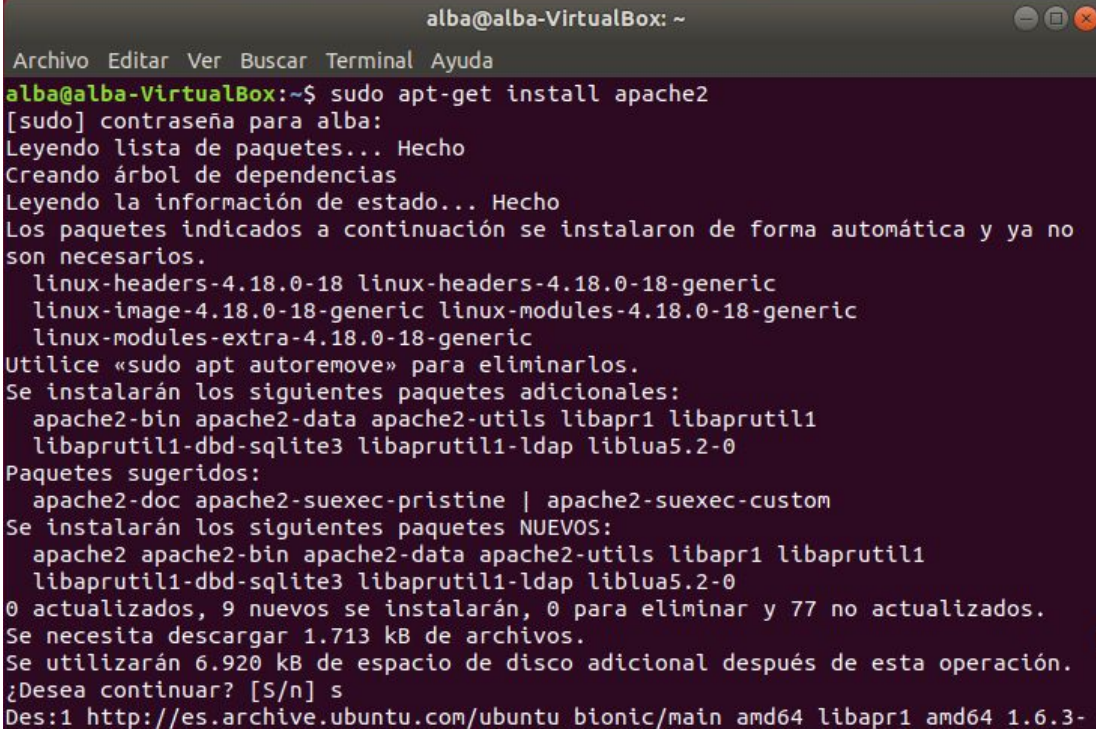


MANUAL INSTALACIÓN Y CONFIGURACIÓN SERVIDOR WEB (HTTP)

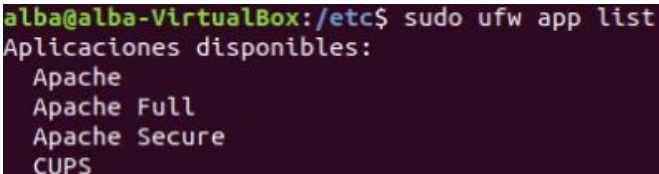
1.- Antes de empezar con la creación del servidor, ejecutamos el comando **sudo apt-get update** para actualizar el índice de los paquetes locales. Esto es para garantizar que en él se reflejan las cargas más recientes de las nuevas versiones de los paquetes.

2.- Para crear y configurar un servidor web, necesitamos instalar el paquete Apache2, mediante el comando **sudo apt-get install apache2**:



```
alba@alba-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Ayuda
alba@alba-VirtualBox:~$ sudo apt-get install apache2
[sudo] contraseña para alba:
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.
  linux-headers-4.18.0-18 linux-headers-4.18.0-18-generic
  linux-image-4.18.0-18-generic linux-modules-4.18.0-18-generic
  linux-modules-extra-4.18.0-18-generic
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
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 liblua5.2-0
0 actualizados, 9 nuevos se instalarán, 0 para eliminar y 77 no actualizados.
Se necesita descargar 1.713 kB de archivos.
Se utilizarán 6.920 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://es.archive.ubuntu.com/ubuntu bionic/main amd64 libapr1 amd64 1.6.3-
```

