TEMA 2

Contenido

Configuración y administración de servidores Web.

Caso práctico
A la empresa BK Programación le ha surgido un nuevo proyecto: una empresa con varias sucursales quiere montar una aplicación web por sucursal.
Ada, la directora, considera que para afrontar este proyecto y atender así la demanda ofrecida, deben configurar un nuevo equipo servidor. Para tal fin se reúne con María:
-Hola María -dijo Ada-, nos han ofrecido un nuevo proyecto relacionado con servicios web, pienso que podemos afrontarlo, pero quería saber tu opinión: ¿con la infraestructura que tenemos ahora ves necesario el montaje de otro equipo servidor dedicado a este proyecto o con lo que tenemos nos arreglamos?
-Pienso -dijo María- que tal como estamos ahora, sí o sí, independientemente de los recursos que consuma este nuevo proyecto necesitamos la configuración de otro equipo servidor. Además debemos configurar dos entornos: el de pruebas y el de producción. ¿Para cuándo sería el proyecto?
-El proyecto debemos entregarlo con fecha final dentro de tres meses.
-Entonces, creo que si todo sigue su cauce normal no tendremos ningún tipo de problema para la ejecución del proyecto. ¿Qué recursos humanos habías pensado y dispones para destinar al proyecto?
-Ahora disponemos de todo el personal de la empresa y cuento contigo y con Juan para que os coordinéis las funciones de este proyecto.
-Pues por mí, no veo objeción al mismo.
-Bien -asintió Ada-, entonces no se hable más, tendremos que configurar otro equipo servidor y aceptamos el proyecto.
Así, la empresa BK Programación envió un presupuesto a la empresa del proyecto, ésta lo aprobó y comenzó el trabajo.
Para afrontar el nuevo proyecto al que se enfrenta BK Programación se acuerda en una reunión en la que asistieron: Ada, María y Juan, quien sería destinado al nuevo proyecto y las funciones a realizar en el mismo. Así, en dicha reunión se determinó que María sería la encargada del montaje, configuración y administración del nuevo equipo servidor y Juan el encargado de coordinar con el resto del personal la creación y funcionamiento de las aplicaciones web del proyecto.
María, entonces, se puso manos a la obra y determinó el siguiente escenario de trabajo para el equipo servidor de este proyecto:

- ✓ *Sistema Operativo: Debian GNU/Linux 6.0*
- ✓ *Servidor Web: Apache (apache2)*
  - *Configuración de Red:*
  - *Servidor Web: 192.168.200.250*
  - *Cliente de pruebas (desde donde se lanza el navegador): 192.168.200.100*

"Se debe hacer todo tan sencillo como sea posible, pero no más sencillo."
Albert Einstein

Hay que tener en cuenta que en el escenario las IP empleadas son **IP privadas**, sin existencia en Internet, por lo cual siempre que se haga referencia a las mismas a través de nombre de dominios, deberá existir un **servidor DNS** que las resuelva en local o bien en su defecto deberán existir las entradas correspondientes en el fichero del sistema local `/etc/hosts`.

1.- Funcionamiento de un servidor Web.

Caso práctico

Para poder llevar a buen fin el proyecto, María, reúne al equipo destinado al mismo, ya que quiere que todo el personal tenga claro los requisitos, entregables y fechas de ejecución del proyecto. Así, en esta reunión informativa para todo el equipo destinado al proyecto, trató los siguientes temas:

1. *Recursos del equipo servidor.*
2. *Conectividad del equipo servidor.*
3. *Servidor web empleado: El porqué de su elección y funcionamiento.*
4. *Posibilidades del servidor web empleado.*
5. *Requisitos de las aplicaciones web del proyecto.*
6. *Entregables y fechas.*

¿Alguna vez te has parado a pensar qué existe detrás de una página web?¿Por qué al escribir un nombre en un navegador puedes visionar una página web?¿Por qué no tienes acceso a determinadas páginas?¿De qué modo puedes impedir el acceso a determinados sitios de una página: por directorio, por usuario? ¿Cómo se puede establecer una comunicación segura en una transición bancaria? ...

Hoy en día utilizamos Internet como una herramienta común: para el trabajo, para el ocio... Pero sin duda el elemento fundamental que usamos no es otro que el navegador, gracias al cual podemos sacar partido a todo lo que se encuentra en Internet: comprar entradas para el cine, acceder a nuestra cuenta bancaria, averiguar el tiempo que hará el fin de semana... pero nada de esto tendría sentido si detrás de cada página web a la que accedemos no existiera un servidor web, el cual permite que la página esté accesible 24x7 (24 horas al día y 7 días a la semana, es decir, siempre).

Detrás de cada página web debe existir un servidor web que ofrezca esa página, bien a los internautas, a los trabajadores de una empresa -por tratarse de una página web interna, de la empresa, no accesible a Internet-, o a todo aquel que disponga de una conexión de red con la cual pueda acceder a la página.

La configuración del servidor web dependerá de las páginas web que ofrezca, así la configuración no será la misma si la página posee contenido estático o no, o si se necesita que modifique el contenido según interacción del usuario, o si se necesita de comunicación segura en la transición de información, o si se debe tener en cuenta el control de acceso a determinados sitios de la página. Por lo tanto según las páginas web que se ofrezcan el servidor web deberá estar configurado para tal fin: con soporte PHP, con soporte de cifrado, con soporte de control de acceso, etc.

Pero ¿un servidor web pueda alojar varias páginas web o solamente una? Es más, ¿puede alojar varios sitios (*Conjunto de páginas web*), dominios de Internet (*Nombre por el cual se reconoce a un grupo de dispositivos o equipos conectados a la red. Éstes pueden ser nombres locales, no existentes en Internet, pero son mayoritariamente utilizados para su uso en Internet, por ejemplo: debian.org*) o solamente uno, esto es, permite hosts virtuales (*Dominios independientes que se pueden alojar en un mismo servidor web*)? Pues, un servidor web puede alojar varias páginas, sitios, dominios de Internet, pero hay que tener en cuenta que la elección del servidor web será muy importante para la configuración y administración de uno o múltiples sitios, ya que: ¿puede el servidor web ser modular -fácilmente se le pueden añadir o quitar características-?, o por la contra si queremos añadirle una funcionalidad que no posea en la instalación base debemos desinstalarlo e instalarlo de nuevo, por ejemplo: hasta ahora el servidor web solamente ofrecía páginas estáticas pero queremos ofrecer también páginas web dinámicas, qué hacemos: modular o nueva instalación.

También tenemos que pensar que todo puede crecer y lo que ahora era un servidor web que ofrecía x número de páginas necesitamos que ofrezca x*y, con lo cual tenemos que prever la escalabilidad del servidor web, y también la estabilidad: ¿cómo se comporta ante mútiples conexiones simultáneas?
De nada servirá tener instalado un servidor web sin saber cómo se va a comportar ofreciendo el servicio, con lo cual será muy importante previamente y durante el funcionamiento del servidor establecer unas pruebas de funcionamiento del mismo y registrar lo acontecido.

Por todo lo anteriormente comentado veremos cómo configurar y administrar el servidor Apache (apache2), ya que soporta: páginas web estáticas, dinámicas, hosts virtuales, seguridad mediante cifrado, autenticación y control de acceso, modularización y monitorización de archivos de registro.

1.1.- Servicio de ficheros estáticos.

¿Es necesario que todas las páginas web se modifiquen constantemente? ¿Un blog sería útil si el contenido no sufre cambios? ¿Y un manual? ¿Si actualizamos un manual la página deja de ser estática?

Todas aquellas páginas web que durante el tiempo no cambian su contenido no necesariamente son estáticas. Una página estática puede modificarse, actualizando su contenido y seguir siendo estática, ¿entonces? Entonces debemos diferenciar cuando accedemos a una página web entre código ejecutable en el lado del servidor y en el lado del cliente -equipo que solicita la página mediante el cliente web (navegador)-. Si al acceder a una página web no es necesaria la intervención de código en el lado del servidor -por ejemplo código PHP- o en el lado del cliente -por ejemplo javascript- entonces entenderemos que la página es estática, si por el contrario es necesaria la intervención en el lado del servidor y/o en el lado del cliente entenderemos que la página es dinámica.

Ofrecer páginas estáticas es simple, puesto que solamente se necesita que el servidor web disponga de soporte html/xhtml/css o incluso solamente html/xhtml. En cuanto a configuración y administración del servidor es el caso más simple: solamente se necesita un soporte mínimo base de instalación del servidor Apache, esto es, no se necesita por ejemplo soporte PHP. En cuanto a rendimiento del servidor, sigue siendo el caso más beneficioso: no necesita de ejecución de código en el lado del servidor para visionar la página y tampoco necesita ejecución de código en el lado del cliente, lo que significa menos coste de CPU y memoria en el servidor y en el cliente, y por lo tanto una mayor rapidez en el acceso a la información de la página.

Para poder ofrecer páginas estáticas mediante el servidor Apache simplemente copias la página en la ruta correspondiente donde quieres que se visione la página. Así por ejemplo cuando se instala Apache en un GNU/Linux Debian 6 se crean una serie de rutas en el equipo servidor similar a la estructura siguiente.

Rutas de interés en la instalación de Apache (apache2)

Rutas de interés en la instalación de Apache (apache2) en un GNU/Linux Debian	
/etc/apache2/ ├── apache2.conf ├── conf.d ├── envvars ├── httpd.conf ├── magic ├── mods-available ├── mods-enabled ├── ports.conf ├── sites-available └── sites-enabled	/etc/apache2/sites-available/ ├── default └── default-ssl /var/www/ └── index.html /etc/apache2/mods-available/mime.conf /etc/apache2/apache2.conf

En la instalación de Apache se crea una página web en `/var/www/index.html` referenciada a través del archivo default (`/etc/apache/sites-available/default`), éste contiene la configuración por defecto, generada en la instalación de Apache, para esa página. Si solamente quieres servir una página web la forma más fácil de hacerlo sería sustituyendo la página `index.html`, referenciada en `default`, por la página que quieres servir, por ejemplo `empresa.html`. Puedes comprobarlo siguiendo el procedimiento:

1. Abres el navegador en la página por defecto creada en la instalación de Apache: `index.html`.
2. Sustituyes los archivos en el servidor. Ten en cuenta que la página a servir debe siempre poseer el nombre `index.html`.
3. Pulsas F5 en el navegador para actualizar la página y la página que verás será la tuya.

Si lo que quieres es servir otra página, por ejemplo `empresa.html`, simplemente no le cambies como antes el nombre, deja el que posee la página. Ahora podrás ver dos páginas en el servidor: la página `index.html` y la página `empresa.html`. Si lo que quieres es servir más páginas pues, como antes, simplemente vas subiendo al servidor las páginas e incluso podrías organizarlas en carpetas.

Te proponemos que hagas un viaje por la página web de documentación de Apache.
http://httpd.apache.org/docs/2.2/es/

1.2.- Contenido dinámico.

"El progreso consiste en el cambio."

Miguel de Unamuno

Muchas veces seguro que te encuentras visitando una página web y la información te parece tan interesante que procedes y guardas en **Favoritos** la dirección URL (*dirección de Internet de un recurso válida para su posible utilización a través de Internet, la cual permite que el navegador la encuentre y la muestre de forma adecuada, por ejemplo: http://www.debian.org*) para una posterior visión, pero cuando de nuevo deseas ver la página resulta que lo que estás viendo no tiene nada que ver o es distinto de lo que esperabas, ¿qué ha ocurrido? Pues puede que la página haya cambiado su contenido o que la página que visitas posee contenido no estático, dinámico, dependiente del código ejecutado en el servidor o en el cliente al acceder a la página.

Imagínate que accedes a una página web y dependiendo si posees una cuenta de usuario u otra el contenido es distinto, o que presionas en una imagen de la página y se produce un efecto en la misma, o que el contenido cambia dependiendo del navegador. De cualquier forma la página ha sido modificada mediante una interacción con el usuario y/o el navegador, por lo tanto nos encontramos con una página dinámica.

Como bien puedes pensar, una página dinámica, necesita más recursos del servidor web que una página estática, ya que consume más tiempo de CPU y más memoria que una página estática. Además la configuración y administración del servidor web será más compleja: cuántos más módulos tengamos que soportar, más tendremos que configurar y actualizar. Esto también tendrá una gran repercusión en la seguridad del servidor web: cuántos más módulos más posibilidades de problemas de seguridad, así si la página web dinámica necesita, para ser ofrecida, de ejecución en el servidor debemos controlar que es lo que se ejecuta.

Algunos módulos con los que trabaja el servidor web Apache para poder soportar páginas dinámicas son: `mod_actions`, `mod_cgi`, `mod_cgid`, `mod_ext_filter`, `mod_include`, `mod_ldap`, `mod_perl`, `mod_php5`, `mod_python`.

En el siguiente enlace a la página de Apache puedes ampliar la información que te proporcionamos sobre los módulos.

http://httpd.apache.org/docs/2.2/es/mod

Abres el navegador y solicitas una página a un servidor web: ¿cuál de las siguientes acciones indica que la página solicitada no es dinámica?

- La página tiene un panel de control, al cual accedes mediante tu usuario y tu contraseña, los cuales nunca cambias. La página entonces establece comunicación con una base de datos y te permite el acceso a tu perfil, distinto del perfil del administrador de la página.
- Al pasar el puntero por encima de una imagen, ésta se redimensiona y al salir vuelve al tamaño original.
- Cuando visitas la página con distintos navegadores aparece un comentario de alerta indicando el navegador con el cual estás accediendo a la página.
- **La página solicitada es un manual sobre el Servidor Apache, y está totalmente escrita en código HTML y CSS.**

Aquellas páginas cuyo contenido no depende da la interacción del usuario, del navegador o un sistema gestor de bases de datos son páginas estáticas.

1.3.- Protocolo HTTP y HTTPS.

¿Quieres conservar la información de forma confidencial? ¿Quieres transferir información de forma segura? Si estás pensando en este tipo de preguntas necesariamente estás pensando en el protocolo HTTPS (*protocolo basado en el protocolo HTTP, destinado a la transferencia segura de datos mediante cifrado, es decir, es la versión segura de HTTP*) y no en el protocolo HTTP (*protocolo usado en cada transacción de la World Wide Web*).

El protocolo HTTPS permite que la información viaje de forma segura entre el cliente y el servidor, por la contra el protocolo HTTP envía la información en texto claro, esto es, cualquiera que accediese a la información transferida entre el cliente y el servidor puede ver el contenido exacto y textual de la información.

Para asegurar la información, el protocolo HTTPS requiere de certificados y siempre y cuando sean validados, la información será transferida cifrada. Pero cifrar la información requiere un tiempo de computación, por lo que será perjudicado el rendimiento del servidor web. Así, ¿es necesario que toda, absolutamente toda, la información sea transferida entre el cliente y servidor de forma cifrada? A lo mejor solamente es necesario que sea cifrada la autenticación a dicha información, por eso en algunas páginas web puede que el servidor esté configurado para que en todo el dominio esté cifrada su información o simplemente el intento de acceso a la misma.

Un servidor web, como Apache, puede emitir certificados, pero puede que en algún navegador sea interpretado como peligroso, esto suele ser debido a que los navegadores poseen en su configuración una lista de Entidades Certificadoras que verifican, autentican y dan validez a los certificados. ¿Tú, confiarías en un DNI que no fuese certificado por una entidad de confianza como el Ministerio del Interior? Pues, lo mismo le pasa a los navegadores, solamente confían en quien confían. Eso no quiere decir que no puedes crear tus certificados en un servidor web, de hecho muchas empresas lo hacen, sobre todo para sitios internos o externos en los que solamente puede acceder personal autorizado por la propia empresa. Ahora si, si utilizas certificados mediante Apache en un sitio visible a través de Internet y accesible por cualquier usuario, o bien eres una empresa o entidad en la que de por si confía el usuario o la imagen de la empresa o entidad quedará muy mal parada, ya que lo más probable es que el usuario no aceptará la comunicación, por visionar en el navegador un aviso de problema de seguridad.

El protocolo HTTPS utiliza cifrado sobre SSL/TLS (*protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet*) que proporcionan autenticación y privacidad. Entonces, si necesitas que la información viaje cifrada debes emplear el protocolo HTTPS, en caso contrario el protocolo HTTP. Hay que dejar claro que la utilización del protocolo HTTPS no excluye ni impide el protocolo HTTP, los dos pueden convivir en un mismo dominio.

