

# Servizo Web: Apache

## ESCENARIO

### Máquinas virtuais:

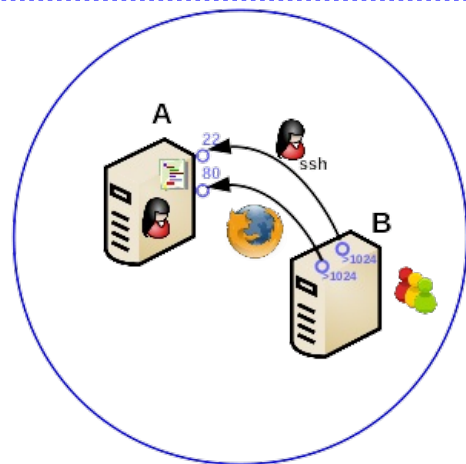
RAM  $\leq$  2048MB    CPU  $\leq$  2    PAE/NX habilitado  
Rede: 192.168.120.0

### Máquina virtual A:

Rede Interna  
Servidor SSH: openssh-server  
Servidor Web: Apache (apache2)  
ISO: Kali Live amd64  
IP/MS: 192.168.120.100/24  
BIOS: Permite arranque dispositivo extraíble: CD/DVD, USB

### Máquina virtual B:

Rede Interna  
Cliente SSH: openssh-client (ssh)  
Cliente Web: Navegador (firefox)  
ISO: Kali Live amd64  
IP/MS: 192.168.120.101/24




**LIMITACIÓN DE RESPONSABILIDADE** O autor do presente documento declina calquera responsabilidade asociada ao uso incorrecto e/ou malicioso que puidese realizarse coa información exposta no mesmo. Por tanto, non se fai responsable en ningún caso, nin pode ser considerado legalmente responsable en ningún caso, das consecuencias que poidan derivarse da información contida nel ou que esté enlazada dende ou hacia el, incluíndo os posibles erros e información incorrecta existentes, información difamatoria, así como das consecuencias que se poidan derivar sobre a súa aplicación en sistemas de información reais e/ou virtuais. Este documento foi xerado para uso didáctico e debe ser empregado en contornas privadas e virtuais controladas co permiso correspondente do administrador desas contornas.

## NOTAS:

- Cliente ssh GNU/Linux: **comando ssh (paquete openssh-client)**
- Servidor SSH GNU/Linux: Paquete **openssh-server** (# apt update && apt -y install openssh-server).  
Ficheiro de configuración: **/etc/ssh/sshd\_config (man sshd\_config)**
- Servidor Web Apache:
  - Paquete apache2 (# apt update && apt -y install apache2).
  - **Nomenclatura versións: 2.X.revision**, onde:
    - X toma valor par → a versión é estable
    - X toma valor impar → a versión é de desenvolvemento

## NOTAS:

- Documentación oficial sobre o Servidor web **Apache (v2.4)** 
- Paquete apache2 (# apt update && apt -y install apache2).
- Configuración en: **/etc/apache2/ (man apache2)**

**apache2.conf** → Ficheiro de configuración principal. Non deberíase modificarse con novas directivas. Así:

- Se se quere extender a configuración global de Apache deberíanse incluír noutros ficheiros de configuración dentro do directorio /etc/apache2/conf-available e activalos en /etc/apache2/conf-enabled (a2enconf, a2disconf).
- No caso de cambiar os portos e socket de conexión deberíase modificar o ficheiro /etc/apache2/ports.conf

Aparecen por liña pares de valores: directiva e argumento. As directivas non distinguen entre maiúsculas e minúsculas e os argumentos si poden distinguir.

**ports.conf** → Ficheiro de configuración que contén as directivas de configuración de portos TCP e direccións IP onde escoitar as conexións do servidor web Apache.

**envvars** → Ficheiro de configuración que contén as variables de entorno que poden ser empregadas na configuración polos ficheiros de configuración que o consideren.

- **APACHE\_RUN\_USER=www-data** → usuario que executa o servizo web Apache
- **APACHE\_RUN\_GROUP=www-data** → grupo que executa o servizo web Apache
- **APACHE\_PID\_FILE=/var/run/apache2\$SUFFIX/apache2.pid** → pid de execución do servizo web Apache
- **APACHE\_ULIMIT\_MAX\_FILES='ulimit -n 65536'** → ulimit determina o número máximo de ficheiros abertos permitidos para servizo web Apache

**conf-available** → Os ficheiros engadidos neste directorio son invocados (include) polo ficheiro de configuración global /etc/conf/apache2.conf. Este directorio é un bo lugar para engadir directivas na configuración. Todos os arquivos neste directorio para ser tidos en conta deben rematar coa extensión **.conf**

Os ficheiros situados neste directorio poden ser habilitados e deshabilitados usando os comandos **a2enconf** e **a2disconf** respectivamente.

**conf-enabled** → Habilita a configuración dos ficheiros. Os ficheiros de configuración habilitados son aqueles que neste directorio conteñen ligazóns ao directorio /etc/apache2/conf-available/

- **a2enconf** → Comando que permite habilitar os ficheiros de configuración situados no directorio /etc/apache2/conf-available/ engadindo ligazóns dende /etc/apache2/conf-enabled a /etc/apache2/conf-available
- **a2disconf** → Comando que permite deshabilitar os ficheiros de configuración activados no directorio /etc/apache2/conf-enabled eliminando ligazóns existentes dende /etc/apache2/conf-enabled a /etc/apache2/conf-available

**mods-available** → Este directorio contén ficheiros que rematan coa extensión **.load** e **.conf**. Os ficheiros **.load** conteñen as directivas de configuración necesarias para cargar o módulo en cuestión. Os ficheiros **.conf** conteñen as directivas necesarias para empregar o módulo en cuestión. Os módulos situados neste directorio poden ser habilitados e deshabilitados usando os comandos **a2enmod** e **a2dismod** respectivamente.

