Tarea para DAW01 de José Luis Comesaña Cabeza.

La arquitectura Web es un modelo compuesto de tres capas, ¿cuáles son y cuál es la función de cada una de ellas?

1. **Capa de presentación** es la encargada de la navegabilidad, validación de los datos de entrada, formateo de los datos de salida, presentación de la web, etc.; se trata de la capa que se presenta al usuario.
   Comprende las responsabilidades de:
   - Navegabilidad del sistema
   - Validación de datos de entrada
   - Formateo de los datos de salida
   - Internacionalización
   - Renderizado de presentación
   - etc
2. **Capa de negocio** es la que recibe las peticiones del usuario y desde donde se le envían las respuestas; en esta capa se verifican que las reglas establecidas se cumplen.
   Comprende las responsabilidades de:
   - Lógica de negocio (o dominio) del sistema.
   - Resultado del análisis funcional (*conjunto de reglas de negocio que abstraen el mundo real*)
   - Capa de presentación y capa de negocio han de ser independientes entre sí en la medida de lo posible
3. **Capa de acceso a datos** o capa de persistencia, es la formada por determinados gestores de datos que se encargan de la lógica de persistencia de las entidades que maneja el sistema en desarrollo:
   - Inserción
   - Eliminación
   - Actualizaciones
   - Búsquedas
   - etc

   No tiene por qué tratarse necesariamente de una base de datos relacional

Una plataforma web es el entorno de desarrollo de software empleado para diseñar y ejecutar un sitio web; destacan dos plataformas web, LAMP y WISA. Explica en qué consiste cada una de ellas.

La plataforma LAMP consiste en un conjunto de herramientas de software libre que al trabajar juntas nos permite disponer de un servidor web. Las siglas de este tipo de plataforma provienen de los términos:

- ✓ **L**inux: Sistema operativo.
- ✓ **A**pache: Servidor web.
- ✓ **M**ySQL: Gestor de bases de datos.
- ✓ **P**HP: Lenguaje interpretado PHP, aunque a veces se sustituye por Perl o Python.

Son de las más apreciadas por los proveedores de servicios web, ya que tienen como ventajas:

- ➔ Compuesta por software libre
- ➔ El código de todos sus componentes está disponible, con lo que podemos realizar las modificaciones que creamos oportunas.

Por su parte, la plataforma WISA está basada en tecnologías desarrolladas por la compañía Microsoft; se trata, por lo tanto, de software propietario. La componen los siguientes elementos:

- ✓ **W**indows: Sistema operativo.
- ✓ **I**nternet Information Services: servidor web.
- ✓ **S**QL Server: gestor de bases de datos.
- ✓ **A**SP o ASP.NET: como lenguaje para scripting del lado del servidor.

A parte del consabido coste económico que requiere la utilización de este tipo de plataforma, tiene como puntos fuertes:

- ➔ Más robusta si se usa para aplicaciones web que funcionen en una intranet o que posean un gran nivel de actividad.
- ➔ Posee multitud de aplicaciones desarrolladas por terceros (*aunque la mayoría son de pago*)

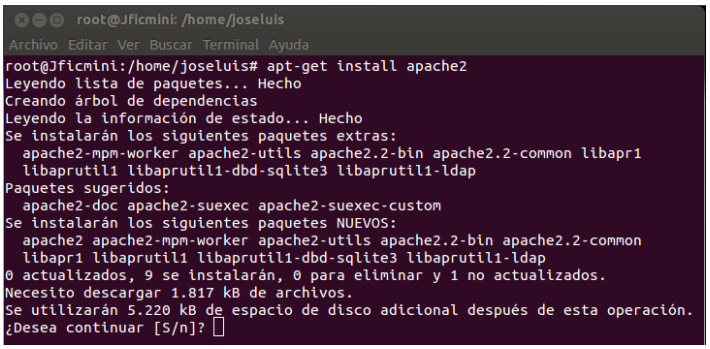
Dispones de una máquina que cuenta con el sistema operativo Ubuntu 12.04 LTS recientemente actualizado, esta máquina tiene el entorno de red configurado y, además, dispones de conexión a Internet. Además, estás trabajando con la cuenta del usuario root. Indica cada uno de los pasos, y comandos implicados en ellos, para conseguir hacer lo siguiente:

Instalar el servidor web Apache desde terminal.

Accedemos a una pantalla de terminal pulsando `Ctrl + Alt + T` y nos logamos como usuario root tecleando `sudo su` y poniendo la contraseña de root que tengamos.
Ahora tecleamos la orden:

```
apt-get install apache2
```

Nos avisa del número de paquetes que se instalarán y el espacio necesario para ello.
Si pulsamos "s" a la pregunta de ¿Desea continuar (s/n)? se comenzará la instalación de forma automática hasta finalizar teniendo instalado el servidos Apache.

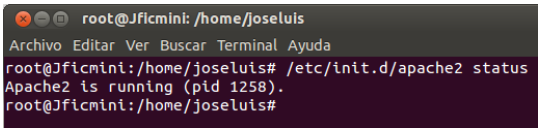


Comprobar que está funcionando el servidor Apache desde terminal.

Para comprobar el funcionamiento de apache desde terminal, estando como root teclearemos:

```
/etc/init.d/apache2 status
```

tras lo que nos informará que se está ejecutando y el número de proceso que tiene.



Comprobar que está funcionando el servidor Apache desde navegador.

Para comprobar el funcionamiento del servidor desde el navegador, abrimos éste, por ejemplo Mozilla Firefox y en su barra de direcciones tecleamos `localhost` o `127.0.0.1` tras lo cual nos aparecerá la pantalla por defecto del servidor "**It works!**".

Cambiar el puerto por el cual está escuchando Apache pasándolo al puerto 82.

Para cambiar el puerto de escucha de Apache hemos de editar el fichero `ports.conf` que se encuentra en `etc/apache2`, por lo que en un terminal como root tecleamos:

1
```
gedit /etc/apache2/ports.conf
```

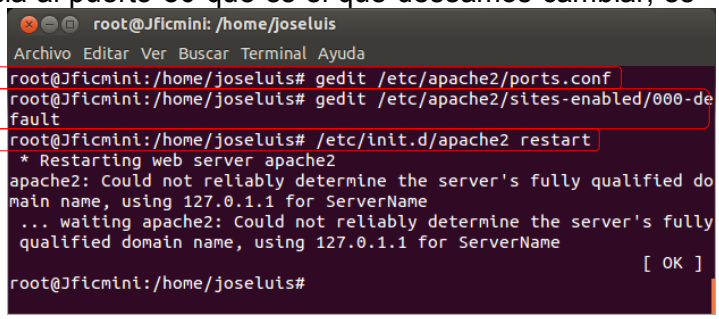
Y modificamos las líneas que hacen referencia al puerto **80** que es el que deseamos cambiar, es decir, cambiaremos:

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

por estas otras:

```
NameVirtualHost *:82
Listen 82
```

Y como nuestro servidor va a ofrecer acceso a páginas web, tendremos que



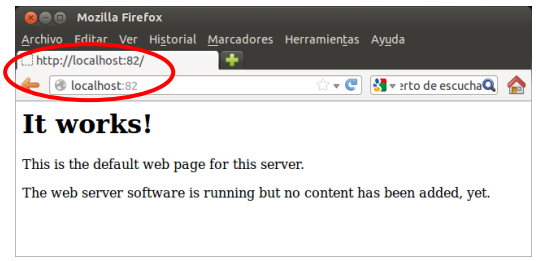
modificar también el puerto de la páginas que sirvamos, por lo que desde el terminal editamos el fichero:

2
```
gedit /etc/apache2/sites-enabled/000-default
```

Y modificamos `VirtualHost *:80` por `VirtualHost *:82`

Tras ello hemos de reiniciar el servidor tecleando:

3
```
/etc/init.d/apache2 restart
```



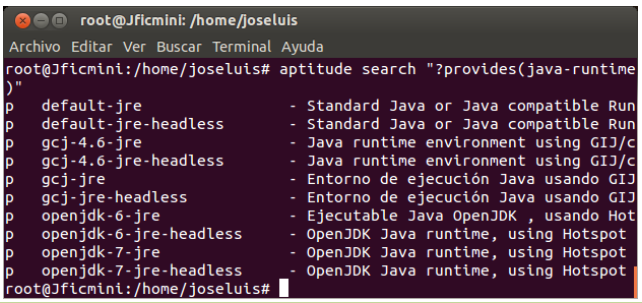
A partir de este momento, si en un navegador tecleamos `localhost` nos saldrá una página de error, pero si ponemos `localhost:82` nos aparecerá la consabida página de "**It Works!**"

Instalar el servidor de aplicaciones Tomcat.

Lo primero que hemos de hacer es instalar el JDK de Java, ya que se usará un conector proporcionado por Java para redirigir las peticiones que se hagan a Apache, a Tomcat.

Para realizar esto buscaremos el paquete Java que más nos interese e instalamos para ello `aptitude` como buscador de versiones tecleando `apt-get install aptitude` y cuando lo tengamos instalado tecleamos:

```
aptitude search "?provides(java-runtime)"
```



Decidimos instalar la primera versión que aparece, por lo que tecleamos:

```
apt-get install default-jre
```

Ahora, y para que esté disponible desde cualquier parte del sistema, tenemos que añadir la dirección donde se encuentra instalado, a la variable de entorno PATH, por lo que editamos como root el fichero profile tecleando:

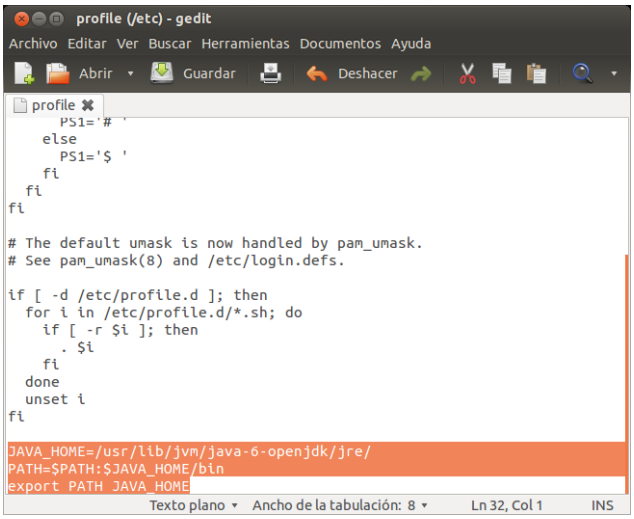
```
gedit /etc/profile
```

y añadimos al final del archivo las líneas:

```
JAVA_HOME=/usr/lib/jvm/java-6-openjdk/jre/
PATH=$PATH:$JAVA_HOME/bin
export PATH JAVA_HOME
```

Lo grabamos y actualizamos las variables de entorno tecleando en el terminal:

```
source /etc/profile
```



Ya tenemos Java preparado y es hora de descargar la última versión estable de Tomcat, para lo cual vamos a la dirección http://apache.rediris.es/tomcat/tomcat-6/ y vemos que la última es la 6.0.36. Vamos a su carpeta bin y nos fijamos en el nombre del fichero comprimido que tendremos que descargar, para volver al terminal de root y proceder a dicha descarga tecleando:

```
wget http://apache.rediris.es/tomcat/tomcat-6/v6.0.36/bin/apache-tomcat-6.0.36.tar.gz
```

Cuando finalizamos su descarga lo descomprimimos tecleando:

```
tar xvzf apache-tomcat-6.0.36.tar.gz
```

y ahora lo movemos a la carpeta donde procederemos a su instalación:

```
mv -v apache-tomcat-6.0.36 /usr/local/
```

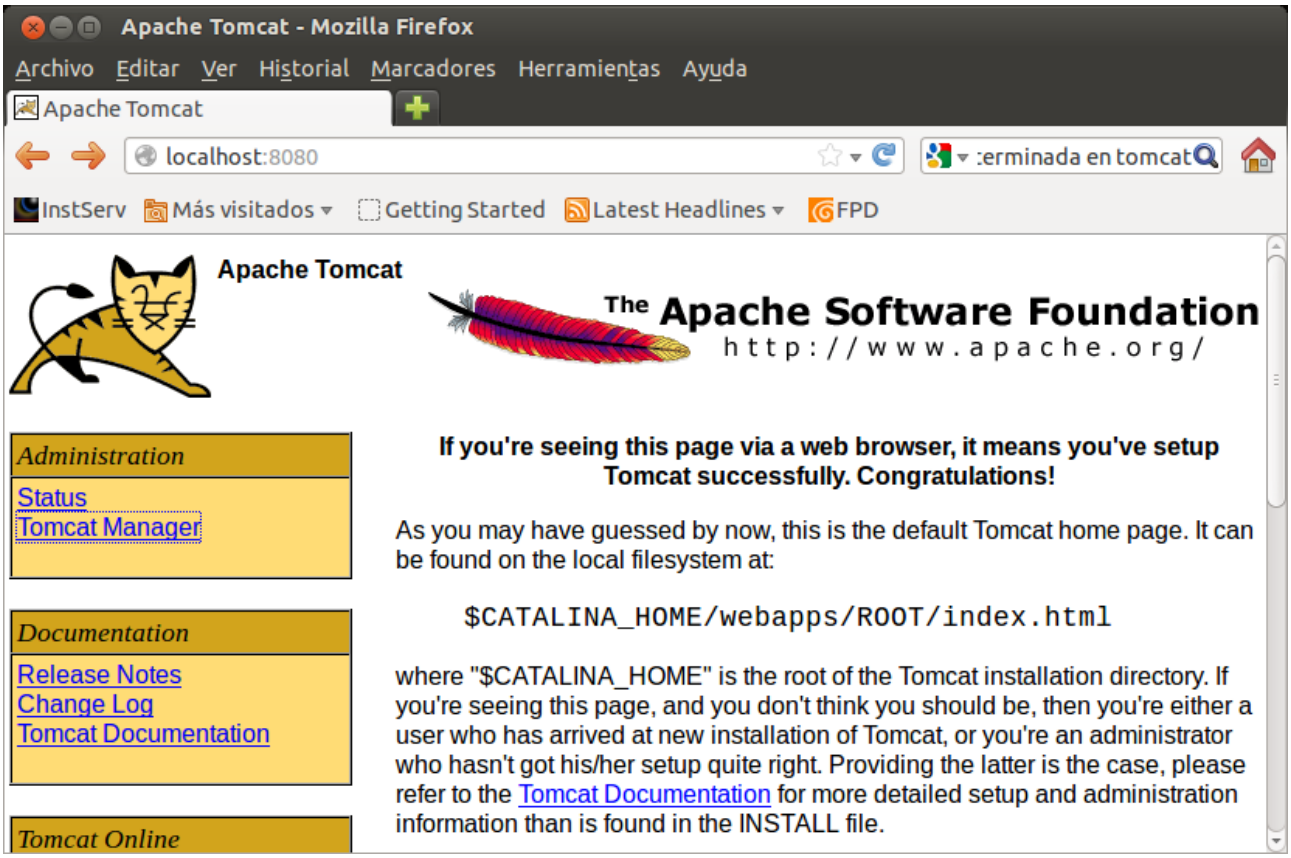
Hacemos un enlace para realizar las actualizaciones de forma más cómoda:

```
ln -s /usr/local/apache-tomcat-6.0.36/ /usr/local/tomcat
```

Y para finalizar arrancamos el servidor Tomcat tecleando:

```
sh /usr/local/apache-tomcat-6.0.36/bin/catalina.sh start
```

Como todo ha ido bien tecleando en el navegador `localhost:8080` accederemos a la página inicial de Tomcat.



Tarea 02 de DAW. José Luis Comesaña

Enunciado.

Una empresa quiere mostrar y operar con su negocio a través de Internet. Así quiere:

- ✓ Una página web visible a cualquiera en Internet que publicite su negocio: quienes somos -que contiene un vídeo presentación de la empresa en formato flv-, clientes habituales, donde estamos, novedades.
- ✓ Un lugar de la página web solamente accesible al personal de la empresa que tenga el rol 'admin'.
- ✓ Asegurar la comunicación del personal de la empresa.

Se pide en un servidor web Apache (apache2):

1. Configurar un **virtualhost basado en nombre** denominado **empresa-tarea-daw02** que permita el acceso de la página web de la empresa en Internet al directorio del servidor web: **todo-empresa-tarea-daw02**
2. Hacer accesible a través de Internet las siguientes URL que identifican a la empresa:
   www.empresa-tarea-daw02.local y **empresa-tarea-daw02.local**
3. Configurar en el servidor el **tipo MIME** posible que permite la identificación correcta del vídeo presentación formato **flv** situado dentro del directorio **videos** y de nombre **entrada.flv**.
4. Crear el subdirectorio **todo-empresa-tarea-daw02/delimitado** teniendo en cuenta que:
   a. El directorio **todo-empresa-tarea-daw02** permite el acceso a cualquier usuario.
   b. El subdirectorio **todo-empresa-tarea-daw02/delimitado** permite el acceso solamente al personal de la empresa que tenga el rol: **admin**.
5. Pemitir el protocolo HTTPS en el **virtualhost empresa-tarea-daw02**
6. Configurar los archivos de registro como sigue:
   a. Identificación log de acceso: **empresa-tarea-daw02-access.log**
   b. Identificación log de error: **empresa-tarea-daw02-error.log**
   c. Alias logformat: **combined**
7. Rotar logs por intervalo temporal: cada 24horas.

NOTAS IMPORTANTES:

- ✓ Los dominios **.local** no existen en Internet, con lo cual la tarea se comprobará en red local. Así para que las URL fuesen visibles en Internet realmente habría que comprar el dominio, dirigirlo a la IP del servidor web y expandirlo mediante Servidores DNS.
- ✓ Para la solución de la tarea simular la página web con dos archivos HTML:
  - ➔ Uno de nombre index.html en la raíz del directorio **todo-empresa-tarea-daw02** que contenga el texto 'ACCESO NO LIMITADO'.
  - ➔ Uno de nombre index.html en la raíz del directorio **todo-empresa-tarea-daw02/delimitado** que contenga el texto 'ACCESO LIMITADO'.
- ✓ La entrega de cada apartado de la tarea consiste en
  - ➔ indicar el archivo a configurar junto con el código necesario para resolver la cuestión correspondiente. Adjuntar los ficheros configurados.
  - ➔ Apoyar la documentación con capturas de pantalla. En alguna de ellas, se debe ver la plataforma con vuestra foto del perfil (la foto que os aparece arriba a la izquierda).

Criterios de puntuación. Total 10 puntos.

La valoración de cada apartado es:

1. 1 punto.
2. 1 punto.

3. 1 punto.
4. 2 puntos.
5. 3 puntos.
6. 1 punto.
7. 1 punto.

Recursos necesarios para realizar la Tarea.

- ✓ Los contenidos de la unidad.
- ✓ Un servidor web Apache2 instalado.
- ✓ Un navegador para comprobar la realización de la tarea.
- ✓ Un procesador de textos para elaborar la documentación y los archivos de la tarea.
- ✓ Un ordenador.
- ✓ Acceso a Internet.

Consejos y recomendaciones.

Ve realizando la tarea de forma secuenciada y al mismo tiempo ve documentando la solución de la misma.

Aunque existen varias posibilidades para controlar el acceso a los usuarios, te recomiendo que comiences a trabajar con la autenticación HTTP Basic. Una vez configurada puedes intentarlo mediante autenticación LDAP.

Te ayudará mucho saber que está pasando en cada momento en tu servidor web, así puedes comprobar en tiempo real que es lo que ocurre en el acceso a los directorios **todo-empresa-tarea-daw02** y **todo-empresa-tarea-daw02/delimitado** mediante el comando:

```
tail -f fichero.log
```

donde `fichero.log` identifica el nombre del fichero de registro a comprobar.

Resolución de la tarea

Para la realización del **punto 1** iremos a la carpeta `/var/www/` y creamos un directorio al que denominaremos "`todo-empresa-tarea-daw02`", para ello abrimos el terminal de Linux y nos logamos como `root` tecleando:

```
sudo su
```

Introducimos la contraseña de superusuario root y ya estamos logados para poder realizar todas las tareas de configuración del virtualhost. Ahora creamos el directorio:

```
mkdir /var/www/todo-empresa-tarea-daw02
```

Y creamos el fichero de configuración para el virtualhost que nos solicita este punto de la tarea:

```
gedit /etc/apache2/sites-available/empresa-tarea-daw02
```

Y tecleamos las siguientes líneas dentro de él:

```
<VirtualHost *:80>
   DocumentRoot /var/www/todo-empresa-tarea-daw02/
   ServerName www.empresa-tarea-daw02.local
   ServerAlias empresa-tarea-daw02.local
</VirtualHost>
```

Grabamos el fichero y procedemos ahora a activar los VirtualHosts recién creados mediante los comandos:

```
a2ensite empresa-tarea-daw02.local
```

Como no hemos dado de alta el dominio en un servidor DNS, para probarlo podemos editar el fichero `/etc/hosts` para que las peticiones a ese dominio vaya a nuestro servidor:

```
gedit /etc/hosts
```

Y dentro del fichero pondremos:

```
127.0.0.1 www.empresa-tarea-daw02.local
```

El **punto 2** de la tarea que indica que debemos hacer accesible a través de internet a las URL que identifican a la empresa, ya lo tenemos hecho mediante las órdenes `ServerName` y `ServerAlias` del apartado anterior.

Ahora, para el **punto 3**, necesitamos indicar que podríamos utilizar un fichero de vídeo de tipo `flv`, para lo que editamos el fichero `mime.conf` del servidor, tecleando:

```
gedit /etc/apache2/mods-available/mime.conf
```

Y vemos que posee una línea donde pone:

```
TypesConfig /etc/mime.types
```

Lo cual indica que los tipos de archivos conocidos se encuentran en ese fichero, por lo que lo editamos tecleando:

```
gedit /etc/mime.types
```

Y dentro del fichero buscamos si ya existe una referencia al tipo de ficheros `flv`, encontrando:

```
video/x-flv flv
```

Que nos dice que los ficheros de tipo `flv` ya están reconocidos por el servidor como ficheros de vídeo. Si no existiera dicha línea seríamos nosotros los que tendríamos que teclearla.

