

## Dos tipos de transferencias

En la página anterior hemos hablado de la manera de transferir información entre el ordenador de un usuario y un servidor web.

Aquí trataremos algo –similar a primera vista– un poco distinto. Es el caso las transferencias –*en los dos sentidos*– **entre servidores** (un servidor **HTTP** y un servidor **FTP**).

En la configuración descrita en la instalación del servidor FTP hemos establecido que ambos servidores tengan por **root** el mismo directorio, pero no es la única opción posible.

Es totalmente factible que uno de los servidores esté alojado en un *hosting* de *Londres* y el otro en *Sydney*, por poner un ejemplo de lugares distantes.

Imaginemos que todo esto es cierto.

## Transferencias FTP

Los dos primeros pasos para poder utilizar las funciones FTP han de ser: **abrir la conexión** y **pasar el login**. El último sería **cerrar la conexión**.

### Abrir la conexión

`$x=ftp_connect (host,pt)`

Esta función –en la que **host** es una cadena con el nombre del servidor FTP (no te olvides de ponerlo entre comillas) y **pt** es el número del puerto a través del cual se efectúa la conexión FTP– **abre** una conexión con el servidor FTP.

Si se omite **pt** se asigna por defecto el valor **21** que es el *puerto* que se usa habitualmente para acceder a este tipo de servidores.

La variable **\$x** recogerá el **identificador de conexión** que será utilizado por las demás funciones.

### «Loguearse»

Utilizaremos este término del *argot informático* –¿horrible, verdad?– para referirnos al hecho de que el usuario se *acredite* como *autorizado* en el servidor FTP.

`ftp_login($x,user,pass)`

## Requisitos del sistema

El uso de estas funciones requiere que PHP tenga activada la opción FTP (*enabled*), que en nuestro caso está activada en la configuración por defecto, tal como puedes ver a través de tu **info.php**, que tendrá un apartado idéntico al que observamos en la siguiente imagen.

ftp

FTP support	enabled
-------------	---------

Esto en cuanto a PHP. Pero además de esta configuración será **imprescindible** disponer de un **servidor FTP accesible y activo**. En este tipo de transferencias intervienen dos servidores: el servidor HTTP (nuestro Apache) y el servidor FTP, cuyo procedimiento de instalación y configuración hemos descrito en el apartado **Servidor de FTP**. Antes de poder utilizar las funciones que aquí describimos deberás tener instalado y activo el servidor FTP que allí se describe.

Nos conviene tener muy presente que esta versión de PHP no admite **localhost** como nombre de servidor (en la versión 4 esto no ocurría) y que por esa razón hemos de referirnos al servidor FTP mediante su dirección IP (**127.0.0.1**, en nuestro caso de servidor local) y que hemos creado diferentes usuarios (con privilegios distintos) entre ellos **admin**.

## Comprobación de login y conexión

```
<?
# conexión con el servidor FTP
if($x=@ftp_connect ("127.0.0.1",21)){
    echo "Conexión FTP activada<br>";
}else{
    echo "No se activo lo conexión FTP<br>";
}
# registro de usuario
if(@ftp_login($x,"super","superi")){
    echo "El login y la password han sido aceptados";
}else{
    echo "Error en login o password";
}
#desconexión
ftp_quit($x);
?>
```

ejemplo91.php

## Lista de contenidos del subdirectorio *curlsophp*

```
<?
if($x=@ftp_connect ("127.0.0.1",21)){
    echo "Conexión FTP activada<br>";
}else{
    echo "No se activo lo conexión FTP";
}
if(@ftp_login($x,"webmaster","webmaster")){
    echo "El login y la password han sido aceptados<BR><BR>";
}else{
    echo "Error en login o password";
}
$lista=ftp_nlist($x,"/curlsophp");
foreach($lista as $c=>$v){
    print "Indice: ".$c." Valor: ".$v."<br>";
}
```

Una vez *abierta* la conexión es preciso comenzar la sesión utilizando la función **ftp\_login** con los siguientes parámetros:

- **\$x**, que es la variable en la que se recogía el resultado de ftp\_connect.
- **user**, que es el *nombre de usuario*.
- **pass**, que es la *password* del usuario.

