



**UMSA**

**UM.MOVI**

**UMSA MOVILIDAD VIRTUAL**



# Desarrollo de aplicaciones Web

Apache – .htaccess



- El archivo .htaccess es un archivo de texto que el servidor Apache utiliza para implementar reglas sobre directorios, archivos y además se utiliza para obligarlo a comportarse de cierta forma frente a algunas URLs en el momento que son accedidas.
- Por ejemplo los servidores suelen usar el .htaccess para reescribir URLs largas y complejas, en otras más simples y fácilmente recordables, permiten bloquear a usuarios por su dirección IP y/o dominio, bloquear bots y arañas web. También permite controlar las páginas de errores cuando estos ocurren del lado del servidor.



- Windows a diferencia de Unix/Linux no permite crear archivos sin nombre, es decir que no tengan ningún valor antes del “.”
- Para poder crear un archivo sin nombre y solo extensión usaremos el siguiente método:
  - Creamos en la raíz de nuestro sitio un archivo cualquiera por ejemplo htaccess.txt
  - Entramos por CMD y nos posicionamos en la carpeta de nuestro sitio web (sino la hemos modificado es la carpeta htdocs dentro de C:/Apache2.4/htdocs)
  - Por ultimo ejecutamos el comando “rename htaccess.txt .htaccess” para colocar el nombre correcto

- Para muchos de los parametros y ajustes utilizados en el archivo .htaccess requerimos activar el modulo Rewrite. Solo es necesario descomentarlo en el httpd.conf de nuestro Apache
- `LoadModule rewrite_module modules/mod_rewrite.so`

- Debemos confirmar que todos los permisos a los directorios que utilizaran el .htaccess permitan su acceso, en el caso del directorio raíz requiera revisar la raíz "/" y la ruta completa en dado caso de que tengamos dos especificaciones por separado:
 

<pre>&lt;Directory /&gt;     Options FollowSymLinks     AllowOverride All     Require all granted &lt;/Directory&gt;</pre>	<pre>&lt;Directory "C:/Apache24/htdocs"&gt;     Options Indexes FollowSymLinks     AllowOverride All     Require all granted &lt;/Directory&gt;</pre>
<pre>&lt;Directory "C:/Sitios/localhost"&gt;     Options Indexes FollowSymLinks     AllowOverride All     Require all granted &lt;/Directory&gt;</pre>	

- Para evitar que los navegadores puedan acceder al archivo .htaccess hay una instrucción dentro del httpd.conf que hace que los archivos que comiencen con “.ht” sea denegado su acceso:

# The following lines prevent .htaccess and .htpasswd files from being

# viewed by Web clients.

```
<Files ".ht*">
```

```
    Require all denied
```

```
</Files>
```

# RewriteRule – Sustituye contenido

- Si queremos que el usuario entre a una liga pero sin que cambie la URL del navegador le mostremos el contenido de otra
- Ejemplo, si tengo mipagina1.html y mipagina2.html y quiero que cuando el usuario entre a “mipagina1.html” le mostremos el contenido de “mipagina2.html” pero que la URL del navegador siga apuntando a “mipagina1.html”
- Colocamos en el archivo .htaccess:

RewriteEngine On

RewriteRule ^mipagina1.html mipagina2.html



- Para cambiar la página de inicio de nuestro dominio, debemos utilizar la siguiente sintaxis ...
- DirectoryIndex inicio.html

- Para que sino ponemos WWW y solo misitio.com nos redireccione a la pagina con WWW.misitio.com
- RewriteCond %{HTTP\_HOST} ^miprueba.com:83 [NC]
- RewriteRule ^(.\*)\$ http://www.miprueba.com:83/\$1 [L,R=301]

- La respuesta 301 significa que fuimos redireccionados y podemos utilizarla cuando una condicion se cumpla ejemplo:
- Cuando alguien entre al dominio.com/carpeta:

Redirect 301 /carpeta/ http://www.undominio.com/

- Cuando alguien intente entrar a un archivo:

Redirect 301 /carpeta/carpeta/index.html  
http://www.undominio.com/

# Options – Mostrar u ocultar listado de Archivos

- # Para MOSTRAR listado de archivos

Options +Indexes

- # Para EVITAR el listado de archivos y marcar Error 303

Options -Indexes

- Se usa para mostrar mas detalle de los archivos (fecha, tamaño, observaciones)
- `#IndexOptions +FancyIndexing`
- Se usa para mostrar la información default de los archivos
- `#IndexOptions -FancyIndexing`



- Para ignorar ciertos archivos o extensiones en la web en el listado de archivos, aunque no se visualizan aquí, si pueden ser accesibles por medio de su ruta completa.

