Protección de directorios

Una vez que tengamos modificada la configuración del servidor Apache estaremos en condiciones de poder proteger directorios –restringir o limitar el acceso a ellos– y también podremos redirigir *peticiones* a páginas predeterminadas en casos concretos.

Este control ha de realizarse mediante ficheros que tienen por nombre **.htaccess** (el primer caracter es el punto) y que no pueden llevar ningún tipo de extensión.

Loa ficheros **.htaccess** pueden ser incluidos en cualquier directorio o subdirectorio del espacio del servidor.

El fichero .htaccess

Las misiones más importantes que puede realizar este fichero son: *redireccionar* y *restringir accesos*.

Veamos cada una de ellas por separado.

Errores y redireccionamiento

Los mensajes de error más frecuentes al intentar acceder a páginas web –y sus causas– son los siguientes:

Error 401
El subdirectorio está protegido por número IP o por password y el intento de acceder a él no ha tenido éxito.
Error 403
El acceso al documento solicitado está prohibido.
Error 404
El documento solicitado no ha sido hallado.
Error 500
Error del servidor. Usualmente este error se da cuando se ha intentado ejecutar de forma incorrecta un CGI, o bien debido a problemas en el servidor.

Los errores de los tipos 403 y 404 suelen producirse en la mayoría de las ocasiones por direcciones incorrectas y –aparte de causar un pésimo efecto– suelen provocar el abandono de la visita.

Un fichero **.htaccess** con este contenido:

Configuración del servidor Apache

Para poder proteger de directorios –mediante un fichero llamado **.htaccess**– es necesario realizar algunas modificaciones en la configuración de Apache.

Abriremos nuestro fichero **httpd.conf** y buscaremos las lineas en las que aparece: AllowOverride **None** y reemplazaremos **None** por **All** guardaremos los cambios en httpd.conf y reiniciaremos nuestro servidor Apache.

En caso de que estemos usando como sistema operativo **Linux Ubuntu** no es necesario realizar ninguna modificación. Bastará matener la configuración establecida durante el proceso de instalación.

Comprobación de la configuración

Empezaremos escribiendo en la barra de direcciones de nuestro navegador (también puedes hacerlo desde este enlace) esta dirección:
http://localhost/cursoPHP/noexisto.html y nos aparecerá un mensaje de error diciendo que no existe ninguna página con ese nombre. Algo lógico, porque realmente no existe.

Ejercicio nº 35

Abre tu editor –no utilices el block de notas porque te dará muchísimos problemas en este caso– y escribe la siguiente línea:
ErrorDocument 404 http://localhost/cursoPHP/index.php
y guarda el documento en el directorio cursoPHP con el nombre (aunque te parezca extraño, no lleva nada delante del punto) **.htaccess**
Pulsa de nuevo en el enlace que tienes aquí arriba –o escribe la dirección en el navegador– y observarás que ahora no dice *página no encontrada* sino que se abre la página principal del Curso.
Edita de nuevo el fichero **.htaccess** y añádele las siguientes líneas:
ErrorDocument 401 http://localhost/cursoPHP/index.php
ErrorDocument 403 http://localhost/cursoPHP/index.php
guardándolo después de haber hecho los cambios.

Crear el fichero de claves y contraseñas

Las restricciones de usuarios mediante **.htaccess** requieren un fichero de claves y contraseñas.

Para crearlo basta con abrir el *block de notas* y escribir **la clave** seguida de **dos puntos** (**:**) y a continuación escribir **la contraseña**.
Podemos poner tantas como deseemos sin más limitaciones que escribir cada bloque *clave:contraseña* en una línea distinta. Este puede ser un ejemplo:

```
pepe:Pepito
pepa:Pepita
```

Podemos guardar este fichero en el *sitio que deseemos* sin que sea necesario que pertenezca al root del servidor.

El directorio **seguridad** que hemos creado cuando tratábamos de INCLUDE (ver página) puede ser un buen sitio. Podemos ponerle cualquier nombre sin que importe que tenga extensión o no la tenga.

Crear un fichero de contraseñas encriptadas

ErrorDocument 401 *pagX*
ErrorDocument 403 *pagY*
ErrorDocument 404 *pagZ*

donde *pagX*, *pagY* y *pagZ* sean direcciones (completas) de las páginas a las que deseamos *redireccionar* el navegador, conseguiría que esos errores *llevaran* al visitante a la página que nosotros deseáramos.

