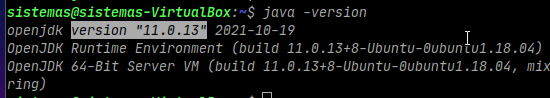
Vamos a instalar el entorno de net beans que vimos ayer, tenemos los pasos en la web.

1º → Instalamos openJDK en su version 11. Hya varias formas:

sudo apt install default-jdk

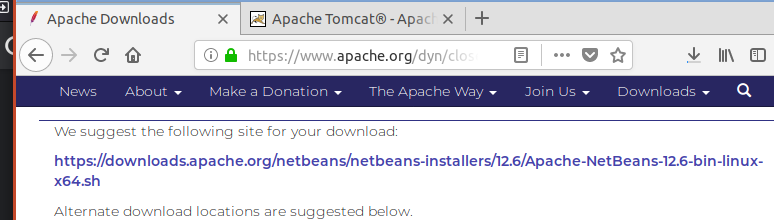
sudo apt install openjdk-11

2º → Comprobamos que la versión que se ha instalado es efectivamente la 11. Es muy importante porque nos pueden salir problemas de compatibilidad.



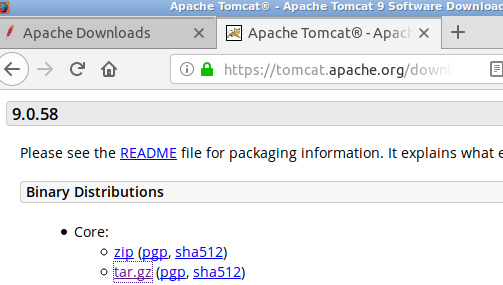
Adicionalmente, vamos a crear un GRUPO tomcat, y un usuario TOMCAT (aunque en teoría se instalan solos, pero bueno, por si acaso). Los usaremos más adelante. (groupadd y useradd cosa)

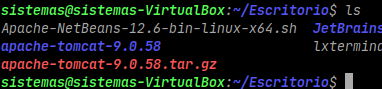
3º → Tenemos que instalar NetBeans. Lo vamos a instalar en su última versión, la 12,6, es la que uso el profe el otro día y le funcionaba. Recemos para que no sea la 12,5 o anterior.



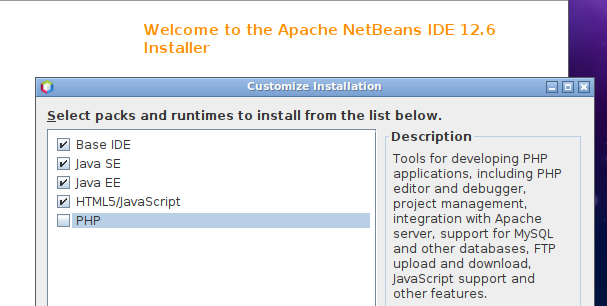
Descarga un .sh

4º → Descargamos también Tomcat en la versión 9 de su página oficial.





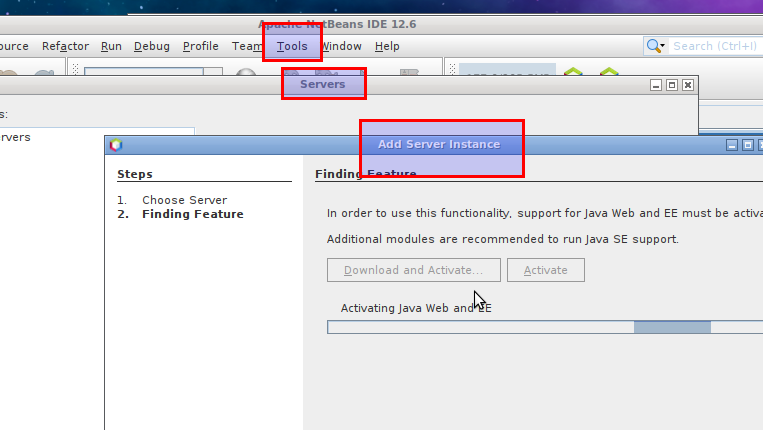
5º → una vez descargado el script, le damos permisos con chmod para poder ejecutarlo

