IES Enric Valor
C/ Duanes, 17 - Pego
96 640 99 60
03007613@edu.gva.es

GENERALITAT VALENCIANA
Conselleria d'Educació, Cultura, Universitats i Ocupació
Finançat per la Unió Europea
Fons Europeus

ACTIVDAD 0302:

Configuración del control de acceso en Apache 2.4

Andrea Berenice Mendez
Despliegue de Aplicaciones Web
2 DAW

ÍNDICE

I. Objetivos

- Entender como funcionan las directivas de Apache para el control de acceso.
- Proteger un directorio dependiente de la dirección IP
- Proteger un directorio con usuario y contraseña
- Configurar el control de acceso con un fichero `.htaccess`.

II. Enunciado

- En la aplicación desplegada anteriormente tendrás que proteger dos directorios:
  1. `restringido`, que solo será accesible desde nuestra dirección IP.
  2. `privado`, que solo será accesible con el usuario `tu_nombre_en_minuscula` y contraseña `1234`.
  3. La configuración estará en un fichero `.htaccess` en el `DocumentRoot`

III. Recursos

Toda la información la podéis extraer de estos recursos:

- Control de acceso. Autorización.
- Autenticación básica.
- Configuración de Apache mediante archivo `.htaccess`.

IV. Producto final

- Fichero zip con la aplicación web preparada para ser desplegada en su contenedor. Es decir, una vez desplegado el contenedor se podrá acceder a la aplicación web con los cambios realizados.
- Documento técnico en PDF donde se documente todo el proceso, siguiendo las indicaciones de la *Guía adaptada de presentación de trabajos con ordenador.*

V. DESARROLLO

1. Creación de directorios y archivos

a) Para comenzar se creará en el directorio ***/web*** dos directorios con los siguientes comandos:

mkdir privado

mkdir restringido

** En el directorio */privado* se restringirá el accesso de los visitantes por medio de contraseña **

** En el directorio */restringido* se limitará el acceso por medio de la IP **

b) Dentro de cada uno de ellos se creará un archivo ***index.html*** con un mensaje que se mostrará una vez se acceda a ellos desde el navegador.

c) Luego desde el directorio principal se crearán dos nuevos directorios que serán volúmenes que permitirán persistir la configuración dentro de los contenedores cada vez que se ejecuten en otra máquina:

mkdir config

mkdir claves

** En el directorio /config se persistirá el archivo con la configuración de apache **

** En el directorio /claves se persistirá el archivo de usuarios y contraseñas **

2. Modificación del archivo de configuración

a) Con el contenedor php-apache_joomla arrancado, se copia el archivo de configuración de apache al directorio ***/config*** creado anteriormente en local:

docker cp php-apache_joomla:/etc/apache2/sites-available/000-default.conf conf/

También puede tomarse este archivo desde la máquina local

b) Se establecen las políticas de acceso utilizando la directiva **<Directory>** dentro del archivo archivo de configuración local. Este archivo será posteriormente mapeado al contenedor **php-apache_joomla** para que las reglas de acceso se apliquen al servidor Apache en ejecución.

000-default.conf

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        ServerName localhost
        DocumentRoot /var/www/html

        <Directory "/var/www/html/privado">
                AuthUserFile
                "/etc/apache2/claves/psswd.txt"
                AuthName "ÁREA PRIVADA"
                AuthType Basic
                Require valid-user
        </Directory>

        <Directory "/var/www/html/restringido">
                Require ip 192.168.18
        </Directory>
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log
        combined
</VirtualHost>
```

#Directory: directorio a proteger.

#AuthUserFile: fichero con la información de los usuarios autorizados y contraseñas (encriptadas) en un directorio no visitable desde Apache.

#AuthName: se establece un nombre al entorno privado a proteger.

#AuthTypeBasic: se establece el tipo de control de acceso que se utilizará para acceder al directorio.

#Require valid-user: se establece el nivel de seguridad.

La directiva ***Require*** utilizada en **<Directory>** o en ficheros **.htacess** permite o deniega el acceso a recurso o conjunto de recurso utilizando alguna de las siguientes opciones:

- `Require all granted`: El acceso es permitido incondicionalmente.
- `Require all denied`: El acceso es denegado incondicionalmente.
- `Require user userid [userid] ...`: El acceso es permitido sólo si los usuarios indicados se han autentificado.
- `Require group group-name [group-name] ...`: El acceso es permitido sólo a los grupos de usuarios especificados.
- `Require valid-user`: El acceso es permitido a los usuarios válidos.
- `Require ip 10 172.20 192.168.2`: El acceso es permitido si se hace desde el conjunto de direcciones especificadas.
- `Require host dominio`: El acceso es permitido si se hace desde el dominio especificado.
- `Require local`: El acceso es permitido desde `localhost`.
- Se puede usar el operador `not` para indicar la denegación

3. Creación del fichero de usuario y contraseña

a) Para crear el fichero de contraseñas para autenticar usuarios se utiliza el siguiente comando desde el directorio ***/claves*** en local:

htpasswd -c passwd.txt andrea *(ver Figura 1)*

**htpasswd gestiona archivos de contraseñas encriptadas.

-c creará un archivo llamado passwd.txt y añadirá el usuario andrea. Luego pedirá una contraseña 1234 que será encriptada.

Esto sólo debe realizarse con el primer usuario, luego se debe remover la -c ya que la creación de otro archivo con el mismo nombre sobreescribirá el primero. **

