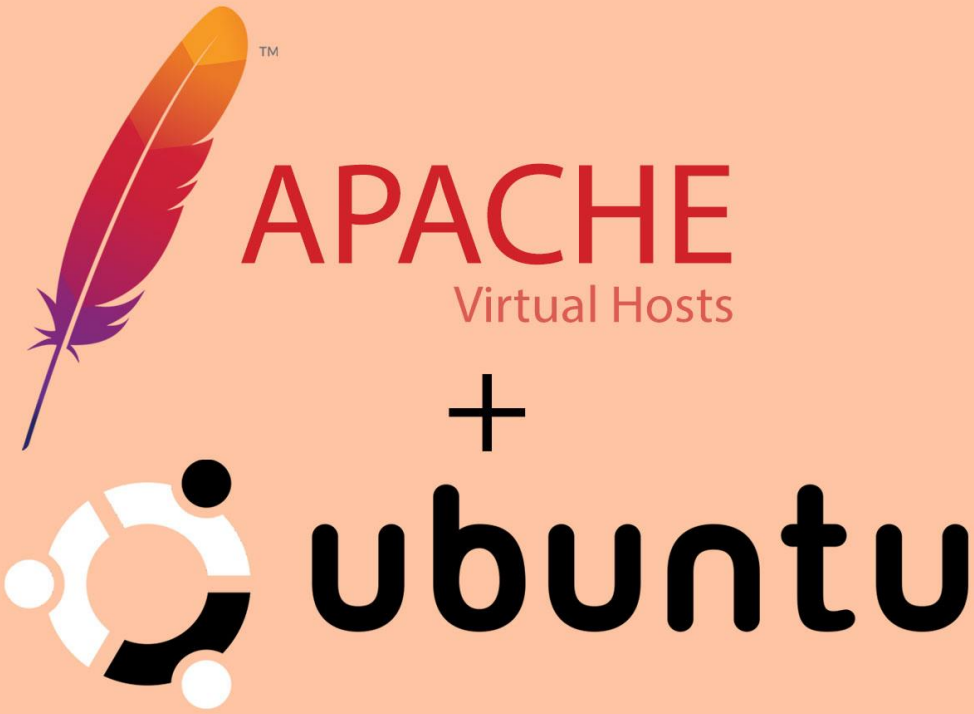
AUTENTICACIÓN Y AUTORIZACIÓN



DAVID SEGURA TRISTANCHO
2º ASIR 2022/23

ÍNDICE

1. Introducción

Dentro del tema que nos concierne de HTTP y el servicio Apache, vamos a dedicar estas páginas al tratamiento de la autenticación para que los sitios estén protegidos de todo aquel que no tenga la autorización para ingresar en ellos. Vamos a verlo desde el punto de vista del método básico (Basic), así como del Digest y el uso útil del archivo .htaccess.

2. Configurar un sitio mediante autenticación para uno o varios usuarios

Esta primera configuración vamos a verla desde el modelo básico. Para ello, y una vez que el servicio apache2 está instalado y activo, creamos un archivo dentro de /*var*/www llamado usuarios donde introduzco a todo aquel usuario que queremos que acceda al sitio en cuestión:

```
usuario@usuario:/var/www/nuevomodulo$ sudo htpasswd -c g_permitidos jignacio
New password:
Re-type new password:
Adding password for user jignacio
usuario@usuario:/var/www/nuevomodulo$ sudo htpasswd g_permitidos antonio
New password:
Re-type new password:
Adding password for user antonio
usuario@usuario:/var/www/nuevomodulo$ sudo htpasswd g_permitidos carlos
New password:
Re-type new password:
Adding password for user carlos
usuario@usuario:/var/www/nuevomodulo$
```

Si visualizamos el archivo, veremos los usuarios creados:

```
usuario@usuario:/var/www$ cat u_permitidos
alan:$apr1$1LUJ6TlM$so7xIaY5dgB/Qvc37.Ivg/
alejandra:$apr1$sPplji0w$vjzxnId0Mc5KRmkTgtVDO.
dani:$apr1$F8Dr7SU2$SE5TLDleuXkPxfKxu.jMO/
diego:$apr1$o.Ng2OU8$bOxUB4AytaY6v.aomI3uD0
jignacio:$apr1$2yXYkffP$1pZ5jQFfPIHPoe1hCcZky/
joaquinma:$apr1$wjQxBQxF$SjCQr6aNmD5sGmE7n.SF7/
antonio:$apr1$CmHcV9c2$yhZ15QT2hc5qGiC/YQsKN/
```

