

PRÁCTICA 4: Instalación y configuración de un servidor Web en Debian.

Servicios de Red e Internet

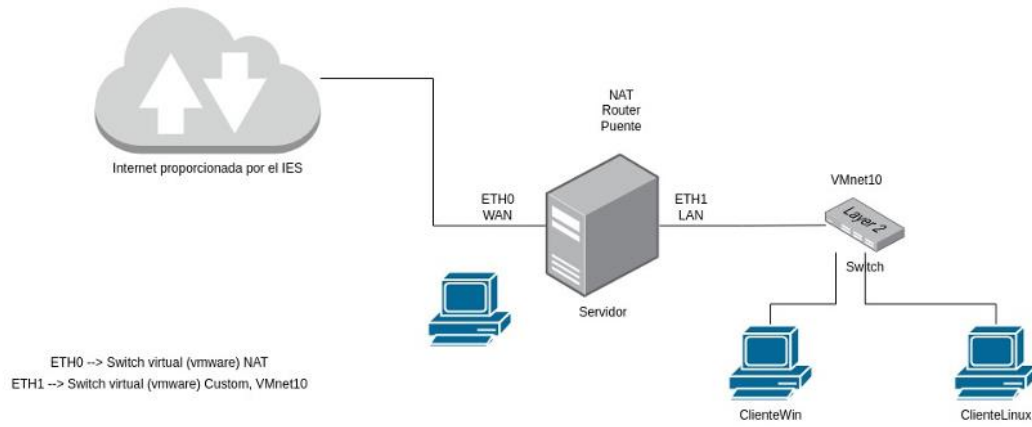
Ana María Serrano Pedrajas

Contenido

1. Introducción.....	2
2. Instala el servicio Apache	3
3. Crear un servidor compartido.....	3
a) Sitios web virtuales	4
b) Redireccionamiento entre directorios	9
c) Seguridad	11
d) Instalar módulos	15
e) Montar un CMS como es Moodle	16
4. Actividades / capturas a entregar	18
1) Captura accediendo desde un cliente a http://www.1asir.local así como a https://www.1asir.local	20
2) Captura accediendo a la sección de 1asir llamada Fundamentos Hardware, donde se aprecie que se ha logueado el usuario llamado “miguel”.	21
3) Captura accediendo a la sección de 1asir llamada Redes.	21
4) Captura accediendo desde un cliente a http://www.2asir.local así como a https://www.2asir.local . Explica como se ha solucionado la problemática asociada al acceso via http.	21
5) Captura accediendo al sitio de 2asir.local desde un cliente con ip 10.0.0.35.	22
6) Captura accediendo desde un cliente a http://www.1daw.local así como a https://www.1daw.local	23
7) Captura donde se muestre que se ha realizado la redirección hacia la sección de sistemas informáticos dentro del sitio 1daw.local.	23

1. Introducción

A partir del siguiente esquema de red, vamos a contestar a una serie de preguntas en las siguientes secciones.

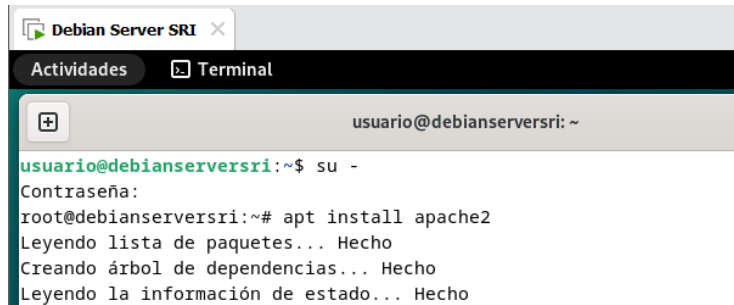


2. Instala el servicio Apache

Estamos utilizando un servidor debian, por lo que solo tenemos ejecutar los siguientes comandos:

su - (cambiar a usuario administrador)

apt install apache2 (instalar paquete necesario)



```
usuario@debianserversri:~$ su -
Contraseña:
root@debianserversri:~# apt install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
```

3. Crear un servidor compartido

La estructura de sitios web a implementar es el siguiente:

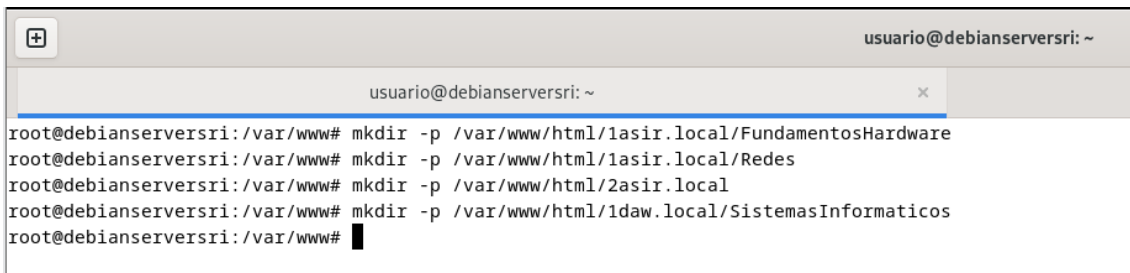
- **Sitio web: Primero de ASIR → www.1asir.local** . Trabajaré en el puerto 80 exclusivamente. Tendrá un archivo html dando la bienvenida al sitio. Tendrá la siguiente estructura de carpetas:
 - **FundamentosHardware**: con acceso restringido cifrado para un grupo de usuarios (miguel, juan e irene). Tendrá un archivo html dando la bienvenida a la sección.
 - **Redes**: al que podrá acceder cualquier usuario. Tendrá un archivo html dando la bienvenida a la sección. Será un directorio virtual.
- **Sitio web: Segundo de ASIR → www.2asir.local** . Trabajaré en el puerto 443 exclusivamente. Instalarás un moodle sobre dicho sitio. Soluciona la problemática asociada al acceso vía http.
 - Se **restringirá** el acceso a la ip **10.0.0.35**
- **Sitio web: Primero de DAW → www.1daw.local** . Trabajaré tanto en los puertos 80 como 443. Tendrá un archivo html dando la bienvenida al sitio. Tendrá la siguiente estructura de carpetas:
 - **SistemasInformáticos**: Tendrá un archivo html dando la bienvenida a la sección.
 - **Redirección**: cada vez que un usuario desee entrar al sitio www.1daw.local será redirigido a la sección Sistemas Informáticos

Debemos crear lo siguiente:

a) Sitios web virtuales

Vamos a crear en primer lugar todas las carpetas que necesitamos para alojar todos los archivos index.html. Utilizaremos los siguientes comandos:

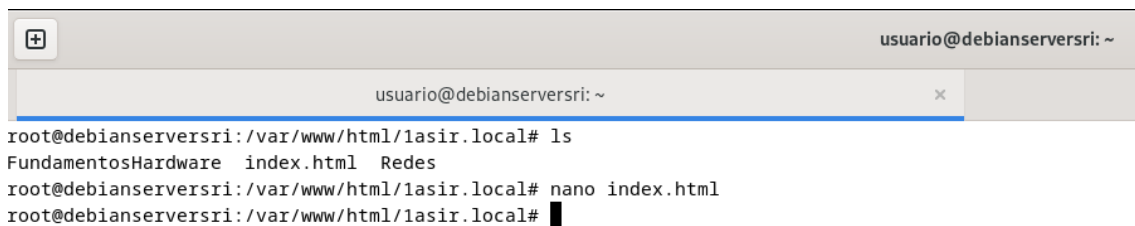
```
mkdir -p /var/www/html/1asir.local/FundamentosHardware
mkdir -p /var/www/html/1asir.local/Redes
mkdir -p /var/www/html/2asir.local
mkdir -p /var/www/html/1daw.local/SistemasInformaticos
```



```
usuario@debianserversri: ~
root@debianserversri:/var/www# mkdir -p /var/www/html/1asir.local/FundamentosHardware
root@debianserversri:/var/www# mkdir -p /var/www/html/1asir.local/Redes
root@debianserversri:/var/www# mkdir -p /var/www/html/2asir.local
root@debianserversri:/var/www# mkdir -p /var/www/html/1daw.local/SistemasInformaticos
root@debianserversri:/var/www#
```

