

Tema 2_a:

Configuración de un servidor Web

1º Parcial

Despliegue de Aplicaciones Web

Ciclo Superior de DAW

Tema2a: Configuración de un servidor Web

- **Objetivo:** *reconocer los parámetros de administración más importantes del servidor Web **Apache**, ampliando su funcionalidad mediante la activación y configuración de nuevos módulos, y creación de sitios virtuales*
- **Ficheros de configuración de Apache**
 - Consideraciones sobre los ficheros de configuración
 - Directivas:
 - En Linux
 - En Windows
- Instalar nuevos módulos
- Crear sitios virtuales
- Realizar pruebas de funcionamiento
- Documentar la configuración del servidor

■ Ficheros de configuración de Apache

■ La configuración de Apache se realiza a través de **archivos de texto** mediante diversas **directivas** que permiten escoger distintas opciones.

■ **directiva de configuración:**

- es una palabra que indica qué parámetro de configuración queremos modificar
- se pueden incluir en cualquiera de los ficheros de configuración
- para escribir una directiva, pondremos el nombre de la directiva seguido de sus parámetros
 - Ej: *DocumentRoot /var/www*
- son las mismas independientemente del sistema operativo, pero la distribución y localización de los archivos son diferentes
- en general, las directivas que incluimos en los ficheros de configuración global de Apache se aplican a todas las páginas del servidor
- las directivas que se pueden usar dependen de los módulos que se hayan habilitado

■ Consideraciones:

- Los ficheros de configuración contienen **una directiva por línea**.
- Puede emplearse una barra invertida \ como último carácter de una línea para indicar que la directiva continúa en la siguiente línea. No puede haber otros caracteres o espacios en blanco entre la barra invertida y el fin de línea.
- Los argumentos de las directivas se separan de éstas por medio de espacios en blanco. Si un argumento contiene espacios, debe ser escrito entre **comillas**.
- Las directivas **no son sensibles a mayúsculas** (case-insensitive), pero muchos de los argumentos sí
- Las directivas no especificadas utilizan su valor por defecto
- **Comentarios:** líneas que comienzan por #, son ignorados. Se pueden incluir en la misma línea que una directiva de configuración.
- Los espacios en blanco a principio de línea y las líneas en blanco, se ignoran –se usan para facilitar la lectura–
- Consultar directivas desde las páginas oficiales:
 - <http://httpd.apache.org/docs/2.4/mod/directives.html>
 - <http://httpd.apache.org/docs/2.4/mod/quickreference.html>

Ficheros y carpetas de configuración de Apache

- **apache2.conf**: fichero principal de configuración
- **conf.d**: directorio de configuración local, contiene ficheros y/o enlaces a ficheros con parámetros globales de configuración generados al instalar *paquetes*
- **envvars**: fichero que almacena variables de entorno del servidor
- **ports.conf**: almacena las directivas que determinan por qué puertos TCP Apache está a la escucha
- **mods-available**: carpeta que contiene todos los módulos instalados y disponibles para activar, con los ficheros para configurar dichos módulos
- **mods-enabled**: con los módulos activos; contiene enlaces simbólicos a los ficheros activos contenidos en **/mods-available**. Cuando un fichero de configuración de módulo está enlazado, éste se carga al iniciar el servidor
- **sites-available**: contiene los ficheros de configuración de *hosts virtuales*, éstos permiten a Apache configurar múltiples sitios, cada uno con su configuración. Contiene el fichero **default** con la configuración del servidor virtual por defecto
- **sites-enabled**: contiene enlaces a ficheros de **/sites-available**. Cuando un fichero de configuración está enlazado, el sitio será activado al iniciar Apache. Contiene el fichero **000-default.conf** que es un enlace al fichero **default**

■ Fichero principal de configuración de Apache: apache2.conf

- Está dividido en tres secciones:
 - **configuración global:** incluye directivas que especifican el funcionamiento del servidor, indican el directorio de los ficheros de configuración, el modo de funcionamiento, etc
 - Ej: ServerRoot /etc/apache2
 - **configuración general del servidor:** incluye directivas que definen el comportamiento del servidor por defecto y de todos los servidores virtuales, excepto en aquellos en los que se definan otras opciones
 - Ej: ServerAdmin webmaster@localhost
 - **configuración de los servidores virtuales:** incluye directivas que definen el comportamiento de un servidor virtual
- para especificar una directiva que se aplique sobre unos determinados documentos o carpetas, podemos utilizar **secciones** de configuración

Ficheros de configuración

Linux (Debian/Ubuntu)

```
# Formato general apache2.conf

# Aquí empieza la Sección 1 (directivas de configuración global)

... Directivas globales

# Aquí empezaría la sección 2 (directivas de funcionamiento del
# servidor principal )
... Directivas de funcionamiento del servidor principal (se heredan en
los servidores virtuales)
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}

include /etc/apache2/mods-enabled/*.load
include /etc/apache2/mods-enabled/*.conf
include /etc/apache2/httpd.conf
include /etc/apache2/ports.conf

.. Directivas de logs y errores
include /etc/apache2/conf.d/

# Aquí empezaría la sección 3 (Servidores virtuales)
include /etc/apache2/sites-enabled/
```

Ficheros y carpetas de configuración de Apache

1. Linux (Debian)

- La carpeta donde se encuentran los ficheros y carpetas de configuración es **/etc/apache2/**
- El fichero principal de configuración de Apache es **apache2.conf**: **/etc/apache2/apache2.conf**, en el están incluidos los demás archivos de configuración:
- /etc/apache2/ports.conf**: configuración de los puertos de escucha
- /etc/apache2/sites-available/000-default.conf**: configuración del servidor virtual por defecto

2. Windows

- La carpeta donde se encuentran los ficheros de configuración es **Apache24\conf**
- El fichero principal de configuración de Apache es **httpd.conf**: **C:\Apache24\conf\httpd.conf**

Lanzar y parar Apache

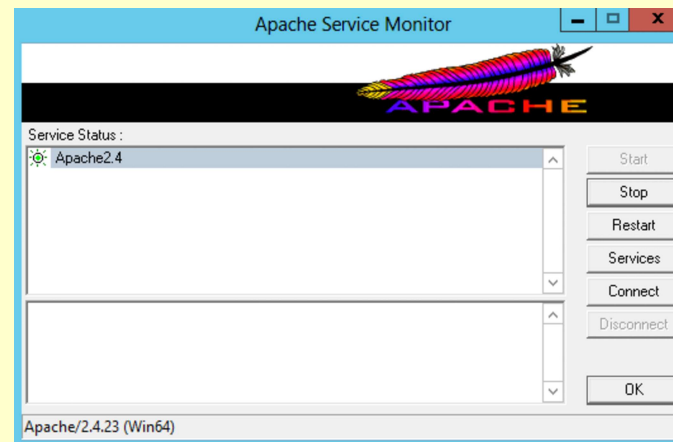
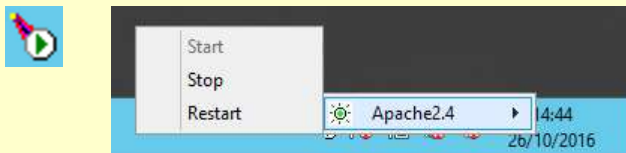
Cada vez que se modifican los ficheros de configuración, para que los cambios tengan lugar, es preciso que se **reinicie** el servidor