El **punto 4** nos pide que creemos un subdirectorio dentro de `todo-empresa-tarea-daw02` para restringir el acceso a todos los usuarios que no tengan el rol de `admin`:

```
mkdir /var/www/todo-empresa-tarea-daw02/delimitado
```

Modificamos el fichero de configuración virtualhost anterior para que permita el uso del fichero htaccess en el directorio que deseamos restringir su acceso:

```
gedit /etc/apache2/sites-available/empresa-tarea-daw02
```

Y le añadimos:

```
<Directory /var/www/todo-empresa-tarea-daw02/delimitado>
   AllowOverride All
</Directory>
```

Y a continuación creamos un fichero `.htaccess` en el directorio que deseamos controlar:

```
gedit /var/www/todo-empresa-tarea-daw02/delimitado/.htaccess
```

Y tecleamos:

```
AuthType Basic
AuthName "Area restringida para administradores"
AuthUserFile /etc/apache2/todo-empresa-tarea-daw02/passwd
AuthGroupFile /etc/apache2/todo-empresa-tarea-daw02/roles
Require group admin
```

Como le indicamos que se restringe el acceso sólo a los pertenecientes al grupo `admin`, hemos de crear dicho grupo en el lugar indicado, en este caso en el fichero `roles` que ubicaremos en `/etc/apache2/todo-empresa-tarea-daw02/`, por lo que tecleamos:

```
gedit /etc/apache2/todo-empresa-tarea-daw02/roles
```

Y dentro del fichero ponemos:

```
admin: root joseluis
```

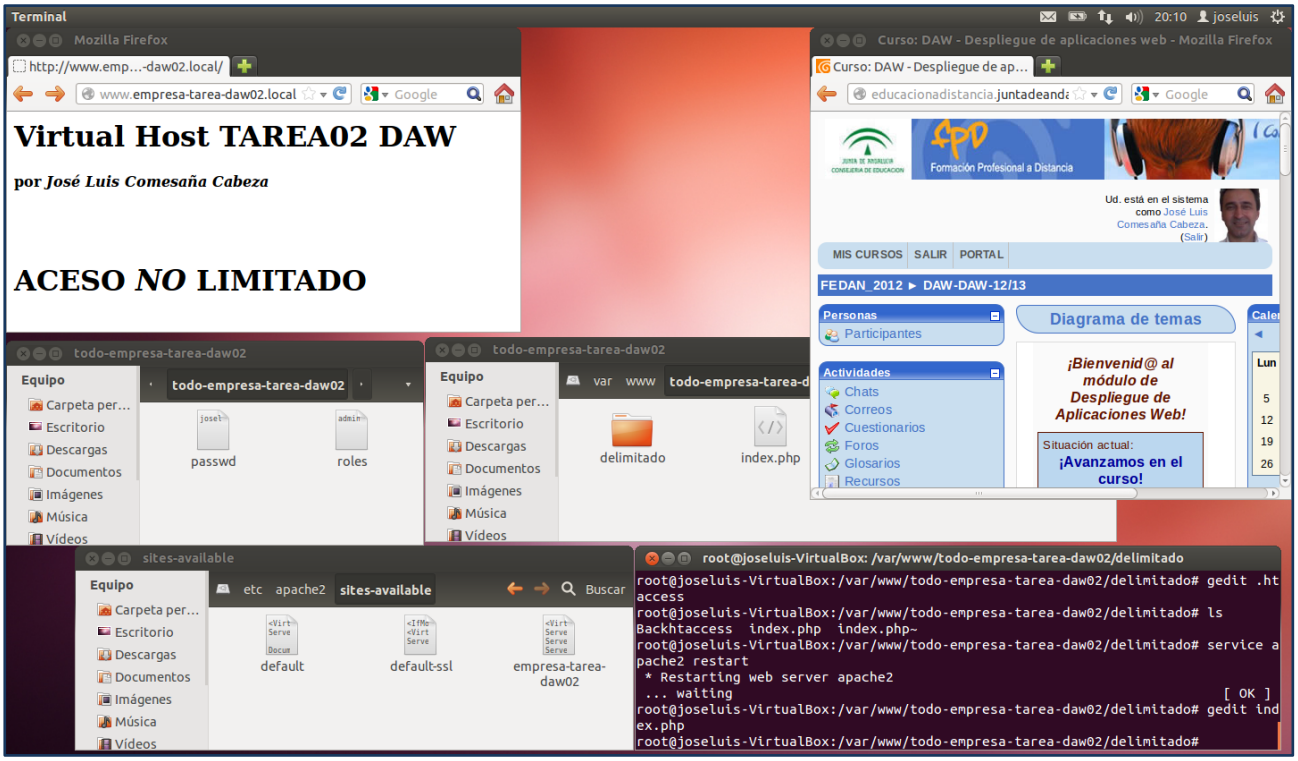
Con lo que asignamos al grupo `admin` los usuarios `root` y `joseluis`

Tenemos creado el grupo, pero no los usuarios, por lo que el siguiente paso será crear sus contraseñas de paso, para lo cual usaremos el mandato `htpasswd`, con el parámetro `-c` si lo vamos a crear por primera vez (*no posee usuarios creados*) o sin dicho parámetro para ir añadiendo nuevos miembros:

```
htpasswd -c /etc/apache2/todo-empresa-tarea-daw02/passwd joseluis
```

Introduzco la contraseña que quiera asignarle y me lo solicita nuevamente para confirmarla.

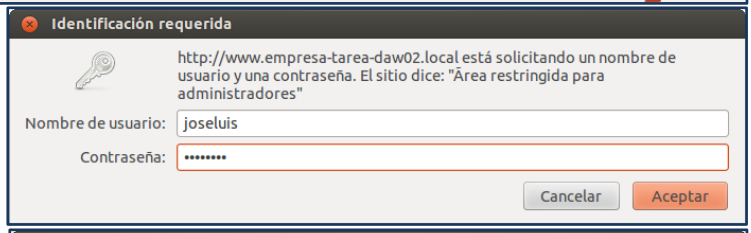
Reiniciamos el servidor con `service apache2 restart` y ya estamos preparados para probarlo, pero antes, como ambas carpetas están vacías, crearemos un index.html para cada una, quedando como muestro a continuación:



Escritorio con página principal de ACCESO NO LIMITADO

Cuando intentamos acceder a "limitado" nos pide usuario/contraseña, y si nos equivocamos nos muestra una pantalla de error:

Cuando introducimos el par usuario/contraseña de forma adecuada:



Conseguimos que nos muestre la página de forma correcta:

El **punto 5** nos pide que permitamos acceder al servidor creado, con HTTPS, para lo que instalaremos el openssl con:

```
apt-get install openssl
```

Y lo activamos con:

```
a2enmod ssl
```

Reiniciamos el servidor web apache:

```
service apache2 restart
```

Creamos ahora una carpeta, en la de apache por ejemplo, que contendrá los certificados, y a la que vamos a llamar `ssl`:

```
mkdir /etc/apache2/ssl
```

Nos situamos en la nueva carpeta

```
cd /etc/apache2/ssl
```

Lo primero es crear la clave privada del servidor. Lo hacemos mediante la generación aleatoria de una cadena de texto, tecleando:

```
openssl genrsa -des3 -out server.key 1024
```

Nos pide una frase de paso que debemos recordar (yo he puesto *cual es mi edad*), ya que la solicitará siempre que reiniciemos el servidor. Si quisiéramos deshabilitar el hecho de que la pida siempre (*no es recomendable*), teclearíamos:

```
openssl rsa -in server.key -out server.pem
```

Ahora generaremos el archivo `csr` mediante la orden:

```
openssl req -new -key server.key -out server.csr
```

Lo cual solicitará una serie de datos que hemos de introducir, teniendo especial atención en poner la web exacta del dominio que queremos activar con ssl. En nuestro caso, cuando nos pregunta "`Common Name`" teclearemos `www.empresa-tarea-daw02.local`

Algunos campos pueden tener valores por defecto, pero si deseamos dejar el campo en blanco tendremos que poner sólo un punto.

Con esto ya podemos emitir certificados para nuestro sitio, por lo que vamos a generar uno autofirmado, ya que no poseemos ninguno de una autoridad certificada. Tecleamos:

```
openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```

Hemos puesto que caduque al año, pero podríamos haber puesto cualquier cantidad de días.

Ya tenemos listos los certificados, por lo que pasaremos ahora a crear el virtualhost que lo contendrá y la carpeta donde lo ubicaremos, por lo que tecleamos:

```
mkdir /var/www/todo-empresa-tarea-daw02-ssl
```

Creamos un fichero `index.html` igual que hicimos para el virtualhost anterior, pero le ponemos un mensaje que nos indique que estamos ante una web segura.

Ahora nos dirigimos a `/etc/apache2/sites-available` y copiamos el contenido del virtualhost del punto anterior "`empresa-tarea-daw02`" y lo llamamos "`empresa-tarea-daw02-ssl`" para lo que tecleamos:

```
cd /etc/apache2/sites-available
cp empresa-tarea-daw02 empresa-tarea-daw02-ssl
```

Y editamos su contenido con `gedit` para cambiarlo por:

```
<VirtualHost *:443>
   ServerAdmin webmaster@empresa-tarea-daw02.local
   ServerName www.empresa-tarea-daw02.local
   ServerAlias empresa-tarea-daw02.local
   DocumentRoot /var/www/todo-empresa-tarea-daw02-ssl/
   SSLEngine on
   SSLCertificateFile /etc/apache2/ssl/server.crt
   SSLCertificateKeyFile /etc/apache2/ssl/server.pem
</VirtualHost>
```

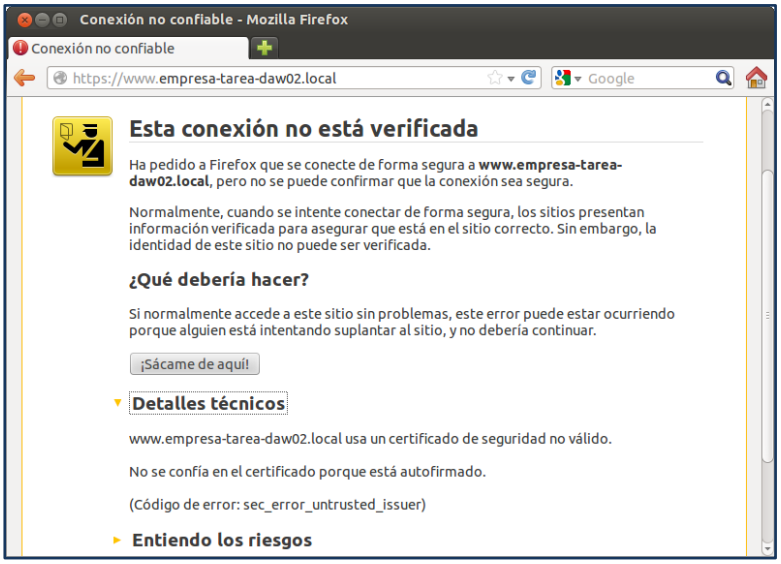
Ya lo tenemos listo y sólo queda recargar y reiniciar apache para que entre en funcionamiento la nueva configuración:

```
service apache2 reload
service apache2 restart
```

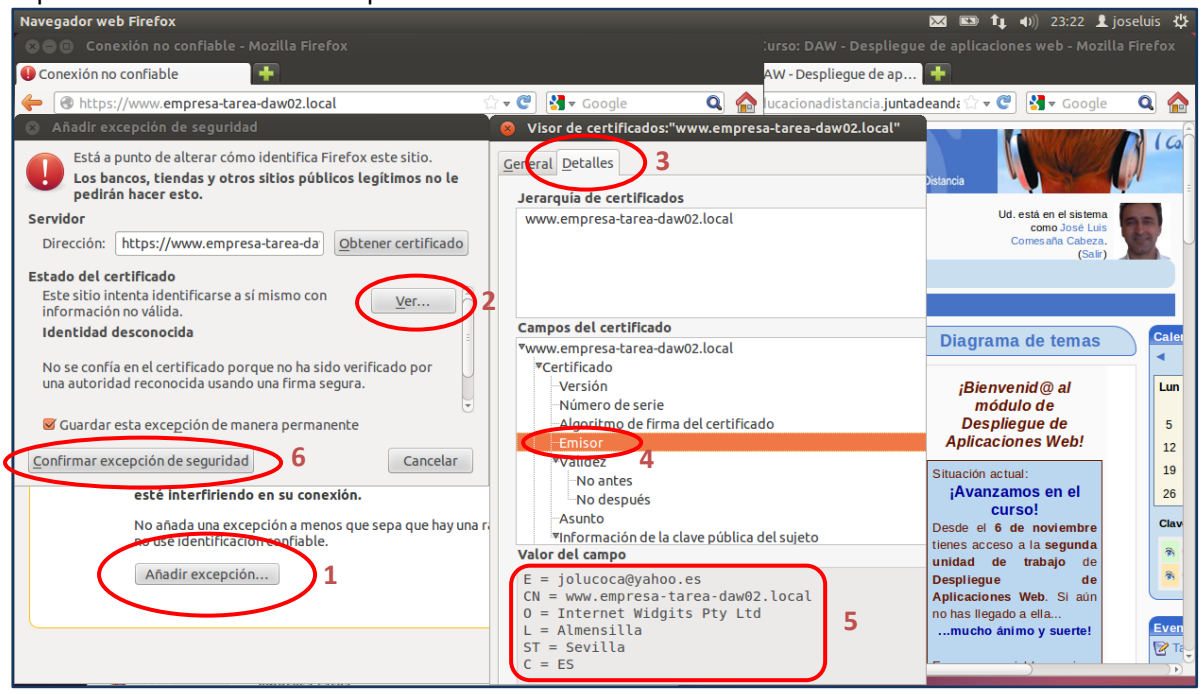
Ahora para comprobar que todo ha salida bien, nos dirigiremos al navegador y en su barra de direcciones teclearemos:

```
https://www.empresa-tarea-daw02.local
```

Y aparece lo siguiente:



Si pulsamos sobre añadir excepción veremos:



Y cuando pulsamos sobre "*Confirmar excepción de seguridad*" nos aparece:

Ahora, para el **punto 6** nos dirigimos a la carpeta donde tenemos configurado el virtualhost y editamos su fichero de configuración tecleando:

```
cd /etc/apache2/sites-available
gedit empresa-tarea-daw02
```

Y añadimos al final del mismo, pero dentro de las etiquetas VirtualHost, las siguientes líneas:

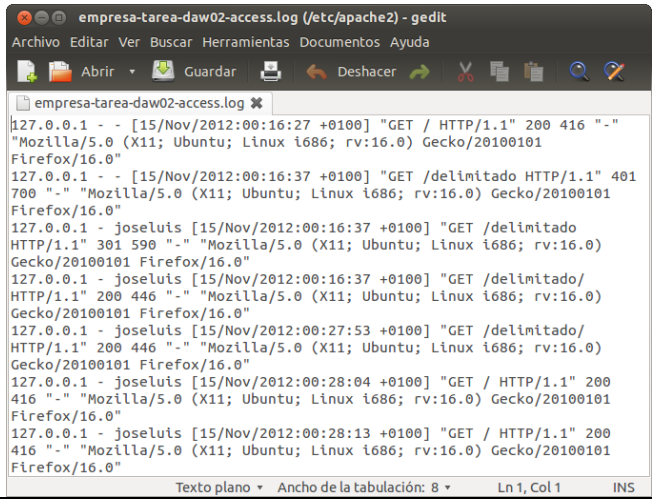
```
LogFormat "%h %l %u %t \"%r\" %>s %b" combined
CustomLog empresa-tarea-daw02-access.log combined
ErrorLog "empresa-tarea-daw02-error.log"
```

Con la primera línea asignamos el alias `combined` a `logformat`, con el que generaremos registros en el formato

- `%h` host
- `%l` identificación_cliente
- `%u` usuario
- `%t` fecha_y_hora
- `\"%r\"` cliente, recurso, protocolo
- `%>s` Código de estado que el servidor envía de vuelta al cliente. Si el código comienza por 2 fue respondido con éxito, por 3 identifican una redirección, por 4 se trata de un error del cliente y por 5 es un error del servidor.
- `%b` Tamaño del objeto retornado por el cliente.

Si cuando nos tenga que presentar los datos, alguno de ellos no existe, pondrá un guión en el lugar que le corresponda.

Con la segunda línea conseguimos indicar que el fichero de registro de acceso se guardará en el directorio del servidor por defecto, con el nombre de `empresa-tarea-daw02-access.log` y con el contenido que se adapte al formato establecido en la línea anterior:



La tercera línea se usa para indicar dónde se ha de guardar el registro de errores, en este caso, en el fichero `empresa-tarea-daw02-error.log` situado en el directorio por defecto del servidor.

Para realizar el **punto 7** y último, a lo introducido en el punto anterior, añadiremos:

```
CustomLog "|/usr/sbin/rotateCustom /etc/apache2/empresa-tarea-daw02-access.log 86400" combined
ErrorLog "|/usr/sbin/rotateError /etc/apache2/empresa-tarea-daw02-error.log 86400" combined
```

Con lo que le indicamos que el fichero del registro de acceso `empresa-tarea-daw02-access.log` que se encuentra situado en `/etc/apache2` lo almacene cada 24 horas (*86400 segundos*) en el fichero `rotateCustom` que se encuentra en `/usr/sbin` utilizando para ello el alias `combined`
Con la segunda línea, decimos que el fichero de registro de errores `empresa-tarea-daw02-error.log` que se encuentra en `/etc/apache2` lo almacene cada 24 horas (*86400 segundos*) en el fichero `rotateError` que se encuentra en `/usr/sbin` utilizando para ello el alias `combined`

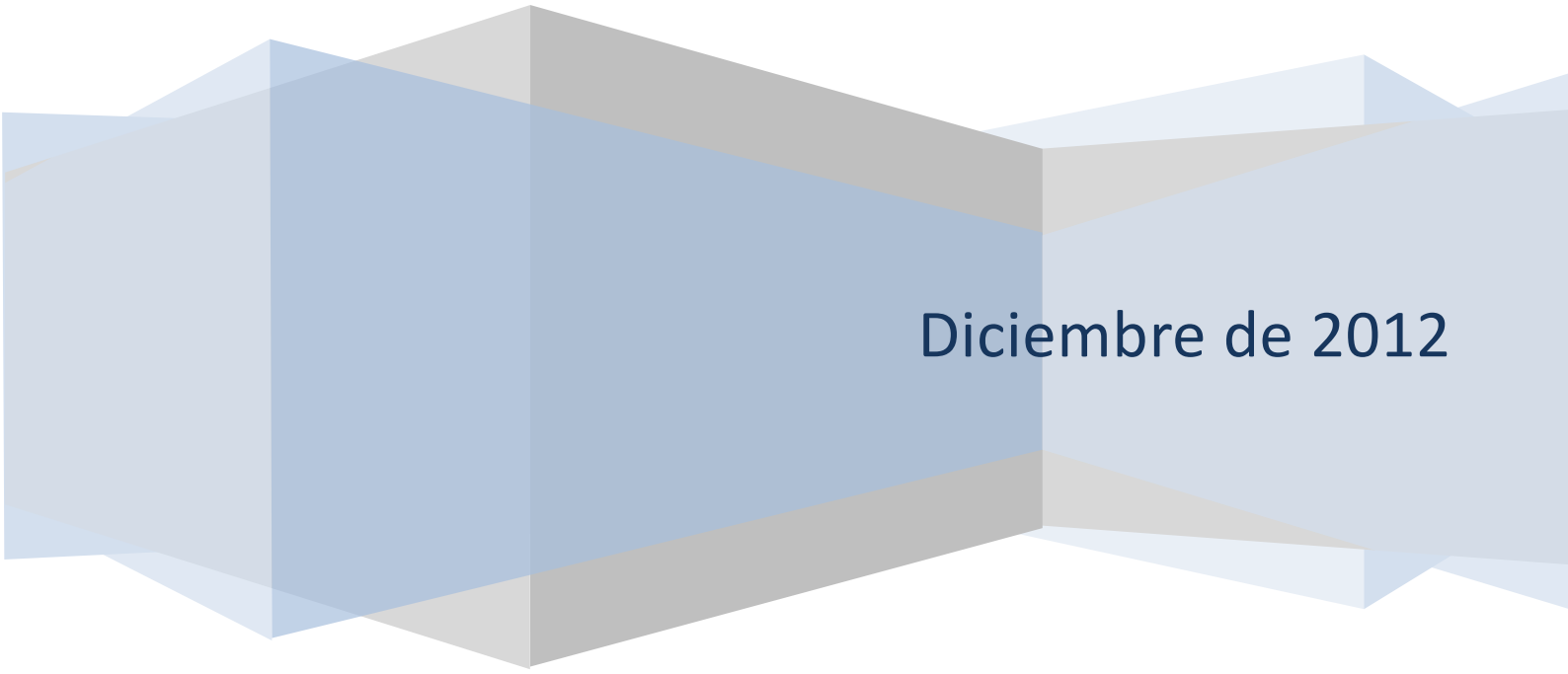
NOTA Cuando reiniciaba o recargaba apache me aparecía un error, el cual conseguí arreglar creando un fichero de nombre `httpd.conf`(*que no existía*), en `/etc/apache2/` y le introduje `ServerName localhost` como único valor.

Módulo Superior de Diseño de Aplicaciones Web

Despliegue de Aplicaciones Web

Tarea 3

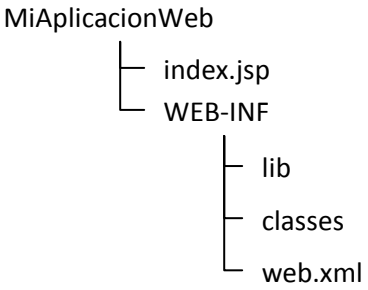
José Luis Comesaña Cabeza



Diciembre de 2012

1. **Una aplicación web puede ser desplegada en diferentes servidores web manteniendo su funcionalidad y sin ningún tipo de modificación en su código debido a la especificación servlet 2.2 , ¿cúal es la estructura de directorios que debe tener?**

   Ha de tener un directorio principal o raíz que contendrá los ficheros estáticos y JSP's. En su interior ha de existir una carpeta denominada `WEB-INF` que contendrá el fichero `web.xml` encargado de configurar la aplicación. Dentro de esta carpeta tendremos dos supcarpetas, una denominada `classes` que contiene los ficheros compilados y otra denominada `lib` que tendrá las librerías adicionales. Al mismo nivel de `WEB-INF` estará el resto de carpetas para ficheros estáticos.

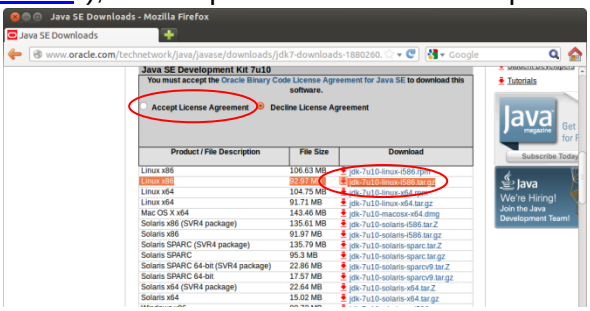
2. ***Ant* se basa en ficheros XML, normalmente configuramos el trabajo a hacer con nuestra aplicación en un fichero llamado *build.xml*. Indica alguna de las etiquetas con las que podemos formar el contenido de este archivo:**

   Con la etiqueta `project` creamos el elemento raíz del fichero xml, por lo que sólo ha de haber uno en todo el fichero, es decir que todas las restantes etiquetas tendrán que estar encerradas entre `<project>` y `</project>`
   Con `<target>` definiremos un conjunto de tareas que deseamos ejecutar sobre nuestra aplicación. Podríamos hacer que unos objetos dependan de otros para que Ant lo trate de forma automática. Con el atributo `name` le daremos nombre a la tarea y con el atributo `depends` le indicaremos de qué otras tareas depende la realización de ésta. También podríamos describir la tarea con el atributo `description`.
   Con la etiqueta `property` podemos indicar qué significa (`value`) la propiedad indicada (`name`)
   `<task>` es la tarea a realizar según el propósito, como manipulación de archivos, comprensión o compilación como por ejemplo:
   `<move file="archivo.origen" todir="dir/destino" />`

3. **Dispones de una máquina que cuenta con el sistema operativo *Ubuntu* recientemente actualizado, en la que está el entorno de red configurado y, además, dispones de conexión a Internet y estás trabajando con la cuenta del usuario root. Indica cada uno de los pasos, y comandos implicados en ellos, (y *añade las capturas de pantalla necesarias*, en alguna de ellas, se debe ver la plataforma con vuestra foto del perfil) para conseguir hacer lo siguiente:**
   1. **Instalar el JDK de Sun.**
   2. **Crea una aplicación web, basta con que crees simplemente un fichero index.html o index.jsp, similarmente a como se ve en el apartado 2.1. del tema, y que como título de la ventana ponga tu localidad y en el cuerpo (Body) ponga tu nombre y apellidos.**
      El nombre de la aplicación, en vez de Aplic_Web como en el ejemplo del apartado, que sea Ejemplotar3.
   3. **Indica los pasos a seguir para generar un log de acceso, no olvides al enviar la tarea, adjuntar el fichero o ficheros que hayas modificado (haz una copia de seguridad por si te equivocas al modificar).**
   4. **Descargar e instalar JBoss Application Server (no instalarlo con Synaptic).**
   5. **Crear usuario para JBoss.**
   6. **Establecer las variables de entorno correspondientes para Java y JBoss.**
   7. **Cambiar la contraseña del administrador de JBoss.**
   8. **Instala y configura Ant.**

   Repuestas a este punto:

3.1 Para poder **instalar JDK de Sun** lo primero es descargarlo desde su propia web (www.oracle.com/technetwork/java/javase/downloads), donde pulsaremos sobre la opción de Java Platform (JDK) 7u10 que es la última versión disponible del JDK. Aceptamos la licencia de agradecimiento y escogemos la tar.gz de 32 bits para Linux y y una vez descargada, la descomprimimos con:




```
tar -xvf jdk-7u10-linux-i586.tar.gz
```

y ya lo tendremos descargado, pero en nuestra carpeta Descargas, por lo que tendremos que crear una carpeta que contendrá esta versión de jdk (previamente nos logamos como superusuarios con `sudo su`):

creamos la carpeta con: `mkdir -p /usr/lib/java-1.7.10`

movemos el jdk descargado y descomprimido: `mv jdk-7u10-linux-i586/* /usr/lib/java-1.7.10/`

Ahora instalamos la máquina virtual java y el compilador con:

```
update-alternatives --install "/usr/bin/java" "java" "/usr/lib/java-1.7.10/bin/java" 1
```

y para el compilador:

```
update-alternatives --install "/usr/bin/javac" "javac" "/usr/lib/java-1.7.10/bin/javac" 1
```

Cuando hayamos finalizado con la instalación, le hemos de indicar que use la versión instalada (ya que en mi caso tengo varias) para lo que ejecutaremos:

```
update-alternatives --config java
```

y seleccionaremos la versión recién instalada.

3.2 Para poder crear una aplicación web en nuestro servidor hemos de crear dos variables de entorno, una para indicar la ubicación de los archivos binarios de Java y otra para indicar la ubicación de los scripts de Tomcat, por lo que editamos el fichero `/etc/profile` y añadimos:

```
CATALINA_HOME=/usr/local/apache-tomcat-6.0.36/
JAVA_HOME=/usr/lib/java-1.7.10/jre/
PATH=$PATH:$JAVA_HOME/bin:$CATALINA_HOME
export PATH JAVA_HOME CATALINA_HOME
```

Una vez grabado el fichero, tenemos que actualizar las variables de entorno tecleando:

```
source /etc/profile
```

Ahora creamos la carpeta que contendrá nuestra aplicación poniéndole el nombre que se indica para esta tarea: `Ejemplotar3`. Y dentro de ella la carpeta `WEB-INF` que contenga a su vez las carpetas `classes` y `lib`

```
mkdir Ejemplotar3
mkdir Ejemplotar3/WEB-INF
mkdir Ejemplotar3/WEB-INF/classes
mkdir Ejemplotar3/WEB-INF/lib
```

Una vez terminada la estructura, creamos nuestro fichero `index.html`, para lo que podemos teclear:

```
gedit Ejemplotar3/index.html
```

y en su interior introducimos:

```
<html>
  <head><title>Almensilla (Sevilla)</title>  </head>
  <body>
     <h1 align="center">Aplicaci&oacute;n web para la tarea 3 de DAW</h1>
     <div align="center">
        <p>Realizado por:</p>
        <p><b><i>Jos&eacute; Luis Comesa&ntilde;a Cabeza</i></b></p>
     </div>
  </body>
</html>
```

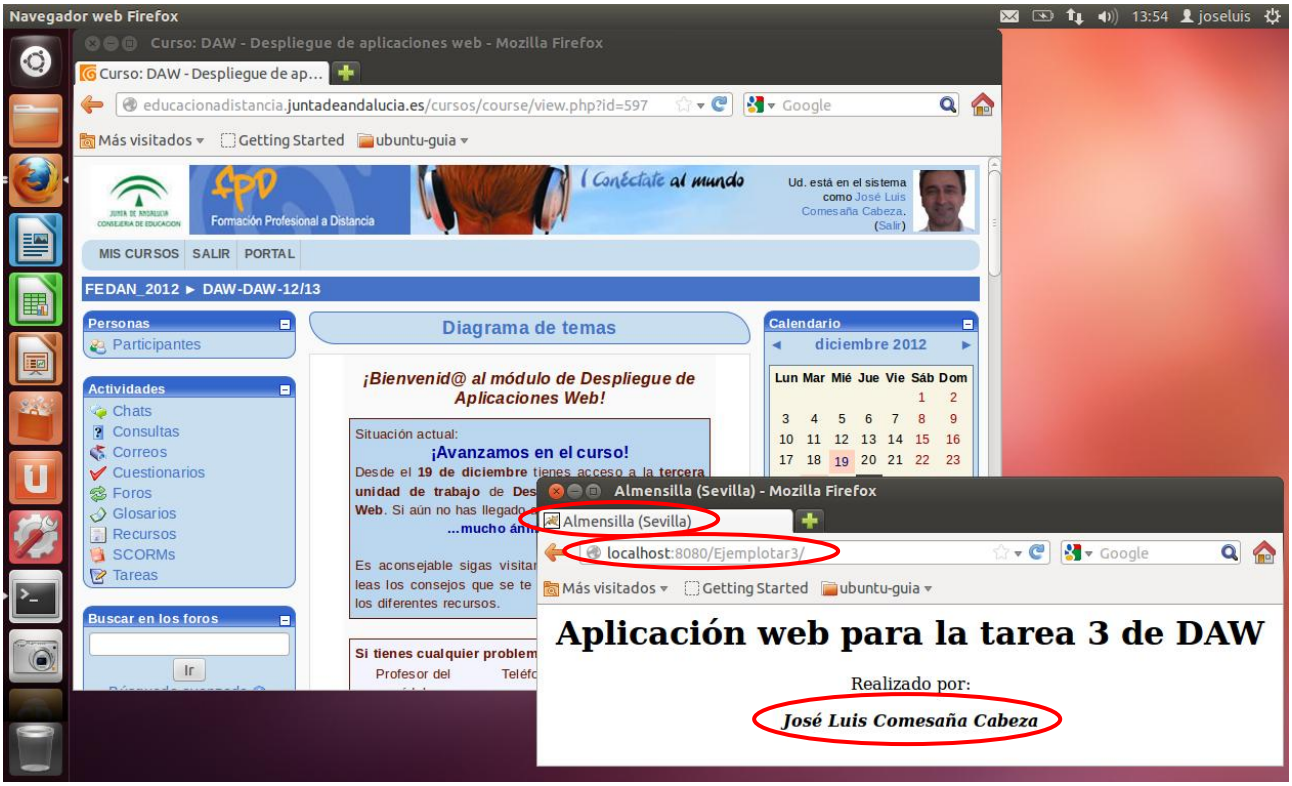
Por último, copiamos la carpeta `Ejemplotar3` completa al subdirectorio `webapps` del directorio donde tenemos instalado tomcat, para lo que tecleamos:

```
cp -r ./Ejemplotar3 /usr/local/apache-tomcat-6.0.36/.
```

Y ya podemos probarlo tecleando en el navegador web:

```
http://localhost:8080/Ejemplotar3
```

obteniendo:



3.3 Para crear un log de acceso:

Modificaremos el fichero `server.xml` para que le afecte a todas las aplicaciones desplegadas en el servidor tomcat. Antes realizaremos una copia mediante:

```
cp /usr/share/tomcat7/conf/server.xml /usr/share/tomcat7/conf/serverBack.xml
```

ya que yo tengo instalada la versión 7.0.34 de tomcat y lo situé en el directorio `/usr/share/tomcat7`.

Ahora edito el fichero con:

```
gedit /usr/share/tomcat7/conf/server.xml
```

y nos desplazamos hasta el apartado donde se indica `<Host name="localhost"…` y descomentamos la línea que comienza por `<Valve className="org.apache.catalina.valves.AccessLogValve"…` y le cambiamos el `pattern="common"` por otro que nos recoja más información, quedando:

```
<Valve className="org.apache.catalina.valves.AccessLogValue" directory="logs"
    prefix="logDeAccesoLocalhost." suffix=".txt"
    pattern=" %h %u %t &quot;%r&quot; %s %b %D" />
```

Utilizándose `directory` para indicarle el nombre del directorio que contendrá el fichero log, `prefix` para decirle cómo queremos que sea el comienzo del nombre del fichero log a crear y `suffix` que le indica con qué caracteres o extensión deseamos que finalice dicho nombre. Así mismo, estamos usando como plantilla o pattern los códigos:

%h el nombre del host (o la dirección IP si hubiésemos utilizado la cláusula `resolveHosts` con valor de falso, y que al no usarla toma el valor de verdadero por defecto)
%u El usuario remoto que fue autentificado
%t Fecha y hora del acceso
%r Primera línea de la petición (método y petición URI)
%s Código de la respuesta HTTP
%b Bytes enviados excluyendo la cabecera HTTP
%D Tiempo necesario para procesar la solicitud, en milisegundos

" Símplemente imprime unas comillas.

Una vez guardado el fichero, iniciamos tomcat con la orden:

```
sh /usr/share/tomcat7/bin/startup.sh
```

Y ya podemos probarlo accediendo a la aplicación creada para que se grabe el acceso realizado. En el navegador web ponemos `localhost:8080/Ejemplotar3/` y aparecerá la pantalla mostrada en el punto anterior.
Una vez realizado dicho acceso al servidor de aplicaciones, si accedemos a `/usr/share/tomcat7/logs` veremos que existe un fichero nuevo denominado: `logDeAccesoLocalhost.2012-12-27.txt` cuyo contenido será, por ejemplo, si hemos llamado a una página inexistente denominada `hola` y posteriormente llamamos a otra, también inexistente, llamada `adios`:

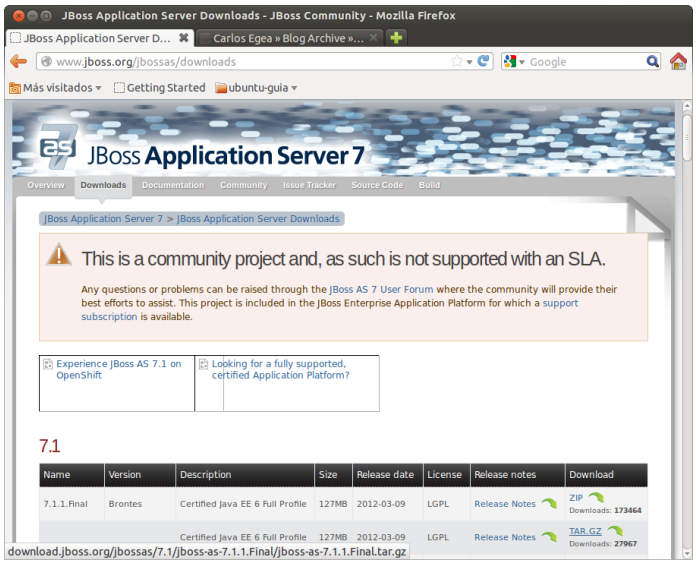
```
127.0.0.1 ???i??? - [27/Dec/2012:20:25:14 +0100] "GET /Ejemplotar3/hola HTTP/1.1" 404 993 134
127.0.0.1 ???i??? - [27/Dec/2012:20:28:05 +0100] "GET /Ejemplotar3/adios HTTP/1.1" 404 995 15
```

(*Adjunto el fichero server.xml y el fichero logDeAccesoLocalhost.2012-12-27.txt*)

3.4 Para poder instalar JBoss Application Server, previamente debemos tener instalado el entorno de ejecución de Java, para lo que podemos verificarlo tecleando `java -version`, y en mi caso me aparece:

```
java versión "1.7.0_10"
Java™ SE Runtime Environment (build 1.7.0 10-b18)
Java HotSpot(TM) Client VM (build 23.6-b04, mixed mode)
```

Es decir, que puedo pasar directamente a la descarga de JBoss, para lo que me dirijo a la dirección:
http://www.jboss.org/jbossas/downloads/



Selecciono la versión `jboss-as-7.1.1.Final.tar.gz`, ya que es la última disponible y requiere de Java 6 o superior, por lo que no debo tener problemas para instalarlo.

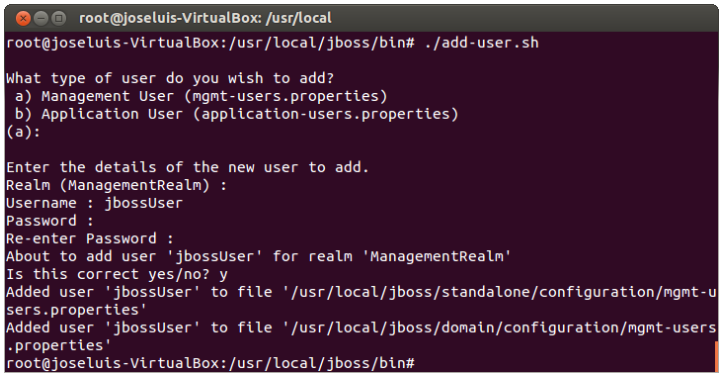
Ahora nos situamos en la carpeta donde se ha realizado la descarga y descomprimimos el fichero descargado:

```
cd /home/joseluis/Descargas
tar xvfz jboss-as-7.1.1.Final.tar.gz
```

A continuación podemos mover el directorio recién creado a la carpeta `jboss` que ubicaremos en `/usr/local`:

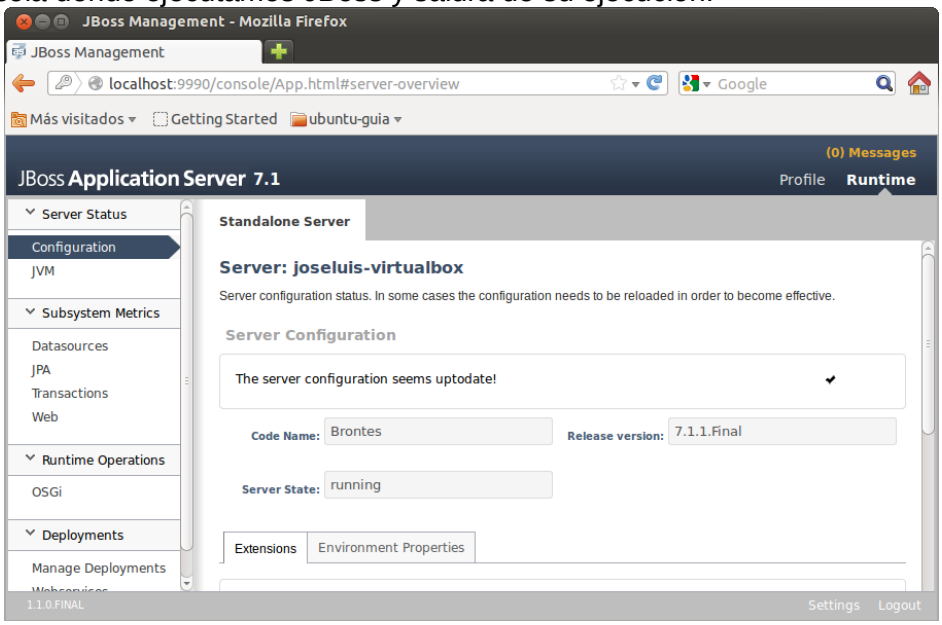
```
mv /home/joseluis/Descargas/jboss-as-7.1.1.Final /usr/local/jboss
```

3.5 Ahora crearemos un usuario para JBoss, y como en esta versión existe un fichero que nos permite realizar este proceso, vamos a utilizarlo, para lo que nos desplazaremos al directorio `bin` del jboss con la orden `cd /usr/local/jboss/bin` y ahora ejecutamos `./add-user.sh` preguntándonos por el tipo, nombre del usuario y su

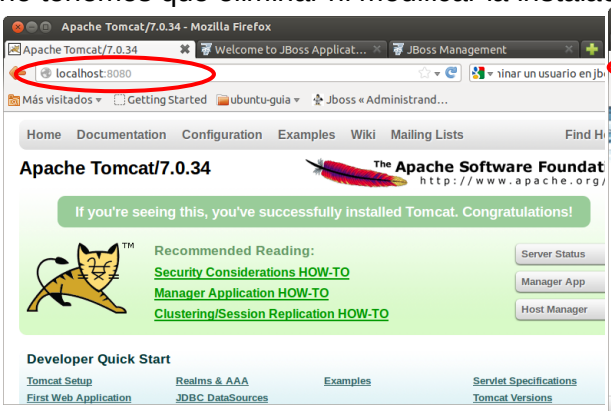


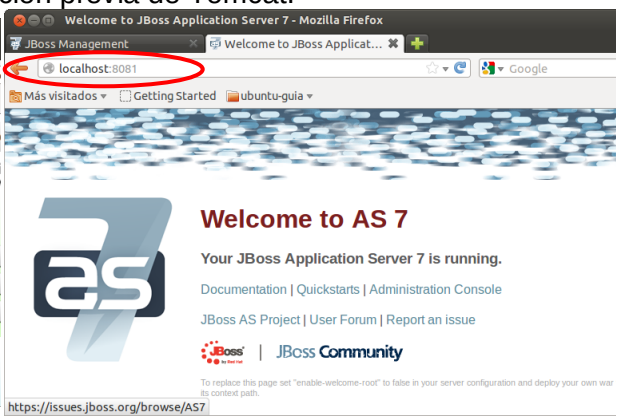
contraseña (por duplicado para confirmarla).

Cuando ya tenemos creado el usuario podemos arrancar JBoss tecleando `/usr/local/jboss/bin/standalone.sh` y para probarlo podemos teclear en la barra de direcciones del navegador `localhost:9990` y aparecerá la pantalla que muestro a continuación (*una vez que hayamos introducido el usuario y contraseña creado anteriormente con* `add-user.sh`). Si deseamos detenerlo sólo tendremos que pulsar **Ctrl-C** en la consola donde ejecutamos JBoss y saldrá de su ejecución.



No podemos poner `localhost:8080` porque el puerto lo tenemos configurado para utilizarlo con Tomcat desde los ejercicios anteriores, pero accediendo a través del puerto `9990` lo que hacemos es abrir la ventana de mantenimiento, la cual nos permite configurar el JBoss, y por tanto, podremos cambiarle el puerto de inicio para las aplicaciones http y que puedan "escuchar" por otro puerto distinto, como por ejemplo por el `8081`. Para ello, y dentro del apartado `Profile`, seleccionamos `General Configuration/Socket Binding/http` pulsamos sobre el botón de `Edit` y cambiamos el valor `8080` por `8081` para `Port`. Pulsamos sobre `Save` y reiniciamos JBoss para que entre en funcionamiento esta nueva configuración y podamos acceder a la página de inicio de JBoss tecleando en el navegador `localhost:8081` con lo que no tenemos que eliminar ni modificar la instalación previa de Tomcat.





3.6 Para establecer las variables de entorno de Java y jboss tenemos que editar el fichero `profile` de la carpeta `/etc`, por lo que tecleamos:

```
gedit /etc/profile
```

y al final añadiríamos:

```
JBOSS_HOME=/usr/local/jboss/
CATALINA_HOME=/usr/share/tomcat7/
JAVA_HOME=/usr/lib/jdk1.7.0_10/jre/
PATH=$PATH:$JAVA_HOME/bin:$CATALINA_HOME:$JBOSS_HOME/bin
export PATH JAVA_HOME CATALINA_HOME JBOSS_HOME
```

Las instrucciones marcadas en color verde ya las tenía creadas para los ejercicios anteriores.
Sólo nos queda ejecutar `source /etc/profile` para que el sistema pueda recoger el contenido de las variables creadas sin que tengamos que reiniciar el equipo

3.7 Para cambiar la contraseña del administrador de JBoss ejecutaremos `add-user.sh` de la carpeta `bin` y la primera pregunta que nos hace es sobre qué tipo de usuario, ***(a)*** para un usuario de administración y ***(b)*** para un usuario de aplicación, por lo que seleccionamos ***(a)*** y pulsamos intro.
A continuación nos pide el nombre del dominio utilizado para asegurar las interfaces de administración, y que por defecto es "`ManagementRealm`". Pulsamos intro para mantener este nombre y pasamos a la pregunta sobre el nombre del usuario, y que en este caso pondremos "admin", aunque no es demasiada buena idea ya que sería fácil de descubrir dicho nombre. Nos pide la contraseña por duplicado y finalizamos indicándole "yes" para que grabe la información.
Si deseamos eliminar un usuario creado, sólo tendremos que editar el fichero `mgmt-users.properties` que se encuentra en `jboss/standalone/configuration` y en su interior aparecerá una línea del tipo:

```
admin=78c1d6fec8b67d3a…
```

La numeración indica su contraseña codificada y `admin` es el nombre del usuario creado con `add-user.sh`, por lo que si borramos esta línea estaremos borrando ese usuario.
Esta misma acción tendremos que hacerla para el fichero del mismo nombre pero de la carpeta `domain/configuration`.

3.8 Para instalar Ant debemos tener previamente instalado Java, y como en mi caso ya lo tengo instalado correctamente, paso a descargar Ant desde su página oficial http://ant.apache.org/bindownload.cgi y seleccionamos para su descarga el fichero comprimido de tipo tar.gz y lo descomprimimos con:

```
tar xvfz apache-ant-1.8.4-bin.tar.gz
```

Ahora movemos la carpeta recién creada a `/usr/local` aprovechando para cambiarle el nombre por otro más corto:

```
mv /home/joseluis/Descargas/apache-ant-1.8.4 /usr/local/apacheAnt184
```

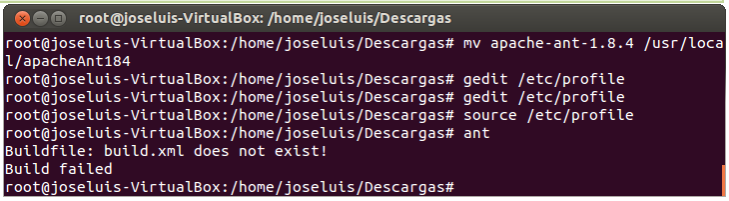
Editamos el fichero `/etc/profile` para incluirle las líneas de path propias de Ant (las marcadas en rojo):

```
ANT_HOME=/usr/local/apacheAnt184/
JBOSS_HOME=/usr/local/jboss/
CATALINA_HOME=/usr/share/tomcat7/
JAVA_HOME=/usr/lib/jdk1.7.0_10/jre/
PATH=$PATH:$JAVA_HOME/bin:$CATALINA_HOME/bin:$JBOSS_HOME/bin:$ANT_HOME/bin
```

para finalizar ejecutamos `source /etc/profile` para que entren en vigor los cambios realizados y si ejecutamos en el terminal la orden `ant` veremos que nos aparece:

```
Buildfile: build.xml does not exist!
Build failed
```

Lo cual nos indica que está totalmente instalado y configurado.

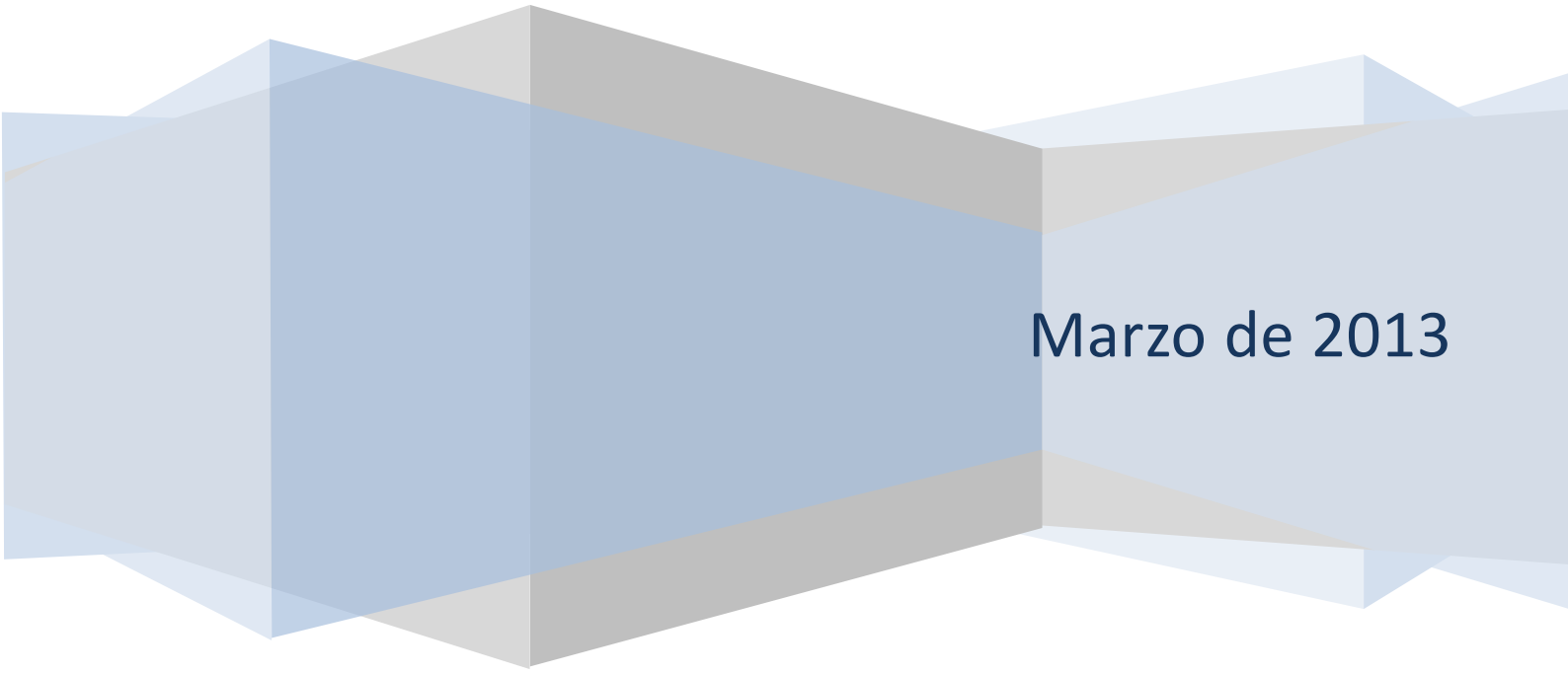


Módulo Superior de Diseño de Aplicaciones Web

Despliegue de Aplicaciones Web

Tarea 4

José Luis Comesaña Cabeza



Marzo de 2013

1. Instalar un servidor ftp: ProFTPD(proftpd)

A la hora de instalar ProFTPd, debemos tener en cuenta que nos debe permitir subir archivos de gran tamaño y que nos permita constantes accesos al servidor, por lo que instalaremos `proftpd` como servidor independiente, para lo cual hemos de buscar su paquete en el sistema.

Para la realización de los ejercicios vamos a logarnos como administradores con `sudo su` y a continuación:

1-1 Actualizamos los repositorios:

```
apt-get update
```

1-2 Actualizamos el sistema operativo

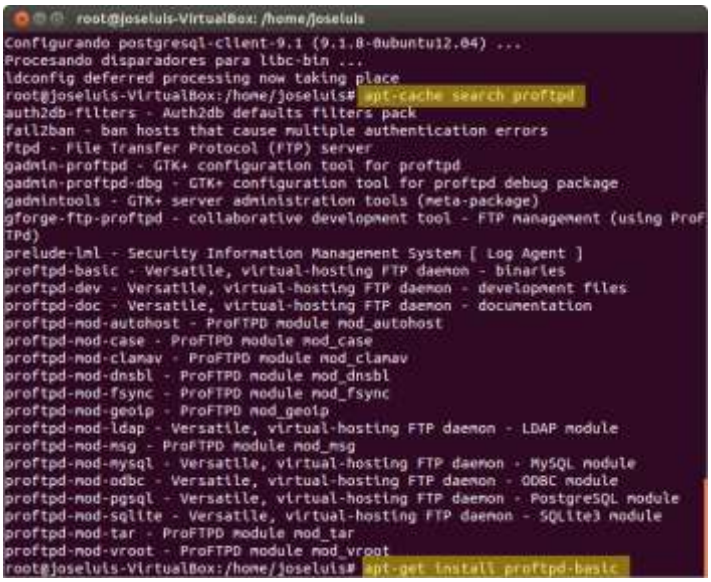
```
apt-get upgrade
```

1-3 Buscamos el paquete proftpd

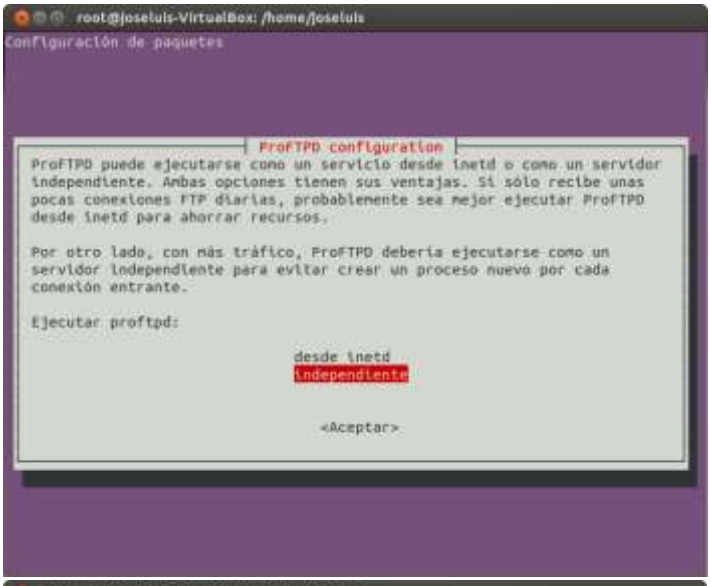
```
apt-cache search proftpd
```

1-4 Vemos que disponemos del paquete basic que es el que instalamos

```
apt-get install proftpd-basic
```



1-5 Lo instalamos como servidor FTP independiente, ya que se pretende tener un tráfico de datos importante

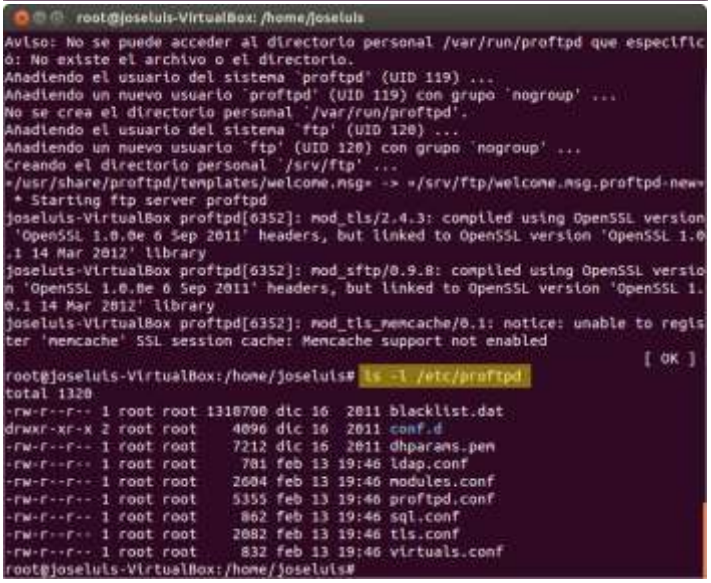


1-6 Comprobamos que ya lo tenemos instalado en /etc

```
ls -l /etc/proftpd
```

1.7 activamos el servicio proftpd:

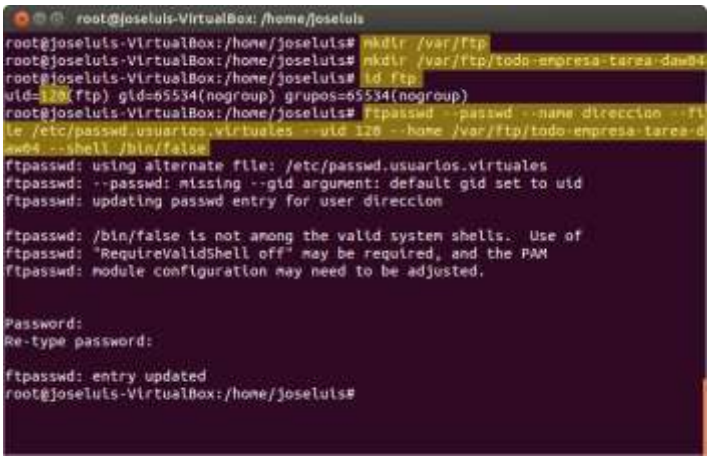
```
service proftpd start
```



2. *Configurar el servidor ProFTPd con:*

2.1. El usuario virtual: `dirección` con permisos de escritura

Para ello hemos de crear el directorio al que deseamos acceder a través de ftp (*esto lo deberíamos haber realizado anteriormente, pero como aún no hemos accedido a él, podemos hacerlo ahora*)




```
mkdir /var/ftp
mkdir   /var/ftp/todo-empresa-tarea-daw04
```

Ahora, para crear el usuario virtual hemos de comprobar el nº de identificación del usuario del sistema ftp, por lo que tecleamos:

```
id ftp
```

y el número que nos da es el que usaremos para la creación del usuario virtual, por lo que pondremos:

```
ftpasswd --passwd --name direccion --file /etc/passwd.usuarios.virtuales --uid 120 --home /var/ftp/todo-empresa-tarea-daw04 --shell /bin/false
```

con lo que le estaremos indicando que vamos a crear un usuario denominado `direccion`, sobre el que nos preguntará una contraseña que será almacenada en `/etc/passwd.usuarios.virtuales`. Este usuario tendrá el número de identificación `120`, y su carpeta principal será la que se encuentra en `/var/ftp/todo-empresa-tarea-daw04`, y no tendrá acceso al shell del sistema.

Las carpetas recién creadas tendrán que ser propiedad del usuario del sistema `ftp` por lo que pondremos:

```
chown ftp /var/ftp -R
```

con lo que le indicamos que el usuario `ftp` será el propietario de la carpeta `ftp` y de todas sus subcarpetas.

2.2 Un virtualhost basado en IP que permita el acceso, mediante ftp, al directorio del servidor ftp: `todo-empresa-tarea-daw04`

Para realizar este punto hemos de editar el fichero de configuración de proftpd:

```
gedit /etc/proftpd/proftpd.conf
```

y descomentamos la línea que referencia a la inclusión del fichero que nos permite configurar virtualhosts (por defecto viene con la extensión `.con`, por lo que también le indicaremos que es `.conf`):

```
Include /etc/proftpd/virtuals.conf
```

Ahora toca comprobar la dirección IP que posee nuestro servidor, por lo que valiéndonos de la orden `ifconfig`, comprobamos que es la `192.168.1.50`, que es la que nos valdrá para crear nuestro virtualhost basado en IP, por lo que entramos en el fichero `virtuals.conf`:

```
gedit /etc/proftpd/virtuals.conf
```

y le añadimos:

```
<VirtualHost 192.168.1.50>
   ServerAdmin       ftpmaster@empresa-tarea-daw04.local
   ServerName        "Servidor FTP para la tarea 4 de DAW (JLC)"
   AuthUserFile      /etc/passwd.usuarios.virtuales
   DefaultRoot       /var/ftp/todo-empresa-tarea-daw04
   RequireValidShell off
   AllowOverwrite    on
</VirtualHost>
```

Con lo que estaríamos indicándole que queremos crear un virtualhost o host virtual para la dirección `192.168.1.50`, al cual le indicamos un email para el administrador, un nombre

para el servidor, que los usuarios autorizados se encuentran en el fichero `/etc/passwd.usuarios.virtuales`, que el directorio raíz por defecto para este virtualhost será `/var/ftp/todo-empresa-tarea-daw04`, que no requiere usuarios con un Shell del sistema válido y que se permite la escritura en él.
Si quisiéramos utilizar otra IP podríamos forzarla utilizando la información ofrecida por `ifconfig` y fijándonos en el interface de entrada a nuestro equipo, y si por ejemplo es el `eth0`, podríamos hacer:

```
ifconfig eth0:1 192.168.100.250
```

con lo que obtendríamos otra conexión IP de número `192.168.100.250` que podríamos utilizar hasta que reiniciemos el sistema. Si la quisiéramos de forma definitiva, tendríamos que modificar el fichero `/etc/network/interfaces`

2.3. El cifrado TLS Explícito para asegurar la comunicación con empresa-tarea-daw04

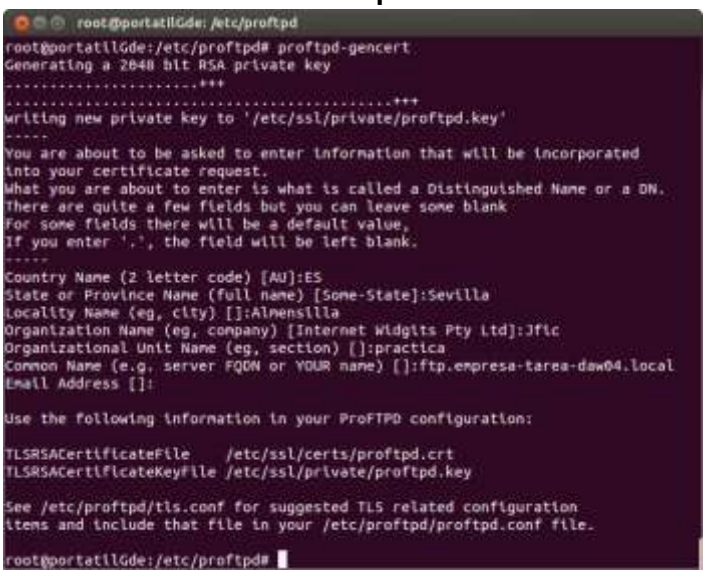
Para realizar un cifrado TLS hemos de descomentar la línea:

```
include /etc/proftpd/tls.conf
```

del fichero `proftpd.conf`, y a continuación creamos las claves públicas y privadas mediante:

```
proftpd-gencert
```

y tras contestar todas las preguntas nos creará los ficheros de certificados (*válidos durante un año*) a los que cambiaremos los permisos, y modificaremos su ubicación:




```
mv /etc/ssl/private/proftpd.key /etc/ssl/
chmod 0600 /etc/ssl/proftpd.key
chmod 0644 /etc/ssl/certs/proftpd.crt
```

Tras este proceso, editaremos el fichero `tls.conf`:

```
gedit /etc/proftpd/tls.conf
```

y descomentaremos las siguientes líneas:

```
<IfModule mod_tls.c>
   <global>
      TLSEngine on
      TLSLog /var/log/proftpd/tls.log
   </global>


   TLSProtocol SSLv23

   <global>
      TLSRSACertificateFile       /etc/ssl/certs/proftpd.crt
      TLSRSACertificateKeyFile    /etc/ssl/proftpd.key
      TLSOptions                  NoCertRequest EnableDialogs NoSessionReuseRequired
      TLSVerifyClient             off
      TLSRequired                 on
      TLSRenegotiate              required off
   </global>
</IfModule>
```

3. Instalar el cliente ftp gráfico FileZilla (filezilla)

Para la instalación de FileZilla procederemos de la siguiente forma:

3.1. como ya actualizamos los repositorios en los puntos 1.1 y 1.2, procedemos a buscar si tenemos el paquete filezilla, por lo que tecleamos:

```
apt-cache search filezilla
```

3.2. Como vemos que sí existe, lo instalamos:

```
apt-get install filezilla
```

3.3. ya podemos arrancarlo con solo teclear su nombre y darle a intro