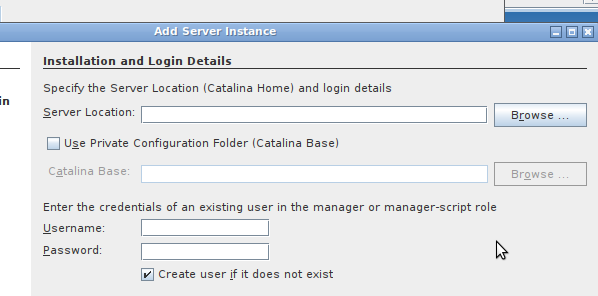


Se nos abre este modo gráfico. Instalamos todo menos php porque no lo vamos a usar. Siguiente siguiente y se instala.

Una vez termina la instalación ya tenemos un ejecutable en el escritorio, podemos abrirlo y enlazar este netbeans al tomcat que hemos descargado.

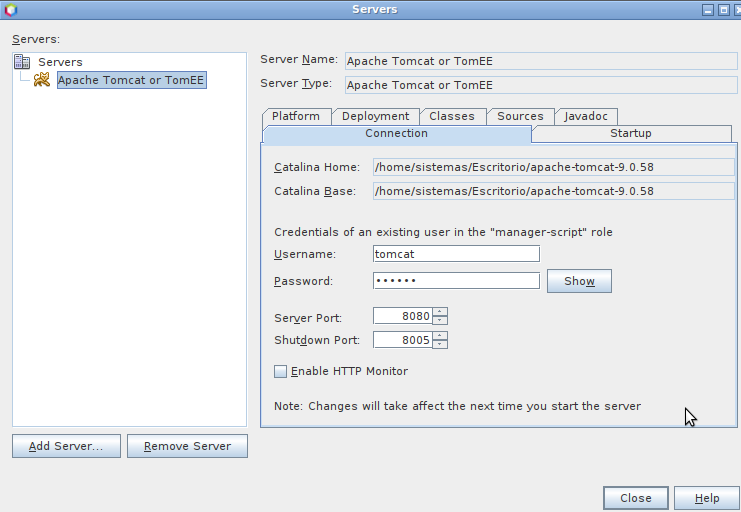
6º → Ahora tenemos que VINCULAR el netbeans a tomcat. Para ellos nos vamos a tools, en la parte de servers, y le damos a añadir nuevo servidor. Tendremos que descargar un pluggin.





Aquí le tenemos que indicar la ruta donde tenemos el targz de tomcat descomprimido, es donde estará nuestro servidor.

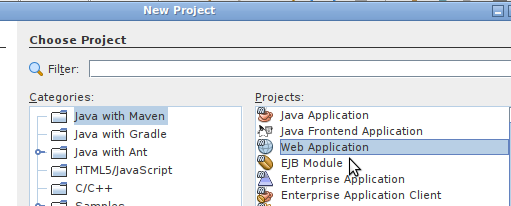
El usuario tiene que ser el usuario tomcat que instalamos previamente.



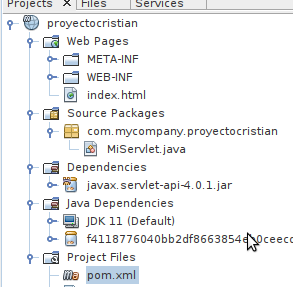
7º → Vamos a crear nuestro primer proyecto de Java con Maven. Maven es un gestor de dependencias, lo que hace es instalar cosas que NO están en el core de Java, unas clases.

Seleccionamos un **proyecto de tipo web.**

Esto lo que hace es cargar una plantilla, y en base a lo que elija, pues trabaja con eso. Esta plantilla NO ESTA actualizada, nada más crearlo ya tenemos errores, hay que tocar cosas.



