

Práctica 3 - Sesiones I y II - Servicios de red avanzados

Entregable

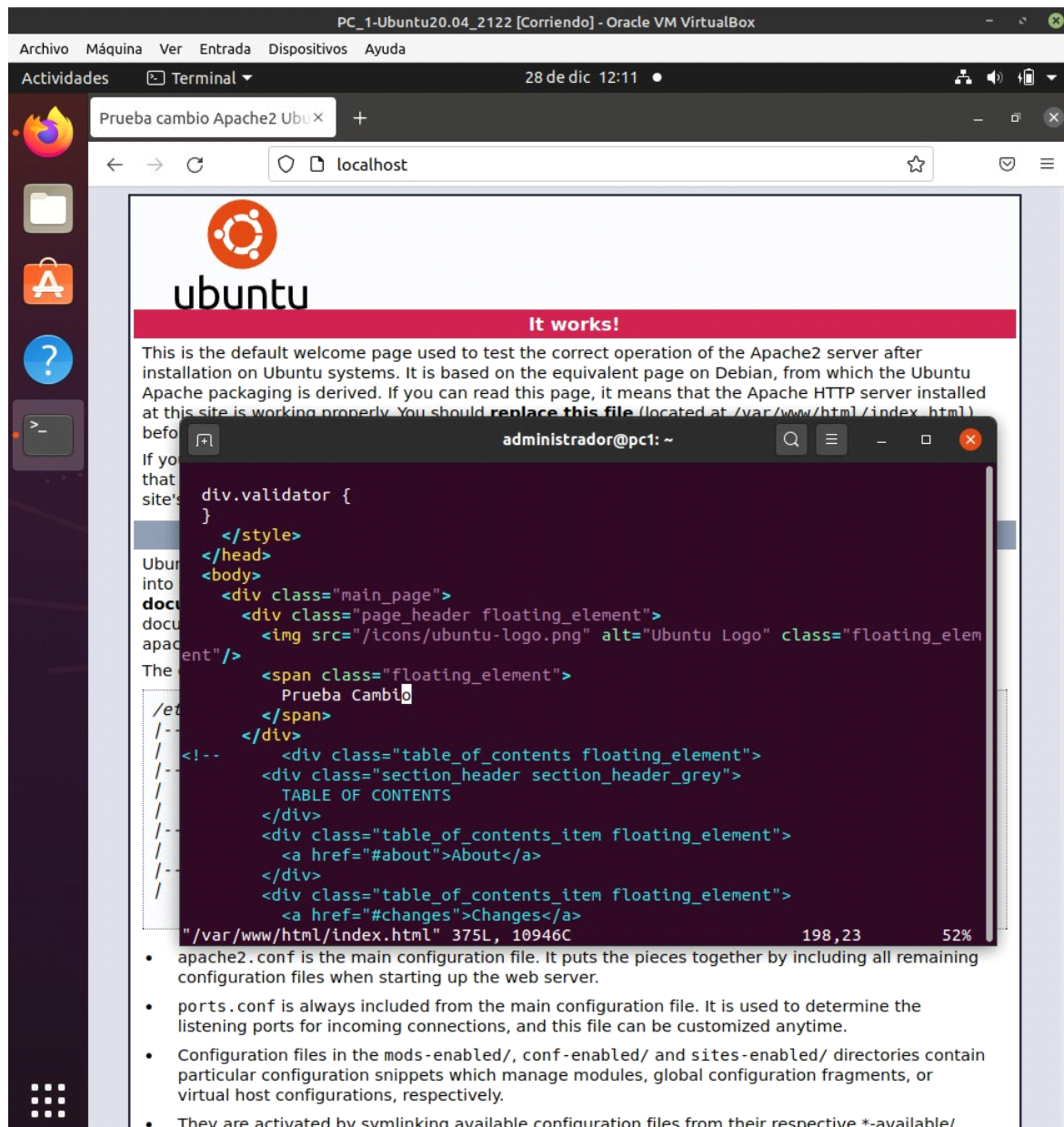
- **Información básica y requisitos para la entrega de las tareas**

- 1) Ser concisos y breves en la respuesta a cada tarea.
- 2) Ceñirse al espacio dedicado para cada tarea.
- 3) No olvidar escribir el nombre de cada integrante de la pareja y la isla en donde normalmente trabaja la pareja.
- 4) No se evaluarán tareas que ya se evaluaron en el laboratorio.
- 5) Adaptar el escenario virtualizado a la isla en donde normalmente trabaja la pareja.