Bien, pero, ¿cómo funcionan? En el protocolo HHTP cuando escribes una dirección URL en el navegador, por ejemplo `http://www.debian.org/index.es.html`, antes de ver la página en el navegador existe todo un juego de protocolos, sin profundizar en todos ellos básicamente lo que ocurre es lo siguiente: se traduce el dominio DNS (*sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a Internet o a una red privada, por ejemplo: apache.org determina un dominio org (organización) y un subdominio que identifica en este caso la máquina o conjunto de máquinas de nombre apache*) por una IP, una vez obtenida la IP se busca en ella si un servidor web aloja la página solicitada en el puerto 80 (*número utilizado en las comunicaciones cliente/servidor, en transmisiones TCP o UDP comprendido entre 1 y 65535, que indica por donde tiene lugar la conexión con un servidor. Están estandarizados, esto es, un servidor suele estar activo siempre por definición en un puerto determinado, pero éste puede que sea modificado en la configuración del servidor. Por ejemplo un servidor web espera en el puerto TCP 80*) , puerto TCP (*es uno de los protocolos fundamentales en Internet. Garantiza que los datos serán entregados en su destino sin errores y una vez recogidos ponerlos en el mismo orden en que se transmitieron*) asignado por defecto al protocolo HTTP. Si el servidor web aloja la página ésta será transferida a tu navegador. Sin embargo cuando escribes en el navegador una dirección URL con llamada al protocolo HTTPS, el procedimiento es similar al anterior pero un poco más complejo, así se traduce el dominio DNS por una IP, con la IP se busca el servidor web que aloja la página solicitada en el puerto 443, puerto TCP asignado por defecto al protocolo HTTPS, pero ahora antes de transferir la página a tu navegador se inicia una negociación SSL, en la que entre otras cosas el servidor envía su certificado -el navegador aunque es poco habitual también puede enviar el suyo-. Si el certificado es firmado por un Entidad Certificadora de confianza se acepta el certificado y se cifra la comunicación con él, transfiriendo así la página web de forma cifrada.

Puedes hacer que un servidor web para una determinada página espere los protocolos HHTP y HTPS en puertos TCP distintos del 80 y 443 respectivamente. Eso sí, cuando visites la página web a mayores en la dirección URL debes especificar el puerto TCP, por ejemplo: `http://www.tupagina.local:8080`, de esta forma el servidor web espera la petición de la página `www.tupagina.local` en el puerto 8080; del mismo modo en la dirección URL: `https://www.tupagina.local:4333` espera la petición de la página `www.tupagina.local` en el puerto 4333. Como ves, puedes configurar los puertos, pero ten en cuenta que cualquiera que quisiera acceder a esas páginas debería saber el puerto TCP de la solicitud. Entonces, quiere decir que ¿aunque no escribas el puerto TCP en las direcciones URL estas se interpretan en el puerto 80 y 443 para el protocolo HTTP y HTTPS respectivamente? Pues si, así es. Es lo mismo escribir `http://www.tupagina.local:80` que `http://www.tupagina.local` y es lo mismo escribir `https://www.tupagina.local:443` que `https://www.tupagina.local`

En la página http://www.warriorsofthe.net/index.html puedes encontrar un vídeo muy ameno sobre el funcionamiento de Internet.

1.4.- Tipos MIME.

¿Cómo se transmite un vídeo por Internet, con qué codificación?¿Cómo sabe un navegador que al seguir un enlace de vídeo el programa que debe utilizar para reproducirlo?

El estándar Extensiones Multipropósito de Correo de Internet o MIME (Multipurpose Internet Mail Extensions), especifica como un programa debe transferir archivos de texto, imagen, audio, vídeo o cualquier archivo que no esté codificado en US-ASCII. **MIME** está especificado en seis RFC (***Request for Comments.*** *Serie de documentos en los que se detalla prácticamente todo lo relacionado con la tecnología de la que se sirve Internet: protocolos, recomendaciones, comunicaciones…*) :

RFC2045	http://tools.ietf.org/html/rfc2045
RFC 2046	http://tools.ietf.org/html/rfc2046
RFC 2047	http://tools.ietf.org/html/rfc2047
RFC 4288	http://tools.ietf.org/html/rfc4288
RFC4289	http://tools.ietf.org/html/rfc4289
RFC2077	http://tools.ietf.org/html/rfc2077

¿Cómo funciona? Imagínate el siguiente ejemplo: Transferencia de una página web.

Cuando un navegador intenta abrir un archivo el estándar MIME le permite saber con qué tipo de archivo está trabajando para que el programa asociado pueda abrirlo correctamente. Si el archivo no tiene un tipo MIME especificado el programa asociado puede suponer el tipo de archivo mediante la extensión del mismo, por ejemplo: un archivo con extensión **.txt** supone contener un archivo de texto.
Bien, pero ¿cómo lo hace?

El navegador solicita la página web y el servidor antes de transferirla confirma que la petición requerida existe y el tipo de datos que contiene. Esto último, mediante referencia al tipo MIME al que corresponde. Este diálogo, oculto al usuario, es parte de las cabeceras HTTP (*Son el lenguaje que utilizan el cliente(navegador web) y el servidor web para comunicarse entre sí. Se puede considerar cada cabecera como un mensaje aparte en el sentido de la comunicación entre el cliente y el servidor. Primero hay unas cuantas preguntas (cabeceras de solicitud), las cuales son respondidas (cabeceras de respuesta)*), protocolo que se sigue en la web.

En ese diálogo, en las cabeceras respuestas del servidor existe el campo `Content-Type`, donde el servidor avisa del tipo MIME de la página. Con esta información, el navegador sabe cómo debe presentar los datos que recibe. Por ejemplo cuando visitas `http://www.debian.org/index.es.html` puedes ver como respuesta en la cabecera del servidor el campo `Content-Type: text/html` , indicando que el contenido de la página web es tipo texto/html:

```
HTTP/1.1 200 OK
Date: Fri, 13 May 2011 18:11:36 GMT
Server: Apache
Last-Modified: Fri, 13 May 2011 16:22:52 GMT
Etag: "3a9b-4a32ab7a76f00"
Accept-Ranges: bytes
Cache-Control: max-age=86400
Expires: Sat, 14 May 2011 18:11:36 GMT
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 4864
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html
Content-Language: es
```

Cada identificador de tipo MIME consta de dos partes. La primera parte indica la categoría general a la que pertenece el archivo como, por ejemplo, **"text"**. La segunda parte del identificador detalla el tipo de archivo específico como, por ejemplo, **"html**". Un identificador de tipo MIME **"text/html"**, por ejemplo, indica que el archivo es una página web estándar.

Los tipos MIME pueden indicarse en tres lugares distintos: el servidor web, la propia página web y el navegador.

- ✓ El servidor debe estar capacitado y habilitado para manejar diversos tipos MIME.
- ✓ En el código de la página web se referencia tipos MIME constantemente en etiquetas link, script, object, form, meta, así por ejemplo:
  - ➔ El enlace a un archivo hoja de estilo CSS:

```
<link href="./miarchivo.css" rel="stylesheet" type="text/css">
```

  - ➔ El enlace a un archivo código javascript:

```
<script language="JavaScript" type="text/javascript" src="scripts/mijavascript.js">
```

- ✓ Con las etiquetas meta podemos hacer que la página participe en el diálogo servidor-cliente, especificando datos MIME:

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

- ✓ El navegador del cliente también participa, además de estar capacitado para interpretar el concreto tipo MIME que el servidor le envía, también puede, en el diálogo previo al envío de datos, informar que tipos MIME puede aceptar la cabecera http_accept, así por ejemplo una cabecera http_accept tipo de un navegador sería:

```
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
```

El valor `*/*` significa que el navegador aceptará cualquier tipo MIME

Complementos del navegador Firefox para ver cabeceras HTTP/HTTPS:

Tamper Data **https://addons.mozilla.org/es-ES/firefox/addon/tamper-data/**

Live HTTP Headers **https://addons.mozilla.org/es-ES/firefox/addon/live-http-headers/**

1.4.1.- Configurar el servidor para enviar los tipos MIME correctos.

En un servidor web podemos especificar el tipo MIME por defecto para aquellos archivos que el servidor no pueda identificar automáticamente como pertenecientes a un tipo concreto, esto es, para aquellos los cuales no se resuelven según su extensión.

Para el servidor web Apache se utilizan dos directivas: **`DefaultType`** y **`ForceType`**.

- ✓ **`DefaultType`** asigna la cabecera Content-Type a cualquier archivo cuya MIME no pueda determinarse desde la extensión del archivo.
- ✓ **`ForceType`** hace que todos los ficheros cuyos nombres tengan una equivalencia con lo que se especifique sean servidos como contenido del tipo MIME que se establezca.

Ejemplos:

- ✓ **`DefaultType text/plain`** : Esto significa que cuando el navegador web solicita y recibe ese archivo como respuesta, desplegará el contenido como un archivo de texto.
- ✓ **`DefaultType text/html`** : Desplegará el contenido como un archivo HTML.
- ✓ **`ForceType image/gif`** **:** Desplegará el contenido como un archivo de imagen gif.
- ✓ **`ForceType video/mp4`** : Desplegará el contenido como un archivo de vídeo mp4.

En el siguiente enlace puedes encontrar más información sobre la directiva DefaultType.
http://httpd.apache.org/docs/2.0/mod/core.html#defaulttype

Puedes consultar más información en la documentación de Apache sobre directivas.
http://httpd.apache.org/docs/2.0/mod/directives.html
http://httpd.apache.org/docs/2.0/mod/quickreference.html

En el servidor web Apache existe el archivo `/etc/apache2/mods-available/mime.conf` donde encontrarás una referencia al archivo /etc/mime.types, el cual contiene la lista de tipos MIME reconocidos por el servidor.

En el siguiente enlace encontrarás la lista oficial de los tipos MIME.
http://www.iana.org/assignments/media-types/

Abres el navegador y solicitas una página web que contiene un vídeo con la extensión .flv a un servidor web Apache: ¿cuáles de las siguientes afirmaciones son correctas teniendo en cuenta que el vídeo puede reproducirse y visualizarse sin problemas?

- [x] **El servidor web no identifica el tipo MIME pero la extensión .flv es reconocida por el navegador, es por esto que el navegador asocia el programa correspondiente al vídeo y se reproduce sin problemas.**
- [ ] El archivo no es reconocido por el servidor web, por lo que el servidor web envía al navegador otro tipo MIME, compatible con el esperado y el vídeo se reproduce sin problemas.
- [x] **Si la extensión .flv no es reconocida por el navegador ni por el servidor web es debido a que el tipo MIME es reconocido por cómo está programada la página web.**
- [ ] El servidor web no identifica el tipo MIME pero como el servidor web reconoce la extensión .flv modifica la programación de la página web incorporando el código necesario para la reprodución del vídeo.

El archivo se reproduce porque el tipo MIME viene especificado por el código programado en la página web o porque programa asociado supone el tipo de archivo mediante su extensión.

2.- Hosts virtuales. Creación, configuración y utilización.

Caso práctico

A la empresa BK Programación le ha surgido el siguiente proyecto: una empresa con varias sucursales quiere montar una aplicación web por sucursal. La empresa en cuestión consta de 7 sucursales. Todas ellas dedicadas a la misma línea de negocio. Así, las aplicaciones tendrán un frontal similar, pero estarán personalizadas dependiendo de la situación de la sucursal, de tal forma que los banners, logos e imágenes de cada aplicación serán monumentos locales a la zona de la sucursal.
El equipo de trabajo del proyecto está coordinado por María, ella es la encargada del montaje, creación y configuración del servidor web donde irán alojadas las aplicaciones web.
La empresa quiere que las sucursales puedan ser localizadas en Internet mediante URLs tipo: `www.sucursal-zonaX.empresa-proyecto.com`*, donde X puede variar de 1 a 7. Además quiere que si las páginas se buscan sin www éstas sigan viéndose, es decir, que* `sucursal-zonaX.empresa-proyecto.com` *se dirija a la misma página que* `www.sucursal-zonaX.empresa-proyecto.com`
La empresa también desea que exista un único panel de control de usuarios, en la URL `www.empresa-proyecto.panel-de-control.com`*, de tal forma que según el perfil que posea el usuario podrá ver un contenido u otro. Así, desea que los comerciales tengan la posibilidad de saber que productos y cantidades de los mismos existen en stock. Al panel de control se accede a través de un enlace configurado en cada aplicación.*
María se reúne con Juan, el encargado del desarrollo de las aplicaciones web, y con Antonio, que ejerce el rol del usuario destinado a comprobar el buen funcionamiento de las aplicaciones haciendo pruebas con distintos navegadores:
—Pienso —dijo María— que la mejor forma de llevar a buen puerto el proyecto se realiza configurando hosts virtuales en el servidor web Apache y no solamente colgando las aplicaciones web en un directorio raíz común para luego, cada una, disponer de su espacio en una carpeta independiente.
—Sí, —dijo Juan—, además tenemos que tener en cuenta la seguridad del panel de control, deberíamos pensar en el protocolo HTTPS, para asegurarnos que la información vaya cifrada.
—Estoy de acuerdo —afirmó María—. Entonces, Antonio, deberás hacer las pruebas mediante HTTP y HTTPS.
—Vale, de acuerdo —dijo Antonio—.

Anteriormente hemos visto como poder alojar múltiples páginas web en el servidor web Apache, pero todas pertenecientes al mismo sitio/dominio, es decir, todas pertenecientes a **empresa.com**, entonces, ¿no se puede alojar páginas de distintos dominios en el mismo servidor web? La respuesta es que si, si se puede, ¿cómo?, mediante la configuración de hostsvirtuales o virtualhosts. Éstos básicamente lo que hacen es permitir que un mismo servidor web pueda alojar múltiples dominios, así configurando hosts virtuales podemos alojar:**empresa1.com** , **empresa2.com**, ..., **empresaN.com** en el mismo servidor web. Cada empresa tendrá su virtualhost único e independiente de las demás.

Aunque como se ha comentado anteriormente cada virtualhost es único e independiente de los demás, todo aquello que no esté incluido en la definición de cada virtualhost se heredará de la configuración principal: **apache2.conf (**`/etc/apache2/apache2.conf`**)**, así. si quieres definir una directiva común en todos los virtualhost no debes modificar cada uno de los virtualhostintroduciendo esa directiva sino que debes definir esa directiva en la configuración principal del servidor web Apache, de tal forma que todos los virtualhost heredarán esa directiva, por ejemplo en apache2.conf puedes encontrar la directiva `Timeout 300`, que establece la directiva `Timeout` igual a 300 segundos, esto es, indica el número de segundos antes de que se cancele un conexión por falta de respuesta.

Existen tres tipos de virtualhost: basados en nombre, basados en IP y basados en varios servidores principales.

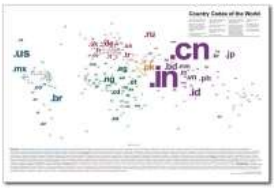
Si no tienes configurado un servidor DNS con las entradas de dominio necesarias, puedes generar estas entradas modificando el archivo `/etc/hosts`, añadiéndolas al final del mismo:

```
#IP nombre-dominio
192.168.200.250 empresa1.com www.empresa1.com
192.168.200.250 empresa2.com www.empresa2.com
```

Cada campo de cada entrada puede ir separado por espacios o por tabulados.

Estas entradas solamente serán efectivas en el equipo en el que se modifique el archivo **`/etc/hosts`**. Así debes modificar el archivo **`/etc/hosts`** en cada equipo que quieres que se resuelvan esas entradas.

2.1.- Virtualhosts basados en nombre



La IP que debemos poner siempre en la definición de la directiva Virtualhost es la IP del servidor web, en nuestro escenario:
`IP Servidor Web=192.168.200.250`

¿Cómo lo haces? Sigues el procedimiento:

1. En la configuración de Apache2 existe un directorio /etc/apache2/sites-available donde se definen los virtualhosts, cada virtualhost en un fichero de texto de configuración distinto, así crea los dos ficheros siguientes en la ruta /etc/apache2/sites-available.
2. Fichero configuración virtualhost: empresa1
```
<VirtualHost IP_Servidor_Web:80>
DocumentRoot /var/www/empresa1/
ServerName www.empresa1.com.
ServerAlias empresa1.com empresa1.es www.empresa1.es
</VirtualHost>
```
3. Fichero configuración virtualhost: empresa2
```
<VirtualHost IP_Servidor_Web:80>
DocumentRoot /var/www/empresa2/
ServerName www.empresa2.com.
ServerAlias empresa2.com empresa2.es www.empresa2.es
</VirtualHost>
```

Explicación fichero virtualhost:

`<VirtualHost IP_Servidor_Web:80>` : Inicio etiqueta virtualhost, define la IP del servidor web donde se aloja la página de la empresa, en este caso empresa1. El puerto TCP para el protocolo HTTP por defecto es el 80, definido en la configuración principal del servidor, mediante la directiva **Listen**, por lo cual no es necesario ponerlo. Se pueden usar varias directivas**Listen** para especificar varias direcciones y puertos de escucha. El servidor responderá a peticiones de cualquiera de esas direcciones y puertos. Por ejemplo, para hacer que el servidor acepte conexiones en los puertos 80 y 8080, usa:

```
Listen 80
Listen 8080
```

Para hacer que el servidor acepte conexiones en dos direcciones IP y puertos diferentes,usa:

```
Listen 192.168.200.250:80
Listen 192.168.200.251:8080
```

- ✓ **`DocumentRoot /var/www/empresa1/`** : Definición de la ruta donde está alojada la página web en el servidor, en este caso: `/var/www/empresa1/` mediante la directiva `DocumentRoot`.
- ✓ **`ServerName www.empresa1.com`** : Definición del nombre DNS que buscará la página alojada en la ruta anterior del servidor mediante la directiva `ServerName`. Es el nombre que escribes en el navegador para visitar la página.
- ✓ **`ServerAlias empresa1.com`** : La directiva `ServerAlias` permite definir otros nombres DNS para la misma página.