1. Linux (Debian)

- Para parar el servicio, se emplea el comando **stop**: `sudo service apache2 stop`
- Para lanzar el servicio, se emplea el comando **start**: `sudo service apache2 start`
- Para reiniciar el servicio, se emplea el comando **restart**: `sudo service apache2 restart`

2. Windows

- Podemos iniciar, parar o reiniciar el servicio desde la interface gráfica, ya sea desde el icono que aparece en el área de notificaciones como desde el propio monitor de Apache



- Directivas de configuración básicas

1. Listen:

- indica a través de qué **puertos** e interfaces IP aceptará peticiones
- por defecto, responde peticiones en todas las interfaces en el puerto que se indique
- **Ejemplo1:** para hacer que el servidor acepte conexiones en los puertos 80 y 8080
 - *Listen 80*
 - *Listen 8080*
- **Ejemplo2:** para atender a dos direcciones IP distintas con puertos diferentes
 - *Listen 192.168.255.5:80*
 - *Listen 192.168.255.8:8080*

- Directivas de configuración básicas

2. ServerRoot:

- **directorio de instalación:** especifica la localización del directorio raíz donde se encuentra instalado Apache, con los ficheros de configuración, error y logs
- esta directiva **no** se debe cambiar, a no ser que se mueva la carpeta de instalación de Apache a otra localización
- **Ejemplo1:**
 - *ServerRoot /etc/apache2*
- **Ejemplo2:**
 - *ServerRoot "C:/xampp/apache"*

- Directivas de configuración básicas

3. Include:

- permite que se incluyan otros ficheros de configuración en tiempo de ejecución
- la ruta a estos ficheros puede ser absoluta o relativa con respecto al directorio indicado en *ServerRoot*
- Esta directiva está en *apache2.conf*
- **Ejemplo:**
 - *Include conf/extra/httpd-languages.conf*
- **Realiza:** Busca alguna línea *include* en algún fichero de configuración

4. DocumentRoot:

- indica el directorio más alto desde el que Apache va a servir los archivos, el servidor añade la ruta indicada en la URL a este directorio
- todos los directorios que va servir Apache deben tener permisos de lectura y ejecución para todos los usuarios, y todos los archivos que sirve deben tener permiso de lectura
 - en Linux, los permisos de archivos y directorios se cambian con el comando **chmod**
- **Ejemplo1:** *DocumentRoot /var/www/html*
 - Si la URL solicitada es 192.180.0.4/index.html Apache servirá el fichero index.html que se encuentra en */var/www/html*
 - Si la URL solicitada es <http://www.miservidor.com/prueba/index.html> Apache servirá el fichero index.html que se encuentra en */var/www/html/prueba*
 - Si la URL solicitada es 192.180.0.4/prueba/index.html Apache servirá el fichero index.html que se encuentra en */var/www/html/prueba*

4. DocumentRoot: (continuación)

- **Ejemplo2:** *DocumentRoot "C:/Apache24/htdocs"*
 - Si la URL solicitada es <http://www.miservidor.com/prueba/index.html> Apache servirá el fichero index.html que se encuentra en */htdocs/prueba*
- **Averigua:** ¿en qué fichero se encuentra esta directiva en Windows?

- Directivas de configuración básicas

5. ErrorLog:

- especifica la localización del fichero que contiene los mensajes de error del servidor
- si la ruta que se indica no es absoluta, se considera relativa al *ServerRoot*
- por defecto, se encuentran en la carpeta **logs** dentro de *ServerRoot*
- **Ejemplo1:** *ErrorLog /var/log/apache2/error_log*
- **Ejemplo2:** *ErrorLog "logs/error.log"*

- Directivas de configuración básicas

5. DirectoryIndex:

- especifica el fichero que se servirá para cada directorio en el caso de que no se especifique ninguno en la URL. Por defecto es **index.html**
- se pueden indicar varios ficheros
- el orden con el que se especifica el nombre de fichero, determinará la prioridad a la hora de decidir que fichero es el que se muestra
- **Ejemplo:**
- *DirectoryIndex **index.php** index.pl index.cgi index.asp **index.html** index.htm \ default.php default.pl default.cgi default.asp default.shtml default.html \ default.htm home.php home.pl home.cgi home.asp home.html*
- *Ejemplo de llamada desde el navegador: 192.168.0.4 => me abre el primer archivo que encuentra que está reflejado DirectoryIndex*
- **Averigua:** en la nueva versión de Apache 2.4 han creado un fichero específico para esta directiva. ¿en qué carpeta se guarda ahora y cómo se llama dicho fichero? ¿en qué fichero se guardaba antes?

- Directivas de configuración básicas

7. ErrorDocument:

- explica qué debe hacerse en el caso de que se produzca un error
- si no se indica nada, se mostrará un mensaje por defecto
- si se indica, podrá mostrarse un texto personalizado o una página web personalizada dentro o fuera de nuestro servidor
- **Ejemplo:**
 - *ErrorDocument 500 "Este fichero no está"*
 - *ErrorDocument 404 errores/error404.html*
 - *ErrorDocument 402 http://www.otroservidor.com/error404externo.html*

- Directivas de configuración básicas

8. Alias:

- permite la definición de **directorios virtuales**:
 - *Directorio que se encuentra en una localización distinta de la que se indica en la URL.*
- de esta forma utilizaremos un nombre alternativo para llegar a un *sitio*, ocultando la ubicación real de la dirección donde se encuentra físicamente
- No tiene por qué encontrarse dentro del árbol de directorios que se crea a partir de *DocumentRoot*, sino que se puede encontrar en cualquier otra localización
 - **DocumentRoot**: para indicar la ruta predeterminada, la de */var/www/html*.
- También permite acortar rutas
- Para que funcione correctamente, es necesario asignar permisos a la carpeta de la ubicación real, con la directiva **<Directory>**
- **Sintaxis**: Alias nombreFicticio ubicaciónReal
- **Ejemplo**: Alias */ciclos/daw* */home/usuario/daw*

- Directivas de configuración básicas

9. Redirect:

- permite redireccionar los ficheros que existían en el espacio de nombres del servidor y han cambiado de ubicación
- tiene preferencia con respecto a la directiva *Alias*, independientemente del orden en el que se sitúen en el fichero de configuración
- **Ejemplo1:** *Redirect /service http://otrodirectorio.ejemplo.com/service*
- **Ejemplo2:** *Redirect permanent /foo http://www.example.com/bar*
- **Nota:** en caso de que dé algún problema, utilizar únicamente *Alias*