A continuación, creamos los “index” o páginas Web principales de cada sitio. Necesitaremos crear los siguientes:

1) /var/www/html/1asir.local/index.html



```
usuario@debianserversri: ~
root@debianserversri:/var/www/html/1asir.local# ls
FundamentosHardware index.html Redes
root@debianserversri:/var/www/html/1asir.local# nano index.html
root@debianserversri:/var/www/html/1asir.local#
```



```
GNU nano 7.2 index.html
<!DOCTYPE html>
<html>
<head>
  <title>Primero de ASIR</title>
</head>
<body>
  <h1>Bienvenido al sitio de Primero de ASIR</h1>
</body>
</html>
```

2) /var/www/html/1asir.local/FundamentosHardware/index.html

```
usuario@debianserversri: ~
usuario@debianserversri: ~
root@debianserversri:/var/www/html/1asir.local# ls
FundamentosHardware index.html Redes
root@debianserversri:/var/www/html/1asir.local# cd FundamentosHardware/
root@debianserversri:/var/www/html/1asir.local/FundamentosHardware# ls
index.html
root@debianserversri:/var/www/html/1asir.local/FundamentosHardware# nano index.html
root@debianserversri:/var/www/html/1asir.local/FundamentosHardware#
```

```
usuario@debianserversri: ~
usuario@debianserversri: ~
GNU nano 7.2 index.html
<!DOCTYPE html>
<html>
<head>
  <title>1 ASIR - Fundamentos de Hardware</title>
</head>
<body>
  <h1>Bienvenido a Fundamentos de Hardware</h1>
</body>
</html>
```

3) /var/www/html/1asir.local/Redes/index.html

```
usuario@debianserversri: ~
usuario@debianserversri: ~
root@debianserversri:/var/www/html/1asir.local# ls
FundamentosHardware index.html Redes
root@debianserversri:/var/www/html/1asir.local# cd Redes
root@debianserversri:/var/www/html/1asir.local/Redes# ls
index.html
root@debianserversri:/var/www/html/1asir.local/Redes# nano index.html
root@debianserversri:/var/www/html/1asir.local/Redes#
```

```
usuario@debianserversri: ~
usuario@debianserversri: ~
GNU nano 7.2 index.html
<!DOCTYPE html>
<html>
<head>
  <title>1 ASIR - Redes</title>
</head>
<body>
  <h1>Bienvenido a Redes</h1>
</body>
</html>
```

4) /var/www/html/2asir.local/index.html

```
usuario@debianserversri: ~  
usuario@debianserversri: ~  
root@debianserversri:/var/www/html/2asir.local# ls  
index.html  
root@debianserversri:/var/www/html/2asir.local# nano index.html  
root@debianserversri:/var/www/html/2asir.local#
```

```
usuario@debianserversri: ~  
usuario@debianserversri: ~  
GNU nano 7.2 index.html  
<!DOCTYPE html>  
<html>  
<head>  
  <title>Segundo de ASIR</title>  
</head>  
<body>  
  <h1>Bienvenido al sitio de Segundo de ASIR</h1>  
</body>  
</html>
```

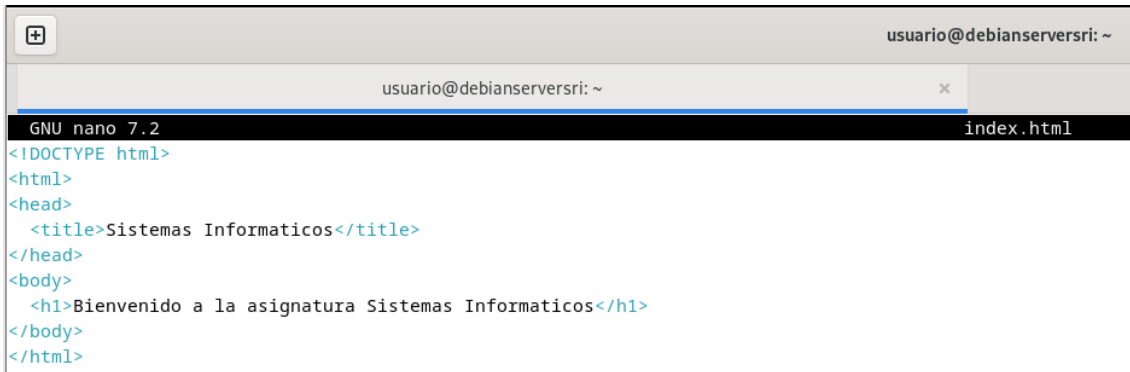
5) /var/www/html/1daw.local/index.html

```
usuario@debianserversri: ~  
usuario@debianserversri: ~  
root@debianserversri:/var/www/html/1daw.local# ls  
index.html SistemasInformaticos  
root@debianserversri:/var/www/html/1daw.local# nano index.html  
root@debianserversri:/var/www/html/1daw.local#
```

```
usuario@debianserversri: ~  
usuario@debianserversri: ~  
GNU nano 7.2 index.html  
<!DOCTYPE html>  
<html>  
<head>  
  <title>Primero de DAW</title>  
</head>  
<body>  
  <h1>Bienvenido al sitio de Primero de DAW</h1>  
</body>  
</html>
```

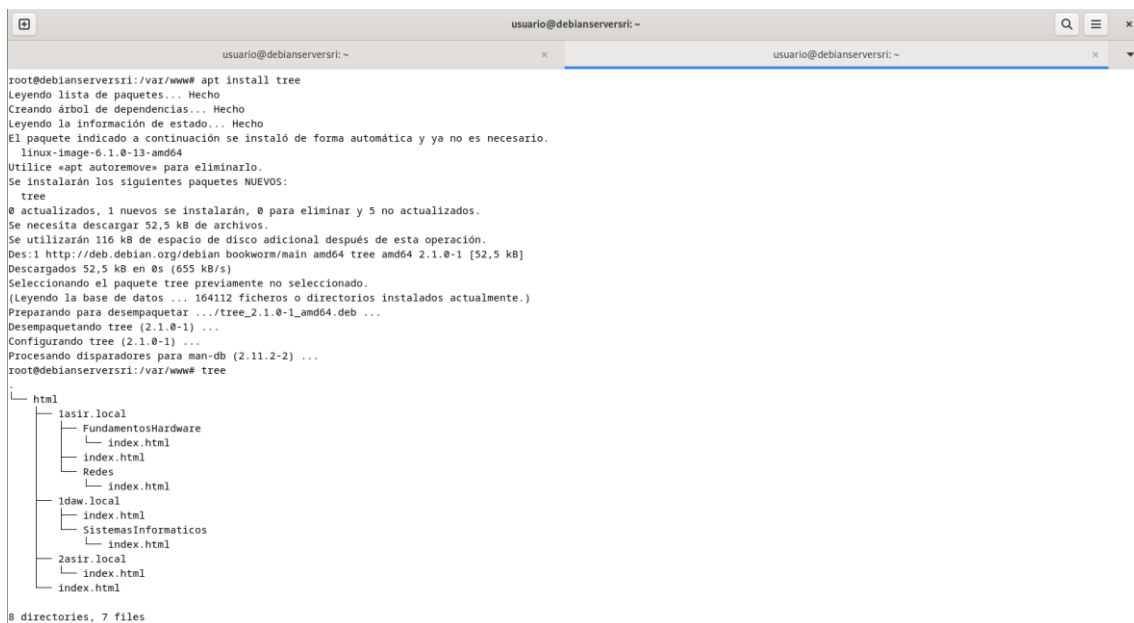
6) /var/www/html/1daw.local/SistemasInformaticos/index.html

```
usuario@debianserversri: ~  
usuario@debianserversri: ~  
root@debianserversri:/var/www/html/1daw.local# ls  
index.html SistemasInformaticos  
root@debianserversri:/var/www/html/1daw.local# cd SistemasInformaticos/  
root@debianserversri:/var/www/html/1daw.local/SistemasInformaticos# ls  
index.html  
root@debianserversri:/var/www/html/1daw.local/SistemasInformaticos# nano index.html  
root@debianserversri:/var/www/html/1daw.local/SistemasInformaticos#
```



```
GNU nano 7.2 index.html
<!DOCTYPE html>
<html>
<head>
  <title>Sistemas Informaticos</title>
</head>
<body>
  <h1>Bienvenido a la asignatura Sistemas Informaticos</h1>
</body>
</html>
```