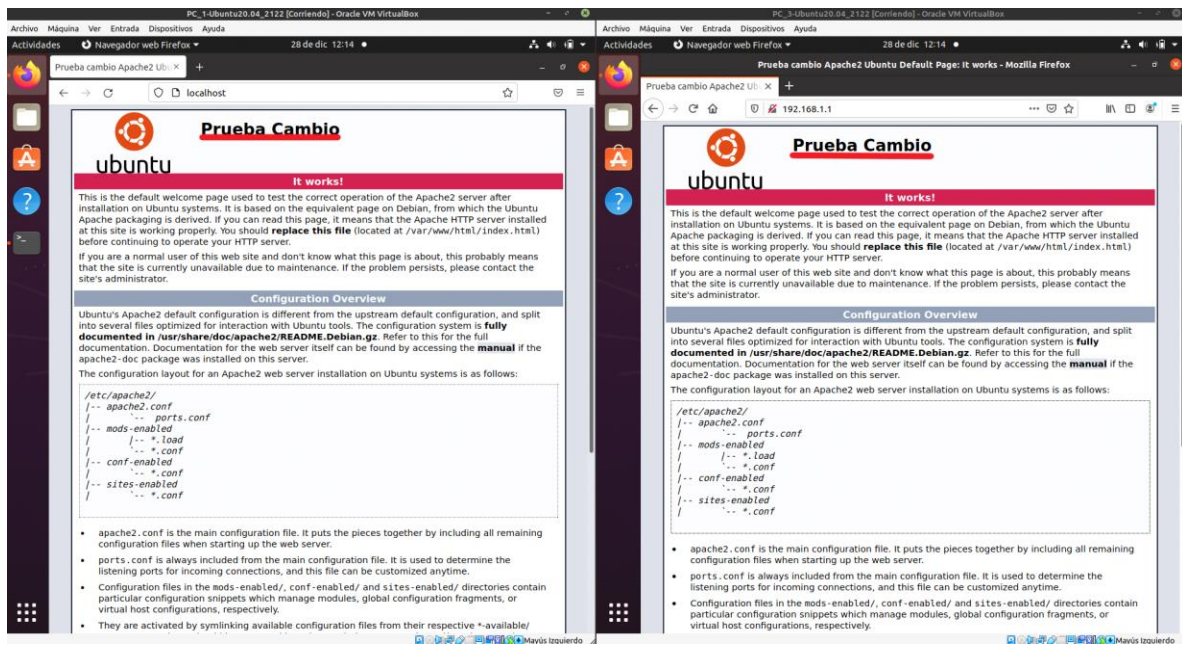
- **Realización práctica: HTTP**

- Habilite el servicio *http* en su equipo servidor. Abra un navegador web y pruebe a visitar la página de inicio desde dicho equipo (<http://localhost> o <http://127.0.0.1>). Modifique el contenido de la página de inicio, y compruebe que la dirección de su servidor es accesible.

Lo primero que se hace es meternos en el navegador en la página <http://localhost>, en la que se ve la página del fondo de la foto, en la que se ve el contenido del archivo `/var/www/html/index.html`. A continuación hacemos este cambio en el archivo `index.html`, en el que cambiamos la cabecera de la página principal del servidor por "Prueba Cambio".



El cambio realizado antes se puede observar si accedemos al localhost del servidor en el navegador web tanto desde el propio servidor como desde el cliente.



- Inspeccione el fichero `apache2.conf` e identifique las directivas más relevantes.