Y en g_permitidos, podremos visualizar todos aquellos usuarios encuadrados dentro del grupo distribuidos entre alumnado y profesorado.

```
usuario@usuario:/var/www$ cat g_permitidos
alumnado: alan alejandra diego dani
profesores: jignacio antonio joaquinma
```

Vamos al archivo de configuración del sitio y establecemos la directiva directorio, especificando la autentificación del archivo para usuarios y el correspondiente a grupos (AuthGroupFile):

```
 GNU nano 4.8                                                nuevomodulo.conf
<VirtualHost 192.168.128.171:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/nuevomodulo

        <Directory "/var/www/nuevomodulo">
                AuthType Basic
                AuthName "Restricted files"
                AuthBasicProvider file
                AuthUserFile "/var/www/u_permitidos"
                AuthGroupFile "/var/www/g_permitidos"
                Require group profesores
        </Directory>
```

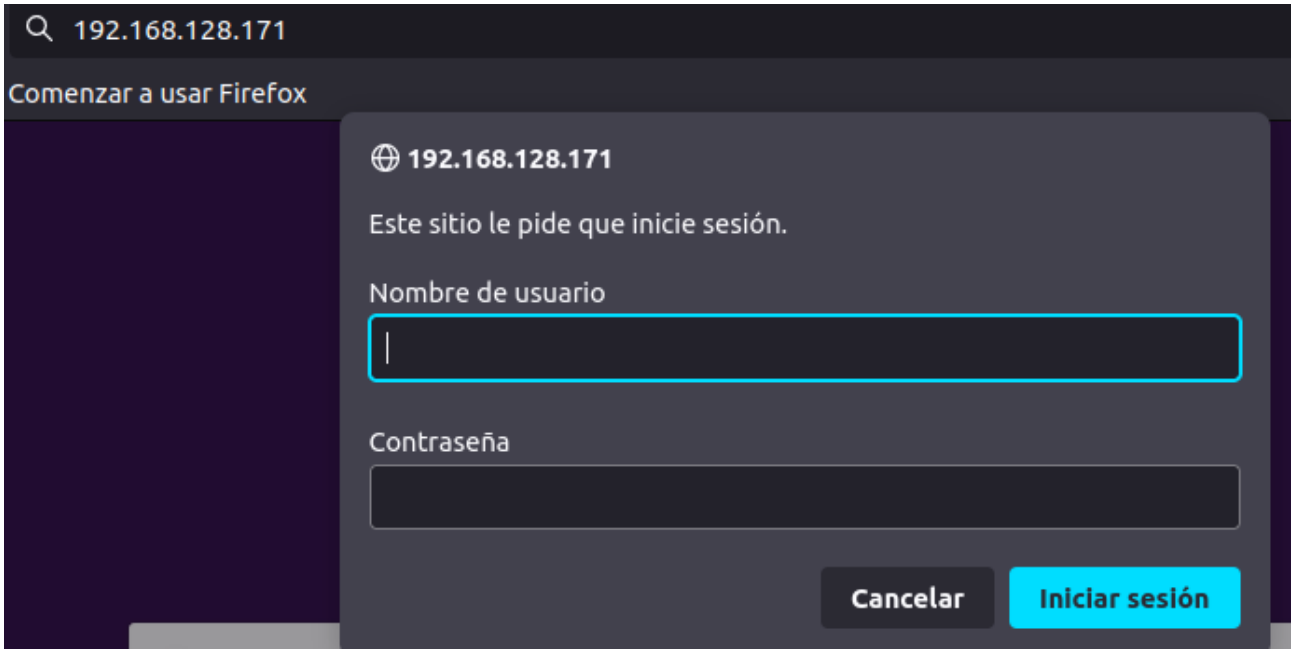
Importante, tendremos que activar el módulo de Group File. En primer lugar, trataremos de localizarlo dentro de los módulos disponibles.

```
usuario@usuario:/etc/apache2$ cd mods-available/
usuario@usuario:/etc/apache2/mods-available$ ls
access_compat.load    authz_user.load     dir.load                log_debug.load       proxy_express.load   session_dbd.load
actions.conf          autoindex.conf      dump_io.load            log_forensic.load    proxy_fcgi.load      session.load
actions.load          autoindex.load      echo.load               lua.load             proxy_fdpass.load    setenvif.conf
alias.conf            brotli.load         env.load                macro.load           proxy_ftp.conf       setenvif.load
alias.load            buffer.load         expires.load            md.load              proxy_ftp.load       slotmem_plain.load
allowmethods.load     cache_disk.conf     ext_filter.load         mime.conf            proxy_hcheck.load    slotmem_shm.load
asis.load             cache_disk.load     file_cache.load         mime.load            proxy_html.conf      socache_dbm.load
auth_basic.load       cache.load          filter.load             mime_magic.conf      proxy_html.load      socache_memcache.load
auth_digest.load      cache_socache.load  headers.load            mime_magic.load      proxy_http2.load     socache_redis.load
auth_form.load        cern_meta.load      heartbeat.load          mpm_event.conf       proxy_http.load      socache_shmcb.load
authn_anon.load       cgid.conf           heartmonitor.load       mpm_event.load       proxy.load           speling.load
authn_core.load       cgid.load           http2.conf              mpm_prefork.conf     proxy_scgi.load      ssl.conf
authn_dbd.load        cgi.load            http2.load              mpm_prefork.load     proxy_uwsgi.load     ssl.load
authn_dbm.load        charset_lite.load   ident.load              mpm_worker.conf      proxy_wstunnel.load  status.conf
authn_file.load       data.load           imagemap.load           mpm_worker.load      ratelimit.load       status.load
authn_socache.load    dav_fs.conf         include.load            negotiation.conf     reflector.load       substitute.load
authnz_fcgi.load      dav_fs.load         info.conf               negotiation.load     remoteip.load        suexec.load
authnz_ldap.load      dav.load            info.load               php7.4.conf          reqtimeout.conf      unique_id.load
authz_core.load       dav_lock.load       lbmethod_bybusyness.load php7.4.load         reqtimeout.load      userdir.conf
authz_dbd.load        dbd.load            lbmethod_byrequests.load proxy_ajp.load      request.load         userdir.load
authz_dbm.load        deflate.conf        lbmethod_bytraffic.load proxy_balancer.conf  rewrite.load         usertrack.load
authz_groupfile.load  deflate.load        lbmethod_heartbeat.load proxy_balancer.load  sed.load             vhost_alias.load
authz_host.load       dialup.load         ldap.conf               proxy.conf           session_cookie.load  xml2enc.load
authz_owner.load      dir.conf            ldap.load               proxy_connect.load   session_crypto.load
```