```
andrea@andrea-VirtualBox:~/0302 - Andrea Mendez/claves$ htpasswd -c psswd.txt andrea

New password:
Re-type new password:
Adding password for user andrea
andrea@andrea-VirtualBox:~/0302 - Andrea Mendez/claves$ cat psswd.txt
andrea:$apr1$6ILNLKcY$0vBJ1eGw0Ewquc3C/3dxW1
andrea@andrea-VirtualBox:~/0302 - Andrea Mendez/claves$
```

Figura 1: Creación de usuario y contraseña con htpasswd

4. Modificación del fichero docker-compose

a) Para que todas las configuraciones establecidas anteriormente tengan efecto en los contenedores, es necesario realizar las siguientes modificaciones en el fichero docker-compose.yml:

docker-compose.yml

```yaml
version: '3.2'
services:
        web:
                image: php:8.3-apache
                build:
                        context: .
                        dockerfile: Dockerfile
                container_name: php-apache_joomla
                ports:
                        - "8080:80"
                volumes:
                        - ./www:/var/www/html
                        - .conf:/000-default.conf:/etc/apache2/sites-available/000-default.conf
                        - ./claves:/etc/apache2/claves
                        - ./php.ini:/usr/local/etc/php.ini
                depends_on:
                        - db
        db:
                image: mysql:8.1
                container_name: mysql_joomla
                environment:
                        - MYSQL_ROOT_PASSWORD=root
                        - MYSQL_DATABASE=joomla_db
                        - MYSQL_USER=andrea
                        - MYSQL_PASSWORD=andrea
                volumes:
                        - ./mysql:/docker-entrypoint-initdb.d/
                        - db_data:/var/lib/mysql
        phpmyadmin:
                image: phpmyadmin:5.2.1
                container_name: phpmyadmin_joomla
                ports:
                        - "8088:80"
                environment:
                        - PMA_HOST=db
                        - PMA_PORT=3306
                        - PMA_USER=root