Podemos instalar la herramienta “tree” para ver cómo queda el árbol de archivos creado (apt install tree). Luego ejecutamos “tree” y vemos que hemos creado nuestros archivos y directorios correctamente:



```
root@debianserver1: /var/www# apt install tree
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  linux-image-6.1.0-13-amd64
Utilice «apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes NUEVOS:
  tree
0 actualizados, 1 nuevos se instalarán, 0 para eliminar y 5 no actualizados.
Se necesita descargar 52,5 kB de archivos.
Se utilizarán 116 kB de espacio de disco adicional después de esta operación.
Des:1 http://deb.debian.org/debian bookworm/main amd64 tree amd64 2.1.0-1 [52,5 kB]
Descargados 52,5 kB en 0s (655 kB/s)
Seleccionando el paquete tree previamente no seleccionado.
(Leyendo la base de datos ... 164112 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../tree_2.1.0-1_amd64.deb ...
Desempaquetando tree (2.1.0-1) ...
Configurando tree (2.1.0-1) ...
Procesando disparadores para man-db (2.11.2-2) ...
root@debianserver1: /var/www# tree
.
├── html
│   ├── 1asir.local
│   │   ├── FundamentosHardware
│   │   │   └── index.html
│   │   ├── index.html
│   │   └── Redes
│   │       └── index.html
│   ├── 1daw.local
│   │   ├── index.html
│   │   └── SistemasInformaticos
│   │       └── index.html
│   └── 2asir.local
│       ├── index.html
│       └── index.html
└── 8 directories, 7 files
```

Ahora nos movemos al directorio de configuración de Apache donde crearemos una versión inicial de los sitios virtuales:

- `/etc/apache2/sites-available/1asir.local.conf`

```
<VirtualHost *:80>
    ServerName www.1asir.local
    DocumentRoot /var/www/html/1asir.local
</VirtualHost>
```

Ejecutamos el comando “a2ensite 1asir.local.conf” y reiniciamos apache con “systemctl restart apache2” para habilitar este sitio web.


```
usuario@debianserversri: ~
usuario@debianserversri: ~
root@debianserversri:/etc/apache2/sites-available# nano 1asir.local.conf
root@debianserversri:/etc/apache2/sites-available# cat 1asir.local.conf
<VirtualHost *:80>
    ServerName www.1asir.local
    DocumentRoot /var/www/html/1asir.local
</VirtualHost>
root@debianserversri:/etc/apache2/sites-available# a2ensite 1asir.local.conf
Enabling site 1asir.local.
To activate the new configuration, you need to run:
    systemctl reload apache2
root@debianserversri:/etc/apache2/sites-available# systemctl restart apache2
root@debianserversri:/etc/apache2/sites-available#
```

- **/etc/apache2/sites-available/1daw.local.conf**

```
<VirtualHost *:80>
    ServerName www.1daw.local
    DocumentRoot /var/www/html/1daw.local
</VirtualHost>
```

Ejecutamos el comando “a2ensite 1daw.local.conf” y reiniciamos apache con “systemctl restart apache2” para habilitar este sitio web.

```
usuario@debianserversri: ~
usuario@debianserversri: ~
root@debianserversri:/etc/apache2/sites-available# nano 1daw.local.conf
root@debianserversri:/etc/apache2/sites-available# cat 1daw.local.conf
<VirtualHost *:80>
    ServerName www.1daw.local
    DocumentRoot /var/www/html/1daw.local
</VirtualHost>
root@debianserversri:/etc/apache2/sites-available# a2ensite 1daw.local.conf
Enabling site 1daw.local.
To activate the new configuration, you need to run:
    systemctl reload apache2
root@debianserversri:/etc/apache2/sites-available# systemctl restart apache2
root@debianserversri:/etc/apache2/sites-available#
```

- **/etc/apache2/sites-available/2asir.local.conf**

```
<VirtualHost *:80>
    ServerName www.2asir.local
    DocumentRoot /var/www/html/2asir.local
</VirtualHost>
```

Ejecutamos el comando “a2ensite 2asir.local.conf” y reiniciamos apache con “systemctl restart apache2” para habilitar este sitio web.

```
usuario@debianserversri: ~
usuario@debianserversri: ~
root@debianserversri:/etc/apache2/sites-available# nano 2asir.local.conf
root@debianserversri:/etc/apache2/sites-available# cat 2asir.local.conf
<VirtualHost *:80>
    ServerName www.2asir.local
    DocumentRoot /var/www/html/2asir.local
</VirtualHost>
root@debianserversri:/etc/apache2/sites-available# a2ensite 2asir.local.conf
Enabling site 2asir.local.
To activate the new configuration, you need to run:
    systemctl reload apache2
root@debianserversri:/etc/apache2/sites-available# systemctl restartrestart apache2
Unknown command verb restartrestart.
root@debianserversri:/etc/apache2/sites-available# systemctl restart apache2
root@debianserversri:/etc/apache2/sites-available#
```

b) Redireccionamiento entre directorios

Tanto para www.1asir.local como para www.1daw.local, en el enunciado se solicita que se tenga acceso a subcarpetas dentro de estos sitios web. Vamos a realizar la configuración de estas carpetas:

- Sitio web www.1asir.local – puerto 80 exclusivamente
 - **FundamentosHardware:** para esta carpeta se requiere un archivo html dando la bienvenida a la sección que ya creamos en un punto anterior, pero además se solicita que exista acceso restringido cifrado para un grupo de usuarios (miguel, juan e irene).
Necesitamos por tanto generar un archivo con las credenciales de estos usuarios, por lo que utilizaremos la utilidad htpasswd de esta forma:

```
htpasswd -c /etc/apache2/.htpasswd miguel
htpasswd /etc/apache2/.htpasswd juan
htpasswd /etc/apache2/.htpasswd irene
```

