

6 Guía Reto y Nginx

MATERIAL

- Los contenidos de la unidad y esta guía
- Máquina Virtual Ubuntu 22.04 Desktop.
- Máquina Virtual Ubuntu 18.04 Desktop.
- Virtualbox
- Ordenador con S.O. Windows 10.
- Navegador para comprobar la realización de la tarea.
- Procesador de textos para elaborar la documentación y los archivos de la tarea.
- Acceso a Internet.

6.1 Reto Servicios Web

El primer apartado no se establecen los pasos guiados, ya que se trata de que el alumno articule una mecánica de revisión para dar los pasos oportunos para conseguir localizar y corregir los fallos que aparecen en la máquina virtual facilitada. Para ello deberán tener en cuenta el listado de puntos de revisión facilitado a modo de protocolo, si bien esta no es una lista cerrada y el alumno deberá incorporar los ítems que considere oportunos para perfeccionar su técnica de revisión.

En este apartado vamos a realizar los pasos para corregir los fallos que se dan en una máquina con un Servicio Web basado en Apache. Para ello se facilitará una máquina virtual de Ubuntu 18.04 Desktop

1. Realizar la configuración correspondiente en la máquina principal (Windows 10 pro) para que si usamos el navegador web poniendo las URL de los sitios creados, podamos acceder a ellos.
2. Se dispone de una MV de Ubuntu en la que se han detectado una serie de errores, se han contabilizado hasta 7 errores. Se adjunta enlace a carpeta compartida de Drive donde se guarda la MV (<https://drive.google.com/drive/folders/1e1tRmhQ0->



[EkMkMCsl7euak0nyYjQveVt?usp=sharing](https://www.dpl.local)). La máquina virtual dispone de dos virtualhosts:

- www.empresa-tarea-daw02.local
- www.dpl.local

Incluso cuenta con certificado de seguridad para el dominio de www.empresa-tarea-daw02.local.

Para solucionar los errores debemos protocolarizar como llevar a cabo la revisión. A continuación, indico una serie de pasos para realizarlo:

- **Revisar en la parte de módulos habilitados de apache, que se encuentren cargados los necesarios.**
- **Revisar que los nombres de los archivos de virtualhost sean correctos y estén habilitados. En el caso de que estén incorrectos los nombres de los archivos corregirlos.**
- **Comprobar que el contenido de los archivos de virtualhost es el correcto.**
- **Comprobar la configuración del archivo de los puertos de Apache (/etc/apache2/ports.conf).**
- **Revisar el archivo de configuración de Apache (/etc/apache2/apache2.conf). Para ello nos podemos apoyar en la comprobación de estado del servicio que nos aportará información valiosa sobre el fallo y en la línea en la que se encuentra dentro del archivo de configuración. Recuerda que lo podemos hacer con la orden `sudo systemctl status apache2`.**
- **Comprobar log de registros de errores de Apache para ver si está haciendo correctamente las autenticaciones.**
- **Verificar permisos de las carpetas y archivos.**

Para cargar la MV en VirtualBox debemos descargarnos el archivo del enlace de Drive. Una vez descargado, descomprimirlo en la carpeta donde se guardan las máquinas virtuales en VirtualBox. Para que nos aparezca en el listado de las máquinas de VirtualBox vamos al menú Máquina>Añadir. Finalizamos cambiando la configuración de la tarjeta de red a adaptador puente y actualizando la MAC en la parte de avanzado de las tarjetas de red.

Credenciales MV Ubuntu:

- usuario: **root**
- contraseña: **@qwerty**



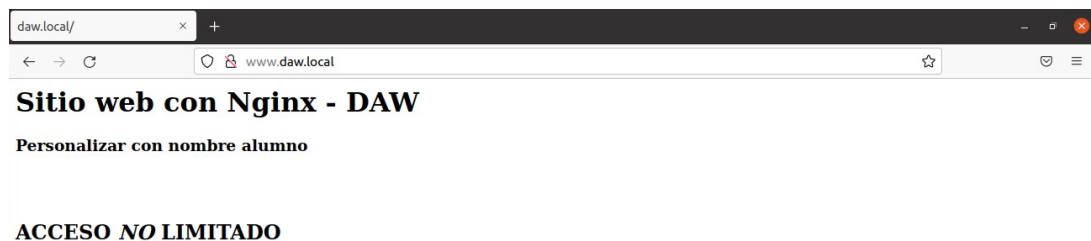
Credenciales MySQL:

- usuario: **root**
- contraseña: **1234**

6.2 Nginx

Realizaremos la instalación del Servidor Web Nginx, para ello podemos emplear la máquina virtual de Ubuntu 22.04 que empleamos para Docker, para evitar que se generen conflictos con el puerto en la que venimos trabajando.

