

Tema 10. FITXERS

La informació que es necessita guardar de forma permanent està emmagatzemada en fitxers. Per això és bastant habitual que les aplicacions, siguin del tipus i entorn que siguin, realitzin operacions amb fitxers del sistema local. Per exemple, posar un simple comptador de visites en una pàgina es pot fer mitjançant un fitxer on es guarden el total de visites rebudes per a que, quan arribi la següent, s'obri el fitxer, es llegeixi el número que conté i es modifiqui convenientment.

Per a poder implementar aquest tipus de tasques, PHP disposa de funcions predefinides per a la utilització i manegament de fitxers. Gràcies a aquestes funcions es podrà accedir a un fitxer creat previament, es podran consultar les dades que contingui, afegir-ne de noves, modificar-les o bé, esborrar-les.

10.1 OPERACIONS AMB FITXERS (NIVELL INTERN)

PHP ofereix funcions que permeten realitzar les operacions típiques de fitxers: obrir, tancar, llegir, escriure, recórrer...

10.1.1 OBRIR UN FITXER

```
fopen(nom, modeObertura [, ruta_cerca])
```

PHP permet obrir fitxers locals o remots. Que el fitxer estigui en el disc dur d'una màquina local o en una altra dependrà de com estigui format el paràmetre `nom`: si comença per **http://** o **ftp://**, indicarà que és un fitxer *remot* i que s'accedirà a ell mitjançant el protocol corresponent. En cas contrari, es tractarà d'un fitxer que PHP cercarà en el sistema de fitxers del servidor, raó per la qual s'haurà de tenir molta cura a l'hora d'indicar el camí o ruta d'accés a l'arxiu que es vol.

Si PHP no pot trobar el fitxer a la ruta especificada per `nom` i el paràmetre `ruta_cerca` està a 1, llavors PHP el cercarà a més als directoris que estiguessin definits a la directiva `include_path` del fitxer de configuració `php.ini`.

El paràmetre `modeObertura` pot tenir els valors de la següent taula:

Valor	Significat
r	Mode només lectura. El punter es col·loca a l'inici del fitxer.
r+	Mode de lectura i escriptura. El punter es col·loca a l'inici del fitxer.
w	Mode de només escriptura. Si no existeix el fitxer, es crea. Si ja existeix, s'esborra tot el contingut.
w+	Mode de lectura i escriptura. Si no existeix el fitxer, es crea. Si ja existeix, s'esborra tot el contingut.
a	Mode de només escriptura. Si no existeix el fitxer, es crea. Si ja existeix, el punter es col·loca al final del fitxer per a afegir les dades.
a+	Mode de lectura i escriptura. Si no existeix el fitxer, es crea. Si ja existeix, el punter es col·loca al final del fitxer per a afegir les dades.

La funció `fopen` retorna un descriptor del fitxer, és a dir, el *canal* pel qual es podrà accedir a ell; per tant, els descriptors seran imprescindibles a l'hora de realitzar operacions sobre fitxers tals com lectura, escriptura, tancat, etc... Per tant, hauran d'existir tantes variables que emmagatzemin descriptors com fitxers oberts simultàniament es vulguin tenir.

En el cas que el fitxer no es pugui obrir per la causa que sigui, `fopen` retornarà `FALSE`.

La forma més habitual d'obrir un fitxer és:

```
<?php
```

```
    if(! $descriptor = fopen("nom_fitxer.ext","r")){
        echo "*** ERROR: el fitxer no es pot obrir. <br>";
    } else{
        echo "Es pot accedir al fitxer mitjançant '$descriptor'.<br>";
    }
}
```

```
?>
```

10.1.2 TANCAR UN FITXER

```
fclose(descriptor_fitxer)
```

Amb la funció `fclose` es tanca el fitxer que està referenciat pel descriptor que es passa com a argument. Evidentment, aquest descriptor és el retornat per la funció `fopen`.

