

# Actividades php

## Implantación de aplicaciones informáticas

- 1.- Crea una página php que contenga tu nombre y apellidos. Visualízalos en una tabla.
- 2.- Crea una tabla que contenga el sistema operativo del servidor donde se está ejecutando tu página, así como la versión de PHP.
- 3.- Crea un documento HTML que pase por sesión tres variables a otro: var1, var2 y var3. Comprueba qué ocurre en el directorio \easyPHP1-7\tmp
- 4.- Crea un documento que pase dos variables a otro mediante URL. Una de ellas debe ser el nombre y la otra los dos apellidos. Visualízalo en una tabla html.
- 5.- Con el documento de la actividad 3 pídele a un compañero que lo realice desde su máquina. ¿Dónde se está creando el fichero sesión en tu máquina o en la suya?
- 6.- Utilizando el menú de la actividad 19 de CSS, introduce código PHP para que pase la variable \$valor a cuatro ficheros php (Inicio, Información, contacto y enlaces). Recupera en cada uno de esos ficheros la variable. ¿Qué método has utilizado? ¿Podrías utilizar cualquiera de los otros dos métodos vistos en clase?
- 7.- Modifica el ejercicio 47 de XHTML para que envíe los datos a recibo\_form.php. Muestra, en este último, los valores seleccionados por el usuario. ¿Encuentras algún problema con estudios? Si el usuario ha utilizado retornos de carro en “otra información” ¿La visualizas correctamente? Utiliza ahora la función nl2br(variable)? ¿Qué pasa ahora?
- 8.- Crea formulario.html que contenga un formulario html que solicite al usuario los siguientes datos:
  - Nombre (texto)
  - Apellidos(texto)
  - Sexo (radio)
  - Sistema Operativo Preferido (select)
  - Si tiene coche (checkbox)
  - Otros comentarios (textarea)Debes enviar estos datos al documento tusdatos.php que muestre los datos introducidos por el usuario en el formulario anterior.
- 9.- Crea un fichero php que muestre el directorio [c:\](#) de tu máquina. ¿Cómo harías para poder realizarlo tanto si el sistema operativo es Windows como si es GNU/Linux?
- 10.- Comprueba la diferencia de las comillas simples y las dobles.
- 11.- Realiza un script PHP que muestre un error si la variable \$b no existe. En caso contrario mostrará un mensaje indicando que sí existe.
- 12.- Crea una actividad para probar el funcionamiento del operador de supresión de error. Indica varios casos en los que utilizarías este operador.
- 13.- Modifica el fichero recibo\_form.php de la actividad 7 para que, si el usuario no marca nada aparezca un “este campo está vacío” en vez del error.
- 14.- Crea un formulario en el que se solicite un nombre. Al pulsar el botón de submit tiene que mostrar el mismo formulario (debe ser el mismo fichero también) con el nombre introducido. NOTA: para hacer referencia al mismo script -->\$\_SERVER["PHP\_SELF"]
- 15.- Crea dos documentos con extensión inc: cabecera.inc y pie.inc. En el primero deben aparecer

las etiquetas del principio de un documento html. En el segundo las de final de un documento html. Inclúyelas en un fichero php con include.

Cambia el nombre de los ficheros .inc y llama al fichero php ¿Qué ocurre, ha mostrado algo?

**16.-** Realiza la actividad anterior con require(). Observa los parecidos y diferencias.

**17.-** Crea los ficheros fichero1.inc, fichero2.inc, fichero3.inc y fichero4.inc que contengan el factorial de 10,9,8 y 7 respectivamente (entendiendo por contener el factorial que muestren la siguiente expresión  $10! = 10 * 9 * 8 * 7 * 6 * 5 * 4 * 3 * 2 * 1 = 3628800$ ) en los ficheros inc deben calcularse los factoriales, no mostrar únicamente el texto.

Estos ficheros serán incluidos dentro de un bucle por un fichero php.

Debes crear, de manera elegante y vistosa (CSS incluidos) un fichero de cabecera de documento y otro de pie de documento. Ponte de acuerdo con tus compañeros porque debe ser intercambiable.

**18.-** Crea el script php es\_primo.inc que devuelva una variable booleana con verdadero o falso según corresponda si el valor contenido en la variable \$numero del documento que incluye a es\_primo.in es primo o no. Debes controlar que exista la variable \$numero en el documento "llamante".

