

Módulo Superior de Diseño de Aplicaciones Web

Despliegue de Aplicaciones Web

Tarea 3

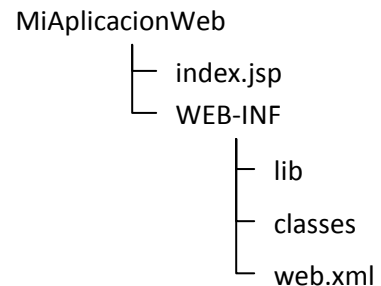
José Luis Comesaña Cabeza



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 , ¿cual 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 subcarpetas, 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 una vez descargada, la descomprimos 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:

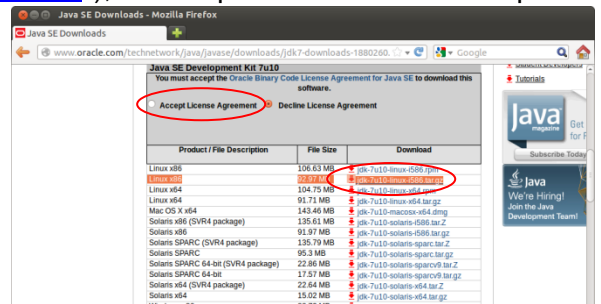
```
gedit Ejemplotar3/index.html
```

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:

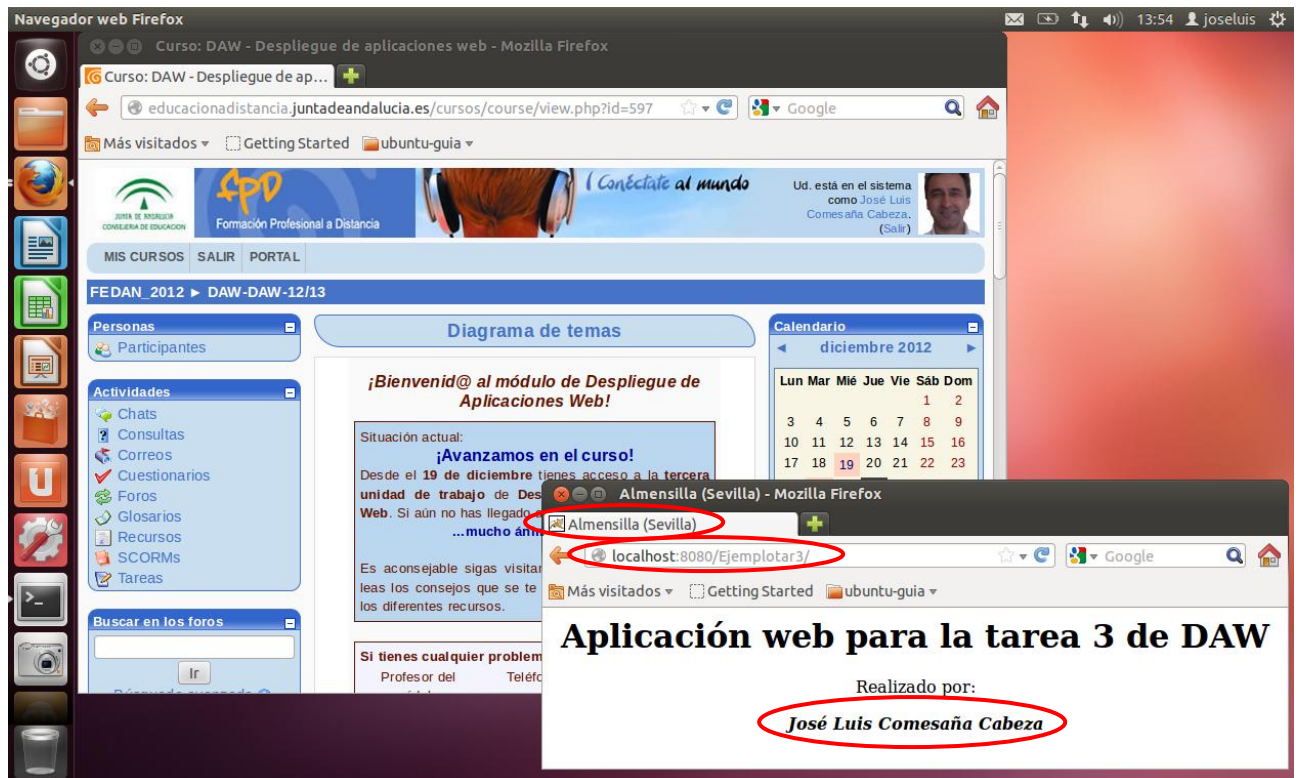
```
cp -r ./Ejemplotar3 /usr/local/apache-tomcat-6.0.36/.
```



