

Java Web Server

Tabla de Contenidos

3. Java Web Server.....	2
3.1 Conceptos generales	2
3.2 Elección de servidor Web con capacidad para servlets.....	3
3.3 Instalación de Tomcat 5.....	4
3.3.1 Comienza la instalación.....	5
3.3.2 Comprobación de funcionamiento.....	8
3.4 Administración y Configuración.....	10
3.4.1 Status.....	10
3.4.2 Administración de archivos	11
3.4.3 Manager	12
3.5 Pool de Conexiones.....	13



3. Java Web Server

3.1 Conceptos Generales

- *Java WebServer*: Es un servidor WEB basado en tecnología JAVA. Cuando Ud. necesite hostear (es decir, mostrar al mundo su página web) o realizar aplicaciones en Intranet (es decir en su red privada o LAN) , Ud. necesitará un Software que le permita justamente hacer esta tarea. Para tal fin, existen diferentes servidores de WEB, y Javaweb Server es uno de ellos.

- *Java Servlets*: Son las repuestas de la tecnología Java a la programación CGI. Son programas que se ejecutan en un servidor web. Basándose en el lenguaje Java, los servidores Web de Java utilizan programas denominados Java Servlets, lo que te permite expandir el servidor Web de forma muy fácil. Los Servlets sustituyen a los antiguos scripts CGI, y ofrecen a los desarrolladores un entorno que podrán usar en cualquier plataforma sin un esfuerzo adicional. Un servlet es de clase Java y normalmente acepta peticiones de tipo HTMLy genera una salida, generalmente de tipo HTML, WML o XML. Los servlet que conforman una aplicación web se ejecutan en un servidor de aplicaciones web (o WebServer)

- *Pool de Conexiones*: se denominan así a las modificaciones que deben hacerse al archivo de configuración del Java Web Server con el fin de quetenga conexión con las diferentes bases de datos a utilizar.

- *JDBC*: La conectividad a la base de datos Java (JDBC : Java Data Base Connectivity) es una interfaz para los desarrolladores Java que escriben los programas que tienen acceso a la información almacenada en bases de datos, plantillas de calculos, etc. Es utilizada para conectar un programa del usuario a la base de datos en segundo plano sin importar que software de administración se utiliza para su control.



3.2 Elección De Servidor Web Con Capacidad Para Servlets

Lo primero que debemos hacer es instalar un servidor Web que soporte servlets Java, o instalar el paquete Servlet en nuestro servidor Web existente. Si estamos usando un servidor Web actualizado, hay muchas posibilidades de que ya tengamos todo lo que necesitamos. Debemos chequear la documentación de nuestro servidor o ver la última lista de servidores que soportan servlets en <http://java.sun.com/products/servlet/industry.html>. Aunque eventualmente queramos desarrollar en un servidor de calidad comercial, cuando estamos aprendiendo es útil tener un sistema gratuito que podemos instalar en nuestra máquina para propósitos de desarrollo y prueba. Aquí están algunas de las opciones más populares:

- *Apache Tomcat*: Tomcat es la implementación de referencia oficial para las especificaciones Servlet 2.2 y JSP 1.1. Puede ser usado como pequeño servidor para probar páginas JSP y servlets, o puede integrarse en el servidor Web Apache. Tomcat, al igual que el propio Apache es gratuito. Sin embargo, también al igual que Apache (que es muy rápido, de gran rendimiento, pero un poco difícil de configurar e instalar), Tomcat requiere significativamente más esfuerzo para configurarlo que los motores de servlets comerciales.
- *JavaServer Web Development Kit (JSWDK)*: El JSWDK es la implementación de referencia oficial para las especificaciones Servlet 2.1 y JSP 1.0. Se usaba como pequeño servidor para probar servlets y páginas JSP antes de desarrollar un completo servidor Web que soporta estas tecnologías. Es gratuito y potente, pero necesita un poco de esfuerzo para instalarlo y configurarlo.
- *Allaire Jrun*: JRun es un motor servlet y JSP que puede conectarse dentro de los servidores Netscape Enterprise o FastTrack, de los servidores Web IIS, Microsoft Personal Web Server, viejas versiones de Apache, O'Reilly's WebSite, o StarNine WebSTAR. Hay una versión gratuita limitada a cinco conexiones simultáneas; la versión comercial elimina esta restricción y añade capacidades como una consola de administración remota.
- *New Atlanta's ServletExec*: ServletExec es un rápido motor de servlets y páginas JSP que puede ser conectado dentro de los servidores web más populares para Solaris, Windows, MacOS, HP-UX y Linux. Podemos descargarlo y usarlo de forma gratuita, pero muchas de sus características avanzadas y utilidades de administración estarán desactivadas hasta que compremos una licencia. El nuevo Atlanta también proporciona un depurador de servlets gratuito que funciona con mucho de los IDEs más populares de Java.
- *Gefion's LiteWebServer (LWS)*. LWS es un pequeño servidor Web gratuito que soporta Servlets versión 2.2 y JSP 1.1. También tienen un plug-in gratuito llamado WAICoolRunner que añade soporte para Servlets 2.2 y JSP 1.1 a los servidores Netscape FastTrack y Enterprise. También venden un gran número de Servlets personalizados y componentes JSP en un paquete llamado InstantOnline.

- Sun's Java Web Server. Este servidor está escrito enteramente en Java y fue uno de los primeros servidores en soportar completamente las especificaciones servlet 2.1 y JSP 1.0. Aunque ya no estará bajo desarrollo activo porque Sun se está concentrando en el servidor Netscape/I-Planet, todavía es una opción popular para aprender servlets y JSP..

En este curso aprenderemos la instalación y configuración del Java Web Server TomCat 5

3.3 Instalación De Tomcat 5

Los servidores de aplicaciones están en constante evolución. Esto no sólo se debe a los constantes desarrollos que se realizan para arreglar posibles problemas, sino que con ello se da soporte a las nuevas versiones de las especificaciones de Java.

A continuación podemos observar en el cuadro comparativo, las versiones correspondientes a Servlet/JSP y a Tomcat :

Especificación Servlet/JSP	Versión de Tomcat
2.4/2.0	5.0.12 Beta
2.3/1.2	4.1.27
2.2/1.1	3.3.1a

Aquí aprenderemos la utilización de la última versión disponible de Tomcat y mostraremos algunas de sus características principales a fin de saber instalarlo y configurarlo.

Aprenderemos también a configurar y utilizar un Pool de conexiones y optimizar de este modo los accesos a la base de datos. En este caso usaremos MySQL como base de datos.

Para bajar la ultima versión de Tomcat es necesario visitar la web oficial.

El sitio para su descarga es :

