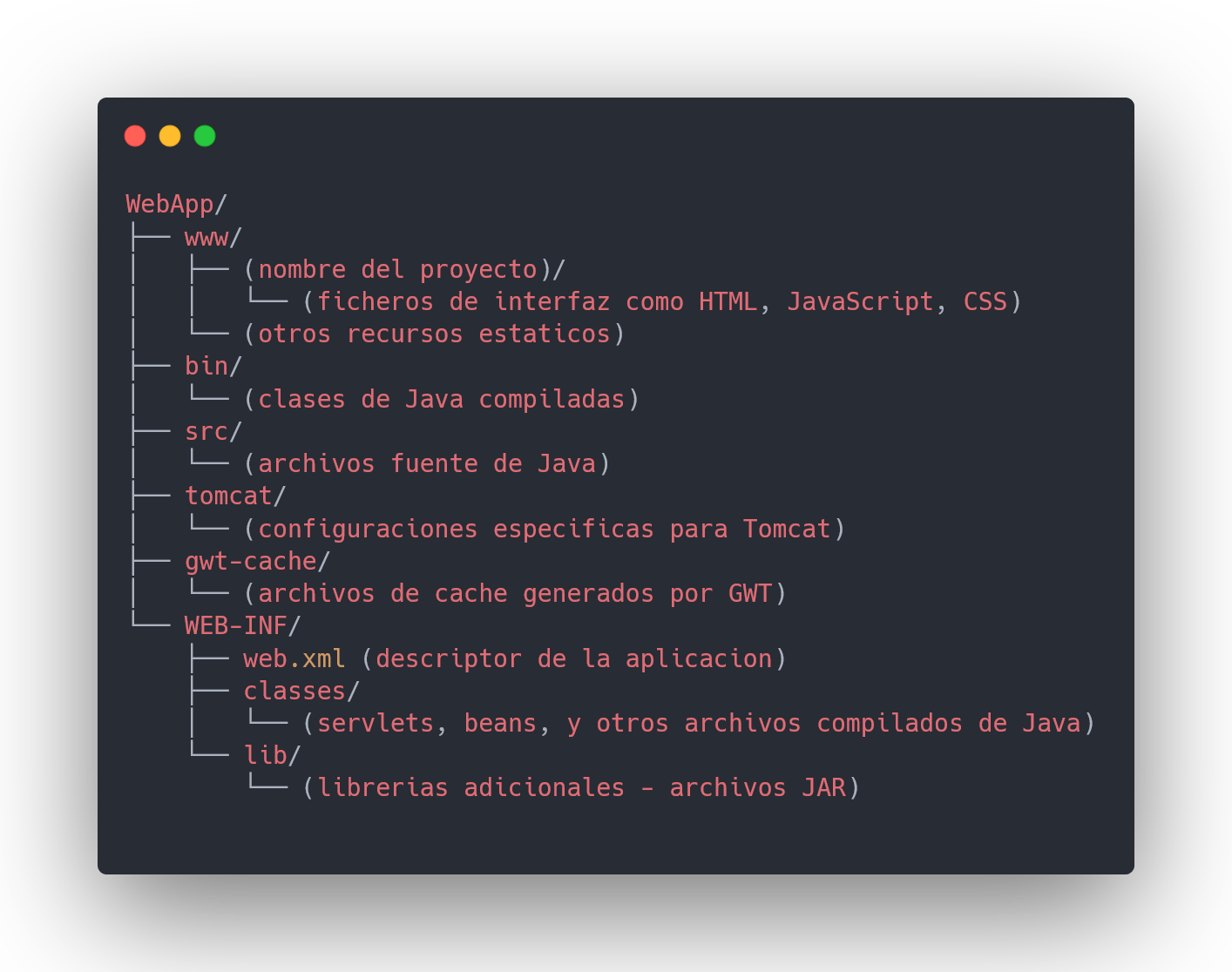


1. **Una aplicación web puede ser desplegada en diferentes servidores web manteniendo su funcionalidad y sin ningún tipo de modificación en su código debido a la especificación servlet 2.2 , ¿cuál es la estructura de directorios que debe tener?**

La estructura de directorios para una aplicación web que cumple con la especificación Servlet 2.2 es la siguiente:



1. **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.**

**<project>**: Esta etiqueta es la raíz del archivo build.xml y define el proyecto en Ant. Contiene información como el nombre del proyecto y la versión, así como las tareas y configuraciones relacionadas.