**mods-enabled** → Habilita os módulos. Os módulos habilitados son aqueles que neste directorio conteñen ligazóns ao directorio /etc/apache2/mods-available/

- **a2enmod** → Comando que permite habilitar os ficheiros dos módulos situados no directorio /etc/apache2/mods-available/ engadindo ligazóns dende /etc/apache2/mods-enabled a /etc/apache2/mods-available
- **a2dismod** → Comando que permite deshabilitar os ficheiros dos módulos activados no directorio /etc/apache2/mods-enabled eliminando ligazóns existentes dende /etc/apache2/mods-enabled a /etc/apache2/mods-available

**sites-available** → Similar ao directorio mods-available, agás que contén as directivas para diferentes VirtualHost. O hostname non ten porque corresponder exactamente co nome do ficheiro. Debian por defecto posúe 2 ficheiros neste directorio:

- 000-default.conf → VirtualHost para o porto TCP 80(http)(Ver arquivo /etc/services)
- default-ssl.conf → VirtualHost para o porto TCP 443(https)(Ver arquivo /etc/services).

Este sitio pode estar activado pero será funcional cando o modulo ssl tamén está activado. Os sitios(arquivos) situados neste directorio poden ser habilitados e deshabilitados usando os comandos **a2ensite** e **a2dissite** respectivamente.

**sites-enabled** → Habilita os VirtualHost(sitios). Os sitios(arquivos) habilitados son aqueles que neste directorio conteñen ligazóns ao directorio /etc/apache2/sites-available/

- **a2ensite** → Comando que permite habilitar os sitios(VirtualHost) situados no directorio /etc/apache2/sites-available/ engadindo ligazóns dende /etc/apache2/sites-enabled a /etc/apache2/sites-available
- **a2dissite** → Comando que permite deshabilitar os sitios(VirtualHost) activados no directorio /etc/apache2/sites-enabled eliminando ligazóns existentes dende /etc/apache2/sites-enabled a /etc/apache2/sites-available

## ■ Directivas:



**Include** → Directiva que permite engadir ficheiros de configuración á configuración global. Esta directiva ignora ficheiros que non finalizan na extensión .conf.

**Require** → Directiva para permitir ou denegar acceso aos recursos.

**DocumentRoot** → Directiva que define a ruta do cartafol onde o servidor web Apache aloxa as páxinas. Por exemplo: /var/www/html

**Listen** → Directiva que define IP/Porto TCP/Protocolo onde escoita o servidor web Apache (Listen 192.168.120.100:8443 https) .

**VirtualHost** → Directiva que define a posibilidade de aloxar varios dominios no mesmo servidor Apache.

**ServerName** → Directiva que define o nome DNS do sitio(dominio) aloxado.

**ServerAlias** → Directiva que define outros nomes DNS para a mesma páxina.

**AllowOverride** → Directiva que especifica que outras directivas poden ser postas en cada ficheiro .htaccess (ficheiros de configuración por directorio).

**ServerSignature** → Directiva que permite a configuración dunha liña de pé de páxina baixo documentos xerados polo servidor (mensaxes de erro, versión...)

**Options ±argumento** → Directiva que controla que funcións do servidor están dispoñibles nun directorio particular. Por exemplo: Options +Indexes especifica que se non existe nun directorio o ficheiro index.html amose o contido do directorio.

**Timeout** → Directiva que especifica a cantidade de tempo que o servidor agardará por certos eventos antes de fallar unha solicitude.

**MaxKeepAliveRequests** → Directiva que especifica o número máximo de peticións permitidas por cada conexión persistente.

**KeepAliveTimeout** → Directiva que especifica a cantidade de tempo que o servidor esperará solicitudes posteriores nunha conexión persistente.

**ErrorLog** → Directiva que especifica o nome do ficheiro no que o servidor rexistrará os erros que atope. Se a ruta do ficheiro non é absoluta, suponse que é relativo ao ServerRoot.

**LogFormat** → Directiva que especifica o formato a empregar para un ficheiro de rexistro.

**ServerRoot** → Directiva que especifica o directorio base da instalación do servidor.

## ■ Seccións:

Prioridade/Alcance			Sección	Aplicación	Exemplo
-	↓	+	Directory .htaccess (AllowOverride ↑)	FileSystem	<Directory /var/www/html/prioridade> (ruta absoluta do sistema de ficheiros <sup>1</sup> )
+		-	Files		
			Location	DocumentRoot	<Location /prioridade> (ruta relativa ao DocumentRoot <sup>1</sup> )
			VirtualHost		

Táboa. Seccións

### Contexto das directivas:



### Estrutura dunha sección:

```
<SectionName [pathDirectorio | "pattern"]>
  Directiva1 valor1
  Directiva2 valor2
  Directiva3 valor3
  ...
  DirectivaN valorN
</SectionName >
```

**Directory** → Sección que inclúe directivas que actúan sobre un directorio. O directorio en cuestión sempre debe ser especificado mediante unha ruta absoluta do sistema de ficheiros.

Xeralmente, so se deberían empregar ficheiros .htaccess cando non se ten acceso ao ficheiro principal de configuración do servidor, por exemplo en servidores compartidos onde non se ten acceso como root no servidor.

**Files** → Sección que inclúe directivas que actúan sobre os ficheiros que se especifiquen.

**Location** → Sección similar á sección Directory. O directorio en cuestión sempre debe ser especificado mediante unha ruta relativa ao DocumentRoot.

**VirtualHost** → Sección que inclúe directivas que actúan soamente sobre un específico VirtualHost (hostname ou IP).

## Resumo Prácticas Exemplos

- No **Exemplo1. Modificar Listen**, veremos como poder escoitar en distintos portos e IPs o protocolo HTTP e o protocolo HTTPS.
- No **Exemplo2. Alojar cartafoles**, veremos como poder alojar múltiples páxinas web no servidor web **Apache**, pero todas pertencentes ao mesmo sitio/dominio, é dicir, todas pertencentes a exemplo.local
- No **Exemplo3. Xerar virtualhost** veremos como poder alojar páxinas de distintos dominios no mesmo servidor web mediante a configuración de hosts virtuais ou virtualhosts.

Os virtualhosts basicamente o que fan é permitir que un mesmo servidor web poida alojar múltiples dominios, así configurando hosts virtuais podemos alojar: exemplo1.local, exemplo2.local..., exemploN.local no mesmo servidor web. Cada empresa terá o seu virtualhost único e independente das demais.

