

Documentación de Apache. Introducción.

Por: Enrique Gallén Roda

Resumen:

A lo largo de este documento podremos ver el contexto del mismo, además del entorno que se utiliza, en este caso Apache, del cual también podremos saber algunas alternativas. Además, en este proyecto se podrá aprender a utilizar Apache, al menos, los primeros pasos. Para empezar, aprenderemos cómo es la instalación de Apache en Ubuntu y luego podremos configurar el firewall y ajustar sus puertos. Posteriormente, crearemos un sitio web, configuraremos el VirtualHost y, por último, activaremos el VirtualHost para que podamos ver nuestro sitio web. Todo esto utilizando los comandos correspondientes en cada caso, que se pueden apreciar en las capturas realizadas. Por último, se muestran los resultados y las conclusiones del proyecto.

Palabras Clave:

Apache, Linux, Ubuntu, servidor, sitio, web, instalación, configuración, VirtualHost.

Índice:

Resumen:	2
Palabras Clave:	2
Índice:	2
Introducción:	3
Apache: Introducción	4
Paso 1: Instalar Apache:.....	4
Paso 2: Ajustar el firewall.....	5
Apache: Configuración	7
Paso 1: Creación del sitio web.....	7
Paso 2: Configuración del VirtualHost.....	8
Paso 3: Activación del VirtualHost.....	9
Resultado:	10
Conclusión:	10
Bibliografía:	11

Introducción:

El contexto de este proyecto es la clase de DAW, con el equipo del centro, utilizando el sistema operativo de Linux, Ubuntu.

Apache es un servidor web gratuito y de código abierto, desarrollado por la Apache Software Foundation. Sirve para alojar páginas web y es muy utilizado por su flexibilidad y fiabilidad. Comenzó en 1995 y fue fundamental en el desarrollo inicial de internet. A continuación algunas alternativas:

- Nginx: Un servidor web de alto rendimiento, conocido por su estabilidad, configuración sencilla y bajo consumo de recursos. Además, se utiliza como proxy inverso y balanceador de carga.
- Microsoft Internet Information Services (IIS): Es el servidor web de Microsoft integrado en sus sistemas operativos Windows, usado para alojar sitios web y servicios en la web, con una fuerte integración con el resto de productos de Microsoft.
- LiteSpeed Web Server: Un servidor web comercial con un rendimiento optimizado para mejorar la velocidad de los sitios web. Ofrece una compatibilidad casi completa con las configuraciones de Apache, lo que facilita la migración.
- Caddy: Un servidor web moderno que destaca por su facilidad de uso y configuración automática de HTTPS. Es conocido por su diseño minimalista y la automatización de tareas complejas.
- Lighttpd: Diseñado para ser ligero en uso de memoria y CPU, es una opción eficiente para servidores con recursos limitados. A menudo se utiliza para sitios web de alto tráfico que necesitan una distribución eficiente de la carga.
- Cherokee: Un servidor web ligero y de alto rendimiento, Cherokee es conocido por su panel de administración de usuario que facilita la configuración, lo que lo hace accesible para principiantes.

La finalidad de este proyecto es aprender a utilizar Apache o, al menos, conseguir interiorizar los conceptos principales. Empezamos con la instalación de Apache y la posterior configuración de un servidor web de Apache en Ubuntu.

Relación de las actividades realizadas:

Apache: Introducción.

Paso 1: Instalar Apache:

Para empezar actualizamos los paquetes locales con este comando:

```
enrique@usuario-OptiPlex-380:~$ sudo apt update
Obj:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Obj:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
Obj:3 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
Des:4 https://download.docker.com/linux/ubuntu jammy InRelease [48,9 kB]
Obj:5 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Des:6 https://deb.nodesource.com/node_12.x focal InRelease [4.583 B]
Err:4 https://download.docker.com/linux/ubuntu jammy InRelease
  Las firmas siguientes no se pudieron verificar porque su clave pública no está disponible: NO_PUBKEY 7
  EA0A9C3F273FCD8
Leyendo lista de paquetes... Hecho
W: Error de GPG: https://download.docker.com/linux/ubuntu jammy InRelease: Las firmas siguientes no se p
udieron verificar porque su clave pública no está disponible: NO_PUBKEY 7EA0A9C3F273FCD8
E: El repositorio «https://download.docker.com/linux/ubuntu jammy InRelease» no está firmado.
N: No se puede actualizar de un repositorio como este de forma segura y por tanto está deshabilitado por
omisión.
N: Vea la página de manual apt-secure(8) para los detalles sobre la creación de repositorios y la config
uración de usuarios.
```