```
filezilla
```

4. Configurar FileZilla con:

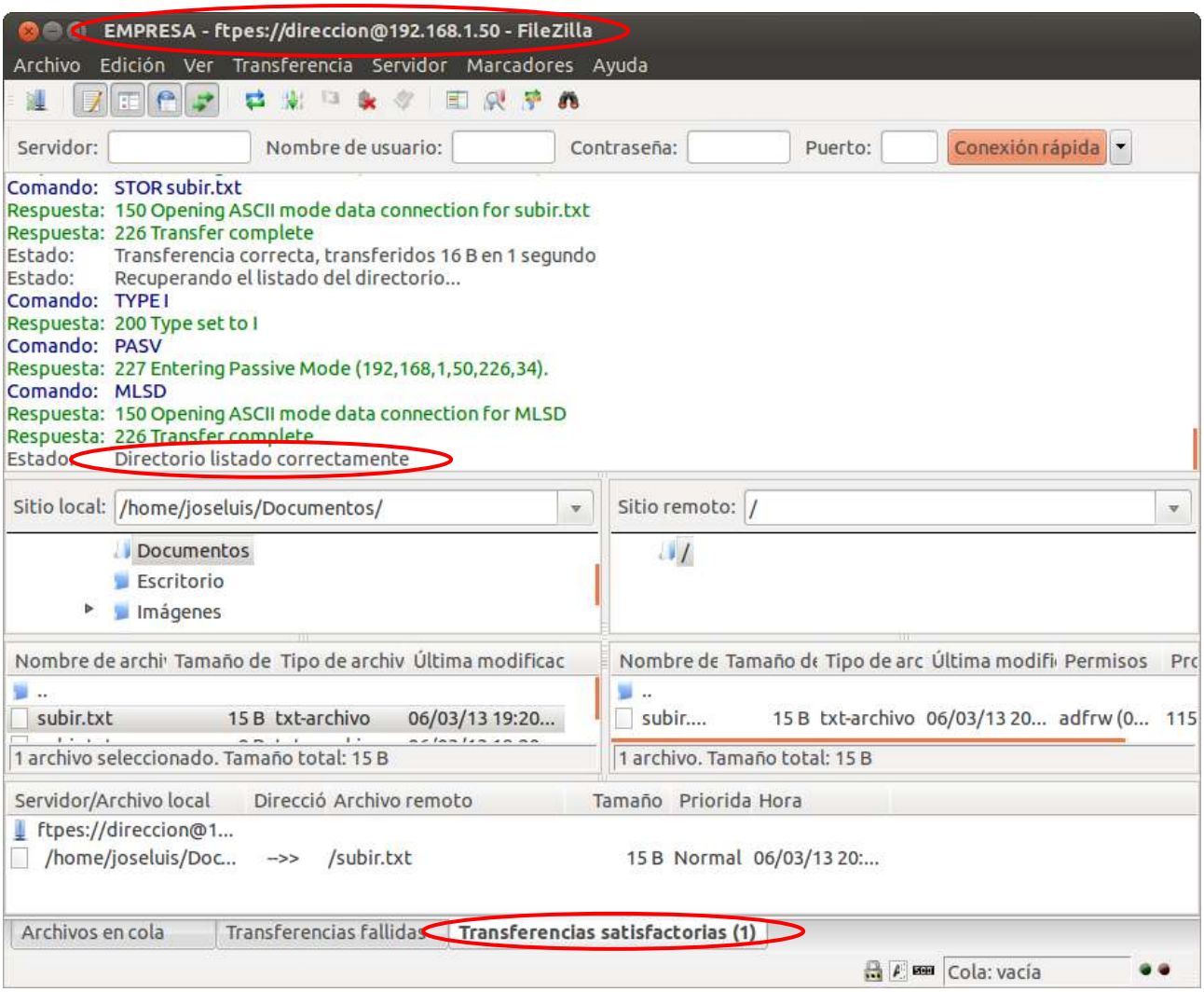
4.1. Una plantilla en el Gestor de Sitios de nombre EMPRESA que contenga la configuración pedida en el enunciado.

Pulsamos sobre `Archivo / Gestor de sitios` y, en la pantalla que aparece, pulsamos sobre el botón de `Nuevo sitio`, quedando a la espera que le indiquemos un nombre para el sitio que vamos a crear. Tecleamos `EMPRESA`, y ahora nos vamos al apartado del servidor donde teclearemos la dirección IP que coincidirá con el servidor que deseamos gestionar. En el puerto no ponemos nada, ya que por defecto se cogerá el 21 como puerto de comunicaciones, y es el que dejamos puesto en la cláusula `port` de proftpd. Utilizaremos FTP como protocolo de transferencia de archivos y cifrado explícito sobre TLS. En el usuario pondremos `direccion` e indicarle que nos pida la contraseña cada vez que nos intentemos conectar.

Al pulsar sobre `Aceptar` ya tendremos definida nuestra plantilla, que podremos utilizar cada vez que queramos pulsando sobre `Archivo / Gestor de sitios` y seleccionando la plantilla que deseemos utilizar. Cuando pulsamos sobre `Conectar` comenzará el proceso de conexión con arreglo a todos los parámetros indicados en los puntos anteriores.

Una vez que intentemos conectarnos nos pedirá la contraseña que pusimos en el punto **2.1**, pulsamos sobre `aceptar` y nos aparece el certificado creado en el punto **2.3** y cuando pulsemos sobre `Aceptar` ya estaremos preparados para poder enviar ficheros desde cualquier punto de nuestro equipo hacia el virtualhost creado para la carpeta `/var/ftp/todo-empresa-tarea-daw04`

4.2 Por último, para probar nuestro servidor vamos a transferir el fichero `subir.txt` desde nuestra carpeta hasta el virtualhost recién creado. Nos tenemos que fijar en los mensajes de filezilla para comprobar que todo ha sido realizado con éxito.



Los ficheros resultantes quedarían así:

proftpd.conf

```
#
# /etc/proftpd/proftpd.conf -- This is a basic ProFTPD configuration file.
# To really apply changes, reload proftpd after modifications, if
# it runs in daemon mode. It is not required in inetd/xinetd mode.
#

# Includes DSO modules
Include /etc/proftpd/modules.conf

# Set off to disable IPv6 support which is annoying on IPv4 only boxes.
# UseIPv6                 on
# If set on you can experience a longer connection delay in many cases.
IdentLookups              off

ServerName                "FTP para la tarea4 de DAW (JLC)"
ServerType                standalone
DeferWelcome              off

MultilineRFC2228          on
DefaultServer             on
ShowSymlinks              on
```

```
TimeoutNoTransfer     600
TimeoutStalled        600
TimeoutIdle           1200

DisplayLogin          welcome.msg
DisplayChdir          .message true
ListOptions           "-l"

DenyFilter            \*.*/

# Use this to jail all users in their homes
DefaultRoot           ~

# Users require a valid shell listed in /etc/shells to login.
# Use this directive to release that constrain.
RequireValidShell     off

# El puerto por defecto es el 21.
Port                  21

# In some cases you have to specify passive ports range to by-pass
# firewall limitations. Ephemeral ports can be used for that, but
# feel free to use a more narrow range.
# PassivePorts      49152 65534

# If your host was NATted, this option is useful in order to
# allow passive tranfers to work. You have to use your public
# address and opening the passive ports used on your firewall as well.
# MasqueradeAddress   1.2.3.4

# This is useful for masquerading address with dynamic IPs:
# refresh any configured MasqueradeAddress directives every 8 hours
<IfModule mod dynmasq.c>
# DynMasqRefresh      28800
</IfModule>

# To prevent DoS attacks, set the maximum number of child processes
# to 30.  If you need to allow more than 30 concurrent connections
# at once, simply increase this value.  Note that this ONLY works
# in standalone mode, in inetd mode you should use an inetd server
# that allows you to limit maximum number of processes per service
# (such as xinetd)
MaxInstances          30

# Set the user and group that the server normally runs at.
User                  proftpd
Group                 nogroup

# Umask 022 is a good standard umask to prevent new files and dirs
# (second parm) from being group and world writable.
Umask                 022  022
# Normally, we want files to be overwriteable.
AllowOverwrite        on

# Uncomment this if you are using NIS or LDAP via NSS to retrieve passwords:
# PersistentPasswd    off

# This is required to use both PAM-based authentication and local passwords
# AuthOrder           mod_auth_pam.c* mod_auth_unix.c

# Be warned: use of this directive impacts CPU average load!
# Uncomment this if you like to see progress and transfer rate with ftpwho
# in downloads. That is not needed for uploads rates.
#
# UseSendFile         off

TransferLog           /var/log/proftpd/xferlog
SystemLog             /var/log/proftpd/proftpd.log

# Logging onto /var/log/lastlog is enabled but set to off by default
#UseLastlog           on

# In order to keep log file dates consistent after chroot, use timezone info
# from /etc/localtime.  If this is not set, and proftpd is configured to
# chroot (e.g. DefaultRoot or <Anonymous>), it will use the non-daylight
# savings timezone regardless of whether DST is in effect.
#SetEnv TZ :/etc/localtime
```

```
<IfModule mod_quotatab.c>
   QuotaEngine off
</IfModule>

<IfModule mod_ratio.c>
   Ratios off
</IfModule>


# Delay engine reduces impact of the so-called Timing Attack described in
# http://www.securityfocus.com/bid/11430/discuss
# It is on by default.
<IfModule mod delay.c>
   DelayEngine on
</IfModule>

<IfModule mod_ctrls.c>
   ControlsEngine     off
   ControlsMaxClients         2
   ControlsLog        /var/log/proftpd/controls.log
   ControlsInterval   5
   ControlsSocket     /var/run/proftpd/proftpd.sock
</IfModule>

<IfModule mod ctrls admin.c>
   AdminControlsEngine        off
</IfModule>

#
# Alternative authentication frameworks
#
#Include /etc/proftpd/ldap.conf
#Include /etc/proftpd/sql.conf

#
# This is used for FTPS connections
#
Include /etc/proftpd/tls.conf

#
# Useful to keep VirtualHost/VirtualRoot directives separated
#
Include /etc/proftpd/virtuals.conf

# A basic anonymous configuration, no upload directories.

# <Anonymous ~ftp>
#   User          ftp
#   Group             nogroup
#   # We want clients to be able to login with "anonymous" as well as "ftp"
#   UserAlias         anonymous ftp
#   # Cosmetic changes, all files belongs to ftp user
#   DirFakeUser   on ftp
#   DirFakeGroup on ftp
#
#   RequireValidShell         off
#
#   # Limit the maximum number of anonymous logins
#   MaxClients        10
#
#   # We want 'welcome.msg' displayed at login, and '.message' displayed
#   # in each newly chdired directory.
#   DisplayLogin              welcome.msg
#   DisplayChdir      .message
#
#   # Limit WRITE everywhere in the anonymous chroot
#   <Directory *>
#     <Limit WRITE>
#       DenyAll
#     </Limit>
#   </Directory>
#
#   # Uncomment this if you're brave.
#   # <Directory incoming>
#   #   # Umask 022 is a good standard umask to prevent new files and dirs
#   #   # (second parm) from being group and world writable.
```

```
#   #   Umask                 022  022
#   #            <Limit READ WRITE>
#   #            DenyAll
#   #            </Limit>
#   #            <Limit STOR>
#   #            AllowAll
#   #            </Limit>
#   # </Directory>
# Include other custom configuration files
Include /etc/proftpd/conf.d/
```

tls.conf

```
#
# Proftpd sample configuration for FTPS connections.
#
# Note that FTPS impose some limitations in NAT traversing.
# See http://www.castaglia.org/proftpd/doc/contrib/ProFTPD-mini-HOWTO-TLS.html
# for more information.
#

<IfModule mod_tls.c>
   <global>
      TLSEngine   on
      TLSLog      /var/log/proftpd/tls.log
   </global>
   TLSProtocol    SSLv23
   <global>
#
# Server SSL certificate. You can generate a self-signed certificate using
# a command like:
#
# openssl req -x509 -newkey rsa:1024 \
#          -keyout /etc/ssl/private/proftpd.key -out /etc/ssl/certs/proftpd.crt \
#          -nodes -days 365
#
# The proftpd.key file must be readable by root only. The other file can be
# readable by anyone.
#
# chmod 0600 /etc/ssl/private/proftpd.key
# chmod 0640 /etc/ssl/private/proftpd.key
#
      TLSRSACertificateFile          /etc/ssl/certs/proftpd.crt
      TLSRSACertificateKeyFile       /etc/ssl/proftpd.key
#
# CA the server trusts...
#TLSCACertificateFile                /etc/ssl/certs/CA.pem
# ...or avoid CA cert and be verbose
#     TLSOptions  NoCertRequest
# ... or the same with relaxed session use for some clients (e.g. FireFtp)
      TLSOptions      NoCertRequest EnableDiags NoSessionReuseRequired
#
#
# Per default drop connection if client tries to start a renegotiate
# This is a fix for CVE-2009-3555 but could break some clients.
#
#TLSOptions                                          AllowClientRenegotiations
#
# Authenticate clients that want to use FTP over TLS?
#
      TLSVerifyClient        off
#
# Are clients required to use FTP over TLS when talking to this server?
#
      TLSRequired            on
#
# Allow SSL/TLS renegotiations when the client requests them, but
# do not force the renegotations.  Some clients do not support
# SSL/TLS renegotiations; when mod_tls forces a renegotiation, these
# clients will close the data connection, or there will be a timeout
# on an idle data connection.
#
      TLSRenegotiate  required off
   </global>
</IfModule>
```

virtuals.conf

```
# Proftpd sample configuration for Virtual Hosts and Virtual Roots.
#
# Note that FTP protocol requires IP based virtual host, not name based.
#

#
# A generic sample virtual host.
#
<VirtualHost 192.168.1.50>
   ServerAdmin             ftpmaster@empresa-tarea-daw04.local
   ServerName              "Servidor FTP para la tarea 4 de DAW (JLC)"
   TransferLog             /var/log/proftpd/xfer/ftp.server.com
   MaxLoginAttempts        3
   RequireValidShell       off
   AuthUserFile              /etc/passwd.usuarios.virtuales
   DefaultRoot             /var/ftp/todo-empresa-tarea-daw04
   AllowOverwrite          on
</VirtualHost>

#
# The vroot module is not required, but can be useful for shared
# directories.
#
<IfModule mod vroot.c>
#VRootEngine on

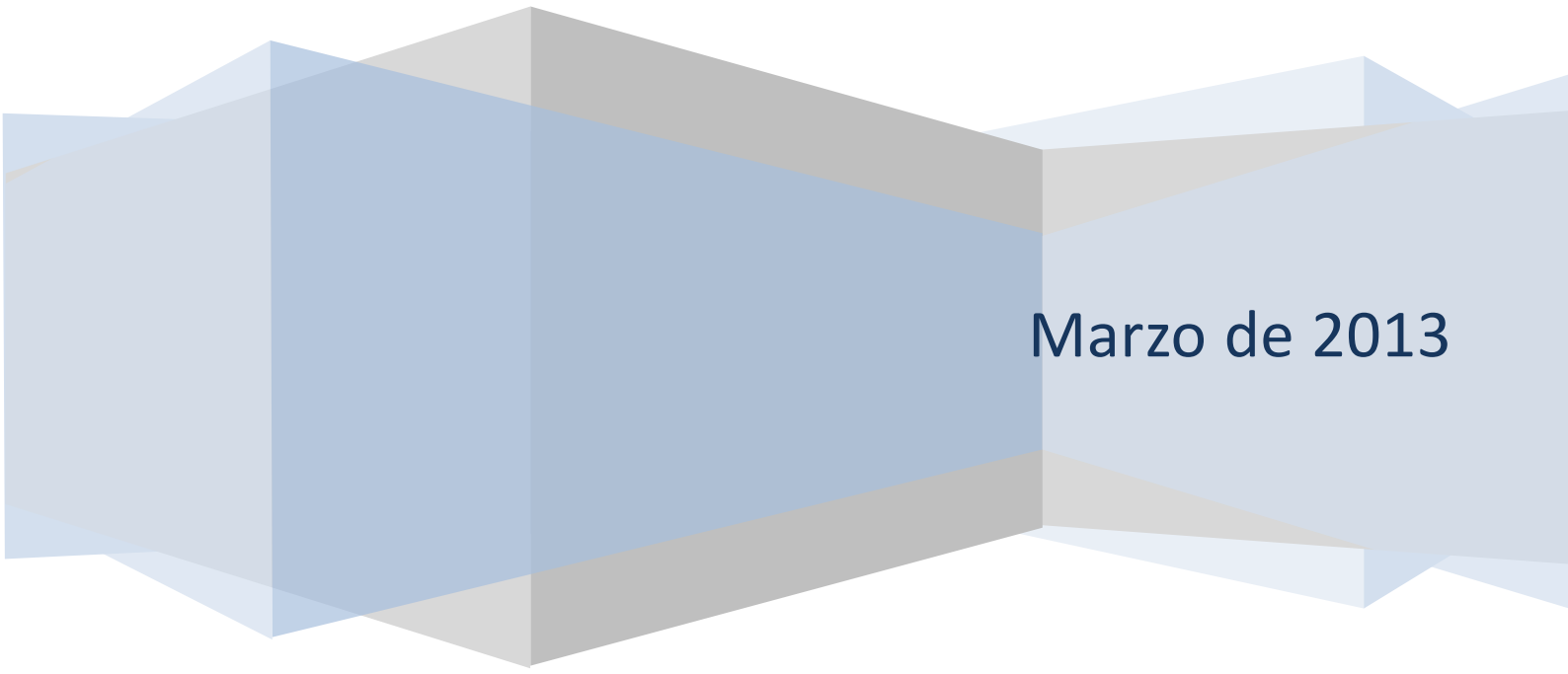
#DefaultRoot ~
#VRootAlias upload /var/ftp/upload
#
#<VirtualHost a.b.c.d>
#VRootEngine on
#VRootServerRoot /etc/ftpd/a.b.c.d/
#VRootOptions allowSymlinks
#DefaultRoot ~
#</VirtualHost>
#
</IfModule>
```

Módulo Superior de Diseño de Aplicaciones Web

Despliegue de Aplicaciones Web

Tarea 5

José Luis Comesaña Cabeza



Marzo de 2013

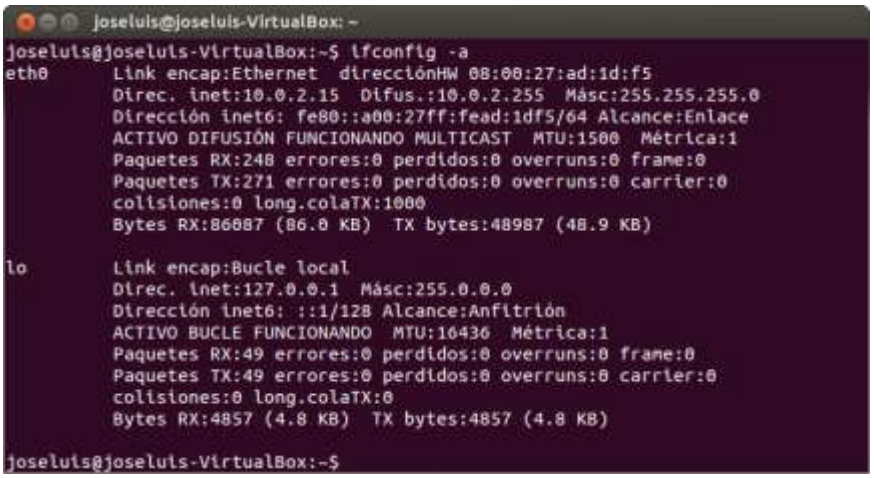
Para la realización de la tarea, donde ponga **xxxxx** debes cambiarlo por tu inicial de nombre seguido por tu primer apellido. Así, en caso de llamarte Antonio López, pues deberás usar alopez.

Para una empresa que opera con su negocio a través de Internet se pide lo siguiente.

1. **Configura tu servidor con la IP 192.168.200.200 (o si usas otra indica cuál).**

   Instala y configura el servidor de nombres bind para que: Administre el dominio dominioempresa-xxxxx-daw05.local.

Para conocer nuestra IP local probamos con el comando `ifconfig -a` lo cual nos da esta salida:



Nos indica que la IP de nuestro servidor es la `10.0.2.15`, y será la que usaremos para la resolución de los ejercicios de esta tarea.
Podríamos obligarle a utilizar otra como la `192.168.200.200` modificando el fichero `/etc/network/interfaces` introduciéndole el contenido:

```
auto lo
iface lo inet loopback auto eth0
iface eth0 inet static
address 192.168.200.200
netmask 255.255.255.0
network 192.168.200.0
broadcast 192.168.200.255
gateway 192.168.200.1
```

Pero no es necesario para los ejercicios que realizamos a continuación, por lo que mantendremos la ofrecida por `ifconfig`.

Primero actualizamos los repositorios del sistema operativo con:

```
apt-get update
```

A continuación actualizamos el sistema operativo:

```
apt-get upgrade
```

Instalamos los paquetes necesarios para el funcionamiento de BIND:

```
apt-get install bind9 bind9utils
```

Verificamos que está instalado y funcionando correctamente:

```
service bind9 status
```

que nos indica que bind9 se está ejecutando

Hacemos una copia de seguridad del archivo que vamos a modificar:

```
cp /etc/bind/named.conf.local{,.original}
```

(esta orden es igual a `cp /etc/bind/named.conf.local /etc/bind/named.conf.original`)

Editamos el fichero named.conf.local tecleando:

```
gedit /etc/bind/named.conf.local
```

Y en su interior tecleamos:

```
zone "dominioempresa-jlcomesana-daw05.local"{
   type master;
   file "/var/lib/bind/master/db.dominioempresa-jlcomesana-daw05.local";
};
```

Como el camino indicado no existe, lo creamos con:

```
mkdir /var/lib/bind/master
```

Comprobamos la sintaxis del archivo de configuración tecleando:

```
named-checkconf /etc/bind/named.conf.local
```

Como no aparece nada significa que es correcto sintácticamente, de lo contrario nos indicaría el error producido.

Crea el registro tipo host tienda (tienda.dominioempresa-xxxxx-daw05.local) que apunte al servidor 192.168.200.200

Ahora, para completar el punto 1 de la tarea y realizar el punto 2, editamos el fichero que indicamos en el apartado `file` del `named.conf.local`, por lo que tecleamos:

```
gedit /var/lib/bind/master/db.dominioempresa-jlcomesana-daw05.local
```

y le incluimos:

```
$ORIGIN dominioempresa-jlcomesana-daw05.local.
$TTL 86400 ; 1 dia
@       IN      SOA     tienda        postmaster (
        1    ; serie
        6H   ; refresco (6 horas)
        1H   ; reintentos (1 hora)
        2W   ; expira (2 semanas)
        3H   ; mínimo (3 horas)
)
         NS     tienda
tienda   IN     A       10.0.2.15
www      IN     A       10.0.2.15
mail     IN     A       10.0.2.16
```

Comprobamos la zona que acabamos de crear (`dominioempresa-jlcomesana-daw05.local`):

```
named-chekzone     dominioempresa-jlcomesana-daw05.local     /var/cache/bind/db.dominioempresa-jlcomesana-daw05.local
```

Como nos sale todo OK vamos a crear el archivo `/var/cache/bind/db.10.0.2` para la zona inversa:

```
gedit /var/lib/bind/master/db.10.0.2
```

y le introducimos las siguientes órdenes:

```
$ORIGIN 2.0.10.in-addr.arpa.
$TTL 86400     ; 1 dia
@       IN      SOA     tienda        postmaster (
        1      ; serie
        6H     ; refresco (6 horas)
        1H     ; reintentos (1 hora)
        2W     ; expire (2 semanas)
        3H     ; mínimo (3 horas)
)
       NS      tienda.dominioempresa-jlcomesana-daw05.local.
15     PTR     tienda.dominioempresa-jlcomesana-daw05.local.
```

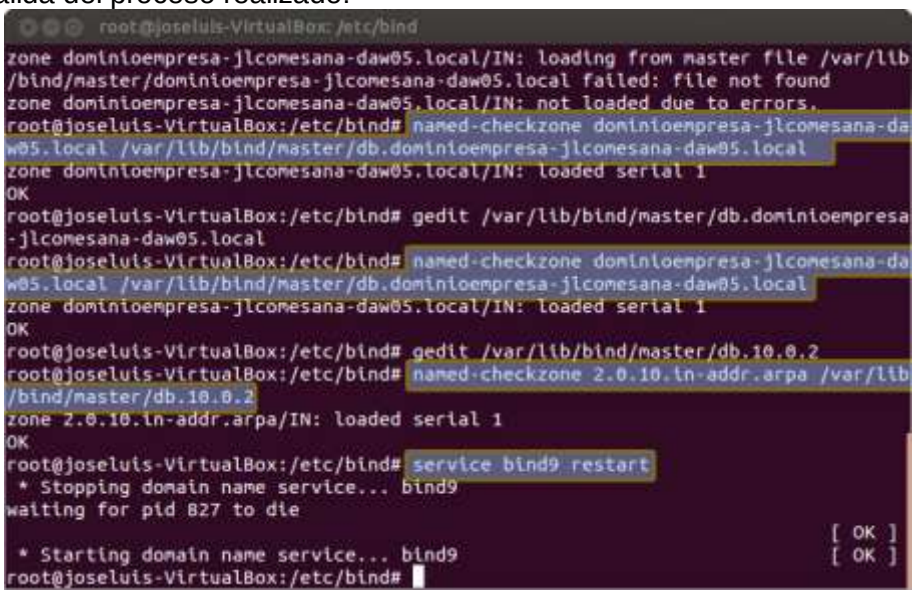
Comprobamos la zona inversa recién creada:

```
named-checkzone 2.0.10.in-addr.arpa /var/cache/bind/db.10.0.2
```

Reiniciamos el servicio:

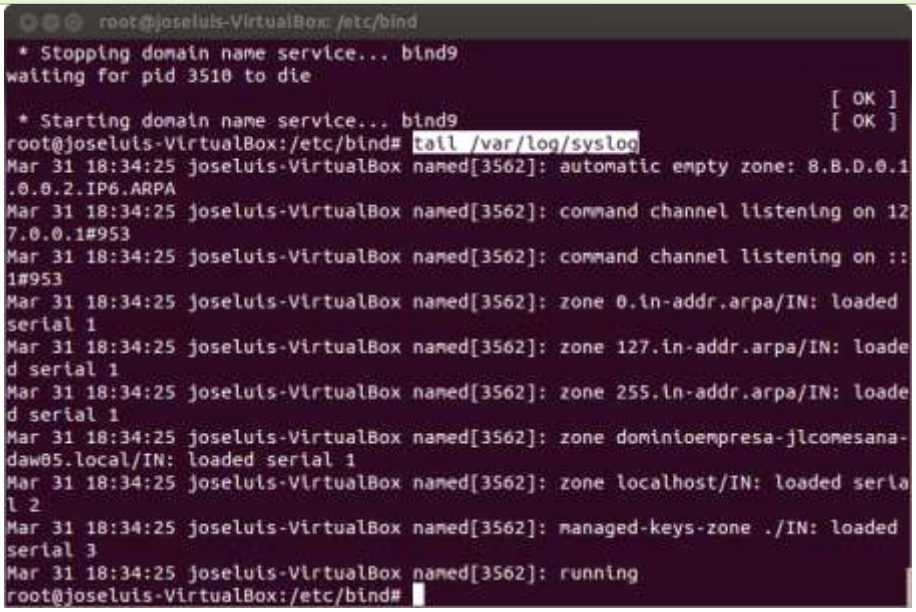
```
service bind9 restart
```

Esta es la salida del proceso realizado:



Revisamos el log para comprobar que todo ha ido bien:

```
tail /var/log/syslog
```



Editamos el fichero `/etc/resolv.conf` para que nuestro servidor resuelva las peticiones DNS:

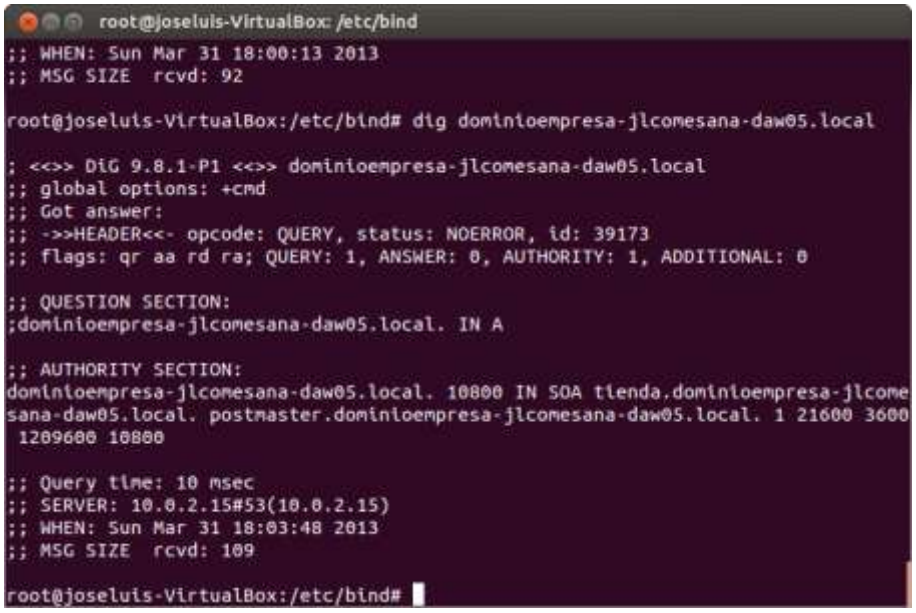
```
gedit /etc/resolv.conf
```

Cambiamos el primer servidor DNS por la IP del nuestro:

```
nameserver  10.0.2.15
nameserver  8.8.8.8
```

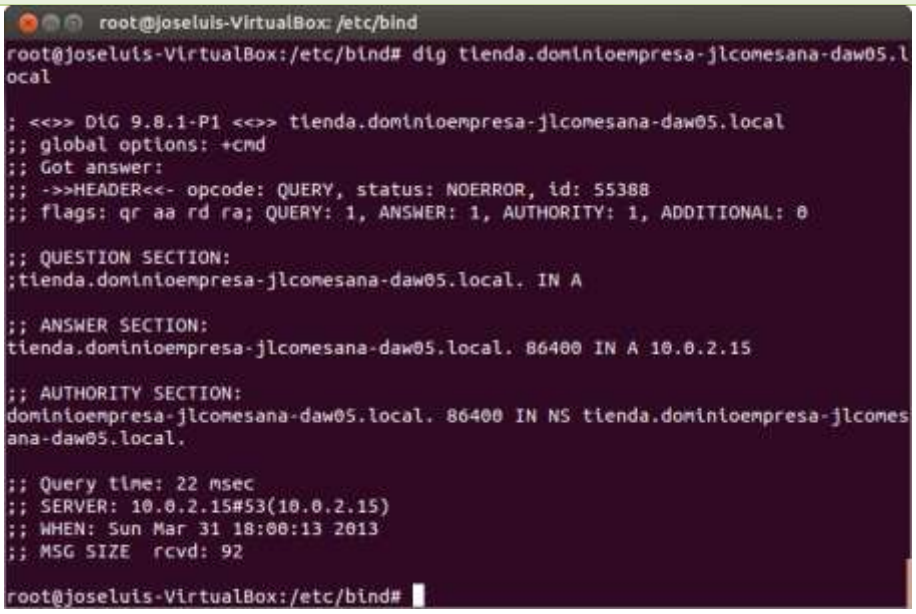
Probamos nuestro servidor de nombres:

```
dig dominioempresa-jlcomesana-daw05.local
```



Probamos también tienda:

```
dig tienda.dominioempresa-jlcomesana-daw05.local
```



Como vemos que en `status` nos aparece `NOERROR`, podemos decir que tendremos concluido el punto 1 de la tarea

2. Configurar un servidor OpenLDAP con:

1º. El nombre de dominioempresa-xxxxx-daw05.local.

Antes de instalar LDAP se han de actualizar los repositorios y el sistema, pero como ya lo hicimos al comienzo del punto 1, no tendremos que realizarlo de nuevo.

Instalamos los paquetes necesarios para el funcionamiento de OpenLDAP, tecleando:

```
apt-get install slapd ldap-utils
```

Y cuando nos solicite la contraseña tendremos que recordarla ya que será la que nos pida para reiniciar los servicios LDAP.

Ahora comprobamos que el servicio está activo, por defecto en el puerto TCP 389, para lo que tecleamos:

```
netstat -natp | grep 389
```

Obteniendo la salida:

```
tcp   0  0    0.0.0.0:389    0.0.0.0:*      ESCUCHAR       5668/slapd
tcp6  0  0    :::389         :::*           ESCUCHAR       5668/slapd
```

Ahora configuramos LDAP utilizando su asistente:

```
dpkg-reconfigure slapd
```

Y nos preguntará:

a. **Deseas omitir la configuración** - Lógicamente contestamos que ***NO***
b. **Nombre del dominio**: *dominioempresa-jlcomesana-daw05.local*
c. **Nombre de la organización**: *dominioempresa-jlcomesana-daw05.local*
d. **Contraseña administrador LDAP**: introducimos la que queramos y la volverá a solicitar como seguridad
e. **Motor de la base de datos a utilizar**: *HDB*
f. **Eliminar BD cuando quitemos slapd**: ***SI***, para evitar confusiones con otras BD
g. **Desea mover la base de datos antigua**: ***SI***
h. **Utilizar la versión 2 de LDAP**: ***NO***, ya que está prácticamente obsoleta

2º. Elabora un fichero `estructura.ldif` para crear una *unidad organizacional* "`tienda`".

```
cd /etc/ldap
gedit estructura.ldif
```

En su interior tecleamos:

```
dn: ou=tienda, dc=dominioempresa-jlcomesana-daw05,dc=local
ou: tienda
objectclass: organizationalUnit
```

3º. Elabora un fichero `grupo.ldif` para crear una *unidad organizacional* "`grupo`".

```
gedit grupo.ldif
```

En su interior tecleamos:

```
dn: ou=grupo,dc=dominioempresa-jlcomesana-daw05,dc=local
ou: grupo
objectclass: organizationalUnit
```

4º. Elabora un fichero `usuarios.ldif` para crear un *grupo* `users`.

```
gedit usuarios.ldif
```

En su interior tecleamos:

```
dn: cn=users,ou=grupo,dc=dominioempresa-jlcomesana-daw05,dc=local
objectclass: posixGroup
objectclass: top
cn: users
userPassword: {crypt}*
gidNumber: 112
```

5º. Agregar los contenidos de los ldif.

```
ldapadd -x -W -D "cd=admin,dc=dominioempresa-jlcomesana-daw05,dc=local" -f estructura.ldif
ldapadd -x -W -D "cd=admin,dc=dominioempresa-jlcomesana-daw05,dc=local" -f grupo.ldif
ldapadd -x -W -D "cd=admin,dc=dominioempresa-jlcomesana-daw05,dc=local" -f usuarios.ldif
```

6º. Revisar el resultado de la inserción.

```
ldapsearch -x -b "dc=dominioempresa-jlcomesana-daw05,dc=local"
```

Y obtenemos la siguiente salida:

```
# extended LDIF
#
# LDAPv3
# base <dc=dominioempresa-jlcomesana-daw05,dc=local> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# dominioempresa-jlcomesana-daw05.local
dn: dc=dominioempresa-jlcomesana-daw05,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: dominioempresa-jlcomesana-daw05.local
dc: dominioempresa-jlcomesana-daw05

# admin, dominioempresa-jlcomesana-daw05.local
dn: cn=admin,dc=dominioempresa-jlcomesana-daw05,dc=local
```

```
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

# tienda, dominioempresa-jlcomesana-daw05.local
dn: ou=tienda,dc=dominioempresa-jlcomesana-daw05,dc=local
ou: tienda
objectClass: organizationalUnit

# grupo, dominioempresa-jlcomesana-daw05.local
dn: ou=grupo,dc=dominioempresa-jlcomesana-daw05,dc=local
ou: grupo
objectClass: organizationalUnit

# users, grupo, dominioempresa-jlcomesana-daw05.local
dn: cn=users,ou=grupo,dc=dominioempresa-jlcomesana-daw05,dc=local
objectClass: posixGroup
objectClass: top
cn: users
gidNumber: 112

# search result
search: 2
result: 0 Success

# numResponses: 6
# numEntries: 5
```

7º. Elabora un fichero `miusuario.ldif` para crear un *usuario* con tu nombre.

```
gedit miusuario.ldif
```

Tecleando en su interior:

```
dn: uid=joseluis,ou=tienda,dc=dominioempresa-jlcomesana-daw05,dc=local
uid: joseluis
cn: José Luis
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
userPassword: {crypt}3kYWXeItEYp1.
shadowLastChange: 14001
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 1012
gidNumber: 112
homeDirectory: /home/joseluis
```

el `password` lo hemos averiguado mediante la orden `slappasswd -h {CRYPT}`
la cual cuando pulsamos *Intro* nos pregunta una contraseña por duplicado para confirmarla y cuando terminamos nos indica cual es su contraseña cifrada, por ejemplo:

```
slappaswd -h {CRYPT}
New password: hola
Re-enter new password: hola
{CRYPT}XM0QyPNgth5WE
```

8º. Agrega el nuevo usuario al nuevo árbol de LDAP.

```
ldapadd -x -W -D "cn=admin,dc=dominioempresa-jlcomesana-daw05,dc=local" -f miusuario.ldif
```

Tras introducir la contraseña de LDAP que nos solicita, y si todo es correcto, nos pone un mensaje indicando que ha añadido una nueva entrada

9º. Revisa el resultado de la inserción.

```
ldapsearch -x -b "dc=dominioempresa-jlcomesana-daw05,dc=local"
```

Y el mensaje nos indica que todo se ha realizado de forma correcta

```
# extended LDIF
#
# LDAPv3
# base <dc=dominioempresa-jlcomesana-daw05,dc=local> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
```

```
# dominioempresa-jlcomesana-daw05.local
dn: dc=dominioempresa-jlcomesana-daw05,dc=local
objectClass: top
objectClass: dcObject
objectClass: organization
o: dominioempresa-jlcomesana-daw05.local
dc: dominioempresa-jlcomesana-daw05

# admin, dominioempresa-jlcomesana-daw05.local
dn: cn=admin,dc=dominioempresa-jlcomesana-daw05,dc=local
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator

# tienda, dominioempresa-jlcomesana-daw05.local
dn: ou=tienda,dc=dominioempresa-jlcomesana-daw05,dc=local
ou: tienda
objectClass: organizationalUnit

# grupo, dominioempresa-jlcomesana-daw05.local
dn: ou=grupo,dc=dominioempresa-jlcomesana-daw05,dc=local
ou: grupo
objectClass: organizationalUnit

# users, grupo, dominioempresa-jlcomesana-daw05.local
dn: cn=users,ou=grupo,dc=dominioempresa-jlcomesana-daw05,dc=local
objectClass: posixGroup
objectClass: top
cn: users
gidNumber: 112

# joseluis, tienda, dominioempresa-jlcomesana-daw05.local
dn: uid=joseluis,ou=tienda,dc=dominioempresa-jlcomesana-daw05,dc=local
uid: joseluis
cn:: Sm9zw6kgTHVpcw==
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
shadowMax: 99999
shadowWarning: 7
loginShell: /bin/bash
uidNumber: 1012
gidNumber: 112
homeDirectory: /home/joseluis

# search result
search: 2
result: 0 Success

# numResponses: 7
# numEntries: 6
```

Tarea para DAW06.

1. **Indica cada uno de los pasos que deberías de dar para proceder a la instalación de phpDocumentor, suponiendo que vas a partir de una máquina en la que tienes instalado la distribución Ubuntu y en la que ya están instalados y correctamente configurados apache y php.**

   Primero hermos de instalar pear, para lo que tecleamos:

   ```
   apt-get install php-pear
   ```

   Ahora configuramos su directorio de trabajo en /var/www:

   ```
   pear config-set data_dir /var/www
   ```

   Pasamos a instalar phpDocumentor con todas sus dependencias:

   ```
   pear install --alldeps PhpDocumentor
   ```

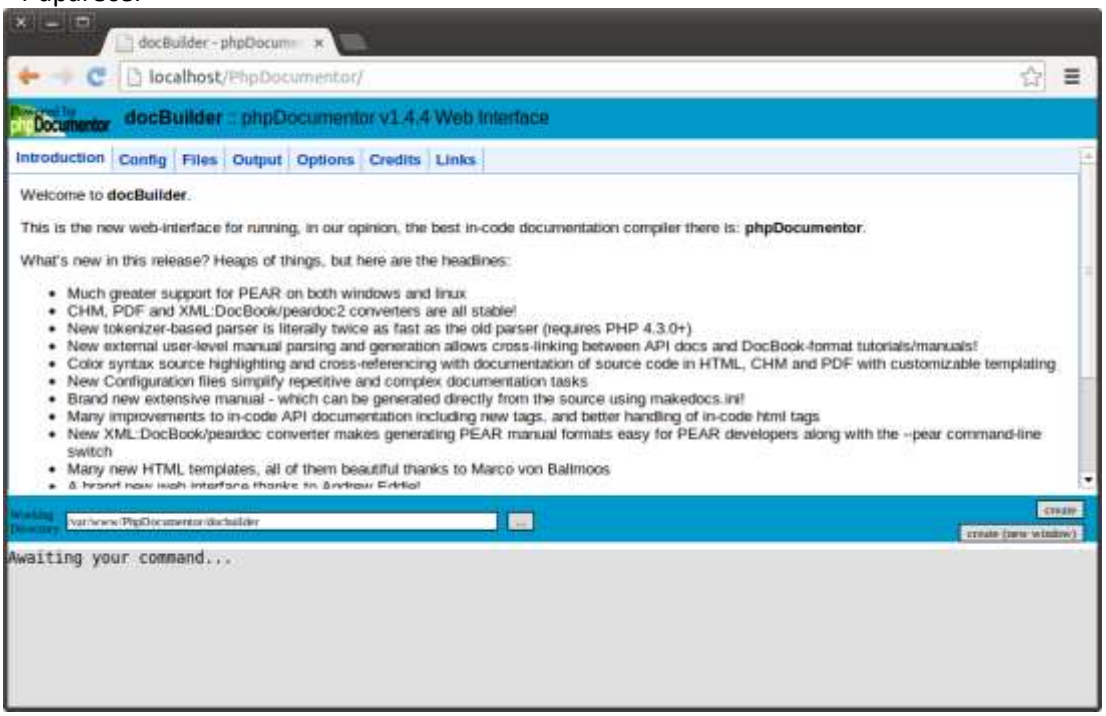
   Ya sólo nos queda crear un directorio de salida para phpDocumentor y cambiar su propietario a www-data:

   ```
   mkdir /var/www/docs
   chown www-data /var/www/docs
   ```

   Para comprobar que todo ha salido de forma correcta, dentro de un navegador tecleamos:

   ```
   http://localhost/PhpDocumentor
   ```

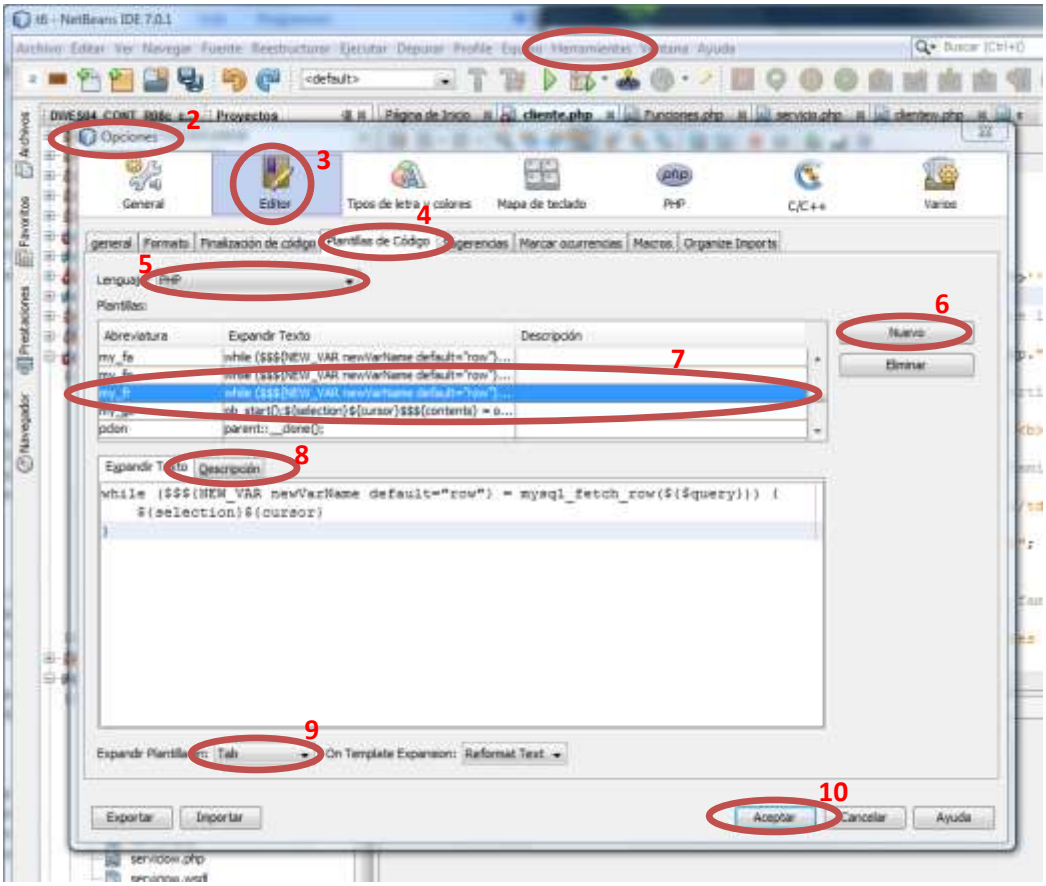
   Y aparece:

2. **Explica en qué consisten las plantillas de código en el caso de Javadoc y cada uno de sus componentes.**

   Con las plantillas Javadoc conseguimos documentar una aplicación, sus clases y métodos, siendo de mucha utilidad para una posterior actualización o refinamiento de la misma. También conseguimos que cuando estemos llamando a un método o clase determinado, se nos muestre una sugerencia para el código, la cual podemos coger con tan solo pulsar Ctrl+Intro o cualquier otra combinación de teclas que definamos.

   Para poder crear nuestras propias plantillas en NetBeans tendremos que seleccionar la opción **Herramientas - Opciones - Editor - Plantillas de Código** y en el apartado de lenguaje seleccionamos aquel al que deseamos aplicarle la nueva plantilla que creemos (*según el orden que muestro en la imagen de la página siguiente*).



Una plantilla se compone de nombre, descripción, contexto en función del lenguaje y un patrón o código de la plantilla. Este último puede estar compuesto de texto fijo o una serie de variables, como pueden ser:

- ✓ `${cursor}`: posición en la que se establecerá el cursor de texto tras desplegar el código de la plantilla.
- ✓ `${enclosing type}`: tipo de la clase en la que nos encontramos
- ✓ `${enclosing_method}`: nombre del método en el que nos encontramos
- ✓ `${year}`: año en curso
- ✓ `${time}`: hora en curso

3. Dispones de una máquina que cuenta con el sistema operativo Ubuntu recientemente actualizado, en la que está el entorno de red configurado y, además, dispones de conexión a Internet y estás trabajando con la cuenta del usuario root. Indica cada uno de los pasos y comandos implicados en ellos para conseguir hacer lo siguiente:

- ✓ **Suponiendo que el sistema ya tiene instalado las siguientes librerías de las que Git depende: `curl`, `zlib`, `openssl`, `expat`, y `libiconv`, pasos para realizar la compilación e instalación de Git considerando que ya disponemos del paquete `git-1.7.9.5.tar.bz2`**