Aínda que como se comentou anteriormente cada virtualhost é único e independente dos demais, todo aquilo que non estea incluído na definición de cada virtualhost herdarase da configuración principal: /etc/apache2/apache2.conf, así, se se quere definir unha directiva común en tódolos virtualhost non se debe modificar cada un dos virtualhost introducindo esa directiva senón que se debe definir esa directiva nun arquivo de configuración dentro de /etc/apache2/conf-available e empregar o comando a2enconf para habilitar esa configuración no servidor web Apache, de tal forma que todos os virtualhost herdarán esa directiva. Por exemplo en /etc/apache2/conf-available/security.conf pódese atopala directiva ServerSignature On, que engade unha liña contendo a versión do servidor e o nome do VirtualHost.

Existe tres tipos de virtualhost: baseados en nome, baseados en IP e baseados en varios servidores principais. Imos centrarnos nos virtualhost baseados en nome.

- No **Exemplo4. Control de acceso** imos tratar distintos tipos de control de acceso (autenticación http basic, IP), os arquivos tipo **.htaccess** e seccións <Directory> e directivas Order, Allow, Deny (todavía funcionais pero desaconsellables) e Require (<RequireAll>)

HTTP proporciona un método de autenticación básico de usuarios: basic. Este método ante unha petición do cliente(navegador web) ao servidor cando se solicita unha URL amosará un diálogo pedindo usuario e contrasinal. Unha vez autenticado o usuario, o cliente volverá facer a petición ao servidor pero agora enviando o usuario e contrasinal, en texto claro (sen cifrar) proporcionados no diálogo. É recomendable entón se se emprega este método que se faga combinado con conexión SSL (HTTPS).

- No **Exemplo5. Prioridade seccións/directivas** imos ver que sección prevalece cando unha mesma directiva é configurada en distintas seccións (Ver Táboa. Seccións)

### Máquina virtual A: Kali amd64

1. Na contorna gráfica abrir un terminal e executar:

```
kali@kali:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.  
kali@kali:~$ passwd kali #Cambiar o contrasinal do usuario kali. Por como contrasinal abc123. (Olo que o contrasinal  
ten un caracter punto final).
```

2. Cambiar hostname da máquina virtual A. Por kaliA como hostname:

```
kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo  
(/etc/sudoers, visudo)  
root@kali:~# echo 'kaliA' > /etc/hostname #Indicar ao sistema o valor do hostname.  
root@kali:~# echo 'kernel.hostname=kaliA' >> /etc/sysctl.conf #Indicar ao kernel o valor do hostname.  
root@kali:~# sysctl -p #Activar o cambio de hostname sen ter que pechar sesión nin reiniciar  
root@kali:~# exit #Sair da consola local sudo na que estabamos a traballar para voltar á consola local de kali.  
kali@kali:~$ exit #Pechar o terminal saíndo da consola local do usuario kali.
```

3. Configurar a rede:

Na contorna gráfica abrir un terminal e executar:

```
kali@kaliA:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.  
kali@kaliA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando  
sudo (/etc/sudoers, visudo)  
root@kaliA:~# /etc/init.d/avahi-daemon stop #Parar o demo avahi-daemon(control resolución de nomes) para  
poder configurar de forma manual a configuración de rede e non ter conflito con este demo.  
root@kaliA:~# /etc/init.d/network-manager stop || pkill NetworkManager #Parar o demo network-  
manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) para poder configurar doutro xeito (co  
comando ip(ifconfig) de forma manual ou mediante networking (ficheiros /etc/init.d/networking, /etc/init.d/networking.d) a  
configuración de rede e non ter conflito con este xestor.  
root@kaliA:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina A, as  
tarxetas de redes: loopback(lo) e interna(eth0).  
root@kaliA:~# ip addr add 192.168.120.100/24 dev eth0 #Configurar a tarxeta de rede interna eth0, coa  
IP: 192.168.120.100 e máscara de subrede: 255.255.255.0.  
root@kaliA:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina A, as  
tarxetas de redes: loopback(lo) e interna(eth0).  
root@kaliA:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface  
de rede local eth0
```

4. Comprobar estado do Servidor SSH:

```
root@kaliA:~# /etc/init.d/ssh status #Comprobar o estado do servidor SSH, por defecto non está arrancado.  
root@kaliA:~# nc -vz localhost 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do servidor ssh está  
en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar  
información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o  
escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.  
root@kaliA:~# nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do  
servidor ssh está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite  
amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito  
facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.  
root@kaliA:~# netstat -natp | grep 22 #Mediante o comando netstat comprobar que o porto 22 do servidor SSH  
está en estado escoita(listen), esperando conexións. A opción -n permite non resolver nomes amosando así soamente as IPs  
e o comando ser máis rápido na execución. A opción -a equivale á opción all o que permite amosar todos os sockets  
(conectores) á escoita no servidor. A opción -t equivale a tcp o que permite buscar soamente información sobre o protocolo  
TCP. A opción -p equivale a program e amosa o PID e nome do programa ao cal pertence o socket.  
root@kaliA:~# ss -natp | grep 22 #Mediante o comando ss comprobar que o porto 22 do servidor SSH está en  
estado escoita(listen), esperando conexións. A opción -n permite non resolver nomes amosando así soamente as IPs e o  
comando ser máis rápido na execución. A opción -a equivale á opción all o que permite amosar todos os sockets  
(conectores) á escoita no servidor. A opción -t equivale a tcp o que permite buscar soamente información sobre o protocolo  
TCP. A opción -p equivale a program e amosa o PID e nome do programa ao cal pertence o socket.  
root@kaliA:~# /etc/init.d/ssh start #Arrancar o servidor SSH.  
root@kaliA:~# /etc/init.d/ssh status #Comprobar o estado do servidor SSH, agora debe estar arrancado.  
root@kaliA:~# find /etc/rc* -name "*ssh*" #Busca polas links runlevels nos cartafolios /etc/rc*  
root@kaliA:~# systemctl enable ssh #Permite que o servizo ssh sexa iniciado no arranque xerando os links nos
```

runlevels (/etc/rcX.d)

root@kaliA:~# find /etc/rc\* -name "\*ssh\*" #Busca polas links runlevels nos cartafolios /etc/rc\*

root@kaliA:~# systemctl is-enabled ssh.service #Amosa se o servizo ssh está enabled ou disabled

root@kaliA:~# nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do servidor ssh está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.

