $a^b$	$\text{pow}(\$a, \$b)$	2	3	8
Raíz cuadrada	$\text{sqrt}(\$a)$	9		3

```
<?
# definimos dos variables numéricas asignándoles
valores
$a=23;
$b=34;
/* Hacemos la suma y escribimos directamente los
resultados utilizando las instrucciones print y echo
*/
Echo "La suma de $a+$b es: ".$a."+".$b."="."($a+$b);
echo "<BR>";

# guardamos el resultado en una nueva variable
$c=$a+$b;
```

```
/* ahora presentamos el resultado utilizando esa  
nueva variable */
```

```
echo "La suma de $a+$b es: ".$a."+".$b."=".$c;  
echo "<BR>";
```

```
/* Modificamos ahora los valores de $a y $b  
comprobando que el cambio no modifica lo contenido  
en la variable $c */
```

```
$a=513; $b=648;
```

```
print "<br> C sigue valiendo: ".$c;
```

```
echo "<br>";
```

```
?>
```

# **13.8      OPERADORES DE ASIGNACIÓN**



```
$a = ($b = 4) + 5;
```

```
// Ahora $a es igual a 9, y $b vale 4.
```



## **13.9 OPERADORES DE COMPARACIÓN**



Nombre	Ejemplo	Resultado
Igualdad	$a = b$	Cierto si $a$ es igual a $b$
Identidad	$a = = b$	Cierta si $a$ es igual a $b$ y son del mismo tipo
Desigualdad	$a \neq b$	Cierto si $a$ no es igual a $b$
Mayor que	$a > b$	Cierto si $a$ es estrictamente mayor que $b$
Menor que	$a < b$	Cierto si $a$ es estrictamente menor que $b$
Mayor o igual que	$a \geq b$	Cierto si $a$ es mayor o igual que $b$
Menor o igual que	$a \leq b$	Cierto si $a$ es menor o igual que $b$

# **13.10 OPERADORES DE INCREMENTO/DECREMENTO**



Nombre	Ejemplo	Resultado
Preincremento	++\$a	Incrementa en 1 \$a y después devuelve \$a
Postincremento	\$a++	Devuelve \$a y después incrementa \$a en 1
Predecremento	--\$a	Decrementa en 1 \$a y después devuelve \$a
Postdecremento	\$a--	Devuelve \$a y después decrementa \$a en 1

# **13.11 OPERADORES LÓGICOS**

Nombre	Ejemplo	Resultado
Y	\$a and \$b \$a && \$b	Cierto si \$a y \$b son ciertos
O	\$a or \$b \$a    \$b	Cierto si \$a o \$b son ciertos
O exclusiva	\$a xor \$b	Cierto si \$a o \$b son ciertos, pero no ambos a la vez
Negación	!\$a	Cierto si \$a no es cierto

# **13.12 OPERADORES DE CADENAS**

Nombre	Ejemplo	Resultado
Concatenación	\$a . \$b	Concatena \$a y \$b
Concatenación y asignación	\$a .= \$b	\$a tiene el contenido de lo que tenía anteriormente más lo que tiene \$b



# **13.13 NÚMEROS ALEATORIOS**

```
<?
# generador de números para el sorteo de la
primitiva
mt_srand(time()); # mejoramos la aleatoriedad
# generamos y presentamos por pantalla los 6 números
echo mt_rand(1,49);
echo " , ";
echo mt_rand(1,49);
echo " , ";
echo mt_rand(1,49);
echo " , ";
echo mt_rand(1,49);
echo " , ";
echo mt_rand(1,49);
```

```
echo " , " ;  
echo mt_rand(1,49) ;  
echo "<br>" ;  
?>
```

**13.14 IF**

```
if ($a > $b) {  
print "a es mayor que b";  
$b = $a;  
}
```

# 13.15 ELSE

```
if ($a > $b) {  
  print "a es mayor que b";  
}  
else {  
  print "a NO es mayor que b";  
}
```

# **13.16 ELSEIF**



```
if ($a > $b) {  
  print "a es mayor que b";  
}  
elseif ($a == $b) {  
  print "a es igual que b";  
}  
else {  
  print "a es mayor que b";  
}
```

# 13.17 WHILE

```
<?
$i=0;
while($i<=10) {
    echo $i;
    echo "<br>";
/* hacemos un "intro" entre número y número*/
    $i++;
}
?>
```