A continuación se muestran algunas directivas relevantes del fichero `apache2.conf`, entre las que se encuentran:

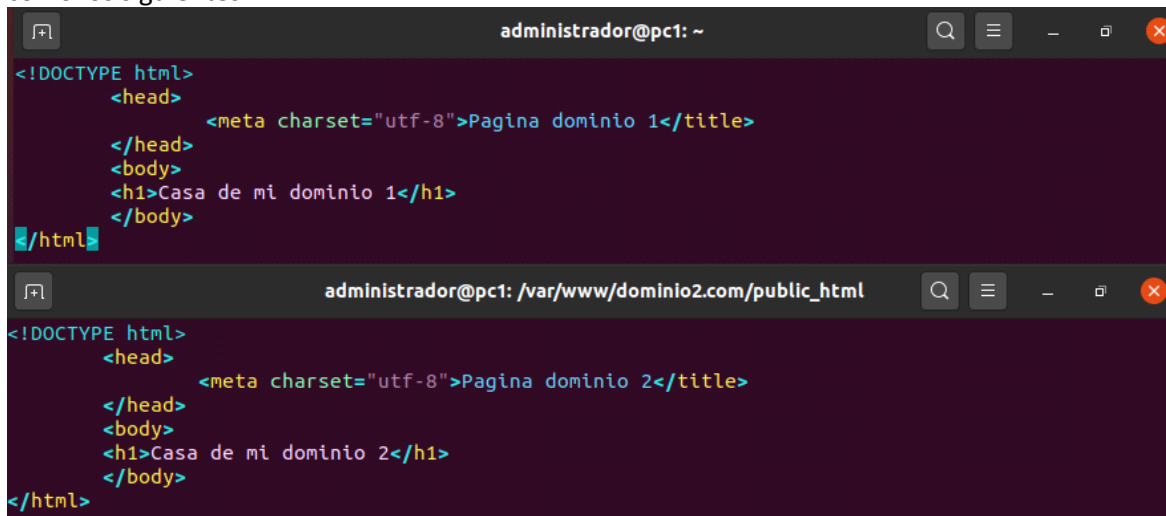
- Timeout: El tiempo que transcurre antes de recibir y enviar time outs.
- MaxKeepAliveRequests: La cantidad máxima de peticiones que se pueden hacer durante una conexión.
- KeepAliveTimeout: Número máximo de segundos que se espera para la siguiente petición del mismo cliente para la misma conexión.
- Errorlog: La localización del archivo de error.

```
#
# Timeout: The number of seconds before receives and sends time out.
#
Timeout 300
#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
KeepAlive On
#
# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited amount.
# We recommend you leave this number high, for maximum performance.
#
MaxKeepAliveRequests 100
#
# KeepAliveTimeout: Number of seconds to wait for the next request from the
# same client on the same connection.
#
KeepAliveTimeout 5
```

```
# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
# container, error messages relating to that virtual host will be
# logged here.  If you *do* define an error logfile for a <VirtualHost>
# container, that host's errors will be logged there and not here.
#
ErrorLog ${APACHE_LOG_DIR}/error.log
#
```

- Cree 2 hosts virtuales con páginas de inicio diferentes y compruebe que son servidos convenientemente ante peticiones del cliente.

Para crear 2 virtual hosts lo primero que hay que hacer es crear el directorio raíz de de cada host virtual, en nuestro caso serán `sudo "mkdir -p /var/www/dominio1.com/public_html"` `"sudo mkdir -p /var/www/dominio2.com/public_html"`, dentro de cada dominio hay que crear un archivo `"index.html"` como los siguientes.



The image shows two terminal windows. The top window is titled 'administrador@pc1: ~' and displays the content of an 'index.html' file for 'dominio 1'. The bottom window is titled 'administrador@pc1: /var/www/dominio2.com/public_html' and displays the content of an 'index.html' file for 'dominio 2'.

```
administrador@pc1: ~
<!DOCTYPE html>
<head>
  <meta charset="utf-8">Pagina dominio 1</title>
</head>
<body>
  <h1>Casa de mi dominio 1</h1>
</body>
</html>

administrador@pc1: /var/www/dominio2.com/public_html
<!DOCTYPE html>
<head>
  <meta charset="utf-8">Pagina dominio 2</title>
</head>
<body>
  <h1>Casa de mi dominio 2</h1>
</body>
</html>
```

Después hay que cambiar los propietarios de los directorios y subdirectorios de los ficheros anteriores al usuario de apache2 `"www-data"` como se ve en la siguiente foto.

```
administrador@pc1: ~  
administrador@pc1:~$ chown -R www-data: /var/www/dominio1.com/  
chown: cambiando el propietario de '/var/www/dominio1.com/public_html/index.html': Operación no permitida  
chown: cambiando el propietario de '/var/www/dominio1.com/public_html': Operación no permitida  
chown: cambiando el propietario de '/var/www/dominio1.com/': Operación no permitida  
administrador@pc1:~$ sudo chown -R www-data: /var/www/dominio1.com/  
administrador@pc1:~$ sudo chown -R www-data: /var/www/dominio2.com/  
administrador@pc1:~$
```

Como se puede observar, hay que hacerlo como superusuario, porque si no no te deja hacerlo.