**<property>**: Se utiliza para definir propiedades personalizadas que pueden ser referenciadas en otros lugares del archivo build.xml. Por ejemplo, para definir rutas de directorios o configuraciones específicas.

**<target>**: Esta etiqueta se utiliza para definir tareas específicas que deben llevarse a cabo. Cada target puede tener un nombre y contendrá una serie de tareas que se ejecutarán en un orden específico.

**<taskdef>**: Se utiliza para definir tareas personalizadas que pueden ser utilizadas en el archivo build.xml. Esto permite extender las capacidades de Ant con tareas personalizadas.

**<javac>**: Esta etiqueta se utiliza para compilar archivos Java. Puede contener opciones y configuraciones específicas de compilación, como la ubicación de los archivos fuente y los archivos de destino.

**<copy>**: Se utiliza para copiar archivos o directorios de un lugar a otro. Puede especificar archivos de origen y directorios de destino.

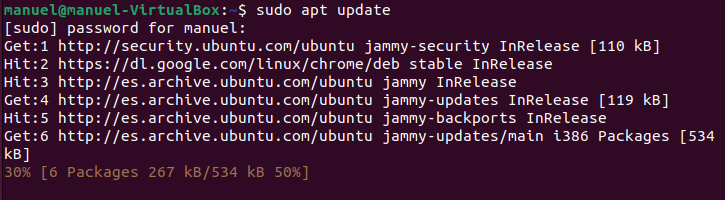
**<delete>**: Esta etiqueta permite eliminar archivos o directorios del sistema de archivos.

**<echo>**: Se utiliza para mostrar mensajes en la salida estándar durante la ejecución del script de construcción. Puede ser útil para proporcionar información sobre lo que está sucediendo en el proceso de construcción.

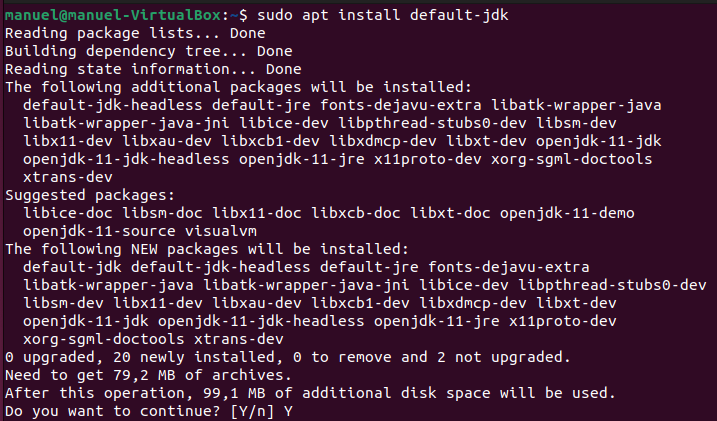
1. **Dispones de una máquina que cuenta con el sistema operativo Ubuntu recientemente actualizado (20.04 LTS), en la que está el entorno de red configurado y, además, dispones de conexión a Internet y estás trabajando con la cuenta del usuario root. Indica cada uno de los pasos, y comandos implicados 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:**

* Instalar el JDK de Sun. (Como probablemente lo hiciste en las tareas anteriores, puedes coger la documentación de la misma tarea que entregaste antes).

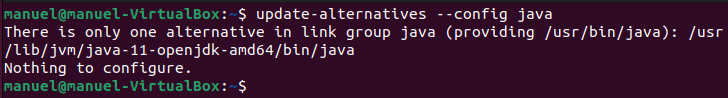
Primero, vamos a actualizar los paquetes de ubuntu usando **apt update**.



Ahora, vamos a instalar el jdk usando **sudo apt install default-jdk**.



Vamos a configurar las variables de entorno para JAVA. Para ello, primero vamos a encontrar la ubicación del JDK:



Ahora que tenemos la ruta de instalación de JDK, vamos a añadirlo a las variables de entorno. Para ello, vamos a modificar el siguiente archivo:



Y añadimos estas 2 líneas al final del archivo:



Vamos a aplicar los cambios sobre las variables de entorno.