- ✓ `</VirtualHost>` : Fin de la etiqueta `VirtualHost`: fin de la definición de este virtualhost para la empresa1.

Si deseas que tu servidor web ofrezca en la misma IP las URL:

```
www.sucursal-zona2.empresa-proyecto.com, sucursal-zona2.empresa-proyecto.com
www.empresa-proyecto.panel-de-control.com.
```

donde las 2 primeras identifican el mismo sitio web y la última otro totalmente distinto. Entonces, ¿podrías utilizar para definir los virtualhosts?

- ○ Un solo fichero.
- ◉ Dos ficheros.
- ○ No se pueden utilizar virtualhosts, debido a que los dominios son distintos.
- ○ Incorrecta la pregunta ya que las URL están mal definidas, no pueden contener el carácter guión.

Si, ya que las 3 URL definen 2 sitios totalmente distintos.

2.2.- Virtualhosts basados en IP

La IP que debemos poner ahora en la definición de la directiva Virtualhost cambia, cada IP corresponde a una interfaz de red del servidor web, en nuestro escenario:

```
IP1_Servidor_Web=192.168.200.250
IP2_Servidor_Web=192.168.200.251
```

Este método no aporta ventajas sobre el anterior, es más, aún puede ser más difícil de mantener si las IP del servidor web se modifican con cierta frecuencia.

¿Cómo lo haces? Sigues el mismo procedimiento usado para los virtualhost basado en nombre, únicamente se diferencia en los ficheros a crear para los virtualhost, así:

1. En la configuración de Apache2 existe un directorio `/etc/apache2/sites-available` donde se definen los virtualhost, cada virtualhost en un fichero de texto de configuración distinto, así crea los dos ficheros siguientes en la ruta `/etc/apache2/sites-available`.
2. Fichero configuración virtualhost: `empresa3`

```
<VirtualHost IP1_Servidor_Web:80>
DocumentRoot /var/www/empresa3/
ServerName 192.168.200.250
</VirtualHost>
```

3. Fichero configuración virtualhost: `empresa4`

```
<VirtualHost IP2 Servidor Web:80>
DocumentRoot /var/www/empresa4/
ServerName 192.168.200.251
</VirtualHost>
```

Explicación fichero virtualhost:

`<VirtualHost>`: Inicio etiqueta virtualhost, define la IP1 del servidor web donde se aloja la página de la empresa, en este caso `empresa3`. El puerto TCP por defecto es el 80, definido en la configuración principal del servidor, mediante la directiva Listen, por lo cual no es necesario ponerlo. Se pueden usar varias directivas **`Listen`** para especificar varias direcciones y puertos de escucha. El servidor responderá a peticiones de cualquiera de esas direcciones y puertos. Por ejemplo, para hacer que el servidor acepte conexiones en los puertos 80 y 8080, usa:

```
<VirtualHost IP1_Servidor_Web:80>
DocumentRoot /var/www/empresa3/
ServerName www.empresa3.com.
ServerAlias empresa3.com empresa3.es www.empresa3.es
</VirtualHost>
```

Para hacer que el servidor acepte conexiones en dos direcciones IP y puertos diferentes, usa:

```
<VirtualHost IP2_Servidor_Web:80>
DocumentRoot /var/www/empresa4/
ServerName www.empresa4.com.
ServerAlias empresa4.com empresa4.es www.empresa4.es
</VirtualHost>
```

- ✓ `DocumentRoot /var/www/empresa3/`: Definición de la ruta donde está alojada la página web en el servidor, en este caso: `/var/www/empresa3/` mediante la directiva `DocumentRoot`.
- ✓ `ServerName www.empresa3.com` : Definición del nombre DNS que buscará la página alojada en la ruta anterior del servidor mediante la directiva `ServerName`. Es el nombre que escribes en el navegador para visitar la página.
- ✓ `ServerAlias empresa3.com` : La directiva `ServerAlias` permite definir otros nombres DNS para la misma página.
- ✓ `</VirtualHost>`: Fin de la etiqueta VirtualHost: fin de la definición de este virtualhost para la empresa3.

En el siguiente enlace encontrarás información sobre la directiva `RewriteRule`, la cual te puede evitar tener que utilizar la directiva `ServerAlias`, pues te permite reescribir las direcciones URL.

http://httpd.apache.org/docs/2.2/mod/mod_rewrite.html#rewriterule

2.3.- Virtualhosts basados en varios servidores principales

"Cuando me preguntan cuándo estará listo un programa, contesto: depende de cuánto trabaje usted en ello."

Richard Stallman

Este método es el más complejo de todos, solo tiene sentido cuando quieras tener varios archivos de configuración **apache2.conf** independientes organizando cada uno sus propios hostsvirtuales, en otro caso, mejor emplear alguno de los dos métodos anteriores.

XAMPP es una forma fácil de instalar y usar el servidor web Apache con MySQL, PHP y Perl. XAMPP, basta con descargarlo, extraerlo y comenzar.

En este momento hay cuatro versiones de XAMPP, para: Linux, Windows, Mac OS X y Solaris.

http://www.apachefriends.org/es/xampp.html

Te proponemos los siguientes enlaces con interesantísimos vídeos sobre la instalación y uso de XAMPP en Windows.

http://informatica.iessanclemente.net/screencast/xampp/

3.- Módulos.

Caso práctico
Está bien -pensó María-. Manos a la obra. Debo montar un nuevo servidor web Apache y necesito... veamos:
- ✓ *Que varias aplicaciones web atiendan en el mismo dominio, tal que:* `sucursal-zonaX.empresa-proyecto.com`, `www.sucursal-zonaX.empresa-proyecto.com`
- ✓ *Un único panel de control de usuarios, en la URL* `www.empresa-proyecto.panel-de-control.com`,
- ✓ *También soporte SSL para cifrado.*
- ✓ *Soporte para páginas dinámicas mediante PHP.*
- ✓ *Y soporte para control de usuarios LDAP.*

Uhm..., ya lo tengo claro. Tengo que montar Apache con varios módulos, así primero instalaré Apache, luego verificaré que módulos vienen instalados por defecto, si me conviene dejarlos instalados o no, igual tengo que desinstalar alguno y tendré que investigar cuales son los módulos nuevos a instalar.
Muy bien, pues lo dicho: ¡Manos a la obra!

La importancia de un servidor web radica en su: estabilidad, disponibilidad y escalabilidad. Es muy importante poder dotar al servidor web de nuevas funcionalidades de forma sencilla, así como del mismo modo quitárselas. Es por esto que la posibilidad que nos otorga el servidor web Apache mediante sus módulos sea uno de los servidores web más manejables y potentes que existen: que necesito soporte SSL pues módulo SSL, que necesito soporte PHP pues módulo PHP, que necesito soporte LDAP pues módulo LDAP, que necesito...
En Debian, y derivados, existen dos comandos fundamentales para el funcionamiento de los módulos en el servidor web Apache: **`a2enmod`** y **`a2dismod`**.

- ✓ **`a2enmod`:** Utilizado para habilitar un módulo de apache. Sin ningún parámetro preguntará que módulo se desea habilitar. Los ficheros de configuración de los módulos disponibles están en **`/etc/apache2/mods-available/`** y al habilitarlos se crea un enlace simbólico desde **`/etc/apache2/mods-enabled/`** .
- ✓ **`a2dismod`**: Utilizado para deshabilitar un módulo de Apache. Sin ningún parámetro preguntará que módulo se desea deshabilitar. Los ficheros de configuración de los módulos disponibles están en `/etc/apache2/mods-available/` y al deshabilitarlos se elimina el enlace simbólico desde `/etc/apache2/mods-enabled/` .
- ✓ Si no dispones de esos comandos para poder habilitar y deshabilitar módulos Apache simplemente haces lo que ellos: crear los enlaces simbólicos correspondientes desde `/etc/apache2/mods-enabled/` hasta `/etc/apache2/mods-available/`.

`a2ensite` es un comando (en Debian y derivados) para habilitar configuraciones de ''sitios web'' en Apache2. Los ficheros de configuración de los ''sitios web'' disponibles (normalmente son configuraciones de hosts virtuales) están en **`/etc/apache2/sites-available/`** y al habilitarlos se crea un enlace simbólico desde **`/etc/apache2/sites-enabled/`**

Puedes consultar más información en la documentación de Apache sobre módulos.
http://httpd.apache.org/docs/2.2/es/mod/

La instalación o desinstalación de un módulo no implica la desinstalación de Apache o la nueva instalación de Apache perdiendo la configuración del servidor en el proceso, simplemente implica la posibilidad de poder trabajar en Apache con un nuevo módulo o no.

3.1.- Operaciones sobre módulos.

"Me lo contaron y lo olvidé. Lo vi y lo entendí. Lo hice y lo aprendí."
Confucio

Los módulos de Apache puedes instalarlos, desinstalarlos, habilitarlos o deshabilitarlos, así, puedes tener un módulo instalado pero no habilitado. Esto quiere decir que aunque instales módulos hasta que los habilites no funcionarán.

En la tabla siguiente encontrarás un resumen de operaciones, ejemplos y comandos necesarios que se le pueden realizar a los módulos:

Operaciones sobre módulos Apache en un GNU/Linux Debian

Operaciones sobre módulos Apache en un en un GNU/Linux Debian	
Instalar un módulo	**Ejemplo: Instalar el módulo ssl**
`apt-get install nombre-modulo`	`apt-get install libapache2-mod-gnutls`
Desinstalar un módulo	**Ejemplo: Desinstalar el módulo ssl**
`apt-get remove nombre-modulo`	`apt-get remove libapache2-mod-gnutls`
Habilitar un módulo	**Ejemplo: Habilitar el módulo ssl**
`a2enmod nombre-modulo-apache`	`a2enmod ssl`
Deshabilitar un módulo	**Ejemplo: Deshabilitar el módulo ssl**
`a2dismod nombre-modulo-apache`	`a2dismod ssl`

Para habilitar un módulo Apache, en Debian, también puedes ejecutar el comando **`a2enmod`** sin parámetros. La ejecución de esté comando ofrecerá una lista de módulos a habilitar, escribes el módulo en cuestión y el módulo se habilitará. Del mismo modo para deshabilitar un módulo Apache, en Debian, puedes ejecutar el comando **`a2dismod`** sin parámetros. La ejecución de esté comando ofrecerá una lista de módulos a deshabilitar, escribes el módulo en cuestión y el módulo se deshabilitará.

Una vez habilitado o deshabilitado los módulos Apache sólo reconocerá estos cambios cuando recargues su configuración, con lo cual debes ejecutar el comando:**`/etc/init.d/apache2 restart`**

Si la configuración es correcta y no quieres reiniciar Apache puedes recargar la configuración mediante el comando: **`/etc/init.d/apache2 reload`**.

Si no dispones de los comandos **`a2enmod`** y **`a2dismod`** puedes habilitar y deshabilitar módulos Apache creando los enlaces simbólicos correspondientes desde **`/etc/apache2/mods-enabled/`** hasta **`/etc/apache2/mods-available/`**, por ejemplo si quisieras habilitar el módulo ssl:

1. Te sitúas en el directorio /etc/apache2/mods-available.
   `cd /etc/apache2/mods-available`
2. Verificas que el módulo aparece en esta ruta y por lo tanto está instalado
   `ls ssl.*`
   Este comando debe listar dos ficheros: ssl.conf (la configuración genérica del módulo) y ssl.load (la librería que contiene el módulo a cargar)
3. Creas el enlace simbólico para habilitar el módulo:
   ```
   ln -s /etc/apache2/mods-available/ssl.load /etc/apache2/mods-enabled/ssl.load
   ln -s /etc/apache2/mods-available/ssl.conf /etc/apache2/mods-enabled/ssl.conf
   ```
   Estos comandos crean los enlaces **`/etc/apache2/mods-enabled/ssl.conf`** y **`/etc/apache2/mods-enabled/ssl.load`** que apuntan a los ficheros **`/etc/apache2/mods-available/ssl.conf`** y **`/etc/apache/mods-available/ssl.load`** respectivamente.
4. Recargas la configuración de Apache:
   `/etc/init.d/apache2 restart`
5. El módulo ssl ya está habilitado.
   Y si quisieras deshabilitarlo, simplemente eliminas en **`/etc/apache2/mods-enabled`** los enlaces simbólicos creados, así si quisieras deshabilitar el módulo `ssl` ejecutarías el siguiente comando:
   `rm -f /etc/apache2/mods-enabled/ssl.*`

Por último, no te olvides recargar la configuración de Apache:**`/etc/init.d/apache2 restart`**

4.- Acceso a carpetas seguras.

Caso práctico

En el transcurso del proyecto sobre aplicaciones web de varias sucursales para una empresa en las oficinas de la empresa BK Programación tuvo lugar la siguiente charla:
Bien, -le dijo Ana a Ada-, ya tenemos casi configurado el servidor web Apache.
- ¿Entonces?- preguntó Ada-
-Nos falta la configuración de la navegación de forma segura, para que la comunicación viaje cifrada.
-¿Os llevará mucho tiempo?
-Bueno...

"Depende qué tan hombre eres."

Miguel de Icaza

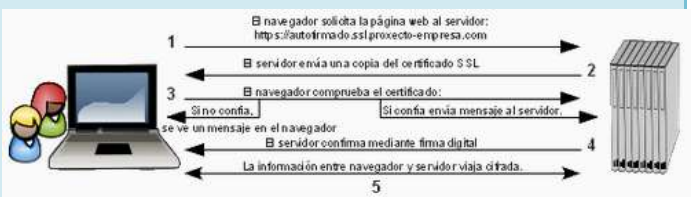
¿Todas las páginas web que están alojadas en un sitio deben ser accesibles por cualquier usuario? ¿Todas las accesibles deben enviar la información sin cifrar, en texto claro? ¿Es necesario que todo el trasiego de información navegador-servidor viaje cifrado?

Existe la posibilidad de asegurar la información sensible que viaja entre el navegador y el servidor, pero esto repercutirá en un mayor consumo de recursos del servidor, puesto que asegurar la información implica en que ésta debe ser cifrada, lo que significa computación algorítmica.

El cifrado al que nos referimos es el cifrado de clave pública o asimétrico: **clave pública (kpub)** y **clave privada (kpriv)**. La **kpub** interesa publicarla para que llegue a ser conocida por cualquiera, la **kpriv** no interesa que nadie la posea, solo el propietario de la misma. Ambas son necesarias para que la comunicación sea posible, una sin la otra no tiene sentido, así una información cifrada mediante la **kpub** solamente puede ser descifrada mediante la **kpriv** y una información cifrada mediante la **kpriv** solo puede ser descifrada mediante la **kpub**.

http://www.criptored.upm.es/intypedia/video.php?id=criptografia-asimetrica&lang=es

En el cifrado asimétrico podemos estar hablando de individuos o de máquinas, en nuestro caso hablamos de máquinas y de flujo de información entre el **navegador (A)** y el **servidor web (B)**. Ver la siguiente tabla como ejemplo de funcionamiento del cifrado asimétrico:

Funcionamiento del cifrado asimétrico.	
	
A(inf) → inf cifrada → B [descifrar inf] → B(inf) = A(inf) **A(inf) → inf cifrada = [(inf)]kpubB → B [inf. cifrada]kprivB → B(inf) = A(inf)**	
Identificación	
A	Navegador web.
inf cifrada = [(inf)]kpubB	Información cifrada mediante la clave pública de B obtenida a través de un certificado digital.
[inf. cifrada]kprivB	Información descifrada mediante la clave privada de B.
B	Servidor web.

Como ves, **A** envía la información cifrada mediante la **kpubB** y **B** la descifra mediante su clave privada**(kprivB)**, por lo que se garantiza la confidencialidad de la información. Pero, ¿estás seguro

que B es quién dice que es? ¿Es quién debe ser? ¿Cómo garantizas la autenticidad de B? Pues ya que supones que B es quien dice ser mediante un certificado digital, debes confiar en ese certificado, así ¿quién emite certificados digitales de confianza? Igual que el DNI es emitido por un entidad certificadora de confianza, el Ministerio del Interior, en Internet existen autoridades de certificación(CA ó AC) que aseguran la autenticidad del certificado digital, y así la autenticidad de B, como: VeriSign y Thawte. Pero, como ya hemos comentado el Servidor Web Apache permite ser CA, por lo que tienes la posibilidad de crear tus propios certificados digitales, ahora bien, ¿el navegador web(A) confiará en estos certificados? Pues, en principio no, por lo que los navegadores avisarán que la página a la cuál intentas acceder en el servidor web representa un peligro de seguridad, ya que no existe en su lista de autoridades certificadoras de confianza. En determinados casos, por imagen, puede ser un problema, pero si la empresa posee una entidad de importancia reconocida o el sitio es privado y no público en Internet o sabes el riesgo que corres puedes aceptar la comunicación y el flujo de información viajará cifrado.

4.1.- Certificados digitales, AC y PKI.