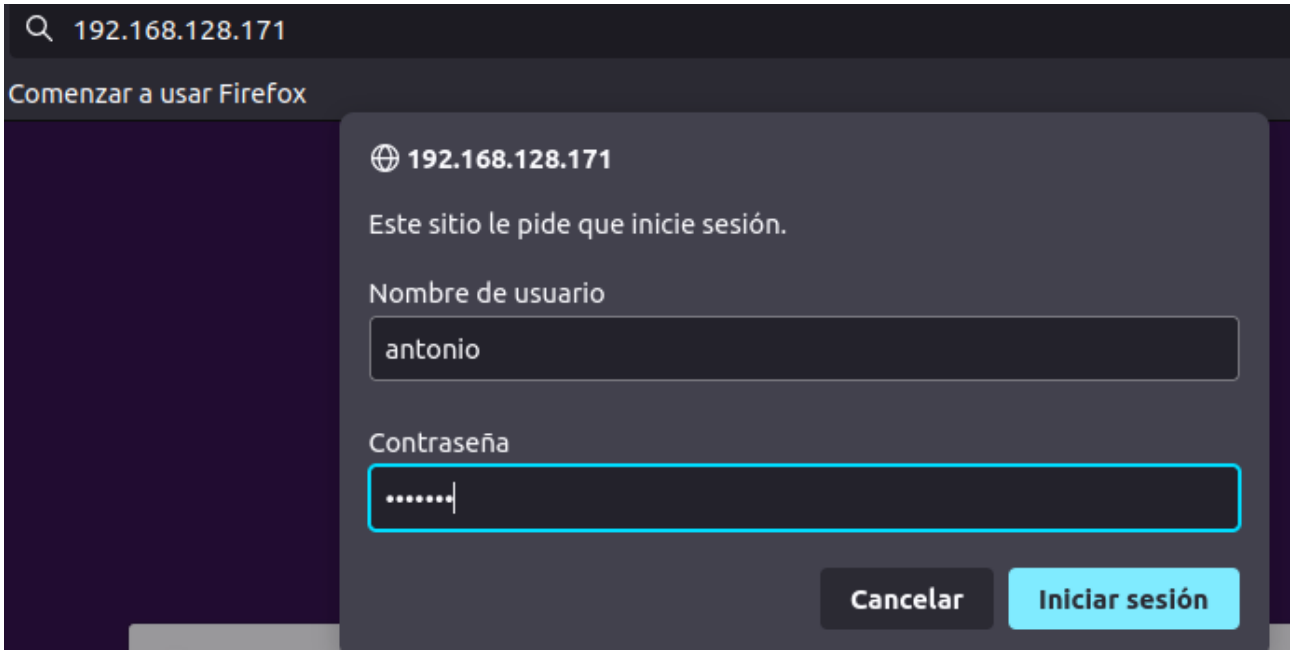
Por ende, lo activamos e iniciamos a posteriori el servicio apache.

