



Programación de sistemas distribuidos

Grado en Ingeniería
Informática

Sara Marcos Cornejo

Práctica 3: Servidor web



UNIVERSIDAD
NEBRIJA

Tabla de contenido

<i>I. Introducción.....</i>	<i>3</i>
<i>II. Objetivos.....</i>	<i>3</i>
<i>III. Elementos.....</i>	<i>3</i>
<i>IV. Funcionamiento.....</i>	<i>4</i>
<i>V. Preguntas, ejercicios.....</i>	<i>5</i>
A. Ejercicio 1.....	5
B. Ejercicio 2.....	6
C. Ejercicio 3.....	7
D. Ejercicio 4.....	9
E. Ejercicio 5.....	10
F. Ejercicio 6.....	11
G. Ejercicio 7.....	13
<i>V. Conclusiones.....</i>	<i>14</i>
<i>VI. Bibliografía.....</i>	<i>14</i>

I. Introducción

Un servidor web es un software que se ejecuta en un servidor y que tiene como función principal atender las solicitudes que llegan por parte de los clientes a través del protocolo HTTP. Su tarea es procesar y responder a las solicitudes de los navegadores o clientes que se conectan a él.

Apache es uno de los servidores web más utilizados en todo el mundo. Es un software de código abierto que funciona en la mayoría de los sistemas operativos, incluyendo Windows, Linux, Unix, entre otros. Al ser de código abierto, permite a los usuarios modificar el código fuente y personalizarlo según sus necesidades.

Apache es muy potente y escalable, lo que lo hace ideal para alojar sitios web de cualquier tamaño y complejidad. Además, es muy fácil de configurar y se pueden agregar diferentes módulos para extender su funcionalidad, como el soporte para SSL/TLS, autenticación, y muchos otros. Es capaz de procesar y servir una variedad de tipos de contenido, incluyendo HTML, CSS, JavaScript, archivos de imagen, audio y video, y otros tipos de archivos multimedia. También admite el procesamiento de solicitudes de diferentes métodos HTTP, como GET, POST, PUT y DELETE.

II. Objetivos

- Primera toma de contacto con esta tecnología.
- Estudiar las funcionalidades y ventajas que ofrece.

III. Elementos

Los elementos clave de Apache incluyen:

- **Servidor web:** Apache es un servidor web que se encarga de recibir las solicitudes HTTP de los clientes y enviar las respuestas correspondientes.
- **Módulos:** Apache es altamente modular, lo que significa que se pueden añadir y quitar módulos para personalizar su funcionamiento. Los módulos pueden ser de autenticación, seguridad, compresión, etc.
- **Virtual hosts:** Apache soporta virtual hosts, lo que permite a un solo servidor físico alojar múltiples sitios web. Cada virtual host tiene su propio nombre de dominio y configuración de sitio.
- **Archivos de configuración:** Apache utiliza archivos de configuración para definir cómo se comporta el servidor. Los archivos de configuración se encuentran en el directorio `/etc/apache2` en sistemas basados en Unix.
- **Acceso a archivos:** Apache proporciona un mecanismo para controlar el acceso a los archivos que sirve. Esto se puede hacer mediante reglas de acceso en los archivos de configuración o mediante autenticación de usuarios.
- **Logs:** Apache lleva un registro de todas las solicitudes que recibe y las respuestas que envía. Los registros se almacenan en archivos de registro que pueden ser analizados para obtener información sobre el tráfico del sitio web y posibles problemas.

IV. Funcionamiento

El cliente (por ejemplo, un navegador web) hace una solicitud HTTP al servidor (que ejecuta Apache) para acceder a un recurso específico (como una página web o una aplicación).

Apache recibe la solicitud y utiliza las reglas de configuración para determinar qué acción debe tomar. En el caso de una solicitud de un archivo, Apache busca el archivo solicitado en el sistema de archivos.

Si se encuentra el archivo, Apache lo entrega al cliente a través de la conexión HTTP. Si el archivo contiene código PHP, por ejemplo, se envía al intérprete de PHP para ser procesado y el resultado se envía al cliente.

Si el archivo no se encuentra en el sistema de archivos, Apache busca en las extensiones compiladas para ver si existe un módulo que pueda manejar la solicitud.