**19.-** Observa los ficheros index.php y /include/mambo.php de joomla.

**20.-** Modifica el fichero datos.php sustituyendo, donde sea posible, los echo por el operador <<<.

**21.-** Crea un formulario HTML que solicite una cadena con delimitador. Envía la cadena y el delimitador (que también solicitará el formulario) a un fichero php. Se mostrará las cadenas que forman la introducida por el usuario en una tabla HTML.

**22.-** Crea un formulario HTML que solicite dos cadenas. Las pasará a un fichero php que las comparará y mostrará cuál de ellas es la mayor.

**23.-** Realiza un fichero php que muestre el efecto de las funciones rtrim, ltrim y trim.

**24.-** Crea en PHP el array escalar automoviles. Debe contener 5 modelos de coches. Se recorrerá el array mostrando su contenido. Se introducirá un nuevo modelo y se volverá a recorrer el array.

**25.-** Crea un array asociativo que contenga los siguientes campos: modelo, marca, fecha, cilindrada, motor, potencia. Este array contendrá los datos de un solo coche. Recórrelo mostrando su contenido en una tabla HTML.

**26.-** ¿Qué realiza el script PHP de la imagen siguiente? Contiene errores ¿Cómo los solucionarías?

```

<?php
$matriz[0]=array("director"=>"Francis Ford Coppola", "pelicula"=>"El padrino II", "calificacion"=>"buenísima")
$matriz[1]=array("director"=>"David Lean", "pelicula"=>"Lawrence de Arabia", "calificacion"=>"muy buena");
foreach ($matriz as $valor) {
    echo $valor["director"];
    echo "<br>";
    echo $valor["pelicula"];
    echo "<br>";
    echo $valor["calificacion"];
    echo "<br>";
}
$matriz[]=array("Torrente", "Santiago Segura", "regularcilla");
foreach ($matriz as $valor) {
    echo $valor
    echo "<br>";
    ;
}
?>

```

**27.-** ¿Cómo crearías un array como el de la actividad 25 que contuviera dos coches? ¿Cómo lo recorrerías? ¿Cómo añadirías un nuevo modelo de coche después de haberlo recorrido?

**28.-** Debes crear un script PHP que recorra un array que contenga títulos de discos, autores y precios. Los debe mostrar en una tabla HTML con un total de precio en la última fila así como un total de los libros mostrados obtenido con la función count().

**29.-** Crea el fichero datos.inc que contenga un script PHP que defina un array con marcas de coches.

Incluye este fichero en un fichero script PHP que muestre un formulario con un campo select cuyas opciones estén tomadas del array definido en datos.inc.

**30.-** Crea un array que contenga libros, autores y precios. Debe estar incluido en un fichero datos.inc.

Crea un formulario que pida un nombre de libro y envíe los datos a buscolibro.php que debe recorrer el array y devolvernos si el libro existe o no. Caso de existir nos mostrará el autor, y el precio.

Si no existe nos enviará al formulario de petición con un mensaje indicando que no se ha encontrado.

**31.-** Crea un formulario que solicite el nombre de usuario y contraseña. Debe existir usuario.inc que contenga la tabla de usuarios con login y password. El password debe estar encriptado con la semilla que decidas.

Cuando el usuario pulse el botón submit se debe comprobar si su login existe en la tabla y si el password es correcto. Si es correcto mostrará una página de bienvenida. Si no lo es volverá a solicitar el login y la contraseña repitiendo el proceso. Esta vez, en el documento de formulario, debe aparecer un mensaje indicando si es el usuario el que no existe o si la contraseña es incorrecta.

**32.-** Modifica el ejercicio 30 para que contenga la función busco() que devuelva un array con los valores encontrados, o vacío si no se hubiera encontrado nada.

**33.-** Incluye la función busco() del ejercicio anterior en un fichero include.

**34.-** Realiza un script php que cree el fichero prueba1.txt con el contenido del un input tipo texto pasado desde otro documento html. Si el tamaño de la cadena no es de 100 caracteres tendrás que añadir espacios al final hasta alcanzarlos. El último carácter que debes introducir es “\n” (fin de línea).