Un certificado digital es un documento electrónico que asocia una clave pública con la identidad de su propietario, individuo o máquina, por ejemplo un servidor web, y es emitido por autoridades en las que pueden confiar los usuarios. Éstas certifican el documento de asociación entre clave pública e identidad de un individuo o máquina (servidor web) firmando dicho documento con su clave privada, esto es, mediante firma digital.

La idea consiste en que los **dos extremos de una comunicación, por ejemplo cliente (navegador web) y servidor (servidor web Apache) puedan confiar directamente entre sí, si ambos tienen relación con una tercera parte, que da fe de la fiabilidad de los dos, aunque en la práctica te suele interesar solamente la fiabilidad del servidor, para saber que te conectas con el servidor que quieres y no con otro servidor -*supuestamente cuando tú te conectas con el navegador al servidor eres tú y no otra persona la que establece la conexión*-.** Así la necesidad de una **Tercera Parte Confiable (TPC ó TTP, Trusted Third Party)** es fundamental en cualquier entorno de clave pública. La forma en que esa tercera parte avalará que el certificado es de fiar es mediante su firma digital sobre el certificado. Por tanto, podremos confiar en cualquier certificado digital firmado por una tercera parte en la que confiamos. La **TPC**que se encarga de la firma digital de los certificados de los usuarios de un entorno de clave pública se conoce con el nombre de **Autoridad de Certificación (AC)**.

El modelo de confianza basado en **Terceras Partes Confiables** es la base de la definición de las **Infraestructuras de Clave Pública (ICP o PKIs, Public Key Infrastructures)**. Una Infraestructura de Clave Pública es un conjunto de protocolos, servicios y estándares que soportan aplicaciones basadas en criptografía de clave pública.

Algunos de los servicios ofrecidos por una **ICP (PKI)** son los siguientes:

- ✓ Registro de claves: emisión de un nuevo certificado para una clave pública.
- ✓ Revocación de certificados: cancelación de un certificado previamente emitido.
- ✓ Selección de claves: publicación de la clave pública de los usuarios.
- ✓ Evaluación de la confianza: determinación sobre si un certificado es válido y qué operaciones están permitidas para dicho certificado.
- ✓ Recuperación de claves: posibilidad de recuperar las claves de un usuario.

Las **ICP (PKI)** están compuestas por:

- ✓ **Autoridad de Certificación (AC)**: realiza la firma de los certificados con su clave privada y gestiona la lista de certificados revocados.
- ✓ **Autoridad de Registro (AR)**: es la interfaz hacia el mundo exterior. Recibe las solicitudes de los certificados y revocaciones, comprueba los datos de los sujetos que hacen las peticiones y traslada los certificados y revocaciones a la **AC** para que los firme.

Existen varios formatos para certificados digitales, pero los más comúnmente empleados se rigen por el estándar UIT-T (*es la organización de las Naciones Unidas para las tecnologías de la información y la comunicación. En su calidad de coordinador mundial de gobiernos y sector privado, la función de la UIT abarca tres sectores fundamentales, a saber: radiocomunicaciones, normalización y desarrollo*) X.509. El certificado X.509 contiene los siguientes campos: versión, nº de serie del certificado, identificador del algoritmo de firmado, nombre del emisor, periodo de validez, nombre del sujeto, información de clave pública del sujeto, identificador único del emisor, identificador único del sujeto y extensiones.

4.2.- Módulo ssl para apache.

Todos los días los bancos efectúan transferencias bancarias, así como también aceptan conexiones a sus páginas web para ofrecer su servicio online. ¿Qué pasaría si cualquiera pudiese interceptar una comunicación bancaria de ese tipo? ¿Sería interesante cifrar la información efectuada antes y durante la conexión bancaria?

El método de cifrado SSL/TLS utiliza un método de cifrado de clave pública (cifrado asimétrico) para la autenticación del servidor.

El **módulo ssl** es quien permite cifrar la información entre navegador y servidor web. En la instalación por defecto éste módulo no viene activado, así que debes ejecutar el siguiente comando para poder activarlo: `a2enmod ssl`

Este módulo proporciona SSL v2/v3 y TLS v1 para el Servidor Apache HTTP; y se basa en Open SSL (*Paquete de herramientas de administración y bibliotecas relacionadas con la criptografía, que suministran funciones criptográficas, entre otros, a navegadores web para acceso seguro a sitios mediante el protocolo HTTPS*) para proporcionar el motor de la criptografía.

En el siguiente enlace puedes encontrar más información sobre el módulo ssl
http://httpd.apache.org/docs/2.2/es/mod/mod_ssl.html

¿Cómo harías, en Debian 6, para deshabilitar el módulo ssl de Apache?
Mediante el comando `a2dismod ssl` y recargando el servicio Apache: `/etc/init.d/apache2 reload ó /etc/init.d/apache2 restart`.
Y ¿cómo lo harías si no dispones del comando para Debian?
Quitando los enlaces existentes en `mods-enabled` que apunten a los archivos ssl correspondientes de la carpeta `mods-available`, por ejemplo en un Debian 6 eliminarías los archivos: `ssl.conf` y `ssl.load` situados en `/etc/apache2/mods-enabled/` y recargando el servicio Apache: `/etc/init.d/apache2 reload` ó `/etc/init.d/apache2 restart`

4.3.- Crear un servidor virtual seguro en Apache (I).

En Debian, Apache posee por defecto en su instalación el fichero `/etc/apache2/sites-available/default-ssl`, que contiene la configuración por defecto de SSL. En su contenido podemos ver las siguientes líneas:

```
SSLEngine on
SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
```

donde,

`SSLEngine on` : Activa o desactiva SSL

`SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem` : Certificado digital del propio servidor Apache

`SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key` : Clave privada del servidor Apache.

Esas líneas lo que quieren decir es que Apache permite conexiones SSL y posee un certificado digital autofirmado por sí mismo -ya que Apache actúa como entidad certificadora-.

Cuando activaste el módulo ssl, mediante el comando a2enmod ssl permitiste que Apache atienda el protocolo SSL. Así, si ahora lanzas el navegador **Firefox** con la dirección de tu servidor web Apache mediante el protocolo HTTPS, verás una imagen similar a la siguiente:

Lo que indica que el certificado digital del servidor no viene firmado por una **AC** contenida en la lista que posee el navegador, sino por el mismo Apache. Si lo compruebas haciendo clic en **Detalles Técnicos** verás algo similar a:

```
192.168.200.250 usa un certificado de seguridad no válido.

     No se confía en el certificado porque está autofirmado.
     El certificado sólo es válido para debian-servidor-fp.

     (Código de error: sec_error_untrusted_issuer)
```

Ahora tienes dos opciones: Confiar en el certificado o no.

- ✓ Si confías haces clic en **Entiendo los riesgos y Añadir excepción...**
  Una vez que confías puedes, antes de **Confirmar excepción de seguridad,** ver el contenido del certificado. Si estás de acuerdo la comunicación se establece y la información viaja cifrada.
- ✓ Si no confías haces clic en **!Sácame de aquí!**

¿Pero...? Como eres AC puedes firmar certificados e incluso puedes generar también tu propio certificado autofirmado similar al que viene por defecto en Apache.

Hay que tener en cuenta que la negociación SSL es dependiente totalmente de la IP, no del nombre del sitio web, así no puedes servir distintos certificados en una misma IP.

4.3.1.- Crear un servidor virtual seguro en Apache (II).

En una distribución Debian 6 el procedimiento para generar un certificado digital sería el siguiente:

1. Instalación del paquete openssl
   ```
   apt-get install openssl
   ```
2. Crear un certificado autofirmado para el servidor web.
   ```
   mkdir /etc/apache2/tus-ssl/
   make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/apache2/tus-ssl/apache.pem
   ```
   Cuando se solicite el nombre del servidor HTTP indicar el nombre DNS que corresponda a la IP del certificado, por ejemplo: `autofirmado.ssl.empresa-proyecto.com`

   El nombre de dominio autofirmado.ssl.empresa-proyecto.com debe resolverse a una IP mediante un servidor DNS o en su defecto mediante el fichero `/etc/hosts`.

   El fichero generado `/etc/apache2/tus-ssl/apache.pem` contiene tanto el certificado del servidor como la clave privada asociada al mismo.

El comando, de Debian, `make-ssl-cert` permite generar certificados autofirmados para pruebas. Los datos de configuración del certificado a generar se indican en `/usr/share/ssl-cert/ssleay.cnf`. Internamente hace uso de las utilidades de la librería `openssl`.

3. Editar la configuración SSL por defecto en el archivo `/etc/apache2/sites-available/default-ssl` para indicar el certificado del servidor y su respectiva clave privada asignando los siguientes valores a los parámetros :

```
SSLEngine on
SSLCertificateFile /etc/apache2/tus-ssl/apache.pem
SSLCertificateKeyFile /etc/apache2/tus-ssl/apache.pem
```

4. Asegúrate que el fichero `/etc/apache2/ports.conf` incluya el valor `Listen 443`
5. Habilita el soporte SSL en Apache y habilita la configuración SSL por defecto:

```
a2enmod ssl
a2ensite default-ssl
/etc/init.d/apache2 restart
```

En el equipo cliente, **192.168.200.100**, lanzas el navegador:

1. Indicar `https://autofirmado.ssl.empresa-proyecto.com` en la barra de direcciones.
2. Dará un aviso de que la AC que firma el certificado del servidor no está reconocida. Añadir la correspondiente excepción de seguridad y permitir la descarga y aceptación del certificado. Antes de aceptarlo puedes ver el contenido del certificado:

4.3.2.- Crear un servidor virtual seguro en Apache (III).

De forma genérica, por si no posees el comando **make-ssl-cert**, puedes emplear el comando **openssl** para generar los certificados. Por ejemplo:

1. Instalación del paquete openssl:
   `apt-get install openssl`
2. Genera el certificado y la clave privada de tu autoridad de certificación (AC)

```
mkdir /etc/apache2/tus-ssl/
cd /etc/apache2/tus-ssl/
```

Puedes ver la ejecución del los dos comandos que se utilizan en el anexo **openssl_autofirmado.txt**

```
openssl req -new -nodes -keyout tupaginaweb.key -out tupaginaweb.csr
```

Este comando genera dos archivos:

- ✓ La clave privada con el que firmarás tus futuros certificados: tupaginaweb.key
- ✓ El certificado con la clave pública de la AC: tupaginaweb.csr

Este comando pedirá algunos datos: nombre de empresa, país, contraseña... La contraseña, puedes omitirla, pero por seguridad es conveniente crearla para utilizarla cuando firmes un certificado SSL .

3. Autofirma el certificado. Puedes hacerlo porque eres una AC, de tal forma que el primer certificado que firmas es el de tu propia AC.

```
openssl x509 -in tupaginaweb.csr -out tupaginaweb.crt -req -signkey tupaginaweb.key -days 3650
```

El campo `days 3650` significa que el certificado de tu AC tardará 10 años en caducar.

4. Editar la configuración SSL por defecto en el archivo `/etc/apache2/sites-available/default-ssl` para indicar el certificado del servidor y su respectiva clave privada asignando los siguientes valores a los parámetros :

```
SSLEngine on
SSLCertificateFile /etc/apache2/tus-ssl/tupaginaweb.crt
SSLCertificateKeyFile /etc/apache2/tus-ssl/tupaginaweb.key
```

5. Asegúrate que el fichero `/etc/apache2/ports.conf` incluya el valor `Listen 443`

6. Habilita el módulo ssl y la configuración SSL por defecto.

```
a2enmod ssl
a2ensite default-ssl
/etc/init.d/apache2 restart
```

En el archivo `/usr/share/doc/apache2.2-common/README.Debian.gz` encontrarás información sobre como configurar SSL y crear certificados autofirmados.

4.4.- Comprobar el acceso seguro al servidor.

"La manera de estar seguro es no sentirse nunca seguro."

Proverbio español

A continuación una serie de actuaciones que te servirán para comprobar que el acceso seguro que estableces con el servidor seguro es el esperado:

- ✓ Siempre que te conectes mediante SSL a una página web y el certificado no sea admitido, debes ver los campos descriptivos del certificado antes de generar la excepción que te permita visitar la página.
- ✓ Debes comprobar en el certificado si la página a la que intentas acceder es la misma que dice el certificado.
- ✓ Típicamente en los navegadores, si no está configurado lo contrario, cuando accedes mediante cifrado SSL a una página web puedes ver en algún lugar del mismo un icono: un candado, por lo cual debes verificar su existencia para asegurarte que estás accediendo por https.
  Incluso si el certificado pertenece a alguna AC que el navegador posee en su lista de AC puedes ver en la barra de direcciones indicaciones del tipo de certificado con el que se cifra la comunicación.
- ✓ Revisar la lista de certificados admitidos que posee tu navegador. En **Firefox**, versión > 3.x , donde x es el número de revisión de la versión 3, puedes verlas dirigiéndote por las pestañas a:
  `Editar → Preferencias → Avanzado → Cifrado → Ver certificados`
  Revisar la lista de revocaciones que posee tu navegador. En **Firefox**, versión > 3.x , donde x es el número de revisión de la versión 3, puedes verlas dirigiéndote por las pestañas a:
  `Editar → Preferencias → Avanzado → Cifrado → Listas de revocación`

Puedes **Importar/Exportar** certificados en los navegadores, con lo cual los puedes llevar a cualquier máquina. Esto es muy útil cuando necesitas un certificado personal en máquinas distintas.

En el siguiente enlace encontrarás información muy interesante, amena y explicativa sobre la seguridad de la información y el cifrado.

http://www.criptored.upm.es/intypedia/index.php?lang=es

5.- Autenticación y control de acceso.

Caso práctico
La reunión tuvo lugar.
El equipo de BK Programación destinado al proyecto de aplicaciones web para varias sucursales de una empresa llegó a un acuerdo para la autenticación y el control de acceso sobre la aplicación de panel de control. Se barajaron varias alternativas: usuarios del sistema, ficheros de usuarios, base de datos SQL y LDAP. Al final se decantaron por dos opciones: ficheros de usuarios para el estado de pruebas y LDAP para la aplicación definitiva, con lo cual establecieron el siguiente protocolo de actuación:

- ✓ *En la aplicación de desarrollo montada por María se realizarán las pruebas, siendo los encargados de las mismas Antonio y Carlos.*
- ✓ *El diseño web de la aplicación recaerá en Ana: banners, logos ...*
- ✓ *Juan se dedicará a la programación del panel de control: autenticación por medio de LDAP*
- ✓ *La encargada de montar el servicio LDAP, integrarlo en Apache y conseguir el control de acceso fue María.*

Ante la espera que María instale y configure Apache con LDAP, y con ello imposibilidad de probar la autenticación por LDAP, María crea un fichero de usuarios para autenticacarse en la aplicación y todos empiezan a trabajar en el resto de las cosas.

Puede que interese impedir el acceso a determinadas páginas ofrecidas por el servidor web, así: ¿crees que a una empresa le interesaría que cualquiera tuviera acceso a determinada información confidencial?, o puede que interese controlar el acceso hacia un servicio a través de la web, como el correo electrónico. Para este tipo de casos tenemos que pensar en la autenticación y el control de acceso.

Cuando nos autenticamos en una web suele transferirse la información de autenticación a una base de datos, que puede existir en la misma máquina que el servidor web o en otra totalmente diferente. Suelen emplearse bases de datos SQL o LDAP para la autenticación de usuarios, siendo OpenLDAP (http://www.openldap.org/) una de las alternativas más empleadas.

Puedes visitar el enlace de wikipedia AAA (*Autenticación, Autorización y Registro. Conjunto de herramientas, procedimientos y protocolos que garantizan un tratamiento coherente de las tareas de autenticación, autorización y registro de actividad de las entidades que tienen acceso a un sistema de información*) donde encontrarás más información referente a la autenticación:

http://es.wikipedia.org/wiki/Protocolo_AAA

HTTP proporciona un método de autenticación básico de usuarios: **basic**. Este método ante una petición del cliente (navegador web) al servidor cuando se solicita una URL mostrará un diálogo pidiendo usuario y contraseña. Una vez autenticado el usuario, el cliente volverá a hacer la petición al servidor pero ahora enviando el usuario y contraseña, en texto claro (sin cifrar) proporcionados en el diálogo. Es recomendable entonces si empleas este método que lo hagas combinado con conexión SSL (HTTPS).