<http://jakarta.apache.org/tomcat/index.html>



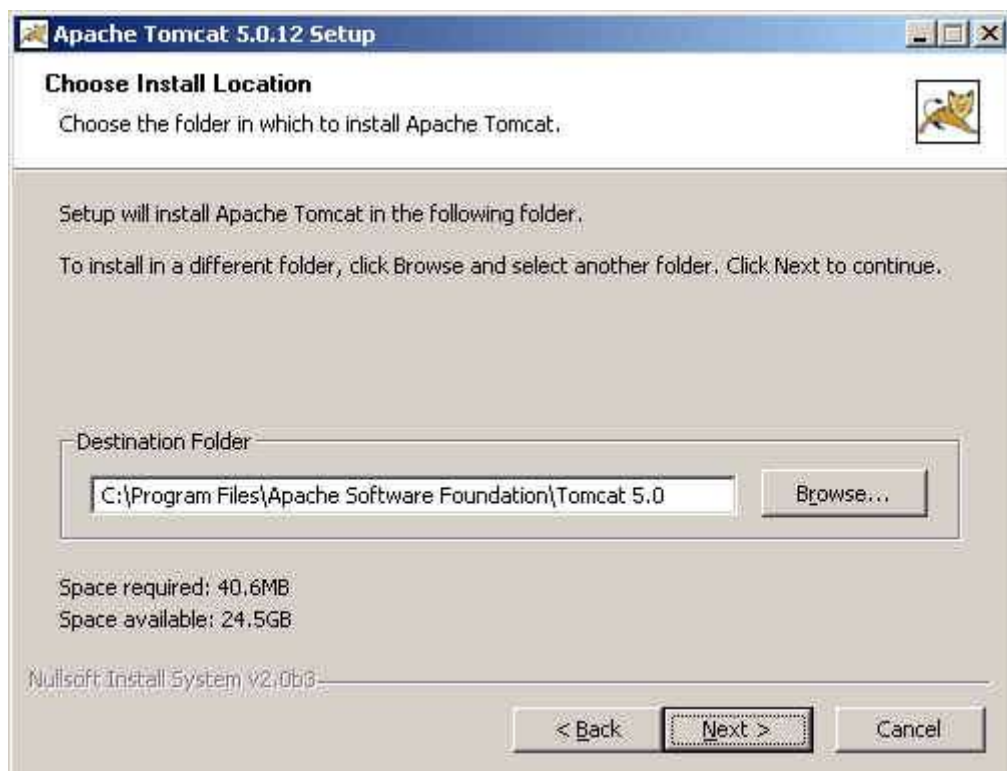
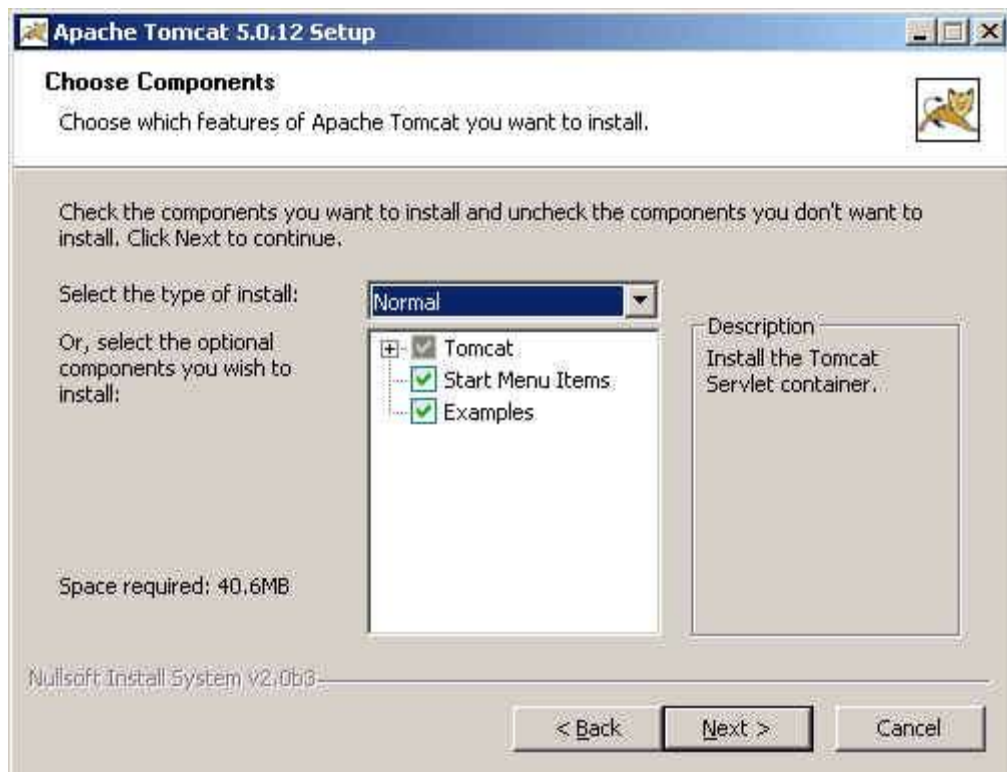
Descargaremos el ejecutable y lo ejecutaremos.