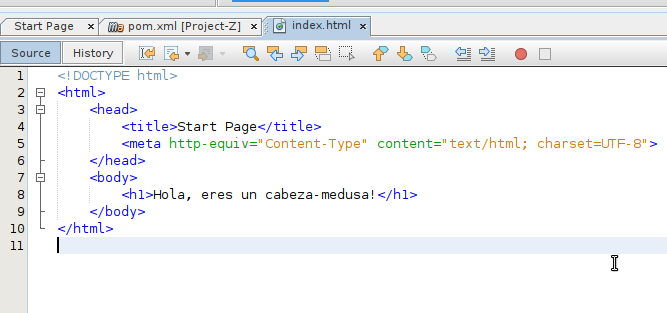
La plantilla es esto básicamente, con un index y todo.



Nos pedirá descargar cosas, y luego editaremos el archivo POM.XML con lo que tenemos en el aula.



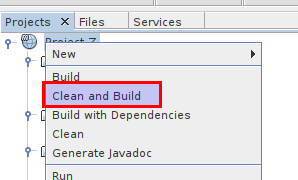
Lo que estamos haciendo es cargar las dependencias de la API 4, meter un ServLet.



Una vez tenemos puesto el pom, vamos a nuestro index. Y le damos al PLAY. Si funciona, se cargará el localhost y se cargará.



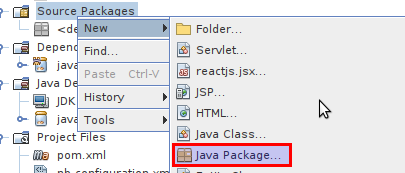
Puede que a veces no funcione, aunque todo este bien, en ese caso, te vas al proyecto y le haces esto:



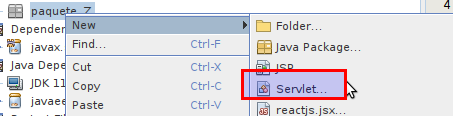
Esto limpia la compilación y la ejecuta de 0, en muchas ocasiones funcionara.

Ahora es cuando vamos a meter los servlet. Un servlet es como el $request de PHP, coge información y muestra información, la usamos internamente para cosas.

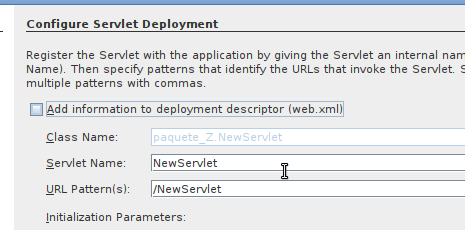
Borramos el que hay y creamos un PAQUETE nuevo, este es el paquete que contendrá nuestro servlet, pero además tendrá otras cosas dentro que configuremos.



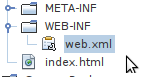
Ahora que tenemos el paquete creado, le metemos un servlet

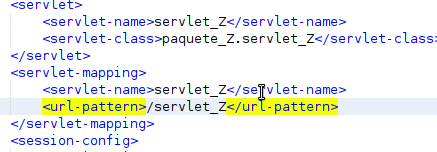


IMPORTANTE marcar la casilla de ADD, esta casilla le da opciones extras importantes para un correcto funcionamiento, como el nombre del paquete, del servlet, la ruta donde esta guardado, etc.



En el momento en que creamos el servlet, se nos crea un xml en la página web





Esto es lo que contiene. Básicamente es que, cuando yo haga peticiones, se cargaran en este servlet.

La configuración de este archivo es muy importante.

Servlet-name tiene que coincidir ARRIBA con abajo, OJO, no tiene que ser el mismo que el servlet físico. Es donde lo voy a guardar. Si no coinciden, nunca va a funcionar.

Servlet-class es INMUTABLE → Es mi servlet físico y mi paquete físico. El servlet name hace referencia a ese servlet físico. Si esto no es idéntico a lo que tengo creado, nunca va a funcionar.

Por último el maping, esto es, en la url, cuando yo ponga miweb/servlet\_z se hace una llamada al servlet name, ese servlet name mira arriba si coincide, y llama al físico.

El mapeado puede ser como me dé la gana, pero apunta al name, el name comprueba si es idéntico, y si lo es, apunta al físico, y se carga.

Entonces yo puedo escribir en la web cosas y se me guardan, ya sea por la URL, por un formulario…

Mañana jugamos con ello.