Si se encuentra una extensión compilada adecuada (por ejemplo, una extensión que maneja solicitudes para una base de datos SQL), Apache carga el módulo y lo utiliza para procesar la solicitud.

Si el servidor necesita acceder a una base de datos SQL para completar la solicitud, utiliza un controlador de base de datos apropiado para conectarse a la base de datos y ejecutar las consultas necesarias.

El resultado de la consulta se devuelve al servidor, que lo procesa y envía la respuesta final al cliente.

El cliente recibe la respuesta y la muestra al usuario.

Ejemplo funcionamiento:

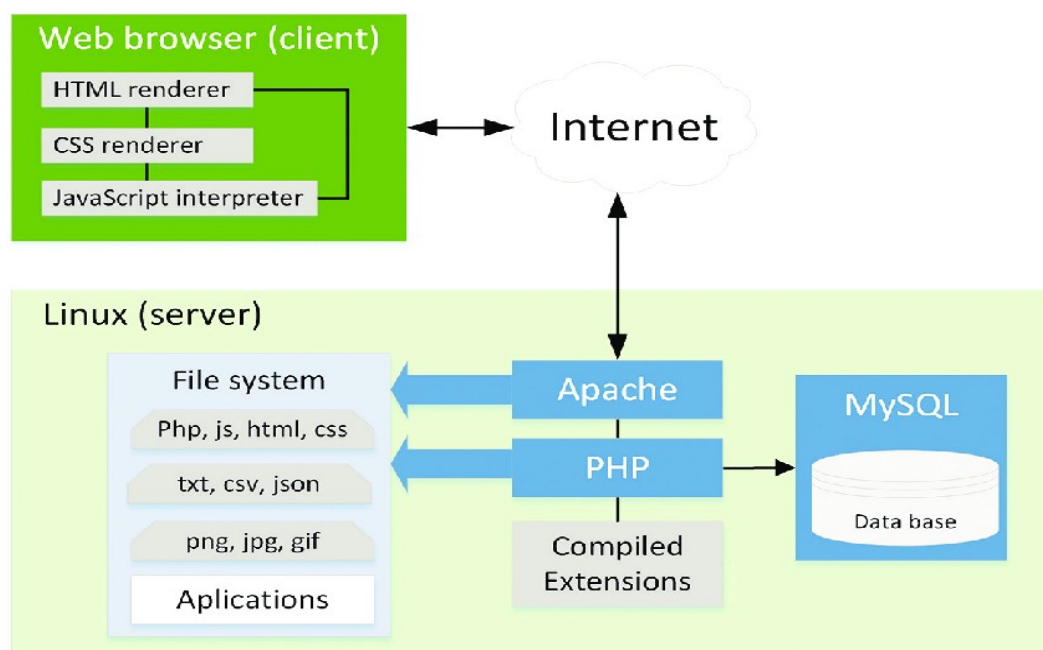


Figura 1. Estructura componentes Apache

V. Preguntas, ejercicios

1. Instalación de Apache 2.4

En primer lugar, se ejecuta el comando `sudo apt update` para actualizar la lista de paquetes disponibles en los repositorios de software.

```
sara@sara-vivobook:~$ sudo apt update
[sudo] contraseña para sara:
Obj:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Obj:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
Des:3 http://packages.microsoft.com/repos/code stable InRelease [3.023 B]
Des:4 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease [107 kB]
Obj:5 https://dl.google.com/linux/chrome/deb stable InRelease
Obj:6 https://deb.nodesource.com/node_14.x jammy InRelease
Obj:7 https://packages.microsoft.com/repos/vscode stable InRelease
Des:8 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Descargados 220 kB en 1s (316 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 203 paquetes. Ejecute «apt list --upgradable» para verlos.
```

Se procede con la instalación de apache con el comando: `sudo apt install apache2`

```
sara@sara-vivobook:~$ sudo apt install apache2
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.
  libjs-highlight.js
Utilice «sudo apt autoremove» para eliminarlo.
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap
0 actualizados, 8 nuevos se instalarán, 0 para eliminar y 203 no actualizados.
```