IMPORTANTE: usar el parámetro “-c” solo en la primera línea para crear el archivo si no existe o solo se creará el último par usuario/password.
Para todos estos usuarios la contraseña será “usuario”.

```
usuario@debianserversri: ~
usuario@debianserversri: ~
root@debianserversri:/etc/apache2/sites-available# htpasswd -c /etc/apache2/.htpasswd miguel
New password:
Re-type new password:
Adding password for user miguel
root@debianserversri:/etc/apache2/sites-available# htpasswd /etc/apache2/.htpasswd juan
New password:
Re-type new password:
Adding password for user juan
root@debianserversri:/etc/apache2/sites-available# htpasswd /etc/apache2/.htpasswd irene
New password:
Re-type new password:
Adding password for user irene
root@debianserversri:/etc/apache2/sites-available#
```

Añadiremos un `<Directory>` a la configuración en `/etc/apache2/sites-available/1asir.local.conf` con la configuración de autenticación básica apuntando al fichero de autenticación generado previamente.

```
<VirtualHost *:80>
    ServerName www.1asir.local
    DocumentRoot /var/www/html/1asir.local

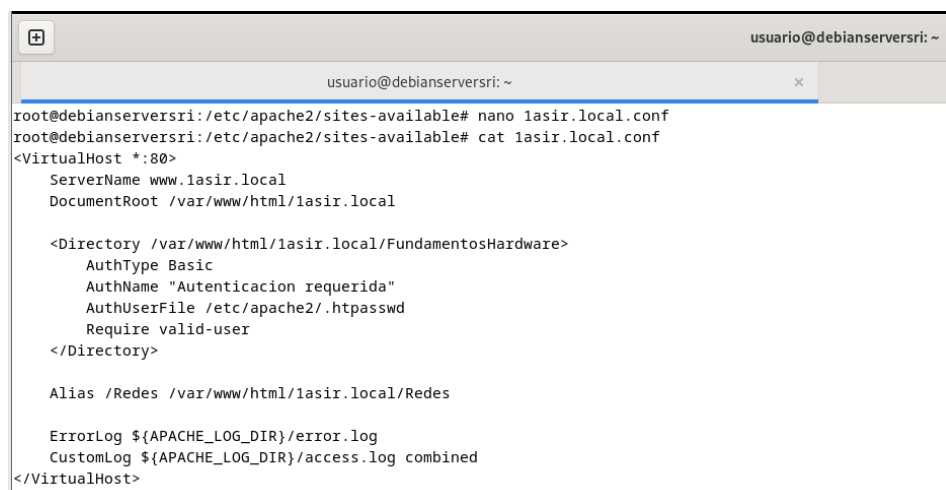
    <Directory /var/www/html/1asir.local/FundamentosHardware>
        AuthType Basic
        AuthName "Autenticacion requerida"
        AuthUserFile /etc/apache2/.htpasswd
        Require valid-user
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Debemos reiniciar Apache para que los cambios surtan efecto con “systemctl restart apache2”.

- **Redes:** para esta carpeta solo se requiere un archivo html dando la bienvenida a la sección, sin ningún requerimiento adicional, por lo que simplemente añadimos un alias a la ubicación a la que queremos redirigir, justo debajo del `<Directory>` anterior. Reiniciamos Apache para comprobar la configuración.

```
Alias /Redes /var/www/html/1asir.local/Redes
```



```
usuario@debianserversri: ~
root@debianserversri:/etc/apache2/sites-available# nano 1asir.local.conf
root@debianserversri:/etc/apache2/sites-available# cat 1asir.local.conf
<VirtualHost *:80>
    ServerName www.1asir.local
    DocumentRoot /var/www/html/1asir.local

    <Directory /var/www/html/1asir.local/FundamentosHardware>
        AuthType Basic
        AuthName "Autenticacion requerida"
        AuthUserFile /etc/apache2/.htpasswd
        Require valid-user
    </Directory>

    Alias /Redes /var/www/html/1asir.local/Redes

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

- Sitio web www.1daw.local – puertos 80 y 443
 - **SistemasInformaticos:** para esta carpeta, solo tendrá un archivo html dando la bienvenida a la sección, pero **no vamos a incluir otro alias** como en el apartado anterior porque se nos pide algo diferente en el siguiente apartado de redirección.

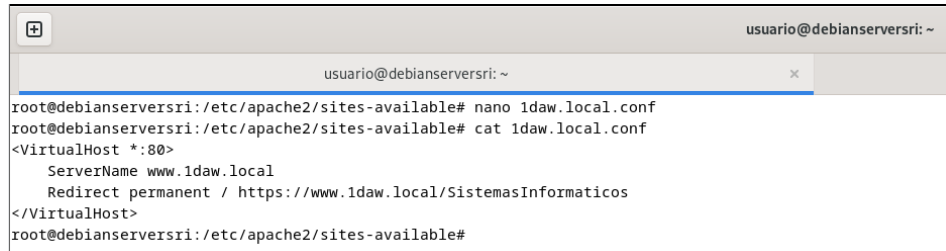
- **Redirección:** en este caso, cada vez que intentamos entrar a www.1daw.local, se nos debe redirigir directamente a la sección “Sistemas Informaticos” cuya configuración hemos completado en el punto anterior. Por tanto, en lugar de un alias, eliminaremos la línea de DocumentRoot

```
DocumentRoot /var/www/html/1daw.local
```

para pasar a tener una redirección:

```
Redirect permanent / https://www.1daw.local/SistemasInformaticos
```

La configuración quedaría así:



```
usuario@debianserversri: ~
usuario@debianserversri: ~
root@debianserversri:/etc/apache2/sites-available# nano 1daw.local.conf
root@debianserversri:/etc/apache2/sites-available# cat 1daw.local.conf
<VirtualHost *:80>
    ServerName www.1daw.local
    Redirect permanent / https://www.1daw.local/SistemasInformaticos
</VirtualHost>
root@debianserversri:/etc/apache2/sites-available#
```

c) Seguridad

Se nos solicita configurar lo siguiente:

- Accesos a recursos
- HTTPS

Vamos a ver qué implica a nivel de configuración. Según el enunciado, tanto para www.2asir.local como para www.1daw.local, se nos requieren cambios en la **configuración de https o restricción de accesos a recursos por IP**. Vamos a ver la configuración requerida para ambos sitios:

- Para el sitio web www.2asir.local – **debe trabajar en el puerto 443 exclusivamente**, para el sitio web www.1daw.local – **debe trabajar tanto en los puertos 80 como en el 443**. El sitio web restante www.1asir.local **se queda en el puerto 80** por lo que no necesita configuración de https.

Para esta tarea, vamos a utilizar OpenSSL para generar un certificado autofirmado y luego configuraremos nuestro servidor web Apache para que lo utilice.

1. Generar la clave privada y el certificado autofirmado:

Primero, generamos una clave privada RSA de 2048 bits con el siguiente comando:

```
openssl genrsa -out autofirmado_privada.key 2048
```

- **openssl genrsa:** Este comando de OpenSSL se utiliza para generar claves RSA.
- **-out autofirmado_privada.key:** Especifica el nombre del archivo donde se guardará la clave privada. En este caso, la guardamos en un archivo llamado "autofirmado_privada.key".

- **2048:** Indica la longitud de la clave en bits. Hemos utilizado 2048 bits, que es un tamaño recomendado para la seguridad actual.