```
tar -xvfz git-1.7.9.5.tar.bz2
cd git-1.7.9.5
apt-get build-dep git-core
apt-get install libssl-dev
make prefix=/usr/local all doc
make prefix=/usr/local install install-doc
```

- ✓ **Cómo obtener Git a través del propio Git para futuras actualizaciones, de manera que descargaría automáticamente el código fuente desde su repositorio.**

  Una vez que tengamos instalado git, teclearíamos en el terminal:

```
git clone git://git.kernel.org/pub/scm/git/git.git
```

- ✓ **Comprobar la versión que se ha instalado de Git.**

```
git --version
```

- ✓ **Establecer el nombre de usuario y dirección de correo electrónico en la configuración de Git.**

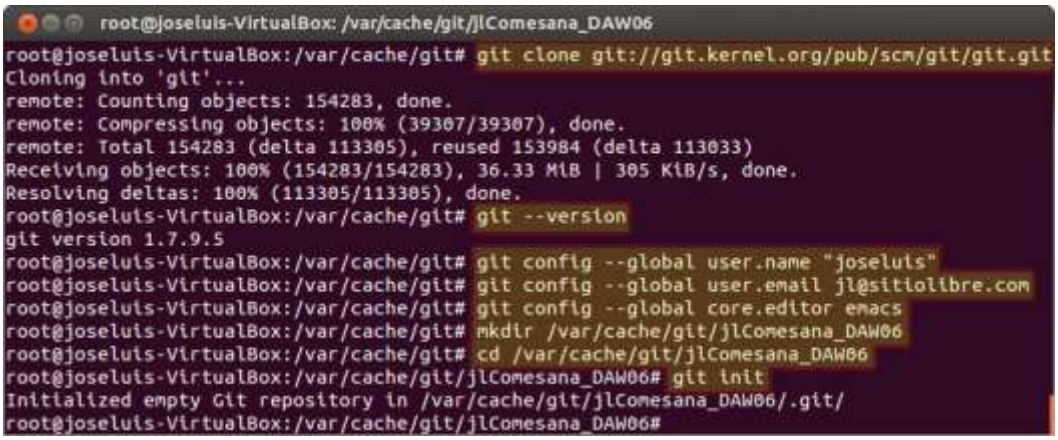
```
git config --global user.name "joseluis"
git config --global user.email jl@sitiolibre.com
```

- ✓ **Cambiar el editor de texto que trae por defecto Git al editor emacs.**

```
git config --global core.editor emacs
```

- ✓ **Dentro de la carpeta `/var/cache/git/` crear una carpeta para un nuevo proyecto denominado `tunombre DAW06` e iniciar un repositorio del nuevo proyecto, donde `tunombre` debes sustituirlo por la inicial de tu nombre, seguido de tu primer apellido.**

```
mkdir /var/cache/git/jlComesana_DAW06
cd /var/cache/git/jlComesana DAW06
git init
```



Módulo: Despliegue de Aplicaciones Web (**DAW**)
I.E.S.: Aguadulce **C.F.G.S. : Desarrollo de Aplicaciones Web**
Fecha: Martes 5 de febrero de 2013 **Hora de Comienzo:** 18:30 horas
Examen: Teórico y Práctico. **Duración:** 3 horas
Nombre: **D.N.I.:**
Centro en el que se realiza el examen:

INSTRUCCIONES:
La puntuación total del examen será de 10 puntos. Parte teórica: 6 puntos. Parte Práctica: 4 puntos.
Cada pregunta del examen teórico puntúa 0,2 puntos. Cada dos preguntas mal quita una bien.

CUESTIONES TEÓRICAS:

1. **Responde: La arquitectura Web consiste en un modelo de tres capas que son:**
   La arquitectura web es un modelo compuesto de tres capas:
   - Capa de Base de Datos, donde estaría toda la documentación de la información que se pretende administrar mediante el servicio web y emplearía una plataforma del tipo MySQL, PostgreSQL, etc.
   - En una segunda capa estarían los servidores de aplicaciones web, ejecutando aplicaciones de tipo Apache, Tomcat, Resin, etc.
   - En una tercera capa estarían los clientes del servicio web al que accederían mediante un navegador web como Firefox, Internet Explorer, Opera, etc.

2. **Si consideramos los componentes de la arquitectura web, podemos afirmar que Tomcat, está encuadrado en:**
   a. Base de datos.
   b. **Servidor de aplicaciones Web**.
   c. Clientes de servicio Web.
   d. Sistema operativo.

3. **Para probar el servlet, una vez arrancado el servidor Tomcat, abrimos un navegador web, en el cual escribiríamos una URL con el siguiente formato:**
   a. http://{address}:{port}/{servletName}
   b. http://{port}:{address}/{servletName}
   c. http://{address}:{servletName}/{port}
   d. http://{servletName}:{port}/{address}

4. Un descriptor de despliegue:
   a. Es un único archivo.
   b. Tiene formato XML.
   c. Describe las características de despliegue de una aplicación web.
   d. Todas son correctas.

5. Sobre los archivos WAR:
   a. No es posible generarlos desde entornos IDE.
   b. **Permiten empaquetar, en una sola unidad, aplicaciones web de Java completas**.
   c. Es una aplicación web formada únicamente por archivos .html
   d. Es un archivo que engloba el protocolo de comunicación de las aplicaciones web generadas con Java.

6. Los middlewares actúan como elementos conectores encargados de distribuir la aplicación web...
   a. De forma horizontal mediante un balanceador software.
   b. De forma vertical mediante un balanceador hardware ftp.
   c. De forma horizontal mediante un balanceador hardware http.
   d. De forma vertical distribuyendo las capas de la aplicación en diferentes máquinas.

7. Para modificar el servidor virtual predeterminado, editamos el archivo
   a. /etc/apache2/sites-available/default
   b. /var/apache/web
   c. /usr/local/apache
   d. /usr/apache/virtual

8. Apache por defecto busca los ficheros en
   a. /usr/local/tomcat/webapps/ROOT
   b. /var/apache/web
   c. /usr/local/ant
   d. /var/www

9. Los dispositivos que, respondiendo únicamente a algoritmos de reparto de carga (Round Robin, LRU, etc.), redireccionan una petición http del usuario a la máquina que, según dicho algoritmo, convenga que se haga cargo de la petición son:
   a. Balanceadores software
   b. Balanceadores http
   c. Balanceadores hardware
   d. Balanceadores localhosters ftp

10. Acerca de Apache podemos decir que:
    a. Es un servidor web poco extendido en el mercado.
    b. No es gratuito, aunque es bastante económico.
    c. Está disponible para GNU/Linux únicamente.
    d. Está estructurado en módulos.

11. Por defecto, Tomcat va a estar escuchando en el puerto:
    a. 8081
    b. 8080
    c. 8088
    d. 127.1

12. Los comandos a2dismod y a2enmod
    a. Permiten siempre y cuando el paquete gnome-modularity esté instalado, habilitar y deshabilitar módulos.
    b. Permiten siempre y cuando el paquete gnome-bobmodular esté instalado, deshabilitar y habilitar módulos.
    c. Son alias de los comandos a2dissite y a2ensite respectivamente, utilizados para desactivar y activar.
    d. **Permiten deshabilitar y habilitar módulos, respectivamente**.

13. **Completa los huecos de las siguientes frases con las palabras adecuadas.** Las aplicaciones web emplean páginas dinámicas, éstas se ejecutan en un servidor web y se muestran en el **navegador** de un equipo cliente que es el que ha realizado previamente la solicitud. Cuando una página web llega al navegador, es posible que también incluya algún programa o fragmento de código que se deba ejecutar. Ese código, normalmente en lenguaje JavaScript, lo ejecutará el propio navegador**.**

14. En un host virtual, encontramos que se atenderán peticiones por cualquier interfaz de red y a través del puerto 80 con:
    a. <VirtualHost *:??>
    b. **<VirtualHost *:80>**
    c. <VirtualHost 192.168.200.0:80>
    d. <VirtualHost *:*>

15. **rotatelog** es un programa proporcionado por Apache que permite:
    a. Rotar logs por tamaño de los mismos.
    b. Rotar logs.
    c. Rotar logs por intervalo de tiempo siempre y cuando en el intervalo de tiempo definido existan nuevos datos.
    d. Todas las anteriores con correctas.

16. Si queremos acceder a una aplicación, llamada **laweb22**, que se está ejecutando en un servidor Tomcat, desde el propio equipo, tendríamos que teclear en el navegador:
    a. http://localhost:8080
    b. http://localhost:80/laweb22
    c. http://localhost:8080/laweb22
    d. http://localhost:8080/tomcat
17. Si al acceder a una página web no es necesaria la intervención de código en el lado del servidor o en el lado del cliente, entonces entenderemos que la página es
    a. Estable.
    b. Dinámica.
    c. Movible.
    d. Xml.

18. Los ataques que suele recibir una aplicación web se pueden centrar en varios objetivos, señala cuáles de los siguientes podrían ser:
    a. Ataques a la computadora del usuario (cliente).
    b. Ataques al servidor.
    c. Ataques al flujo de información entre cliente y servidor.
    d. Todos los anteriores son correctos.

19. En la estructura de directorios que forman una aplicación web, el directorio que contiene el fichero web.xml es:
    a. WEB-INF
    b. raíz
    c. lib
    d. classes

20. El método quizás más sencillo para desplegar una aplicación que se utiliza durante la etapa de desarrollo, es copiar la carpeta correspondiente a la aplicación en la carpeta $CATALINA_HOME/ teniendo en cuenta que la variable $CATALINA_HOME es la ruta de los scripts que emplea Tomcat.
    a. web.xml
    b. index
    c. **webapps**
    d. classes

21. Tomcat es:
    a. Un navegador web.
    b. **Un servidor de aplicaciones**.
    c. Un servlet.
    d. Un sistema operativo.

22. Apache-Ant es una herramienta que se usa en programación para la realización de tareas mecánicas y repetitivas, posee la ventaja de no depender de los comandos shell de cada sistema operativo, ya que se basa en archivos de configuración:
    a. **XML**.
    b. HTML.
    c. ANT.
    d. ZIP.

23. En Apache las directivas SSLCertificateFile y SSLCertificatKeyFile definen:
    a. La clave privada del servidor Apache y el certificado digital del propio servidor Apache respectivamente.
    b. **El Certificado digital del propio servidor Apache y la clave privada del servidor Apache respectivamente.**
    c. La clave pública del navegador y la clave privada del servidor respectivamente.
    d. La clave privada del navegador y la clave privada del servidor respectivamente.

24. En Tomcat el motor JSP encargado de compilar las páginas JSP, en código java, en servlets que puedan ser manejados por Catalina se denomina:
    a. Jasper.
    b. Coyote.
    c. RoadRunner.
    d. Catalina.
25. La inyección de comandos SQL
    a. Consiste en identificar, en todo momento, quién es el usuario que está accediendo.
    b. Es lo mismo que autenticación con certificados.
    c. Es una técnica para explotar aplicaciones web que no validan la información suministrada por el cliente para generar consultas SQL peligrosas.
    d. Consiste en la validación de entrada SQL.

26. Acerca del balanceo de carga, señala la afirmación que es falsa.
    a. Consiste en establecer un método de reparto de la carga de peticiones entre los servidores del cluster.
    b. No es posible aplicarlo con Tomcat.
    c. Trata de que se minimice el tiempo de respuesta a las solicitudes de los clientes.
    d. Todas son correctas.

27. Señala qué característica es falsa acerca del servidor de aplicaciones JBoss.
    a. Es de código abierto.
    b. Está implementado en su totalidad en Java.
    c. Es únicamente un "EJB Container".
    d. Funciona únicamente en servidores Microsoft Windows.

28. ¿Cuál de las siguientes afirmaciones es falsa?
    a. Un archivo .war puede estar formado por varios archivos .ear
    b. El comando #jar cvf permite generar archivos .war
    c. El comando #jar cvf permite generar archivos .ear
    d. Un archivo .ear puede estar formado por varios archivos .war

29. Con Ant, un task o tarea es
    a. El elemento raíz del fichero XML.
    b. Un conjunto de tareas que queremos aplicar a nuestra aplicación en algún momento.
    c. Un código ejecutable que aplicaremos a nuestra aplicación.
    d. Una propiedad o property es, simplemente, algún parámetro (en forma de par nombre-valor) que necesitamos para procesar nuestra aplicación.

30. Sobre los tipos MIME, DefaultType text/plain significa que cuando el navegador web solicita y recibe ese archivo como respuesta:
    a. Desplegará el contenido como un archivo de texto.
    b. Desplegará el contenido como un archivo HTML.
    c. Desplegará el contenido como un archivo de imagen gif.
    d. Desplegará el contenido como un archivo de vídeo mov.

Módulo: Despliegue de Aplicaciones Web (**DAW**)
I.E.S.: Aguadulce **C.F.G.S. : Desarrollo de Aplicaciones Web**
Fecha: Martes 5 de febrero de 2013 **Hora de Comienzo:** 18:30 horas
Examen: Teórico y Práctico. **Duración:** 3 horas
Nombre: **D.N.I.:**
Centro en el que se realiza el examen:

CUESTIONES PRÁCTICAS:

Suponiendo que dispones de una máquina que cuenta con un sistema operativo Linux recientemente actualizado, esta máquina tiene el entorno de red configurado y, además, dispones de conexión a Internet. Además, estás trabajando con la cuenta del usuario root.

1. **Explica qué hace cada una de las instrucciones siguientes.** (0,9 puntos)

   - **`#apt-get install apache2`**

   Instalar el servidor web Apache desde terminal.

   - **`#/etc/init.d/apache2 status`**

   Comprobar que está funcionando el servidor Apache desde terminal.

   - **`http://localhost ó también http://127.0.0.1`**

   Comprobar que está funcionando el servidor Apache desde navegador.

   - **`#/etc/init.d/apache2 restart`**

   Reiniciar el servidor Apache

2. **Si edito el fichero /etc/apache2/sites-enabled/000-default y en él cambio donde aparece** `<VirtualHost *:80>` **por:** `<VirtualHost *:82>`, **¿qué estoy haciendo?** (0,5 puntos)

   Estoy cambiando el puerto por defecto por el cual está escuchando el virtualhost en Apache, pasándolo al puerto 82

3. **Si quiero instalar Tomcat, explica para qué sirve hacer:** (0,9 puntos)

   - `#aptitude search "?provides(java-runtime)"`

   Para instalar cualquier versión de Tomcat es necesario tener instalado JDK (Kit de desarrollo de Java), para buscar el paquete ejecuto esa instrucción.

   - ¿dónde y para qué se meten estas líneas?

     ```
     JAVA_HOME=/usr/lib/jvm/java-6-openjdk/jre/
     PATH=$PATH:$JAVA_HOME/bin
     export PATH JAVA_HOME
     ```

Sirven para crear una variable de entorno, para indicar en donde se ha instalado y añadir a la variable PATH el directorio en donde se encuentran los archivos binarios para que puedan ser invocados desde cualquier parte, para ello se edita **/etc/profile** introduciendo esas líneas.

- `#source /etc/profile`

Actualizamos las variables de entorno mediante el comando

```
#catalina start
```

Arrancar el servicio correspondiente a Tomcat mediante el script "catalina"

- `http://127.0.0.1:8080`

Comprobar desde el navegador que el Tomcat está funcionado.

4. **Tenemos que configurar un virtualhost basado en nombre, denominado casa, que permita el acceso de la página web de una empresa en Internet, al directorio del servidor web casa. A continuación se ponen instrucciones referentes a eso. Explica los pasos que seguirías para configurar el virtual host.** (0,8 puntos)

```
a2ensite casa
/etc/init.d/apache2 restart


<VirtualHost *:80>
      DocumentRoot /var/www/casa
</VirtualHost>


mkdir /var/www/casa
cd  /etc/apache2/sites-available
```

Acceder a la ruta /etc/apache2/sites-available

cd /etc/apache2/sites-available

Crear el directorio casa:

mkdir /var/www/casa

Crear el fichero casa con el contenido:

<VirtualHost *:80>

DocumentRoot /var/www/casa

</VirtualHost>

Habilitar el virtualhost creado: a2ensite casa

Recargar ó reiniciar Apache: /etc/init.d/apache2 restart

5. **Dado el siguiente el fichero de un virtual host, explica para qué sirven los bloques señalados en negrita:** (0,5 puntos)

```
<VirtualHost *:80>
DocumentRoot /var/www/todo-empresa-tarea-daw02
ServerName www.empresa-tarea-daw02.local
ServerAlias empresa-tarea-daw02.local

<Directory /var/www/todo-empresa-tarea-daw02/videos>
        DefaultType video/x-flv
#       ForceType video/x-flv
</Directory>

<Directory /var/www/todo-empresa-tarea-daw02/delimitado/>
        AllowOverride AuthConfig
        Options Indexes FollowSymLinks MultiViews
        Order allow,deny
        allow from all
</Directory>

SSLEngine on
SSLCertificateFile    /etc/apache2/tus-ssl/empresa-tarea-daw02.crt
SSLCertificateKeyFile /etc/apache2/tus-ssl/empresa-tarea-daw02.key

ErrorLog ${APACHE_LOG_DIR}/empresa-tarea-daw02-error.log
LogLevel warn
CustomLog ${APACHE_LOG_DIR}/empresa-tarea-daw02-access.log combined

</VirtualHost>
```

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, para aquellos los cuales no se resuelven según su extensión. Para el servidor Apache se utilizan dos directivas: DefaultType y ForceType. Por tanto estamos configurando para que sea posible la identificación correcta del vídeo presentación formato flv situado dentro del directorio videos.

En el segundo bloque en negrita, se activa SSL y se indica el certificado digital del servidor Apache y la clave privada del servidor

El tercer bloque permite registrar todos los errores que encuentre Apache. Con combinedse puede especificar una gran variedad de cosas, usuario que ha realizado la petición, etc.

6. **Dado os dos siguientes fragmentos, en relación a JBoss Application Server, explica qué es lo que se pretende hacer con dichos fragmentos de código.** (0,4 puntos)

```
#group add jboss
#useradd -s /bin/bash -g jboss jboss
#passwd jboss
#usermod -d /usr/local/jboss/jboss-6.0.0.Final/ jboss
```

Crear el usuario de JBoss que posee y dirige JBoss, se crea un grupo JBoss y un usuario llamado JBoss al se agrega al grupo creado