1. Instalar el Servidor Web Nginx, aplicar las reglas de firewall y los permisos correspondientes. Crear un virtualhost o dominio que se llame daw. Comprobar que funciona. Dentro del sitio poner un index.html que tenga el siguiente aspecto:



Instalar Nginx

Antes de instalar el Nginx debemos de verificar que nuestra máquina se encuentra actualizada. Para ello lanzamos los siguientes comandos:

Para comprobar si hay cambios en las versiones de las aplicaciones que tenemos en nuestra máquina con respecto a los repositorios.

#sudo apt update

Tras lanzar el anterior comando procedemos a realizar la actualización de los archivos que ha detectado que existe una actualización:

#sudo apt upgrade

Instalamos el Nginx:

#sudo apt-get install nginx



Ajustes firewall

Comprobamos las configuraciones de aplicación con las que ufw sabe trabajar escribiendo lo siguiente:

#sudo ufw app list

```
informatica@informatica-VirtualBox:/etc/nginx/sites-enabled$ sudo ufw app list
[sudo] contraseña para informatica:
Aplicaciones disponibles:
  CUPS
  Nginx Full
  Nginx HTTP
  Nginx HTTPS
```

Los tres perfiles disponibles son:

- **Nginx Full.** Este perfil abre el puerto 80 (tráfico web normal, no cifrado) y el puerto 443 (tráfico TLS/SSL cifrado)
- **Nginx HTTP.** Este perfil abre solo el puerto 80 (tráfico web normal, no cifrado)
- **Nginx HTTPS.** Este perfil abre solo el puerto 443 (tráfico TLS/SSL cifrado)

En un entorno de trabajo se recomienda emplear el perfil más restrictivo. En nuestro caso sólo vamos a tener tráfico en el puerto 80, por lo tanto es suficiente con habilitar el segundo perfil.

Para habilitarlo:

#sudo ufw allow 'Nginx HTTP'

Para ver las reglas que tiene activas:

#sudo ufw status

En el caso de que nos salga el siguiente mensaje:

```
informatica@informatica-VirtualBox:/etc/nginx/sites-enabled$ sudo ufw status
Estado: inactivo
```

No tenemos habilitado el firewall, para habilitarlos ejecutamos:

#sudo ufw enable



Comprobamos que está activo con:

#sudo ufw status

```
informatica@informatica-VirtualBox:/etc/nginx/sites-enabled$ sudo ufw enable
El cortafuegos está activo y habilitado en el arranque del sistema
informatica@informatica-VirtualBox:/etc/nginx/sites-enabled$ sudo ufw status
Estado: activo

Hasta          Acción      Desde
-----
Nginx HTTP     ALLOW       Anywhere
Nginx HTTP (v6) ALLOW       Anywhere (v6)
```

En el caso de que quisiéramos deshabilitar el firewall lo haríamos con el siguiente comando:

#sudo ufw disable

Y si quisiéramos habilitar un puerto en concreto que esté asociado a algún servicio procederíamos de la siguiente manera:

#sudo ufw allow *número_de_puerto*

Para denegar el acceso a través de un puerto:

#sudo ufw deny *número_de_puerto*

Esto mismo es aplicable a un servicio en concreto. Las reglas son las mismas lo que tendríamos que poner en vez de un número de puerto el nombre del servicio (p.e. **#sudo ufw allow ssh**).