3.3.1 Comienza La Instalación.

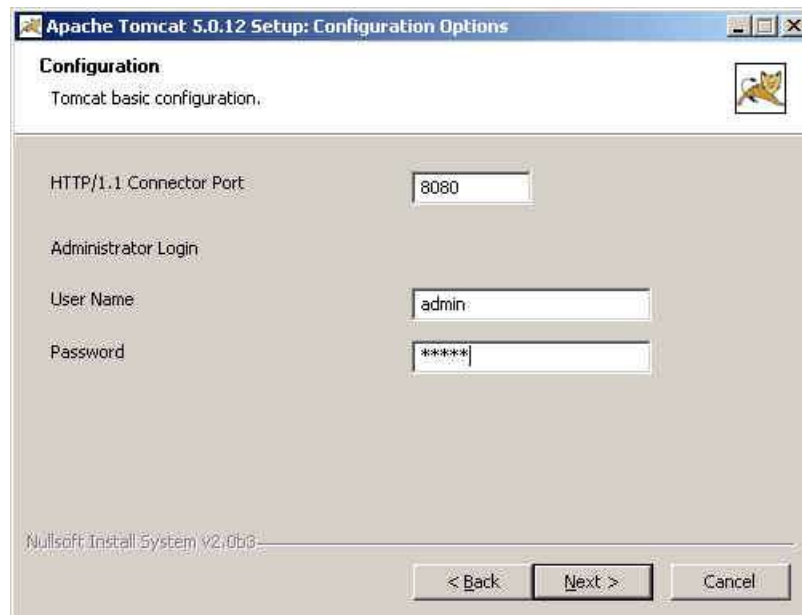
Una vez que hemos bajado el software, ejecutaremos el archivo de instalación.. A continuación podremos observar la siguiente pantalla de bienvenida (welcome) :



Ahora elegiremos los componentes a instalar. En este caso seleccionaremos el tipo de instalación Normal como se puede observar en la pantalla :



Seleccionamos un usuario y password para administración. Este usuario y password deben ser recordados porque serán los utilizados para la futura administración.



Posteriormente elegimos la ruta de la versión de Java a utilizar.



Allí hacemos clic en el botón de Instalar y comienza la instalación de Tomcat 5. Al finalizar, nos mostrará la pantalla que se muestra a continuación :