A continuación, en el directorio `/etc/apache2/sites-available` tenemos que modificar los archivos configuración de cada uno de los hosts virtuales que se van a crear.

```
administrador@pc1: /etc/apache2/sites-available  
GNU nano 4.8 dominio1.com.conf  
<VirtualHost *:80>  
    ServerName dominio1.com  
    ServerAlias www.dominio1.com  
    ServerAdmin webmaster@dominio1.com  
    DocumentRoot /var/www/dominio1.com/public_html  
  
    <Directory /var/www/dominio1.com/public_html>  
        Options -Indexes +FollowSymLinks  
        AllowOverride All  
    </Directory>  
    ErrorLog ${APACHE_LOG_DIR}/dominio1.com-error.log  
    CustomLog ${APACHE_LOG_DIR}/dominio1.com-access.log combined  
</VirtualHost>  
  
[ 13 líneas leídas ]  
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Texto ^J Justificar ^C Posición  
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar ^T Ortografía ^_ Ir a línea
```



```
administrador@pc1: /etc/apache2/sites-available
GNU nano 4.8                               dominio2.com.conf          Modificado
<VirtualHost *:80>
    ServerName dominio2.com
    ServerAlias www.dominio2.com
    ServerAdmin webmaster@dominio2.com
    DocumentRoot /var/www/dominio2.com/public_html

    <Directory /var/www/dominio2.com/public_html>
        Options -Indexes +FollowSymLinks
        AllowOverride All
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/dominio2.com-error.log
    CustomLog ${APACHE_LOG_DIR}/dominio2.com-access.log combined
</VirtualHost>

^G Ver ayuda  ^O Guardar   ^W Buscar    ^K Cortar Tex^J Justificar ^C Posición
^X Salir      ^R Leer fich.^_ Reemplazar ^U Pegar      ^T Ortografía ^_ Ir a línea
```

Una vez hecho esto, hay que crear los enlaces simbólicos entre los directorios *sites-available* y *sites-enabled*, esto se hace con la orden *a2ensite* y después de eso hay que comprobar si la configuración hecha tiene algún error, esto se hace con la orden *apachectl configtest*.

```
administrador@pc1: ~
administrador@pc1:~$ sudo a2ensite dominio1.com
Enabling site dominio1.com.
To activate the new configuration, you need to run:
    systemctl reload apache2
administrador@pc1:~$ sudo a2ensite dominio2.com
Enabling site dominio2.com.
To activate the new configuration, you need to run:
    systemctl reload apache2
administrador@pc1:~$ sudo apachectl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK
```

