

Cómo instalar Java SDK en Ubuntu 8.04 (Hardy Heron)

Juan-Antonio Martínez-Comeche
Manuel Blázquez Ochando

- **Problema:** En muchas ocasiones se necesita tener instalada una Máquina Virtual Java para poder ejecutar aplicaciones o software basados en el lenguaje de programación Java. La Máquina Virtual Java (JVM) es un programa independiente de la plataforma (Windows, Linux, Mac OS...) que permite ejecutar instrucciones escritas en el código binario de Java (el Java bytecode). Si, con el tiempo, deseamos también modificar el código de esas aplicaciones para adaptarlas a nuestras necesidades específicas, o bien queremos desarrollar programas en Java, una opción aceptable consiste en instalar Java SDK (Java Software Development Kit), pues incluye, además del JRE (Java Runtime Environment) que permite que corran las aplicaciones o software desarrollados en Java, un grupo de utilidades que son de gran ayuda a la hora de programar. Sin embargo, resulta conveniente configurar correctamente algunas variables de entorno para lograr que este programa de código libre cedido por Sun sea perfectamente operativo.
- **Solución:** El procedimiento que figura a continuación para instalar la Máquina Virtual Java de Sun (Java SDK versión 6) en Ubuntu 8.04 (Hardy Heron), logrando su correcto funcionamiento, se basa esencialmente en las siguientes páginas:
 - <http://www.ubuntu-es.org/node/55/>
 - <http://www.kevinboone.com/classpath.html/>
 - <http://www.guia-ubuntu.org/index.php?title=java>
- **Otra Solución:** En este mismo sitio figura otro procedimiento de instalación y configuración de la Máquina Virtual Java de Sun, consistente en descargarse la última versión disponible en el sitio web de Sun. Si prefiere instalar la última versión de JDK, puede consultar http://www.comeche.es/documents/Software/como_instalar_java_sdk_ubuntu_910.pdf



Proceso de Instalación:

- **Paso 1:** Se puede acudir directamente al sitio web de Sun, <http://java.sun.com/javase/downloads/index.jsp> para descargarnos la última versión disponible de “Java SE Development Kit (JDK)”, pero resulta mucho más sencillo y a la postre más seguro descargarnos la versión de Sun disponible en los repositorios de Ubuntu. Por ejemplo, a la hora de escribir estas líneas la última actualización de Java JDK en las páginas de Sun es la 6.10, mientras que en los repositorios de Ubuntu es la 6.7. En consecuencia, basta ejecutar el programa gestor de paquetes de Ubuntu 8.04 (Sistema → Administración → Synaptic), y buscar por “sun”. De esta forma nos aseguramos que la versión de Java que instalaremos será la de Sun Microsystems, y no otra.
- **Paso 2:** Marcar el paquete “sun-java-jdk” para instalar. Para ello, basta hacer doble clic sobre él. Aparecerá una ventana con otros paquetes que deberán instalarse simultáneamente. En el caso de “sun-java-jdk”, por ejemplo, aparecerán “libc6”, “libx11-6”, “sun-java6-bin”, “debconf”, “sun-java6-demo”, “sun-java6-doc” y “sun-java6-source”. Hacer clic en el botón “Marcar” para aceptar la instalación de estos paquetes adicionales. A continuación, pulsar el botón “Aplicar” para proceder a la instalación de todos ellos.
- **Paso 3:** En ocasiones se tiene alguna versión anterior de java jdk o de java jre instalada previamente. Para asegurarnos de que la versión que se utilizará de la Máquina Virtual Java es la instalada en el paso anterior, conviene instalar el paquete “galternatives” y comprobarlo. Debemos cercionarnos de que el paquete “galternatives” está disponible en los repositorios que hayamos elegido. Para ello, abrimos Synaptic y buscamos por “galternatives”. Si no apareciese el paquete, marcar en Configuración → Repositorios → Software de terceros los repositorios

necesarios y pulsar en el botón “Recargar”. A continuación marcamos el paquete “galternatives” para su instalación. No hay en este caso dependencias. Hacer clic en “Aplicar”. También se puede instalar desde consola tecleando “sudo apt-get install galternatives”.