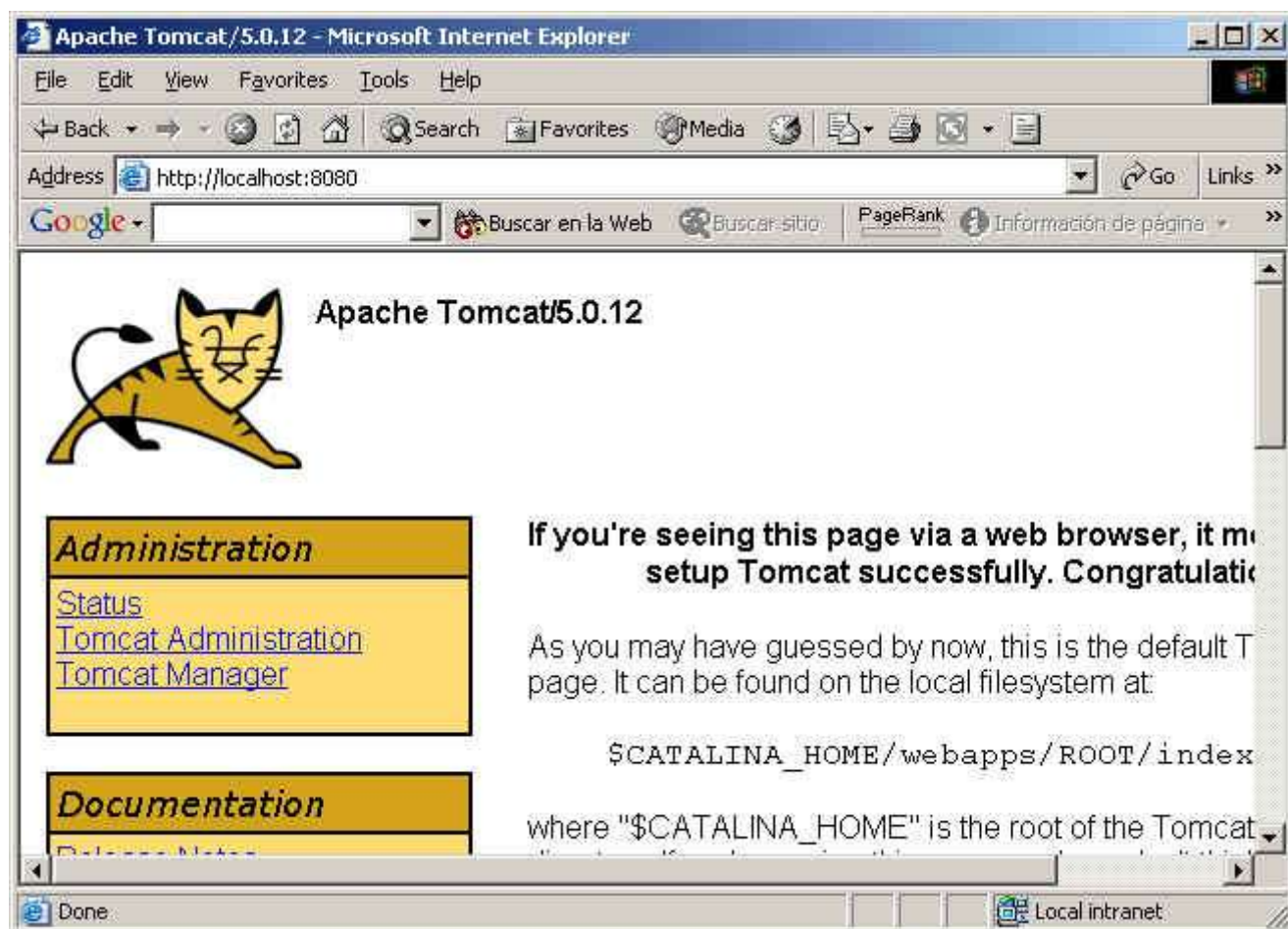


Como las cajas de verificación de “Run Apache Tomcat” y “Show Readme” están marcados, nos mostrará el archivo de lectura previo (readme) y comenzará a correr Tomcat 5.

3.3.2 Comprobación De Funcionamiento

Abrimos el navegador y nos comunicaremos a través de “localhost” mediante el puerto 8080 como se muestra :