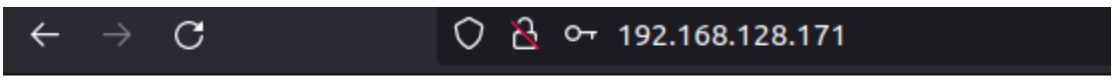
```
usuario@usuario:/etc/apache2/mods-available$ sudo a2enmod authz_groupfile
Considering dependency authz_core for authz_groupfile:
Module authz_core already enabled
Enabling module authz_groupfile.
To activate the new configuration, you need to run:
  systemctl restart apache2
usuario@usuario:/etc/apache2/mods-available$ sudo service apache2 restart
```

Y si reiniciamos el servicio y accedo al navegador con el directorio especifico, me pedirá el nombre de usuario y su contraseña. Si lo hago, por ejemplo, con el alumno Alan, no podré acceder:



En cambio, si lo hago con el profesor Antonio, el contenido del documento me será mostrado:





DocumentaciÃ³n confidencial solo para los usuarios autorizados

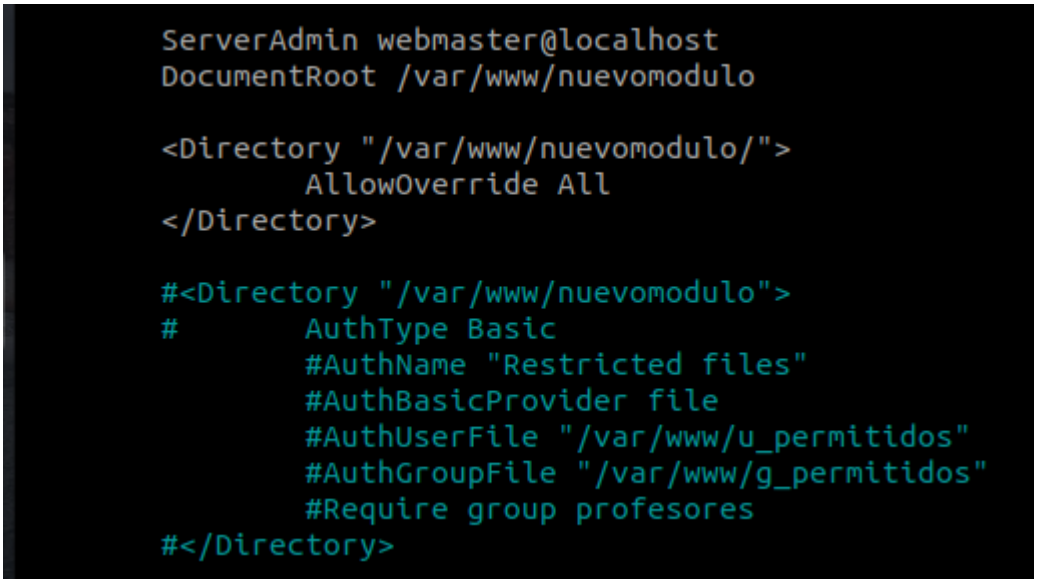
3. Archivos .htaccess

El archivo .htaccess de una web será como la tabla de los mandamientos para el **servidor Apache**, que cumplirá estrictamente lo que se le indique en el archivo .htaccess. No tiene por qué existir un solo archivo .htaccess, sino que puede existir uno para cada directorio si lo consideras necesario.