0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10



**13.18 FOR**

```
/* ejemplo 1 */
```

```
for ($i =1; $i<=10;$i++) {  
    print $i;  
}
```

```
/* ejemplo 2 */
```

```
for ($i=1;;$i++)  
{  
    if ($i>10) {  
        break;  
    }  
    print $i;  
    $i++;  
}
```

```
/* ejemplo 3 */
```

```
$i=1;  
for (;;) {  
    if ($i>10) break;  
    print $i;  
    $i++;  
}
```

```
/* ejemplo 4 */
```

```
for ($i=1;$i<=10;print$i, $i++);
```

# **13.19 SWITCH**

If	Switch
<pre>if (\$i == 0) {   print "i es igual a 0"; } if (\$i == 1) {   print "i es igual a 1"; } if (\$i == 2) {   print "i es igual a 2"; }</pre>	<pre>switch(\$i) {   case 0:     print "i es igual a 0";     break;   case 1:     print "i es igual a 1";     break;   case 2:     print "i es igual a 2";     break; }</pre>



```
switch ($i) {  
    case 0:  
        print "1 es igual a 0";  
    case 1:  
        print "1 es igual a 1";  
    case 2:  
        print "1 es igual a 2";  
}
```

### ***/\* ejemplo 1\*/***

```
switch ($i) {  
  case 0:  
  case 1:  
  case 2:  
    print "i es menor que 3, pero  
no negativo";  
    break;  
  case 3:  
    print "i es 3";  
}
```

### ***/\*ejemplo 2\*/***

```
switch ($i) {  
  case 0:  
    print "i es 0";  
    break;  
  case 1:  
    print "i es 1";  
    break;  
  case 2:  
    print "i es 2";  
    break;  
    default:  
    print "i no 0, 1 o 2";  
}
```

# **13.20    FUNCIONES INTEGRADAS Y DE USUARIO**



```
function func ($arg_1, $arg_2, ..., $arg_n) {  
echo "Funci&oacute;n de ejemplo.";  
return $retval;  
}
```

```
<?
```

```
function doble($numero)  
{  
return ($numero*2);  
}
```

```
$b=7;
```

```
$c=doble($b);
```

```
echo "el número $b, y su doble $c";
```

```
?>
```

# **13.21 INCLUDE**

```
<?
/* Fichero funciones.inc.php */
function doble($numero) {
return ($numero*2); }

function mitad($numero) {
return (int)($numero/2); }
?>
```

```
<?
/* fichero prueba.php */
include( "funciones.inc.php" );

$a=6;
echo $a;
echo "<br>";
echo doble($a);
echo "<br>";
echo mitad($a);
?>
```



# **13.22 REQUIRE**

```
<?
/* fichero prueba.php */
require( "funciones.inc.php" );

$a=6;
echo $a;
echo "<br>";
echo doble($a);
echo "<br>";
echo mitad($a);
?>
```

## **13.23 INCLUDE\_ONCE() Y REQUIRE\_ONCE()**

```
<?
/* fichero prueba.php */
include_once( "funciones.inc.php" );

$a=6;
echo $a;
echo "<br>";
echo doble($a);
echo "<br>";
echo mitad($a);
?>
```

```
<?
/* fichero prueba.php */
require_once( "funciones.inc.php" );

$a=6;
echo $a;
echo "<br>";
echo doble($a);
echo "<br>";
echo mitad($a);
?>
```

## **13.24 ELEMENTOS DE ORIENTACIÓN A OBJETO**

```
<?
class Caja{
    var $alto;
    var $ancho;
    var $largo;
    var $contenido;
    var $color;

function introduce($cosa){
    $this->contenido = $cosa;
}
function muestra_contenido(){
    echo $this->contenido; }
}
?>
```

```
$micaja = new Caja;  
$micaja->introduce("algo");  
$micaja->muestra_contenido();
```



## **13.25 GESTIÓN DE ERRORES**

- E\_ERROR: errores fatales en tiempo de ejecución.
- E\_WARNING: advertencia en tiempo de ejecución (no son fatales).
- E\_PARSE: errores fatales en tiempo de compilación.
- E\_NOTICE: avisos en tiempo de ejecución (menos importantes que las advertencias).

## **13.26 EXCEPCIONES EN PHP**

```
try {  
    if(!@include('/path/to/fichero.php')) {  
        throw new Exception('Error al cargar el fichero');  
    }  
}  
catch(Exception $e) {  
    print $e->getMessage();  
}
```