Para verificar que se instaló correctamente: `sudo apachectl -v` (muestra la versión de Apache)

```
sara@sara-vivobook:~$ sudo apachectl -v
Server version: Apache/2.4.52 (Ubuntu)
Server built:   2023-01-23T18:34:42
sara@sara-vivobook:~$
```

2. Por defecto se instalará en el directorio `/var/www`. ¿Qué puedes explicar de la estructura de directorios?

La estructura de directorios predeterminada de Apache en el directorio `/var/www` es la siguiente:

El directorio predeterminado donde se almacenan los archivos del sitio web en Apache es `/var/www/html`. Dentro de este directorio, se pueden crear subdirectorios y archivos para organizar el contenido del sitio web. (imágenes, archivos CSS, HTML,...)

El directorio `/var/www/icons` contiene iconos que se utilizan para mostrar en el directorio predeterminado de Apache.

Para almacenar archivos de inclusión de servidor que se utilizan en varios lugares del sitio web se utiliza el directorio `/var/www/includes`.

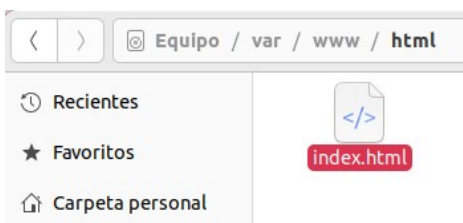
El directorio `/var/www/cgi-bin` se utiliza para alojar scripts CGI, que se almacenan en este directorio y se ejecutan desde allí.

Para almacenar archivos de registro de Apache se utiliza el directorio `/var/www/logs`, que registran información sobre las solicitudes de los clientes y otros eventos importantes del servidor.

`/var/www/error`: Este directorio contiene archivos HTML de error que se muestran cuando se produce un error en el servidor

Además, existen algunos directorios opcionales tales como: `/var/www/cgi-bin-ssl` (para alojar scripts CGI que se ejecutan en modo seguro (SSL)) o `/var/www/private` (para almacenar archivos privados relacionados con la configuración del servidor web).

Como se puede observar, se instala en `var/www` el `index.html` correspondiente:



Apache2 Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in [/usr/share/doc/apache2/README.Debian.gz](#)**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
/   |-- ports.conf
|-- mods-enabled
/   |-- *.load
/   |-- *.conf
|-- conf-enabled
/   |-- *.conf
|-- sites-enabled
/   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host

3. Crear 2 VirtualHosting

nombre_apellido_1.nebrija.es y nombre_apellido2.nebrija.es

Nota: En la memoria sólo figura la creación y configuración para uno de los Virtualhost. Se debe hacer lo mismo para el otro Virtualhost.

En primer lugar, se crean los dos directorios correspondientes mediante el comando:

```
sara@sara-vivobook:~$ sudo mkdir /var/www/sara_marcos_1.nebrija.es
```

A continuación, asigne la propiedad del directorio con la variable de entorno \$USER:

```
sara@sara-vivobook:~$ sudo chown -R $USER:$USER /var/www/sara_marcos_1.nebrija.es
```

Para asegurarse de que sus permisos sean correctos y permitir al propietario leer, escribir y ejecutar los archivos, y a la vez conceder solo permisos de lectura y ejecución a los grupos y terceros, puede ingresar el siguiente comando

```
sara@sara-vivobook:~$ sudo chmod -R 755 /var/www/sara_marcos_1.nebrija.es
```

A continuación, se crea una página de ejemplo index.html utilizando nano:

```
sara@sara-vivobook:~$ sudo nano /var/www/sara_marcos_1.nebrija.es/index.html
sara@sara-vivobook:~$
```