En la autenticación HTTP Basic es muy típico utilizar archivos .htaccess en los directorios que queremos controlar el acceso. Puedes encontrar un ejemplo sobre basic con https en el archivo `virtualhost-ssl-basic`:

```
<IfModule mod_ssl.c>
#<VirtualHost  default :443>
<VirtualHost 192.168.200.250:443>
   ServerAdmin web-autenticacion@empresa-proyecto.com
   ServerName web-con-autenticacion-basic.empresa-proyecto.com
   DocumentRoot /var/www/web-con-autenticacion-basic
   <Directory /var/www/web-con-autentitacion-basic/>
      AllowOverride AuthConfig
```

```
      Options Indexes FollowSymLinks MultiViews
      Order allow,deny
      allow from all
   </Directory>
   ErrorLog ${APACHE_LOG_DIR}/error-web-autenticacion-basic.log
   LogLevel warn
   CustomLog ${APACHE_LOG_DIR}/ssl_access-web-autenticacion-basic.log combined
   SSLEngine on
   SSLCertificateFile    /etc/apache2/tus-ssl/tupaginaweb.crt
   SSLCertificateKeyFile /etc/apache2/tus-ssl/tupaginaweb.key
</VirtualHost>
</IfModule>
```

y un ejemplo sobre .htaccess en el archivo `htaccess` :

```
AuthType Basic
AuthName "Web con Autenticacion Basic"
AuthUserFile /etc/apache2/web.htpasswd
##Require valid-user
Require user user1
```

Para usar archivos **.htaccess**, necesitas tener una configuración en el servidor que permita poner directivas de autenticación en estos archivos, mediante la directiva **`AllowOverride`**, así:

```
AllowOverride AuthConfig
```

Puedes visitar el siguiente enlace donde encontrarás más información referente a la autenticación http basic:

http://httpd.apache.org/docs/2.0/es/howto/auth.html

También se puede controlar el acceso mediante IP. Puedes encontrar un ejemplo en el archivo **virtualhost-control-por-IP**:

```
<VirtualHost IP_Servidor_Web:80>
   Alias /carpeta-controlada "/usr/srv/control/carpeta-controlada/"
   <Directory "/usr/srv/control/carpeta-controlada/">
      Order deny,allow
      Deny from all
      Allow from IP permiso concedido
   </Directory>
   DocumentRoot /usr/srv/control/carpeta-controlada
   ServerName www.empresa.com.
   ServerAlias empresa.com
</VirtualHost>
```

5.1.- Autenticar usuarios en apache mediante LDAP.

Se ha comentado en el apartado anterior que el servidor web Apache permite la autenticación de usuarios mediante LDAP. Esto es posible mediante los módulos **`ldap`** y **`authnz_ldap`**.

En **este anexo** se encuentra cómo instalar y configurar un servidor OpenLDAP pero también deberías visitar la siguiente web

http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m6/instalacin_y_configuracin_de_openldap.html

Para una instalación de OpenLDAP en Debian 6 visita el enlace al siguiente documento:

Instalación y configuración del servidor OpenLDAP en Debian 6

En el siguiente enlace encontrarás más información sobre la autenticación LDAP para el servidor web Apache mediante el módulo `authnz_ldap`.

http://httpd.apache.org/docs/2.2/mod/mod_authnz_ldap.html

Para el buen funcionamiento de lo expuesto a continuación se asume que tanto Apache2 como OpenLDAP están instalados y configurados:

1. Habilita el soporte LDAP para Apache2:

   ```
   a2enmod authnz_ldap
   /etc/init.d/apache2 restart
   ```

2. Configura el virtualhost **`autenticacion-ldap-apache`** como sigue:

```
<VirtualHost *:80>
   DocumentRoot /var/www/autenticacion-ldap
   ServerName www.empresa-proyecto.panel-de-control.com
   ServerAlias www.autenticacion-ldap.empresa-proyecto.com
   <Directory /var/www/autenticacion-ldap>
      AllowOverride All
   </Directory>
   ErrorLog /var/log/apache2/error-autenticacion-ldap.log
   LogLevel warn
   CustomLog /var/log/apache2/access-autenticacion-ldap.log combined
</VirtualHost>
```

La directiva **`AllowOverride All`** es necesaria para habilitar ficheros .htaccess

3. Crea el fichero **`/var/www/autenticacion-ldap/.htaccess`** que permite configurar la autenticación ldap para el virtualhost anterior:

```
AuthName "Autenticacion por LDAP"
AuthType Basic
AuthBasicProvider ldap
AuthzLDAPAuthoritative on
AuthLDAPUrl ldap://127.0.0.1/ou=usuarios,dc=proyecto,dc=com?uid
Require ldap-user user1LDAP
```

La directiva **`Require ldap-user admin`** permite la autenticación al usuario **`user1LDAP`**, todos los demás usuarios tienen el acceso denegado.

4. Accede a la URL: **`www.empresa-proyecto.panel-de-control.com`** ó **`www.autenticacion-ldap.empresa-proyecto.com`**

6.- Monitorización del acceso: Archivos de registro (logs).

Caso práctico
¿Qué, quién, dónde, cuándo, por qué ha pasado? Eso es lo que queremos saber en todo momento -comentó María-. Recordad que es necesario guardar los archivos de registro al menos durante 1 año según la LSSI/CE. Tenemos que estar preparados ante cualquier petición de los logs (requerimiento judicial) por parte de las administraciones. Es por esto que tú, Antonio, vas a realizar una batería de pruebas: accesos a páginas existentes y no existentes, búsqueda de listado de ficheros y no solamente el index.html, accesos no permitidos a bases de datos, accesos controlados por IP, por usuario, etc.
Muy bien, eso está hecho -dijo Antonio-.

Tan importante como es configurar un servidor web lo es mantener y comprobar su correcto funcionamiento, y para ello debes ayudarte de los logs o archivos de registro que te permiten revisar y estudiar su funcionamiento

Apache permite mediante diversas directivas crear archivos de registro que guardarán la información correspondiente a las conexiones con el servidor. Esta información es guardada en formato CLF **(Common Logon Format)** por defecto. Ésta es una especificación utilizada por los servidores web para hacer que el análisis de registro entre servidores sea mucho más sencillo, de tal forma que independientemente del servidor web utilizado podamos emplear el mismo método de análisis de registro, ya sea mediante lectura, mediante programas ejecutables (scripts) o mediante programas propios de análisis de registro.

En un archivo de registro en formato CLF cada línea identifica una solicitud al servidor web. Esta línea contiene varios campos separados con espacios**.** Cada campo sin valor es identificado con un guión (-). Los campos empleados en una configuración por defecto de Apache2 son los definidos en la siguiente tabla:

Ejemplo log Apache en formato CLF		
192.168.200.100 - - [05/May/2011:17:19:18 +0200] "GET /index.html HTTP/1.1" 200 20		
Campos (especificadores)	**Definición**	**Ejemplo**
`host (%h)`	Identifica el equipo cliente que solicita la información en el navegador.	192.168.200.100
`ident (%l)`	Información del cliente cuando la máquina de éste ejecuta `identd` y la directiva `IdentityCheck` está activada.	
`authuser ( %u)`	Nombre de usuario en caso que la URL solicitada requiera autenticación HTTP.	
`date ( %t)`	Fecha y hora en el que se produce la solicitud al servidor. Va encerrado entre corchetes. Este campo tiene su propio formato: `[dia/mes/año:hora:minuto:segundo zona]`	[05/May/2011:17:19:18 +0200]
`request (%r)`	Petición del cliente, esto es, la página web que está solicitando. En el ejemplo: `/index.html`, esto es, dentro de la raíz del dominio que se visite la página	/index.html
`status ( %s ó %>s)`	Identifica el código de estado HTTP de tres dígitos que se devuelve al cliente.	200
`Bytes (%b)`	Sin tener en cuenta las cabeceras HTTP el número de bytes devueltos al cliente.	20

Cada campo tiene su especificador, el cual se emplea en las directivas de Apache para indicar que campo queremos registrar.

6.1.- Directivas para archivos de registro.

El contexto de aplicación de todas las directivas que se indican a continuación en la siguiente tabla puede ser el de la configuración principal del servidor así como el de la configuración de los host virtuales.

Directivas para archivos de registro.	
Directivas	**Definicion**
`TransferLog`	Directiva que define el nombre del archivo de registro o al programa al que se envía la información de registro. Emplea los especificadores asignados por la directiva `LogFormat`.
`LogFormat`	Directiva que define el formato del archivo de registro asignado con la directiva `TransferLog`
`ErrorLog`	Directiva que permite registrar todos los errores que encuentre Apache. Permite guardar la información en un archivo de registro o bien en `syslog`
`CustomLog`	Directiva similar a la directiva `TransferLog`, pero con la particularidad que permite personalizar el formato de registro empleando los especificadores anteriormente vistos.
`CookieLog`	Directiva que define el nombre del archivo de registro donde registrar información sobre cookies

La tabla siguiente muestra la sintaxis y el uso de las anteriores directivas:

Sintaxis y uso de directivas para archivos de registro	
Directiva TransferLog	
Sintaxis	`TransferLog nombre_fichero_archivo_registro | tubería_para_enviar_al_programa_la_información de_registro`
Uso	`TransferLog logs/acceso a empresa1.log`
Directiva LogFormat	
Sintaxis	`LogFormat nombre_fichero_archivo_registro [opcional_alias] [opcional_alias] permite definir un logformat con un nombre de tal forma que cuando hacemos referencia al nombre lo hacemos al logformat vinculado.`
Uso	`LogFormat logs/acceso_a_empresa1.log`
Directiva ErrorLog	
Sintaxis	`ErrorLog nombre_fichero_archivo_registro`
Uso	`ErrorLog logs/acceso a empresa1.log`
Directiva CustomLog	
Sintaxis	`CustomLog nombre_fichero_archivo_registro| tubería_para_enviar_al_programa_la_información de_registro [variable_de_entorno_opcional]`
Uso	`CustomLog logs/acceso_a_empresa1.log`
Directiva CookieLog	
Sintaxis	`CookieLog nombre_fichero_archivo_registro`
Uso	`CookieLog logs/acceso a empresa1.log`

En **GNU/Linux** puedes **comprobar en tiempo real** desde un terminal en el equipo que guarda los logs *-que puede ser el propio equipo servidor web-* que es lo que ocurre cuando accedes a una página web observando el contenido de los archivos de registro mediante el comando: `tail -f nombre_archivo_de_registro.log`

6.2.- Rotación de los archivos de registro (I).

Como los archivos de registro a medida que pasa el tiempo van incrementando su tamaño, debe existir una política de mantenimiento de registros para que éstos no consuman demasiados recursos

en el servidor, así es conveniente rotar los archivos de registro, esto es, hay que depurarlos, comprimirlos y guardarlos. Básicamente tienes dos opciones para rotar tus registros: **`rotatelogs`** un programa proporcionado por Apache, o **`logrotate`**, una utilidad presente en la mayoría de los sistemas GNU/Linux.

No debes olvidar que la información recopilada en los **`ficheros log`** se debe conservar al menos durante 1 año por eventuales necesidades legales, de este modo, además de rotarlos se opta habitualmente por **`comprimir logs`**.

Uso de rotatelogs

```
CustomLog "|ruta_rotatelogs ruta_log_a_rotar numero_segundos|tamaño_máximoMB" alias_logformat
```

Ejemplos

Rotar el archivo de registro access.log cada 24horas

```
CustomLog "|/usr/sbin/rotatelogs /var/log/apache2/access.log 86400" common
```

Rotar el archivo de registro access.log cada vez que alcanza un tamaño de 5 megabytes

```
CustomLog "|/usr/sbin/rotatelogs /var/logs/apache2/access.log 5M" common
```

Rotar el archivo de registro error.log cada vez que alcanza un tamaño de 5 megabytes y el archivo se guardará con el sufijo de formato : YYYY-mm-dd-HH_MM_SS (Año-Mes-Día-Hora_Minutos_Segundos)

```
ErrorLog "|/usr/sbin/rotatelogs /var/logs/errorlog.%Y-%m-%d-%H %M %S 5M" common
```

Los ficheros rotados por intervalo de tiempo, lo harán siempre y cuando en el intervalo de tiempo definido existan nuevos datos.

Por defecto, si no se define formato mediante ningún modificador % para guardar los archivos de registro, el sufijo nnnnnnnnnn (10 cifras) se agrega automáticamente y es el tiempo en segundos traspasados desde las 24 horas (medianoche).

El **`alias logformat`** es muy interesante, porque permite definir un grupo de modificadores en una palabra, de tal forma que incorporando esa palabra en la directiva log correspondiente estás activando todo un grupo de modificadores. En Apache existen predefinidos en el archivo **`/etc/apache2/apache2.conf`** los alias **`logformat`**: `vhost_combined`, `combined`, `common`, `referer` y `agent`, que puedes ver a continuación

Alias predefinidos

Alias logformat predefinidos en /etc/apache2/apache2.conf

```
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined
LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %O" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
```

Es conveniente que le des una visita al manual de **rotatelogs:** `man rotatelogs`**.**

6.2.1.- Rotación de los archivos de registro (II).

El programa **logrotate** rota, comprime y envía archivos de registro a diario, semanalmente, mensualmente o según el tamaño del archivo. Suele emplearse en una tarea diaria del cron (*Programa empleado en sistemas GNU/Linux para la automatización de tareas a intervalos regulares: minutos, horas, días …*).

En **Debian** puedes encontrar los siguientes archivos de configuración para **logrotate**:

- ✓ **/etc/logrotate.conf** : Define los parámetros globales, esto es, los parámetros por defecto de logrotate. Te mostramos un fichero de este tipo:

```
# ejecutar "man logrotate" para más información

# rotar log semanalmente
weekly

# mantener logs durante 4 semanas
rotate 4

# rotar y crear nuevo log aunque esté vació el anterior
create
# descomentar si quieres comprimir logs
#compress

# ubicación de paquetes para el rotado de logs
include /etc/logrotate.d

# los logs wtmp o btmp los haremos rotar aquí
/var/log/wtmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}

/var/log/btmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}

# los logs del sistema se pueden rotar aquí
```

- ✓ **/etc/logrotate.d/apache2** : Define para apache2 el rotado de logs, todos aquellos parámetros que no se encuentren aquí recogen su valor del fichero /etc/logrotate.conf. Puedes ver un archivo tipo a continuación:

```
/var/log/apache2/*.log {
        weekly  → rotar log semanalmente
        missingok
        rotate 52 → mantener los logs durante 52 semanas
        compress → Archivos comprimidos mediante gzip por defecto
        delaycompress
        notifempty
        create 640 root adm -> rotar y crear nuevo log, aunque esté vacio el anterior, con
permisos 640, usuario root y grupo adm
        sharedscripts
        postrotate → Una vez rotado se recarga la configuración de apache2
                /etc/init.d/apache2 reload > /dev/null
        endscript
}
```

Uso de logrotate

Comprobar la correcta configuración de la rotación de un log

```
/usr/sbin/logrotate -d /etc/logrotate.d/apache2
```

Forzar la ejecución de logrotate

```
/usr/sbin/logrotate-f /etc/logrotate.conf
```

/etc/cron.daily/logrotate: Fichero tipo para ejecutar logrotate diariamente en el cron

```
 #!/bin/sh

test -x /usr/sbin/logrotate || exit 0

/usr/sbin/logrotate /etc/logrotate.conf
```

Ejemplo para añadir al archivo crontab del sistema (crontab -e)

```
               # Rotar logs de apache con logrotate a las 3 am

0 03 * * * root /usr/sbin/logrotate /etc/logrotate.conf > /dev/null 2>&1
```

El **rotado de logs** descrito anteriormente lo podemos aplicar a cualquier otra herramienta del sistema. Es conveniente que le des una visita al manual de **logrotate:** `man logrotate`**.**

Busca las palabras escondidas en la sopa de letras

1. Formato archivo de registro
2. Dominios independientes configurables
3. Protocolo criptográfico
4. Sistema de autenticación de usuarios
5. Archivos de registro
6. Verisign, Thawte, …
7. Asegurar http
8. Políticas para los cambios de registro

Z	V	J	M	Q	T	B	Z	A	M	I	V
F	H	I	P	S	U	Y	L	H	S	N	H
V	B	A	R	P	E	C	O	I	R	S	S
A	D	T	U	T	Y	I	G	D	O	P	L
L	X	H	U	P	U	M	S	F	L	X	A
T	J	T	I	N	O	A	E	T	B	D	E
Z	C	L	F	Y	E	U	L	S	Z	X	G
N	H	V	B	N	L	T	S	H	R	K	J
A	C	F	G	B	K	P	B	N	O	T	O
F	T	Y	U	I	T	K	P	E	S	S	W
K	O	R	O	T	A	C	I	O	N	H	T
W	S	X	H	V	G	H	R	U	B	N	Y
F	G	H	J	K	L	R	T	F	I	V	S

CLF
VIRTUALHOST
SSL
LDAP
LOGS
AC
HTTPS
ROTACIÓN

Rellena los huecos con la palabra: `a2ensite` `available` `443` `enabled` `a2enmod` `80`

Servidores web

En apache2 utilizas el comando		para habilitar módulos	a2enmod
El puerto TCP	suele identificar a HTTPS		443
En apache2 utilizas el comando		para habilitar sitios	a2ensite
En apache2 el directorio mods-		contiene los módulos habilitados	enabled
En apache2 el directorio mods-		contiene los módulos posibles	available
El puerto TCP	suele identificar a HTTP		80

7.- Despliegue de aplicaciones sobre servidores Web.