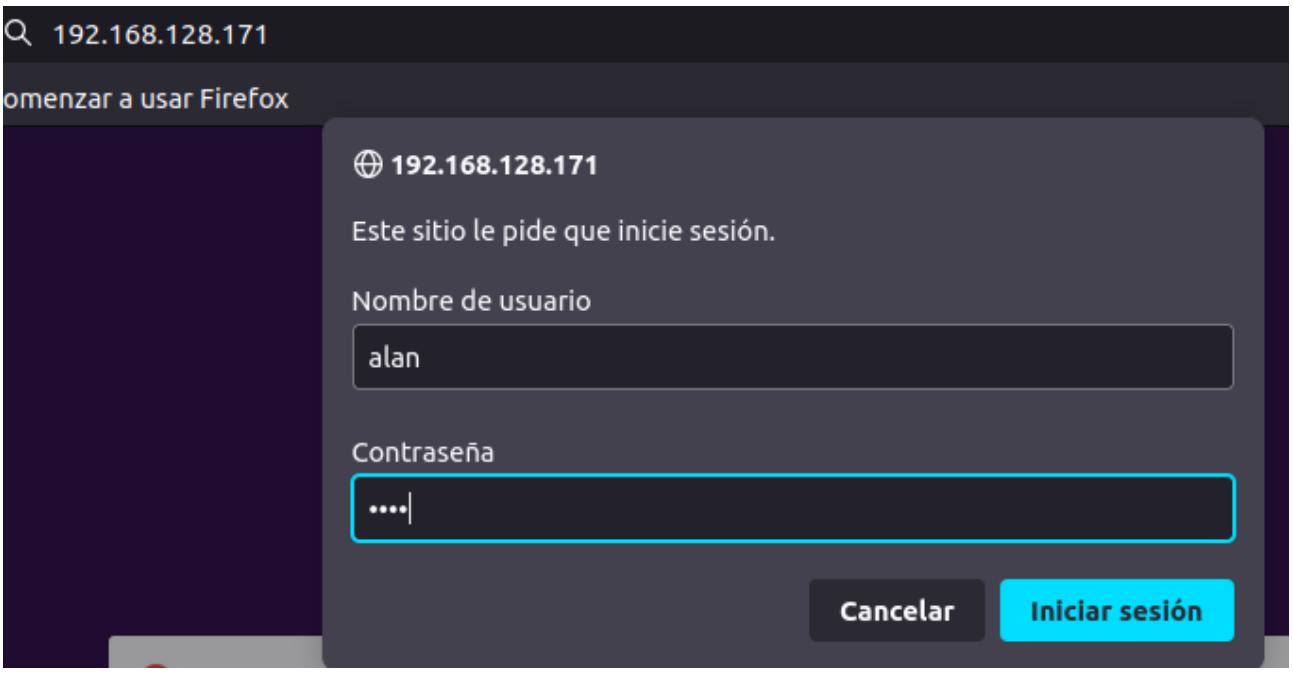
El comportamiento de este archivo es jerárquico en el árbol de directorios de tu web, por lo que, si tienes un archivo .htaccess en el directorio raíz de la web, las instrucciones que indiques en ese archivo se aplicarán a toda ella. En cambio, si creas un archivo en alguna de las carpetas, las directrices de ese archivo solo se aplicarán al contenido de esa carpeta y subcarpetas, por lo que no afectará al resto de directorios de la web.