root@kaliA:~# ssh -v kali@localhost #Comprobar se o servidor SSH está activo e podemos conectarnos a el dende localhost co usuario kali e o seu contrasinal. Se é a primeira vez que nos conectamos o servidor avísanos se estamos de acordo coa autenticación. Respostamos yes e pulsamos Enter. A opción -v (modo verbose) aporta información máis detallada da conexión.

kali@kaliA:~\$ exit #Saír da consola remota ssh a que acabamos de acceder, para voltar á consola local de **root**.

root@kaliA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.

kali@kaliA:~\$



## Máquina virtual B: Kali amd64

### 5. Configuración da rede. Na contorna gráfica abrir un terminal e executar:

```
kali@kali:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
```

```
kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
root@kali:~# /etc/init.d/avahi-daemon stop #Parar o demo avahi-daemon(control resolución de nomes) para poder configurar de forma manual a configuración de rede e non ter conflito con este demo.
```

```
root@kali:~# /etc/init.d/network-manager stop || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) para poder configurar doutro xeito (co comando ip(ifconfig) de forma manual ou mediante networking (ficheiros /etc/init.d/networking, /etc/init.d/networking.d) a configuración de rede e non ter conflito con este xestor.
```

```
root@kali:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina B as tarxetas de redes: loopback(lo) e interna(eth0).
```

```
root@kali:~# ip addr add 192.168.120.101/24 dev eth0 #Configurar a tarxeta de rede interna eth0, coa IP: 192.168.120.101 e máscara de subrede: 255.255.255.0.
```

```
root@kali:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina B as tarxetas de redes: loopback(lo) e interna(eth0).
```

```
root@kali:~# ping -c4 192.168.120.101 #Comprobar mediante o comando ping a conectividade coa interface de rede local eth0
```

```
root@kali:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface de rede da máquina virtual A
```

```
root@kali:~# echo '192.168.120.100 kaliA' >> /etc/hosts #Engadir no ficheiro /etc/hosts, é dicir, na táboa estática de búsqueda para nomes de host (DNS) o nome kaliA, para que atenda á IP 192.168.120.100
```

```
root@kali:~# ping -c4 kaliA #Comprobar mediante o comando ping a conectividade coa interface de rede da máquina virtual A
```

### 6. Cambiar hostname da máquina virtual B. Por kaliB como hostname:

```
root@kali:~# echo 'kaliB' > /etc/hostname #Indicar ao sistema o valor do hostname.
```

```
root@kali:~# echo 'kernel.hostname=kaliB' >> /etc/sysctl.conf #Indicar ao kernel o valor do hostname.
```

```
root@kali:~# sysctl -p #Activar o cambio de hostname sen ter que pechar sesión nin reiniciar
```

```
root@kali:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de kali.
```

```
kali@kali:~$ exit #Pechar o terminal saíndo da consola local do usuario kali.
```

### 7. <sup>SSH</sup> **B → A** Acceder mediante SSH dende a máquina virtual B á máquina virtual A. Dende agora executaremos sempre os comandos dende a máquina virtual B, a través da consola SSH:

Na contorna gráfica abrir un terminal e executar:

```
kali@kaliB:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
```

```
kali@kaliB:~$ nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
```

```
kali@kaliB:~$ nc -vz kaliA 22 #Mediante o comando nc(netcat) comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
```

```
kali@kaliB:~$ ssh -v kali@192.168.120.100 #Comprobar se o servidor SSH está activo e podemos conectarnos a el. Agora accedemos como o usuario kali a través da conexión cifrada SSH.
```

```
kali@kaliA:~$
```

### 8. Activar Servidor Web Apache:

```
kali@kaliA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
root@kaliA:~# /etc/init.d/apache2 status #Comprobar o estado do servidor web Apache.
```

```
root@kaliA:~# /etc/init.d/apache2 start #Iniciar o servidor web Apache.
```

```
root@kaliA:~# /etc/init.d/apache2 status #Comprobar o estado do servidor web Apache.
```

```
root@kaliA:~# nc -vz 192.168.120.100 80 #Mediante o comando nc(netcat) comprobar se o porto 80 do servidor web Apache está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 80 é o porto TCP a escanear.
```



No caso da distribución Kali xa temos instalado o servidor web Apache, pero nunha distribución baseada en Debian poderíamos instalalo do seguinte xeito:

```
# apt update #Actualizar o listado de paquetes dos repositorios (/etc/apt/sources.list, /etc/apt/sources.list.d)
# apt search apache2 #Buscar calquera paquete que coincida co patrón de búsqueda apache2
# apt -y install apache2 #Instalar o paquete apache2, é dicir, instalar o servidor HTTP apache2. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.
```

9. Lanzar na máquina virtual B (Kali) un navegador e visitar a IP 192.168.120.100 ou a URL <http://192.168.120.100>

10. Permisos apache:

```
root@kaliA:~# chown -R www-data. /var/www/html/ #Cambiar usuario propietario www-data e grupo propietario www-data a toda a árbore de ficheiros e directorios que colgan do directorio DocumentRoot de Apache: /var/www/html
root@kaliA:~# chmod 444 /var/www/html/index.html #Cambiar a só lectura os permisos ugo do ficheiro index.html situado en /var/www/html, é dicir, establecer os permisos r--r--r-- (soamente lectura para o usuario propietario, o grupo propietario e o resto do mundo)
root@kaliA:~# /etc/init.d/apache2 restart #Reiniciar o servidor web Apache.
root@kaliA:~# /etc/init.d/apache2 status #Comprobar o estado do servidor web Apache.
```

11. Actualizar na máquina virtual B (Kali) a páxina referente á URL <http://192.168.120.100>

## 12. Exemplo1. Modificar directiva Listen.

Na instalación defínese en ports.conf a directiva Listen nos portos TCP 80 e 443(ver /etc/services), atendendo a calquera IP do servidor. Imos configurar esta directiva para que atenda:

- a. **Listen 80 →** Todas as interfaces do servidor no porto TCP 80 (Configuración por defecto):

```
root@kaliA:~# grep Listen /etc/apache2/ports.conf #Buscar no ficheiro /etc/apache2/ports.conf
mediante o comando grep o patrón de texto 'Listen'
root@kaliA:~# nc -vz localhost 80 #Mediante o comando nc(netcat) comprobar se o porto 80 do servidor
web Apache está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que
permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e
de igual xeito facer o escaneo ao/s porto/s solicitados. O número 80 é o porto TCP a escanear.
root@kaliA:~# nc -vz 192.168.120.100 80 #Mediante o comando nc(netcat) comprobar se o porto 80 do
servidor web Apache está en estado escoita(listen) na IP 192.168.120.100, esperando conexións. A opción -v
corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z
permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 80 é o porto
TCP a escanear.
```

- b. **Listen 3800 →** Todas as interfaces do servidor a escoita no porto TCP 3800 ademais do porto TCP 80):

```
root@kaliA:~# sed -i 's/^Listen 80/Listen 80\nListen 3800/' /etc/apache2/ports.conf #Pór
debaixo da liña Listen 80, outra liña co contido Listen 3800 para indicar que agora o servidor web Apache tamén está
a escoita no porto TCP 3800
root@kaliA:~# /etc/init.d/apache2 reload #Recargar a configuración do Servidor Web Apache
root@kaliA:~# nc -vz localhost 80 3800 #Mediante o comando nc(netcat) comprobar se os portos 80 e
3800 do servidor web Apache están en estado escoita(listen), esperando conexións. A opción -v corresponde á opción
verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver
PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. Os números 80 e 3800 son os portos TCP
a escanear.
root@kaliA:~# nc -vz 192.168.120.100 80 3800 #Mediante o comando nc(netcat) comprobar se os
portos 80 e 3800 do servidor web Apache están en estado escoita(listen) na IP 192.168.120.100, esperando conexións.
A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A
opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. Os números
80 e 3800 son os portos TCP a escanear.
```

- c. **Listen IP:4300 →** IP escoita no porto TCP 4300

```
root@kaliA:~# echo 'Listen 192.168.120.100:4300' >> /etc/apache2/ports.conf #Engadir a liña
Listen 192.168.120.100:4300 no ficheiro /etc/apache2/ports.conf para indicar que agora o servidor web Apache tamén
está a escoita no porto TCP 4300 na IP 192.168.120.100
root@kaliA:~# /etc/init.d/apache2 reload #Recargar a configuración do Servidor Web Apache
root@kaliA:~# nc -vz 192.168.120.100 4300 #Mediante o comando nc(netcat) comprobar se o port 4300
do servidor web Apache está en estado escoita(listen) na IP 192.168.120.100, esperando conexións. A opción -v
corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z
permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao porto solicitado. O número 4300 é o ports
TCP a escanear.
```

- d. **Listen IP:8443 http →** IP escoita no porto TCP 8443 atendendo o protocolo HTTP

```
root@kaliA:~# echo 'Listen 192.168.120.100:8443 http' >> /etc/apache2/ports.conf #Engadir
a liña Listen 192.168.120.100:8443 http no ficheiro /etc/apache2/ports.conf para indicar que agora o servidor web
Apache tamén está a escoita no porto TCP 8443 na IP 192.168.120.100 o protocolo http
root@kaliA:~# /etc/init.d/apache2 reload #Recargar a configuración do Servidor Web Apache
root@kaliA:~# nc -vz 192.168.120.100 8443 #Mediante o comando nc(netcat) comprobar se o port 8443
do servidor web Apache está en estado escoita(listen) na IP 192.168.120.100, esperando conexións. A opción -v
corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z
permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao porto solicitado. O número 8443 é o ports
TCP a escanear.
```

### 13. Exemplo2. Aloxa Cartafoles.

Na instalación defínese na directiva DocumentRoot o cartafol onde Apache aloxa as páxinas, sendo este: /var/www/html/, de tal xeito que incorporando ficheiros e cartafoles dentro desa ruta poderase acceder ao contido aloxado nos mesmos.

1. Acceder ao servidor web e crear/copiar varios ficheiros(cartafoles) en /var/www/html/  
root@kaliA:~# cat > /var/www/html/info.php <<EOF #Comezo do ficheiro a creari  
<?php Comezo do código PHP  
    phpinfo(); Función phpinfo(), a cal amosa información sobre a configuración de PHP  
?> Fin do código PHP  
EOF Fin do ficheiro a crear /var/www/html/info.php  
root@kaliA:~# cp -pv /var/www/html/index.html /var/www/html/index2.html #Copiar o ficheiro  
/var/www/html/index.html en /var/www/html/index2.html en modo verbose (detallado) e preservando permisos e  
datas.  
root@kaliA:~# chown -R www-data. /var/www/html/ #Cambiar usuario propietario www-data e grupo  
propietario www-data a toda a árbore de ficheiros e directorios que colgan do directorio DocumentRoot de Apache:  
/var/www/html  
root@kaliA:~# chmod 444 /var/www/html/info.php #Cambiar a só lectura os permisos **ugo** do ficheiro  
info.php situado en /var/www/html, é dicir, establecer os permisos r--r--r-- (soamente lectura para o usuario  
propietario, o grupo propietario e o resto do mundo)

2. Acceder dende calquer equipo cliente(kaliB) ás seguintes direccións web:

http://192.168.120.100/index2.html

http://192.168.120.100/info.php

**IMPORTANTE:** Como se pode observar as páxinas cargan sen ter que reiniciar o servidor:

```
root@kaliA:~# /etc/init.d/apache2 restart #Reiniciar o servidor web Apache.
```

Isto é debido a que o Servidor Web Apache está activo e sempre está a exportar o valor da directiva **DocumentRoot**. Polo tanto como **DocumentRoot** toma o valor **/var/www/html/** namentras o Servidor Web Apache estea activo todo o que aí se garda estará exposto polo servidor.