- Directivas de configuración básicas

10. Options:

- controla qué características del servidor están disponibles para un determinado directorio
- las características se hacen disponibles si se listan, y **no** disponibles si se preceden por un signo “–”
- algunas **opciones** son:
 - ***Indexes***: si se solicita una URL que apunta a un directorio y en dicho directorio no se encuentra ninguno de los ficheros indicados por *DirectoryIndex*, se mostrará una lista de los archivos que contiene dicho directorio
 - ***FollowSymLinks***: el servidor seguirá enlaces simbólicos en este directorio
- **Ejemplo:** *Options Indexes –FollowSymLinks*

■ Etiquetas de configuración por secciones

- para especificar una directiva que se aplique sobre unos determinados documentos o carpetas, podemos utilizar **secciones de configuración**, y de esa forma afectan únicamente a ese bloque
- para definir una sección de configuración lo haremos usando una sintaxis parecida a la de HTML, empezando la sección con la directiva de sección entre menor y mayor que <directiva argumento1>, y terminando con el nombre de la misma directiva también entre menor y mayor que pero con una barra delante </directiva>

■ Tipos de bloques :

1. Directory

- Las directivas que se encuentren dentro de una etiqueta <**Directory**> sólo se aplican a ese directorio, a sus subdirectorios y a los contenidos
- El directorio debe indicarse con la **ruta absoluta**
- **Ejemplo:**

```
<Directory /var/www/html/ciclos>
```

```
#Aquí las directivas que sólo se aplicarán a los documentos contenidos en  
/var/www/html/ciclos
```

```
Options Indexes FollowSymLinks
```

```
</Directory>
```

Etiquetas de configuración por secciones

2. Files

- Las directivas que se encuentren dentro de una etiqueta **<Files>** sólo se aplican a al fichero que se indicada, independientemente del directorio en el que esté

- Ejemplo:**

```
<Files "privado.html">  
    #Aquí las directivas  
</Files>
```

3. Location

- Las directivas que se encuentren dentro de una etiqueta **<Location>** sólo se aplican en el ámbito de la URL indicada en la etiqueta

- Ejemplo:**

```
<Location /privado1>  
    #Aquí las directivas  
</Location /privado1>
```

- Amplía:** consulta en la documentación oficial de Apache la definición de las diferentes directivas <https://httpd.apache.org/docs/2.4/mod/core.html#page-header>

- **Tarea1_a: consultar ficheros de configuración de Apache en Linux**
- En esta actividad tienes que consultar los ficheros de configuración y observar los valores de ciertas directivas. Inicia sesión con un usuario que tenga privilegios de administración en el servidor de Linux y abre un terminal accediendo al directorio **/etc/apache2**
 1. Haz un listado del directorio y observa los ficheros de configuración que contiene.
 2. Abre el fichero **/etc/apache2/apache2.conf** y analiza su configuración.
 - Observa si aparece la directiva *Include*. ¿A qué ficheros y directorios incluye?
 - Cuál es el valor de las siguientes directivas:
 - *ServerRoot*
 - *User* y *Group*
 - *TimeOut*

■ **Tarea1_a: consultar ficheros de configuración de Apache en Linux** *(continuación)*

3. Consulta la documentación oficial de Apache en <httpd.apache.org/docs/2.4> y responde:
 - ¿Se permiten las conexiones persistentes, es decir, que todas las conexiones de un usuario se atienden en la misma conexión TCP? ¿Qué directiva define este comportamiento?
 - ¿Cuál es el fichero de errores? ¿Qué directiva lo define?
4. Consulta el fichero **/etc/apache2/ports.conf** y comprueba cuál es el puerto en el que Apache escucha las peticiones. ¿En qué puerto escuchará también si se habilita el módulo *modssl*?

Tarea1_b: Configuración básica de Apache en Linux

- En esta actividad tienes que modificar los ficheros de configuración para cambiar los archivos que se sirven por defecto, el puerto de escucha del servidor y el directorio en el que se almacenan los ficheros que se van a servir.
- 1. Realiza copias de seguridad –**cp**- de los ficheros de configuración (**apache2.conf**, **ports.conf**, **sites-available/000-default.conf**).
- 2. En el directorio **/var/www/html**, crea los siguientes archivos –**gedit**- con el contenido que prefieras, el objetivo es poder distinguir unos archivos de otros, y directorios –**mkdir**-. Comprobar que los permisos de los ficheros son 644 y del directorio 755 –**ls -l**-
 - index.html => como ya existe, crea una copia
 - indice.html
 - personal/alumnos.html
 - personal/profesores.html
- 3. Abre el navegador de la máquina cliente y establece conexiones con cada uno de los archivos (<http://192.168.0.1/nombreArchivo>). ¿Se muestra correctamente el contenido?. Prueba también a acceder a la IP del servidor sin indicar ningún archivo (<http://192.168.0.1/>).

Tarea1_b: Configuración básica de Apache en Linux (*continuación*)

4. Haz las modificaciones adecuadas para que se sirva por defecto el archivo **indice.html**
5. Haz las modificaciones precisas para que en el directorio **personal** no se muestre el listado de archivos cuando se pide la raíz del mismo
6. Modifica el mensaje de error para el **error 404**
7. Crea una redirección, de forma que cuando se haga una petición a <http://192.168.0.1/portal>, se redirija a la página <http://edu.xunta.es>
8. Modifica el puerto en el que escucha el servidor Apache al puerto **8081**
9. Crea el directorio **/home/administrador/apuntes**, que contendrá un fichero cualquiera. Crea un alias para que este directorio sea accesible desde <http://192.168.0.1/apuntes>. Como aún no hemos visto las directivas de acceso, copia la configuración para el directorio **/var/www** del archivo **apache2.conf** para este nuevo directorio

Tarea1_b: Configuración básica de Apache en Linux (*continuación*)

10. Modifica el directorio desde el que Apache va a servir los archivos a **/home/administrador/web**, incluye en ese directorio un archivo para que puedas comprobar que, efectivamente, son esos los ficheros que se sirven
 - **Documenta** cada uno de los pasos anteriores, incluyendo breves explicaciones y pantallazos
 - **No olvides restaurar los ficheros** a su estado original (sobre todo respecto al directorio en el que se encuentran los archivos que va a servir Apache y el puerto de escucha)

- **Tarea1_c: Configuración básica de Apache en Windows**
- En esta actividad tienes que modificar los ficheros de configuración para cambiar los archivos que se sirven por defecto, el puerto de escucha del servidor y el directorio en el que se almacenan los ficheros que se van a servir
- **Enunciado**
 1. Realiza copias de seguridad del fichero de configuración (**httpd.conf**). Lee el contenido completo para ver cómo se colocan las diferentes directivas
 2. En el directorio **c:\Apache24\htdocs**, crea los siguientes archivos -con el contenido que prefieras, el objetivo es poder distinguir unos archivos de otros- y directorios:
 - index.html
 - indice.html
 - personal/alumnos.html
 - personal/profesores.html
 3. Abre el navegador de la máquina cliente y establece conexiones con cada uno de los archivos (<http://192.168.0.2/nombreArchivo>). Prueba también a acceder a la IP del servidor sin indicar ningún archivo (<http://192.168.0.2/>)