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.AccessLogValve" 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 autenticado
- `%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

" S mplemente 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 aplicaci n creada para que se grabe el acceso realizado. En el navegador web ponemos `localhost:8080/Ejemplotar3/` y aparecer  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 ser , por ejemplo, si hemos llamado a una p gina inexistente denominada `hola` y posteriormente llamamos a otra, tambi n 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 ejecuci n de Java, para lo que podemos verificarlo tecleando `java -version`, y en mi caso me aparece:

```
java versi n "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 direcci n:

<http://www.jboss.org/jbossas/downloads/>

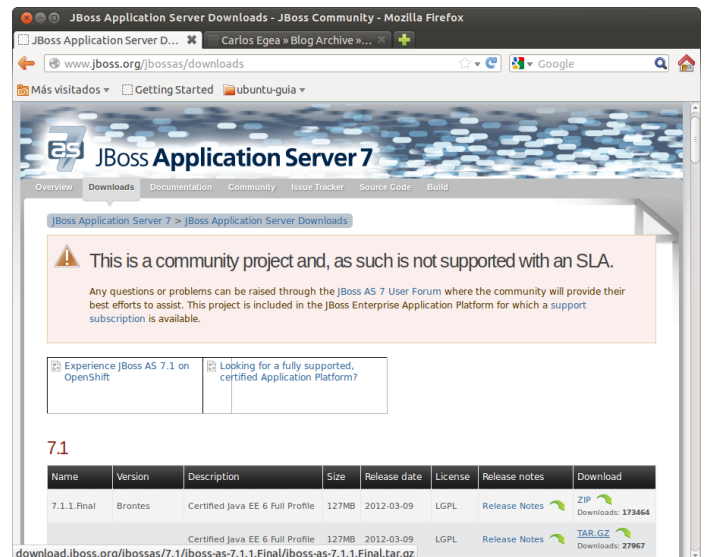
Selecciono la versi n `jboss-as-7.1.1.Final.tar.gz`, ya que es la  ltima disponible y requiere de Java 6 o superior, por lo que no debo tener problemas para instalarlo.

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

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

A continuaci n podemos mover el directorio reci n 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 versi n 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` pregunt ndonos por el tipo, nombre del usuario y su

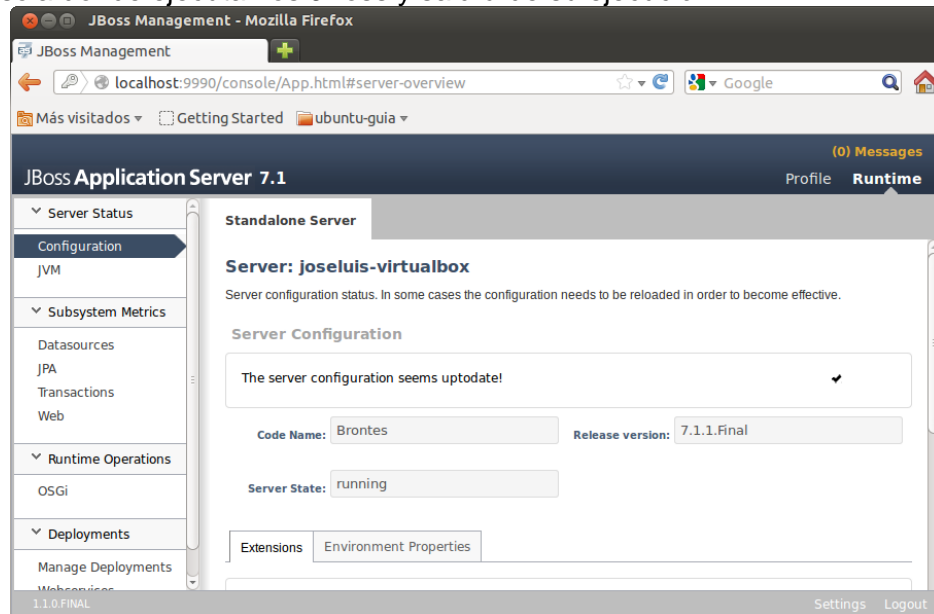
```
root@joseluis-VirtualBox: /usr/local
root@joseluis-VirtualBox: /usr/local/jboss/bin# ./add-user.sh