Una vez aplicados los cambios, podemos ver que nuestras variables de entorno se han aplicado correctamente imprimiendo su valor por consola.

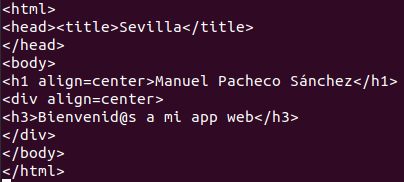


* Crea una aplicación web, basta con que crees simplemente un fichero index.html o index.jsp, similarmente a como se ve en el apartado 2.1. del tema, y que como título de la ventana ponga tu localidad y en el cuerpo (Body) ponga tu nombre y apellidos.

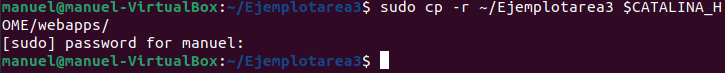
Primero, vamos a crear un directorio donde crear nuestra aplicación, y dentro del directorio vamos a crear un archivo .jsp:



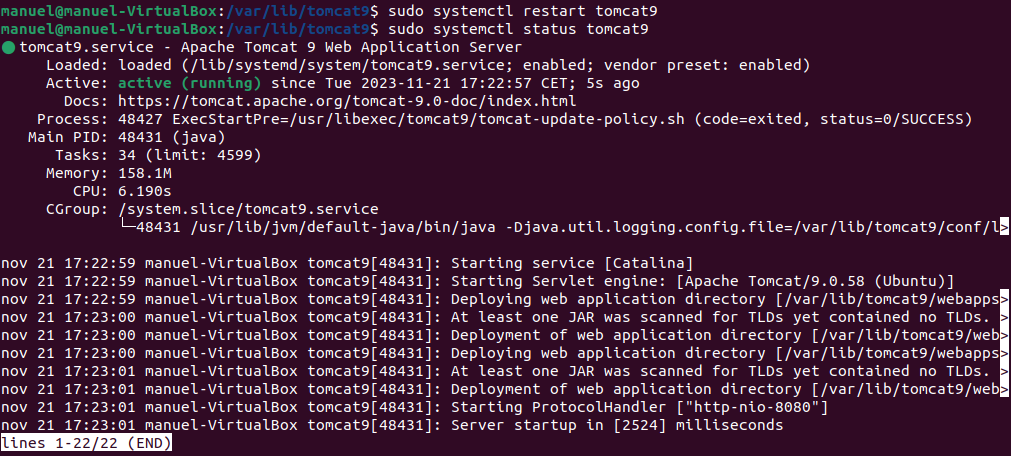
Dentro de este archivo, vamos a crear nuestra pequeña aplicación web.



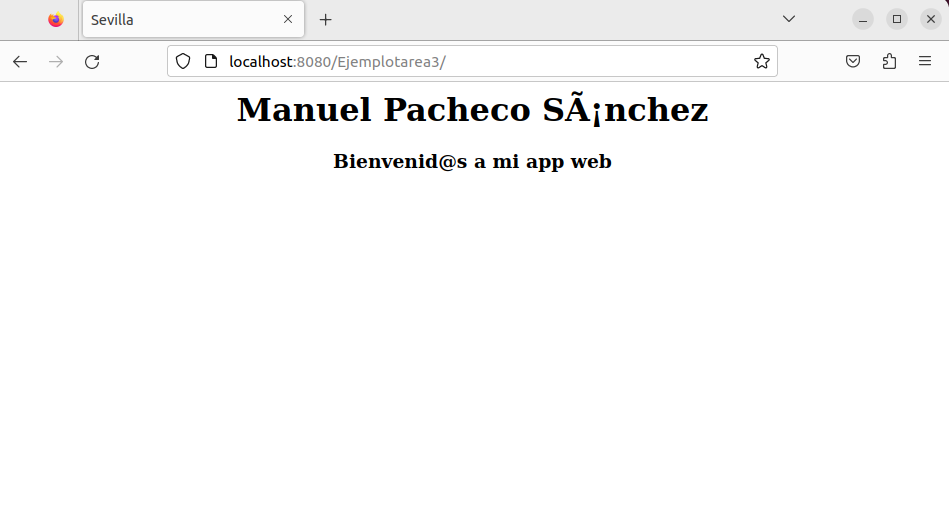
Para desplegar esta aplicación, tenemos que copiar el directorio en **webapps** de Tomcat.