#### 14. Exemplo3. Xerar virtualhost: Virtualhost baseados en nome.

1. Engadir no directorio /etc/apache2/sites-available/ os seguintes bloques de configuración de virtualhosts. Cada bloque pertence a un arquivo .conf:

##### Arquivo empresa1.conf

```
#Configuración virtualhost: empresa1
<VirtualHost *:80>
DocumentRoot /var/www/empresa1/
ServerName www.empresa1.com
ServerAlias empresa1.com empresa1.es www.empresa1.es
</VirtualHost>
```

##### Arquivo empresa2.conf

```
#Configuración virtualhost: empresa2
<VirtualHost *:80>
DocumentRoot /var/www/empresa2/
ServerName www.empresa2.com
ServerAlias empresa2.com empresa2.es www.empresa2.es
</VirtualHost>
```

##### Explicación bloques configuración virtualhost:

- <VirtualHost \*:80> → Inicio etiqueta virtualhost.
- DocumentRoot /var/www/empresa1/ → Definición da ruta onde está aloxada a páxina web no servidor, neste caso: /var/www/empresa1/ mediante a directiva DocumentRoot.
- ServerName www.empresa1.com → Definición do nome DNS que buscará a páxina aloxada na ruta anterior do servidor mediante a directiva ServerName. É o nome que escribes no navegador para visitar a páxina.
- ServerAlias empresa1.com → A directiva ServerAlias permite definir outros nomes DNS para a mesma páxina.
- </VirtualHost> → Fin da etiqueta VirtualHost: fin da definición deste virtualhost para empresa1.

2. Xerar os directorios /var/www/empresa1 e /var/www/empresa2, os ficheiros index.html dentro deles e establecer permisos para que Apache poida acceder a eses ficheiros index.html.

```
root@kaliA:~# mkdir /var/www/empresa1 /var/www/empresa2 #Crear os directorios
/var/www/empresa1 e /var/www/empresa2
root@kaliA:~# echo 'empresa1 contido' > /var/www/empresa1/index.html Crear o ficheiro
/var/www/empresa1/index.html co contido: empresa1 contido
root@kaliA:~# echo 'empresa2 contido' > /var/www/empresa2/index.html Crear o ficheiro
/var/www/empresa2/index.html co contido: empresa2 contido
root@kaliA:~# chown -R www-data. /var/www/empresa1 /var/www/empresa2 #Cambiar
usuario propietario www-data e grupo propietario www-data a toda a árbore de ficheiros e directorios que colgan dos
directorios /var/www/empresa1 e /var/www/empresa2
```

3. Actualizar a configuración de Apache para ter en conta os novos cambios:

```
root@kaliA:~# a2ensite empresa1 Comando que permite habilitar a configuración do VirtualHost
empresa1, é dicir, comando que permite habilitar o ficheiro do VirtualHost empresa1 situado no directorio
/etc/apache2/sites-available/empresa1.conf engadindo a ligazón correspondente dende /etc/apache2/sites-
enabled/empresa1.conf a /etc/apache2/sites-available/empresa1.conf
root@kaliA:~# a2ensite empresa2 Comando que permite habilitar a configuración do VirtualHost
empresa2, é dicir, comando que permite habilitar o ficheiro do VirtualHost empresa2 situado no directorio
/etc/apache2/sites-available/empresa2.conf engadindo a ligazón correspondente dende /etc/apache2/sites-
enabled/empresa2.conf a /etc/apache2/sites-available/empresa2.conf
```

4. Acceder dende o equipo cliente kaliB ás seguintes direccións web:

```
http://192.168.120.100/empresa1/index.html
http://192.168.120.100/empresa2/index.html
```

NOTA: Como se pode observar agora as páxinas non cargan, porque o DocumentRoot non toma o valor /var/www/html senón os valores /var/www/empresa1 e /var/www/empresa2 para empresa1 e empresa2 respectivamente (configurados nos VirtualHost correspondentes en sites-available), e polo tanto, non están activos no servidor na raíz do VirtualHost por defecto (000-default.conf)

5. Actualizar o arquivo /etc/hosts no cliente kaliB:

```
root@kaliA:~# exit #Saír da consola remota ssh na que estamos a traballar, para voltar á consola local do
usuario kali na máquina kaliB.
kali@kaliB:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co
comando sudo (/etc/sudoers, visudo)
root@kaliB:~# echo '192.168.120.100 www.empresa1.com empresa1.com empresa1.es
www.empresa1.es' >> /etc/hosts #Engadir no ficheiro /etc/hosts, é dicir, na táboa estática de búsqueda
para nomes de host (DNS) os nomes www.empresa1.com, empresa1.com, empresa1.es e www.empresa1.es para
que atendan á IP 192.168.120.100
root@kaliB:~# echo '192.168.120.100 www.empresa2.com empresa2.com empresa2.es
www.empresa2.es' >> /etc/hosts #Engadir no ficheiro /etc/hosts, é dicir, na táboa estática de búsqueda
para nomes de host (DNS) os nomes www.empresa2.com, empresa2.com, empresa2.es e www.empresa2.es para
que atendan á IP 192.168.120.100
root@kaliB:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de
kali.
kali@kaliB:~$
```

6. Acceder de novo dende o equipo cliente kaliB ás seguintes direccións web:

```
http://www.empresa1.com/index.html
http://empresa1.com/index.html
http://empresa1.es/index.html
http://www.empresa1.es/index.html

http://www.empresa2.com/index.html
http://empresa2.com/index.html
http://empresa2.es/index.html
http://www.empresa2.es/index.html
```

**IMPORTANTE:** Como se pode observar agora as páxinas non cargan, porque aínda que actualizamos a configuración dos sitios(VirtualHost) do Servidor Web Apache, é necesario recargar o servidor para que se atenda á nova configuración realizada. Así, temos que recargar o servidor:

```
root@kaliA:~# /etc/init.d/apache2 reload #Recargar a configuración do Servidor Web Apache
```

7. Unha vez recargada a configuración do Servidor Web Apache acceder de novo dende o equipo cliente kaliB ás anteriores direccións web:

```
root@kaliA:~# /etc/init.d/apache2 reload #Recargar a configuración do Servidor Web Apache

http://www.empresa1.com/index.html
http://empresa1.com/index.html
http://empresa1.es/index.html
http://www.empresa1.es/index.html

http://www.empresa2.com/index.html
http://empresa2.com/index.html
http://empresa2.es/index.html
http://www.empresa2.es/index.html
```

**IMPORTANTE:** a2ensite + reload → Unha vez activado os sitios(a2ensite) e recargado(reload) o servidor as páxinas(VirtualHost) cargan.