IndexIgnore \*.gif

# `<files></files>` Para limitar accesos

Para Apache 2.2

- Por medio del .htaccess podemos restringir acceso a ciertos archivos que no queremos sean visualizados por el navegador, esto mediante la instrucción `<files></files>`:

```
<files archivo.php>
```

```
order allow,deny
```

```
deny from all
```

```
</files>
```

# `<files></files>` Para limitar accesos

Para Apache 2.4

- Por medio del .htaccess podemos restringir acceso a ciertos archivos que no queremos sean visualizados por el navegador, esto mediante la instrucción `<files></files>`:

```
<files archivo.php>
```

```
Require all denied
```

```
</files>
```

- 400 Bad Request Petición errónea
- 401 Authorization Required Autorización requerida. Se muestra cuando un usuario intenta acceder a una zona protegida sin autorización.
- 403 Forbidden Acceso denegado. Se muestra cuando un usuario intenta acceder a una zona a la que no tiene permisos.
- 404 Not Found Uno de los errores más comunes. Se muestra cuando una página no existe en el servidor. Resulta muy útil para mostrar información al usuario de lo que ocurre exactamente.
- 500 Internal Server Error Se muestra cuando se ha producido un error interno en el servidor. Normalmente por un error en la ejecución de scripts CGI

# ErrorDocument - Errores personalizados

- Por medio de la instrucción ErrorDocument podemos personalizar por medio de una pagina los errores comunes de HTTP:
- ErrorDocument 400 /error/badrequest.html
- ErrorDocument 401 /error/authreqd.html
- ErrorDocument 403 /error/forbid.html
- ErrorDocument 404 /error/notfound.html
- ErrorDocument 500 “Error numero 500 consulta a sistemas”



- Es común que otros sitios nos “roben” imágenes, colocándolas en su sitio y consumiendo nuestro ancho de banda sin ningún beneficio para nuestro sitio, esto se puede evitar de la siguiente forma:

RewriteEngine on

RewriteCond %{HTTP\_REFERER} !^\$

RewriteCond %{HTTP\_REFERER} !^http://(www\.)?mi-dominio.com/.\*\$ [NC]

RewriteRule \.(jpg|jpeg|gif|png|bmp)\$ - [F]

# RewriteRule \.(jpg|jpeg|gif|png|bmp)\$ http://www.servidor-  
imagenes.com/no-hotlink.gif [R,L]

# Deny From – Para bloquear direcciones en Apache 2.2

- <Limit GET HEAD POST>
- order allow,deny
- deny from 000.000.000.000
- deny from 111.000.000.000
- deny from 222.000.000.000
- allow from all
- </limit>

## Deny From – Para bloquear direcciones en Apache 2.4

- <Limit GET HEAD POST>
- Require not ip 50.62.136.183
- </limit>

- Existen varias formas de agregar seguridad a los sitios, por ejemplo:
  - Básica
  - Básica con encriptación
  - Digestiva con encriptación
- En todos los casos, requeriremos un archivo adicional para almacenar los usuarios y contraseñas en formato usuario:contraseña, sin importar si son encriptadas o no
- Por estandar, los archivos de contraseñas normalmente se colocan en un archivo con nombre .htXXX para que no sean accedidos desde fuera

- Es un archivo utilizado para crear usuarios/contraseñas para permitir accesos al sitio web. Sirve para el modo Básico encriptado o no encriptado
- En Windows se crea de la misma forma que el archivo .htaccess, es decir renombrando por medio de MS-DOS
- Dentro del archivo colocamos los usuarios deseados con su contraseña seguida de ":" después del login:

usuario1:pass1

usuario2:pass2

usuario3:pass3



\*.htpasswd: Bloc de notas

Archivo

Edición

Formato

Ver

Ayuda

usuario1:1234

usuario2:5678

- El archivo se coloca en la carpeta del sitio al cual le queremos agregar seguridad




# htpasswd (encriptada) Apache 2.4

- Para crear contraseñas encriptadas con MD5 la opción es usar la herramienta proveída por apache para la creación del archivo:
- C:\Apache24\bin\htpasswd.exe
- La invocamos desde MS:DOS de la siguiente forma:

htpasswd -c nombredelarchivo nombredelusuario

htpasswd -c .htpasswd usuario1

 .htpasswd: Bloc de notas

Archivo Edición Formato Ver Ayuda

|usuario1:\$apr1\$Y2MWN/OD\$G00zbMxVdIsI4q0hPv

- Al presionar enter nos pedirá 2 veces la contraseña deseada para el usuario
- El “-c” es para crear el archivo, cuando se agreguen usuarios se repite la misma sintaxis eliminando el “-c” lo cual agregara usuarios al archivo.
- El archivo se guarda en la misma carpeta BIN, hay que trasladarlo a la carpeta del sitio

- Para utilizar contraseñas encriptadas por MD5 requerimos activar el módulo digestor en el httpd.conf:

`LoadModule auth_digest_module modules/mod_auth_digest.so`

Después de modificar el archivo httpd.conf es necesario reiniciar el servicio de apache para que los cambios surtan efecto

Al modificar los archivos .htaccess o .htpasswd no requiere reiniciar los servicios, solo refrescar el sitio