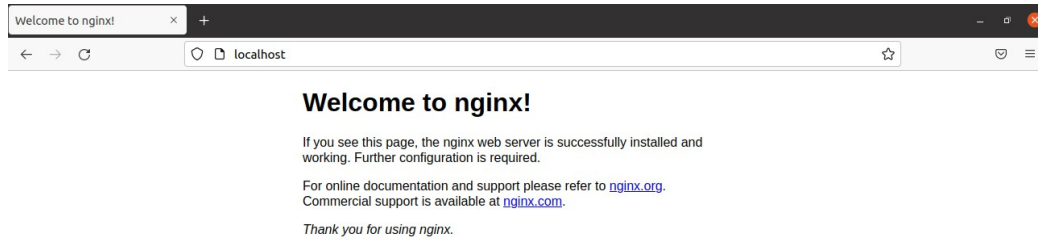
Realizamos la comprobación de que está operativo el Nginx:

#sudo systemctl status nginx

```
informatica@informatica-VirtualBox:/etc/nginx/sites-enabled$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2021-11-08 08:42:34 WET; 2h 4min ago
     Docs: man:nginx(8)
   Process: 5260 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Process: 5261 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 5262 (nginx)
    Tasks: 2 (limit: 2312)
   Memory: 2.4M
   CGroup: /system.slice/nginx.service
           └─5262 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             └─5263 nginx: worker process

nov 08 08:42:34 informatica-VirtualBox systemd[1]: Starting A high performance web server and a reverse proxy server...
nov 08 08:42:34 informatica-VirtualBox systemd[1]: Started A high performance web server and a reverse proxy server.
```

Comprobamos el funcionamiento desde el navegador, poniendo en la barra de direcciones <http://localhost> o 127.0.0.1 o la dirección IP de la máquina con el servidor web.



Del mismo modo que hacíamos con Apache podemos hacer con el Nginx para administrar el servicio:

Iniciarlo:

```
#sudo systemctl start nginx
```

Pararlo:

```
#sudo systemctl stop nginx
```

Reiniciar el servicio:

```
#sudo systemctl restart nginx
```

Recargar el servicio, tener en cuenta que con este comando se recargan las configuraciones sin perder las conexiones establecidas:

```
#sudo systemctl reload nginx
```

Si queremos habilitar el servicio de Nginx en el arranque:

```
#sudo systemctl enable ufw
```

En el caso de que quisiéramos que al arrancar la máquina no se active de manera automática el Nginx.

```
#sudo systemctl disable ufw
```

Configurar sitios virtuales o bloques

Los sitios virtuales o bloques como los define Nginx son espacios de carpetas públicas donde se guardarán los archivos que vamos a servir. Por defecto el Nginx dispone de un directorio donde se suministra la página a la que accedemos y nos da la bienvenida si hacemos y localhost con un navegador. Esta carpeta está ubicada en:

```
/var/www/html/
```

En el caso de que quisiéramos disponer de nuevos sitios virtuales creamos la carpeta con el nombre del dominio para identificarlo a partir de la ruta:

```
/var/www/
```



Como nos pide la actividad que creemos el dominio www.daw.local, vamos a crear el directorio de la carpeta pública donde se va a alojar:

```
#sudo mkdir -p /var/www/daw/html/
```

-p indica que nos cree cualquier carpeta anterior que sea necesaria para obtener la ruta que le pedimos.

Asignamos como propietario de la carpeta creada a \$USER de manera recursiva para que sea aplicado a todos los directorios y subdirectorios.

```
#sudo chown -R $USER:$USER /var/www/daw/html/
```

Aplicamos los permisos de lectura, escritura y ejecución correspondientes a las carpetas (Propietario-Grupo-Otros).

```
#sudo chmod 755 /var/www/daw/html/
```

Después de dar los permisos, creamos el index.html.

```
#sudo nano /var/www/daw/html/index.html
```

```
GNU nano 4.8 /var/www/daw/html/index.html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <h1>Sitio web con Nginx - DAW</h1>
    <h3>Personalizar con nombre alumno</h3>
  </head>
  <body>
    <br>
    <br>
    <h2>ACCESO <i>NO</i> LIMITADO</h2>
  </body>
</html>
```