Una vez definida la función de este trascendental y peculiar archivo, yo lo voy a crear dentro del directorio /var/www/nuevomodulo, el utilizado para las practicas anteriores en clase. Aquí me traigo el contenido del directory creado para este método básico:

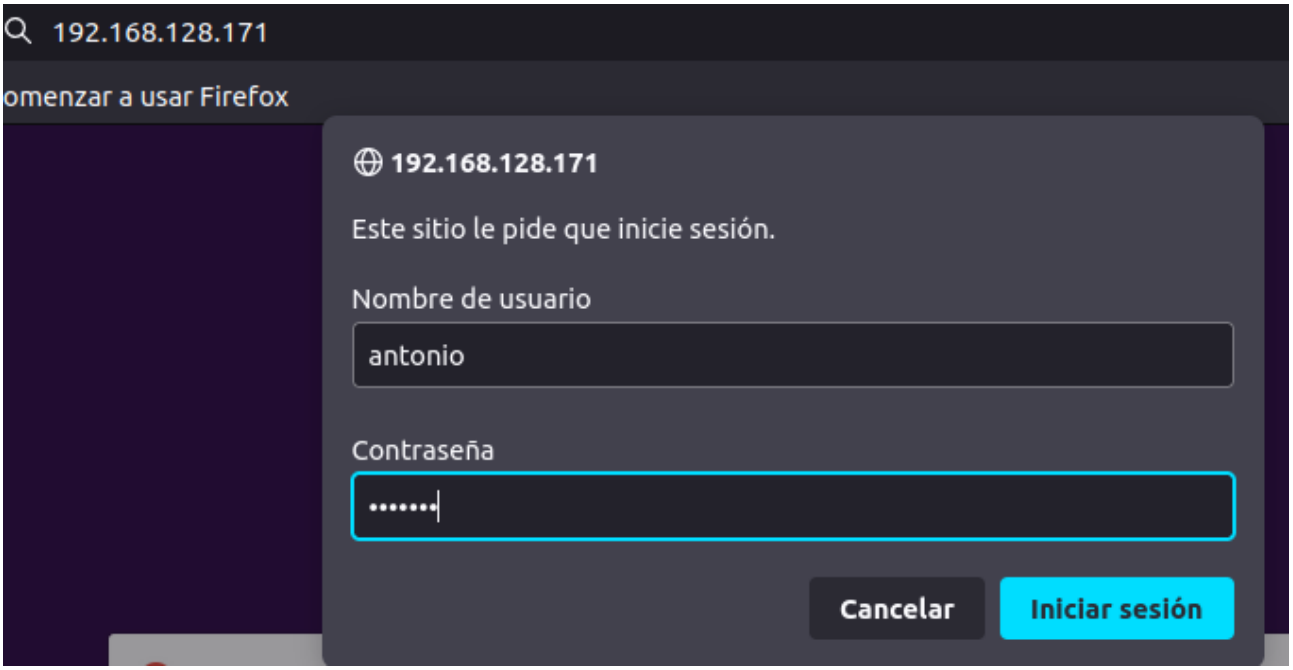
Si accedemos al archivo de configuración del sitio, lo que haremos será descomentar lo hecho en el anterior paso para la autentificación de grupo. En su lugar, crearemos un nuevo directory en el que voy a indicar la ruta del sitio y el permiso para sobrescribir todo a aquel que esté autorizado para ello:

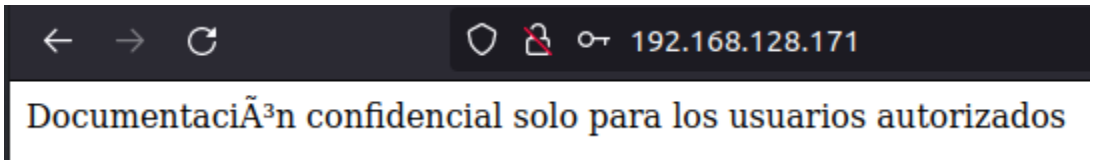


Reinicio el servicio y si vamos al navegador como antes veremos cómo, del mismo modo, el alumno Alan no podrá acceder al mensaje.



Mientras tanto, el profesor Antonio sí llegará a leer el contenido alojado en el sitio.