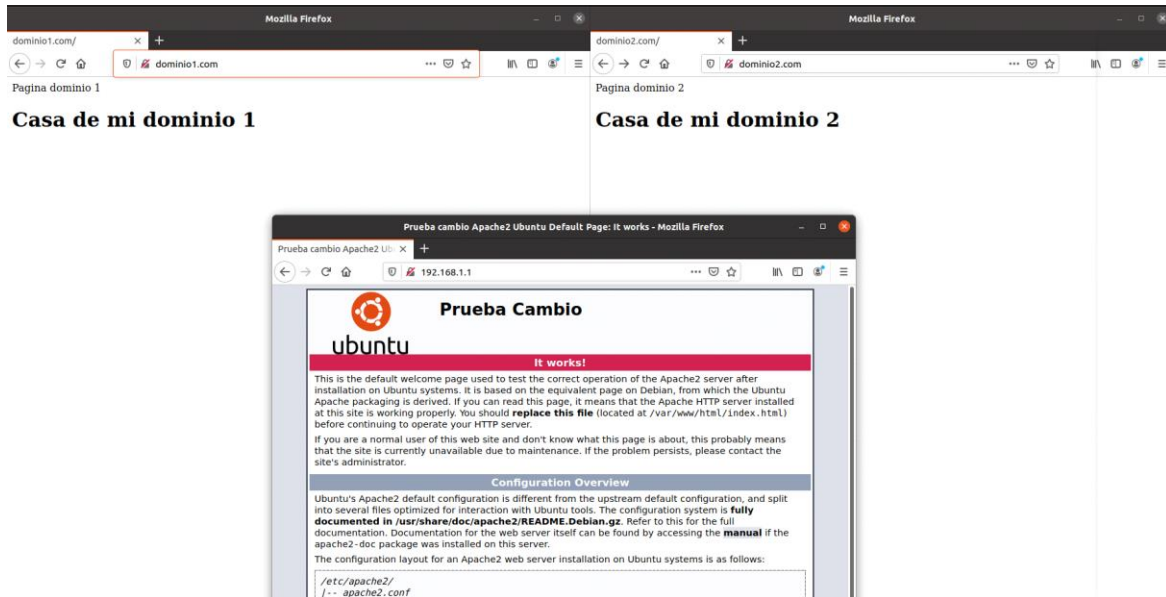
En la imagen se puede observar que aparece un error de 'ServerName', esto no supondrá un problema siempre que aparezca la sentencia "Syntax OK".

Para crearlas entradas de los hosts hay que configurar DNS para los distintos dominios virtuales, cosa que se puede hacer de forma manual en el fichero *etc/hosts* como se muestra a continuación.

```
administrador@pc1: ~  
GNU nano 4.8 /etc/hosts  
127.0.0.1 pc1  
127.0.1.1 administrador-PC1  
192.168.1.1 dominio1.com  
192.168.1.1 dominio2.com  
  
# 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  
  
[ 11 líneas escritas ]  
^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Text ^J Justificar ^C Posición  
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar ^T Ortografía ^_ Ir a línea
```

Para terminar, reiniciamos el servicio apache2 y comprobamos desde el servidor y desde el cliente (las fotos están en ese orden) que todo funciona perfectamente





- Cree una página de acceso restringido (es decir, que requiera usuario y contraseña antes de mostrarla) en <http://<IP o nombre del servidor>/restringida/>. Utilice como credenciales de acceso el usuario *admin* y la contraseña *1234*.

Para crear una página de acceso restringido hay que hacer como en el punto anterior, crear primero la carpeta del directorio del archivo `sudo mkdir -p /var/www/html/mi_zona_restringida` a continuación, en el archivo `/etc/apache2/apache2.conf` hay que escribir "AllowOverride All" en los 3 lugares posibles.

```

administrador@pc1: ~
GNU nano 4.8 /etc/apache2/apache2.conf Modificado
# the latter may be used for local directories served by the web server. If
# your system is serving content from a sub-directory in /srv you must allow
# access here, or in any related virtual host.
<Directory />
    Options FollowSymLinks
    AllowOverride All
    Require all denied
</Directory>

<Directory /usr/share>
    AllowOverride All
    Require all granted
</Directory>

<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>

^G Ver ayuda ^O Guardar ^W Buscar ^K Cortar Tex ^J Justificar ^C Posición
^X Salir ^R Leer fich. ^E Reemplazar ^U Pegar ^T Ortografía ^_ Ir a línea

```


Después de esto hay que crear un fichero de contraseñas con la utilidad "htpasswd" para el usuario admin, como se muestra a continuación, en el que cuando te piden la contraseña metemos 1234.

```
administrador@pc1:~$ sudo htpasswd -c /usr/local/apache/passwd/passwords admin
New password:
Re-type new password:
Adding password for user admin
```

A continuación, hay que editar el archivo `/var/www/html/mi_zona_restringida` con las directivas de la foto.

```
administrador@pc1: ~
GNU nano 4.8 /var/www/html/mi_zona_restringida/.htaccess
AuthType Basic
AuthName "Directorio con control de acceso"
# opcionales
AuthUserFile "/usr/local/apache/passwd/passwords"
Require user miusuario
```

Como última opción hay que reiniciar el servicio apache2, con lo que si nos metemos en la dirección del archivo en el navegador, ya sea del servidor o del cliente y nos pide el usuario y la contraseña (admin y 1234, respectivamente), como queríamos.