Esta función devuelve un valor *booleano* que será **1** en el caso en que se inicie la sesión correctamente o **NUL** si no lo hace.

## Cerrar la conexión

Mediante la función:

### ftp\_quit(\$x)

se **cierra** la conexión abierta con el *identificador* indicado en la variable **\$x**.

## Gestión de directorios en el servidor FTP

Una vez *logueados* y con la conexión activa ya podremos utilizar *funciones FTP* tales como:

### ftp\_cdup(\$x)

Nos sitúa en el directorio raíz del servidor FTP.

### ftp\_pwd(\$x)

Nos devuelve una cadena con el nombre del directorio actual.

### ftp\_chdir(\$x, nuevodir)

Cambia el acceso actual al directorio especificado por la cadena **nuevodir**, en caso de que exista.

### ftp\_pwd(\$x)

Indica el nombre del directorio al que estamos accediendo en este momento.

### ftp\_mkdir(\$x, nomdir)

Crea un **subdirectorio** –en el directorio actual– cuyo nombre es el nombre indicado en la cadena **nomdir**.

### ftp\_rmdir(\$x, nomdir)

**Borra** el *directorio* especificado en la cadena **nomdir**.

Para que un directorio pueda **ser borrado** se requiere que esté **vacio** y que esté incluido *dentro* del directorio *actual*.

## Información sobre los

```

}
print "<H1>Lista completa</H1>";
$listacompleta=ftp_rawlist($x, "/");
foreach($listacompleta as $c=>$v){
    print "Indice: ".$c." Valor: <H1>".$v."</H1>";
}
ftp_quit($x);
?>

```

ejemplo92.php

El resultado de la ejecución del script anterior podría producir una salida similar a esta:

```

Conexión FTP activada
El login y la password han sido aceptados
Indice: 0 Valor: //configuracion.txt
Indice: 1 Valor: //AreaAlumnos

Lista completa

Indice: 0 Valor:


| 1          | 2a | 2b  | 2c  | 3     | 4   | 5  | 6     | 7                 | 8 | 9 |
|------------|----|-----|-----|-------|-----|----|-------|-------------------|---|---|
| -rw-r--r-- | 1  | ftp | ftp | 17064 | Mar | 04 | 21:11 | configuracion.txt |   |   |



Indice: 1 Valor:


| 1          | 2a | 2b  | 2c  | 3 | 4   | 5  | 6     | 7           | 8 | 9 |
|------------|----|-----|-----|---|-----|----|-------|-------------|---|---|
| drwxr-xr-x | 1  | ftp | ftp | 0 | Mar | 20 | 18:06 | AreaAlumnos |   |   |


```

Tal como puedes ver en la imagen, la cadena devuelta por la función *ftp\_rawlist* tiene dos resultados distintos. La primera de las cadenas comienza por **-** lo cual indica que se trata de un archivo y documento. En el segundo de los casos ese primer carácter es **d** e indica que se trata de un directorio.

Los *nueve caracteres* siguientes especifican los permisos de acceso a los ficheros y/o directorios. Se subdividen en tres bloques de igual tamaño que corresponden a los tres niveles de usuarios habituales en sistemas Unix/Linux (propietario, grupo y resto de usuarios). Para nuestros propósitos bastará con que consideremos los privilegios del primer bloque, es decir los del *propietario*.

El primero carácter de cada bloque sólo puede ser **r** ó **-**. Si se trata de un *fichero* y está marcado con **r** indica que se permite el acceso a él en modo *lectura* y si se trata de un directorio indica que está permitida la visualización de su contenido.

El segundo de los caracteres (puede ser **w** ó **-**) indica, si se trata de un fichero, que está permitida la modificación del fichero. Cuando se trata de un directorio significa que se pueden añadir o suprimir ficheros.

El tercero de los caracteres indicaría (**x** ó **-**) que el fichero -si se trata de un ejecutable- tiene permisos para ser ejecutado. Cuando se trata de un directorio, indica que pueden conocerse los atributos de los ficheros que contiene y que está permitido el acceso a él y a sus subdirectorios.