4. Configurar un sitio mediante autenticación para varios usuarios bajo el método Digest

Creamos un archivo dentro de /*var*/www llamado u_digest2 donde introduzco a todo aquel usuario que queremos que acceda al sitio en cuestión.

```
usuario@usuario:/var/www$ sudo htdigest -c u_digest2 g_permitidos jignacio
Adding password for jignacio in realm g_permitidos.
New password:
Re-type new password:
usuario@usuario:/var/www$ sudo htdigest u_digest2 g_permitidos antonio
Adding user antonio in realm g_permitidos
New password:
Re-type new password:
usuario@usuario:/var/www$ sudo htdigest u_digest2 g_permitidos carlos
Adding user carlos in realm g_permitidos
New password:
Re-type new password:
```

Si visualizamos el archivo, podremos comprobar como los usuarios recién creados se insertan en base a usuarios (u_permitidos) o grupo (g_permitidos).

```
usuario@usuario:/var/www$ cat u_digest2
jignacio:g_permitidos:f0b7c3dee8e4ee4ecb345a776a24d2f5
go:u_permitidos:f670c4ac68c88c5b2d061a5c9172fa71
alejandra:u_permitidos:6acf94b3e064a65c5ceb966334821613
alan:u_permitidos:a64222a7625e0993dba7acc01f71db40
antonio:g_permitidos:f7209cf429aeadce17acfc11836328e6
carlos:g_permitidos:b8a2ad5593dbb2bc5a7ec3aae36e9dbb
```

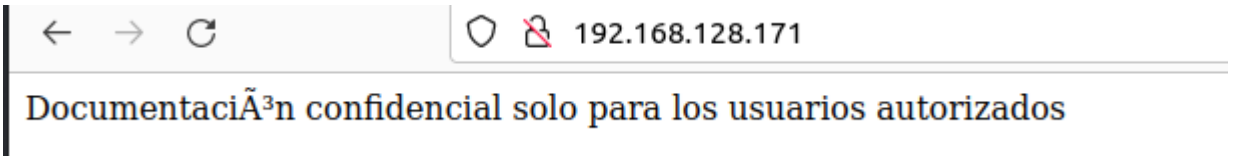
Creados nuestros usuarios, accedemos al archivo de configuración del sitio y añadimos el directory específico para Digest, donde indico como AuthType Digest, en AuthName g_permitidos, es decir, los usuarios que van a tener acceso, y como Require solo los usuarios válidos.

```
 GNU nano 4.8                                    nuevomodulo.conf
<VirtualHost 192.168.128.171:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/nuevomodulo

        <Directory "/var/www/nuevomodulo">
                AuthType Digest
                AuthName "g_permitidos"
                AuthBasicProvider file
                AuthUserFile "/var/www/u_digest2"
                Require valid-user
        </Directory>
```

Un detalle trascendental que no puede quedar atrás. Hay que activar el módulo encargado de gestionar este método digest (auth_digest). Una vez activado será necesario reiniciar el servicio apache.

```
usuario@usuario:/etc/apache2/mods-available$ sudo a2enmod auth_digest
Considering dependency authn_core for auth_digest:
Module authn_core already enabled
Enabling module auth_digest.
To activate the new configuration, you need to run:
  systemctl restart apache2
usuario@usuario:/etc/apache2/mods-available$ sudo service apache2 restart
```

Dados todos estos pasos, si cualquiera de los usuarios introducidos en el archivo digest, dentro de g_permitidos, accede con la ip al navegador podrá leer el contenido del documento para cuyo acceso está autorizado.



He aquí un inciso. Si hubiera optado por realizar este proceso Digest mediante el uso del archivo .htaccess, únicamente habría que descomentar en el archivo de configuración del sitio el apartado directory e insertar todo lo descomentado dentro de .htaccess, tal y como se puede apreciar en la siguiente imagen:

```
  GNU nano 4.8                                                    .htaccess
AuthType Digest
AuthName "g_permitidos"
AuthBasicProvider file
AuthUserFile "/var/www/u_digest2"
Require valid-user
```

¿En qué se diferencia el método Digest del basic? Podemos afirmar que el método Digest es más seguro sobre todo a nivel de posible captura de tráfico, ya que el intercambio de hashes entre nuestro navegador y el servidor es más complejo.

Por dar un ejemplo, podemos hacer aún más seguro el acceso a nuestras páginas sensibles haciendo uso conjunto de HTTPS y el método de autenticación Digest, es decir, que en primer lugar nuestro servidor web utilizará SSL o TLS en las conexiones y sobre la conexión encriptada, usar el método de autenticación Digest.