3.- Antes de probar el servidor, es necesario modificar los ajustes del cortafuegos de tal manera que se garantice el acceso externo a los puertos web por defecto. Todo ello teniendo en cuenta que tenemos un cortafuegos UFW configurado para restringir el acceso al servidor. Durante la instalación, Apache en sí, se registra en el UFW para proveer los perfiles que permitan habilitar o deshabilitar su acceso a través del cortafuegos. Para ver la lista de los perfiles de aplicación introducimos el comando **sudo ufw app list**. Se debería desplegar una aplicaciones disponibles:



```
alba@alba-VirtualBox:/etc$ sudo ufw app list
Aplicaciones disponibles:
  Apache
  Apache Full
  Apache Secure
  CUPS
```

4.- Añadimos Apache a la lista de acceso del cortafuegos:

```
alba@alba-VirtualBox:/etc$ sudo ufw allow 'Apache'  
Regla añadida  
Regla añadida (v6)
```

5.- Se puede verificar el cambio ejecutando el comando **sudo ufw status**. Debería desplegarse la siguiente lista:

```
alba@alba-VirtualBox:/etc$ sudo ufw status  
Estado: activo  
  
Hasta          Acción         Desde  
-----          -  
20/tcp         ALLOW          Anywhere  
21/tcp         ALLOW          Anywhere  
990/tcp        ALLOW          Anywhere  
40000:50000/tcp ALLOW          Anywhere  
Apache         ALLOW          Anywhere  
20/tcp (v6)   ALLOW          Anywhere (v6)  
21/tcp (v6)   ALLOW          Anywhere (v6)  
990/tcp (v6)  ALLOW          Anywhere (v6)  
40000:50000/tcp (v6) ALLOW          Anywhere (v6)  
Apache (v6)   ALLOW          Anywhere (v6)
```

Como se puede observar, la aplicación se ha activado, y está permitido el acceso al servidor web.

6.- Al finalizar el proceso de instalación, Ubuntu 18.04 iniciará Apache. Para entonces el servidor web debería encontrarse activo y en ejecución. Nos aseguramos de que el servidor está activo y en ejecución escribiendo el comando **sudo systemctl status apache2**:

```
alba@alba-VirtualBox:~$ sudo systemctl status apache2  
● apache2.service - The Apache HTTP Server  
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:   
   Drop-In: /lib/systemd/system/apache2.service.d  
            └─apache2-systemd.conf  
   Active: active (running) since Thu 2019-06-06 08:52:07 CEST; 1min 40s ago  
   Main PID: 12274 (apache2)  
     Tasks: 55 (limit: 4915)  
    CGroup: /system.slice/apache2.service  
            └─12274 /usr/sbin/apache2 -k start  
              └─12275 /usr/sbin/apache2 -k start  
                └─12276 /usr/sbin/apache2 -k start  
  
jun 06 08:52:07 alba-VirtualBox systemd[1]: Starting The Apache HTTP Server...  
jun 06 08:52:07 alba-VirtualBox apachectl[12263]: AH00558: apache2: Could not re  
jun 06 08:52:07 alba-VirtualBox systemd[1]: Started The Apache HTTP Server.
```