Tarea1_c: Configuración básica de Apache en Windows (*continuación*)

4. Haz las modificaciones adecuadas para que se sirva por defecto el archivo **indice.html**
5. Haz las modificaciones precisas para que en el directorio **personal** no se muestre el listado de archivos cuando se pide la raíz del mismo
6. Modifica el mensaje de error para el **error 404**
7. Crea el directorio **C:\Users\Administrador\apuntes**, que contendrá un fichero cualquiera. Crea un alias para que este directorio sea accesible desde <http://192.168.0.2/apuntes>
8. Crea una redirección, de forma que cuando se haga una petición a <http://192.168.0.2/portal>, se redirija a la página <http://edu.xunta.es>
9. Modifica el puerto en el que escucha el servidor Apache al puerto **8080**

Tarea1_c: Configuración básica de Apache en Windows (*continuación*)

10. Modifica el directorio desde el que Apache va a servir los archivos a **C:\Users\Administrador\web**, incluye en ese directorio un archivo para que puedas comprobar que, efectivamente, son esos los ficheros que se sirven
 - Documenta cada uno de los pasos anteriores, incluyendo breves explicaciones y pantallazos
 - No olvides restaurar los ficheros a su estado original (sobre todo respecto al directorio en el que se encuentran los archivos que va a servir Apache y el puerto de escucha)

Módulos de Apache

- El diseño de Apache es modular => permite personalizar y configurar de forma independiente cada uno de sus componentes
- El núcleo de Apache (**core server**) proporciona la funcionalidad básica del servidor web, y es la única que se incluye en el servidor principal
- Las funcionalidades adicionales se pueden añadir gracias a la **carga de módulos** que también se pueden incluir en la instalación inicial de Apache
- Consultar módulos desde las páginas oficiales:
 - <https://modules.apache.org/>
 - <http://httpd.apache.org/docs/2.4/mod/>
- Cada módulo tiene un conjunto de **directivas específicas** que permiten su gestión
- En **Linux**:
 - **/etc/apache2/mods-available**: módulos instalados y disponibles para activar. Contiene los ficheros **.load** –para cargar un módulo- y los **.conf** -configuración básica para iniciar el módulo
 - **/etc/apache2/mods-enabled**: módulos habilitados (se cargan al iniciar Apache)

Módulos de Apache

■ Cómo añadir módulos:

- **Estáticamente:** se añaden cuando se compila Apache
- **Dinámicamente:** en el momento de lanzar el servidor, se cargan junto aquellos módulos que implementen la configuración que necesitemos. Los módulos cargados se gestionan en la configuración del servidor. Estos módulos se denominan *objetos dinámicos compartidos* o **módulos DSO** -Dynamic Shared Object –
 - Hay que compilar el servidor con la opción DSO
 - **Ventajas:**
 - Servidor más flexible
 - Más sencillo el prototipado y desarrollo de módulos
 - **Desventajas:**
 - Servidor más lento en el arranque
 - Servidor más lento en funcionamiento

Módulos de Apache

Directivas de módulos:

a) LoadModule

- Permite cargar módulos dinámicos
- **Sintaxis:** `LoadModule nombre_módulo ruta_módulo`
- **Ej:** abrir el fichero **dir.load** y observa cómo se usa la directiva

```
LoadModule dir_module /usr/lib/apache2/modules/mod_dir.so
```

b) <IfModule nombre_modulo> ... </IfModule>

- Permite especificar directivas que se tendrán en cuenta si el módulo está cargado al arrancar o reiniciar el servidor
- **Ej:** abre el fichero **dir.conf** y observa cómo se añaden directivas a esta sección

```
<IfModule mod_dir.c>  
    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml  
</IfModule>
```

■ Consultar Módulos

- Podemos utilizar diferentes comandos para consultar los módulos que están instalados, tanto los estáticos como los dinámicos
- **Estáticos:**
 - Ubuntu: **sudo apache2ctl -l**
 - Windows: **C:\Apache24\bin\httpd.exe -l**
- **Dinámicos:**
 - Ubuntu: mostrar con **ls** el contenido de la carpeta **/etc/apache/mods-enabled** (donde están los módulos activos)
 - Windows: consultar las directivas **LoadModule** del fichero de configuración **httpd.conf**

■ Consultar Módulos

- **Dinámicos disponibles:** (en los repositorios de Apache para Ubuntu)
 - Ubuntu:
 - mostrar con **ls** el contenido de la carpeta **/usr/lib/apache/modules** para los que ya están **instalados** y disponibles para cargar
 - para mostrar los paquetes disponibles en los **repositorios** de Ubuntu que permiten **instalar módulos adicionales** en Apache, es decir, no están instalados, **para buscar:** **sudo apt-cache search libapache2-mod**
 - Windows: consultar la carpeta **C:\Apache24\modules**

■ Instalación de Módulos

- Ubuntu: **sudo apt-get install libapache2 nombre_módulo** y tras realizar la instalación, aparecerá en la carpeta **mods-available**
 - **Ej:** **sudo apt-get install libapache2-mod-auth-mysql**
(para instalar el módulo de autenticación en MySQL)

Habilitar Módulos

- Ubuntu: **sudo a2enmod nombre_módulo** => y se creará el enlace simbólico en **mods_enabled**
- Windows: desde el fichero de configuración **httpd.conf** eliminar los comentarios de las directivas **LoadModule** e **Include** correspondientes a ese módulo

Deshabilitar Módulos

- Ubuntu: **sudo a2dismod nombre_módulo** => y se borrará el enlace simbólico de **mods_enabled**
- Windows: desde el fichero de configuración **httpd.conf** **comentar** las directivas **LoadModule** e **Include** correspondientes a ese módulo

Añadir directivas a cada Módulo

- Ubuntu: en el fichero de extensión **.conf** correspondiente al módulo, que se encuentra en el directorio **/etc/apache2/mods-available**
- Windows: en el fichero de extensión **.conf** correspondiente al módulo, que se encuentra en el directorio **C:\Apache24\conf\extra**
- Desde cualquiera de los ficheros de configuración, dentro de la sección **<IfModule nombre_módulo></IfModule>**

- **Tarea2_a: consultar módulos de Apache en Linux**
- Revisa los apuntes y contesta las siguientes cuestiones:
 1. Comprueba los módulos estáticos que se han cargado al compilar el servidor
 2. Comprueba los módulos que se han cargado dinámicamente al arrancar el servidor
 3. Comprueba los módulos dinámicos disponibles instalados
 4. Comprueba qué módulos adicionales se pueden instalar
 5. Comprueba específicamente si aparecen el de **php**, el **jk** y el de **userdir** (**|grep**)
 6. Comprueba los módulos ya instalados y disponibles para cargar
 7. Comprueba los enlaces que se han creado de los módulos activos (**ls -l**)

