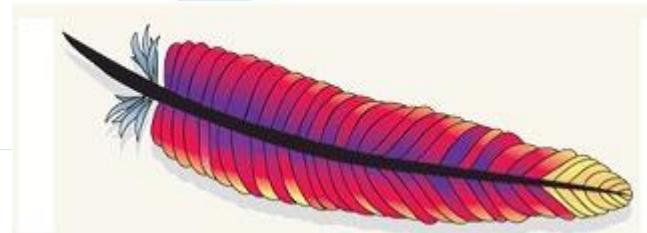


UT2

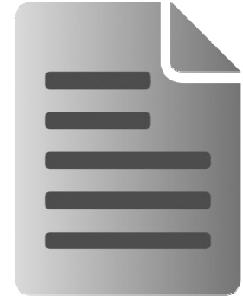
Lenguaje PHP



Lenguaje PHP: manejo de ficheros en el servidor

1º.- Abrir o crear el fichero:

```
$fichero = fopen("ruta_y_nombre", "modo");
```



El "**modo**" puede ser:

"r": lectura desde el comienzo del archivo.

"r+": lectura y escritura desde el comienzo del archivo.

"w": lo borra y escribe desde el comienzo del archivo. Si no existe lo crea.

"w+": lo borra, lee y escribe desde el comienzo. Si no existe lo crea.

"a": escritura desde el final del archivo. Si no existe lo crea.

"a+": lectura y escritura desde el final del archivo. Si no existe lo crea.

Emplear la barra / en las rutas

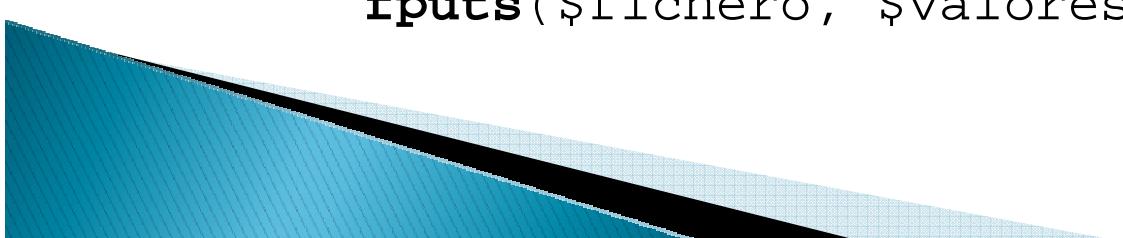
Lenguaje PHP: ficheros

2º.- Posicionarse en el fichero:

```
$fin = feof($fichero)      // true si llega al final  
rewind($fichero)          // rebobina al principio  
fseek($fichero, posicion_en_bytes)  
$posicion_en_bytes = ftell($fichero)
```

3º.- Leer o escribir en el fichero:

```
$linea = fgets($fichero)      // lee una línea y avanza  
$caracter = fgetc($fichero)    // lee un carácter y avanza  
  
fputs($fichero, $valores)    // escribe y avanza
```



No olvidemos escribir al final de cada
línea el "\r\n" o PHP_EOL

Lenguaje PHP: ficheros

4º.- Cerrar el fichero:

```
fclose($fichero);
```

Otras funciones con ficheros:

```
bool file_exists( "ruta_del_fichero" )  
unlink( "ruta_del_fichero" ) // borra el fichero  
copy( "ruta_fichero_origen" , "ruta_fichero_destino" )  
rename( "fichero_original" , "fichero_nuevo" )  
$bytes = filesize( $fichero )  
string  filetype( $fichero )  
string  filemtime( $fichero )  
mkdir( "ruta_del_directorio" ) // crear directorio
```

Lenguaje PHP: ficheros

EJEMPLO: otros modos de leer un fichero:

```
<?php
    if (file_exists("fichero.txt")) {
        echo "<br><H2>Contenido del fichero:</H2>";
        // lee el fichero en una única string
        $lineasDelFichero = readfile("fichero.txt");
        echo $lineasDelFichero, "<br><br>";
        // muestra el contenido del fichero
        include("fichero.txt");
        echo "<br><br>";
        // lee el fichero y lo guarda en un array que se muestra por líneas
        $arrayLineasDelFichero = file("fichero.txt");
        foreach($arrayLineasDelFichero as $linea=>$texto) {
            echo "Linea: ",$linea," Texto: ",$texto,"<br>";
        }
        echo "<br><br>Ultima modificación a las: <br>";
        echo date("h:i:s A", filemtime ("fichero.txt"));
        echo "<br>del día: <br>";
        echo date("j-n-Y", filemtime ("fichero.txt"));
    } else
        echo "El fichero \"fichero.txt\" no existe";
    ?>
```

Lenguaje PHP: ficheros

EJEMPLO: creación, escritura y lectura de líneas en un fichero

```
<?php  
    $f1 = fopen("fichero.txt", "w+");      // borra el contenido y lo crea si no existe  
    fputs($f1, "Primera linea<br>\r\n"); // escribe al inicio del fichero. Salto de línea!  
    fclose($f1);                         // cierra el fichero  
  
    $f1 = fopen("fichero.txt", "a+");      // abrimos para añadir al final (append)  
    fputs($f1, "Añadido al final<br>\r\n");  
    fclose($f1);  
  
    echo "<br><H2>Contenido del fichero:</H2>"; // imprime el contenido del fichero  
    include("fichero.txt");  
  
    $f1 = fopen("fichero.txt", "r");       // abrimos para leer desde el principio  
    echo "<br><h2>Lee los 2 primeros caracteres de la primera línea:</h2>";  
    $c = fgetc($f1);        echo "Primer carácter: ", $c, "<br>";  
    $c = fgetc($f1);        echo "Segundo carácter: ", $c, "<br>";  
    fclose($f1);  
  
    echo "<br><H2>Contenido de todo el fichero:</h2>";  
    $f1 = fopen("fichero.txt", "r");       // lo abre desde el principio  
    while (!feof($f1)) {  
        $linea = fgets($f1);  
        echo $linea;  
    }  
    fclose($f1);  
  
    unlink("fichero.txt");   // borra el fichero  
?>
```

Lenguaje PHP: ficheros

EJEMPLO: guardar los datos de un formulario validado en un fichero y luego **leerlo** para mostrarlo:

```
<html>
    <head></head>
<body>

<!-- VALIDACION -->
<?php
    function procesarValor (&$dato) {
        $dato = trim($dato);
        $dato = stripslashes($dato);
        $dato = htmlspecialchars($dato);
        return $dato;
    }

    function imprimeError ($error) {
        echo "<span style = \"color: red;\">".$error."</span>";
    }

    // Variables para almacenar los valores leídos del formulario:
    $nombre = $apellidos = $edad = "";

    // Variables para almacenar los posibles errores encontrados:
    $errorNombre = $errorApellidos = $errorEdad = "";

    $validacionOK = true;
```