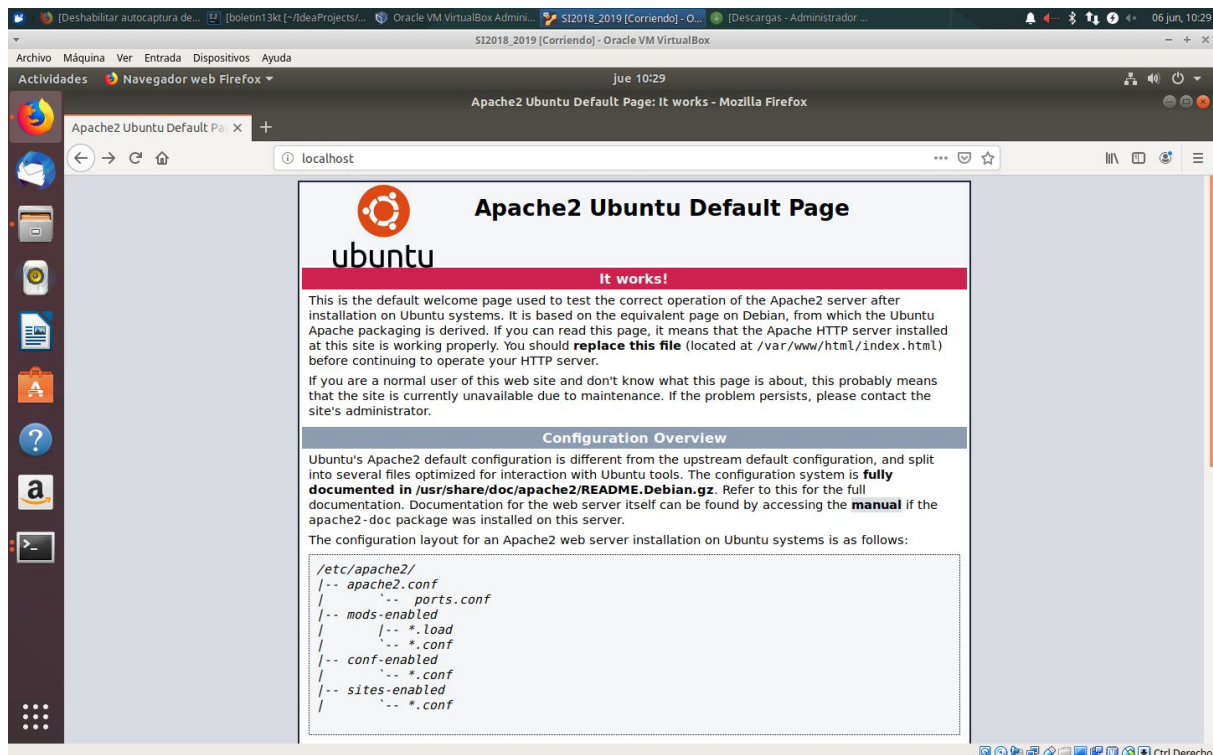
Como se puede apreciar, el servidor se ha iniciado con éxito. Sin embargo, la mejor forma de realizar esta comprobación es pedirle al servidor Apache que muestre una página.

7.- Podemos acceder a la página por defecto de Apache para confirmar que éste se está ejecutando correctamente a través de nuestra dirección IP. Si no sabemos la dirección IP de

nuestro servidor, podemos obtenerla de diferentes maneras desde la línea de comandos introduciendo **hostname -I**:

```
alba@alba-VirtualBox:~$ hostname -I  
10.0.2.15
```

8.- Cuando tengamos la dirección IP de nuestro servidor, la introducimos en la barra de direcciones del navegador. A continuación, deberíamos ver la página web predeterminada de Ubuntu 18.04:



La ejecución de esta página implica que Apache se encuentra funcionando correctamente. Asimismo, incluye información básica sobre la localización de archivos y directorios relevantes de Apache.

9.- Al utilizar el servidor web Apache, podemos usar sitios virtuales (similares a los bloques de servidor en Nginx), permitiendo encapsular detalles de configuración, así como alojar más de un dominio en un solo servidor. Configuraremos un dominio llamado ejemplo.com, pero se puede utilizar el dominio que se desee. Apache en Ubuntu 18.04 tiene un bloque de servidor predeterminado y activo para servir los documentos del directorio /var/www/html. A pesar de que funciona adecuadamente como configuración para un único sitio, puede ser muy difícil de mantener y controlar cuando se tienen múltiples sitios. En vez de modificar /var/www/html, creamos una estructura de directorios dentro de /var/www para nuestra página ejemplo.com, dejándola así: /var/www/html como se encuentra por defecto, y

sirviendo de directorio predeterminado en caso que la solicitud de un cliente no coincida con las otras páginas. Creamos el directorio para ejemplo.com usando la opción -p de tal manera que se creen los directorios padres necesarios:

```
alba@alba-VirtualBox:~$ sudo mkdir -p /var/www/ejemplo.com/html  
[sudo] contraseña para alba:
```