```
JAVA_HOME=/usr/lib/jvm/java-6-sun/jre
JBOSS_HOME=/usr/local/jboss/jboss-6.0.0.Final
PATH=$PATH:$JAVA_HOME/bin:$JBOSS_HOME/bin
export PATH JAVA_HOME JBOSS_HOME
```

Establecer las variables de entorno JAVA_HOME **y** JBOSS_HOME: Estas variables son interesantes para indicar las rutas donde se ha instalado Java y JBoss. Estás rutas serán empleadas en los archivos de configuración de dichas aplicaciones. Esas modificaciones se realizan en el archivo **/etc/profile**:

Módulo: Despliegue de Aplicaciones Web (**DAW**)
I.E.S.: Aguadulce **C.F.G.S. : Desarrollo de Aplicaciones Web**
Fecha: Jueves 13 de junio de 2013 **Hora de Comienzo:** 15:30 horas
Examen: Teórico y Práctico. **Duración:** 2 horas 30 minutos
Nombre: **D.N.I.:**
Centro en el que se realiza el examen:

INSTRUCCIONES:
La puntuación total del examen será de 10 puntos. Parte teórica: 6 puntos. Parte Práctica: 4 puntos.
El alumnado que superó el examen de febrero debe hacer:

- Preguntas teóricas desde la 21 a la 40. (0,30 puntos cada pregunta. Cada dos mal quita una bien).
- De la parte práctica los ejercicios desde el 5 en adelante. Puntúan respectivamente:
  - Ejercicio 5 → 0,6 puntos
  - Ejercicios 6, 7 y 9 → 0,8 puntos cada uno.
  - Ejercicio 8 → 1 punto.

El resto del alumnado debe hacer el examen completo:

- Cada pregunta de teoría vale 0,15 puntos. Cada dos mal quita una bien.
- En la parte práctica, los ejercicios (1,2,3,4,5,9) valen 0,4 puntos cada uno. El 6 y 7 vale 0,5 puntos, y el 8 vale 0,6 puntos.

Cualquier decisión en la parte práctica, por algo que no quedara claro en el enunciado, coméntalo justificando la decisión.

CUESTIONES TEÓRICAS:

1. **Si queremos probar un servlet, una vez arrancado el servidor Tomcat, abrimos un navegador web, en el cual escribiríamos una URL con el siguiente formato:**
   a. http://{port}:{address}/{servletName}
   b. http://{address}:{servletName}/{port}
   c. http://{address}:{port}/{servletName}
   d. http://{servletName}:{port}/{address}

2. **La forma más habitual de integrar la ejecución de páginas PHP en el servidor web Apache, es utilizar...**
   a. Un servidor de aplicaciones.
   b. Una solución como FastCGI.
   c. El módulo mod_php.
   d. Tomcat.

3. **El servidor Tomcat se gestiona a través de un script denominado:**
   a. tomcatina
   b. sys.ctl
   c. catalina
   d. apache2

4. **Señala cuál no es un factor de los que han impulsado el uso de los servicios Web.**
   a. Ancho de banda más barato.
   b. Sistemas operativos Microsoft Windows de "Open Source".
   c. Éxito de la computación extendida.
   d. Contenido web más dinámico.

5. **Los middlewares actúan como elementos conectores encargados de distribuir la aplicación web...**
   a. De forma horizontal mediante un balanceador software.
   b. De forma vertical mediante un balanceador hardware FTP.
   c. De forma horizontal mediante un balanceador hardware HTTP.
   d. De forma vertical distribuyendo las capas de la aplicación en diferentes máquinas.

6. **La arquitectura Web es un modelo de tres capas que son:**
   a. Base de Datos, Sistema Operativo, Navegador Web.
   b. Base de Datos, Servidor Linux, Servidor IIS.
   c. Base de Datos, Servidor de aplicaciones Web, clientes de servicio Web.
   d. Base de Datos, Clientes HTML , Clientes XML.

7. **Una plataforma WIMP usa MySQL, PHP y ¿qué sistema operativo?**
   a. Microsoft Windows.
   b. Linux.
   c. Unix.
   d. MAC.

8. **Por defecto, Tomcat va a estar escuchando en el puerto:**
   a. 8081
   b. 8088
   c. 127.1
   d. 8080

9. **Señala la opción incorrecta. El programa rotatelog es un programa proporcionado por Apache que permite:**
   a. **Solamente rotar logs por tamaño de los mismos.**
   b. Rotar logs por tamaño de los mismos.
   c. Rotar logs.
   d. Rotar logs por intervalo de tiempo siempre y cuando en el intervalo de tiempo definido existan nuevos datos.

10. En un host virtual, encontramos que se atenderán peticiones por cualquier interfaz de red y a través del puerto 80 con:
    a. <VirtualHost *:??>
    b. **<VirtualHost *:80>**
    c. <VirtualHost 192.168.200.0:80>
    d. <VirtualHost *:*>

11. **rotatelog** es un programa proporcionado por Apache que permite:
    a. Rotar logs por tamaño de los mismos.
    b. Rotar logs.
    c. Rotar logs por intervalo de tiempo siempre y cuando en el intervalo de tiempo definido existan nuevos datos.
    d. Todas las anteriores con correctas.

12. Si queremos acceder a una aplicación, llamada **iesaguadulce**, que se está ejecutando en un servidor Tomcat, desde el propio equipo, tendríamos que teclear en el navegador:
    a. http://localhost:8080
    b. http://localhost:80/iesaguadulce
    c. http://localhost:8080/iesaguadulce
    d. http://localhost:8080/tomcat

13. **Los ataques que suele recibir una aplicación web se pueden centrar en varios objetivos, señala cuáles de los siguientes podrían ser:**
    a. Ataques a la computadora del usuario (cliente).
    b. Ataques al servidor.
    c. Ataques al flujo de información entre cliente y servidor.
    d. Todos los anteriores son correctos.

14. En una comunicación cliente(A)-servidor(B) mediante el cifrado asimétrico, 'A' envía la información cifrada mediante la clave (señala la opción) y 'B' la descifra mediante su clave privada.
a. privada de 'B'
b. privada y pública de 'B'
c. privada y pública de 'A'
d. pública de 'B'

15. Si configuramos las sesiones persistentes de forma global tenemos que manipular el archivo:
a. <CATALINA_HOME>/conf/context.xml.
b. <CATALINA_HOME>/bin/context.xml.
c. <CATALINA_HOME>/META-INF/context.xml.
d. <CATALINA_HOME>/config/context.xml.

16. ¿Qué archivo contiene información sobre la configuración web de Tomcat?
a. PHP.ini
b. host-manager.xml
c. build.xml
d. $CATALINA_HOME.

17. En Apache las directivas SSLCertificateFile y SSLCertificatKeyFile definen:
a. La clave privada del servidor Apache y el certificado digital del propio servidor Apache respectivamente.
b. La clave pública del navegador y la clave privada del servidor respectivamente.
c. **El certificado digital del propio servidor Apache y la clave privada del servidor Apache respectivamente.**
d. La clave privada del navegador y la clave privada del servidor respectivamente.

18. Según la especificación httpd 2.2 una aplicación web puede ser desplegada en diferentes servidores web manteniendo su funcionalidad y sin ningún tipo de modificación en su código.
a. Verdadero.
b. Falso.

19. La inyección de comandos SQL
a. Consiste en identificar, en todo momento, quién es el usuario que está accediendo.
b. Es lo mismo que autenticación con certificados.
c. Es una técnica para explotar aplicaciones web que no validan la información suministrada por el cliente para generar consultas SQL peligrosas.
d. Consiste en la validación de entrada SQL.

20. Uno de los requisitos previos para instalar JBoss es tener instalado:
a. Una distribución Linux.
b. Microsoft Windows.
c. Mozilla Firefox.
d. JDK.

21. Asocia las siguientes directivas del servidor ProFTPD con su posible uso (escribe el número):
a. Escuchar la conexión en un puerto TCP. → 2
b. Define el fichero de autenticación de usuarios. → 4
c. Definición de la ruta que sirve ProFTPFD. → 1
d. Configura el nombre que se muestra en la conexión de los usuarios. → 3
(1) DocumentRoot
(2) Port 2121
(3) Server Name
(4) AuthUserFile

22. El protocolo FTP permite cifrar la información entre el cliente y el servidor FTP.
a. Sí, mediante el método AUTH basic auto.
b. Sí, independientemente del modo de funcionamiento usado.
c. No.
d. Sí, siempre y cuando el cliente sea el origen de la información.

23. Los permisos rw-r--r-- identifican:
a. Los permisos: 655.
b. Los permisos: 644.
c. Los permisos: 544.
d. Los permisos: 744.

24. Los comandos close y disconnect:
a. Terminan la sesión ftp y salen de la consola ftp.
b. Salen a línea de comandos del sistema operativo temporalmente sin cortar la conexión. Para volver debes teclear exit en la línea de comandos.
c. Son alias de los comandos qut y ext.
d. Finalizan la sesión ftp sin cerrar la consola ftp.

25. Los comandos quit y exit
a. Finalizan la sesión ftp sin cerrar la consola ftp.
b. Salen a línea de comandos del sistema operativo temporalmente sin cortar la conexión. Para volver debes teclear exit en la línea de comandos.
c. Terminan la sesión ftp y salen de la consola ftp.
d. Son alias de los comandos qut y ext.

26. Dos nombres DNS distintos que identifican a la misma máquina y a la misma IP, pueden servir una configuración ftp distinta mediante el servidor PropFTPD.
a. No, es imposible.
b. No, es imposible para cualquier servidor ftp.
c. Sí, siempre y cuando cada virtualhost, así como el servidor principal, deben servir en un puerto TCP distinto.
d. Sí, es posible, configurando virtualhosts.

27. El servidor ftp ProFTPD permite configurar hosts virtuales:
a. No.
b. Sí, pero sólo en Windows
c. Sí, por ejemplo basados en IP.
d. La configuración de hosts virtuales únicamente es posible en servidores web.

28. ¿Qué registro de recursos permite combatir el correo electrónico no deseado?
a. SOA
b. CNAME
c. A
d. SPF

29. El servidor DNS permite traducir una IP a su nombre de dominio correspondiente únicamente cuando es una IP versión 4.
a. Sí, puesto que la IP versión 6 no lo permite.
b. No.
c. Sí, mediante resolución directa.
d. Sí, mediante resolución inversa.

30. El comando host:
a. No permite realizar resoluciones inversas y directas.
b. Solamente permite realizar resoluciones inversas.
c. Solamente permite realizar resoluciones directas, esto es, de nombres de dominio DNS a IP.
d. Permite realizar resoluciones inversas y directas.

31. En las consultas iterativas, el cliente:
a. Está interesado en la mejor respuesta por parte del servidor DNS consultado.
b. Quiere que el servidor DNS le resuelva totalmente la consulta.
c. Quiere que el servidor DNS no use recursividad y sí iteración con otros servidores DNS.
d. Quiere que le iteren el servidor DNS habilitando y deshabilitando la tarjeta de red cada tres segundos.

32. La entrada dn: **uid=usuario,ou=People,dc=empresa,dc=local** es:
a. Una entrada definida incorrectamente.
b. Una entrada definida sobre Java bajo Linux.
c. Una entrada dentro del dominio empresa.local.
d. Un objeto no unívoco.

33. ¿Qué se debe actualizar siempre cada vez que modificas la base de datos del servidor DNS?
a. Tiempo de vida mínimo.
b. Intervalo de reintento.
c. Intervalo de actualización.
d. Número de serie.

34. El directorio LDAP tiene una estructura en forma de árbol denominado
a. ND
b. CN
c. objectClass
d. DIT

35. Una plantilla en Javadoc se compone de
a. Un nombre, una descripción, un contexto en función del lenguaje y un pattern.
b. Un nombre, una descripción.
c. Un nombre, un contexto en función del lenguaje y un pattern.
d. Un nombre y un pattern.

36. Para establecer la ruta donde guardar la documentación generada por phpDocumentor, es necesario establecer el valor deseado en el archivo .ini de nuestro proyecto, cambiando en él el contenido de la línea:
a. target.
b. directory.
c. destination.
d. path.

37. Indica cuál de los siguientes no es un estado en el que se pueden encontrar los archivos, de un proyecto integrado con Git.
a. Confirmado.
b. Modificado.
c. Preparado.
d. Inmutable.

38. ¿Qué sección de Git almacena los metadatos y la base de datos de objetos para el proyecto?
a. Git directory.
b. working directory.
c. tating directory.
d. staging area.

39. Los sistemas de control de versiones son programas que permiten...
- **a. Gestionar un repositorio de archivos y sus distintas versiones.**
- b. Documentar aplicaciones.
- c. Insertar comentarios en el código fuente de las aplicaciones.
- d. Compilar aplicaciones web.

40. ¿Qué comando muestra cómo se encuentran los archivos de nuestro proyecto en relación al repositorio del servidor?
- **a. git status**
- b. git clone
- c. git add
- d. git cvs

CUESTIONES PRÁCTICAS:

Suponiendo que dispones de una máquina que cuenta con un sistema operativo Linux recientemente actualizado, esta máquina tiene el entorno de red configurado y, además, dispones de conexión a Internet. Además, estás trabajando con la cuenta del usuario `root`.

1. Contesta brevemente, ¿para qué sirven cada una de las instrucciones siguientes?

- **`#apt-get install apache2`**

Instalar el servidor web Apache desde terminal.

- **`#/etc/init.d/apache2 status`**

Comprobar que está funcionando el servidor Apache desde terminal.

- **`http://localhost`**

Comprobar que está funcionando el servidor Apache desde navegador.

- **`#/etc/init.d/apache2 restart`**

Reiniciar el servidor Apache

2. Instalando Tomcat, explica para qué sirve hacer cada una de las instrucciones siguientes:

- `#aptitude search "?provides(java-runtime)"`

Para instalar cualquier versión de Tomcat es necesario tener instalado JDK (Kit de desarrollo de Java), para buscar el paquete ejecuto esa instrucción.

- ¿Dónde y para qué se introducen estas líneas?

```
JAVA_HOME=/usr/lib/jvm/java-6-openjdk/jre/
PATH=$PATH:$JAVA_HOME/bin
export PATH JAVA_HOME
```

Sirven para crear una variable de entorno, para indicar en donde se ha instalado y añadir a la variable PATH el directorio en donde se encuentran los archivos binarios para que puedan ser invocados desde cualquier parte, para ello se edita **/etc/profile** introduciendo esas líneas.

- `#source /etc/profile`

Actualizamos las variables de entorno mediante el comando

```
#catalina start
```

Arrancar el servicio correspondiente a Tomcat mediante el script "catalina"

- `http://127.0.0.1:8080`

Comprobar desde el navegador que el Tomcat está funcionado.

- ```
  # a2enmod proxy
  # a2enmod proxy_ajp
  # a2enmod proxy_balancer
  ```

Cargar esos módulos para conseguir que Apache funcione como Proxy

3. Tenemos que configurar un virtualhost basado en nombre, denominado empresaDAW, que permita el acceso de la página web de una empresa en Internet, al directorio del servidor web empresaDAW. A continuación se ponen instrucciones referentes a eso. Explica los pasos que seguirías para configurar el virtual host.

```
a2ensite empresaDAW

/etc/init.d/apache2 restart


<VirtualHost *:80>
      DocumentRoot /var/www/empresaDAW
</VirtualHost>


mkdir /var/www/empresaDAW

cd  /etc/apache2/sites-available
```

Acceder a la ruta /etc/apache2/sites-available

cd /etc/apache2/sites-available

Crear el directorio empresaDAW:

mkdir /var/www/empresaDAW

Crear el fichero empresaDAW con el contenido:
```

<VirtualHost *:80>

DocumentRoot /var/www/empresaDAW

</VirtualHost>

Habilitar el virtualhost creado: a2ensite empresaDAW

Recargar ó reiniciar Apache: /etc/init.d/apache2 restart

4. **Para crear un servidor seguro en Apache, con una distribución Ubuntu el procedimiento sería el siguiente, describe brevemente lo que describe las instrucciones:**

- `apt-get install openssl`

  Instalar el paquete openssl

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

  `make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/apache2/tus-ssl/apache.pem`

  Crear un certificado autofirmado para el servidor web, cuando se solicite el nombre del servidor HTTP se indicará el nombre DNS que corresponda a la IP del certificado

- Editamos el contenido de **`/etc/apache2/sites-available/default-ssl`** siendo:

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

  Indicar el certificado del servidor y su respectiva clave privada asignando esos valores a los parámetros.

- Hemos de asegurarnos de que el fichero `/etc/apache2/ports.conf` incluya ¿qué valor?

  El valor Listen 443

- Comenta para qué sirven estas instrucciones

```
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
```

  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. Útiles sobre todo si no dispones de los comandos **a2enmod** y **a2dismod** para habilitar y deshabilitar módulos Apache

5. **En relación a JBoss Application Server, explica qué es lo que se pretende con estos fragmentos de código.**

```
#group add jboss
#useradd -s /bin/bash -g jboss jboss
#passwd jboss
#usermod -d /usr/local/jboss/jboss-6.0.0.Final/ jboss
```

Crear el usuario de JBoss que posee y dirige JBoss, se crea un grupo JBoss y un usuario llamado JBoss al se agrega al grupo creado

```
JAVA_HOME=/usr/lib/jvm/java-6-sun/jre
JBOSS_HOME=/usr/local/jboss/jboss-6.0.0.Final
PATH=$PATH:$JAVA_HOME/bin:$JBOSS_HOME/bin
export PATH JAVA_HOME JBOSS_HOME
```

Establecer las variables de entorno JAVA_HOME **y** JBOSS_HOME: Estas variables son interesantes para indicar las rutas donde se ha instalado Java y JBoss. Estás rutas serán empleadas en los archivos de configuración de dichas aplicaciones. Esas modificaciones se realizan en el archivo **/etc/profile**:

- groupadd jboss

Crear el grupo para JBoss.

- useradd -s /bin/bash -g jboss jboss

Crear el usuario jboss.

- sh /usr/local/jboss/bin/standalone.sh

Arrancar JBoss

6. Responde explicando brevemente par qué sirven las siguientes instrucciones que ejecutemos en un terminal, en relación a la instalación y configuración de servidores FTP.

- `ls -l /etc/proftpd`

Comprobar que proftpd está instalado en /etc

- `service proftpd start`

Iniciar el servicio proftpd:

- `id ftp`

comprobar el número de identificación del usuario del sistema ftp

- `ftpasswd --passwd --name examen --file /etc/passwd.usuarios.virtuales --uid 133 --home /var/ftp/todo-examenDAW --shell /bin/false`

Estamos indicando que vamos a crear un usuario denominado examen, sobre el que nos preguntará una contraseña que será almacenada en /etc/passwd.usuarios.virtuales. Este usuario tiene el número de identificación 133, su carpeta principal será la que hay en /var/ftp/todo-examenDAW. No tendrá acceso al shell del sistema.

- `chown ftp /var/ftp -R`

Estamos haciendo que el usuario ftp sea el propietario de la carpeta ftp y de todas sus subcarpetas.

- Editamos con: `gedit /etc/proftpd/virtuals.conf` y añadimos lo siguiente:

```
<VirtualHost 192.168.1.23>
      ServerAdmin ftpadmin@examenDAW.local
      ServerName “Servidor FTP para el examen de DAW”
      AuthUserFile /etc/passwd.usuarios.virtuales
      DefaultRoot /var/ftp/todo-examenDAW
      RequireValidShell off
      AllowOverwrite on
</VirtualHost>
```

Estamos diciendo que queremos crear un virtualhost para la dirección 192.168.1.23, al cual le indicamos un email para el administrador, un nombre para el servidor, decimos que que los usuarios autorizados se encuentran en el fichero /etc/passwd.usuarios.virtuales. También decimos que el directorio raíz para este virtualhost es /var/ftp/todo-examenDAW, que no requiere usuarios con un shell del sistema válido y que se permite la escritura en él.

7. Sobre la configuración de DNS, comenta brevemente para qué sirven las siguientes instrucciones.

- `apt-get install bind9 bind9utils`

Instalamos los paquetes necesarios para el funcionamiento de BIND:

- `service bind9 status`

  Verificamos que está instalado y funcionando correctamente:

- Editamos el fichero `/etc/bind/named.conf` y escribimos en él:

```
zone “dominioempresa-examen-daw05.local”{
                type master;
                file “/var/lib/bind/master/db.dominioempresa-examen-
daw05.local”;
        };
```

  Configuro el servidor de DNS como maestro.

- Si modificamos el fichero /etc/resolv.conf para que sólo tenga activa la siguiente línea:

  nameserver 127.0.0.1

  Establecemos el servidor DNS activo solamente es el local.

- Si ejecutamos en una consola: `$ nslookup www.google.es`, ¿qué obtendremos?

  nos dirá cual es la IP del servidor de Google.

- `netstat -natp | grep named` y también `netstat -naup | grep named`

  Verifica en qué puertos TCP y UDP está activo el servidor bind9, para ello comprueba el servicio named

- ¿Para qué sirven los comandos “named-checkconf” y “named-checkzone” ?

  Para realizar una verificación de los ficheros de configuración y de zona por posibles fallos

8. **Acerca de la configuración de LDAP, aquí tienes varias líneas de código que te pueden resultar útiles y que corresponden a unos ficheros de extensión .ldif. Se pide que indiques el contenido de tres ficheros a los que llamaríamos estructura.ldif, grupo.ldif y usuarios.ldif, que generarías suponiendo que tienes un nombre de dominio: examenDAW.local**

```
dn: cn=users,ou=grupo,dc=lperengano,dc=local
objectclass: posixGroup
```

```
objectclass: top
cn: users
gidNumber: 113
```

```
dn: ou=tienda,dc=lperengano,dc=local
objectClass: organizationalUnit
ou:currucucu
```

```
dn: ou=grupo,dc=lperengano,dc=local
ou: grupo
objectclass: organizationalUnit
```

- **Contenido del fichero estructura.ldif para crear una unidad organizacional "junio".**

dn: ou=junio,dc=examenDAW,dc=local
ou:junio
objectClass: organizationalUnit

- **Contenido del fichero grupo.ldif para crear una unidad organizacional "daw_daw".**

dn: ou=daw_daw,dc=examenDW,dc=local
ou: daw-daw
objectclass: organizationalUnit

- **Contenido de un fichero usuarios.ldif para crear un grupo users_daw.**

dn: cn=users:daw, ou=daw_daw,dc=examenDAW,dc=local
objectclass: posixGroup
objectclass: top
cn: users_daw
gidNumber: 114

- **¿Para qué sirve esta instrucción, supuesto que en usuario.ldif tenemos los datos de un usuario?**

```
ldapadd -x -W -D “cn=admin,dc=undominio,dc=local” -f usuario.ldif
```

Agrega el nuevo usuario al nuevo árbol de LDAP en el dominio: undominio.

- **Completa el comando supuesto que hemos obtenido al ejecutarlo lo siguiente:**

```
adding new entry "uid=upruebas,ou=usuarios,dc=proyecto-empresa,dc=local"
```

```
ldapadd -x -D cn=admin,dc=                ,dc=      -w admin -f usuario.ldif
```

ldapadd -x -D cn=admin,dc=proyecto-empresa,dc=local -w admin -f usuario.ldif

- **¿Qué haría la siguiente instrucción?**

```
ldapsearch -h 192.168.200.250 -x -b dc=proyecto-empresa,dc=local
"(objectclass=*)"
```

Conectaría con el servidor LDAP en la IP 192.168.200.250 para buscar el DIT del dominio proyecto-empresa.local.

9. **En relación a Git, y a su instalación y configuración, comenta brevemente para qué sirven las siguientes instrucciones:**

- **apt-get install php-pear**

Instalar el paquete phppear, o sea, el entorno de desarrollo y sistema de distribución para componentes de código php.

- **git config --global core.editor Emaus**

Cambiar el editor de texto que trae por defecto Git al editor emacs.

- **git config --global user.name “alumn”**

  git config --global user.email alumn@example.com

Establecer el nombre de usuario y dirección de correo electrónico en la configuración de Git.

- **Cómo obtener Git a través del propio Git para futuras actualizaciones, de manera que descargaría automáticamente el código fuente desde su repositorio.**

Utilizando el comando clone de git configuramos su propio repositorio para actualizar el código fuente:

git clone git://git.kernel.org/pub/scm/git/git.git