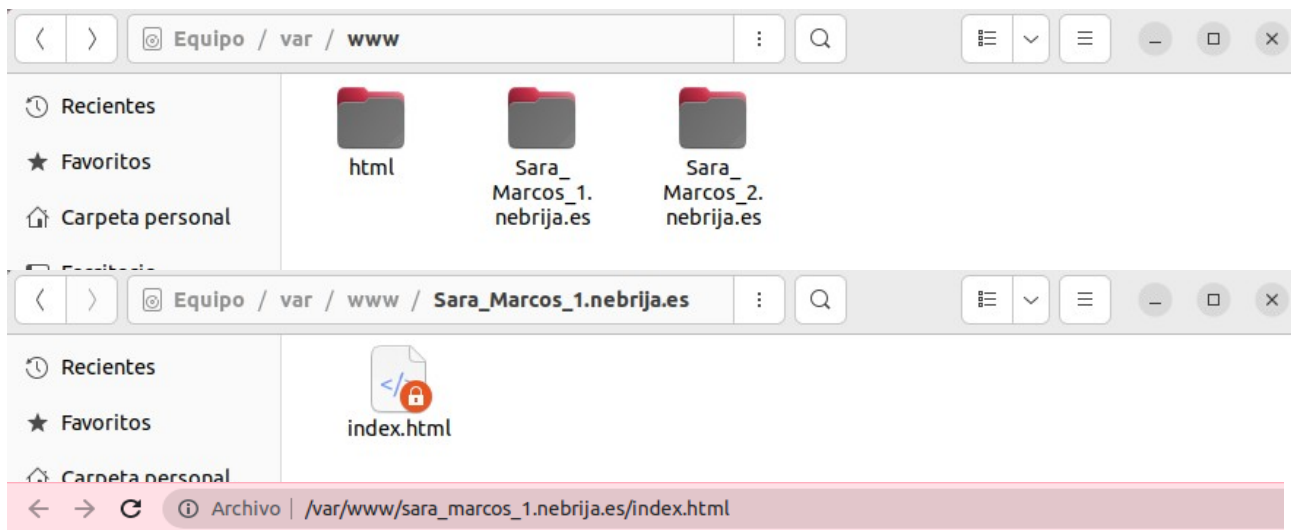
Este comando abrirá el archivo "index.html" en el editor de texto nano con permisos de superusuario, lo que permitirá editar el archivo si se desea.

Se añade al fichero html la descripción que se desee:

```
GNU nano 6.2 /var/www/sara_marcos_1.nebrija.es/index.html
<html>
<head>
  <title>Sitio web de Sara Marcos 1</title>
</head>
<body>
  <h1>Bienvenido al sitio web de Sara Marcos 1</h1>
  <p>Este es el sitio web de Sara Marcos 1.</p>
</body>
</html>
```

Guardar y cerrar el archivo al terminar.

El resultado obtenido es el siguiente:



Bienvenido al sitio web de Sara Marcos 1

Este es el sitio web de Sara Marcos 1.

← → ↻ Archivo | /var/www/sara_marcos_1.nebrija.es/index.html

Bienvenido al sitio web de Sara Marcos 2

Este es el sitio web de Sara Marcos 2.

Como se puede observar, se han creado correctamente los respectivos directorios, cada uno con su fichero index.html correspondiente, con el contenido que se ha añadido mediante HTML con el último comando.

El siguiente paso es crear un archivo de host virtual con las directivas correctas para que Apache proporcione este contenido. Se crea un archivo de configuración nuevo de VirtualHost en el directorio "/etc/apache2/sites-available/" para cada host, con el comando:

```
sara@sara-vivobook:~$ sudo nano /etc/apache2/sites-available/sara_marcos_1.nebrija.es.conf
```

Posteriormente se añade el siguiente bloque de código en el archivo de configuración de VirtualHost: (Guardar y cerrar el archivo al terminar).

```
sara@sara-vivobook: ~  
GNU nano 6.2 /etc/apache2/sites-available/Sara_Marcos_1.n  
<VirtualHost *:80>  
    ServerName Sara_Marcos_1.nebrija.es  
    ServerAlias www.Sara_Marcos_1.nebrija.es  
    DocumentRoot /var/www/Sara_Marcos_1.nebrija.es/public_html  
    ErrorLog ${APACHE_LOG_DIR}/error.log  
    CustomLog ${APACHE_LOG_DIR}/access.log combined  
</VirtualHost>
```


Este bloque de código define la configuración básica necesaria para que Apache pueda servir contenido para el dominio `sara_marcos_1.nebrija.es`, incluyendo la ubicación del directorio raíz de documentos, los archivos de registro de acceso y de errores y las direcciones IP y puertos en los que se escucharán las solicitudes.