Lenguaje PHP: ficheros

```
// Validación:  
if ($_SERVER[ "REQUEST_METHOD" ] == "POST" ) {  
    if (empty($_POST[ "nombre" ])) {  
        $errorNombre = "Se requiere el nombre";  
        $validacionOK = false;  
    } else {  
        $nombre = procesarValor($_POST[ "nombre" ]);  
        // Valida si el nombre solo contiene letras y espacios:  
        if (!preg_match("/^[a-zA-Z ]*$/",$nombre)) {  
            $errorNombre = "Solo se permite letras y espacios";  
            $validacionOK = false;  
        }  
    }  
  
    if (empty($_POST[ "apellidos" ])) {  
        $errorApellidos = "Se requiere los apellidos";  
        $validacionOK = false;  
    } else {  
        $apellidos = procesarValor($_POST[ "apellidos" ]);  
        // Valida si los apellidos solo contienen letras y espacios:  
        if (!preg_match("/^[a-zA-Z ]*$/",$apellidos)) {  
            $errorApellidos = "Solo se permite letras y espacios";  
            $validacionOK = false;  
        }  
    }  
}
```

Lenguaje PHP: ficheros

```
if ($_POST["edad"] == "") {
    $errorEdad = "Se requiere la edad";
    $validacionOK = false;
} else {
    $edad = procesarValor($_POST["edad"]);
    if (!is_numeric($edad)) { // Valida si la edad es un número
        $errorEdad = "Solo se permite números";
        $validacionOK = false;
    } else {
        if ($edad <= 0) {
            $errorEdad = "Número incorrecto para la edad";
            $validacionOK = false;
        }
    }
}
?>

<h2>GUARDAR DATOS EN UN FICHERO</h2> <p>* campo obligatorio</p>
<form method="post" action=<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>>
    <table><tr>
        <td>Nombre:</td><td><input type="text" name="nombre" value=<?php echo $nombre;?>>
        * <?php imprimeError($errorNombre); ?></td>
    </tr><tr>
        <td>Apellidos:</td><td><input type="text" name="apellidos" value=<?php echo
        $apellidos;?>> * <?php imprimeError($errorApellidos); ?></td>
    </tr><tr>
        <td>Edad:</td><td><input type="text" name="edad" value=<?php echo $edad;?>>
        * <?php imprimeError($errorEdad); ?></td>
    </tr>
    </table>
    <input type="submit" name="submit" value="ENVIAR">
</form>
```

Lenguaje PHP: ficheros

```
<!-- VALORES LEIDOS DEL FORMULARIO -->
<?php
    if ($_SERVER[ "REQUEST_METHOD" ] == "POST" ) {
        if ($validacionOK == true) {
            // Guarda los valores del formulario al final del archivo
            // Lo crea si no existe
            $fichero = fopen("datos.txt", "a+");
            // Recorre los campos del formulario y los guarda en el archivo:
            foreach($_POST as $campo=>$valor) {
                if ($campo != "submit") {
                    fputs($fichero, $campo.": ".$valor.PHP_EOL);
                }
            }

            echo "<span style= \"color: blue;\">";
            echo "<br><h3>Contenido del archivo:<br></h3>";
            // Lee todos los datos escritos hasta el momento:
            rewind($fichero); // rebobina el fichero para colocarse al principio
            while (!feof($fichero)) {
                $linea = fgets($fichero);
                echo $linea, "<br>";
            }
            fclose($fichero);
            echo "</span>";
        }
    }
?>
</body>
</html>
```

La constante **PHP.EOL** (end of line)
equivale a los caracteres CR y LF: \r y \n

Lenguaje PHP: ficheros

ACTIVIDAD: creación de un contador de visitas

- Desarrolla un script llamado **contador.php** que muestre un contador de visitas que será *almacenado en un archivo local al servidor*.
- Cuando se visite esta página, deberá mostrar al usuario *cuántas visitas lleva la web* (incluyéndole a él); y la *fecha y hora del último* usuario que la visitó (el anterior a él).
- Para mostrar la cantidad de visitas, debes *usar imágenes* para cada uno de los dígitos del número de visitas:

La primera vez que se visite la web se creará el archivo para el contador y se almacenará un 1. En las siguientes visitas se abrirá el archivo para leer su valor y luego incrementarlo en 1 y guardarlo.



Lenguaje PHP: subida de ficheros al servidor

- Primero debemos consultar la configuración del archivo C:\xampp\PHP\php.ini:
abrir el fichero o consultararlo visitando esta web:

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



Verificar las directivas del **php.ini** relativas a la subida de ficheros:

```
file_uploads=On

upload_tmp_dir="C:\xampp\tmp"

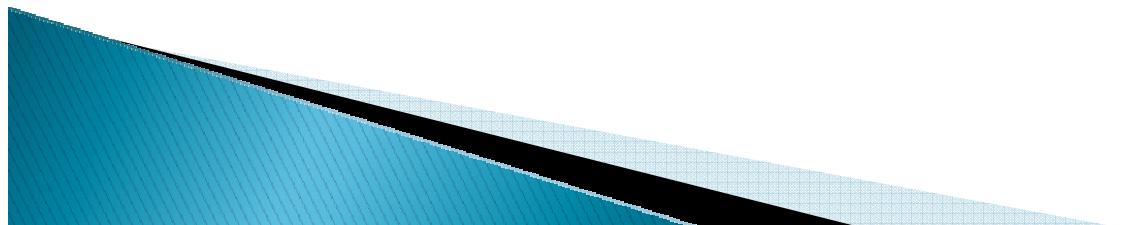
upload_max_filesize=2M

max_file_uploads=20
```

Cuando publicamos nuestra web en un hosting no podemos editar el fichero php.ini; pero sí podemos consultararlo con la función phpinfo()

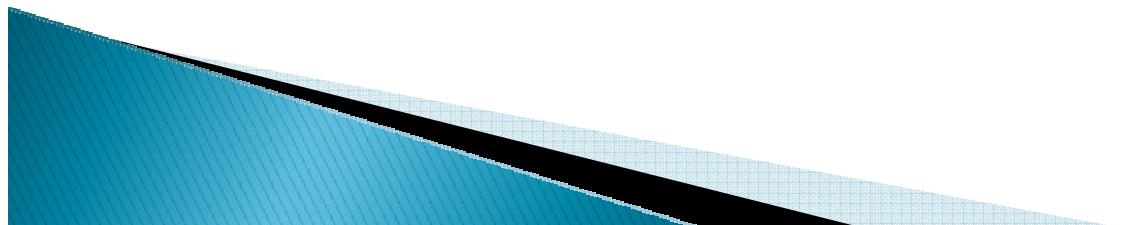
Lenguaje PHP: uploading

- En el **<form>** hemos de incluir el **method="post"** y **enctype="multipart/form-data"**.
- Para crear un control de usuario que le permite subir un fichero: etiqueta **<input type=""file"" name="xxx">**.
- Una vez enviado el formulario, el fichero transferido se guarda en un **directorio temporal del servidor** con un nombre que se le asignada de forma automática.
- El array asociativo **\$_FILES['indice']['valor']** recoge la información relativa a la transferencia del fichero/s.