Documenta cada uno de los pasos anteriores, incluyendo breves explicaciones y pantallazos de las diferentes carpetas que has consultado

- **Tarea2_b: módulo UserDir en Linux**

- Investiga la funcionalidad del módulo **userdir**, habilítalo y comprueba cómo funciona realizando la siguiente configuración:
 1. Comprueba que el módulo **userdir** no está habilitado
 2. Habilita el módulo **userdir** y comprueba que está activo, indicando qué enlaces se han creado
 3. Muestra el contenido del archivo de configuración del módulo e interpreta las directivas propias de este módulo
 4. Crea un nuevo usuario (`sudo adduser nombreusu`) y crea un fich html y comprueba q se puede entrar (nuevo usu+ crear `public_html`+ fich.html)
 5. Crea algún fichero en el directorio correspondiente para el usuario *administrador* y comprueba que se puede acceder desde la máquina cliente
 6. Deshabilita el módulo **userdir** y verifica que ya no esté activo

Documenta cada uno de los pasos anteriores, incluyendo breves explicaciones y pantallazos de cada paso realizado

- Sol Tarea2_b: habilitar el módulo UserDir de Apache en Linux
- Módulo UserDir
 - Este módulo hace posible que todos los usuarios con acceso al servidor tengan una carpeta propia en la que pueden alojar sus páginas y archivos dentro de su carpeta personal, y estén disponibles desde Apache
 - Para acceder a estas páginas desde el navegador hay que añadir a la dirección URL del servidor el usuario:
 - Sintaxis: http://nombre_servidor/~nombre_usuario
 - Fichero de configuración de *userdir*:
 - Ubuntu: `/etc/apache2/mods-available/userdir.conf`
 - Windows: `C:\Apache24\conf\extra\httpd-userdir.conf`
 - Carpeta predeterminada:
 - Ubuntu: `/home/usuario/public_html` (esa carpeta hay q crearla para cada usu)
 - Windows: `C:\Users\usuario\My Documents\My Website`
(en la máquina virtual de Windows Server 2012, no se llama My Documents, sino **Documents**, por lo que hay que cambiarlo en el fichero de configuración)

■ **Tarea2_c: consultar módulos de Apache y habilitar *userdir* en Windows**

1. Comprueba los módulos estáticos que se han cargado al compilar el servidor
2. Comprueba los módulos que se han cargado dinámicamente al arrancar el servidor
3. Comprueba los módulos dinámicos disponibles instalados
4. Comprueba que el módulo **userdir** no está habilitado
5. Habilita el módulo **userdir** y comprueba que está activo, indicando qué enlaces se han creado
6. Muestra el contenido del archivo de configuración del módulo e interpreta las directivas propias de este módulo
7. Crea algún fichero en el directorio correspondiente para el usuario *administrador* y comprueba que se puede acceder desde la máquina cliente.
8. Deshabilita el módulo **userdir** y verifica que ya no esté activo

Documenta cada uno de los pasos anteriores, incluyendo breves explicaciones y pantallazos

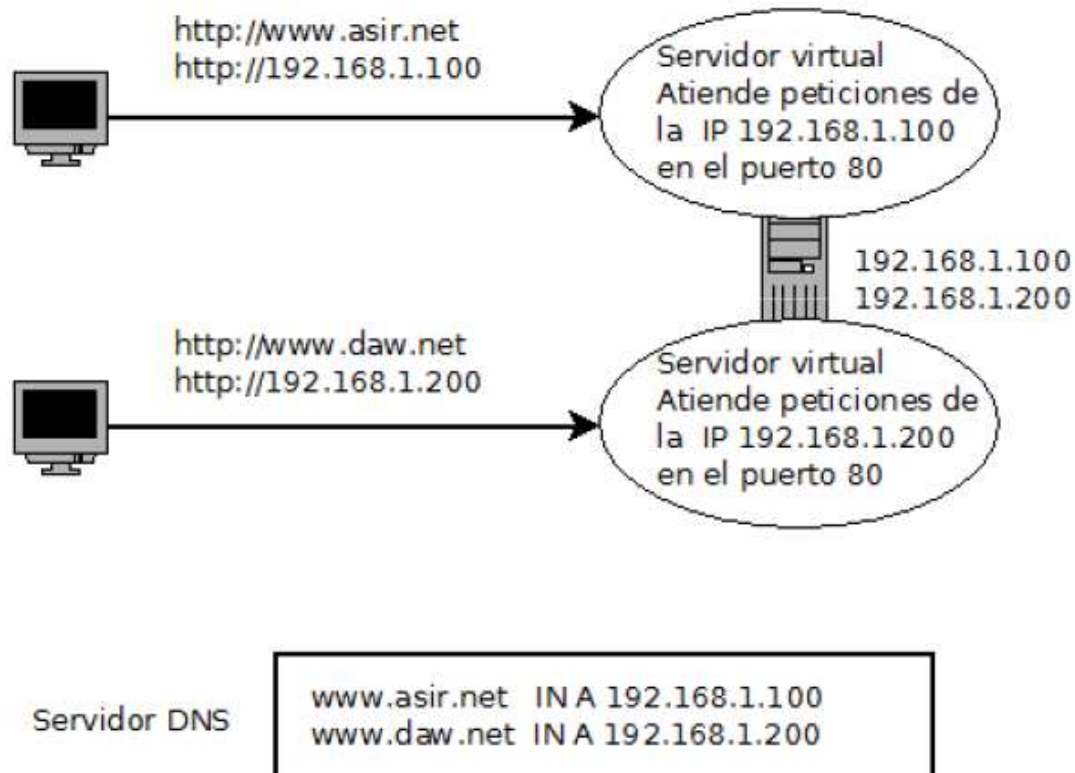
- Directivas de configuración básicas

- Alojamiento compartido/virtual: *Virtual Hosting*

- Es la unidad básica que describe a un **sitio** o **dominio**
- El término *virtual hosting* hace referencia a la práctica de **alojar más de un sitio web en una misma máquina**, simulando que existen varias máquinas –*hosts*- con sus respectivos sitios web sobre un único servidor web
- Cada **dominio** configurado enviará al visitante a un directorio específico manteniendo la información del mismo. A simple vista, nunca indicará que el mismo servidor es también responsable de otros sitios
- Este esquema es flexible sin limitación de software -siempre y cuando el servidor pueda manejar la carga-
- En este apartado, veremos cómo se puede servir contenido diferente a distintos visitantes dependiendo del dominio que estén solicitando
- **Ej:** crear dos dominios en un mismo servidor
 - *uno.midominio.es*
 - *dos.otrodominio.com*