10.- A continuación, asignamos el usuario propietario del directorio, mediante la variable de entorno \$USER:

```
alba@alba-VirtualBox:~$ sudo chown -R $USER:$USER /var/www/ejemplo.com/html  
alba@alba-VirtualBox:~$
```

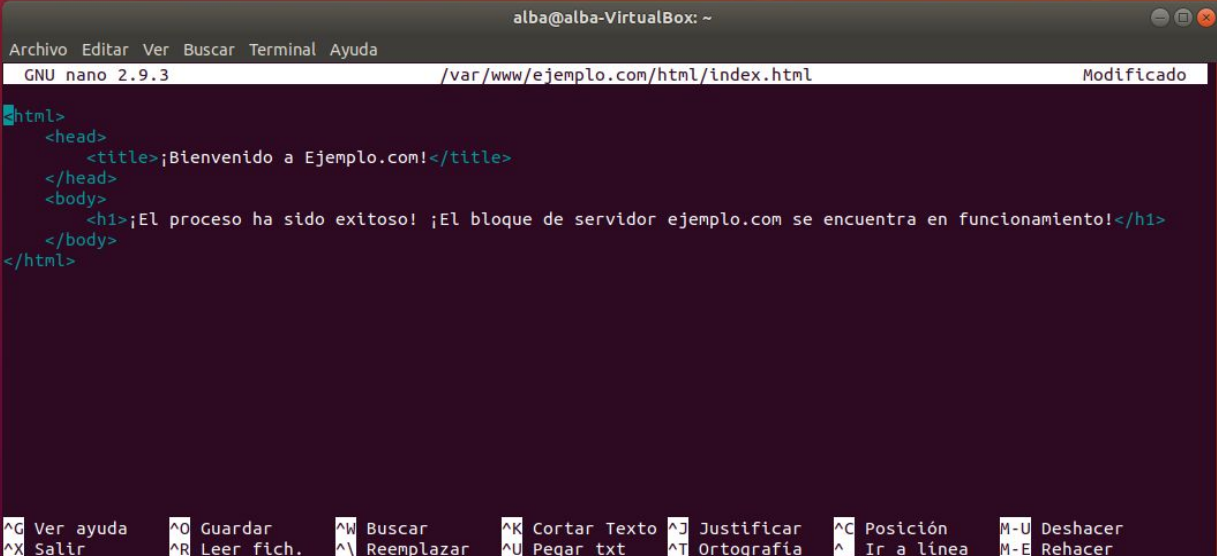
11.- Los permisos de los directorios raíz para la web no se modificarán a no ser que modifiquemos el valor de unmask. Sin embargo, podemos asegurarnos mediante el siguiente comando:

```
alba@alba-VirtualBox:~$ sudo chmod -R 755 /var/www/ejemplo.com  
alba@alba-VirtualBox:~$
```

12.- Después creamos una página de ejemplo index.html usando nano o el editor que prefiramos:

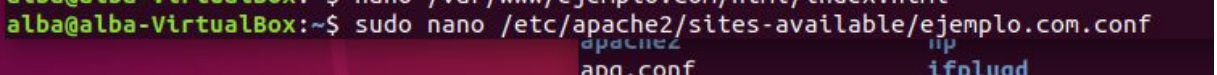
```
alba@alba-VirtualBox:~$ nano /var/www/ejemplo.com/html/index.html  
alba@alba-VirtualBox:~$
```


13.- Dentro del archivo, añadimos el código en HTML. Guardamos y cerramos el archivo cuando terminemos de editarlo.:



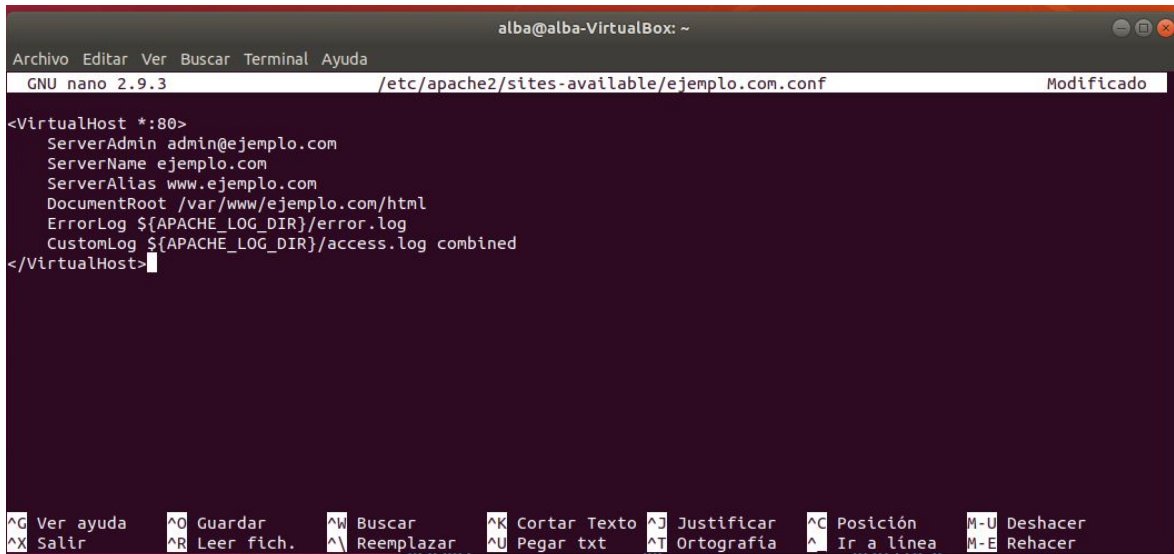
```
alba@alba-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 /var/www/ejemplo.com/html/index.html Modificado
html>
<head>
  <title>;Bienvenido a Ejemplo.com!</title>
</head>
<body>
  <h1>;El proceso ha sido exitoso! ;El bloque de servidor ejemplo.com se encuentra en funcionamiento!</h1>
</body>
</html>
^G Ver ayuda  ^O Guardar    ^W Buscar    ^K Cortar Texto ^J Justificar  ^C Posición   M-U Deshacer
^X Salir      ^R Leer fich. ^\ Reemplazar ^U Pegar txt  ^T Ortografía ^ Ir a línea  M-E Rehacer
```

14.- Para que Apache muestre este contenido, es necesario crear un archivo de alojamiento virtual. En vez de modificar directamente la configuración predeterminada que se encuentra en `/etc/apache2/sites-available/000-default.conf`, creamos una nueva en `/etc/apache2/sites-available/ejemplo.com.conf`:



```
alba@alba-VirtualBox: ~$ sudo nano /etc/apache2/sites-available/ejemplo.com.conf
```

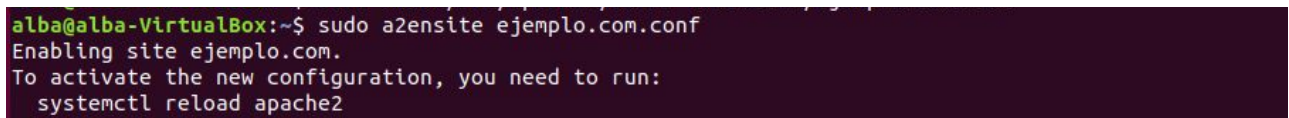
15.- Introducimos siguiente bloque de configuración, muy similar al predeterminado, pero contiene información actualizada de directorios y dominio. Actualizamos el apartado DocumentRoot a nuestro nuevo directorio y ServerAdmin a una dirección de correo electrónico que el administrador del sitio ejemplo.com pueda acceder. También añadimos dos apartados: ServerName, que establece el dominio base que debería coincidir con la definición del sitio virtual, y ServerAlias, que define otros nombres que serán atendidos de la misma forma como si fuesen el dominio base. Guardamos y cerramos el archivo cuando terminemos:



```
alba@alba-VirtualBox: ~
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.9.3 /etc/apache2/sites-available/ejemplo.com.conf Modificado

<VirtualHost *:80>
  ServerAdmin admin@ejemplo.com
  ServerName ejemplo.com
  ServerAlias www.ejemplo.com
  DocumentRoot /var/www/ejemplo.com/html
  ErrorLog ${APACHE_LOG_DIR}/error.log
  CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

16.- Habilitamos el archivo usando la herramienta a2ensite:



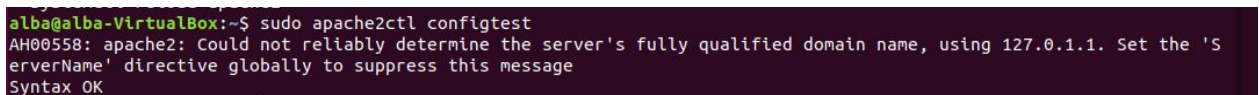
```
alba@alba-VirtualBox:~$ sudo a2ensite ejemplo.com.conf
Enabling site ejemplo.com.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