Herencias

El archivo **.htaccess** provoca **herencia**. Eso significa que las especificaciones incluidas en un directorio –sean restricciones o redirecciones– son efectivas en todos los subdirectorios que contiene, incluso en el caso de que esos subdirectorios tengan su propio **.htaccess** y que en él se establezcan condiciones distintas a las anteriores.

Al crear el fichero del ejercicio nº 22 y guardarlo en *cursoPHP* las condiciones establecidas afectarán a todos sus subdirectorios (puedes probar a abrir una página con un nombre cualquiera en el subdirectorio images). Por eso, si pretendemos que desde subdirectorios distintos se redireccione a páginas distintas tendremos que incluir un .htaccess en cada uno de ellos y omitirlo en el directorio que los contenga.

Protección de directorios

Son muchas las posibilidades que ofrece .htaccess a la hora de restringir el acceso a un directorio determinado.

Entre otras opciones, se puede denegar el acceso a todos los usuarios; denegar el acceso con *excepciones*, autorizar a todos (equivale a no restringir), autorizar con *excepciones* o requerir clave y contraseña.

Lo que hemos denominado *excepciones* también permite una serie de alternativas tales como: una IP determinada, un rango de IP's, nombres de dominio, etcétera.

Sólo comentaremos la forma de protección de directorios mediante claves de usuario y contraseña.

Restricción de acceso a usuarios no autorizados

Este tipo de protección requiere crear *un fichero de claves y contraseñas* y *configurar* de forma adecuada **.htaccess**

Lo relativo a la creación de los primeros lo tenemos detallado aquí a la derecha en sus dos opciones:

Apache posee una utilidad que permite la creación de ficheros de claves con *contraseñas encriptadas*. Se trata de un programa llamado **htpasswd.exe** que está en el subdirectorio **bin** del servidor.

Para **crear** un *nuevo fichero* el procedimiento sería el siguiente:

En la línea de comandos: **Inicio->Ejecutar**

debemos escribir:

path **htpasswd -c** *nombre y path del fichero de claves usuario*

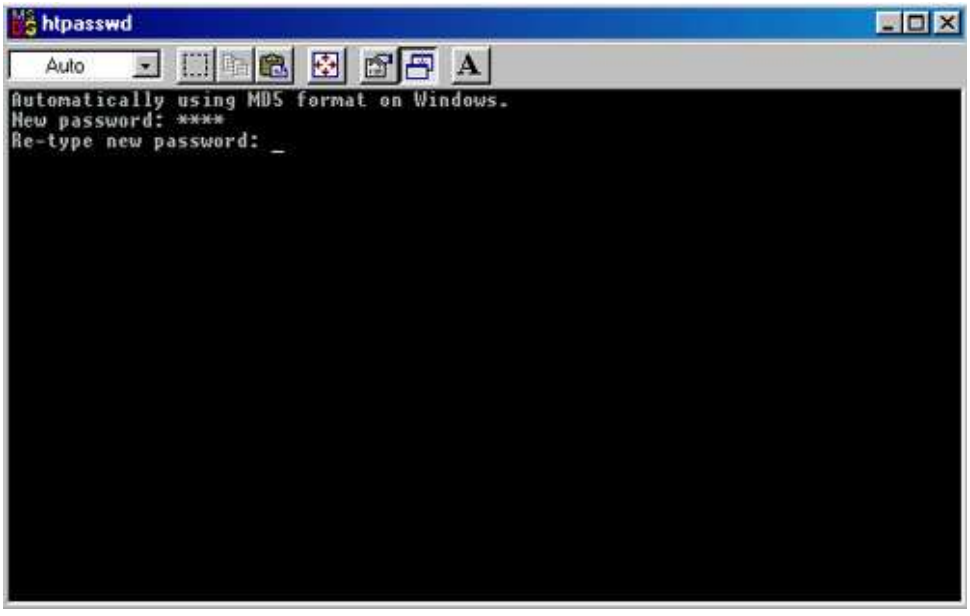
si se trata de un sistema operativo **Linux Ubuntu** habría que escribir en la consola

htpasswd -c *nombre y path del fichero de claves usuario*

En nuestra configuración y para **crear un fichero** con la palabra clave **pepe** escribiríamos:

C:/ServidoresLocales/Apache/bin/htpasswd -c C:/ServidoresLocales/Apache/seguridad/misclaves.txt pepe

y aparecería una ventana de MS-DOS en la que deberemos escribir la **password** para ese usuario.



Añadir usuarios a un fichero de contraseñas encriptadas

Porcederíamos de la misma forma. Volveríamos a ejecutar **htpasswd** con la nueva clave pero sin utilizar **-c**.