`fclose` retorna `TRUE` si el fitxer es tanca correctament i `FALSE` en cas contrari.

Donat que un fitxer obert consumeix recursos del sistema, és convenient tancar aquells que es preveu que no s'han d'utilitzar més. En el cas que no es tanqui expressament un fitxer, PHP ho farà automàticament en terminar d'executar l'script.

10.1.3 LECTURA DES D'UN FITXER

PHP té diferents i variades formes de llegir el contingut d'un fitxer:

- `fgetc(descriptor_fitxer)`: retorna un caràcter del fitxer. Si s'ha arribat al final del fitxer retorna `FALSE`.
- `fgets(descriptor_fitxer[,total_cars_a_llegir])`: retorna una cadena de `total_cars_a_llegir - 1` caràcters o de menor longitud si s'ha trobat un canvi de línia (que s'inclouria a la cadena a retornar) o s'ha arribat al final del fitxer.
- `fgetss(descriptor_fitxer, tamany, etiq_permeses)`: és idèntica a `fgets`, excepte que treu del text llegit les etiquetes HTML que hagi trobat. En el paràmetre `etiq_permeses` es poden indicar les etiquetes HTML que sí es volen mantenir.
- `fread(descriptor_fitxer, total_cars_a_llegir)`: és idèntica a `fgets`, excepte que no deixa de llegir quan es troba un canvi de línia i en que retorna exactament `total_cars_a_llegir` (si els té).
- `fscanf(descriptor_fitxer, format[, var1...])`: segons llegeixi l'entrada, la processa amb el format especificat en el paràmetre `format` per a assignar-la convenientment a es variables que s'indiquen.
- `feof(descriptor_fitxer)`: no és una funció de lectura pròpiament dita, però és molt útil quan es llegeixen fitxers: indica si s'ha arribat al final (no queden més dades per llegir) o no.

```
        while (!feof($descriptor)){
            .....
        }
```
- `file(nom_fitxer [, ruta_cerca])`: llegeix tot el contingut d'un fitxer i el retorna en forma d'*array*: una línia a cada posició. Aquesta funció utilitza el nom del fitxer i no un descriptor.
- `readfile(nom_fitxer [, ruta_cerca])`: llegeix el contingut d'un fitxer i el mostra per la sortida estàndard. Retorna el total de caràcters llegits.

10.1.4 RECÒRRER UN FITXER

En moltes ocasions es té la necessitat de col·locar-se en una determinada part d'un fitxer per a poder llegir o escriure cadenes de text a partir d'aquesta posició. Per a poder realitzar aquesta funcionalitat PHP ofereix les següents funcions:

- `rewind(descriptor)`: situa el punter de lectura/escriptura al principi del fitxer.
- `fseek(descriptor, desplaçament [, des_don])`: desplaça la posició del punter de lectura/escriptura `desplaçament` posicions. El tercer paràmetre pot tenir els valors **SEEK_SET**, **SEEK_CUR** i **SEEK_END**, que significarà que els desplaçaments són relatius al principi del fitxer, a la posició actual del punter o al final del fitxer (llavors `desplaçament` serà negatiu), respectivament.
- `ftell(descriptor)`: retorna la posició del punter.

10.1.5 ESCRIPTURA EN UN FITXER

Per a escriure, es disposa de 2 funcions que es poden utilitzar de forma indistinta, ja que es poden considerar àlies l'una de l'altra:

```
fwrite(descriptor, cadena[, total_caracters]);
```

```
fputs(descriptor, cadena[, total_caracters]);
```

Escriuen la cadena (completa) passada com a paràmetre. Si s'utilitza el tercer paràmetre, no més s'escriuran els `total_caracters` indicats. Retorna el total de caràcters escrits o **FALSE** en cas de produir-se algun error.