(Nota: vamos a utilizar la función `str_repeat($caracter, vecesqueserepite)` para rellenar hasta 100).

**35.-** En el fichero creado en la actividad anterior, y recogiendo también un input tipo texto, vamos a añadir una línea respetando la anterior.

**36.-** Crea un script php que lea el fichero prueba1.txt con la función `fgets` y lo muestre en un tabla HTML.

**37.-** Modifica el script de la actividad 36 sustituyendo la función `fgets` por `file`.

**38.-** Vamos a crear un programa PHP que nos permita añadir, modificar, borrar y listar una agenda de teléfonos.

Vamos a realizarla por partes:

**1°.-** Creamos un AGENDA.INC que, de momento, va a contener las siguientes constantes:

SEP\_CAMPOS con valor ;

SEP\_REGISTROS con valor \n

FICH\_DATOS con valor **agenda.txt**

CAMPONOMBRE con valor **0**

CAMPOAPELLIDOS con valor **1**

CAMPOTELEFONO con valor **2**

CAMPOPROVINCIA con valor **3**

**2°.-** Creamos un DATOSAGENDA.INC que contenga un array de 10 elementos, cada uno de los cuales contendrá: nombre, apellidos, teléfono y provincia.

**3°.-** Vamos a crear la función **anadeRegistro(\$df,\$matriz)** (siendo df el descriptor de fichero, y matriz una array escalar que en la posición 0 contenga nombre, en la posición 1 apellidos, en la posición 2 teléfono, y en la 3 provincia). La función cogerá la matriz y la añadirá en una nueva línea al fichero referenciado por \$df. Este fichero debe estar previamente abierto para escritura. Cada uno de los campos se separará del siguiente por SEP\_CAMPOS. Y al final del array se añadirá SEP\_REGISTROS.

En el último campo no debe introducirse el separador de campos.

Esta función la añadiremos a AGENDA.INC.

**4°.-** Crearemos el script CreaAgenda.php que leerá el array contenido en DATOSAGENDA.INC y lo añadirá (utilizando `anadeRegistro`) a FICH\_DATOS.

Al final del proceso nos mostrará un mensaje indicando que se ha creado el fichero de datos correctamente.

**5°.-** Realizaremos el script Unregistromas.php al que se le pasarán nombre, Apellidos, teléfono y provincia. Generará con estos datos un array escalar y lo añadirá, utilizando `anadeRegistro` a FICH\_DATOS.

**6°.-** Vamos a crear la función **devuelveRegistros(\$campo,\$valor, \$fichero)**. La pasaremos como parámetros la posición del campo por el que queremos buscar (0,1,2,3), el valor que queremos encontrar, y el descriptor del fichero. Con el valor o valores encontrados generará un array escalar que contendrá en cada celda otro array

con NOMBRE, APELLIDOS, TELÉFONO Y PROVINCIA de los datos encontrados. Caso de no haber encontrado ninguno devolverá la tabla a “”.  
Añadiremos la función a AGENDA.INC.

7º.- Utilizando devuelveRegistros, crearemos Busca.php que, pasándole el campo por el que queremos buscar y el valor, devolverá un listado con los elementos encontrados.

8º.- Vamos a crear la función **borraRegistros(\$campo,\$valor, \$fichero)** donde \$campo será el valor de la constante (NOMBRE, APELLIDOS, PROVINCIA O TELÉFONO), \$valor es el contenido que se querrá borrar, y \$fichero el nombre del fichero (ojo, no el descriptor) del que queremos borrar los registros encontrados. Esta función recorrerá el \$fichero y obtendrá los registros que contengan en el campo indicado el \$valor. Modificará el fichero para que no aparezcan los valores. Devolverá la cantidad de registros eliminados.  
Añadiremos borraRegistros en AGENDA.INC.

9º.- Crearemos el script BORRA.PHP que hará una llamada a borraRegistros eliminando del fichero FICH\_DATOS los registros que contengan el valor indicado en el campo correspondiente.

39.- Crea el script sesionuno.php que cree una sesión y le añada la variable campo1 con el valor “sesiónuno”. Debe existir un href que “apunte” a sesiondos.php.

Crea el script sesiondos.php que obtenga el valor del campo1 creado en sesionuno.php, lo muestre en el navegador y añada campo2 con el valor “sesióndos”. Debe apuntar a sesiontres.php.