**13.27 FORMULARIOS.  
PROCESAMIENTO DE DATOS  
RECIBIDOS DESDE EL CLIENTE.  
MÉTODOS DE ENVÍO DE DATOS  
DESDE EL CLIENTE AL SERVIDOR**

```
<FORM NAME="formulario" METHOD="GET"  
ACTION="pagina.php">
```

```
<FORM NAME="formulario" METHOD="POST"  
ACTION="pagina.php">
```

```
<form name="datos" method="POST"
action="formu2.php">
nombre
<input type="text" name="nombre">
<br>
email
<input type="text" name="email">
<br>
<input type="submit" name="enviar" value="enviar">
</form>
```

```
<?
/* fichero formu2.php */

$nombre=$_POST['nombre'];
$email=$_POST['email'];

$nombre2=$_REQUEST['nombre'];
$email2=$_REQUEST['email'];

echo "$nombre, $nombre2";
echo "<br>";
echo "$email, $email2";
?>
```



## **13.28 USO DE LA FUNCION ISSET**

```
<?
if (isset($_POST['enviar']))
{
/* estamos aquí porque hemos pulsado el botón enviar
*/

$nombre=$_POST['nombre'];
$email=$_POST['email'];

echo "$nombre";
echo "<br>";
echo "$email";
}
else
{
```

```
/* si estamos aquí es porque enviar no ha sido
pulsado */
?>
<form name="datos" method="POST" action="">
nombre
<input type="text" name="nombre">
<br>
email
<input type="text" name="email">
<br>
<input type="submit" name="enviar" value="enviar">
</form>
<?
}
?>
```

nombre alumno

email alumno@alumno.com

enviar

alumno, alumno

alumno@alumno.com, alumno@alumno.com

# **13.29 SESIONES**

```
<?
session_start();
?>
<html>
<head>
<title>Generar variable de sesi&acute;n</title>
</head>
<body>
<?
$_SESSION["mivariabledesesion"] = "Hola este es el
valor de la variable de sesi&acute;n";
?>
</body>
</html>
```

```
<?
session_start();
?>
<html>
<head>
<title>Leo variable se sesi&oacute;n</title>
</head>
<body>
Muestro esa variable:
<?
echo $_SESSION["mivariabledesesion"];
?>
</body>
</html>
```

Función	Descripción
Session_id()	Nos devuelve el identificador de la sesión.
Session_destroy()	Dá por abandonada la sesión eliminando variables e identificador.
Session_unregister('variable')	Abandona una variable sesión.



```
<? session_start();

if (!isset($_SESSION["cuenta_paginas"])){

    $_SESSION["cuenta_paginas"] = 1;
}else{
    $_SESSION["cuenta_paginas"]++;
}
?>

<html>
<head>
<title>Contar p&aacute;ginas vistas por un usuario
en toda su sesi&oacute;n</title>
</head>
```

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

# **13.30 PROBLEMAS DE AUTENTICACIÓN DE USUARIOS**

```
<?
/* login.php */

session_start();

if (isset($_POST['enviar']))
{
/* hemos pulsado enviar en nuestra página */

$usuario=$_POST['usuario'];
$contrasena=$_POST['contrasena'];
if (($usuario=='alumno') and
($contrasena=='alumno'))
{
echo "usuario correcto";
```

```
echo "<br>";
$_SESSION['usuario']=$usuario; /* la variable de
sesi&oacute;n es inicializada */

echo "<a href='privada.php'>Pulsa aqu&iacute; para
ir a la zona privada</a>";

}
else
{
echo "usuario incorrecto";
echo "<br>";
echo "<a href=''>Pulsa aqu&iacute; para volver a
intentarlo</a>";
}
}
```

```
else
{
/* no hemos pulsado todavía enviar en nuestra página
*/
?>
<form name="login" method="POST" action=" ">
usuario
<input type="text" name="usuario">
<br>
contrase&ntilde;a
<input type="password" name="contrasena">
<br><br>
<input type="submit" name="enviar" value="enviar">
</form>
<? } ?>
```

usuario

contraseña

usuario incorrecto  
[Pulsa aqui para volver a intentarlo](#)

usuario correcto  
[Pulsa aqui para ir a la zona privada](#)

```
<?
/* privada.php */

session_start();
if ((isset($_SESSION['usuario'])))
{
/* estamos de forma legal en la página podremos
acceder a los servicios de la página*/

echo "bienvenido a la p&aacute;gina";

}
else
{
```



```
/* si estamos aquí es porque la variable de sesión  
no tiene valor */  
echo "no deberías estar aquí";  
echo "<br>";  
echo "<a href='login.php'>Accede como es  
debido</a>";  
}  
  
?>
```

bienvenido a la página

no deberías estar aquí  
[Accede como es debido](#)

# **13.31 MECANISMOS PARA MANTENER EL ESTADO ENTRE CONEXIONES**

- En PHP las conexiones que se mantienen pueden tener tres estados, Normal (0), Aborted (1) y Timeout (2).
- Una función muy útil para estos casos, es *connection\_status()* que devuelve el estado de la conexión.