En el siguiente paso utilizamos esta clave para crear un certificado autofirmado válido por 365 días:

openssl req -x509 -new -nodes -key autofirmado_privada.key -sha256 -days 365 -out autofirmado_certificado.crt

- **openssl req:** Este comando se utiliza para generar solicitudes de firma de certificado (CSR) y certificados autofirmados.
- **-x509:** Indica que queremos generar un certificado autofirmado.
- **-new:** Especifica que estamos creando una nueva solicitud de certificado.
- **-nodes:** Evita que se nos solicite una contraseña para la clave privada.
- **-key autofirmado_privada.key:** Especifica el archivo de la clave privada que se utilizará para firmar el certificado.
- **-sha256:** Utiliza el algoritmo SHA-256 para firmar el certificado.
- **-days 365:** Indica la validez del certificado en días. En este caso, el certificado será válido por un año.
- **-out autofirmado_certificado.crt:** Especifica el nombre del archivo donde se guardará el certificado.

Durante la creación del certificado, OpenSSL nos solicita información sobre nuestra organización y dominio. Rellenamos los campos con datos relevantes para nuestro sitio web:

```
usuario@debianserversri: ~
usuario@debianserversri: ~
root@debianserversri:/home/usuario# openssl genrsa -out autofirmado_privada.key 2048
root@debianserversri:/home/usuario# openssl req -x509 -new -nodes -key autofirmado_privada.key -sha256 -days 365 -out autofirmado_certificado.crt
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:Cordoba
Locality Name (eg, city) []:Cordoba
Organization Name (eg, company) [Internet Widgits Pty Ltd]:IES Gran Capitan
Organizational Unit Name (eg, section) []:Dpto Informatica
Common Name (e.g. server FQDN or YOUR name) []:Ana Serrano
Email Address []:a22sepean@iesgrancapitan.org
root@debianserversri:/home/usuario# ls -l autofirmado*
-rw-r--r-- 1 root root 1521 nov 17 15:33 autofirmado_certificado.crt
-rw----- 1 root root 1704 nov 17 15:32 autofirmado_privada.key
root@debianserversri:/home/usuario#
```

Nos movemos al directorio de configuración de Apache:

```
cd /etc/apache2/sites-enabled
```

Habilitamos los sitios SSL en nuestro servidor web:

```
a2enmod ssl
systemctl restart apache2
```

```
root@debianserversri:/home/usuario# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
    systemctl restart apache2
(reverse-i-search)`rest': sudo systemctl ^Cstart bind9
root@debianserversri:/home/usuario# systemctl restart apache2
```

Copiamos el certificado y la clave privada que generamos previamente a las ubicaciones `/etc/ssl/certs` y `/etc/ssl/private`, respectivamente.

```
cp /home/usuario/autofirmado_privada.key /etc/ssl/private/
cp /home/usuario/autofirmado_certificado.crt /etc/ssl/certs/
```

A continuación, editamos con nano los archivos de configuración de los virtualhost de nuestros sitios web en Apache para habilitar HTTPS y especificar la ubicación de la clave privada y el certificado.

Añadiremos a ambos sitios www.2asir.local y www.1daw.local la siguiente configuración en un VirtualHost *:443:

```
SSLEngine on
SSLCertificateFile      /etc/ssl/certs/autofirmado_certificado.crt
SSLCertificateKeyFile   /etc/ssl/private/autofirmado_privada.key
```

Así es como quedaría la configuración de `/etc/apache2/sites-available/2asir.local.conf`, recordemos que para este sitio web no queremos que pueda ser accedido en http, según el enunciado se indica que trabajará en el puerto 443 exclusivamente, por lo que se realiza una redirección a protocolo seguro https, cosa que indicamos en nuestro VirtualHost *:80 junto con los certificados en el VirtualHost *:443:

```
<VirtualHost *:80>
    ServerName www.2asir.local
    Redirect permanent / https://www.2asir.local/
</VirtualHost>
<VirtualHost *:443>
    ServerName www.2asir.local
    DocumentRoot /var/www/html/2asir.local

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/autofirmado_certificado.crt
    SSLCertificateKeyFile /etc/ssl/private/autofirmado_privada.key

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Y así es como quedaría la configuración de **/etc/apache2/sites-available/1daw.local.conf**. Para este caso concreto se requiere una redirección permanente a /SistemasInformaticos como es requerido en el enunciado. También añadimos la siguiente línea para lograr esta redirección. Este sitio web debe funcionar en ambos protocolos http y https, por lo que añadimos la redirección en ambos.

```
RedirectMatch permanent ^/$ /SistemasInformaticos/
```

Con lo cual el archivo de configuración completo queda:

```
<VirtualHost *:80>
    ServerName www.1daw.local
    DocumentRoot /var/www/html/1daw.local
    RedirectMatch permanent ^/$ /SistemasInformaticos/
</VirtualHost>

<VirtualHost *:443>
    ServerName www.1daw.local
    DocumentRoot /var/www/html/1daw.local

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/autofirmado_certificado.crt
    SSLCertificateKeyFile /etc/ssl/private/autofirmado_privada.key
    RedirectMatch permanent ^/$ /SistemasInformaticos/

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Reiniciamos Apache como siempre para que los cambios surtan efecto con “systemctl restart apache2”.

- Finalizada la configuración de https para completar el apartado de seguridad, vemos que el enunciado también nos requiere una restricción para el sitio web www.2asir.local – se debe restringir el acceso a este sitio solo a la ip 10.0.0.35.

Para este cometido, vamos a extender la configuración del **archivo /etc/apache2/sites-available/2asir.local.conf** añadiendo un <Directory> de la siguiente forma:

```
<VirtualHost *:80>
    ServerName www.2asir.local
    Redirect permanent / https://www.2asir.local/
</VirtualHost>

<VirtualHost *:443>
    ServerName www.2asir.local
    DocumentRoot /var/www/html/2asir.local

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/autofirmado_certificado.crt
    SSLCertificateKeyFile /etc/ssl/private/autofirmado_privada.key

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    <Directory /var/www/html/2asir.local>
        Options Indexes FollowSymLinks
        AllowOverride All
        <RequireAll>
            Require all granted
            Require not ip 10.0.0.35
        </RequireAll>
    </Directory>
</VirtualHost>
```

Reiniciamos como siempre Apache ejecutando “systemctl restart apache2” para que los cambios surtan efecto.

d) Instalar módulos

- PHP
- Mysql

Vamos a instalar los módulos solicitados para PHP ejecutando el siguiente comando:

```
apt install php libapache2-mod-php php-mysql php-xml php-curl php-zip php-gd php-intl php-mbstring
```

El resultado de ejecutar el comando:


```
usuario@debianserversri: ~  
Configurando php8.2-mbstring (8.2.24-1~deb12u1) ...  
Creating config file /etc/php/8.2/mods-available/mbstring.ini with new version  
Configurando php-mbstring (2:8.2+93) ...  
Configurando php8.2-intl (8.2.24-1~deb12u1) ...  
Creating config file /etc/php/8.2/mods-available/intl.ini with new version  
Configurando php-intl (2:8.2+93) ...  
Configurando php8.2-curl (8.2.24-1~deb12u1) ...  
Creating config file /etc/php/8.2/mods-available/curl.ini with new version  
Configurando php8.2-xml (8.2.24-1~deb12u1) ...  
Creating config file /etc/php/8.2/mods-available/dom.ini with new version  
Creating config file /etc/php/8.2/mods-available/simplexml.ini with new version  
Creating config file /etc/php/8.2/mods-available/xml.ini with new version  
Creating config file /etc/php/8.2/mods-available/xmlreader.ini with new version  
Creating config file /etc/php/8.2/mods-available/xmlwriter.ini with new version  
Creating config file /etc/php/8.2/mods-available/xsl.ini with new version  
Configurando php8.2-gd (8.2.24-1~deb12u1) ...  
Creating config file /etc/php/8.2/mods-available/gd.ini with new version  
Configurando php-gd (2:8.2+93) ...  
Configurando php-xml (2:8.2+93) ...  
Configurando php-curl (2:8.2+93) ...  
Configurando php-mysql (2:8.2+93) ...  
Configurando php8.2-zip (8.2.24-1~deb12u1) ...  
Creating config file /etc/php/8.2/mods-available/zip.ini with new version  
Configurando php-zip (2:8.2+93) ...  
Procesando disparadores para libapache2-mod-php8.2 (8.2.24-1~deb12u1) ...  
Procesando disparadores para libc-bin (2.36-9+deb12u9) ...  
Procesando disparadores para php8.2-cli (8.2.24-1~deb12u1) ...  
root@debianserversri:/etc/apache2/sites-available#
```