Caso práctico
La empresa ha quedado muy contenta con el proyecto realizado por BK Programación, con lo cual ha considerado la posibilidad de contratarlos para un nuevo proyecto: la creación de una tienda virtual para la venta del material de la empresa a través de Internet. Para ello mantuvieron una reunión con los siguientes integrantes de BK Programación: Ada, la directora de la empresa y Juan el encargado de desarrollo de aplicaciones web.
-Juan -comentó-, pienso que se podría aprovechar para este proyecto varias aplicaciones de software libre, así el costo se abarataría y la comunidad de programadores es una garantía para la estabilidad del proyecto.
-Entonces -preguntó el representante de la empresa-, el desarrollo del proyecto mediante software libre y no la creación de una tienda virtual propia ¿reduciría el costo y el tiempo de desarrollo del proyecto?
-Sí, -dijo Juan-, existen varias aplicaciones de software libre en el mercado para tiendas virtuales, como: OpenCart, Magento, osCommerce.
-¿Cuál nos recomiendas?
-Pues, hoy en día, OpenCart, pero cualquiera de las tres son una buena elección.

Normalmente las aplicaciones sobre servidores web necesitan de los siguientes elementos para su correcto funcionamiento: `soporte php` y `soporte sql`.

El servidor web puede tener soporte php, pero el soporte sql debe ser ofrecido por otro servidor al que pueda acceder el servidor web. Este servidor con soporte sql puede estar configurado en el mismo equipo que el servidor web o en otro.

El procedimiento suele ser el siguiente:
1. Se descarga la aplicación.
2. Se configura para que sea visible a través del servidor web.
3. Suele traer una página de instalación que verifique si el servidor web cumple los requisitos para la instalación de la aplicación.
4. Es necesaria antes de finalizar el proceso de instalación autenticarse al servidor sql con un usuario con permisos para crear/modificar una base de datos. Puede que previamente se tenga que crear la base de datos para que el proceso de instalación genere las tablas necesarias en la misma.
5. Se pide un usuario y contraseña para poder acceder a la aplicación web.
6. Fin de la instalación.

A continuación, en el siguiente documento puedes ver un **ejemplo basado en la aplicación Opencart**.

En este documento se supone que tienes funcionando el siguiente entorno básico: Apache, MySQL y PHP. En Debian 6, instalado Apache, puedes lograrlo con el comando:

```
apt-get install libapache2-mod-auth-mysql mysql-server-5.5 php5-mysql curl php5-curl php5-gd libgd-tools
```

Otra buena opción sería instalar el paquete XAMMP para GNU/Linux:

En los siguientes enlaces encontrarás demos de las aplicaciones para tienda virtual: OpenCart, Magento, osCommerce.
http://www.opencart.com/index.php?route=demonstration/demonstration
http://demo.magentocommerce.com/
http://demo.oscommerce.com/

Anexo I - /etc/apache/sites-available/default

```
<VirtualHost *:80>
   ServerAdmin webmaster@localhost

   DocumentRoot /var/www
   <Directory />
      Options FollowSymLinks
      AllowOverride None
   </Directory>
   <Directory /var/www/>
      Options Indexes FollowSymLinks MultiViews
      AllowOverride None
      Order allow,deny
      allow from all
   </Directory>

   ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
   <Directory "/usr/lib/cgi-bin">
      AllowOverride None
      Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
      Order allow,deny
      Allow from all
   </Directory>

   ErrorLog ${APACHE_LOG_DIR}/error.log

   # Possible values include: debug, info, notice, warn, error, crit,
   # alert, emerg.
   LogLevel warn

   CustomLog ${APACHE_LOG_DIR}/access.log combined

    Alias /doc/ "/usr/share/doc/"
    <Directory "/usr/share/doc/">
        Options Indexes MultiViews FollowSymLinks
        AllowOverride None
        Order deny,allow
        Deny from all
        Allow from 127.0.0.0/255.0.0.0 ::1/128
    </Directory>

</VirtualHost>
```

Anexo II - /etc/mime.types

```
#############################################################################
#
#  MIME-TYPES and the extensions that represent them
#
#  This file is part of the "mime-support" package.  Please send email (not a
#  bug report) to mime-support@packages.debian.org if you would like new types
#  and/or extensions to be added.
#
#  The reason that all types are managed by the mime-support package instead
#  allowing individual packages to install types in much the same way as they
#  add entries in to the mailcap file is so these types can be referenced by
#  other programs (such as a web server) even if the specific support package
#  for that type is not installed.
#
#  Users can add their own types if they wish by creating a ".mime.types"
#  file in their home directory.  Definitions included there will take
#  precedence over those listed here.
#
#  Note: Compression schemes like "gzip", "bzip", and "compress" are not
#  actually "mime-types".  They are "encodings" and hence must _not_ have
#  entries in this file to map their extensions.  The "mime-type" of an
#  encoded file refers to the type of data that has been encoded, not the
#  type of encoding.
#
#############################################################################


application/activemessage
application/andrew-inset                        ez
application/annodex                             anx
application/applefile
application/atom+xml                            atom
application/atomcat+xml                         atomcat
application/atomserv+xml                        atomsrv
application/atomicmail
application/batch-SMTP
application/beep+xml
application/bbolin                              lin
application/cals-1840
application/cap                                 cap pcap
application/commonground
application/cu-seeme                            cu
application/cybercash
application/davmount+xml                        davmount
application/dca-rft
application/dec-dx
application/docbook+xml
application/dsptype                             tsp
application/dvcs
application/ecmascript                          es
application/edi-consent
application/edi-x12
application/edifact
application/eshop
application/font-tdpfr
application/futuresplash                        spl
application/ghostview
application/hta                                 hta
application/http
application/hyperstudio
application/iges
application/index
application/index.cmd
application/index.obj
application/index.response
application/index.vnd
application/iotp
application/ipp
application/isup
application/java-archive                        jar
application/java-serialized-object              ser
application/java-vm                             class
application/javascript                          js
application/m3g                                 m3g
application/mac-binhex40                        hqx
```

```
application/mac-compactpro                      cpt
application/macwriteii
application/marc
application/mathematica                         nb nbp
application/ms-tnef
application/msaccess                            mdb
application/msword                              doc dot
application/mxf                                 mxf
application/news-message-id
application/news-transmission
application/ocsp-request
application/ocsp-response
application/octet-stream                        bin
application/oda                                 oda
application/ogg                                 ogx
application/parityfec
application/pdf                                 pdf
application/pgp-encrypted
application/pgp-keys                            key
application/pgp-signature                       pgp
application/pics-rules                          prf
application/pkcs10
application/pkcs7-mime
application/pkcs7-signature
application/pkix-cert
application/pkix-crl
application/pkixcmp
application/postscript                          ps ai eps epsi epsf eps2 eps3
application/prs.alvestrand.titrax-sheet
application/prs.cww
application/prs.nprend
application/qsig
application/rar                                 rar
application/rdf+xml                             rdf
application/remote-printing
application/riscos
application/rss+xml                             rss
application/rtf                                 rtf
application/sdp
application/set-payment
application/set-payment-initiation
application/set-registration
application/set-registration-initiation
application/sgml
application/sgml-open-catalog
application/sieve
application/slate
application/smil                                smi smil
application/timestamp-query
application/timestamp-reply
application/vemmi
application/whoispp-query
application/whoispp-response
application/wita
application/x400-bp
application/xhtml+xml                           xhtml xht
application/xml                                 xml xsl xsd
application/xml-dtd
application/xml-external-parsed-entity
application/xspf+xml                            xspf
application/zip                                 zip
application/vnd.3M.Post-it-Notes
application/vnd.accpac.simply.aso
application/vnd.accpac.simply.imp
application/vnd.acucobol
application/vnd.aether.imp
application/vnd.android.package-archive         apk
application/vnd.anser-web-certificate-issue-initiation
application/vnd.anser-web-funds-transfer-initiation
application/vnd.audiograph
application/vnd.bmi
application/vnd.businessobjects
application/vnd.canon-cpdl
application/vnd.canon-lips
application/vnd.cinderella                      cdy
application/vnd.claymore
application/vnd.commerce-battelle
```

```
application/vnd.commonspace
application/vnd.comsocaller
application/vnd.contact.cmsg
application/vnd.cosmocaller
application/vnd.ctc-posml
application/vnd.cups-postscript
application/vnd.cups-raster
application/vnd.cups-raw
application/vnd.cybank
application/vnd.dna
application/vnd.dpgraph
application/vnd.dxr
application/vnd.ecdis-update
application/vnd.ecowin.chart
application/vnd.ecowin.filerequest
application/vnd.ecowin.fileupdate
application/vnd.ecowin.series
application/vnd.ecowin.seriesrequest
application/vnd.ecowin.seriesupdate
application/vnd.enliven
application/vnd.epson.esf
application/vnd.epson.msf
application/vnd.epson.quickanime
application/vnd.epson.salt
application/vnd.epson.ssf
application/vnd.ericsson.quickcall
application/vnd.eudora.data
application/vnd.fdf
application/vnd.ffsns
application/vnd.flographit
application/vnd.framemaker
application/vnd.fsc.weblaunch
application/vnd.fujitsu.oasys
application/vnd.fujitsu.oasys2
application/vnd.fujitsu.oasys3
application/vnd.fujitsu.oasysgp
application/vnd.fujitsu.oasysprs
application/vnd.fujixerox.ddd
application/vnd.fujixerox.docuworks
application/vnd.fujixerox.docuworks.binder
application/vnd.fut-misnet
application/vnd.google-earth.kml+xml            kml
application/vnd.google-earth.kmz                kmz
application/vnd.grafeq
application/vnd.groove-account
application/vnd.groove-identity-message
application/vnd.groove-injector
application/vnd.groove-tool-message
application/vnd.groove-tool-template
application/vnd.groove-vcard
application/vnd.hhe.lesson-player
application/vnd.hp-HPGL
application/vnd.hp-PCL
application/vnd.hp-PCLXL
application/vnd.hp-hpid
application/vnd.hp-hps
application/vnd.httphone
application/vnd.hzn-3d-crossword
application/vnd.ibm.MiniPay
application/vnd.ibm.afplinedata
application/vnd.ibm.modcap
application/vnd.informix-visionary
application/vnd.intercon.formnet
application/vnd.intertrust.digibox
application/vnd.intertrust.nncp
application/vnd.intu.qbo
application/vnd.intu.qfx
application/vnd.irepository.package+xml
application/vnd.is-xpr
application/vnd.japannet-directory-service
application/vnd.japannet-jpnstore-wakeup
application/vnd.japannet-payment-wakeup
application/vnd.japannet-registration
application/vnd.japannet-registration-wakeup
application/vnd.japannet-setstore-wakeup
application/vnd.japannet-verification
application/vnd.japannet-verification-wakeup
application/vnd.koan
```

```
application/vnd.lotus-1-2-3
application/vnd.lotus-approach
application/vnd.lotus-freelance
application/vnd.lotus-notes
application/vnd.lotus-organizer
application/vnd.lotus-screencam
application/vnd.lotus-wordpro
application/vnd.mcd
application/vnd.mediastation.cdkey
application/vnd.meridian-slingshot
application/vnd.mif
application/vnd.minisoft-hp3000-save
application/vnd.mitsubishi.misty-guard.trustweb
application/vnd.mobius.daf
application/vnd.mobius.dis
application/vnd.mobius.msl
application/vnd.mobius.plc
application/vnd.mobius.txf
application/vnd.motorola.flexsuite
application/vnd.motorola.flexsuite.adsi
application/vnd.motorola.flexsuite.fis
application/vnd.motorola.flexsuite.gotap
application/vnd.motorola.flexsuite.kmr
application/vnd.motorola.flexsuite.ttc
application/vnd.motorola.flexsuite.wem
application/vnd.mozilla.xul+xml                xul
application/vnd.ms-artgalry
application/vnd.ms-asf
application/vnd.ms-excel                       xls xlb xlt
application/vnd.ms-lrm
application/vnd.ms-pki.seccat                  cat
application/vnd.ms-pki.stl                     stl
application/vnd.ms-powerpoint                  ppt pps
application/vnd.ms-project
application/vnd.ms-tnef
application/vnd.ms-works
application/vnd.mseq
application/vnd.msign
application/vnd.music-niff
application/vnd.musician
application/vnd.netfpx
application/vnd.noblenet-directory
application/vnd.noblenet-sealer
application/vnd.noblenet-web
application/vnd.novadigm.EDM
application/vnd.novadigm.EDX
application/vnd.novadigm.EXT
application/vnd.oasis.opendocument.chart           odc
application/vnd.oasis.opendocument.database        odb
application/vnd.oasis.opendocument.formula         odf
application/vnd.oasis.opendocument.graphics        odg
application/vnd.oasis.opendocument.graphics-template       otg
application/vnd.oasis.opendocument.image           odi
application/vnd.oasis.opendocument.presentation    odp
application/vnd.oasis.opendocument.presentation-template   otp
application/vnd.oasis.opendocument.spreadsheet     ods
application/vnd.oasis.opendocument.spreadsheet-template          ots
application/vnd.oasis.opendocument.text            odt
application/vnd.oasis.opendocument.text-master     odm
application/vnd.oasis.opendocument.text-template           ott
application/vnd.oasis.opendocument.text-web        oth
application/vnd.osa.netdeploy
application/vnd.openxmlformats-officedocument.spreadsheetml.sheet        xlsx
application/vnd.openxmlformats-officedocument.spreadsheetml.template             xltx
application/vnd.openxmlformats-officedocument.presentationml.presentation pptx
application/vnd.openxmlformats-officedocument.presentationml.slideshow           ppsx
application/vnd.openxmlformats-officedocument.presentationml.template            potx
application/vnd.openxmlformats-officedocument.wordprocessingml.document          docx
application/vnd.openxmlformats-officedocument.wordprocessingml.template          dotx
application/vnd.palm
application/vnd.pg.format
application/vnd.pg.osasli
application/vnd.powerbuilder6
application/vnd.powerbuilder6-s
application/vnd.powerbuilder7
application/vnd.powerbuilder7-s
application/vnd.powerbuilder75
```

```
application/vnd.powerbuilder75-s
application/vnd.previewsystems.box
application/vnd.publishare-delta-tree
application/vnd.pvi.ptid1
application/vnd.pwg-xhtml-print+xml
application/vnd.rapid
application/vnd.rim.cod                         cod
application/vnd.s3sms
application/vnd.seemail
application/vnd.shana.informed.formdata
application/vnd.shana.informed.formtemplate
application/vnd.shana.informed.interchange
application/vnd.shana.informed.package
application/vnd.smaf                            mmf
application/vnd.sss-cod
application/vnd.sss-dtf
application/vnd.sss-ntf
application/vnd.stardivision.calc               sdc
application/vnd.stardivision.chart              sds
application/vnd.stardivision.draw               sda
application/vnd.stardivision.impress            sdd
application/vnd.stardivision.math               sdf
application/vnd.stardivision.writer             sdw
application/vnd.stardivision.writer-global      sgl
application/vnd.street-stream
application/vnd.sun.xml.calc                    sxc
application/vnd.sun.xml.calc.template           stc
application/vnd.sun.xml.draw                    sxd
application/vnd.sun.xml.draw.template           std
application/vnd.sun.xml.impress                 sxi
application/vnd.sun.xml.impress.template        sti
application/vnd.sun.xml.math                    sxm
application/vnd.sun.xml.writer                  sxw
application/vnd.sun.xml.writer.global           sxg
application/vnd.sun.xml.writer.template         stw
application/vnd.svd
application/vnd.swiftview-ics
application/vnd.symbian.install                 sis
application/vnd.triscape.mxs
application/vnd.trueapp
application/vnd.truedoc
application/vnd.tve-trigger
application/vnd.ufdl
application/vnd.uplanet.alert
application/vnd.uplanet.alert-wbxml
application/vnd.uplanet.bearer-choice
application/vnd.uplanet.bearer-choice-wbxml
application/vnd.uplanet.cacheop
application/vnd.uplanet.cacheop-wbxml
application/vnd.uplanet.channel
application/vnd.uplanet.channel-wbxml
application/vnd.uplanet.list
application/vnd.uplanet.list-wbxml
application/vnd.uplanet.listcmd
application/vnd.uplanet.listcmd-wbxml
application/vnd.uplanet.signal
application/vnd.vcx
application/vnd.vectorworks
application/vnd.vidsoft.vidconference
application/vnd.visio                           vsd
application/vnd.vividence.scriptfile
application/vnd.wap.sic
application/vnd.wap.slc
application/vnd.wap.wbxml                       wbxml
application/vnd.wap.wmlc                        wmlc
application/vnd.wap.wmlscriptc                  wmlsc
application/vnd.webturbo
application/vnd.wordperfect                     wpd
application/vnd.wordperfect5.1                  wp5
application/vnd.wrq-hp3000-labelled
application/vnd.wt.stf
application/vnd.xara
application/vnd.xfdl
application/vnd.yellowriver-custom-menu
application/x-123                               wk
application/x-7z-compressed                     7z
application/x-abiword                           abw
application/x-apple-diskimage                   dmg
```

```
application/x-bcpio                             bcpio
application/x-bittorrent                        torrent
application/x-cab                               cab
application/x-cbr                                   cbr
application/x-cbz                                   cbz
application/x-cdf                                   cdf cda
application/x-cdlink                                vcd
application/x-chess-pgn                             pgn
application/x-core
application/x-cpio                                  cpio
application/x-csh                                   csh
application/x-debian-package                        deb udeb
application/x-director                              dcr dir dxr
application/x-dms                                   dms
application/x-doom                                  wad
application/x-dvi                                   dvi
application/x-httpd-eruby                           rhtml
application/x-executable
application/x-font                                  pfa pfb gsf pcf pcf.Z
application/x-freemind                              mm
application/x-futuresplash                          spl
application/x-gnumeric                              gnumeric
application/x-go-sgf                                sgf
application/x-graphing-calculator                   gcf
application/x-gtar                                  gtar tgz taz
application/x-hdf                                   hdf
application/x-httpd-php                             phtml pht php
application/x-httpd-php-source                      phps
application/x-httpd-php3                            php3
application/x-httpd-php3-preprocessed               php3p
application/x-httpd-php4                            php4
application/x-httpd-php5                            php5
application/x-ica                                   ica
application/x-info                                  info
application/x-internet-signup                       ins isp
application/x-iphone                                iii
application/x-iso9660-image                         iso
application/x-jam                                   jam
application/x-java-applet
application/x-java-bean
application/x-java-jnlp-file                        jnlp
application/x-jmol                                  jmz
application/x-kchart                                chrt
application/x-kdelnk
application/x-killustrator                          kil
application/x-koan                                  skp skd skt skm
application/x-kpresenter                            kpr kpt
application/x-kspread                               ksp
application/x-kword                                 kwd kwt
application/x-latex                                 latex
application/x-lha                                   lha
application/x-lyx                                   lyx
application/x-lzh                                   lzh
application/x-lzx                                   lzx
application/x-maker                                 frm maker frame fm fb book fbdoc
application/x-mif                                   mif
application/x-ms-wmd                                wmd
application/x-ms-wmz                                wmz
application/x-msdos-program                         com exe bat dll
application/x-msi                                   msi
application/x-netcdf                                nc
application/x-ns-proxy-autoconfig                   pac dat
application/x-nwc                                   nwc
application/x-object                                o
application/x-oz-application                        oza
application/x-pkcs7-certreqresp                     p7r
application/x-pkcs7-crl                             crl
application/x-python-code                           pyc pyo
application/x-qgis                                  qgs shp shx
application/x-quicktimeplayer                       qtl
application/x-redhat-package-manager                rpm
application/x-ruby                                  rb
application/x-rx
application/x-sh                                    sh
application/x-shar                                  shar
application/x-shellscript
application/x-shockwave-flash                       swf swfl
```

```
application/x-silverlight                       scr
application/x-stuffit                           sit sitx
application/x-sv4cpio                           sv4cpio
application/x-sv4crc                        sv4crc
application/x-tar                           tar
application/x-tcl                           tcl
application/x-tex-gf                        gf
application/x-tex-pk                        pk
application/x-texinfo                       texinfo texi
application/x-trash                         ~ % bak old sik
application/x-troff                         t tr roff
application/x-troff-man                     man
application/x-troff-me                      me
application/x-troff-ms                      ms
application/x-ustar                         ustar
application/x-videolan
application/x-wais-source                   src
application/x-wingz                         wz
application/x-x509-ca-cert                  crt
application/x-xcf                           xcf
application/x-xfig                          fig
application/x-xpinstall                     xpi

audio/32kadpcm
audio/3gpp
audio/amr                                   amr
audio/amr-wb                                awb
audio/amr                                   amr
audio/amr-wb                                awb
audio/annodex                               axa
audio/basic                                 au snd
audio/flac                                  flac
audio/g.722.1
audio/l16
audio/midi                                  mid midi kar
audio/mp4a-latm
audio/mpa-robust
audio/mpeg                                  mpga mpega mp2 mp3 m4a
audio/mpegurl                               m3u
audio/ogg                                   oga ogg spx
audio/parityfec
audio/prs.sid                               sid
audio/telephone-event
audio/tone
audio/vnd.cisco.nse
audio/vnd.cns.anp1
audio/vnd.cns.inf1
audio/vnd.digital-winds
audio/vnd.everad.plj
audio/vnd.lucent.voice
audio/vnd.nortel.vbk
audio/vnd.nuera.ecelp4800
audio/vnd.nuera.ecelp7470
audio/vnd.nuera.ecelp9600
audio/vnd.octel.sbc
audio/vnd.qcelp
audio/vnd.rhetorex.32kadpcm
audio/vnd.vmx.cvsd
audio/x-aiff                                aif aiff aifc
audio/x-gsm                                 gsm
audio/x-mpegurl                             m3u
audio/x-ms-wma                              wma
audio/x-ms-wax                              wax
audio/x-pn-realaudio-plugin
audio/x-pn-realaudio                        ra rm ram
audio/x-realaudio                           ra
audio/x-scpls                               pls
audio/x-sd2                                 sd2
audio/x-wav                                 wav

chemical/x-alchemy                          alc
chemical/x-cache                            cac cache
chemical/x-cache-csf                        csf
chemical/x-cactvs-binary                    cbin cascii ctab
chemical/x-cdx                              cdx
chemical/x-cerius                           cer
chemical/x-chem3d                           c3d
chemical/x-chemdraw                         chm
```

```
chemical/x-cif                          cif
chemical/x-cmdf                         cmdf
chemical/x-cml                          cml
chemical/x-compass                      cpa
chemical/x-crossfire                    bsd
chemical/x-csml                         csml csm
chemical/x-ctx                          ctx
chemical/x-cxf                          cxf cef
#chemical/x-daylight-smiles             smi
chemical/x-embl-dl-nucleotide           emb embl
chemical/x-galactic-spc                 spc
chemical/x-gamess-input                 inp gam gamin
chemical/x-gaussian-checkpoint          fch fchk
chemical/x-gaussian-cube                cub
chemical/x-gaussian-input               gau gjc gjf
chemical/x-gaussian-log                 gal
chemical/x-gcg8-sequence                gcg
chemical/x-genbank                      gen
chemical/x-hin                          hin
chemical/x-isostar                      istr ist
chemical/x-jcamp-dx                     jdx dx
chemical/x-kinemage                     kin
chemical/x-macmolecule                  mcm
chemical/x-macromodel-input             mmd mmod
chemical/x-mdl-molfile                  mol
chemical/x-mdl-rdfile                   rd
chemical/x-mdl-rxnfile                  rxn
chemical/x-mdl-sdfile                   sd sdf
chemical/x-mdl-tgf                      tgf
#chemical/x-mif                         mif
chemical/x-mmcif                        mcif
chemical/x-mol2                         mol2
chemical/x-molconn-Z                    b
chemical/x-mopac-graph                  gpt
chemical/x-mopac-input                  mop mopcrt mpc zmt
chemical/x-mopac-out                    moo
chemical/x-mopac-vib                    mvb
chemical/x-ncbi-asn1                    asn
chemical/x-ncbi-asn1-ascii              prt ent
chemical/x-ncbi-asn1-binary             val aso
chemical/x-ncbi-asn1-spec               asn
chemical/x-pdb                          pdb ent
chemical/x-rosdal                       ros
chemical/x-swissprot                    sw
chemical/x-vamas-iso14976               vms
chemical/x-vmd                          vmd
chemical/x-xtel                         xtel
chemical/x-xyz                          xyz

image/cgm
image/g3fax
image/gif                               gif
image/ief                               ief
image/jpeg                              jpeg jpg jpe
image/naplps
image/pcx                               pcx
image/png                               png
image/prs.btif
image/prs.pti
image/svg+xml                           svg svgz
image/tiff                              tiff tif
image/vnd.cns.inf2
image/vnd.djvu                          djvu djv
image/vnd.dwg
image/vnd.dxf
image/vnd.fastbidsheet
image/vnd.fpx
image/vnd.fst
image/vnd.fujixerox.edmics-mmr
image/vnd.fujixerox.edmics-rlc
image/vnd.mix
image/vnd.net-fpx
image/vnd.svf
image/vnd.wap.wbmp                      wbmp
image/vnd.xiff
image/x-canon-cr2                       cr2
image/x-canon-crw                       crw
```

```
image/x-cmu-raster                          ras
image/x-coreldraw                           cdr
image/x-coreldrawpattern                    pat
image/x-coreldrawtemplate                   cdt
image/x-corelphotopaint                     cpt
image/x-epson-erf                           erf
image/x-icon                                ico
image/x-jg                                  art
image/x-jng                                 jng
image/x-ms-bmp                              bmp
image/x-nikon-nef                           nef
image/x-olympus-orf                         orf
image/x-photoshop                           psd
image/x-portable-anymap                     pnm
image/x-portable-bitmap                     pbm
image/x-portable-graymap                    pgm
image/x-portable-pixmap                     ppm
image/x-rgb                                 rgb
image/x-xbitmap                             xbm
image/x-xpixmap                             xpm
image/x-xwindowdump                         xwd

inode/chardevice
inode/blockdevice
inode/directory-locked
inode/directory
inode/fifo
inode/socket

message/delivery-status
message/disposition-notification
message/external-body
message/http
message/s-http
message/news
message/partial
message/rfc822                              eml

model/iges                                  igs iges
model/mesh                                  msh mesh silo
model/vnd.dwf
model/vnd.flatland.3dml
model/vnd.gdl
model/vnd.gs-gdl
model/vnd.gtw
model/vnd.mts
model/vnd.vtu
model/vrml                                  wrl vrml
model/x3d+vrml                              x3dv
model/x3d+xml                               x3d
model/x3d+binary                            x3db

multipart/alternative
multipart/appledouble
multipart/byteranges
multipart/digest
multipart/encrypted
multipart/form-data
multipart/header-set
multipart/mixed
multipart/parallel
multipart/related
multipart/report
multipart/signed
multipart/voice-message

text/cache-manifest                         manifest
text/calendar                               ics icz
text/css                                    css
text/csv                                    csv
text/directory
text/english
text/enriched
text/h323                                   323
text/html                                   html htm shtml
text/iuls                                   uls
text/mathml                                 mml
text/parityfec
```

```
text/plain                                  asc txt text pot brf
text/prs.lines.tag
text/rfc822-headers
text/richtext                               rtx
text/rtf
text/scriptlet                              sct wsc
text/t140
text/texmacs                                tm ts
text/tab-separated-values                   tsv
text/uri-list
text/vnd.abc
text/vnd.curl
text/vnd.DMClientScript
text/vnd.flatland.3dml
text/vnd.fly
text/vnd.fmi.flexstor
text/vnd.in3d.3dml
text/vnd.in3d.spot
text/vnd.IPTC.NewsML
text/vnd.IPTC.NITF
text/vnd.latex-z
text/vnd.motorola.reflex
text/vnd.ms-mediapackage
text/vnd.sun.j2me.app-descriptor            jad
text/vnd.wap.si
text/vnd.wap.sl
text/vnd.wap.wml                            wml
text/vnd.wap.wmlscript                      wmls
text/x-bibtex                               bib
text/x-boo                                  boo
text/x-c++hdr                               h++ hpp hxx hh
text/x-c++src                               c++ cpp cxx cc
text/x-chdr                                 h
text/x-component                            htc
text/x-crontab
text/x-csh                                  csh
text/x-csrc                                 c
text/x-dsrc                                 d
text/x-diff                                 diff patch
text/x-haskell                              hs
text/x-java                                 java
text/x-literate-haskell                     lhs
text/x-makefile
text/x-moc                                  moc
text/x-pascal                               p pas
text/x-pcs-gcd                              gcd
text/x-perl                                 pl pm
text/x-python                               py
text/x-scala                                scala
text/x-server-parsed-html
text/x-setext                               etx
text/x-sh                                   sh
text/x-tcl                                  tcl tk
text/x-tex                                  tex ltx sty cls
text/x-vcalendar                            vcs
text/x-vcard                                vcf

video/3gpp                                  3gp
video/annodex                               axv
video/dl                                    dl
video/dv                                    dif dv
video/fli                                   fli
video/gl                                    gl
video/mpeg                                  mpeg mpg mpe
video/mp4                                   mp4
video/quicktime                             qt mov
video/mp4v-es
video/ogg                                   ogv
video/parityfec
video/pointer
video/vnd.fvt
video/vnd.motorola.video
video/vnd.motorola.videop
video/vnd.mpegurl                           mxu
video/vnd.mts
video/vnd.nokia.interleaved-multimedia
video/vnd.vivo
```

```
video/x-flv                             flv
video/x-la-asf                          lsf lsx
video/x-mng                             mng
video/x-ms-asf                          asf asx
video/x-ms-wm                           wm
video/x-ms-wmv                          wmv
video/x-ms-wmx                          wmx
video/x-ms-wvx                          wvx
video/x-msvideo                         avi
video/x-sgi-movie                       movie
video/x-matroska                        mpv mkv

x-conference/x-cooltalk                 ice

x-epoc/x-sisx-app                       sisx
x-world/x-vrml                          vrm vrml wrl
```