Vamos a reiniciar Tomcat para aplicar todos estos cambios.

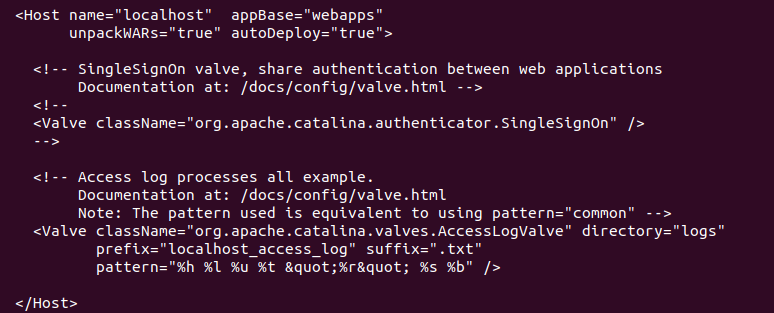


Y ahora vamos a ver que nuestra aplicación está desplegada.

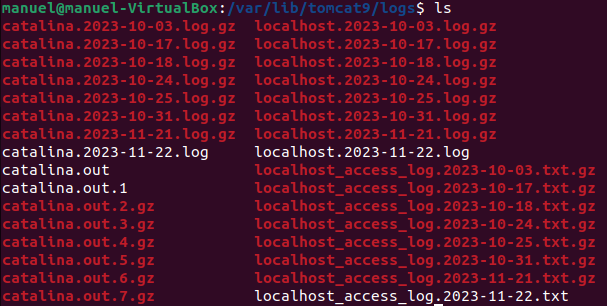


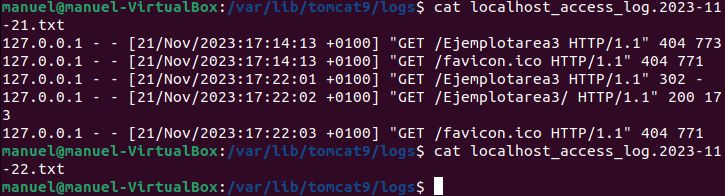
* Indica los pasos a seguir para generar un log de acceso, no olvides al enviar la tarea, adjuntar el fichero o ficheros que hayas modificado (haz una copia de seguridad por si te equivocas al modificar). También se deben adjuntar los ficheros con las capturas de pantalla hechas desde Ubuntu (Al pulsar PrtSc sale una ventana para seleccionar donde guardar la captura de pantalla)

Para generar un log de acceso sobre nuestra aplicación web, vamos a modificar el fichero **/var/lib/tomcat9/conf/server.xml**. En este fichero, vamos a añadir las siguientes configuraciones, las cuales por defecto ya tenía integradas en mi fichero.



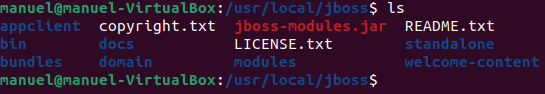
Estos logs se van a generar en el directorio **/var/lib/tomcat9/logs** con el formato especificado en el archivo de configuración. Podemos ver que ha generado un log para el dia 21-11 y otro para el dia 22-11, ya que son los 2 días en los que he levantado mi aplicación web. En el fichero del dia 21 debe haber contenido, ya que accedí a la app, y en el del 22 no debe haber nada, ya que hoy no he accedido aún.