Instalamos también MySQL como es requerido, tanto la parte cliente como la del servidor con el siguiente comando:

```
apt install mariadb-client mariadb-server
```

e) Montar un CMS como es Moodle

Vamos a instalar este popular CMS.

En primer lugar lo descargamos con el siguiente comando estando dentro de la carpeta /var/www/html:

```
wget https://download.moodle.org/download.php/direct/stable405/moodle-4.5.zip
```

A continuación, descomprimos el archivo con el siguiente comando:

```
unzip moodle-latest-405.zip
```

```
usuario@debianserversri: ~  
creating: moodle/blocks/settings/classes/  
creating: moodle/blocks/settings/classes/privacy/  
inflating: moodle/blocks/settings/classes/privacy/provider.php  
inflating: moodle/blocks/settings/styles.css  
inflating: moodle/blocks/settings/renderer.php  
creating: moodle/blocks/settings/db/  
inflating: moodle/blocks/settings/db/upgrade.php  
inflating: moodle/blocks/settings/db/access.php  
inflating: moodle/blocks/settings/block_settings.php  
inflating: moodle/blocks/settings/edit_form.php  
inflating: moodle/blocks/settings/version.php  
creating: moodle/blocks/settings/lang/  
creating: moodle/blocks/settings/lang/en/  
inflating: moodle/blocks/settings/lang/en/block_settings.php  
creating: moodle/blocks/settings/amd/  
creating: moodle/blocks/settings/amd/src/  
inflating: moodle/blocks/settings/amd/src/settingsblock.js  
creating: moodle/blocks/settings/amd/build/  
inflating: moodle/blocks/settings/amd/build/settingsblock.min.js.map  
inflating: moodle/blocks/settings/amd/build/settingsblock.min.js  
creating: moodle/blocks/calendar_upcoming/  
creating: moodle/blocks/calendar_upcoming/classes/  
creating: moodle/blocks/calendar_upcoming/classes/privacy/  
inflating: moodle/blocks/calendar_upcoming/classes/privacy/provider.php  
creating: moodle/blocks/calendar_upcoming/tests/  
creating: moodle/blocks/calendar_upcoming/tests/generator/  
inflating: moodle/blocks/calendar_upcoming/tests/generator/lib.php  
creating: moodle/blocks/calendar_upcoming/tests/behat/  
inflating: moodle/blocks/calendar_upcoming/tests/behat/block_calendar_upcoming_frontpage.feature  
inflating: moodle/blocks/calendar_upcoming/tests/behat/block_calendar_upcoming_course.feature  
inflating: moodle/blocks/calendar_upcoming/upgrade.txt  
creating: moodle/blocks/calendar_upcoming/db/  
inflating: moodle/blocks/calendar_upcoming/db/upgrade.php  
inflating: moodle/blocks/calendar_upcoming/db/access.php  
inflating: moodle/blocks/calendar_upcoming/version.php  
creating: moodle/blocks/calendar_upcoming/lang/  
creating: moodle/blocks/calendar_upcoming/lang/en/  
inflating: moodle/blocks/calendar_upcoming/lang/en/block_calendar_upcoming.php
```

Podemos borrar ya el zip descargado:

```
rm moodle-latest-405.zip
```

Modificamos los permisos necesarios para que Apache pueda acceder a los archivos ejecutando el comando siguiente:

```
chown -R www-data:www-data moodle
```

```
usuario@debianserversri: ~  
usuario@debianserversri: ~  
root@debianserversri:/var/www/html# ls -l  
total 28  
drwxr-xr-x  4 www-data www-data 4096 nov 17 12:55 lasir.local  
drwxr-xr-x  3 www-data www-data 4096 nov 17 13:02 ldaw.local  
drwxr-xr-x  2 www-data www-data 4096 nov 17 12:58 2asir.local  
-rw-r--r--  1 www-data www-data 10701 nov 11 14:00 index.html  
drwxr-xr-x 65 www-data www-data 4096 oct  5 12:43 moodle  
root@debianserversri:/var/www/html# chown -R www-data:www-data moodle  
root@debianserversri:/var/www/html#
```

Recordemos que el acceso al Moodle según el enunciado colgaría sobre el raíz del sitio web www.2asir.local. Este sitio tenía una restricción configurada en **/etc/apache2/sites-available/2asir.local.conf** en el que solo la ip 10.0.0.35 está habilitada para visitarlo.

Moodle también necesita una base de datos vacía para poder instalarse, así que vamos a crearla. Ejecutamos el comando “mysql -u root -p” y escribimos las siguientes sentencias dentro del cliente de mysql:

```
CREATE DATABASE moodledb CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
```

```
CREATE USER 'moodleuser'@'localhost' IDENTIFIED BY 'usuario';
```

```
GRANT ALL PRIVILEGES ON moodledb.* TO 'moodleuser'@'localhost';
```

```
FLUSH PRIVILEGES;
```

```
EXIT;
```

```
usuario@debianserversri: ~
root@debianserversri:/home/usuario# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 32
Server version: 10.11.6-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE moodledb CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
Query OK, 1 row affected (0.005 sec)

MariaDB [(none)]> CREATE USER 'moodleuser'@'localhost' IDENTIFIED BY 'usuario';
Query OK, 0 rows affected (0.005 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON moodledb.* TO 'moodleuser'@'localhost';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> FLUSH;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '' at line 1
MariaDB [(none)]>
MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> EXIT;
Bye
root@debianserversri:/home/usuario#
```

Visto esto, ya podemos comenzar con las pruebas.

4. Actividades / capturas a entregar

Para estas actividades vamos a usar nuestro cliente Windows ya configurado.

Se adjunta la configuración final de todos los sitios virtuales.