                        - PMA_PASSWORD=root
                depends_on:
volumes:
        db_data:
        www:
```

#Se monta el volumen de configuración de apache
#Se monta el volumen de claves, que se creará en el contenedor.

b) Una vez modificado el archivo, se ejecuta el siguiente comando para que los cambios tengan efecto:

docker-compose up -d –build

5. Verificación de los cambios realizados

a) Para verificar el control de acceso por usuario y contraseña, se accede desde el navegador a las siguientes URL *(ver Figura 2, 3 y 4):*

http://*localhost:8080/privado/index.html*

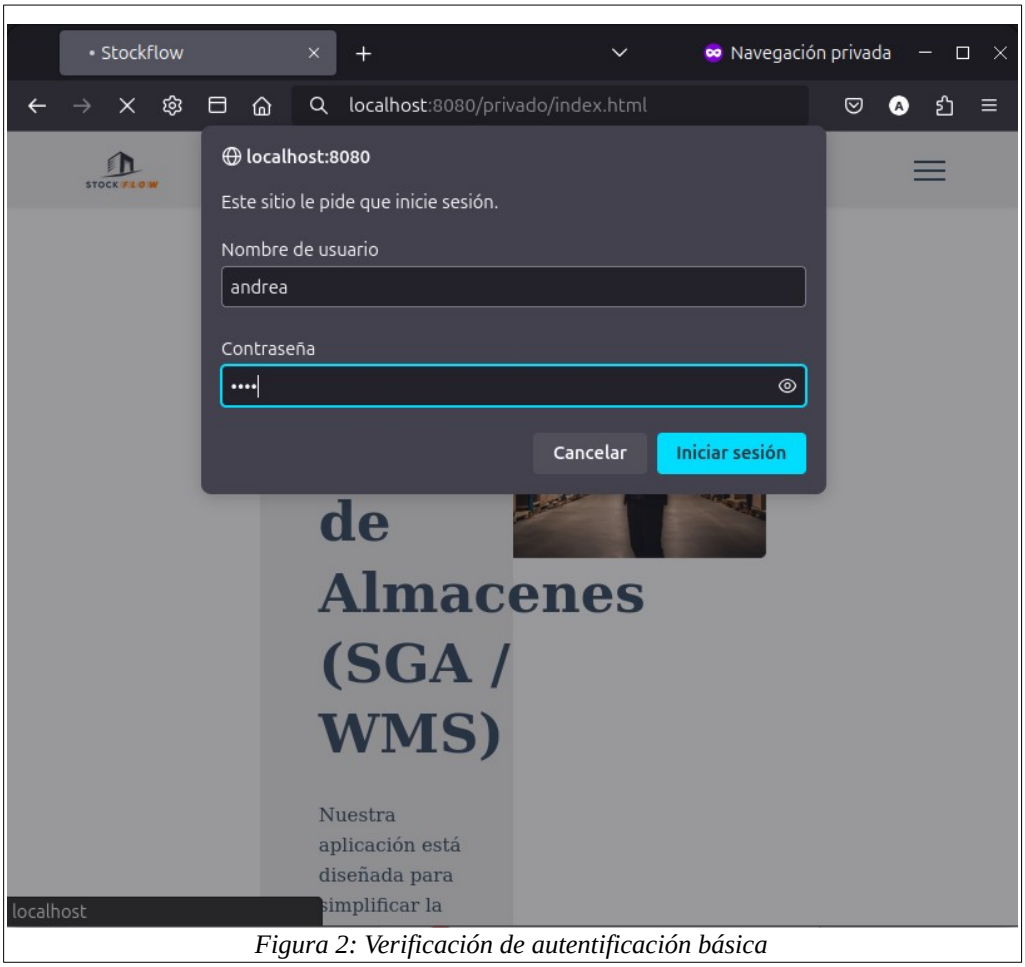

Figura 2: Verificación de autentificación básica

Figura 3: Verificación de autentificación II

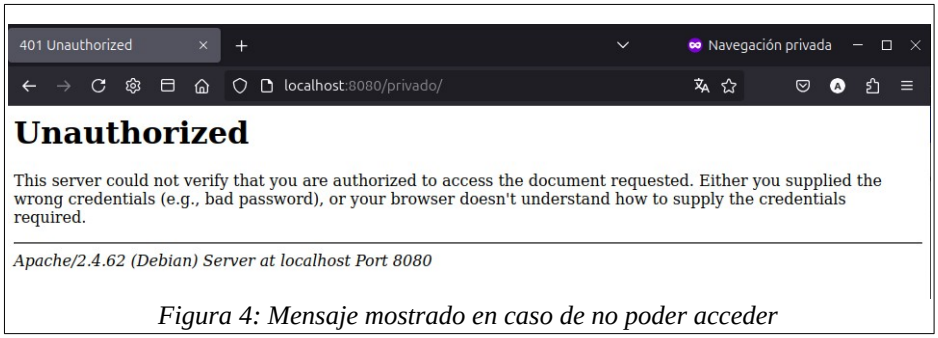

Figura 4: Mensaje mostrado en caso de no poder acceder

b) Para verificar el control de acceso por IP, se accede desde el navegador a las siguientes URL *(ver Figura 5 y 6) :*

http://localhost:8080/restringido/index.html

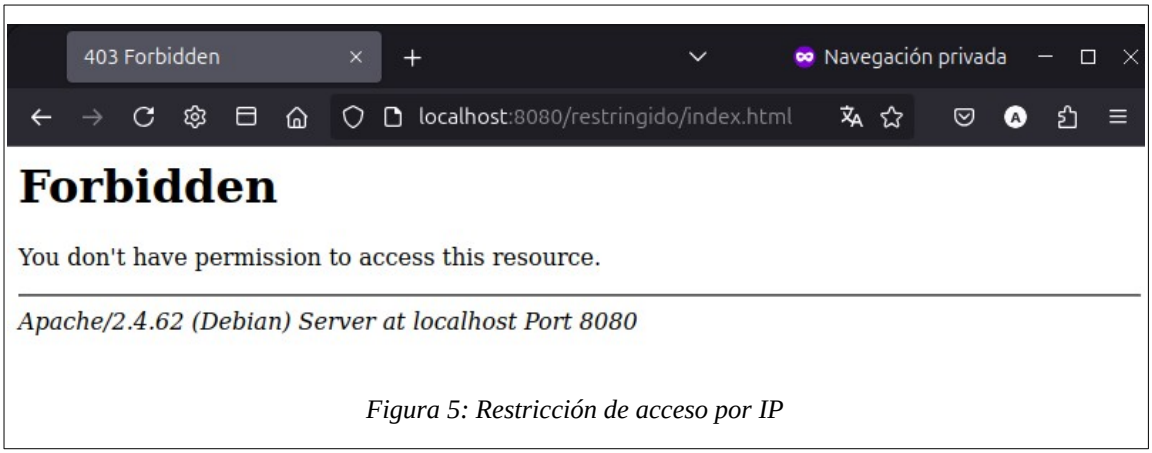

Figura 5: Restricción de acceso por IP

** Acceso permitido si es desde la IP especificada **

Figura 6: Ejemplo de acceso desde la IP especificada

6. Configuración de Apache mediante archivo .htaccess

El archivo htaccess (acceso de hipertexto) es un archivo oculto que se utililiza para configurar funciones adicionales para sitios web alojados en el servidor web Apache. Se puede reescribir la URL, proteger directorios con contraseña, habilitar la protección de enlaces directos, no permitir el acceso a direcciones IP específicas, entre otras cosas.

Este se coloca directamente en el directorio que se desea configurar y las reglas aplicadas afectan solo a ese directorio y sus subdirectorios.

a) Se restablece el archivo de configuración de Apache dejandolo en su configuración por defecto.

b) Se modifica el archivo docker-compose para no montar el volumen ***/conf.***

b) Desde el directorio ***/privado*** se crea el archivo **.htaccess** con las siguientes directivas:

nano .htaccess

→

.htaccess