Lenguaje PHP: uploading

- Array asociativo bidimensional `$_FILES['indice']['valor']` usado para manejar los archivos subidos :
 - '**indice**': nombre (name) del campo `<input type="file" name="nombre_del_índice">` usado para la transferencia.
 - '**valor**': array asociativo con los siguientes campos:
 - 'name': nombre original del archivo
 - 'type': formato del archivo (text/html, image/png, etc)
 - 'tmp_name': nombre con el que se ha guardado en el directorio temporal del server
 - 'error': 1 si hubo error en la transferencia; 0 en caso de éxito
 - 'size': bytes que ocupa el archivo



Lenguaje PHP: uploading

- Por último debemos:

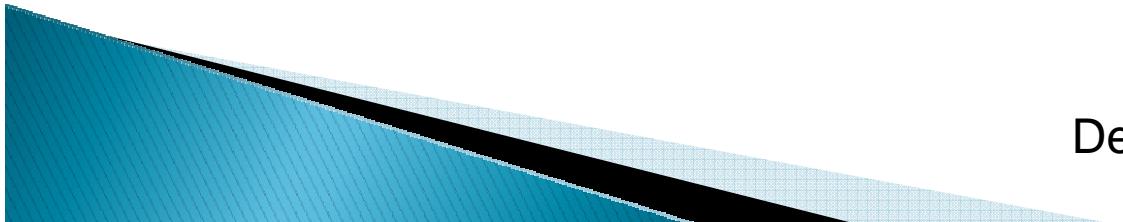
- **Copiar** el archivo de la carpeta temporal (con el nombre automático que tiene) a la carpeta definitiva del server.
- **Borrar** de la carpeta temporal el archivo temporal.

```
$rutaDef = "C:/xampp/htdocs/upload/" ; // directorio de destino
if (!is_dir($rutaDef)) // crea el directorio de destino si no existe
    mkdir($rutaDef);

$nombreTmp = $_FILES['nombre_input_type_file']['tmp_name'];
$nombreDef = $rutaDef . $_FILES['nombre_input_type_file']['name'];

$resultado = copy($nombreTmp,$nombreDef); // copia el archivo

if ($resultado == 0)
    unlink($nombreTmp); // borra el archivo temporal
```



Debemos usar la barra / para las rutas

Lenguaje PHP: uploading

- **EJEMPLO:** formulario que sube dos archivos al servidor:

```
<html>
    <head></head>
<body>

<!-- VALIDACIÓN DEL FORMULARIO -->
<?php
    $errorValidacion = " ";
    if ($_SERVER[ "REQUEST_METHOD" ] == "POST" ) {
        $validacionOK = true;
        foreach($_FILES as $archivo=>$valores) {
            if (empty($valores['name'])) {
                $validacionOK = false;
                $errorValidacion = "Debe especificar un archivo";
            }
        }
    }
?>

<h2>SUBIDA DE FICHEROS AL SERVIDOR</h2>
<form enctype="multipart/form-data" method="post"
        action=<?php echo htmlspecialchars($_SERVER[ "PHP_SELF" ]); ?>">
    <table>
        <tr> <td>Archivo 1:</td> <td> <input type="file" name="archivo1">
            <?php echo $errorValidacion; ?> </td> </tr>
        <tr> <td>Archivo 2:</td> <td> <input type="file" name="archivo2">
            <?php echo $errorValidacion; ?> </td> </tr>
    </table>
    <input type="submit" name="enviar" value="ENVIAR">
</form>
<hr>
```

Lenguaje PHP: uploading

```
<!-- FICHEROS SUBIDOS CON EL FORMULARIO -->
<?php
if ($_SERVER[ "REQUEST_METHOD" ] == "POST" && $validacionOK == true) {
    foreach($_FILES as $archivo=>$valores) {
        // Imprime una tabla con la información de cada archivo subido
        echo "<table border=\\"1\\" style= \\\"color: blue;\\\">";
        echo "<tr>";
        echo "<td colspan=\\"2\\">"; echo $archivo; echo "</td>";
        echo "</tr>";
        foreach ($valores as $indice=>$valor) {
            echo "<tr><td>"; echo $indice; echo "</td>";
            echo "<td>"; echo $valor; echo "</td></tr>";
        }
        echo "</table>";
        echo "<br>";

        // Copia el archivo a la carpeta de destino y borra el archivo temporal
        $rutaDef = "C:/xampp/htdocs/upload/";
        if (!is_dir($rutaDef)) // Crea el directorio de destino si no existe
            mkdir($rutaDef);
        $nombreTmp = $_FILES[$archivo][ 'tmp_name' ];
        $nombreDef = $rutaDef . $_FILES[$archivo][ 'name' ];
        $resultado = copy($nombreTmp,$nombreDef);
        if ($resultado == 0)
            unlink($nombreTmp); // Borra el archivo temporal
    }
}
?>
</body>
</html>
```

Lenguaje PHP: uploading

- **EJEMPLO:** form que sube un archivo si es una imagen JPG, GIF o PNG y < 200 KB

```
<html>
    <head></head>
<body>

    <!-- VALIDACIÓN DEL FORMULARIO -->
    <?php
        $errorValidacion = " ";
        if ($_SERVER["REQUEST_METHOD"] == "POST") {
            $validacionOK = true;
            if (empty($_FILES['imagen']['name'])) {
                $validacionOK = false;
                $errorValidacion .= "Debe especificar un archivo de imagen.";
            } else {
                $tipo = $_FILES['imagen']['type'];
                if (!($tipo=="image/pjpeg" || $tipo=="image/jpeg"
                    || $tipo=="image/gif" || $tipo=="image/png")) {
                    $validacionOK = false;
                    $errorValidacion .= "Formato ".$tipo." no permitido. JPG, GIF o PNG.";
                }
                $tamaño = $_FILES['imagen']['size'];
                if ($tamaño > 200000 || $tamaño == 0) { // máximo 200K = 200000B
                    $validacionOK = false;
                    $errorValidacion .= $tamaño." bytes excede los 200 KB permitidos.";
                }
            }
        }
    ?>
```

localhost/upload_imagen.php

SUBIDA DE IMAGENES AL SERVIDOR

Solo puede subir JPG y GIF de hasta 200 KB

Imagen: Seleccionar archivo | Ningún archivo seleccionado

ENVIAR

name	Sin nombre.jpg
type	image/jpeg
tmp_name	C:\xampp\tmp\phpA135.tmp
error	0
size	47014