Tenemos que crear el archivo de configuración del dominio creado. Al igual que el Apache debemos crear el archivo de configuración en sites-available, de la siguiente forma:

```
#sudo nano /etc/nginx/sites-available/daw
```

```
GNU nano 4.8 /etc/nginx/sites-available/daw
server {
    listen 80;
    listen [::]:80;

    root /var/www/daw/html;
    index index.html index.htm index.nginx-debian.html;

    server_name daw.local www.daw.local;

    location / {
        try_files $uri $uri/ =404;
    }
}
```



Ahora creamos el enlace simbólico a la carpeta de sites-enabled para que quede habilitado:

```
#sudo ln -s /etc/nginx/sites-available/daw /etc/nginx/sites-enabled/
```


Descomentamos la siguiente línea dentro del nginx.conf:

```
# server_names_hash_bucket_size 64;
```

Para ello lo editamos así:

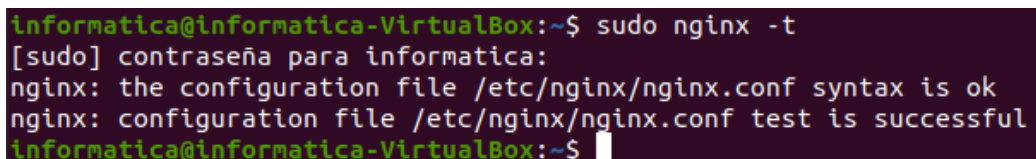
```
# sudo nano /etc/nginx/nginx.conf
```

Quedando:



Comprobamos que la configuración está realizada de manera correcta:

```
#sudo nginx -t
```



Añadimos la relación dominio – IP de la máquina servidora para que al poner en la barra de direcciones el dominio sepa resolver a que máquina debe dirigirse para mostrarla.

```
# sudo nano /etc/hosts
```




```
GNU nano 4.8 /etc/hosts
127.0.0.1    localhost www.daw.local
127.0.1.1    informatica-VirtualBox

# The following lines are desirable for IPv6 capable hosts
::1         ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
```

Ponemos la URL en el navegador y accedemos:



Sitio web con Nginx - DAW

Personalizar con nombre alumno

ACCESO NO LIMITADO

Reiniciamos el servicio.

#sudo systemctl restart nginx

2. Configurar el puerto del Servidor Web Nginx para que sirva las peticiones web en el puerto 85.

En el archivo de configuración del sitio debemos sustituir en los Listen el puerto 80 por el 85.

#sudo nano /etc/nginx/sites-available/daw



```
GNU nano 4.8 /etc/nginx/sites-available/daw
server {
    listen 85;
    listen [::]:85;

    root /var/www/daw/html;
    index index.html index.htm index.nginx-debian.html;

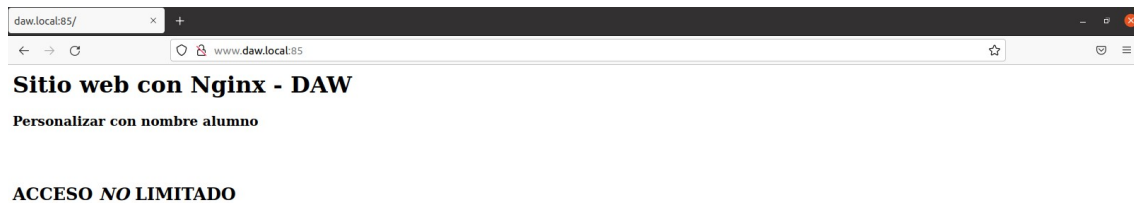
    server_name daw.local www.daw.local;

    location / {
        try_files $uri $uri/ =404;
    }
}
```

Reiniciamos el servicio.

#sudo systemctl restart nginx

Accedemos a la página del sitio indicándole el puerto 85.



3. Crear un acceso limitado dentro del dominio www.daw.local para que controle el acceso de los usuarios al mismo.

Editamos el archivo de configuración de nuestro sitio y añadimos las líneas de control de la autenticación básica.



```
GNU nano 4.8 /etc/nginx/sites-available/daw
server {
    listen 85;
    listen [::]:85;

    root /var/www/daw/html;
    index index.html index.htm index.nginx-debian.html;

    server_name daw.local www.daw.local;

    location / {
        try_files $uri $uri/ =404;
        ##Autenticación Basic
        auth_basic "Acceso restringido";
        auth_basic_user_file /etc/nginx/daw/.htpasswd;
    }
}
```

Comprobamos que no hay ningún tipo de error en la configuración de nginx.