- **/etc/apache2/sites-available/1asir.local.conf**

```
usuario@debianserversri: ~
root@debianserversri:/etc/apache2/sites-available# cat 1asir.local.conf
<VirtualHost *:80>
    ServerName www.1asir.local
    DocumentRoot /var/www/html/1asir.local

    <Directory /var/www/html/1asir.local/FundamentosHardware>
        AuthType Basic
        AuthName "Autenticacion requerida"
        AuthUserFile /etc/apache2/.htpasswd
        Require valid-user
    </Directory>

    Alias /Redes /var/www/html/1asir.local/Redes

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

<VirtualHost *:443>
    ServerName www.1asir.local
    DocumentRoot /var/www/html/404/

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/autofirmado_certificado.crt
    SSLCertificateKeyFile /etc/ssl/private/autofirmado_privada.key

    <Directory /var/www/html/404>
        Options -Indexes +FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
root@debianserversri:/etc/apache2/sites-available#
```

- **/etc/apache2/sites-available/2asir.local.conf**

```
usuario@debianserversri: ~
root@debianserversri:/etc/apache2/sites-available# cat 2asir.local.conf
<VirtualHost *:80>
    ServerName www.2asir.local
    Redirect permanent / https://www.2asir.local/
</VirtualHost>

<VirtualHost *:443>
    ServerName www.2asir.local
    DocumentRoot /var/www/html/2asir.local/moodle

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/autofirmado_certificado.crt
    SSLCertificateKeyFile /etc/ssl/private/autofirmado_privada.key

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined

    <Directory /var/www/html/2asir.local/moodle>
        Options Indexes FollowSymLinks
        AllowOverride All
        <RequireAll>
            Require all granted
            Require not ip 10.0.0.35
        </RequireAll>
    </Directory>
</VirtualHost>
root@debianserversri:/etc/apache2/sites-available#
```

- **/etc/apache2/sites-available/1daw.local.conf**

```
usuario@debianserversri: ~
root@debianserversri:/etc/apache2/sites-available# cat 1daw.local.conf
<VirtualHost *:80>
    ServerName www.1daw.local
    DocumentRoot /var/www/html/1daw.local
    RedirectMatch permanent ^/$ /SistemasInformaticos/
    #Redirect permanent / http://www.1daw.local/SistemasInformaticos
</VirtualHost>

<VirtualHost *:443>
    ServerName www.1daw.local
    DocumentRoot /var/www/html/1daw.local

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/autofirmado_certificado.crt
    SSLCertificateKeyFile /etc/ssl/private/autofirmado_privada.key
    RedirectMatch permanent ^/$ /SistemasInformaticos/

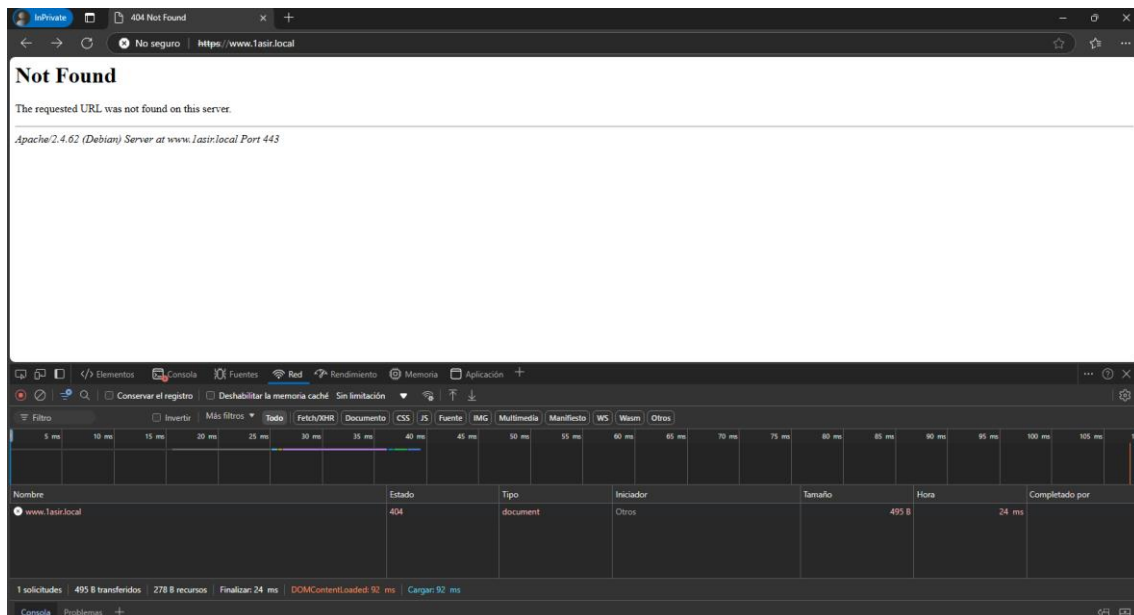
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
root@debianserversri:/etc/apache2/sites-available#
```

1) Captura accediendo desde un cliente a <http://www.1asir.local> así como a <https://www.1asir.local>.

Visitamos en primer lugar <http://www.1asir.local>:



Visitamos a continuación <https://www.1asir.local> (puerto 443). Podemos ver que nos devuelve un 404 porque lo estamos redirigiendo a un recurso inexistente para que no lo encuentre y el cliente pueda entender que esta URL no está habilitada.



Esta fue la parte de la configuración encargada de este comportamiento (el recurso `/var/www/html/404` no existe):