Lenguaje PHP: uploading

```
<h2>SUBIDA DE IMAGENES AL SERVIDOR</h2>    <h4>JPG y GIF de hasta 200 KB</h4>
<form enctype="multipart/form-data" method="post"
    action=<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>>
    Imagen: <input type="file" name="imagen"> <?php echo $errorValidacion; ?> <br>
    <input type="submit" name="enviar" value="ENVIAR">
</form>  <hr>

<?php
    if ($_SERVER["REQUEST_METHOD"] == "POST" && $validacionOK == true) {
        echo "<table border=\"1\" style= \"color: blue;\">";
        foreach ($_FILES['imagen'] as $indice=>$valor) {
            echo "<tr>";  echo "<td>";  echo $indice;  echo "</td>";
            echo "<td>";  echo $valor;  echo "</td>";  echo "</tr>";
        }
        // Copia el archivo temporal a la carpeta de destino
        $rutaDef = "./upload/";
        if (!is_dir($rutaDef))
            mkdir($rutaDef);
        $nombreTmp = $_FILES['imagen']['tmp_name'];
        $nombreDef = $rutaDef . $_FILES['imagen']['name'];
        $resultado = copy($nombreTmp,$nombreDef);
        if ($resultado == 0)
            unlink($nombreTmp); // Borra el archivo temporal
        echo "<tr>";  echo "<td colspan=\"2\">"; // Dibuja la imagen subida
        echo "<img src=\"\" . $nombreDef . "\">";
        echo "</td>";  echo "</tr>";  echo "</table>";
    }
?
</body>
</html>
```

Lenguaje PHP: compresión de archivos en el server

- La compresión de archivos en el server resulta útil para:
 - Economizar espacio en el server
 - Ahorrar tiempo en las descargas
- Para saber si el intérprete de PHP de nuestro server soporta las funciones para la compresión de archivos debemos consultar la configuración del archivo **php.ini** (mediante la función **phpinfo()**) y verificar las directivas de la **librería Zlib**:



zlib

ZLib Support	enabled
Stream Wrapper	compress.zlib://
Stream Filter	zlib.inflate , zlib.deflate
Compiled Version	1.2.7.3
Linked Version	1.2.7.3

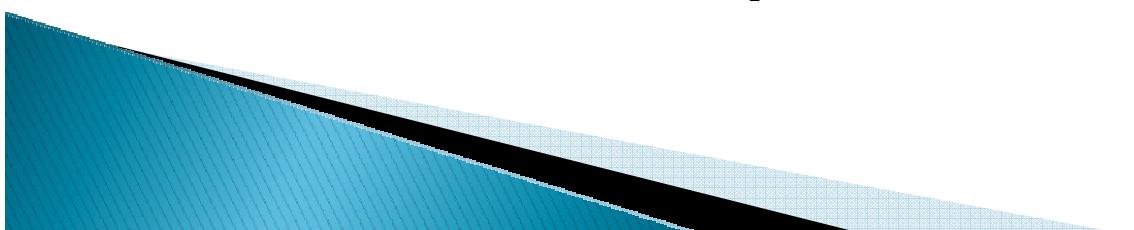
Lenguaje PHP: compresión de archivos

- **EJEMPLO:** sube un archivo al servidor y lo comprime

```
<html>
    <head></head>
<body>

<h2>SUBE UN ARCHIVO COMPRIMIDO AL SERVIDOR</h2>
<form enctype="multipart/form-data" method="post"
        action=<?php echo htmlspecialchars($_SERVER['PHP_SELF']);?>>
    Archivo: <input type="file" name="archivo"> <br>
    <input type="submit" name="enviar" value="SUBIR">
</form> <hr>

<?php
    if ($_SERVER['REQUEST_METHOD'] == "POST" && !empty($_FILES['archivo']['name'])) {
        // Copia el archivo temporal a la carpeta de destino
        $carpetaDestino = "./upload_zip/";
        if (!is_dir($carpetaDestino))
            mkdir($carpetaDestino);
        $nombreTmp = $_FILES['archivo']['tmp_name'];
        $nombre = $_FILES['archivo']['name'];
        $nombreDef = $carpetaDestino . $nombre;
        $resultado = copy($nombreTmp, $nombreDef);
        // Borra el archivo temporal
        if ($resultado == 0)
            unlink($nombreTmp);
```



Lenguaje PHP: compresión de archivos

```
// Crea el archivo .zip dentro de la carpeta de destino
$nombreZip = $nombreDef . ".zip";
$archivoZip = new ZipArchive;
if ($archivoZip->open($nombreZip, ZipArchive::CREATE) !== TRUE)
    exit("No se puede crear el archivo".$archivoZip);

// Añade el archivo al .zip
$archivoZip->addFile($nombreDef, $nombre);
// se puede añadir más archivos al mismo .zip: $archivoZip->addFile(otro,otro);

// Cierra el archivo .zip
$archivoZip->close();

// Borra el archivo original subido para dejar solo el comprimido
unlink($nombreDef);

// Lista el directorio de Destino con todos los .zip
echo "<h3>Listado de archivos subidos y comprimidos al servidor:</h3>";
$listadoFicheros = scandir($carpetaDestino);
for ($i = 2; $i < count($listadoFicheros); $i++) {
    $nombreFichero = $listadoFicheros[$i];
    echo $nombreFichero . "<br>";
}
?>
</body>
</html>
```

Lenguaje PHP: compresión de archivos

- **ACTIVIDAD:** Modifica el ejemplo anterior para que el usuario indique cuántos archivos desea subir al servidor de una tacada y al subirlos se compriman **todos en un único archivo zip**:

localhost/upload_many_in_zip.php

¿Cuántos archivos desea subir dentro del mismo archivo .zip?



localhost/upload_many_in_zip2.php

SUBIDA DE ARCHIVOS COMPRIMIDOS

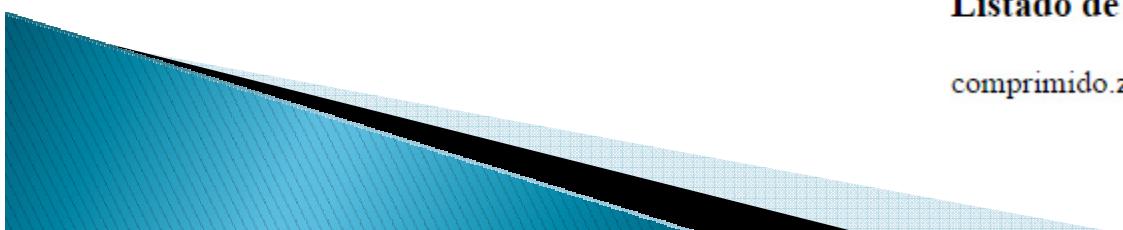
Archivo 0: calculadora.php
Archivo 1: favicon.ico



localhost/upload_many_in_zip3.php

Listado de archivos subidos y comprimidos al servidor:

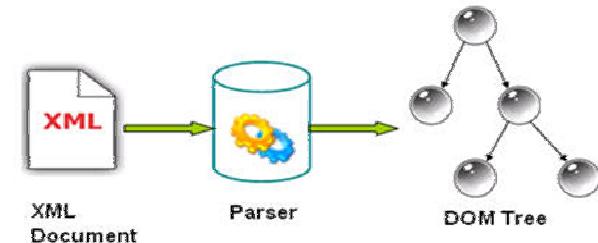
comprimido.zip



Lenguaje PHP: manejo de ficheros XML

- PHP integra la librería **SimpleXML Parser** (reconocedor), que permite:

- Verificar si un archivo XML es bien formado.
- Abrir un archivo XML válido para leer sus elementos y atributos.