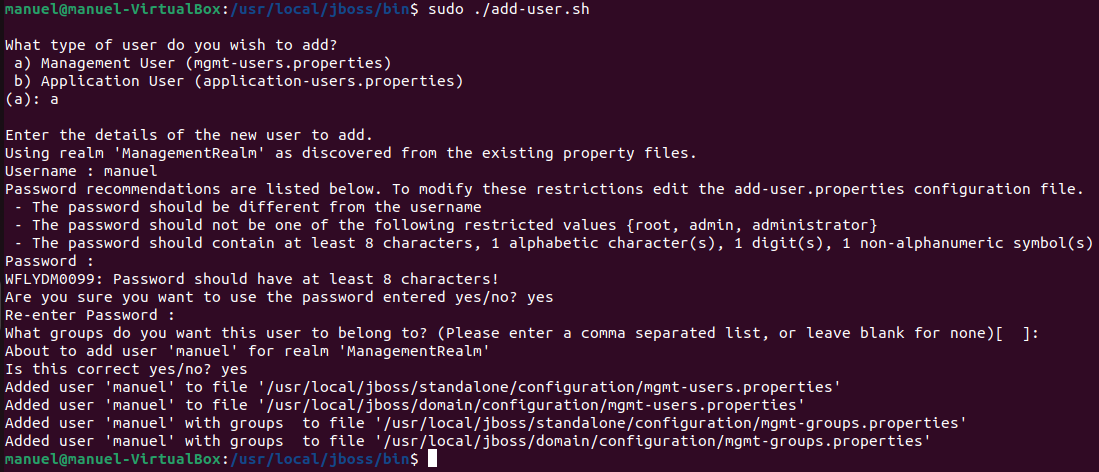
* Descargar e instalar la última versión de JBoss Application Server (no instalarlo con Synaptic, apt-get o similar).

Vamos a descargar la última versión estable de JBoss desde la web de [JBoss](https://jbossas.jboss.org/downloads). En mi caso, he instalado WildFly ya que mi versión de Java no soporta la última versión de JBoss, y WildFly es la nueva versión de la aplicación. Esto nos descargará un zip con la aplicación contenida. Vamos a descomprimirlo y a mover el directorio resultante a nuestra carpeta **/usr/local/jboss**. El contenido del directorio debe ser este:



* Crear usuario para JBoss.

Para crear un usuario para JBoss, vamos a acceder al directorio **/usr/local/jboss/bin** y vamos a ejecutar el script **add-user.sh**.

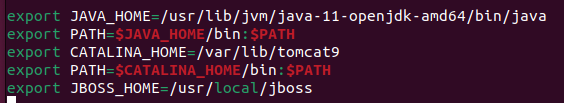


* Establecer las variables de entorno correspondientes para Java y JBoss.

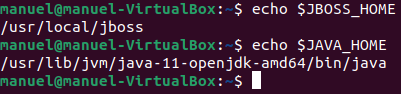
Para modificar las variables de entorno, vamos a modificar el fichero **bashrc.**



Al final de este archivo, vamos a introducir las siguientes líneas para definir nuestras variables de entorno:

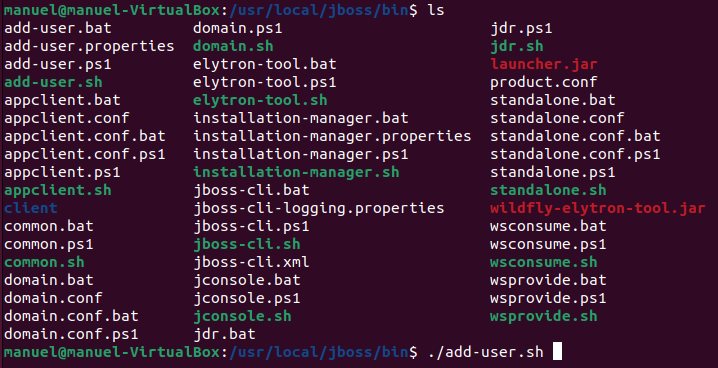


Al hacer **echo** para imprimir el valor contenido en las variables de entorno, se nos debe imprimir la ruta de instalación de estos servicios.

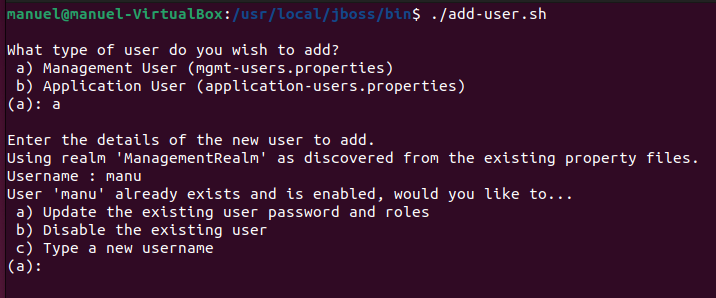


* Cambiar la contraseña del administrador de JBoss.