Ahora instalamos apache con el paquete de *apache2*:

```
enrique@usuario-OptiPlex-380:~$ sudo apt install apache2
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data apache2-utils
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
Se actualizarán los siguientes paquetes:
  apache2 apache2-bin apache2-data apache2-utils
4 actualizados, 0 nuevos se instalarán, 0 para eliminar y 202 no actualizados.
Se necesita descargar 1.696 kB de archivos.
Se utilizarán 0 B de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2 amd64 2.4.52-1ubuntu4.6 [97,8
kB]
Des:2 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-bin amd64 2.4.52-1ubuntu4.6 [
1.345 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-data all 2.4.52-1ubuntu4.6 [1
65 kB]
Des:4 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-utils amd64 2.4.52-1ubuntu4.6
[89,1 kB]
Descargados 1.696 kB en 1s (2.893 kB/s)
(Leyendo la base de datos ... 355868 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../apache2_2.4.52-1ubuntu4.6_amd64.deb ...
Desempaquetando apache2 (2.4.52-1ubuntu4.6) sobre (2.4.52-1ubuntu4.4) ...
Preparando para desempaquetar .../apache2-bin_2.4.52-1ubuntu4.6_amd64.deb ...
Desempaquetando apache2-bin (2.4.52-1ubuntu4.6) sobre (2.4.52-1ubuntu4.4) ...
Preparando para desempaquetar .../apache2-data_2.4.52-1ubuntu4.6_all.deb ...
Desempaquetando apache2-data (2.4.52-1ubuntu4.6) sobre (2.4.52-1ubuntu4.4) ...
Preparando para desempaquetar .../apache2-utils_2.4.52-1ubuntu4.6_amd64.deb ...
Desempaquetando apache2-utils (2.4.52-1ubuntu4.6) sobre (2.4.52-1ubuntu4.4) ...
Configurando apache2-bin (2.4.52-1ubuntu4.6) ...
Configurando apache2-data (2.4.52-1ubuntu4.6) ...
Configurando apache2-utils (2.4.52-1ubuntu4.6) ...
Configurando apache2 (2.4.52-1ubuntu4.6) ...
Could not execute systemctl: at /usr/bin/deb-systemd-invoke line 142.
apache-htcacheclean.service is a disabled or a static unit not running, not starting it.
Procesando disparadores para man-db (2.10.2-1) ...
Procesando disparadores para ufw (0.36.1-4build1) ...
Se han actualizado las reglas para el perfil «Apache»
El cortafuegos se ha recargado
```

Paso 2: Ajustar el firewall

Ajustamos la configuración del firewall para permitir el acceso externo a los puertos web por defecto.

Se enumeran los perfiles de aplicación *ufw* con este comando:

```
enrique@usuario-OptiPlex-380:~$ sudo ufw app list
Aplicaciones disponibles:
  Apache
  Apache Full
  Apache Secure
  CUPS
  OpenSSH
```

En este punto recomienda habilitar el perfil más restrictivo:

```
enrique@usuario-OptiPlex-380:~$ sudo ufw allow 'Apache'
Omitiendo adición de regla ya existente
Omitiendo adición de regla ya existente (v6)
```

Para verificar los cambios, se puede utilizar el siguiente comando:

```
enrique@usuario-OptiPlex-380:~$ sudo ufw status
Estado: activo

Hasta          Acción      Desde
-----
80/tcp          ALLOW       Anywhere
Apache          ALLOW       Anywhere
8080/tcp        ALLOW       Anywhere
80/tcp (v6)     ALLOW       Anywhere (v6)
Apache (v6)     ALLOW       Anywhere (v6)
8080/tcp (v6)   ALLOW       Anywhere (v6)
```

El resultado proporciona una lista del tráfico de HTTP que permite, ya con el puerto 80 activado, pero hubo un problema con la configuración de comandos:

```
enrique@usuario-OptiPlex-380:~$ sudo systemctl status apache2
* apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: failed (Result: exit-code) since Mon 2023-10-02 17:28:40 CEST; 2min 18s ago
     Docs: https://httpd.apache.org/docs/2.4/
    CPU: 22ms

oct 02 17:28:40 usuario-OptiPlex-380 systemd[1]: Starting The Apache HTTP Server...
oct 02 17:28:40 usuario-OptiPlex-380 apachectl[41265]: apache2: Could not open configuration file /etc/ab
oct 02 17:28:40 usuario-OptiPlex-380 apachectl[41262]: Action 'start' failed.
oct 02 17:28:40 usuario-OptiPlex-380 apachectl[41262]: The Apache error log may have more information.
oct 02 17:28:40 usuario-OptiPlex-380 systemd[1]: apache2.service: Control process exited, code=exited,
oct 02 17:28:40 usuario-OptiPlex-380 systemd[1]: apache2.service: Failed with result 'exit-code'.
oct 02 17:28:40 usuario-OptiPlex-380 systemd[1]: Failed to start The Apache HTTP Server.
lines 1-13/13 (END)
```