¡Cuidado!

El modificador **-c destruiría el fichero anterior, si existiera** y crearía uno nuevo.

El proceso ahora sería:

encriptadas o sin encriptar.

Para el caso *específico* de nuestro servidor Apache, el fichero **.htaccess** ha de contener:

AuthType Basic

No permite modificación e indica el tipo de autentificación requerida.

AuthName "*nuestro texto* **"**

El texto que escribamos aquí aparecerá como mensaje en la ventana en la que nos pedirá la clave

AuthUserFile "*path***"**

Entre esas comillas debes escribir el *nombre del fichero de contraseñas especificando su path completo.*

require valid-user

Este texto indica que para acceder se requiere un usuario válido.

Con nuestra configuración de Apache no es necesario especificar en **.htaccess** la forma de encriptación de contraseñas. El propio servidor interpreta el contenido del fichero y aplica u omite los criterios de encriptación.

A riesgo de parecerte pesados tenemos que volver a insistir que no todos los **hosting** tienen habilitada esta opción, pero además hemos de hacer mención a otro detalle muy importante.

La configuración que hemos comentado **no es válida** para todos los servidores.

Según como esté configurado el servidor, la versión del software que utilice, etcétera no sería extraño que se necesitara esta otra sintaxis:

AuthType Basic
AuthName "*Texto*"
AuthTextUserFile *fichero*
required valid-user
AuthTextCrypt *On/Off*

u otras similares que pueden inducirnos al error. Lo mejor, en caso de servidores ajenos, es *consultar* al administrador del sistema sobre estos aspectos y recabarle detalles sobre la sintaxis específica de su configuración.

C:/ServidoresLocales/Apache/bin/htpasswd C:/ServidoresLocales/Apache/seguridad/misclaves.txt luis

Habríamos creado así nuestro fichero con claves encriptadas. Si pretendiéramos visualizarlo nos aparecería lo siguiente:

```
pepe:$apr1$EC4.....$7Z3.p2tv2QpzrZbo4bI2j0
luis:$apr1$SU4.....$iU8a.YTo.ZvYyRggDAvTC.
```

¡Cuidado!

Bajo **Linux**, antes de añadir el usuario *luis* tendríamos que asegurarnos de que el fichero **misclaves.txt** tenga permisos para poder efectuar modificaciones en su cnotenido.

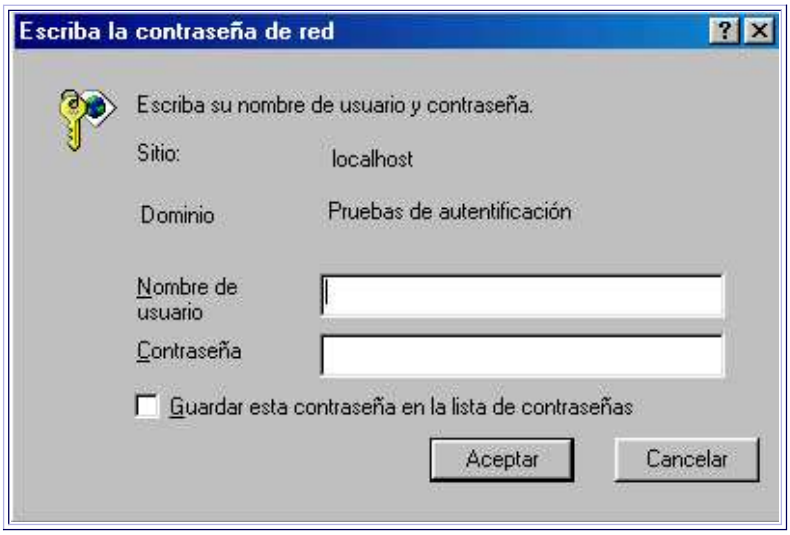
Un ejemplo de .htaccess

Supongamos que tenemos un directorio llamado **protegido** en cualquier parte del servidor (por ejemplo dentro de htdocs) la forma de protegerlo sería crear un fichero con nombre .htaccess con un contenido como este:

```
AuthType Basic
AuthName "Pruebas de autentificación"
AuthUserFile "C:/ServidoresLocales/Apache/seguridad/misclaves.txt"
require valid-user
```

y guardarlo en ese directorio.

Al acceder al directorio **protegido** aparecerá una ventana como esta:



y si al cabo de *tres intentos* no escribimos *la clave y contraseña adecuadas* se producirá un **Error 401**.