#sudo nginx -t

```
Informatica@Informatica-VirtualBox:~$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Instalamos la utilidad de apache para controlar la autenticación básica con la siguiente orden:

#sudo apt-get install apache2-utils

```
Informatica@Informatica-VirtualBox:~$ sudo apt-get install apache2-utils
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  libfprint-2-tod1 liblvm10
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  libapr1 libaprutil1
Se instalarán los siguientes paquetes NUEVOS:
  apache2-utils libapr1 libaprutil1
0 actualizados, 3 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 261 kB de archivos.
Se utilizarán 970 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```

Creamos un directorio para el sitio y el usuario “dpl” y nos pedirá la contraseña con los comandos:

#sudo mkdir /etc/nginx/daw

#cd /etc/nginx/daw

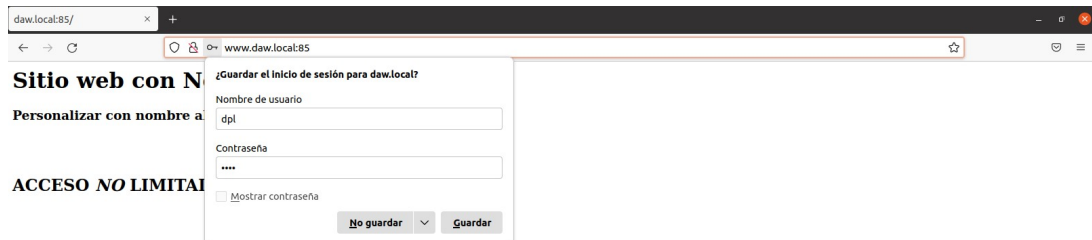
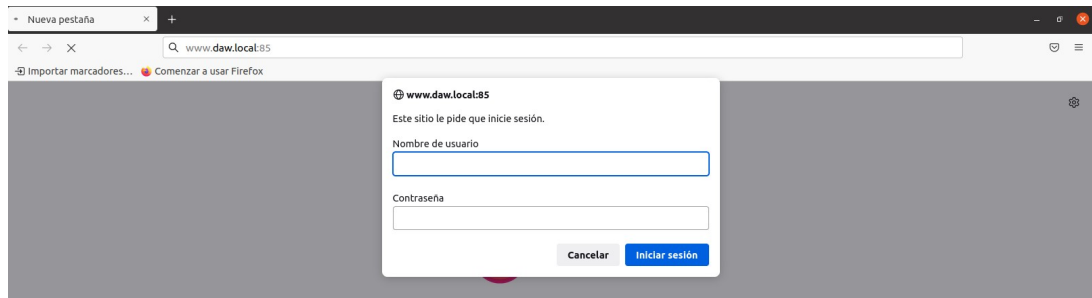
#htpasswd -c -m .htpasswd dpl



```
informatica@informatica-VirtualBox:~$ cd /etc/nginx/
informatica@informatica-VirtualBox:~$ cd /etc/nginx$
informatica@informatica-VirtualBox:~$ ls
conf.d          modules-available  sites-enabled
fastcgi.conf    modules-enabled    snippets
fastcgi_params  nginx.conf         uwsgi_params
koi-utf         proxy_params      win-utf
koi-win         scgi_params
mime.types      sites-available
informatica@informatica-VirtualBox:~$ sudo mkdir daw
informatica@informatica-VirtualBox:~$ cd daw/
informatica@informatica-VirtualBox:~$ sudo htpasswd -c -m .htpasswd dpl
New password:
Re-type new password:
Adding password for user dpl
informatica@informatica-VirtualBox:~$ sudo systemctl restart nginx
```

Reiniciamos la máquina para que sean efectivos los cambios.

#sudo systemctl restart nginx



Sitio web con Nginx - DAW

Personalizar con nombre alumno

ACCESO LIMITADO