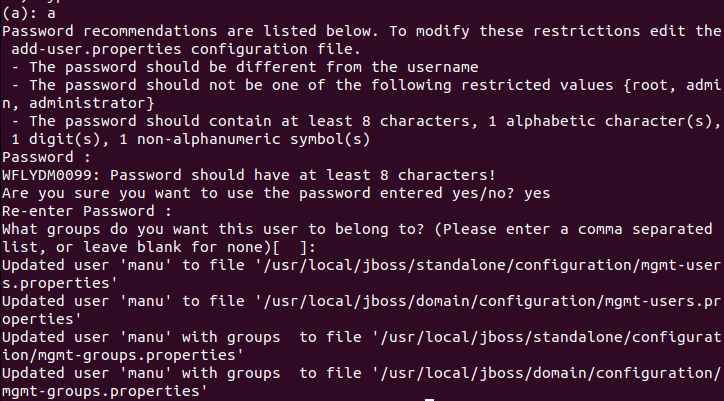
Para cambiar la contraseña de un usuario en JBoss, vamos a acceder a la carpeta donde lo tenemos instalado, y vamos a ejecutar el script **add-user.sh**.



En este script, vamos a “añadir un usuario” y vamos a introducir el nombre del usuario administrador. Nos dirá que el usuario ya existe, y nos dará la opción de modificar su contraseña.

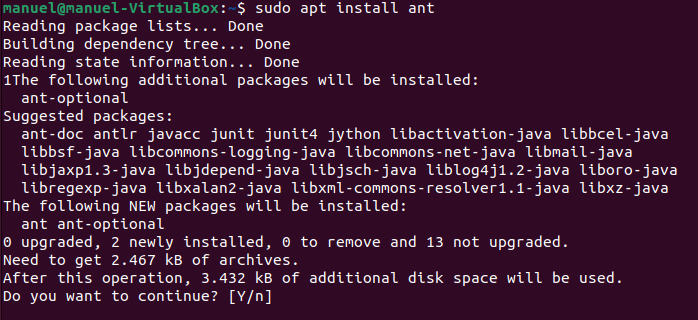


Introducimos la nueva contraseña deseada y el usuario admin se habrá actualizado.

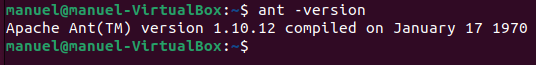


* Instala y configura Ant.

Para instalar Ant, vamos a ejecutar desde bash el comando **sudo apt install ant**.



Para verificar la versión que hemos instalado y que Ant se ha instalado correctamente, usamos el comando **ant -version**.



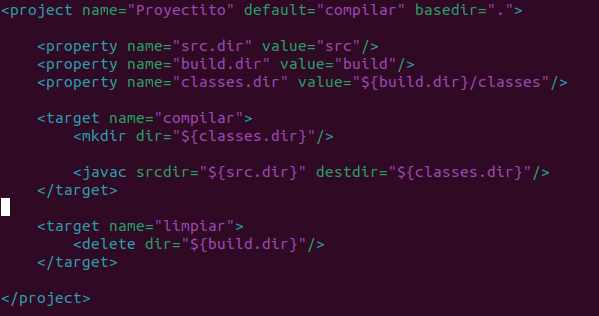
Vamos a añadir Ant a las variables de entorno. Para ello, nuevamente vamos a modificar el fichero **~/.bashrc** y vamos a añadir estas líneas al final del fichero.



Podemos verificar que se ha aplicado correctamente al imprimir el valor de la variable de entorno por consola.



Vamos a crear un pequeño **build.xml** para probar que Ant funciona correctamente.



Vamos a ejecutar Ant sobre el directorio donde hemos creado el fichero **build.xml** y comprobar que devuelve “BUILD SUCCESSFUL”.