En general, donat que són molt lentes totes les operacions en les quals està impliat un accés al sistema de fitxers, es recorre a l'escriptura mitjançant buffers per a fer-les més eficients. Això vol dir que l'escriptura no es fa efectiva en el fitxer fins que no s'omple aquesta memòria intermitja. Per defecte, el buffer d'escriptura de PHP té un tamany de 8 Kb, però si en algun moment es vol canviar aquesta xifra per una altra, es pot fer amb la funció `set_file_buffer()`:

```
set_file_buffer(descriptor,buffer_tamany);
```

10.2 INFORMACIÓ SOBRE FITXERS

- `file_exists(nom_fitxer)`: comprova si el fitxer existeix. Retorna TRUE si existeix i FALSE en cas contrari.
- `is_file(nom)`: retorna TRUE si el nom indicat és un fitxer i FALSE en cas contrari.
- `is_dir(nom)`: retorna TRUE si el nom indicat és un directori i FALSE en el cas que no ho sigui o no existeixi.
- `is_executable(nom_fitxer)`: retorna TRUE si el nom indicat és un fitxer que té permisos d'execució (LINUX) o és un fitxer executable (extensió EXE, COM o BAT).

10.3 OPERACIONS AMB FITXERS (NIVELL EXTERN)

- `copy(origen, destí)`: copia un fitxer en un destí. Retorna TRUE si no hi ha hagut cap error.
- `rename(nom_original, nom_final)`: canvia el nom a un fitxer o directori.

10.4 MANEGAMENT DE DIRECTORIS

Existeixen funcions específiques per al manegament i utilització de directoris. Amb elles es poden crear, esborrar, llegir les entrades que té un directori, avariguar en quin directori ens trobem, canviar a un altre, etc...

- `opendir(nom_directori)`: obre el directori indicat com a paràmetre. Retorna el descriptor del directori.

Si s'escriuen scripts per a plataformes Windows s'ha de tenir en compte en benefici de la portabilitat: utilitzar el caràcter "/" com a separador de directoris i no fer referència a la unitat d'emmagatzematge.

```
<?php
    opendir("/mis documentos/llibre");
?>
```

- `readdir(descriptor)`: llegeix una entrada del directori a partir del descriptor retornat per `opendir`.
- `closedir(descriptor)`: tanca el directori indicat pel descriptor.
- `rewinddir(descriptor)`: situa el punter de lectura al principi del directori.
- `getcwd()`: retorna el directori actual de treball (també anomenat actiu) de l'script.
- `chdir(nou_directori)`: estableix `nou_directori` com a nou directori de treball. Retorna FALSE en cas d'error.

Pe. A continuació teniu un exemple d'un navegador d'arxius senzill.

```

<?
if (isset($_GET['nuevo_dir']))
    $nuevo_dir=$_GET['nuevo_dir'];
else { // es la primera vez que se ejecuta el script
    $nuevo_dir=getcwd();
    if (isset($_SERVER["WINDIR"])); {
        $nuevo_dir=substr($nuevo_dir, 2); // quitamos las 2 letras de la unidad
        $nuevo_dir=strtr($nuevo_dir, "\\ ", "/");
    }
}
echo "<h2>CARPETA: <i>$nuevo_dir</i></h2>";

if (!$df_dir = opendir($nuevo_dir))
    die("<h3>*** ERROR: no se ha podido entrar en ($nuevo_dir)</h3>");

while (($item = readdir($df_dir)) !== false) {
    if ($item == ".") continue;

    if (es_directorio($nuevo_dir, $item)) {
        pon_url($nuevo_dir, $item);
    } else
        echo "<img src='/icons/generic.gif'>$item<br>";
}
closedir($df_dir);

function pon_url($un_dir, $un_item)
{
    $ini_etiq="<a href='$_SERVER["PHP_SELF"]"?nuevo_dir";
    $fin_etiq="</a><br>";

    if ($un_item == "..") {
        if (substr_count($un_dir, "/") >= 1) {
            $un_dir=strtr(dirname($un_dir), "\\ ", "/"); // Win devuelve '/'
            echo "<img src='/icons/back.gif'>$ini_etiq=$un_dir'><font
size=+2>..</font>$fin_etiq";
        }
    } else {
        if ($un_dir=="")
            echo "<img
src='/icons/folder.gif'>$ini_etiq=$un_item'>$un_item$fin_etiq";
        else
            echo "<img
src='/icons/folder.gif'>$ini_etiq=$un_dir/$un_item'>$un_item$fin_etiq";
    }
}

function es_directorio($un_dir, $un_item)
{ // en windows no puede haber un nombre de directorio con dos '/' : '/'
mal
    if ($un_dir == '/')
        $fich_a_preguntar="/$un_item";
    else
        $fich_a_preguntar="$un_dir/$un_item";
    return (is_dir("$fich_a_preguntar"));
}
?>