Al final se pudo permitir el acceso al servidor web de Apache y finalizar la instalación con la comprobación del servidor web gracias a esta lista de comandos:

```

enrique@usuario-OptiPlex-380:~$ sudo service apache2 stop
enrique@usuario-OptiPlex-380:~$ sudo apt-get purge apache2 apache2-utils apache2.2-bin apache2-common
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
El paquete «apache2.2-bin» no está instalado, no se eliminará
E: No se ha podido localizar el paquete apache2-common
enrique@usuario-OptiPlex-380:~$ sudo apt-get autoremove
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 206 no actualizados.
enrique@usuario-OptiPlex-380:~$ whereis apache2
apache2: /usr/sbin/apache2 /usr/lib/apache2 /etc/apache2 /usr/share/apache2 /usr/share/man/man8/apache2.
8.gz
enrique@usuario-OptiPlex-380:~$ apache2: /etc/apache2
Orden «apache2:» no encontrada. Quizá quiso decir:
  la orden «apache2» del paquete deb «apache2-bin (2.4.52-1ubuntu4.6)»
Pruebe con: sudo apt install <nombre del paquete deb>
enrique@usuario-OptiPlex-380:~$ apache2:/etc/apache2
bash: apache2:/etc/apache2: No existe el archivo o el directorio
enrique@usuario-OptiPlex-380:~$ sudo rm -rf/etc/apache2
rm: opción incorrecta -- «/»
Pruebe 'rm --help' para más información.
enrique@usuario-OptiPlex-380:~$ whereis apache2
apache2: /usr/sbin/apache2 /usr/lib/apache2 /etc/apache2 /usr/share/apache2 /usr/share/man/man8/apache2.
8.gz
enrique@usuario-OptiPlex-380:~$ sudo rm -rf /usr/sbin/apache2 /usr/lib/apache2 /etc/apache2 /usr/share/a
pache2 /usr/share/man/man8/apache2.8.gz

```

Apache: Configuración.

Paso 1: Creación del sitio web.

Para la configuración de un virtual host debemos empezar creando un sitio web, para ello creamos la carpeta en la que almacenará, esto lo hacemos ejecutando el siguiente comando:

```
enrique@usuario-OptiPlex-380: /var/www/gci
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
enrique@usuario-OptiPlex-380:~$ sudo mkdir /var/www/gci/
[sudo] contraseña para enrique:
```

Ahora hay que crear un archivo HTML en el directorio que hemos creado, se utilizan estos dos comandos:

```
enrique@usuario-OptiPlex-380:~$ cd /var/www/gci/
enrique@usuario-OptiPlex-380:/var/www/gci$ nano index.html
```

Abrimos el archivo index.html con el editor de texto Nano:

```
enrique@usuario-OptiPlex-380:/var/www/gci$ sudo nano index.html
```

Y ahora introducimos el siguiente código en index.html:

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
GNU nano 6.2                                index.html *
<html>
<head>
  <title> Ubuntu rocks! </title>
</head>
<body>
  <p> I'm running this website on an Ubuntu Server server!
</body>
</html>

^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich. ^\ Reemplazar  ^U Pegar      ^J Justificar ^_ Ir a línea
```

Paso 2: Configuración del VirtualHost.

Empezamos accediendo al directorio de Apache y editando el archivo de configuración con los siguientes comandos:

```
enrique@usuario-OptiPlex-380: /etc/apache2/sites-available
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
enrique@usuario-OptiPlex-380:/etc/apache2/sites-available$ sudo cp 000-default.conf gci.conf
[sudo] contraseña para enrique:
enrique@usuario-OptiPlex-380:/etc/apache2/sites-available$ sudo nano gci.conf
enrique@usuario-OptiPlex-380:/etc/apache2/sites-available$
```

Una vez dentro, lo editamos y añadimos las líneas 11 y 12. También debemos quitar el # de la línea 9 (aunque en la captura aparezca, en el archivo original se quitó):