17.- Deshabilitamos el sitio por defecto definido en 000-default.conf:



```
alba@alba-VirtualBox:~$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

18.- A continuación, probamos la configuración en busca de errores. Debería aparecernos una salida con el siguiente mensaje:

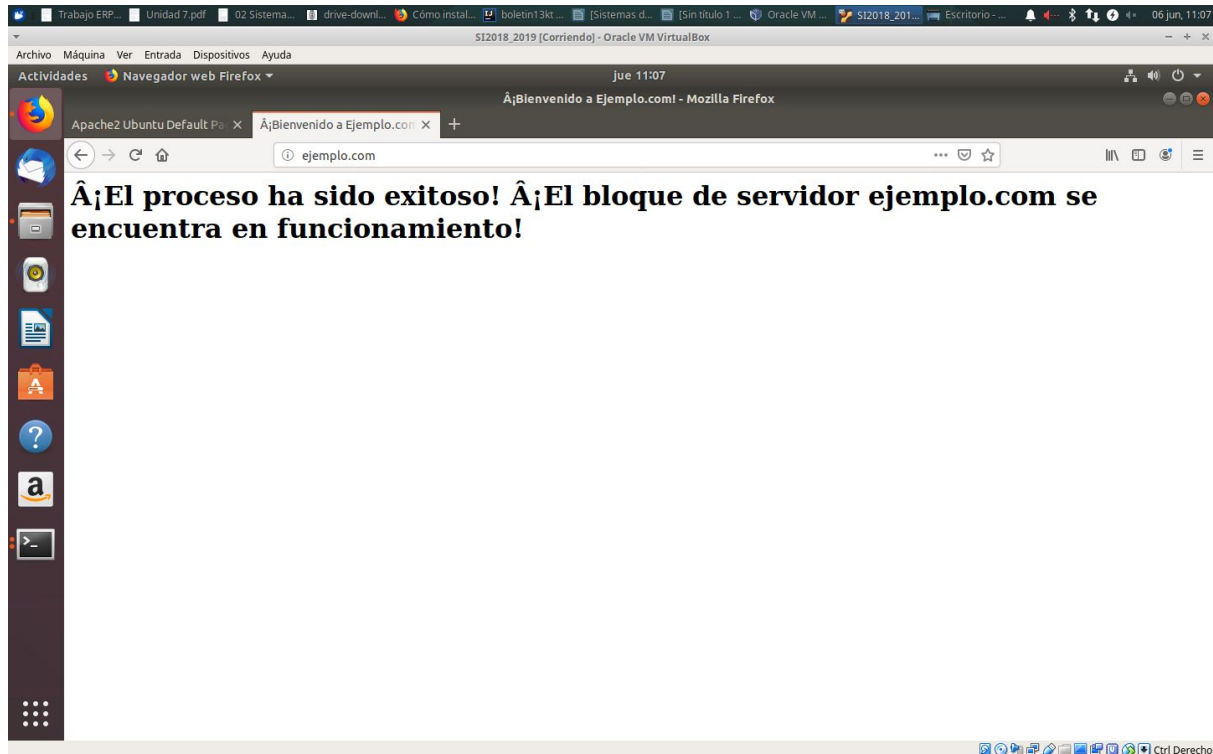


```
alba@alba-VirtualBox:~$ sudo apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK
```

19.- Reiniciamos Apache para que se apliquen los cambios:

```
alba@alba-VirtualBox:~$ sudo systemctl restart apache2
alba@alba-VirtualBox:~$
```

20.- Finalmente, podemos comprobar si funciona introduciendo en nuestro navegador la dirección `http://ejemplo.com`, donde deberíamos ver algo similar:



Comandos de interés

Ahora que ya tenemos un servidor web activo y en ejecución, es interesante familiarizarse con algunos comandos básicos de administración:

- **Para detener nuestro servidor web:**

```
sudo systemctl stop apache2
```

- **Para iniciar nuestro servidor web:**

```
sudo systemctl start apache2
```

- **Para detener y reiniciar el servidor en un solo paso:**

```
sudo systemctl restart apache2
```

- **Si solamente estamos realizando cambios en la configuración, podemos volver a cargar Apache sin necesidad de perder las conexiones que pudieran estar activas. Para ello, se usaría el comando:**

```
sudo systemctl reload apache2
```

- **Por defecto, Apache se configura para iniciarse automáticamente cuando el servidor arranca. Si no queremos esto, se puede deshabilitar esta configuración introduciendo el comando:**

```
sudo systemctl disable apache2
```

- **Para volver a habilitar el servicio durante la ejecución. Tras ingresar este comando, Apache debería iniciarse automáticamente durante el inicio del servidor.:**

```
sudo systemctl enable apache2
```

Archivos y directorios importantes de Apache

- Contenido:

`/var/www/html`: es donde se encuentra el contenido web. Por defecto, consiste únicamente en la página predeterminada de Apache que vimos antes, y se aloja en el directorio `/var/www/html`. Esto se puede modificar en los archivos de configuración de Apache.

- Configuración del servidor

`/etc/apache2`: es el directorio de configuración de Apache. Todos los archivos de configuración de Apache se localizan en éste.

`/etc/apache2/apache2.conf`: es el archivo principal de configuración de Apache. Al modificarlo se realizan cambios en la configuración global de Apache. Este archivo es el responsable de la carga de una gran variedad de otros archivos en el directorio de configuración.

`/etc/apache2/ports.conf`: este archivo especifica los puertos a los que Apache escuchará. Por defecto, Apache escucha el puerto 80.

`/etc/apache2/sites-available/`: es el directorio donde se alojan los diferentes sitios virtuales que podrían habilitarse. Apache no utilizará los archivos de configuración que se encuentren en este directorio a menos que se encuentren enlazados con el directorio `sites-enabled`. Normalmente, la configuración de todos los bloques de servidores se hace en este directorio,

para después ser habilitados mediante su enlace con los directorios usando el comando `a2ensite`.

`/etc/apache2/sites-enabled/`: es el directorio donde se alojan los diferentes sitios virtuales que se encuentren habilitados. Normalmente, éstos se crean usando `a2ensite` para enlazar los archivos de configuración que se encuentran en el directorio `sites-available`. Apache lee los archivos de configuración y los enlaces que se encuentren en este directorio en el momento de su arranque o reinicio, para después compilar una configuración completa.

`/etc/apache2/conf-available/`, `/etc/apache2/conf-enabled/`: estos directorios presentan la misma relación que se da entre los directorios `sites-available` y `sites-enabled`, solo que éstos son usados para guardar los fragmentos de configuración que no pertenecen a un sitio virtual. Los archivos dentro del directorio `conf-available` se pueden habilitar con el comando `a2enconf` y deshabilitar con el comando `a2disconf`.

`/etc/apache2/mods-available/`, `/etc/apache2/mods-enabled/`: estos directorios contienen tanto los módulos disponibles como los habilitados. Los archivos terminados en `.load` contienen fragmentos que permiten cargar módulos específicos, mientras que los archivos terminados en `.conf` contienen la configuración de dichos módulos. Los módulos se pueden habilitar y deshabilitar usando los comandos: `a2enmod` y `a2dismod`.

- Archivos de registro del servidor

`/var/log/apache2/access.log`: por defecto, toda solicitud realizada al servidor web se registra en este archivo, excepto que se configure Apache para hacerlo de una manera distinta.

`/var/log/apache2/error.log`: por defecto, los errores se registran en este archivo. El apartado `LogLevel`, dentro de la configuración de Apache, especifica el nivel de detalle con el cual se registra el contenido del error.