En sesióntres.php deben mostrarse campo1 y campo2. Se añadirá campo3 con “sesióntres”. Debe existir un “apunte” a sesiónuno.php.

Cuando el script llegue de nuevo a sesiónuno.php y exista campo3 con el valor “sesióntres” mostrará un mensaje indicando que ya se ha realizado todo el circuito.

40.- Modifica los script de la actividad 39 para que el recorrido sólo pueda hacerse tres veces.

41.- Asigna, en sesionuno.php el identificador de sesión que desees. Muestra el identificador, obtenido con la función session\_id() en los demás script.

42.- Modifica el script sesiontres.php para que apunte, en vez de a sesionuno.php a sesioncuatro.php. Sesioncuatro.php debe mostrar todos los valores de la sesión creados anteriormente y añadir un campo4 con el valor “sesióncuatro”. Este script apuntará a sesionuno.php.

43.- Asigna en sesiondos.php un nuevo identificador de sesión. Debes poder leer los valores asignados en sesionuno.php en sesiontres.php, y los valores de sesiondos.php en sesioncuatro.php, ¿Cómo lo harías?

44.- Crea un script llamado informacion.php con la función phpinfo(). Obtén el directorio donde se guardan los ficheros de sesión de tu instalación, así como el camino del fichero php.ini. ¿Cuántos minutos de vida tienen las sesiones que se crean en tu servidor?

45.- Modifica los script de la actividad 42 para que se destruya la sesión al realizar los tres recorridos. Observa en el directorio de sesiones cómo van evolucionando los ficheros.

46.- Pon la variable enable-trans-id del fichero php.ini a 1. Reinicia Apache. Ejecuta el script de la actividad anterior. Observa la ventana Navigator de tu navegador ¿Qué está ocurriendo con la URL?

Cambialá de nuevo a 0. Vuelve a reiniciar Apache.

**47.-** Describe el proceso de vida de una sesión: Qué ficheros se crean y dónde; cómo se asignan valores y dónde se guardan; cómo se pueden añadir nuevos valores, modificar y acceder a los existentes, etc.

**48.-** Modifica la aplicación realizada en examen parcial de php para que sólo permita tres intentos de login por sesión.

**49.-** Modifica la aplicación que realizaste con el resto de compañeros de clase para que guarde en un fichero log los accesos a tu página por parte de los distintos usuarios.

Una vez realizado esto, muestra las veces que un usuario ha accedido a tu página.

Utiliza la variable `$_SERVER["REMOTE_ADDR"]` para visualizar la IP desde la que está accediendo.

¿Se te ocurre alguna forma de guardar también la página que visita después de estar en la tuya?

**50.-** Realización del foro con MySql y PHP.

Formularios: primerforum.php  
anadecomentario.php  
anaderespuesta.php  
configtema.php

Tablas:

comentarios

id  
comentario  
autor  
id\_tema  
texto

respuestas

id\_res  
id  
autor\_res  
respuesta

tema

id\_tema  
tema

**51.-** Realiza un documento html con un formulario que contenga:

Dos botones submit: uno borrar  
otro añadir  
un textarea

Los valores de estos campos se pasarán por `$_POST` a un script que mostrará el valor del botón pulsado, y el valor del textarea con `addslashes($_POST["campotextarea"])` y `htmlspecialchars($_POST["campotextarea"])`

**52.-** Crea una estructura de tablas que pueda contener un menú con una profundidad de dos opciones como mucho. En el menú principal el elemento podrá ser o aplicación o submenú. Caso de ser aplicación lanzará el script php correspondiente. En el caso de ser un submenú mostrará las opciones que forman parte de él.

Este menú deberá estar integrado en el foro de la actividad 50.

**53.-** Creamos la función PHP “muestrareloj()” con los siguientes argumentos:

- \$nombreform
- \$campo
- \$caja

Esta función deberá crear un formulario dentro del div \$caja. Este formulario contendrá un campo tipo texto name=\$campo con clase=horas.

Dentro de la función muestrareloj() se creará la función JavaScript reloj() que, cada minuto, actualizará la hora en formato digital mostrándola en el campo \$campo.

Debemos crear la CSS que contenga la #\$caja y .horas.

Introduce esta función en un include y utilízala en primerforum.php