A continuación, se habilitarán los VirtualHosts y se reiniciará el servidor Apache para que los cambios surtan efecto:

```
sara@sara-vivobook: ~  
sara@sara-vivobook:~$ sudo a2ensite Sara_Marcos_1.nebrija.es.conf  
[sudo] contraseña para sara:  
Enabling site Sara_Marcos_1.nebrija.es.  
To activate the new configuration, you need to run:  
    systemctl reload apache2  
sara@sara-vivobook:~$ sudo systemctl restart apache2  
sara@sara-vivobook:~$
```

Al buscar `http://sara_marcos_2.nebrija.es` el resultado obtenido es el siguiente:



4. ¿Qué configuración debes hacer en el servidor de nombres de tú portátil/ordenador?

Como se puede observar en el apartado 3, no se ha podido acceder al sitio web de http://sara_marcos_2.nebrija.es con la configuración realizada hasta el momento.

Para que funcione correctamente, se debe hacer una configuración de DNS.

Para realizar esto, se puede editar el archivo `/etc/hosts` en su computadora y agregar las siguientes líneas para asignar las direcciones IP de su servidor a los nombres de host:

- `<direccion_IP_del_servidor> sara_marcos_1.nebrija.es`
- `<direccion_IP_del_servidor> sara_marcos_2.nebrija.es`

Después de agregar estas líneas, guarde y cierre el archivo `/etc/hosts`.

```

sara@/etc/hosts: orden no encontrado
sara@sara-vivobook:~$ sudo nano /etc/hosts
GNU nano 6.2 /etc/hosts
127.0.0.1 localhost
127.0.1.1 sara-VivoBook-ASUSLaptop-X521EA-S533EA

192.168.1.135 sara_marcos_1.nebrija.es sara_marcos_2.nebrija.es
# 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

```

Una vez realizado esto, se intenta acceder al sitio web de nuevo. El resultado obtenido es el siguiente:

← → No es seguro | sara_marcos_1.nebrija.es < ☆ Actualizar ⓘ

Bienvenido al sitio web de Sara Marcos 1

Este es el sitio web de Sara Marcos 1.

← → No es seguro | sara_marcos_2.nebrija.es < ☆ Actualizar ⓘ

Bienvenido al sitio web de Sara Marcos 2

Este es el sitio web de Sara Marcos 2.

Funciona correctamente.

5. ¿Eres capaz de añadir autenticación básica a tú servidor? ¿Puedes añadir usuario y contraseña distintos para cada virtual host?

Sí, esto es posible. Se realiza de la siguiente forma:

En primer lugar se crea un archivo de contraseñas para el usuario en cuestión utilizando el comando htpasswd. Se debe introducir el siguiente comando, que pedirá una contraseña para el usuario: user1

```

sara@sara-vivobook:~$ sudo htpasswd -c /etc/apache2/.htpasswd user1
New password:
Re-type new password:
Adding password for user user1

```

Posteriormente se debe acceder de nuevo al fichero de configuración del VirtualHost que se desee configurar, y añadir el siguiente código. Guardar y cerrar al terminar.

```

GNU nano 6.2 /etc/apache2/sites-available
<VirtualHost *:80>
    ServerName sara_marcos_2.nebrija.es
    ServerAlias www.sara_marcos_2.nebrija.es
    DocumentRoot /var/www/sara_marcos_2.nebrija.es

    <Directory /var/www/sara_marcos_2.nebrija.es>
        Options Indexes FollowSymLinks
        AllowOverride All

        AuthType Basic
        AuthName "Acceso restringido"
        AuthUserFile /etc/apache2/.htpasswd
        Require user user1
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/sara_marcos_2_error.log
    CustomLog ${APACHE_LOG_DIR}/sara_marcos_2_access.log combined
</VirtualHost>

```

En este código se activa la autenticación básica, se establece el nombre que se mostrará en la ventana de autenticación, se indica la ruta del archivo de contraseñas creado en el paso anterior y se especifica que solo el usuario "user1" tendrá acceso.

