

## ACTIVIDAD 3

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

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

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

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

Vamos a realizarla por partes:

1º.- 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.

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

8°.- 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.

5.- 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 la 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.





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


7.- Asigna, en `sesionuno.php` el identificador de sesión que desees. Muestra el identificador, obtenido con la función `session_id()` en los demás script.


8.- 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`.

9.- 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?

10.- 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?





11.- 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.

12.- 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.

13.- 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.

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

15.- 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?

## 16.- Realización del foro con MySql y PHP.

Formularios: `primerforum.php`

`anadecomentario.php` `anaderespuesta.php` `configtema.php`

Tablas:

Comentarios

Respuestas

Id comentario autor id\_tema texto

Id\_res id

Autor\_res respuesta

tema

## 17.- Realiza un documento html con un formulario que contenga:

Dos botones submit: uno borrar

Otro añadir

Id\_tema tema

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"])`

18.- 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.

19.- 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`