- Una vez teniendo el archivo con usuario y contraseñas (Básico o Básico con MD5) en el formato especificado, agregamos las instrucciones para leerlo en el archivo .htaccess:

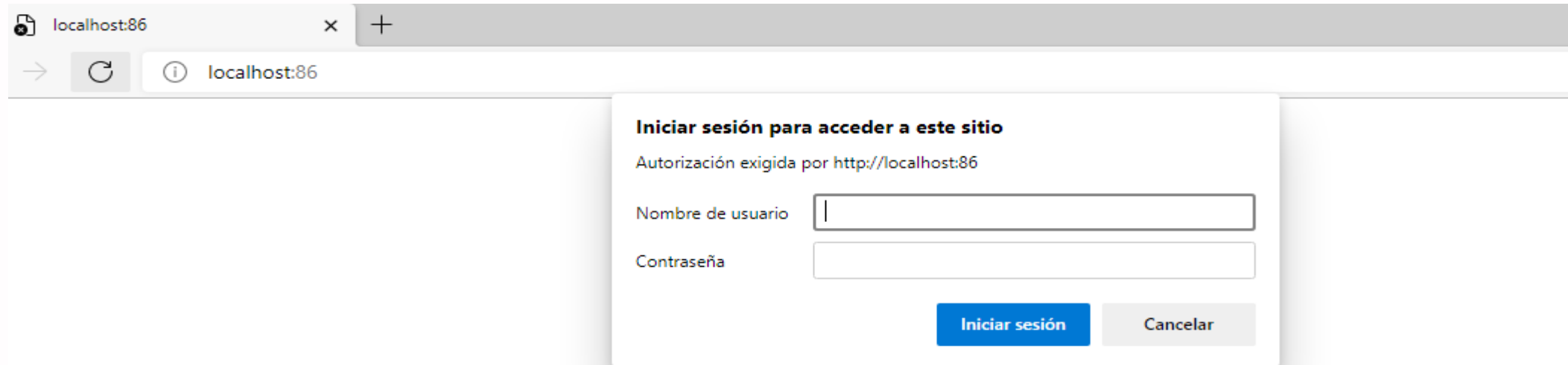
AuthType Basic

AuthUserFile "C:/Sitios/Localhost/.htpasswd"

AuthName "Necesita validarse"

require valid-user

- Al querer entrar nos dirá algo como esto:



The image shows a web browser window with a single tab titled 'localhost:86'. The address bar also displays 'localhost:86'. A modal dialog box is centered on the screen, titled 'Iniciar sesión para acceder a este sitio'. Below the title, it says 'Autorización exigida por http://localhost:86'. There are two input fields: 'Nombre de usuario' and 'Contraseña'. At the bottom right of the dialog are two buttons: 'Iniciar sesión' (highlighted in blue) and 'Cancelar'.

localhost:86

→ ↻ ⓘ localhost:86

**Iniciar sesión para acceder a este sitio**  
Autorización exigida por http://localhost:86

Nombre de usuario

Contraseña

Iniciar sesión Cancelar

- Tambien podemos especificar un usuario deseado de la lista de usuarios del httpasswd:

AuthType Basic

AuthUserFile "C:/Sitios/Localhost/.htpasswd"

AuthName "Necesita validarse"

require user usuario2



- Dentro de la misma carpeta BIN se encuentra el archivo htdigest.exe que nos permite crear contraseñas MD5 pero con un ámbito o real, la sintaxis es similar a la de htpasswd:

htdigest -c nombredelarchivo realm nombredelusuario

htdigest -c .htdigest seguridad usuario1

- Al presionar enter nos pedirá 2 veces la contraseña deseada para el usuario
- El “-c” es para crear el archivo, cuando se agreguen usuarios se repite la misma sintaxis eliminando el “-c” lo cual agregara usuarios al archivo.
- El archivo se guarda en la misma carpeta BIN, hay que trasladarlo a la carpeta del sitio

- Para utilizar el archivo generado por el htdigest en el .htaccess colocamos:

AuthType Digest

AuthName seguridad #el realm

AuthDigestNonceLifetime 1

AuthDigestDomain http://localhost:86 #opcional

AuthDigestProvider file

AuthUserFile "C:/Sitios/LocalHost/.htdigest"

require valid-user

- Con la autenticación básica, la contraseña se envía casi sin formato (codificada en base64) al servidor y en el lado del servidor se obtiene un hash y se compara con la contraseña hash (almacenada en un archivo htpasswd o similar). Con la autenticación implícita, la contraseña hash se envía al servidor (con algunos datos definidos por el servidor agregados para que los ataques de reproducción no funcionen). Pero para verificar la contraseña, debe tener la contraseña simple en el lado del servidor (o algo parecido a la contraseña simple). Esto significa que si el atacante obtiene acceso al archivo htpasswd, necesita descifrar todas las contraseñas antes de que puedan usarse para la autenticación básica, mientras que si obtiene acceso al archivo htdigest, puede usarlo directamente para la autenticación implícita.