- Directivas de configuración básicas
- Alojamiento compartido/virtual: *Virtual Hosting*
 - **Objetivos:**
 - Reducir el número de máquinas físicas
 - Aprovechar mejor los recursos de cada equipo (uso de CPU, memoria, ...)
 - Aprovechar mejor las direcciones IP disponibles
 - Apache admite **diferentes configuraciones** de servidores virtuales:
 - a) basados en **dirección IPs**: cada sitio tiene su propia dirección IP
 - b) basados en **nombres**: con varias direcciones simbólicas para una misma dirección IP
 - c) basados en **puertos**: cambia los puertos de escucha combinando con opciones anteriores

■ Alojamiento compartido basado en IP's

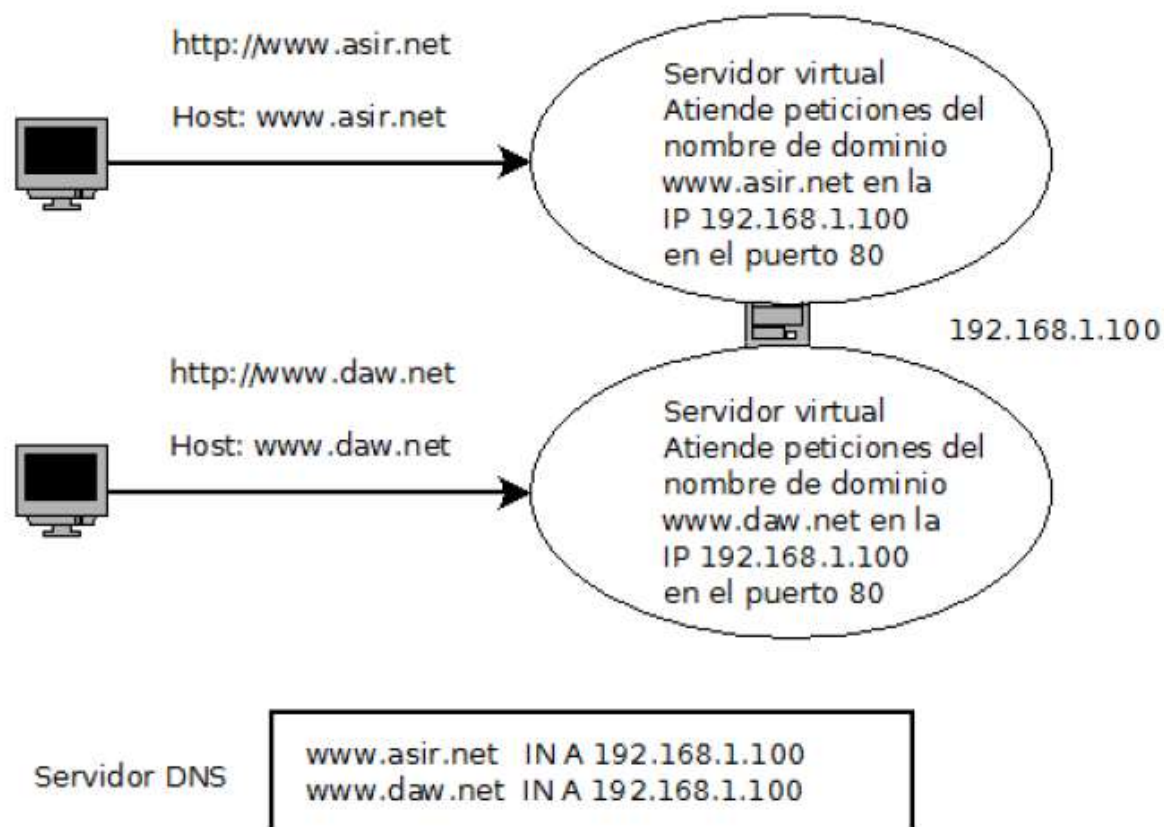


- El servidor tendrá diferentes direcciones IP por cada servidor web virtual.

- Cada servidor virtual atenderá peticiones en una dirección IP diferente.

- A efectos de los usuarios es como si existiesen varios servidores web, uno en cada dirección IP.

Alojamiento compartido basado en nombres

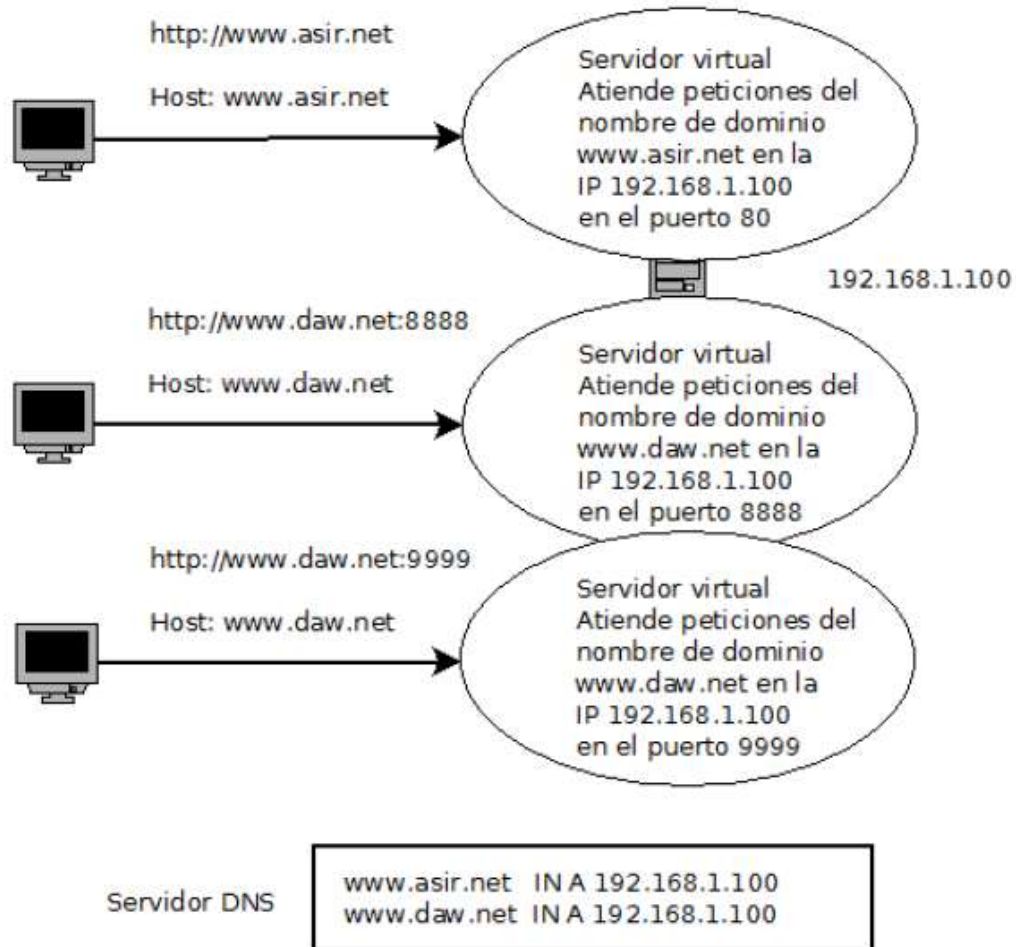


- El servidor permite alojar varios nombres de dominio sobre la misma dirección IP.