A continuación, se habilitará el archivo de configuración del virtual host y se reiniciará el servidor Apache para que los cambios surtan efecto:

```
sara@sara-vivobook:~$ sudo a2ensite sara_marcos_2.nebrija.es
[sudo] contraseña para sara:
Site sara_marcos_2.nebrija.es already enabled
sara@sara-vivobook:~$ sudo systemctl reload apache2
sara@sara-vivobook:~$
```

El resultado obtenido es el siguiente:

Al realizar la petición de acceso a sara_marcos_2.nebrija.es, se solicitará un nombre de usuario y contraseña para poder acceder al sitio. Una vez introducidas las credenciales correctas configuradas previamente, se accede al sitio con normalidad de la siguiente forma:

The first screenshot shows a browser window with the address bar set to sara_marcos_2.nebrija.es. A modal dialog titled 'Iniciar sesión' is displayed, showing the URL http://sara_marcos_2.nebrija.es and a warning that the connection is not private. It contains input fields for 'Nombre de usuario' and 'Contraseña', and buttons for 'Cancelar' and 'Iniciar sesión'.

The second screenshot shows the same dialog, but the 'Nombre de usuario' field is now filled with 'user1'.

The third screenshot shows the browser window after a successful login. The address bar now includes a warning 'No es seguro' (Not secure). The page content displays 'Bienvenido al sitio web de Sara Marcos 2' and a message 'Este es el sitio web de Sara Marcos 2.'

Bienvenido al sitio web de Sara Marcos 2

Este es el sitio web de Sara Marcos 2.

6. Si realizas una petición GET a uno de los 2 virtual host. Detalla las cabeceras que recibes

de la solicitud.

Para obtener las cabeceras de respuesta del virtual host sara_marcos_2.nebrija.es, se puede ejecutar el siguiente comando, que mostrará las cabeceras de la respuesta, pero no el contenido del cuerpo de la respuesta:

```
sara@sara-vivobook:~$ curl -I http://sara_marcos_2.nebrija.es
HTTP/1.1 401 Unauthorized
Date: Sat, 25 Mar 2023 21:36:40 GMT
Server: Apache/2.4.52 (Ubuntu)
WWW-Authenticate: Basic realm="Acceso restringido"
Content-Type: text/html; charset=iso-8859-1
```

Para ver todo el contenido de la respuesta, se puede usar:

```
sara@sara-vivobook:~$ curl -i sara_marcos_2.nebrija.es
HTTP/1.1 401 Unauthorized
Date: Sat, 25 Mar 2023 21:42:25 GMT
Server: Apache/2.4.52 (Ubuntu)
WWW-Authenticate: Basic realm="Acceso restringido"
Content-Length: 471
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>401 Unauthorized</title>
</head><body>
<h1>Unauthorized</h1>
<p>This server could not verify that you
are authorized to access the document
requested. Either you supplied the wrong
credentials (e.g., bad password), or your
browser doesn't understand how to supply
the credentials required.</p>
<hr>
<address>Apache/2.4.52 (Ubuntu) Server at sara_marcos_2.nebrija.es Port 80</address>
</body></html>
```

La respuesta obtenida tras la petición GET al sitio web sara_marcos_2.nebrija.es muestra las siguientes cabeceras:

- **HTTP/1.1 401 Unauthorized:** Indica que la respuesta tiene un código de estado 401 que corresponde a una respuesta de "No autorizado".
- **Date:** Sat, 25 Mar 2023 21:42:25 GMT: Indica la fecha y hora de la respuesta.
- **Server:** Apache/2.4.52 (Ubuntu): Indica el servidor utilizado para dar la respuesta.
- **WWW-Authenticate:** Basic realm="Acceso restringido": Es una cabecera de autenticación básica que indica al navegador que se requiere una autenticación para acceder al recurso y se muestra un mensaje de autenticación básica en el navegador.
- **Content-Length:** 471: Indica el tamaño en bytes del cuerpo de la respuesta.
- **Content-Type:** text/html; charset=iso-8859-1: Indica el tipo de contenido del cuerpo de la respuesta, que es HTML con codificación de caracteres ISO-8859-1.