Anexo III - /etc/apache2/sites-available/default-ssl

```
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
   ServerAdmin webmaster@localhost

   DocumentRoot /var/www
   <Directory />
      Options FollowSymLinks
      AllowOverride None
   </Directory>
   <Directory /var/www/>
      Options Indexes FollowSymLinks MultiViews
      AllowOverride None
      Order allow,deny
      allow from all
   </Directory>

   ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
   <Directory "/usr/lib/cgi-bin">
      AllowOverride None
      Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
      Order allow,deny
      Allow from all
   </Directory>

   ErrorLog ${APACHE_LOG_DIR}/error.log

   # Possible values include: debug, info, notice, warn, error, crit,
   # alert, emerg.
   LogLevel warn

   CustomLog ${APACHE_LOG_DIR}/ssl_access.log combined

   Alias /doc/ "/usr/share/doc/"
   <Directory "/usr/share/doc/">
      Options Indexes MultiViews FollowSymLinks
      AllowOverride None
      Order deny,allow
      Deny from all
      Allow from 127.0.0.0/255.0.0.0 ::1/128
   </Directory>

   #   SSL Engine Switch:
   #   Enable/Disable SSL for this virtual host.
   SSLEngine on

   #   A self-signed (snakeoil) certificate can be created by installing
   #   the ssl-cert package. See
   #   /usr/share/doc/apache2.2-common/README.Debian.gz for more info.
   #   If both key and certificate are stored in the same file, only the
   #   SSLCertificateFile directive is needed.
   SSLCertificateFile    /etc/ssl/certs/ssl-cert-snakeoil.pem
   SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

   #   Server Certificate Chain:
   #   Point SSLCertificateChainFile at a file containing the
   #   concatenation of PEM encoded CA certificates which form the
   #   certificate chain for the server certificate. Alternatively
   #   the referenced file can be the same as SSLCertificateFile
   #   when the CA certificates are directly appended to the server
   #   certificate for convinience.
   #SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.crt

   #   Certificate Authority (CA):
   #   Set the CA certificate verification path where to find CA
   #   certificates for client authentication or alternatively one
   #   huge file containing all of them (file must be PEM encoded)
   #   Note: Inside SSLCACertificatePath you need hash symlinks
   #         to point to the certificate files. Use the provided
   #         Makefile to update the hash symlinks after changes.
   #SSLCACertificatePath /etc/ssl/certs/
   #SSLCACertificateFile /etc/apache2/ssl.crt/ca-bundle.crt

   #   Certificate Revocation Lists (CRL):
   #   Set the CA revocation path where to find CA CRLs for client
```

```
#   authentication or alternatively one huge file containing all
#   of them (file must be PEM encoded)
#   Note: Inside SSLCARevocationPath you need hash symlinks
#         to point to the certificate files. Use the provided
#         Makefile to update the hash symlinks after changes.
#SSLCARevocationPath /etc/apache2/ssl.crl/
#SSLCARevocationFile /etc/apache2/ssl.crl/ca-bundle.crl

#   Client Authentication (Type):
#   Client certificate verification type and depth.  Types are
#   none, optional, require and optional_no_ca.  Depth is a
#   number which specifies how deeply to verify the certificate
#   issuer chain before deciding the certificate is not valid.
#SSLVerifyClient require
#SSLVerifyDepth  10

#   Access Control:
#   With SSLRequire you can do per-directory access control based
#   on arbitrary complex boolean expressions containing server
#   variable checks and other lookup directives.  The syntax is a
#   mixture between C and Perl.  See the mod_ssl documentation
#   for more details.
#<Location />
#SSLRequire (    %{SSL_CIPHER} !~ m/^(EXP|NULL)/ \
#            and %{SSL_CLIENT_S_DN_O} eq "Snake Oil, Ltd." \
#            and %{SSL_CLIENT_S_DN_OU} in {"Staff", "CA", "Dev"} \
#            and %{TIME_WDAY} >= 1 and %{TIME_WDAY} <= 5 \
#            and %{TIME_HOUR} >= 8 and %{TIME_HOUR} <= 20       ) \
#           or %{REMOTE_ADDR} =~ m/^192\.76\.162\.[0-9]+$/
#</Location>

#   SSL Engine Options:
#   Set various options for the SSL engine.
#   o FakeBasicAuth:
#     Translate the client X.509 into a Basic Authorisation.  This means that
#     the standard Auth/DBMAuth methods can be used for access control.  The
#     user name is the `one line' version of the client's X.509 certificate.
#     Note that no password is obtained from the user. Every entry in the user
#     file needs this password: `xxj31ZMTZzkVA'.
#   o ExportCertData:
#     This exports two additional environment variables: SSL_CLIENT_CERT and
#     SSL_SERVER_CERT. These contain the PEM-encoded certificates of the
#     server (always existing) and the client (only existing when client
#     authentication is used). This can be used to import the certificates
#     into CGI scripts.
#   o StdEnvVars:
#     This exports the standard SSL/TLS related `SSL_*' environment variables.
#     Per default this exportation is switched off for performance reasons,
#     because the extraction step is an expensive operation and is usually
#     useless for serving static content. So one usually enables the
#     exportation for CGI and SSI requests only.
#   o StrictRequire:
#     This denies access when "SSLRequireSSL" or "SSLRequire" applied even
#     under a "Satisfy any" situation, i.e. when it applies access is denied
#     and no other module can change it.
#   o OptRenegotiate:
#     This enables optimized SSL connection renegotiation handling when SSL
#     directives are used in per-directory context.
#SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
<FilesMatch "\.(cgi|shtml|phtml|php)$">
   SSLOptions +StdEnvVars
</FilesMatch>
<Directory /usr/lib/cgi-bin>
   SSLOptions +StdEnvVars
</Directory>

#   SSL Protocol Adjustments:
#   The safe and default but still SSL/TLS standard compliant shutdown
#   approach is that mod_ssl sends the close notify alert but doesn't wait for
#   the close notify alert from client. When you need a different shutdown
#   approach you can use one of the following variables:
#   o ssl-unclean-shutdown:
#     This forces an unclean shutdown when the connection is closed, i.e. no
#     SSL close notify alert is send or allowed to received.  This violates
#     the SSL/TLS standard but is needed for some brain-dead browsers. Use
#     this when you receive I/O errors because of the standard approach where
#     mod_ssl sends the close notify alert.
#   o ssl-accurate-shutdown:
```

```
    #      This forces an accurate shutdown when the connection is closed, i.e. a
    #      SSL close notify alert is send and mod_ssl waits for the close notify
    #      alert of the client. This is 100% SSL/TLS standard compliant, but in
    #      practice often causes hanging connections with brain-dead browsers. Use
    #      this only for browsers where you know that their SSL implementation
    #      works correctly.
    #   Notice: Most problems of broken clients are also related to the HTTP
    #   keep-alive facility, so you usually additionally want to disable
    #   keep-alive for those clients, too. Use variable "nokeepalive" for this.
    #   Similarly, one has to force some clients to use HTTP/1.0 to workaround
    #   their broken HTTP/1.1 implementation. Use variables "downgrade-1.0" and
    #   "force-response-1.0" for this.
    BrowserMatch "MSIE [2-6]" \
       nokeepalive ssl-unclean-shutdown \
       downgrade-1.0 force-response-1.0
    # MSIE 7 and newer should be able to use keepalive
    BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown

</VirtualHost>
</IfModule>
```