- **Paso 4:** Lanzamos el programa galternatives. Para ello, en una consola basta teclear “sudo galternatives”. Saldrá una ventana con el programa.
- **Paso 5:** En el menú del margen izquierdo, deslizar el scroll hasta localizar y seleccionar la línea “java”. En el margen derecho deberá estar marcada la ruta del programa recién instalado, en nuestro caso, “/usr/lib/jvm/java-6-sun/jre/bin/java”. En caso contrario, efectuar las correcciones necesarias para que la ruta corresponda a la versión recién instalada. Salir del programa (Fichero-->Salir).
- **Paso 6:** Comprobar que estamos usando la versión correcta. Para ello, tecleamos en un terminal “sudo java -version”. Debemos obtener el mensaje correspondiente a la versión recién instalada; en nuestro caso, “java version 1.6.0_07”.
- **Paso 7:** La variable de entorno JAVA_HOME nunca es configurada por ningún JDK. Hay muchos programas que, sin embargo, necesitan tener configurada la variable JAVA_HOME para poder ser instalados correctamente. En consecuencia, conviene configurar dicha variable adecuadamente. Para ello debemos anotar en qué directorio se ha instalado la Máquina Virtual Java, esto es, en el que se halla el Java SDK. En nuestro caso, “/usr/lib/jvm/java-6-sun”. Ésta será la ruta que deberá poseer dicha variable.
- **Paso 8:** La variable de entorno CLASSPATH no es tampoco configurada durante la instalación previamente realizada. Sin embargo, dicha variable debe contener la ruta de los directorios en los que el compilador de Java (javac.exe) podrá localizar los archivos .java (código fuente de programas Java) para compilar, así como cualquier otro archivo class (programa ya compilado) referenciado o empleado en el programa fuente. De ahí que en CLASSPATH deban figurar también las rutas de las librerías que se empleen en nuestros programas. Si se desean señalar varias rutas, deberemos separarlas en Linux mediante ':' (dos puntos). Además, conviene terminar las rutas de CLASSPATH siempre con un punto '.' (punto), pues ello nos permitirá compilar y/o ejecutar cualquier programa en Java situado en el directorio donde trabajemos en ese momento. En consecuencia, en nuestro caso, incluiremos en este ejemplo tres rutas en nuestra variable CLASSPATH:
 - El directorio donde se hallan las librerías de Java de la Máquina Virtual recién instalada: /usr/lib/jvm/java-6-sun/lib
 - El directorio donde vayamos a incluir los programas Java que desarrollemos en el futuro, por ejemplo: /home/juan
 - El directorio actual, representado mediante un punto

En consecuencia, CLASSPATH deberá figurar en nuestro caso de la siguiente manera: “/usr/lib/jvm/java-6-sun/lib:/home/juan:.”

- **Paso 9:** Para el establecimiento de estas variables de entorno, basta modificar el archivo “.bashrc”, localizado y oculto en el directorio raíz de cada usuario. Por ejemplo: /home/juan/.bashrc, donde “juan” es el nombre de nuestra carpeta de usuario. Abrimos con gedit el archivo “.bashrc” tecleando “sudo gedit /home/juan/.bashrc”.
- **Paso 10:** Al final del archivo añadimos las dos siguientes líneas de texto:
export JAVA_HOME="/usr/lib/jvm/java-6-sun"
export CLASSPATH="/usr/lib/jvm/java-6-sun/lib:/home/juan:.”
- **Paso 11:** Una vez añadidas estas líneas al final del archivo, guardamos los cambios. De esta manera, hemos instalado y configurado la Máquina Virtual Java adecuadamente.
- **Paso 12:** Es importante reiniciar el sistema para que todas las variables de entorno adquieran los valores recién introducidos. Para comprobar que todo está correcto, abrimos de nuevo un terminal y tecleamos “echo \$JAVA_HOME”. Deberá darnos la ruta introducida anteriormente. De igual forma, si tecleamos “echo \$CLASSPATH” deberá mostrar las rutas tecleadas previamente.