## 15. Exemplo4. Control de acceso.

### A. Control de acceso por HTTP Basic

Na autenticación HTTP Basic é moi típico utilizar **arquivos .htaccess** nos directorios que queremos controlar o acceso. Os arquivos .htaccess son ficheiros de configuración do propio directorio onde exista.

Para usar arquivos .htaccess, necesítase ter unha configuración no servidor que permita poñer directivas de autenticación nestes arquivos, mediante a directiva AllowOverride, tal como segue: AllowOverride AuthConfig

**NOTA:** Visitar o seguinte enlace para ver unha explicación, máis polo miúdo, sobre á autenticación http basic: [Autenticación y autorización](#)



#### Procedemento:

1. Modificar arquivo **/etc/apache2/conf-available/security.conf** e engadir o seguinte bloque:

```
<Directory /var/www/html/auth-empresa>  
AllowOverride Authconfig  
</Directory>
```

2. Crear o contrasinal para o usuario nome\_usuario no ficheiro de contrasinais /etc/apache2/web.htpasswd:

```
root@kaliA:~# htpasswd -c /etc/apache2/web.htpasswd nome_usuario
```

3. Crear o directorio /var/www/html/auth-empresa:

```
root@kaliA:~# mkdir /var/www/html/auth-empresa
```

4. Configuralo servidor para o acceso sexa permitido mediante autenticación: usuario/contrasinal empregando un arquivo .htaccess:

```
root@kaliA:~# cat /var/www/html/auth-empresa/.htaccess Amosar contido arquivo .htaccesss  
  
AuthType Basic  
AuthName "Web con Autenticacion Basic"  
AuthBasicProvider file  
AuthUserFile /etc/apache2/web.htpasswd  
##Require valid-user  
Require user nome_usuario
```

5. Xerar dentro do directorio /var/www/html/auth-empresa o ficheiro secret.txt e establecer permisos para que Apache poida acceder a ese ficheiro secret.txt

```
root@kaliA:~# echo 'S3c3eT contido' > /var/www/html/auth-empresa/secret.txt Crear o ficheiro /var/www/html/auth-empresa/secret.txt co contido: S3cr3T contido  
root@kaliA:~# chown -R www-data. /var/www/html/auth-empresa #Cambiar usuario propietario www-data e grupo propietario www-data a toda a árbore de ficheiros e directorios que colgan do directorio /var/www/html/auth-empresa  
root@kaliA:~# chmod 400 /var/www/html/auth-empresa/.htaccess #Cambiar a só lectura os permisos ugo do ficheiro .htaccess situado en /var/www/html/auth-empresa, é dicir, establecer os permisos r----- \(soamente lectura para o usuario propietario\)
```

6. Actualizar a configuración de Apache para ter en conta os novos cambios:

```
root@kaliA:~# /etc/init.d/apache2 reload #Recargar a configuración do Servidor Web Apache
```



## 7. Actualizar o arquivo /etc/hosts no cliente kaliB:

```
root@kaliA:~# exit #Saír da consola remota ssh na que estamos a traballar, para voltar á consola local do usuario kali na máquina kaliB.  
kali@kaliB:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)  
root@kaliB:~# echo '192.168.120.100 auth-empresa.local' >> /etc/hosts #Engadir no ficheiro /etc/hosts, é dicir, na táboa estática de búsqueda para nomes de host (DNS) o nome auth-empresa.local para que atenda á IP 192.168.120.100  
root@kaliB:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de kali.  
kali@kaliB:~$
```

## 8. Acceder de novo dende o equipo cliente kaliB á seguinte dirección web:

<http://auth-empresa.local>

**IMPORTANTE:** Como estamos a empregar o sitio por defecto de Apache (DocumentRoot → /var/www/html), aínda que configuramos a autenticación, durante todo o proceso o arquivo secret.txt antes de recargar o servidor estivo visible e accesible. Mellor sería ter configurado isto mediante VirtualHost, tal que non se activaría o sitio ata que executáramos o comando a2ensite correspondente:

**a2ensite + reload → Unha vez activado o sitio(a2ensite) e recargado(reload) o servidor a páxina(VirtualHost) carga.**

## B. Control de acceso por IP (Order, Deny, Allow)



1. Tamén pódese controlar o acceso mediante IP. No seguinte exemplo IP\_permiso\_concedido define a IP que unicamente ten permiso de acceso. Copiamos este bloque ao arquivo **/etc/apache2/sites-available/controlIP.conf**

```
<VirtualHost *:80>
Alias /cartafol-controlado "/var/www/control/cartafol-controlado/"
<Directory "/var/www/control/cartafol-controlado/">
Order deny,allow
Deny from all
#Allow from IP_permiso_concedido
Allow from 192.168.120.101
</Directory>
DocumentRoot /var/www/control/cartafol-controlado/
ServerName www.empresa.local
ServerAlias empresa.local
</VirtualHost>
```

Actualmente as directivas: Order, Deny e Allow están en desuso e xa non son necesarias. Son substituídas pola directiva Require e o contenedor <RequireAll>

2. Crear o seguinte contido:

```
root@kaliA:~# mkdir -p /var/www/control/cartafol-controlado #Crear a estrutura arbórea de
directorios ata inclusive o directorio cartafol-compartido
root@kaliA:~# echo 'Contido control' > /var/www/control/cartafol-
controlado/control.txt #Crear o ficheiro control.txt no directorio anterior (/var/www/control/cartafol-
controlado → DocumentRoot)
```