El signo **-** significa la negación del atributo en todas las opciones.  
El siguiente carácter, el número **1**, está asociado con sistemas Linux/Unix e indicaría el número de *vínculos duros* contra el archivo, que es otra cosa que una forma de asignar nombres distintos a un mismo fichero.

Los dos grupos siguientes -parece que no demasiado relevantes para nuestros propósitos- ser los nombres del usuario y grupo al que pertenece.

A continuación aparece el tamaño del archivo (cero si se trata de un directorio), la fecha y hora de su creación y el nombre del archivo o directorio.

## Una miscelánea de las funciones FTP

```

<?
# Conexión con el el servidor ftp utilizando su dirección IP
if(!$x=@ftp_connect ("127.0.0.1",21)){
    echo "No se activo lo conexión FTP";
    exit();
}
# Identificación de usuario webmaster (manejaremos ficheros en Apache)
if(!@ftp_login($x, "webmaster", "webmaster")){
    echo "Error en login o password";
    exit();
}
/* comprobamos el nombre del directorio actual del servidor FTP
que será el root correspondiente al usuario registrado (aparecerá /) */
echo "El directorio actual es: ", ftp_pwd($x), "<br>";

```

## contenidos de los directorios del servidor FTP

### ftp\_nlist(\$x, nomdir)

Devuelve una *array* *escalar* con los nombres de los *ficheros* y *subdirectorios* contenidos en el directorio que se indica en *nomdir*.

Si se trata del *directorio actual*, el parámetro *nomdir* puede especificarse como una cadena vacía ("").

Si la información se refiere a un *subdirectorio* del *actual* bastará poner su nombre como valor del parámetro *nomdir*.

En cualquier otro caso *nomdir* contendrá la *ruta* completa.

### ftp\_rawlist(\$x, nomdir)

Igual que la función anterior, *ftp\_rawlist* también devuelve un *array* *escalar*, pero en este caso con información ampliada.

Este *array* detalla, además del nombre del *fichero*, el *tamaño*, el *tipo*, la *fecha de la última modificación* y los *permisos* de lectura y/o escritura.

## Transferencia de ficheros

Las transferencias de ficheros pueden realizarse en *ambos sentidos*.

### Desde el servidor FTP hasta el servidor HTTP

Mediante la función:

### ftp\_get(\$x, nloc, nrem, modo)

se transfiere un *fichero* desde un *servidor FTP* hasta un directorio del *servidor HTTP* en el que se está ejecutando PHP.

La cadena *nloc* contiene el *nombre* con el que el *fichero* será *copiado* en el *directorio actual del servidor web* y la cadena *nrem* contiene el nombre (incluyendo el *path*) del *fichero* que debe ser transferido.

El parámetro *modo* puede contener uno de estos valores: **FTP\_ASCII** o **FTP\_BINARY**

### Desde el servidor HTTP hasta el servidor FTP

Para realizar transferencias en *sentido contrario* al anterior se utiliza la siguiente sintaxis:

### ftp\_put(\$x, nrem, nloc, modo)

Se comporta de forma idéntica a la

```
/* intentamos cambiar a un subdirectorio indicando la ruta absoluta
partiendo del directorio root del usuario actual.
En caso de error (ruta incorrecta o falta de permisos de accesos
nos daría un mensaje de error. Si el cambio tiene éxito nos indicaría
el nombre del nuevo directorio */
if(!@ftp_chdir($x, "/cursoph/pdf")){
    print "No tienes permisos de acceso a este directorio<br>";
    print "o la ruta es incorrecta.¡Comprueba los datos!<br>";
}else{
    echo "Hemos cambiado al directorio: ", ftp_pwd($x), "<br>";
}
# comprobamos el nombre del sistema operativo del servidor de FTP
echo "El S.O: del servidor FTP es: ", ftp_systype ($x), "<br>";
/* obtenemos una matriz conteniendo la lista de ficheros y directorios
del subdirectorio "cursoph/fuentes" del del directorio actual*/
$lista=ftp_nlist($x, "/cursoph/fuentes");
# escribimos la lista de ficheros contenidos en ese directorio
echo "Lista de ficheros del subdirectorio cursoph/fuentes<br>";
foreach ($lista as $valor){
    echo $valor, "<br>";
}
# obtenemos una lista completa de los contenidos de ese subdirectorio
$lista=ftp_rawlist($x, "/cursoph/fuentes");
# ordenamos el array que contiene la lista anterior
sort($lista);
echo "Contenidos del subdirectorio cursoph/fuentes<br>";
/* extrae los elementos del array eliminando los espacios repetidos
mediante la funcion preg_replace en la que \s+ indica uno o más espacios
que serán sustituidos por uno solo ( ' ' ) */
foreach($lista as $v){
    $v=preg_replace('/\s+/', ' ', $v);
    # imprimimos la cadena completa
    print "<br><br><br>". $v. "<br>";
    # convertimos la cadena en un array
    # utilizando los espacios como separadores
    $extrae=explode(" ", $v);
    # leemos los elementos del array y comentamos sus valores
    foreach($extrae as $indice=>$cont){
        switch($indice){
            case 0:
                print "El elemento de indice ".$indice." es: ".$cont."<br>";
                if (substr($cont,0,1)=="d"){
                    print "Es un directorio<br>";
                }elseif(substr($cont,0,1)=="-"){
                    print "Es un fichero<br>";
                }
                if (substr($cont,1,1)=="r"){
                    print "Tiene permisos de LECTURA<br>";
                }elseif(substr($cont,1,1)=="-"){
                    print "No tiene permisos de LECTURA<br>";
                }
                if (substr($cont,2,1)=="w"){
                    print "Tiene permisos de ESCRITURA<br>";
                }elseif(substr($cont,2,1)=="-"){
                    print "No tiene permisos de ESCRITURA<br>";
                }
                break;
            case 4:
                print "El tamaño de este fichero es: ".$cont." bytes<br>";
                break;
            case 8:
                print "El nombre del fichero o directorio es: ".$cont."<br>";
                break;
        }
    }
}
# regresamos al directorio cursoph
ftp_chdir($x, "/cursoph/");
/* creamos un subdirectorio (del directorio actual que es cursoph)
con nombre experimento anteponiendo @# para evitar mensajes de error
en caso de que ya existiera */
@ftp_mkdir($x, "experimento");
/* copiamos el fichero enol.jpg desde el directorio que se indica
en el tercer parámetro (cursoph)
al directorio del servidor FTP que se indica
en el segundo parámetro. Le ponemos por nombre lago_enol.jpg */
ftp_put($x, "../cursoph/experimento/lago_enol.jpg",
        "../cursoph/enol.jpg", FTP_BINARY);
# obtenemos el tamaño del fichero transferido
```

función anterior. La cadena **nrem** sigue siendo el *nombre* y el *path* del servidor FTP (donde vamos a copiar el fichero) y **nloc** contiene el nombre del fichero en el servidor web (origen de la transferencia).

### Modificación de ficheros en el servidor FTP

**ftp\_rename(\$x,nant,nnuevo)**

Cambia el nombre del fichero **nant** por el indicado en la cadena **nnuevo**.

**ftp\_delete(\$x,fichero)**

Elimina -en el servidor FTP- el fichero indicado en la cadena **fichero**.

### Información sobre ficheros del en el servidor FTP

**ftp\_size(\$x,nomfile)**

Devuelve el tamaño (en bytes) del fichero que se indica en la cadena **nomfile**.

**ftp\_mdtm(\$x,nomfile)**

Esta función devuelve la *fecha de la última modificación* del fichero indicado en la cadena **nomfile**. Esta fecha se indica en *tiempo Unix*.

#### Ejercicio nº 31

Crea un formulario y un script mediante los cuales puedas *subir* al directorio Documentacion del servidor FTP un fichero cualquiera.

Utiliza el nombre de usuario y contraseña adecuados para poder acceder al directorio indicado y evita que quede copia del fichero en el servidor Apache.