Anexo IV - openssl_autofirmado.txt

```
/etc/apache2/tus-ssl# openssl  req -new -nodes -keyout tupaginaweb.key -out tupaginaweb.csr
Generating a 1024 bit RSA private key
.....................................++++++
.++++++
writing new private key to 'tupaginaweb.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:ES
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:autofirmado.ssl.empresa-proyecto.com
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
root@debian-servidor-fp:/etc/apache2/tus-ssl#   openssl   x509   -in   tupaginaweb.csr   -out
tupaginaweb.crt -req -signkey tupaginaweb.key -days 3650
Signature ok
subject=/C=ES/ST=Some-State/O=Internet Widgits Pty Ltd/CN=autofirmado.ssl.empresa-proyecto.com
Getting Private key
root@debian-servidor-fp:/etc/apache2/tus-ssl#
```

Anexo V - Instalación y configuración de OpenLDAP

Para simplificar la administración de los usuarios del sistema es ideal utilizar una base de datos accesible mediante LDAP. Almacenar las cuentas de usuario de forma centralizada en un único repositorio facilitará la creación, modificación y eliminación de cuentas de usuario y grupos de usuarios. Será necesario configurar los PCs de la red para que utilicen el servidor LDAP como servidor de autentificación.

Instalación de OpenLDAP

El servidor OpenLDAP está disponible en el paquete **slapd** por tanto, lo instalaremos utilizando apt-get. También nos conviene instalar el paquete **ldap-utils** que contiene utilidades adicionales:

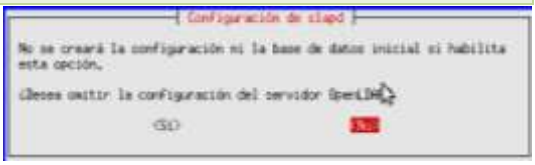
```
// Instalación del servidor LDAP
sudo apt-get install slapd ldap-utils
```

Configuración inicial de OpenLDAP

Los archivos de configuración del servidor LDAP se almacenan en la carpeta /etc/ldap/. En lugar de editar manualmente dichos archivos, es mejor lanzar el asistente de configuración de slapd. Para ello debemos ejecutar el siguiente comando:

```
//Lanzar el asistente de configuración de slapd
sudo dpkg-reconfigure slapd
```

Lo primero que nos pregunta el asistente es si deseamos omitir la configuración del servidor LDAP:



Asistente de configuración de slapd

Obviamente responderemos que no, ya que precisamente lo que queremos es configurar el servidor LDAP.

Después nos preguntará si queremos que se elimine la base de datos cuando quitemos slapd. Para evitar confusiones con bases de datos anteriores, lo mejor es responder Sí:

Pregunta sobre la eliminación de la base de datos

Luego nos preguntará si deseamos utilizar LDAP versión 2, respondemos que no ya que apenas se utiliza.

Utilización LDAP versión 2

Con esto habremos concluido la configuración inicial del servidor LDAP.

Arranque y parada manual del servidor LDAP

El servidor LDAP, al igual que todos los servicios en Debian, dispone de un script de arranque y parada en la carpeta /etc/init.d.

```
// Arrancar o reiniciar el servidor LDAP
sudo /etc/init.d/slapd restart

// Parar el servidor LDAP
sudo /etc/init.d/slapd stop
```

Anexo VI - Instalación y configuración del servidor OpenLDAP en Debian 6

```
Escenario: debian-servidor-fp --> IP: 192.168.200.250
Procedimiento a realizar como usuario root:

1) Actualiza el sistema operativo.
root@debian-servidor-fp:~# apt-get update
root@debian-servidor-fp:~# apt-get ugrade

2) Instala los paquetes necesarios para el funcionamiento de OpenLDAP. La instalación te pedirá una contraseña, como puedes ver a continuación la contraseña es 'admin'
root@debian-servidor-fp:~# apt-get install slapd ldap-utils
Administrator password: admin
Confirm password: admin

3) Verifica que el servidor OpenLDAP está activo, por defecto,  en el puerto TCP 389
root@debian-servidor-fp:~# netstat -natp | grep 389
tcp        0      0 0.0.0.0:389             0.0.0.0:*               LISTEN      1775/slapd
tcp6       0      0 :::389                  :::*                    LISTEN      1775/slapd

4) Configura el servidor OpenLDAP. Los valores utilizados los puedes ver a continuación del comando
root@debian-servidor-fp:~# dpkg-reconfigure slapd
 Omit OpenLDAP config ? No
 Domain name : proyecto.com
 organisation name : proyecto.com
 admin password : admin
 admin password : admin
 database module to use : HDB
 delete database when purging the package ? No
 Move the previous database ? Si
 Allow LDAPv2 ? No

5) Continuación de la configuración del servidor OpenLDAP. Edita el archivo /etc/ldap/slapd.d/cn=config/olcDatabase\=\{1\}hdb.ldif y cambia todas las cadenas 'dc=nodomain' por 'dc=proyecto,dc=com', similar a como se expone a continuación:
root@debian-servidor-fp:~# cat /etc/ldap/slapd.d/cn=config/olcDatabase\=\{1\}hdb.ldif | sed -e "s/dc=nodomain/dc=proyecto,dc=com/g" > a.txt
root@debian-servidor-fp:~# mv a.txt /etc/ldap/slapd.d/cn=config/olcDatabase\=\{1\}hdb.ldif
root@debian-servidor-fp:~# nano /etc/ldap/slapd.d/cn=config/olcDatabase\=\{1\}hdb.ldif
dn: olcDatabase={1}hdb
objectClass: olcDatabaseConfig
olcDatabase: {1}hdb
olcDbDirectory: /var/lib/ldap
olcSuffix: dc=proyecto,dc=com
olcAccess: {0}to attrs=userPassword,shadowLastChange by self write by anonymous auth by dn="cn=admin,dc=proyecto,dc=com" write by * none
olcAccess: {1}to dn.base="" by * read
olcAccess: {2}to * by self write by dn="cn=admin,dc=proyecto,dc=com" write by * read
olcLastMod: TRUE
olcRootDN: cn=admin,dc=proyecto,dc=com
olcRootPW:: e1NTSEF9bThuNDVrOGZCRVhHVzlBYUpud0ZGYkI1QUtvanVsSnE=
olcDbCheckpoint: 512 30
olcDbConfig: {0}set_cachesize 0 2097152 0
olcDbConfig: {1}set_lk_max_objects 1500
olcDbConfig: {2}set_lk_max_locks 1500
olcDbConfig: {3}set_lk_max_lockers 1500
olcDbIndex: objectClass eq
structuralObjectClass: olcHdbConfig
entryUUID: 1e80cb3e-1f44-1030-9fab-8b0ca1ca9cc2
creatorsName: cn=admin,cn=config
createTimestamp: 20110530200658Z
entryCSN: 20110530200658.710565Z#000000#000#000000
modifiersName: cn=admin,cn=config
modifyTimestamp: 20110530200658Z

7) Activa los cambios del servidor OpenLDAP
root@debian-servidor-fp:~# /etc/init.d/slapd restart
Stopping OpenLDAP: slapd.
Starting OpenLDAP: slapd.

8) Testea el servidor OpenLDAP:
root@debian-servidor-fp:~# slaptest
hdb_db_open: database "dc=proyecto,dc=com": unclean shutdown detected; attempting recovery.
```

```
hdb_db_open: database "dc=proyecto,dc=com": recovery skipped in read-only mode. Run manual
recovery if errors are encountered.
config file testing succeeded

9) Instala los paquetes necesarios para que Apache funcione con LDAP
root@debian-servidor-fp:~# apt-get install libapache2-mod-vhost-ldap

10) Habilita el módulo LDAP para Apache:
root@debian-servidor-fp:~# a2enmod authnz_ldap

11) Reinicia Apache:
root@debian-servidor-fp:~# /etc/init.d/apache2 restart

12) Crea la estructura básica del dominio LDAP mediante la ejecucion de un fichero basica.ldif
root@debian-servidor-fp:~# nano basica.ldif
# Objetos raiz del dominio
dn: dc=proyecto,dc=com
objectClass: top
objectClass: dcObject
objectclass: organization
o: proyecto.com
dc: proyecto
description:  Raiz de dominio

# Usuarios
dn: ou=usuarios,dc=proyecto,dc=com
objectClass: organizationalUnit
ou: usuarios

# Grupos
dn: ou=grupos,dc=proyecto,dc=com
objectClass: organizationalUnit
ou: grupos

root@debian-servidor-fp:~# ldapadd -x -D cn=admin,dc=proyecto,dc=com -w admin -f basica.ldif
adding new entry "dc=proyecto,dc=com"

adding new entry "ou=usuarios,dc=proyecto,dc=com"

adding new entry "ou=grupos,dc=proyecto,dc=com"

13) Añadiendo un usuario a LDAP de nombre pruebas y contraseña: 123456 mediante el archivo
usuario.ldif:
root@debian-servidor-fp:~# cat usuario.ldif
# Usuario
dn: uid=pruebas,ou=usuarios,dc=proyecto,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: pruebas
sn: daw02
givenName: Pruebas
cn: Pruebas daw02
displayName: Pruebas DAW02
uidNumber: 10000
gidNumber: 10000
userPassword: 123456
gecos: Pruebas DAW02
loginShell: /bin/bash
homeDirectory: /home/pruebas
shadowExpire: -1
shadowFlag: 0
shadowWarning: 7
shadowMin: 8
shadowMax: 999999
shadowLastChange: 10877
mail: pruebas.daw02@proyecto.com
initials: PD

root@debian-servidor-fp:~# ldapadd -x -D cn=admin,dc=proyecto,dc=com -w admin -f usuario.ldif
adding new entry "uid=pruebas,ou=usuarios,dc=proyecto,dc=com"

14) Reiniciar LDAP y Apache
root@debian-servidor-fp:/etc/apache2/sites-available# /etc/init.d/slapd restart

root@debian-servidor-fp:/etc/apache2/sites-available# /etc/init.d/apache2 restart
```

Anexo VII.- Despliegue aplicación Opencart.

"El movimiento se demuestra andando."

Diógenes de Sínope

Procede con el siguiente ejemplo: **Instalación de OpenCart**

1. **Descarga y descomprime la aplicación:**
   - ✓ En la página de descarga de OpenCart(http://www.opencart.com/index.php?route=download/download) puedes ver los requisitos para la instalación de Opencart: **Web Server** (preferably Apache) , **PHP** (at least 5.2) , **MySQL** , **Curl** , **Fsock**
   - ✓ Descarga el último paquete estable de Opencart de la página web de descarga en `/tmp/pruebas`

     ```
     mkdir /tmp/pruebas
     wget -c http://opencart.googlecode.com/files/opencart_v1.4.9.5.zip (6 MB)
     ```
   - ✓ Descomprime el paquete

     ```
     cd /tmp/pruebas
     apt-get install unzip
     unzip opencart_v1.4.9.5.zip
     ```
2. Lee el fichero de instalación `install.txt`.
3. Crea el virtualhost para Opencart:
   - ✓ Copia la carpeta upload en el servidor web. Para ello genera en `/etc/apache2/sites-available/` un virtualhost de nombre `tienda-virtual` como el siguiente:

     ```
     <VirtualHost 192.168.200.250:80
        DocumentRoot /var/www/tienda-virtual
        ServerName              ww.tienda-virtual.empresa-proyecto.com              ErrorLog
        /var/log/apache2/error_tienda-virtual.log
        CustomLog  var/log/apache2/access_tienda-virtual.log  "%h  %l  %u  %t  \"%r\"  %>s  %b
        \"%{Referer}i\" %I %O"
     </VirtualHost>
     ```
   - ✓ Ahora mueve la carpeta `upload` con el nombre tienda-virtual en `/var/www/tienda-virtual`
   - ✓ Activa el sitio nuevo tienda-virtual: `a2ensite tienda-virtual`
   - ✓ Recarga la configuración de Apache:`/etc/init.d/apache2 reload`
   - ✓ Verifica que los siguientes ficheros y carpetas tengan permisos de escritura en `/var/www/tienda-virtual/`: `chmod 0755` ó `0777` para: `image/`, `image/cache/`, `image/data/`, `system/cache/`, `system/logs/`, `download/`, `config.php`, `admin/config.php`
4. **Crea la base de datos para OpenCart y el usuario con permisos en la misma**:
   Asegúrate que posees una base de datos mysql para Opencart y un usuario distinto de root con permisos en la misma:
   - ✓ Primero, debes crear una nueva base de datos para tu sitio Opencart:
     `/usr/bin/mysql -h127.0.0.1 -uroot -p -e "CREATE DATABASE db_opencart;"`
     donde:
     - ➔ **`root`** es el usuario administrador de MySQL y por lo tanto tiene los privilegios para crear una base de datos.
     - ➔ **`db_opencart`** es el nombre de la base de datos de opencart que acabas de crear.

     MySQL te pide la contraseña del usuario root y luego crea los archivos iniciales de la base de datos.
   - ✓ Segundo, creas el usuario con privilegios en la base de datos de nuevo se requiere la contraseña de root-.

     ```
     /usr/bin/mysql          -h127.0.0.1          -uroot          -p          -e          "GRANT
     SELECT,UPDATE,INSERT,DELETE,DROP,INDEX,ALTER,CREATE ON "db_opencart".*
     TO "db_user_opencart"@localhost IDENTIFIED BY 'opencart';"
     ```

     donde:
     - ➔ '**`db_opencart`**' es el nombre de tu base de datos
     - ➔ '**`db_user_opencart@localhost`**' es el nombre de usuario de MySQL que posee los privilegios en la base de datos '`db opencart`'.

- ➔ '`opencart`' es la contraseña requerida para iniciar sesión como el usuario '`db_user_opencart`' en MySQL

✓ Tercero, para activar los nuevos cambios ejecuta:

```
/usr/bin/mysql -h127.0.0.1 -uroot -p -e "flush privileges;"
```

Alternativamente puedes usar, si lo posees, tu panel de control Web o bien phpMyAdmin para crear la base de datos '`db opencart`' y el usuario '`db user opencart`'

5. Visita la página principal de tu Opencart, por ejemplo: http://www.tienda-virtual.empresa-proyecto.com/
6. Sigue las instrucciones que aparecen en pantalla.
7. Una vez acabada la instalación borra la carpeta `install`.
8. Puedes ya visitar tu tienda online en: `http://www.tienda-virtual.empresa-proyecto.com/` y tu panel de administración en: `http://www.tienda-virtual.empresa-proyecto.com/admin/`