El cuerpo de la respuesta contiene un mensaje HTML que explica que el servidor no puede

verificar que el usuario está autorizado para acceder al recurso solicitado debido a que no se proporcionaron las credenciales correctas o el navegador no pudo entender cómo suministrar las credenciales requeridas.

Si se desea obtener la respuesta con las cabeceras de autorización, se puede utilizar el siguiente comando:

```
sara@sara-vivobook:~$ curl -i -u user1:user1pw http://sara_marcos_2.nebrija.es
HTTP/1.1 200 OK
Date: Sat, 25 Mar 2023 21:44:07 GMT
Server: Apache/2.4.52 (Ubuntu)
Last-Modified: Sat, 25 Mar 2023 15:17:21 GMT
ETag: "cb-5f7bb019ec8d1"
Accept-Ranges: bytes
Content-Length: 203
Vary: Accept-Encoding
Content-Type: text/html

<html>
  <head>
    <title>Sitio web de Sara Marcos 2</title>
  </head>
  <body>
    <h1>Bienvenido al sitio web de Sara Marcos 2</h1>
    <p>Este es el sitio web de Sara Marcos 2.</p>
  </body>
</html>
```

Las cabeceras obtenidas corresponden a una respuesta exitosa (HTTP/1.1 200 OK) a una solicitud GET al sitio web sara_marcos_2.nebrija.es. Las cabeceras incluyen información sobre el servidor (Server: Apache/2.4.52 (Ubuntu)), la última modificación del archivo (Last-Modified: Sat, 25 Mar 2023 15:17:21 GMT) y el tamaño del archivo en bytes (Content-Length: 203). También se incluyen algunas cabeceras de control de caché (ETag: "cb-5f7bb019ec8d1" y Accept-Ranges: bytes) y de codificación (Vary: Accept-Encoding). El cuerpo de la respuesta es un documento HTML que muestra el mensaje "Bienvenido al sitio web de Sara Marcos 2".

7. ¿Cómo ampliarías esta práctica?

Para ampliar esta práctica se podría configurar SSL/TLS en los virtual hosts para que las conexiones sean seguras. Los pasos a seguir para realizar esto serían los siguientes:

- Primero, se deberá generar un certificado SSL/TLS: Para esto se puede utilizar una autoridad de certificación (CA) o generar un certificado autofirmado.
- Después, configurar Apache para habilitar SSL/TLS: Se debe asegurarse de que el módulo SSL esté cargado en la configuración de Apache. Luego se debe modificar el archivo de configuración de Apache para incluir la información del certificado SSL/TLS.
- Una vez realizado esto, configurar el VirtualHost para habilitar SSL/TLS: Se debe agregar la configuración de SSL/TLS específica del VirtualHost al archivo de configuración de Apache.
- Por último, reiniciar el servicio de Apache: Para que los cambios surtan efecto, se debe reiniciar el servicio de Apache.

Una vez realizados estos pasos, el sitio web debería estar configurado para aceptar conexiones seguras mediante HTTPS.

V. Conclusiones:

En esta práctica se ha hecho un análisis muy completo sobre el funcionamiento de Apache 2.4. Después de haber profundizado sobre su instalación, la estructura de directorios por defecto, la creación de VirtualHosts, la configuración del servidor de nombres, la adición de autenticación básica y la obtención de cabeceras en una solicitud GET, se puede concluir que la comprensión y manejo de estos conceptos es fundamental para el desarrollo y mantenimiento de sitios web y servicios en línea.

Apache es una herramienta muy poderosa y versátil que permite a los desarrolladores y administradores de sistemas gestionar el tráfico y la seguridad de sus aplicaciones. La práctica se puede ampliar de muchas maneras, como por ejemplo añadiendo nuevos VirtualHosts, configurando SSL/TLS, implementando un balanceo de carga, entre otros. En definitiva, conocer y entender el funcionamiento de los servidores web es fundamental para cualquier profesional que se dedique al mundo de la tecnología.

VI. Bibliografía:

<https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-20-04-es>
https://linuxhint.com/configure_apache_virtualhost_ubuntu/
<https://francisconi.org/linux/comandos/nano>
<https://geekland.eu/estructura-de-directorios-en-linux/>
<https://www.websecurity.digicert.com/es/es/security-topics/what-is-ssl-tls-https>