Intenta mejorar el script para limitar el tamaño del archivo y restringir la transferencia a formatos \*.jpg ó \*.gif.

#### ¡Cuidado!

Observa los path de los ejemplos. Al anteponer **../** estaremos indicando una ruta absoluta desde al root de servidor FTP y con **./** aludimos a un

```
echo "El tamaño de fichero transferidos es: ",
ftp_size($x,"../cursophp/experimento/lago_enol.jpg")," bytes<br>";
/* escribimos la fecha de la última modificación del fichero transferido
que coincidirá con la fecha y hora en la que se realizó la transferencia.
Convertimos a formato de fecha convencional el tiempo UNIX que devuelve
la función ftp_mdtm */
print "La fecha de modificación del fichero es:";
print date("d-m-Y H:i:s",ftp_mdtm($x,"./experimento/lago_enol.jpg"));
# cambiamos el nombre del fichero lago_enol.jpg por lago_covadonga.jpg
# en el servidor FTP
@ftp_rename($x,"./experimento/lago_enol.jpg",
            "./experimento/lago_covadonga.jpg");
/* creamos un enlace de descarga directa del fichero haciendo una llamada
mediante el protocolo ftp:// utilizando la sintaxis:
ftp://usuario:contraseña@nombre del servidor
seguidos de la ruta (en el servidor FTP) y el nombre del fichero */
print "<BR><a href='ftp://webmaster:webmaster@localhost'";
print "/cursophp/experimento/lago_covadonga.jpg'> Descargar</a>";
/* transferimos al directorio cursophp con nombre liborio.jpg
un fichero procedente del directorio experimento
cuyo nombre es lago_covadonga.jpg*/
ftp_get($x,"../cursophp/liborio.jpg",
        "../experimento/lago_covadonga.jpg",FTP_ASCII);
/* comprimimos un fichero alojado en cursophp
para transferirlo comprimido al servidor FTP */
#empezamos leyendo el fichero y guardándolo en una cadena
$f1=fopen("cabina.jpg","r");
while (!feof($f1)) {
    $cadena .= fgets($f1,1024);
}
fclose($f1);
# comprimimos la cadena obtenida del fichero anterior
$scl=gzencode($cadena,3,FORCE_GZIP);
# guardamos la cadena comprimida en un fichero
$f=fopen("cabina.jpg.gz","w");
fwrite($f,$scl);
fclose($f);
/* al servidor el fichero comprimido. No es necesario indicar
la ruta actual ya que ha sido creado en el mismo directorio
en el que se está ejecutando el script */
ftp_put($x, "../experimento/cabina.jpg.gz",
        "cabina.jpg.gz",FTP_BINARY);
#eliminamos el fichero comprimido del directorio cursophp
unlink("cabina.jpg.gz");
# cerramos la conexión con el servidor ftp
ftp_quit($x);
# establecemos un enlace de descarga para el fichero comprimido
print "<BR><a href='ftp://webmaster:webmaster@localhost'";
print "/cursophp/experimento/cabina.jpg.gz'>Descarga comprimido</a>";
?>
```

ejemplo93.php

#### ¡Cuidado!

Al ejecutar el script del ejemplo anterior **por segunda vez** (sobre **Linux Ubuntu**) puede aparecerte un mensaje de error del tipo **overwrite permission denied**.

Este problema puede ser causado por un defecto de configuración del sevidor FTP. Hemos podido comprobar que, algunas veces, por una extraña razón, aparecen en el fichero de configuración dos líneas segudas diciendo **AllowOverwrite Off** y **AllowOverwrite On**. La configuración correcta es **AllowOverwrite On** (para permitir sobrescribir). Bastaría con eliminar la línea marcada con Off (o reemplazar el Off por On) para solventar el problema.

Anterior



Índice



Siguiente



subdirectorío del actual.

Si te aparece el error que comentamos al margen puedes editar el fichero de configuración (con GADMIN PROFTPD) y sustituir su contenido por el que puedes encontrar en un fichero llamado **config\_proftpd.txt** en el directorio **soft\_linux** del CD.