3. Actualizar a configuración de Apache para ter en conta os novos cambios:

```
root@kaliA:~# a2ensite controlIP Comando que permite habilitar a configuración do VirtualHost
controlIP, é dicir, comando que permite habilitar o ficheiro do VirtualHost controlIP situado no directorio
/etc/apache2/sites-available/controlIP.conf engadindo a ligazón correspondente dende /etc/apache2/sites-
enabled/controlIP.conf a /etc/apache2/sites-available/controlIP.conf
root@kaliA:~# /etc/init.d/apache2 reload #Recargar a configuración do Servidor Web Apache
```

4. Lanzar no navegador da máquina virtual B (KaliB) unha nova lapela coa URL **http://192.168.120.100/control/cartafol-controlado/control.txt** Que acontece? Por que?  
Pois obtemos erro porque agora non se espera na IP ese cartafol senón un nome DNS por medio do VirtualHost xerado. Así deberíamos crear en /etc/hosts unha entrada como a seguinte:  
  
192.168.120.100 empresa.local www.empresa.local  
  
de tal xeito que se visitáramos **http://empresa.local** veríamos o esperado.
5. E se visitamos a URL **http://kaliA/control/cartafol-controlado/control.txt** ? Pois máis do mesmo, porque o nome DNS non se corresponde e polo tanto visitaríase o DocumentRoot de 000-default, e como /var/www/html/control/cartafol-controlado non existe, obteríamos erro.
6. E se visitamos a URL **http://empresa.local** ?  
Pois agora si que visitaríamos o esperado.
7. E se visitamos a URL **http://empresa.local/cartafol-controlado** ?  
Pois seguiríamos vendo o esperado, xa que existe un Alias definido no VirtualHost de xeito que somos redireccionados a /var/www/control/cartafol-controlado, visualizando o contido esperado.

### C. Control de acceso por IP (Require e <RequireAll>)



1. Imos facer de novo a opción B (controlar o acceso mediante IP) pero agora empregando a directiva aconsellada por Apache: **Require** (<RequireAll>). Así, modificamos o anterior bloque VirtualHost do arquivo **/etc/apache2/sites-available/controlIP.conf** tal como segue:

```
<VirtualHost *:80>
Alias /cartafol-controlado "/var/www/control/cartafol-controlado/"
<Directory "/var/www/control/cartafol-controlado/">
#Order deny,allow
#Deny from all
#Allow from 192.168.120.101
#Require ip IP_permiso_concedido
Require ip 192.168.120.101
</Directory>
DocumentRoot /var/www/control/cartafol-controlado/
ServerName www.empresa.local
ServerAlias empresa.local
</VirtualHost>
```

2. Actualizar a configuración de Apache para ter en conta os novos cambios:

```
root@kaliA:~# a2ensite controlIP Comando que permite habilitar a configuración do VirtualHost controlIP, é dicir, comando que permite habilitar o ficheiro do VirtualHost controlIP situado no directorio /etc/apache2/sites-available/controlIP.conf engadindo a ligazón correspondente dende /etc/apache2/sites-enabled/controlIP.conf a /etc/apache2/sites-available/controlIP.conf
root@kaliA:~# /etc/init.d/apache2 reload #Recargar a configuración do Servidor Web Apache
```

3. Lanzar no navegador da máquina virtual B (KaliB) unha nova lapela coa URL **<http://192.168.120.100/control/cartafol-controlado/control.txt>** Que acontece? Por que?

Pois obtemos erro porque agora non se espera na IP ese cartafol senón un nome DNS por medio do VirtualHost xerado. Así deberíamos ter en /etc/hosts unha entrada como a seguinte:

```
192.168.120.100 empresa.local www.empresa.local
```

de tal xeito que se visitáramos <http://empresa.local> veríamos o esperado.

4. E se visitamos a URL **<http://kaliA/control/cartafol-controlado/control.txt>** ? Pois máis do mesmo, porque o nome DNS non se corresponde e polo tanto visitaría o DocumentRoot de 000-default, e como /var/www/html/control/cartafol-controlado non existe, obteríamos erro.
5. E se visitamos a URL **<http://empresa.local>** ?  
Pois agora si que visitaríamos o esperado.
6. E se visitamos a URL **<http://empresa.local/cartafol-controlado>** ?  
Pois seguiríamos vendo o esperado, xa que existe un Alias definido no VirtualHost de xeito que somos redireccionados a /var/www/control/cartafol-controlado, visualizando o contido esperado.

## 16. Exemplo5. Prioridade seccións/directivas:

Á hora de configurar o servidor web Apache temos que ter en conta que é o que acontece cando unha mesma directiva pertence a distintas seccións. Cal é a directiva que se atende? Cal é a prioridade? Imos revisar a prioridade empregado a directiva Options co argumento Indexes:

- Options +Indexes → no caso de non existir un index.html permite ver o contido do cartafol visitado.
- Options -Indexes → no caso de non existir un index.html non permite ver o contido do cartafol visitado amosando o erro 403(Forbidden).

a. Crear o seguinte contido:

```
root@kaliA:~# mkdir /var/www/html/prioridade #Crear o directorio prioridade no
DocumentRoot(/var/www/html) do sitio por defecto que configura Apache (000-default.conf)
root@kaliA:~# echo 'Contido f1.txt' > /var/www/html/prioridade/f1.txt #Crear o ficheiro f1.txt no
directorio anterior (/var/www/html/prioridade → DocumentRoot)
```

b. Visitar <http://localhost/prioridade>

c. Modificar arquivo **/etc/apache2/conf-available/security.conf** e engadir o seguinte bloque:

### Sección Directory

```
<Directory /var/www/html/prioridade>
Options -Indexes
</Directory >
```

Recargar a configuración:

```
root@kaliA:~# /etc/init.d/apache2 reload #Recargar a configuración do Servidor Web Apache
```

d. Visitar de novo <http://localhost/prioridade>

Agora non se pode visualizar a páxina amosándose o erro 403 Forbidden

e. Modificar de novo o arquivo **/etc/apache2/conf-available/security.conf** e engadir o seguinte bloque:

### Sección Location

```
<Location /prioridade>
Options +Indexes
</Location >
```

f. Visitar de novo <http://localhost/prioridade>

Agora si se pode visualizar a páxina

Ricardo Feijoo Costa



This work is licensed under a **Creative Commons Attribution-ShareAlike 4.0 International License**