What type of user do you wish to add?
a) Management User (mgmt-users.properties)
b) Application User (application-users.properties)
(a):

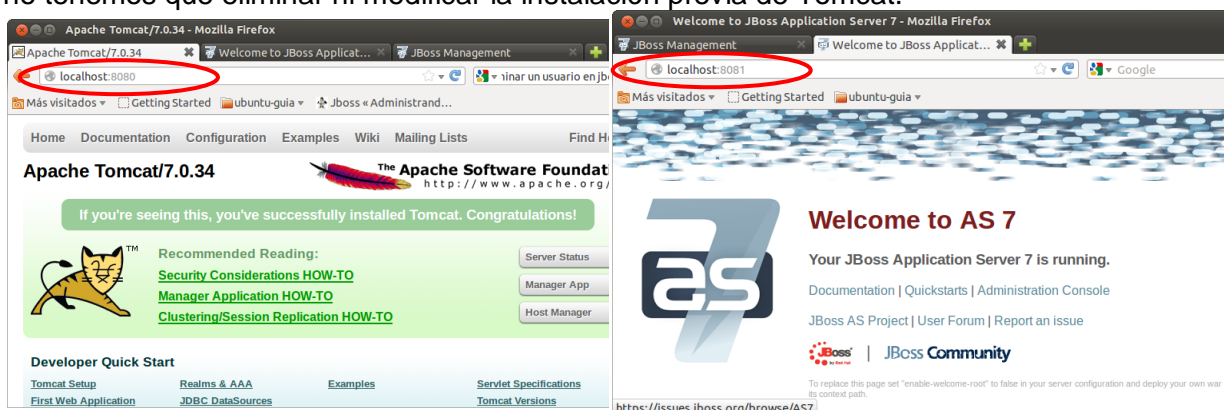
Enter the details of the new user to add.
Realm (ManagementRealm) :
Username : jbossUser
Password :
Re-enter Password :
About to add user 'jbossUser' for realm 'ManagementRealm'
Is this correct yes/no? y
Added user 'jbossUser' to file '/usr/local/jboss/standalone/configuration/mgmt-users.properties'
Added user 'jbossUser' to file '/usr/local/jboss/domain/configuration/mgmt-users.properties'
root@joseluis-VirtualBox: /usr/local/jboss/bin#
```

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.

```
root@joseluis-VirtualBox: /home/joseluis/Descargas
root@joseluis-VirtualBox: /home/joseluis/Descargas# mv apache-ant-1.8.4 /usr/local/apacheAnt184
root@joseluis-VirtualBox: /home/joseluis/Descargas# gedit /etc/profile
root@joseluis-VirtualBox: /home/joseluis/Descargas# gedit /etc/profile
root@joseluis-VirtualBox: /home/joseluis/Descargas# source /etc/profile
root@joseluis-VirtualBox: /home/joseluis/Descargas# ant
Buildfile: build.xml does not exist!
Build failed
root@joseluis-VirtualBox: /home/joseluis/Descargas#
```