```

10.5 OPERACIONS AMB DIRECTORIS

Algunes funcions bàsiques per treballar amb directoris:

- `mkdir(nom_directori, permisos)`: crea el directori amb els permisos indicats.
- `rmdir(nom_directori)`: esborra el directori indicat.

10.6 ALTRES FUNCIONS

Aquestes són funcions que d'alguna manera estan relacionades amb els fitxers:

- `basename(ruta)`: aquesta no és una funció de fitxers pròpiament dita sinó que és de cadenes de caràcters, ja que tant el paràmetre com el resultat de la funció és un *string*; donada una cadena de caràcters que defineixen una ruta o posició d'un fitxer en el disc dur, retorna els caràcters que estan després de l'últim caràcter que s'utilitza com a separador.
Pe. si li apliquem a aquesta funció la cadena */hola/que/tal* retornaria */tal*
- `dirname(ruta)`: fa l'invers de la funció `basename`, és a dir, retorna la ruta de directoris i subdirectoris on està ubicat el fitxer: els directoris pels quals s'ha de passar per a trobar-lo.
Pe. si li apliquem a aquesta funció la cadena */hola/que/tal* retornaria */hola/que*
- `disk_free_space(nom_directori)`: pren una ruta a un directori i retorna l'espai lliure en octets (*bytes*) disponibles en el sistema de fitxers que conté al directori.
- `disk_total_space(nom_directori)`: pren una ruta a un directori i retorna el tamany del sistema de fitxers on està situat el directori.

10.7 TRANSFERÈNCIA DE FITXERS ENTRE CLIENT I SERVIDOR

10.7.1 PUJAR FITXERS AL SERVIDOR

No és estrany trobar-se amb pàgines que permeten transmetre fitxers des del client al servidor, és a dir, el que es coneix com *pujar* fitxers o fer *uploads*. Casos com aquests es troben, per exemple, en pàgines on s'ha d'omplir un currículum vitae i un dels camps a omplir és incloure una fotografia de la persona interessada. També en els clients de correu hi ha la possibilitat de poder compondre un missatge adjuntant fitxers.

El problema de pujar fitxers al servidor es divideix en 2 escenaris: un és la pàgina HTML on l'usuari indicarà (mitjançant etiquetes) quin fitxer del seu disc local vol pujar; l'altre es troba en la part de programació: gràcies a la pàgina HTML (en realitat, gràcies al protocol HTTP), el fitxer ja ha estat transferit al servidor i ara el que s'ha de fer és poder accedir a ell per a realitzar les tasques que es considerin oportunes.

A la pàgina HTML només es necessiten 2 etiquetes: la etiqueta **<input>** (en realitat tantes com fitxers es vulguin pujar), gràcies a la qual el navegador mostrarà una capsa i un botó de navegació (*browsing*), per a que l'usuari seleccioni el fitxer que vol.

```
<input type='file' name='nom_fitxer_a_pujar'>
```

La segona etiqueta necessària és la que permet construir un formulari (**form**) però amb una sèrie de clàusules addicionals:

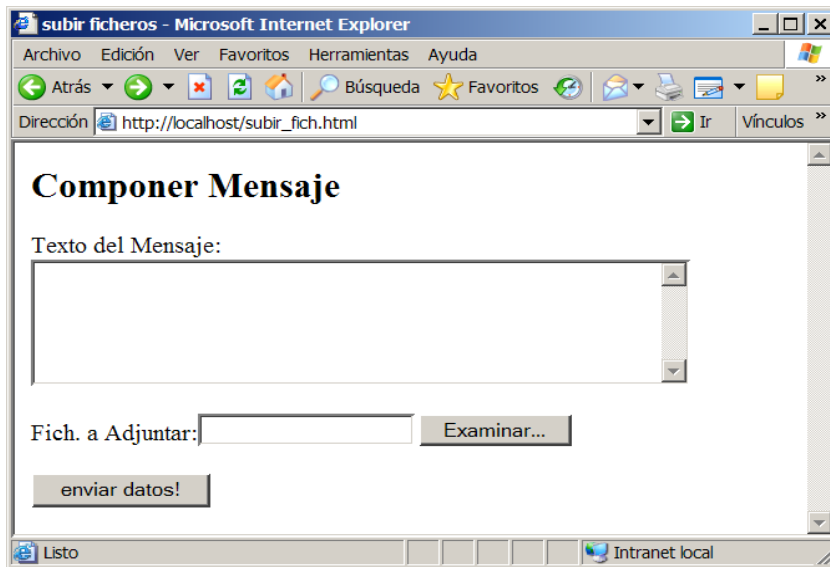
```
<form action='pujar_fitxer.php'  
      method='post'  
      enctype='multipart/form-data'>
```

S'ha d'indicar que el mètode de transferència de les dades del formulari és mitjançant POST (viatgen juntament amb la pàgina) i que viatjaran codificats com *multipart/form-data*.

Opcionalment es pot indicar el tamany màxim que podrà tenir el fitxer per a poder ser enviat amb un camp de dades ocult (llavors es tindrien 3 etiquetes). Aquest camp s'ha de dir **max_file_size**, en cas de no dir-li així s'agafarà la directiva **upload_max_filesize**.

```
<input type='hidden' name='max_file_size' value='1000'>
```


Pe. A continuació teniu un exemple d'una pàgina HTML des de la que es podria enviar un missatge juntament amb un fitxer adjunt:



```
<HTML>
<HEAD>
<TITLE> subir ficheros </TITLE>
</HEAD>
<BODY>
  <h2>Componer Mensaje</h2>
  <FORM method='post' action='subir_fich.php'
        enctype='multipart/form-data'>
    <INPUT type='hidden' name='max_file_size' value='250000'>

    Texto del Mensaje:<br>
    <textarea cols='50' rows='5' name=.texto.>
    </textarea>
    <br><br>Fich. a Adjuntar:<input type='file' name='f_adjunto'>
    <br><br><input type='submit' value='enviar datos!'>
  </FORM>
</BODY>
</HTML>
```

Després de premer el botó d'enviar, el fitxer es troba en el servidor. Evidentment, el fitxer enviat no està en un directori a l'atzar: s'emmagatzema en el directori que s'especifica en el fitxer `php.ini` mitjançant la directiva `upload_tmp_dir`.

Per a poder treballar amb els fitxers enviats així, PHP proporciona l'*array* associatiu **\$_FILES**. Aquest array té dos dimensions: a la primera columna s'indiquen els fitxers que han estat pujats utilitzant com a clau els identificadors que s'indiquen a l'etiqueta **<input>**, mentre que les claus de la segona columna es corresponen amb les característiques de cada fitxer:

Clau	Descripció
name	Nom del fitxer enviat.
tmp_name	Nom del fitxer temporal on està guardat el fitxer acabat de pujar. Si no s'ha pogut realitzar aquesta operació (per exemple, el fitxer és massa gran), llavors el seu valor serà none.
size	Tamany en caràcters del fitxer.
type	El tipus MIME del fitxer (indicat pel client).
error	Codi d'error retornat: indica com ha anat la transferència.