- Cada servidor virtual atiende las peticiones de un nombre de dominio.

- Hay que configurar un servidor DNS que asocie los nombres de dominio con la misma dirección IP.

Alojamiento compartido basado en puertos



- Cada servidor virtual atiende peticiones en una **dirección IP** y/o **dominio:puerto** diferentes.

- Consiste en combinar el alojamiento basado en IP y/o en nombres con el uso de varios puertos a las escucha.

- Directivas de configuración básicas

- Alojamiento compartido basado en nombre

- Es la forma de alojamiento virtual más utilizada
- Permite alojar múltiples dominios en una única máquina
- El servidor confía en el cliente para que le indique el nombre del host como parte de las cabeceras HTTP
- Normalmente, es más sencillo que el basado en IP, ya que únicamente son necesarios dos pasos:
 1. Configurar el **servidor DNS** para que relacione la dirección simbólica con la IP adecuada (o modificar el fichero **hosts** del cliente en **/etc** indicando **ip dirección**; en Windows se ubica en **C:\Windows\System32\Drivers\etc**)
 - Ej: 192.168.0.4 uno.midominio.es
 2. Configurar Apache para que reconozca las diferentes direcciones simbólicas
- **Ventajas:**
 - Ahorro de equipos
 - Reduce el problema de la escasez de direcciones IP (ahorro de IP's)
 - Facilita la administración centralizada

■ Alojamiento compartido: *Virtual Hosting*

- Los archivos **virtual host** son archivos que especifican la configuración actual de un virtual host e indican cómo el servidor Apache va a responder a varias solicitudes de dominio
- La versión precompilada (.deb) para Debian/Ubuntu tiene habilitados servidores virtuales => tiene creado y habilitado un **servidor virtual por defecto (000-default.conf)**
- La versión de Windows no tiene habilitados por defecto los servidores (usa el servidor principal)
- **Configuración en Linux**
 - **/etc/apache2/sites-available:** directorio que contiene los ficheros de configuración de servidores virtuales disponibles
 - por defecto están creados el fichero **000-default.conf** con la configuración del servidor virtual por defecto y el fichero **default-ssl.conf**, con la configuración de un servidor por defecto seguro

```
root@base:/etc/apache2/sites-available# ls
000-default.conf  default-ssl.conf
```

- Directivas de configuración básicas
- Alojamiento compartido: *Virtual Hosting*
- Configuración en Linux
 - **/etc/apache2/sites-enabled**: directorio que contiene enlaces simbólicos a los ficheros de sites-available
 - Los enlaces serán los servidores virtuales habilitados
 - Se incluyen en **apache2.conf** en orden alfabético
 - Por defecto está creado el fichero **000-default** que es un enlace al fichero **000-default.conf** del directorio **sites-available**
 - Listar los sitios virtuales habilitados: **ls /etc/apache2/sites-enabled**

```
root@base:/etc/apache2/sites-enabled# ls  
000-default.conf
```

```
root@base:/etc/apache2/sites-enabled# ls -l  
total 0  
lrwxrwxrwx 1 root root 35 Nov 22 11:47 000-default.conf -> ../sites-available/000-default.conf
```


- Directivas de configuración básicas
- Alojamiento compartido basado en nombre
 - Pasos y directivas a utilizar:
 1. Habilitar los servidores virtuales por nombre => usar la directiva **NameVirtualHost** e indicar en qué **IP** y **puertos** se aceptarán peticiones
 - Ej: **NameVirtualHost** *:80 => se aceptarán peticiones de cualquier IP en el puerto 80
 2. Crear un bloque **<VirtualHost>** para cada uno de los diferentes sitios que se van servir. El argumento de este bloque debe coincidir con el de la directiva **NameVirtualHost**. Dentro de cada uno de estos bloques se precisan, al menos, dos directivas:
 - **ServerName** para designar el nombre del sitio/dominio que se sirve
 - **DocumentRoot** para indicar donde se encuentran los ficheros de ese sitio en el árbol de directorios
 3. Muchos sitios web quieren ser accesibles a través de varios nombres => emplear la directiva **ServerAlias**
 4. Pueden incluirse otras directivas que indiquen cómo se sirve el sitio: **DirectoryIndex**, directivas de control de acceso, etc.
 - Para saber si una directiva puede situarse en un bloque **VirtualHost** debe consultarse el apartado **Context** de la documentación oficial de dicha directiva

- Directivas de configuración básicas
- Alojamiento compartido basado en nombre
- Dónde situar las directivas en Ubuntu
 - **NameVirtualHost** debe situarse en el fichero **/etc/apache2/ports.conf**
 - Los bloques **<VirtualHost>** deben situarse cada uno en un archivo separado dentro del directorio **/etc/apache/sites-available** y tener un nombre significativo (se recomienda utilizar el que se indica en la directiva **ServerName**). En Apache 2.4 es necesario que estos ficheros tengan la extensión **.conf**
 - Ej: **example.com.conf**
- Comandos
 - **Activar un sitio virtual**: crea un enlace simbólico en **sites_enabled**
 - **sudo a2ensite nombreServidor.conf**
 - **Desactivar un sitio virtual**: borra el enlace simbólico de **sites_enabled**
 - **sudo a2dissite nombreServidor.conf**
 - Acordarse de **reiniciar** Apache!!

- Directivas de configuración básicas
- Alojamiento compartido basado en nombre
- Ejemplos:

```
<VirtualHost *:80>
    ServerName uno.midominio.es
    ServerAlias uno.midominio.com
    DocumentRoot /var/www/uno
    DirectoryIndex indice.html
</VirtualHost>

<VirtualHost *:80>
    ServerName dos.otrodominio.com
    DocumentRoot /var/www/dos
</VirtualHost>
```

/etc/apache2/sites-available/example.com.conf

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

- Directivas de configuración básicas
- Alojamiento compartido basado en nombre
- Dónde situar las directivas en Windows
- Todas están en el fichero C:\Apache24\conf\extra\httpd-vhost.conf

```
<VirtualHost _default_:80>
DocumentRoot "${SRVROOT}/htdocs"
#ServerName www.example.com:80
</VirtualHost>

# Add any other Virtual Hosts below

#<VirtualHost *:80>
#   ServerAdmin webmaster@dummy-host.example.com
#   DocumentRoot "${SRVROOT}/docs/dummy-host.example.com"
#   ServerName dummy-host.example.com
#   ServerAlias www.dummy-host.example.com
#   ErrorLog "logs/dummy-host.example.com-error.log"
#   CustomLog "logs/dummy-host.example.com-access.log" common
#</VirtualHost>
```

- Para poder utilizar alojamiento virtual hay que incluir dicho fichero en **httpd.conf** (descomentando la línea correspondiente)

```
# Virtual hosts
Include conf/extra/httpd-vhosts.conf
```

- Directivas de configuración básicas
- Alojamiento compartido basado en nombre
- Comprobar el correcto funcionamiento de los virtual hosts configurados
 - Desde el navegador del cliente, comprobar que se puede entrar en los diferentes dominios

`http://example.com`

Success! The example.com virtual host is working!