```
AuthUserFile "/etc/apache2/claves/psswd.txt"
AuthName "ÁREA PRIVADA"
AuthType Basic
Require valid-user
```

c) Desde el directorio ***/restringido*** se crea el archivo **.htaccess** con las siguientes directivas:

nano .htaccess

→

.htaccess
Require ip 192.168.18

d) Una vez creados los archivos, se ejecuta el siguiente comando para que los cambios tengan efecto:

docker-compose up -d –build

e) Luego se verifica el control de acceso por usuario y contraseña desde el navegador accediendo a las siguientes URL *(ver Figura 7)*:

http://*localhost:8080/privado/index.html*

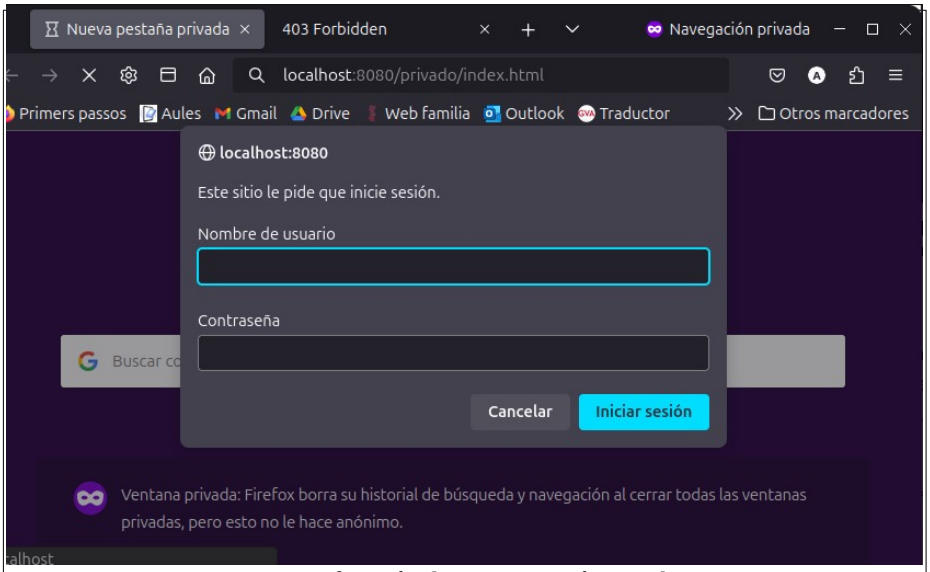

Figura 7: Verificación de autenticación con .htaccess

e) Luego se verifica el control de acceso por IP desde el navegador accediendo a las siguientes URL *(ver Figura 8)*:

http://localhost:8080/restringido/index.html

Figura 8: Restricción de acceso por IP con .htaccess

7. Diferencias entre <Directory> y .htaccess

<Directory>	.htaccess
VENTAJAS: - Al estar configurado en el archivo principal, no necesita buscar archivos *.htaccess* en cada solicitud. - Solo los administradores con acceso al servidor pueden modificar estas configuraciones, aumentando la seguridad. **DESVENTAJAS:** - Solo los administradores pueden modificarlo	**VENTAJAS:** - Se utiliza para configurar accesos y permisos de manera local dentro de un directorio. - Los usuarios con acceso al directorio pueden realizar cambios sin tener que acceder al archivo principal del servidor. **DESVENTAJAS:** - Apache verifica la existencia de este tipo de archivos en cada solicitud, lo que puede ralentizar el servidor si hay muchos accesos.