Els codis d'error que podem trobar són:

Valor	Constant	Significat
0	UPLOAD_ERR_OK	Sense errors.
1	UPLOAD_ERR_INI_SIZE	El fitxer és més gran que el tamany indicat a la directiva upload_max_filesize del fitxer de configuració php.ini.
2	UPLOAD_ERR_FORM_SIZE	El fitxer és més gran que el tamany indicat en el camp ocult max_file_size.
3	UPLOAD_ERR_PARTIAL	El fitxer ha estat parcialment transferit.
4	UPLOAD_ERR_NO_FILE	No s'ha transferit el fitxer.

També existeixen un parell de funcions molt útils per a treballar amb aquests fitxers:

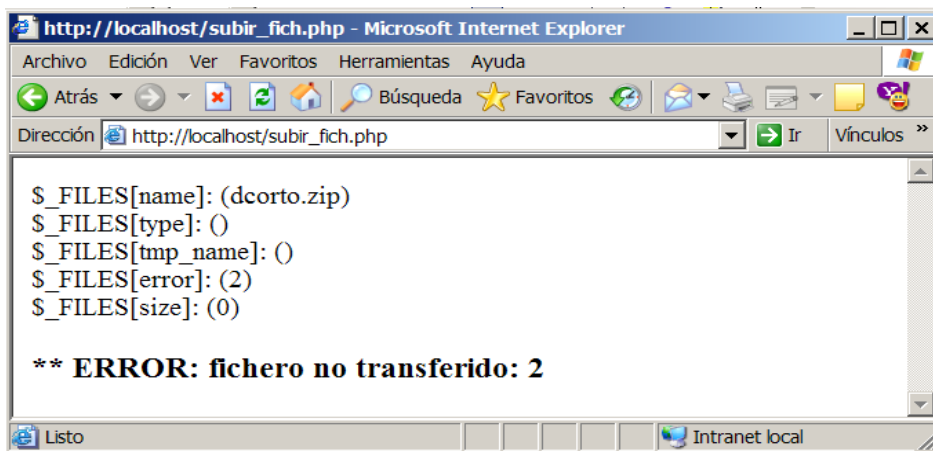
- `is_uploaded_file($_FILES['identificador_fitxer']['tmp_name'])`: indica si el fitxer ha estat pujat.
- `move_uploaded_file(nom_fitxer_original, destí)`: serveix per a renombrar o moure el fitxer temporal a un que ho serà permanent.

Pe. A continuació teniu l'exemple del fitxer que tractaria les dades enviades per la pàgina HTML anterior:

```
<?
foreach ($_FILES['f_adjunto'] as $clave => $valor)
    echo "\$_FILES[$clave]: ($valor)<br>";

if (!is_uploaded_file($_FILES['f_adjunto']['tmp_name']))
{
    $error=$_FILES['f_adjunto']['error'];
    die("<h3>** ERROR: fichero no transferido: $error </h3>");
}

if ($_FILES['f_adjunto']['type'] != 'application/x-zip-compressed')
    echo "<h3>** ERROR: el fichero enviado no está comprimido</h3>";
?>
```



10.7.2 DIRECTIVES DE php.ini INVOLUCRADES

S'han de considerar 3 directives del fitxer de configuració `php.ini` que determinaran el comportament de l'interpret PHP en el tema de la transferència de fitxers:

Directiva	Descripció
<code>file_uploads</code>	En funció del valor que tingui (<code>on</code> , <code>off</code>) permetrà o no pujar fitxers al servidor.
<code>upload_tmp_dir</code>	Directorio en el servidor on s'emmagatzemen de forma temporal els fitxers que han estat pujats.
<code>upload_max_size</code>	Limita el tamany màxim que podrà tenir un fitxer per a poder transferir-se al servidor. Aquest valor està per sobre del valor donat en el camp ocult identificat com a <code>max_file_size</code> .