```
*gci.conf
/etc/apache2/sites-available
Guardar

1 <VirtualHost *:80>
2     # The ServerName directive sets the request scheme, hostname and port that
3     # the server uses to identify itself. This is used when creating
4     # redirection URLs. In the context of virtual hosts, the ServerName
5     # specifies what hostname must appear in the request's Host: header to
6     # match this virtual host. For the default virtual host (this file) this
7     # value is not decisive as it is used as a last resort host regardless.
8     # However, you must set it for any further virtual host explicitly.
9     #ServerName gci.example.com
10
11     ServerAdmin enrique@localhost.com
12     DocumentRoot /var/www/gci/|
13
14     # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
15     # error, crit, alert, emerg.
16     # It is also possible to configure the loglevel for particular
17     # modules, e.g.
18     #LogLevel info ssl:warn
19
20     ErrorLog ${APACHE_LOG_DIR}/error.log
21     CustomLog ${APACHE_LOG_DIR}/access.log combined
22
23     # For most configuration files from conf-available/, which are
24     # enabled or disabled at a global level, it is possible to
25     # include a line for only one particular virtual host. For example the
26     # following line enables the CGI configuration for this host only
27     # after it has been globally disabled with "a2disconf".
28     #Include conf-available/serve-cgi-bin.conf
29 </VirtualHost>
30
31 # vim: syntax=apache ts=4 sw=4 sts=4 sr noet

Texto plano  Anchura del tabulador: 4  Ln 12, Col 31  INS
```


Paso 3: Activación del VirtualHost.

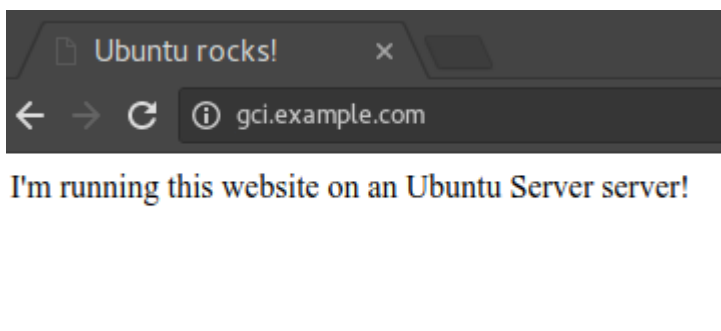
Después de configurar el VirtualHost debemos ejecutar el archivo con el siguiente comando:

```
enrique@usuario-OptiPlex-380:/etc/apache2/sites-available$ sudo a2ensite gci.conf
Enabling site gci.
To activate the new configuration, you need to run:
    systemctl reload apache2
```

Reiniciamos Apache, y así poder visualizarlo, con el siguiente comando:

```
enrique@usuario-OptiPlex-380:/etc/apache2/sites-available$ systemctl reload apache2
enrique@usuario-OptiPlex-380:/etc/apache2/sites-available$
```

El resultado:



Resultados y conclusiones.

Resultado:

En el caso específico del proyecto sobre Apache, se inicia con una introducción que contextualiza el uso de Apache en la clase de Desarrollo de Aplicaciones Web (DAW) con el sistema operativo Linux, Ubuntu. Se presentan diversas alternativas a Apache, como Nginx, Microsoft IIS, LiteSpeed, Caddy, Lighttpd y Cherokee, destacando sus características.

El propósito principal del proyecto es aprender a utilizar Apache, comenzando con la instalación y configuración en Ubuntu. Se detallan las actividades realizadas, como la instalación de Apache y la configuración del firewall. Aunque se encuentran dificultades con algunos comandos, al final se logra permitir el acceso al servidor web de Apache. Luego, se describe la configuración del sitio web, la creación de un virtual host y la activación del mismo. Se proporcionan comandos específicos para llevar a cabo estas acciones. El texto culmina con la visualización del resultado obtenido tras reiniciar Apache.

Conclusión:

Desde mi punto de vista, el trabajo es bastante completo, debido a la aportación de la información por parte del profesor como su ayuda en clase. Personalmente me ha costado bastante, ya que este tipo de contenido no es el que mejor se me da, aunque he de admitir que es bastante útil. Además, mi trabajo en clase ha sido escaso debido a mi frustración personal, todo esto ha propiciado que, en pocas palabras, *“me pille el toro”*. A pesar del poco tiempo disponible, he podido acabar el proyecto que junto a las capturas realizadas en clase y a la información de las web enlazadas por el profesor, he podido remontar. En definitiva, un proyecto completo, que te enseña bien los pocos contenidos que tiene en base a Bash, aunque para llevarlo correctamente hay que trabajarlo día a día.

Bibliografía:

- [Cómo instalar el servidor web Apache en Ubuntu 20.04 \(digitalocean.com\)](https://www.digitalocean.com/articles/how-to-install-apache-on-ubuntu-20-04)
- [Instalación y configuración de Apache \(ubuntu.com\)](https://ubuntu.com/server/docs/apache-installation)
- [Chat GPT](#) (utilizado para complementar información)