- **Ejemplo:** apertura y **verificación** del archivo XML datos.xml:

```
<?php
    libxml_use_internal_errors(true);

$xml = simplexml_load_file("datos.xml");
if ($xml === false) {
    echo "Errores encontrados en el archivo XML: ."."<hr>";
    foreach(libxml_get_errors() as $error)
        echo "<br>". $error->message;
} else
    // imprime el árbol de elementos del
    // archivo XML mediante arrays anidados:
    print_r($xml);
?>
```

La librería verifica el archivo .xml (bien formado) pero no lo valida con el DTD.

datos.xml

```
<?xml version='1.0' encoding='UTF-8'?>
<document>
    <user valor=4>Carlos</wronguser>
    <email>carlos@mimail.com</wrongemail>
</DOCument>
```

Lenguaje PHP: manejo de ficheros XML

- Al abrir el archivo XML, el parser genera un árbol DOM con los elementos encontrados:
- **Lectura de elementos** de un archivo XML:

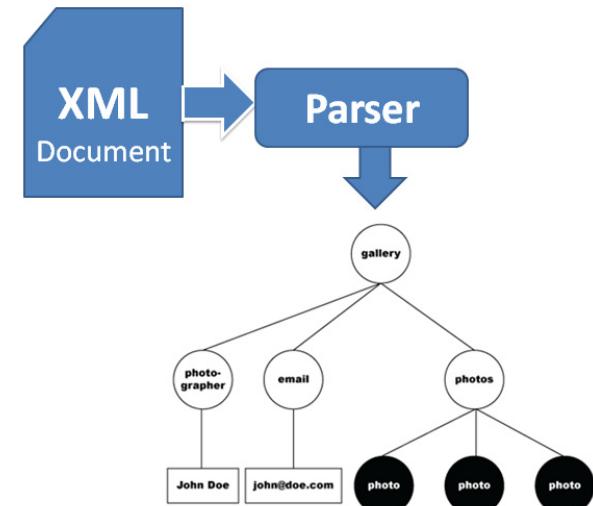
```
<?php
    libxml_use_internal_errors(true);

    $ficha = simplexml_load_file("ficha.xml");
    if ($ficha === false)
        echo "Error en el archivo XML";
    print_r($ficha);
    echo "<br> <hr>";

    echo $ficha->nombre . "<br>";
    foreach ($ficha->apellido as $valor)
        echo $valor . "<br>";
    echo $ficha->direccion . "<br>";

    // Equivale a: echo $ficha->apellido[0]."<br>";
    //                 echo $ficha->apellido[1]."<br>";

    echo "<hr>";
?>
```



1 y ?: **elemento->subelemento->subelemento...**
+ y *: **elemento[posicion]->subelemento...**

datos.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ficha [
    <!ELEMENT ficha (nombre, apellido+, direccion)>
    <!ELEMENT nombre (#PCDATA)>
    <!ELEMENT apellido (#PCDATA)>
    <!ELEMENT direccion (#PCDATA)>
]>
<ficha>
    <nombre>Angel</nombre>
    <apellido>Romero</apellido>
    <apellido>Santana</apellido>
    <direccion>c/Sin nombre, 36</direccion>
</ficha>
```

Lenguaje PHP: manejo de ficheros XML

- **Actividad:** hemos exportado de una BBDD a un archivo de texto las notas de un examen y queremos mostrarlas como una página HTML:

1º.- Diseña un **DTD** con una gramática que permita guardar notas de alumnos:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE calificaciones [
    <!ELEMENT calificaciones (asignatura, alumno*)>
    <!ELEMENT asignatura (#PCDATA)>
    <!ELEMENT alumno (nombre, nota)>
    <!ELEMENT nombre (#PCDATA)>
    <!ELEMENT nota (#PCDATA)>
]>
```

¡OJO! Los nombres de los elementos XML son case-sensitive. Deben llamarse en PHP tal cual se llaman en el archivo DTD.

2º.- Crea un **archivo .xml** que cumpla la gramática anterior y que guarde 5 notas. Valídalo con:
https://validator.w3.org/#validate_by_input

3º.- Implementa una aplicación en PHP que **abra el archivo XML** y muestre las notas en una **tabla**:



The screenshot shows a web browser window with the URL "localhost/xml.php" in the address bar. The page title is "CALIFICACIONES". Below the title, the word "DSW" is displayed. A table is shown with the following data:

Luis	5,4
Ana	7,9
Jose	6,8
Antonio	8,5
Marta	7,3

Lenguaje PHP: manejo de ficheros XML

- Lectura de atributos de elementos XML: usa un array asociativo en el que el índice es el atributo: **elemento['atributo']**

```
<html> <body bgcolor="#82A2CC> <h3>LIBRERIA</h3>
<?php
    libxml_use_internal_errors(true);

    $bookstore = simplexml_load_file("libros.xml");
    if ($bookstore === false)
        echo "Error en el archivo XML";
    print_r($bookstore);
    echo "<hr>";
    echo "LIBRERIA". "<br><br>";
    echo "<table border=\"1\">";
    foreach ($bookstore->book as $book) {
        echo "<tr><td>";
        echo $book['category'];
        echo "</td><td>";
        echo $book->title;
        echo "</td><td>";
        echo $book->title['lang'];
        echo "</td><td>";
        echo $book->author;
        echo "</td></tr>";
    }
    echo "</table>";
?
</body></html>
```

¡OJO! Recuerda que XML es case-sensitive para el nombre de los elementos y de sus atributos.

libros.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE bookstore [
    <!ELEMENT bookstore (book*)>
    <!ELEMENT book (title, author)>
    <!ATTLIST book category CDATA #REQUIRED>
    <!ELEMENT title (#PCDATA)>
    <!ATTLIST title lang CDATA #REQUIRED>
    <!ELEMENT author (#PCDATA)>
]>
<bookstore>
    <book category="COOKING">
        <title lang="en">Everyday Italian</title>
        <author>Giada De Laurentiis</author>
    </book>
    <book category="CHILDREN">
        <title lang="en">Harry Potter</title>
        <author>J K. Rowling</author>
    </book>
</bookstore>
```

Referencia de la librería SimpleXML Parser:
http://www.w3schools.com/php/php_ref_simplexml.asp

Lenguaje PHP: envío de emails

- Uso de la librería **PHPMailer** para posibilitar el envío de correos electrónicos a través de aplicaciones web escritas en PHP.



- Para realizar el envío se emplea una cuenta en algún servidor de correo electrónico gratuito (GMail, Yahoo, Outlook); siendo este servidor SMTP quien efectúa la transmisión.

Debemos descomprimir la carpeta con los fuentes de la librería y copiarla como un subdirectorio de htdocs



Lenguaje PHP: envío de emails

- Ejemplo para enviar un correo electrónico por medio de una cuenta en un servidor de GMail:

```
<?php
    require_once('PHPMailer-master/class.phpmailer.php');

    $destinatario = "alumno@gmail.com";
    $asunto = "Probando mails desde php";
    $texto = "Hola, he sido escrito desde PHP.";

    enviarEmail($destinatario, $asunto, $texto);

    function enviarEmail($direccion, $asunto, $texto){
        $mail = new phpmailer();
        $mail->IsSMTP();
        $mail->Mailer = "smtp";

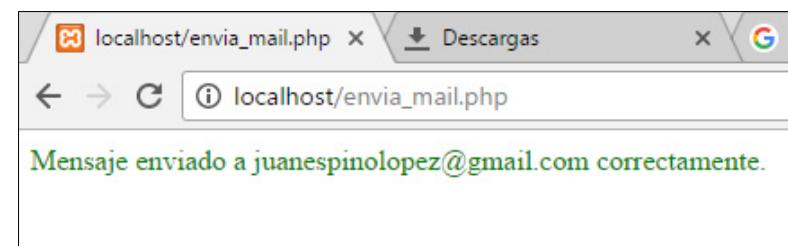
        // Rellenar los datos con una cuenta de GMail:
        $mail->Host = "smtp.googlemail.com";
        $mail->SMTPAuth = true;
        $mail->SMTPSecure = "ssl";
        $mail->Port = 465;
        $mail->Username = "alumno@gmail.com";           // Cuenta del emisor
        $mail->Password = "clave";                     // Contraseña del emisor
```

Lenguaje PHP: envío de emails

```
$mail->From = "alumno@gmail.com";           // Dirección del emisor
$mail->FromName = "FULANITO DE TAL";         // Remitente que le aparece al receptor

$mail->Timeout=30;
$mail->AddAddress($direccion);                // Dirección del destinatario
$mail->Subject = $asunto;                      // Asunto del email
$mail->Body = $texto;                          // Contenido del email
$mail->AltBody = $texto;                       // Contenido del email alternativo

$exito = $mail->Send();
$intentos=1;
while ((!$exito) && ($intentos < 5)) {
    sleep(5);
    $exito = $mail->Send();
    $intentos++;
}
if ($exito) {
    echo "<p style=\"color:green;\"> Mensaje enviado a " . $direccion . " correctamente.</p>";
} else {
    echo "<p style=\"color:red;\"> No se ha podido enviar el correo electronico: " . $mail->ErrorInfo . "</p>";
}
?>
```



Lenguaje PHP: envío de emails

- **Actividad:** desarrollar una aplicación web en PHP llamada **enviamail.php** que muestre al usuario un formulario para enviar incidencias por correo electrónico a la cuenta del alumno:

Tras validar los campos, se enviará un mensaje de correo cuyo asunto será el mismo que el de la incidencia; y cuyo contenido se formará extrayendo los valores del resto de campos del formulario:

```
Urgente: Sí
Tipo: Hardware
Fecha: dd/mm/aaaa
Identificador: 1
E-mail de contacto: usuario@mail.com
Descripción: detalle de la incidencia
```

ENVÍO DE INCIDENCIAS POR EMAIL

* campo obligatorio

Asunto: Resumen... *

Urgente:

Tipo: Hardware Redes Software

Fecha: dd/mm/aaaa *

Identificador: 1 *

E-mail de contacto: usuario@mail.com *

Por favor, detalle su incidencia...

Descripción:

ENVIAR

Además se deberá guardar en el servidor un archivo cuyo nombre sea el asunto del correo (la incidencia) y con el mismo contenido que el mensaje creado.

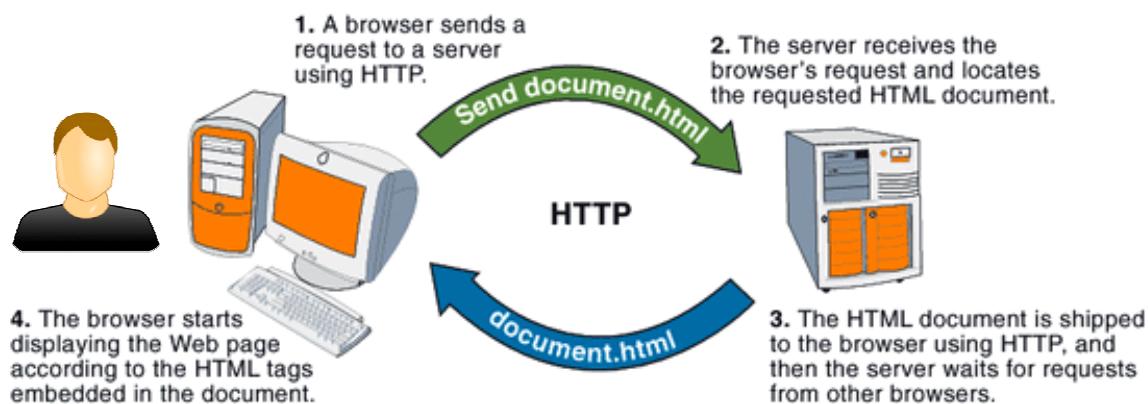
Lenguaje PHP: recordando al usuario

- Es deseable que el servidor web que aloja una aplicación pueda **recordar información sobre los usuarios** que lo han visitado.
- Así la aplicación podría ofrecerles en futuras visitas una experiencia más personal.
- Y podría mostrarles la aplicación usando directamente:
 - El lenguaje elegido
 - La última sección visitada
 - La última búsqueda realizada
 - Etc



Lenguaje PHP: recordando al usuario

- El problema es que **HTTP** es un protocolo **sin estado** (*stateless*): no guarda información de las conexiones anteriores → no mantiene información entre las peticiones de los clientes.
- Trata cada conexión que haga el mismo cliente al servidor como si fueran conexiones independientes → "el server no le recuerda".



Las siguientes veces que acceda el mismo usuario a la misma web, el servidor HTTP no le recuerda y lo trata como si fuera la primera vez que accede.

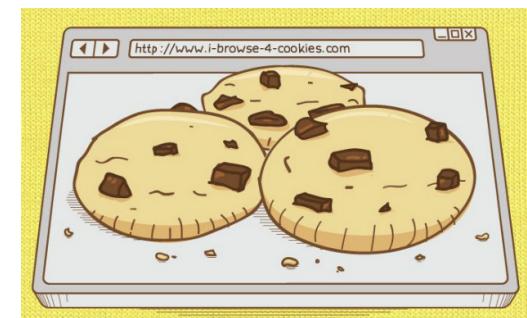
Lenguaje PHP: cookies

- Esto se solventa guardando datos sobre el cliente mediante alguno es estos mecanismos:

- **Cookies**: en el client-side
- **Sesiones**, logs y BBDDs: en el server-side



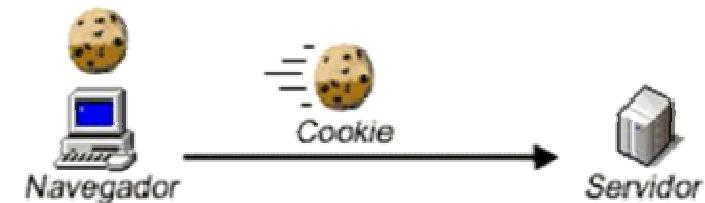
- Una **cookie** es información sobre el cliente (las preferencias del usuario) que se almacena en el propio equipo del cliente.
- El navegador se encarga de almacenarla en el disco del usuario a petición de:
 - La web del cliente: mediante **JavaScript**
 - El servidor web: mediante **PHP**



Lenguaje PHP: cookies

- El **servidor puede recuperar** en cualquier momento la información almacenada en la cookie del equipo del cliente.
- Esta información puede ser usada por el servidor para **personalizar las futuras visitas** de ese cliente a la web.
- Las cookies se almacenan de manera distinta según el navegador usado por el cliente para visitar la web.

Las cookies son independientes (propias) en cada navegador, pues el servidor web los considera clientes HTTP distintos aunque se conecten a través de la misma dirección IP.



Cookies	
Name	Value
_utma	208036121926835014...
_utmc	208036121926835014...
_utmz	79472861...
gravatar	melianabogados...
wordpress_e	1b53979227...
wordpress_lo	in_e124e1...
wordpress_test_	UnionAu...
wp-settings-2	WP+...
wp-settings-time-2	1408827702
wpcm-from-email	1

Lenguaje PHP: cookies y los navegadores



Screenshot of Mozilla Firefox's Privacy settings and Cookies dialog.

Privacy settings:

- Tracking: Use Tracking Protection is checked.
- History: Remember history is checked.
- Location Bar: History, Bookmarks, Open tabs are checked.

Cookies dialog:

- Search: []
- The following cookies are stored on your computer:

 - 01iftt.com
 - 148927072.log.optimizely.com
 - 1x.io
 - 3lift.com
 - abestserver.com
 - acloudweb.com

Screenshot of Mozilla Firefox's Application Basics information page.

Name	Firefox
Version	48.0.2
Build ID	20160823121617
Update History	Show Update History
Update Channel	release
User Agent	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
OS	Windows_NT 6.1 x86
Profile Folder	Show Folder
Enabled Plugins	about:plugins
Build Configuration	about:buildconfig

Internet Explorer: archivos de texto independientes



Configuración de Internet e Historial

Página principal: http://www.google.es

Historial de exploración:

- Eliminar archivos temporales, historial, cookies, etc.
- Eliminar el historial de exploración al salir

Búsqueda:

- Cambiar las opciones predeterminadas de búsqueda.

Pestañas:

- Cambiar la forma en que las páginas web se muestran en las pestañas.

Apariencia:

- Colores, Idiomas, Fuentes

Archivos temporales de Internet:

Internet Explorer almacena copias de páginas web, imágenes y multimedia para un acceso posterior más rápido.

Comprobar si hay nuevas versiones de las páginas guardadas:

- Cada vez que visite la página web
- Cada vez que inicie Explorer
- Automáticamente (selecciónada)
- Nunca

Espacio en disco que se va a usar (8 a 1024 MB): 50

Ubicación actual: C:\Users\Juan\AppData\Local\Microsoft\Windows\Temporary Internet Files

Historial:

Especifique el número de días que Internet Explorar debe guardar la lista de sitios web visitados.

Conservar páginas en el historial por estos días: 20

Aceptar Cancelar

Google Chrome:

chrome://settings/clearBrowserData

Ajustes

Google Chrome puede utilizar servicios web para mejorar la experiencia de navegación de los usuarios. Puedes...

Borrar datos de navegación

Utilizar un servicio de predicciones para...

Utilizar un servicio de predicciones para...

Enviar a Google automáticamente información...

Obtener protección para ti y para tu dispositivo...

Utilizar un servicio web para revisar la ortografía...

Enviar automáticamente estadísticas de uso...

Enviar una solicitud de seguimiento con tu...

Historial de navegación - 23 elementos

Historial de descargas

Cookies y otros datos de sitios y de complementos

Archivos e imágenes almacenados en caché - menos de 400 MB

Contraseñas

Datos de Autocompletar formulario

Datos de aplicaciones alojadas

Licencias de contenido multimedia

Borrar datos de navegación Cancelar

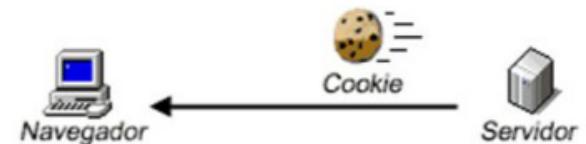
No se borrarán algunas opciones de configuración que pueden reflejar hábitos de navegación: [Más información](#)

Lenguaje PHP: manejo de cookies

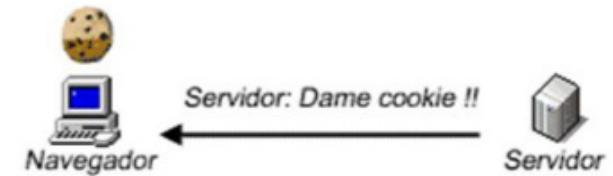
- En PHP se emplea la función `setcookie()` para **crear/asignar** valores a una cookie:

```
bool setcookie (string $nombreCookie, string $valor, int $caducidad)
```

Si la cookie no existe la crea con el valor;
y si existe sobreescribe su valor ("w" de fopen).



- Para **recuperar** (leer) el valor de una cookie se consulta el array superglobal asociativo `$_COOKIE['nombre_cookie']`



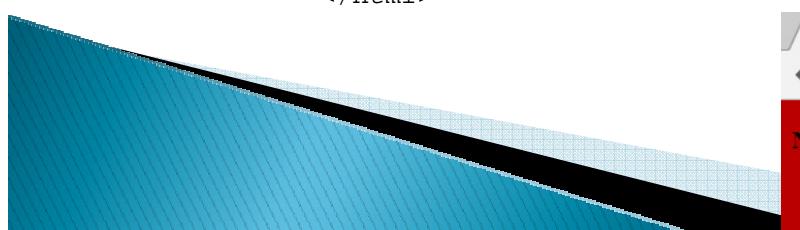
- Para **borrar** una cookie se le asigna una fecha de caducidad ya pasada, anterior a la actual (p.e. `time() - 3600`).

Lenguaje PHP: manejo de cookies

- **Ejemplo:** cookie que permite recordar si un usuario ya ha visitado una web; pero solo lo recuerda durante un minuto porque expira la cookie:

```
<?php // EL MANEJO DE LAS COOKIES DEBE SER LO PRIMERO DEL ARCHIVO
    if(!isset($_COOKIE['te_conozco'])){
        // Si no existe la crea para recordar que se ha visitado la web
        // La crea con una caducidad de un minuto --> se olvidará en breve
        // La fecha de expiración se expresa en segundos:
        $caducidad = time() + 60;
        $valor = "si";
        setcookie('te_conozco', $valor, $caducidad);
        $mensaje = "No te conozco, es la primera vez que me visitas.";
        $color = "#BB0000";
    } else {
        // Si ya existe la cookie es que la web ha sido visitada por ese navegador
        $mensaje = "Yo a ti ya te conozco.<br>Pero creo que te olvidaré en breve.";
        $color = "#00BB00";
    }
?>
<!DOCTYPE html>
<html>
<head> <title>Prueba de cookie</title> </head>
<body bgcolor=" <?php echo $color;?> ">
    <h3> <?php echo $mensaje; ?> </h3>
</body>
</html>
```

Las cookies se deben enviar antes que cualquier salida que genere la página (antes que <html>, <head> o un simple espacio en blanco).



Lenguaje PHP: manejo de cookies

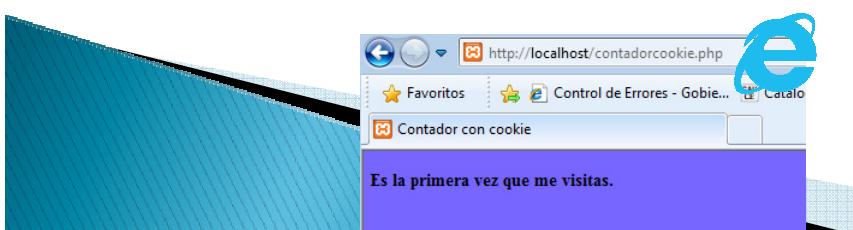
- **Ejemplo:** página contadorcookie.php capaz de contar cada vez que un usuario (un navegador) en particular la ha visitado.

```
<?php // EL MANEJO DE LAS COOKIES DEBE SER LO PRIMERO DEL ARCHIVO
if(!isset($_COOKIE['contador'])){
    // No existe la cookie --> nunca se ha visitado --> crear el contador a 1
    $contador = 1;
    $caducidad = time() + 365*24*60*60; // El contador caduca dentro de un año
    setcookie('contador', $contador, $caducidad);
    $mensaje = "Es la primera vez que me visitas.";
} else {
    // Ya existe la cookie --> ya se había visitado --> incrementar contador
    $contador = $_COOKIE['contador'];
    $contador++;
    $caducidad = time() + 365*24*60*60;
    setcookie('contador', $contador, $caducidad);
    $mensaje = "Me has visitado con este navegador ".$contador." veces.";
}
?>
<!DOCTYPE html>
<html>
<head> <title>Contador con cookie</title> </head>
<body bgcolor="#7766FF">
    <h3> <?php echo $mensaje; ?> </h3>
</body>
</html>
```

Se usa una cookie para almacenar en el equipo del cliente un número con su contador personal.

Prueba luego a resetear el contador → ¿esperar un año?
Mejor borra la cookie de tus navegadores y comprueba

Las cookies son únicas para cada navegador (el cliente HTTP); y no para cada usuario del SO, ni equipo (IP):



Lenguaje PHP: manejo de cookies

- **Actividad:** crea una página llamada contadorcompleto.php capaz de contar:
 - El número total de visitas registradas entre todos los usuarios.
 - El número de visitas que ha hecho cada usuario en particular.

Debes integrar en el mismo archivo PHP los dos tipos de contadores realizados:

- El contador global que usa un **fichero** almacenado en el servidor.
- El contador particular que usa una **cookie** almacenada en cada cliente.

Primero visitarla con un navegador



localhost/contadorcompleto.php

localhost/contadorcompleto.php

Esta página ha sido visitada 204 veces en total

La última vez fue a las 05:26:23 AM

204

Tú ya me has visitado con este navegador 159 veces

Y justo después con otro



localhost/contadorcompleto.php

localhost/contadorcompleto.php

Esta página ha sido visitada 205 veces en total

La última vez fue a las 05:26:25 AM

205

Tú ya me has visitado con este navegador 15 veces

Lenguaje PHP: manejo de cookies

- Es posible almacenar en una cookie más de un valor empleando un array asociativo para almacenar pares campo-valor:

```
setcookie ("nombreCookie[campo]", $valor, $cad);
```

```
<?php
    // Comprueba si ya existe la cookie
    if (!isset($_COOKIE['cookieArray'])){
        echo "<br>No existe la cookie.";
        echo "<br>Se creará un array con los valores:";
        echo "<br>temperatura=frio, estacion=invierno, meses=4";
        setcookie("cookieArray[temperatura]", "frio", time()+3600);
        setcookie("cookieArray[estacion]", "invierno", time()+3600);
        setcookie("cookieArray[meses]", 4, time()+3600);
    } else {
        echo "Sí existe la cookie.";
        echo "<br>Tiene guardados el siguiente array de valores:";
        $cookieArray = $_COOKIE['cookieArray'];
        foreach ($cookieArray as $campo=>$valor) {
            echo "<br>".$campo." = ".$valor;
        }
    }
?>
```

El nombre del campo de la cookie no va entrecomillado

También se puede leer el array asociativo así:

```
$cookieArray = $_COOKIE['cookieArray'];
$temperatura = $cookieArray['temperatura'];
$estacion = $cookieArray['estacion'];
$meses = $cookieArray['meses'];
```

Lenguaje PHP: manejo de cookies

- **Actividad:** crea una página llamada cookiearray.php que permita almacenar en una única cookie (en un array asociativo) varias preferencias del usuario:

- Su nombre.
- El color de fondo que prefiere para la web (rojo, verde o azul).
- El lenguaje elegido para el texto de la web (español o inglés).

Cuando el usuario cargue la página; si existe la cookie, la página se mostrará aplicando las preferencias del usuario que tiene almacenadas.

El usuario podrá cambiar sus preferencias y serán guardadas en la misma cookie.

Cookie array

localhost/cookiearray.php

Personalice el sitio web:

Nombre: NAVEGANTE | verde ▾ | inglés ▾ | GUARDAR

Hi NAVEGANTE,
This web saves your preferences in a cookie with:
your name, background color and language selected.

Al cambiar sus preferencias se almacenan en la cookie

Cookie array

localhost/cookiearray.php

Personalice el sitio web:

Nombre: NAVEGANTE | azul ▾ | español ▾ | GUARDAR

Hola NAVEGANTE,
Esto es una página que guarda tus preferencias en una cookie con:
tu nombre, color de fondo y el lenguaje seleccionados.

Modifica la cookie para que guarde también la fecha y hora del último acceso del usuario y se la muestre.