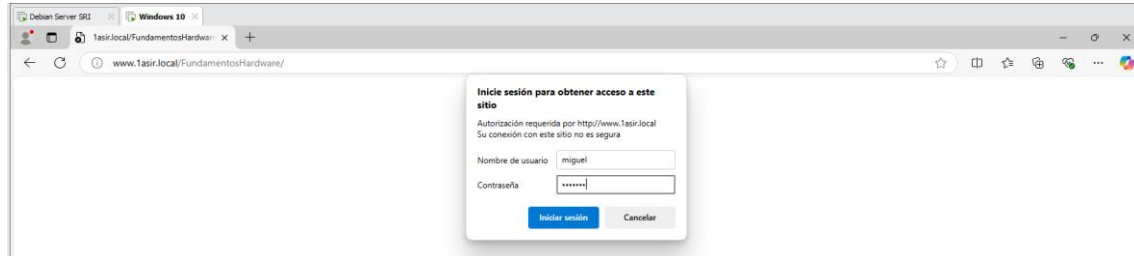
```
<VirtualHost *:443>
    ServerName www.1asir.local
    DocumentRoot /var/www/html/404/

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/autofirmado_certificado.crt
    SSLCertificateKeyFile /etc/ssl/private/autofirmado_privada.key

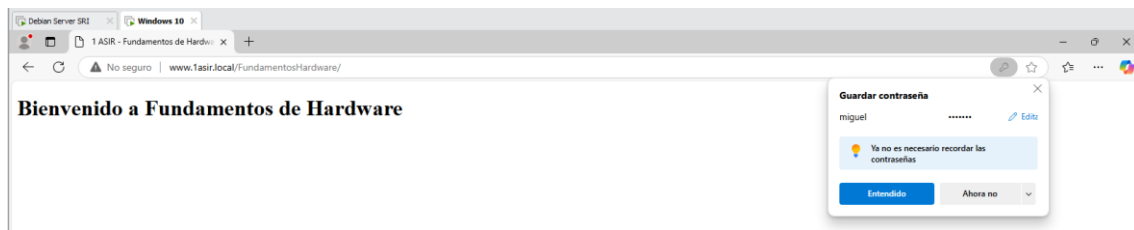
    <Directory /var/www/html/404>
        Options -Indexes +FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

2) Captura accediendo a la sección de 1asir llamada Fundamentos Hardware, donde se aprecie que se ha logueado el usuario llamado “miguel”.

Se dispara el diálogo de autenticación básica e introducimos “miguel” como nombre de usuario y como contraseña introducimos “usuario”.



Vemos que pasamos la autenticación y visualizamos correctamente a su página html.

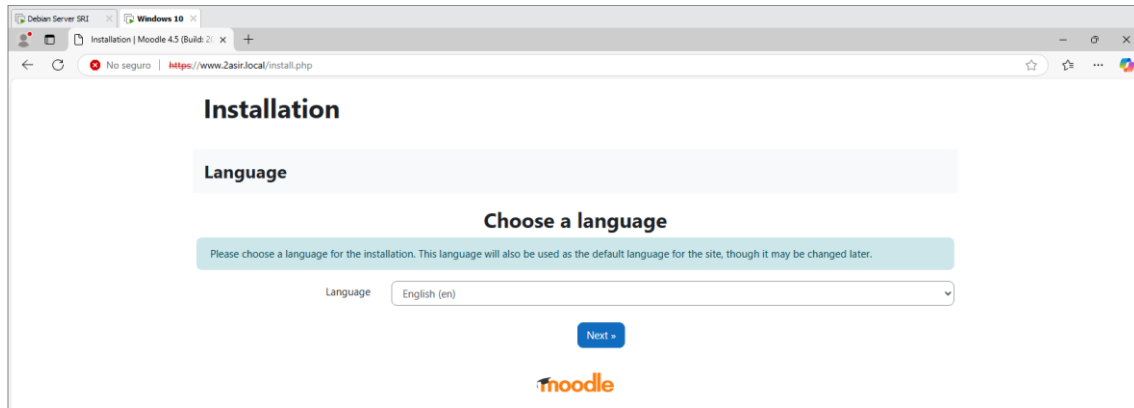


3) Captura accediendo a la sección de 1asir llamada Redes.

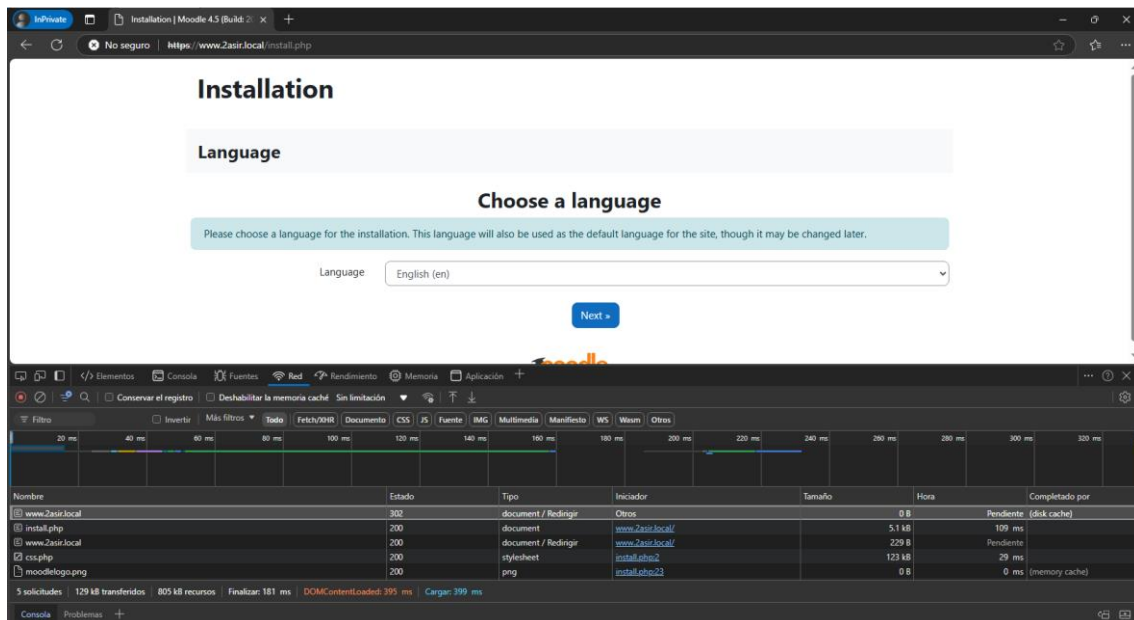


4) Captura accediendo desde un cliente a <http://www.2asir.local> así como a <https://www.2asir.local>. Explica como se ha solucionado la problemática asociada al acceso via http.

Comprobamos el acceso a <https://www.2asir.local> como solicita el enunciado. Podemos que ver que, tal como se requiere, se muestra la página inicial de Moodle, en este caso para su instalación al no estar aún instalado.

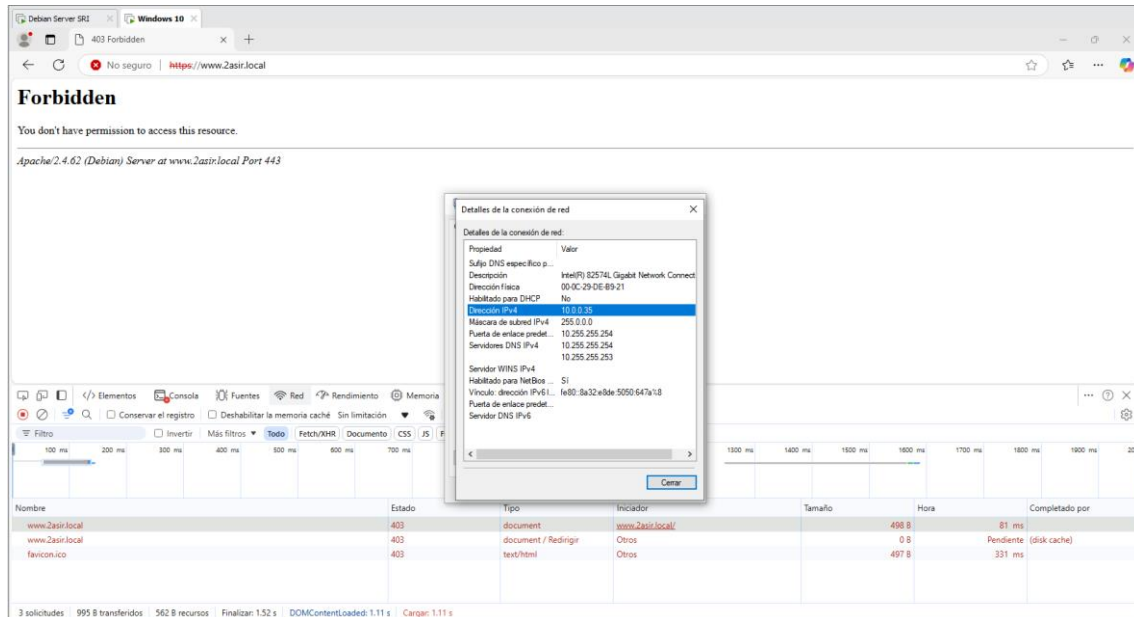


Con respecto a la **problemática asociada al acceso vía http** (puerto 80), si no configuramos el sitio virtual correctamente, nos abrirá la página por defecto de Apache. Lo que hicimos en la configuración es redirigir del puerto 80 al 443 de forma permanente. Podemos ver que esta redirección (302) funciona correctamente al abrir <http://www.2asir.local> y se nos redirecciona a la versión segura <https://www.2asir.local> que es lo esperado:



5) Captura accediendo al sitio de 2asir.local desde un cliente con ip 10.0.0.35.

Configuramos nuestro cliente Windows con la IP fija 10.0.0.35 y comprobamos como se nos deniega el acceso a 2asir.local.



6) Captura accediendo desde un cliente a <http://www.1daw.local> así como a <https://www.1daw.local>.

Ambos caminos nos llevarán a la página de SistemasInformaticos que es lo esperado pero tenemos que poder acceder tanto por http como por https como indica el enunciado.

En el caso de http:



Y en el caso de https:



7) Captura donde se muestre que se ha realizado la redirección hacia la sección de sistemas informáticos dentro del sitio 1daw.local.

Vemos en la consola de desarrollo como se lanza el 301 de redirección hacia /SistemasInformaticos.