`http://test.com`

Success! The test.com virtual host is working!

- **Tarea3_a: crear sitios virtuales en Linux**

- Configura Apache para que sirva dos sitios web que tendrán las siguientes características:

1. Crea y habilita un servidor virtual para el dominio **daw.xunta.es**

- su directorio base será **/var/www/daw**
- contendrá una página llamada **daw.html**, que será la que se sirve por defecto
- si no se indica ningún archivo, no se mostrará el listado del directorio raíz
- tendrá un alias **daw.xunta.com**

2. Crea y habilita un servidor virtual para el dominio **asir.xunta.es**

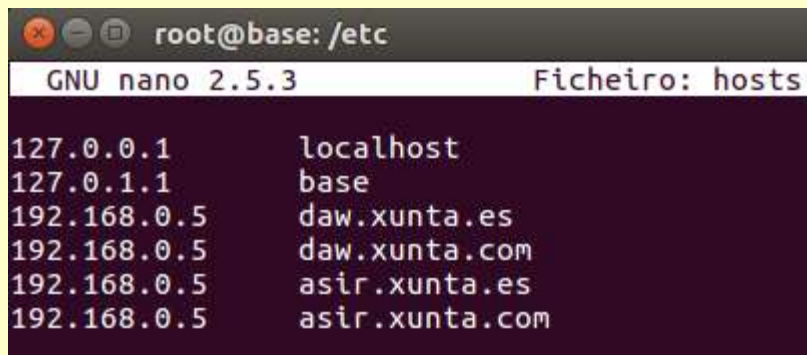
- su directorio base será **/var/www/asir**
- contendrá una página llamada **asir.html**, que será la que se sirve por defecto
- si no se indica ningún archivo, se mostrará el listado del directorio raíz
- tendrá un alias **asir.xunta.com**

- Comprueba el correcto funcionamiento de los virtual hosts. Documenta cada uno de los pasos anteriores, incluyendo breves explicaciones y pantallazos
- Si es necesario, utiliza otras fuentes para averiguar cómo se configuran los servidores virtuales, y cítalas en tu documento

■ Sol Tarea3_a: crear sitios virtuales en Linux

1. Configurar servidor DNS

- Como no tenemos ningún servidor DNS, relacionamos las direcciones simbólicas con la IP del servidor desde el archivo **hosts** de la máquina cliente



```
root@base: /etc
GNU nano 2.5.3      Ficheiro: hosts
127.0.0.1          localhost
127.0.1.1          base
192.168.0.5        daw.xunta.es
192.168.0.5        daw.xunta.com
192.168.0.5        asir.xunta.es
192.168.0.5        asir.xunta.com
```

2. Designar qué IP's y puertos aceptarán peticiones

- En versiones anteriores a 2.4, para activar los servidores virtuales por nombre, es necesario añadir la directiva **NameVirtualHost** en **/etc/apache2/ports.conf** para todas las direcciones IP y en el puerto 80 (y en versiones anteriores a la 2, había que deshabilitar el servidor virtual por defecto con a2dissite 000-default.conf)

```
Listen 80
NameVirtualHost *:80
```

3. Crear la carpeta /DAW y dentro un fichero y una carpeta con otro fichero desde **/var/www** crear con **mkdir** la carpeta DAW, y dentro un fichero **daw.html**

■ Sol Tarea3_b: crear sitios virtuales en Windows

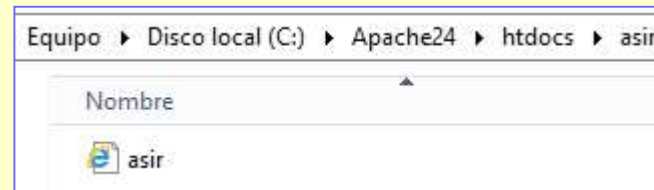
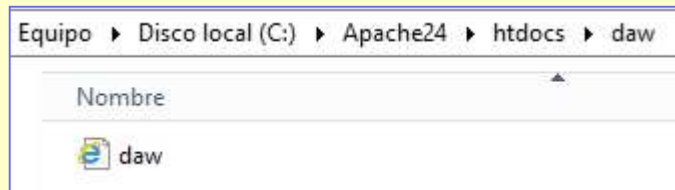
1. Configurar servidor DNS

- Como no tenemos ningún servidor DNS, relacionamos las direcciones simbólicas con la IP del servidor desde el archivo **hosts** de la máquina cliente situado en **C:\WINDOWS\system32\drivers\etc**

```
127.0.0.1      localhost
192.168.0.4    daw.xunta.es
192.168.0.4    daw.xunta.com
192.168.0.4    asir.xunta.es
192.168.0.4    asir.xunta.com
```

2. Crear la estructura de carpetas con el contenido del host virtual

- carpeta **daw** dentro de **C:\Apache24\htdocs**, y dentro al menos el fichero **daw.html**
- carpeta **asir** dentro de **C:\Apache24\htdocs**, y dentro al menos el fichero **asir.html**



3. Modificar el archivo **httpd.conf** que se encuentra en **C:\Apache24\conf** para que incluya el fichero **httpd-vhosts.conf** descomentando la línea:

- `#Include conf/extra/httpd-vhosts.conf`

```
# Virtual hosts
Include conf/extra/httpd-vhosts.conf
```


■ **Tarea3_b: crear sitios virtuales en Windows**

- Configura Apache para que sirva dos sitios web que tendrán las siguientes características:
 1. Crea y habilita un servidor virtual para el dominio **daw.xunta.es**
 - su directorio base será **C:\Apache24\htdocs\daw**
 - contendrá una página llamada **daw.html**, que será la que se sirve por defecto
 - si no se indica ningún archivo, no se mostrará el listado del directorio raíz
 - tendrá un alias **daw.xunta.com**
 2. Crea y habilita un servidor virtual para el dominio **asir.xunta.es**
 - su directorio base será **C:\Apache24\htdocs\asir**
 - contendrá una página llamada **asir.html**, que será la que se sirve por defecto
 - si no se indica ningún archivo, se mostrará el listado del directorio raíz
 - tendrá un alias **asir.xunta.com**
- Comprueba el correcto funcionamiento de los virtual hosts. Documenta cada uno de los pasos anteriores, incluyendo breves explicaciones y pantallazos
- Si es necesario, utiliza otras fuentes para averiguar cómo se configuran los servidores virtuales, y cítalas en tu documento