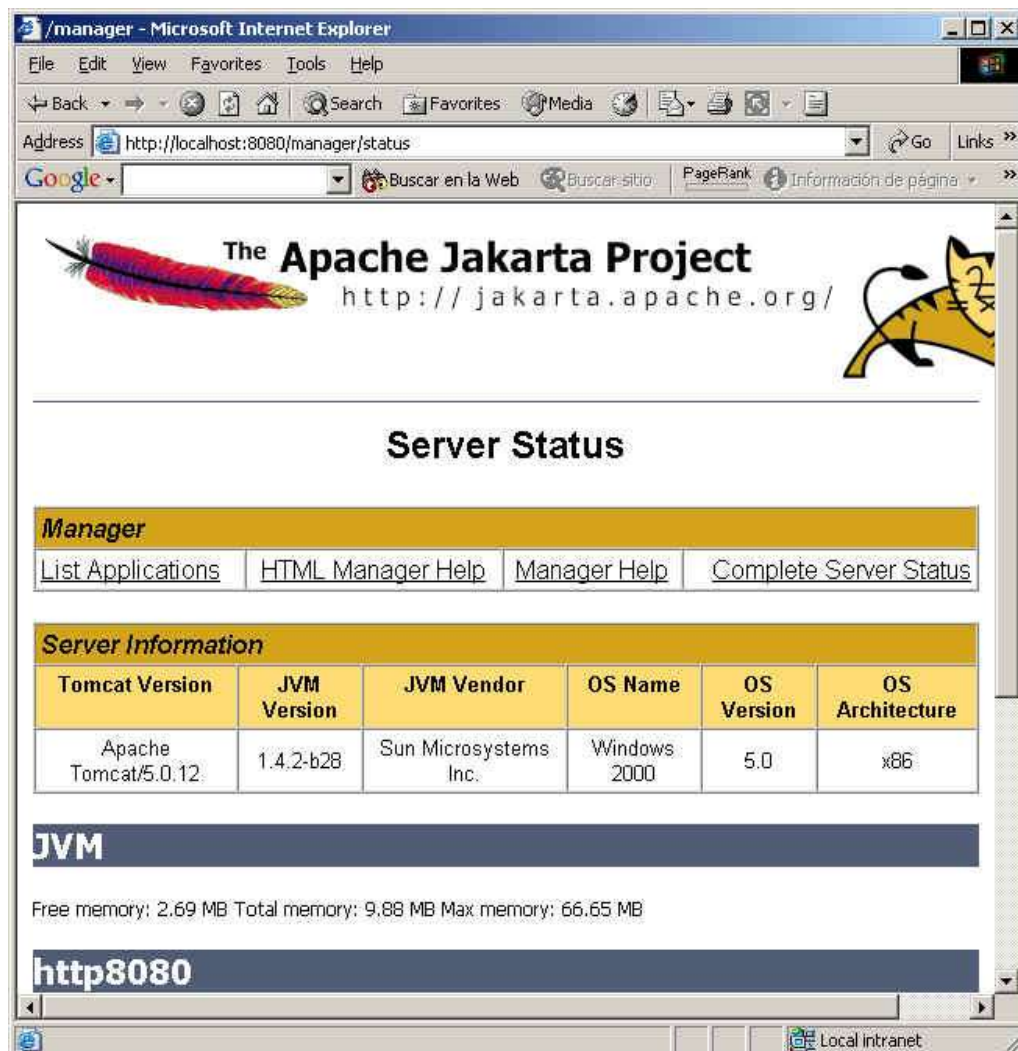
Si Ud. Puede ver esta pantalla es porque Tomcat 5 se ha instalado correctamente.



3.4 Administración Y Configuración

3.4.1 Status

En la parte de administración tenemos unos enlaces para ver comprobar el estado de Tomcat. Para ello, como lo muestra la pantalla que a continuación puede verse, debemos conectarnos con un navegador al siguiente link : <http://localhost:8080/manager/status>



The screenshot shows a Microsoft Internet Explorer window titled "/manager - Microsoft Internet Explorer". The address bar displays "http://localhost:8080/manager/status". The page content includes the Apache Jakarta Project logo and a cat illustration. Below the header, the "Server Status" section is visible, containing a "Manager" section with links to "List Applications", "HTML Manager Help", "Manager Help", and "Complete Server Status". The "Server Information" section contains a table with the following data:

Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture
Apache Tomcat/5.0.12	1.4.2-b28	Sun Microsystems Inc.	Windows 2000	5.0	x86

Below the table, the "JVM" section shows memory usage: "Free memory: 2.69 MB Total memory: 9.88 MB Max memory: 66.65 MB". The "http8080" section is also visible at the bottom.

3.4.2 Administración De Archivos

Para acceder a la pantalla de administración, deberemos introducir el siguiente link en el navegador:

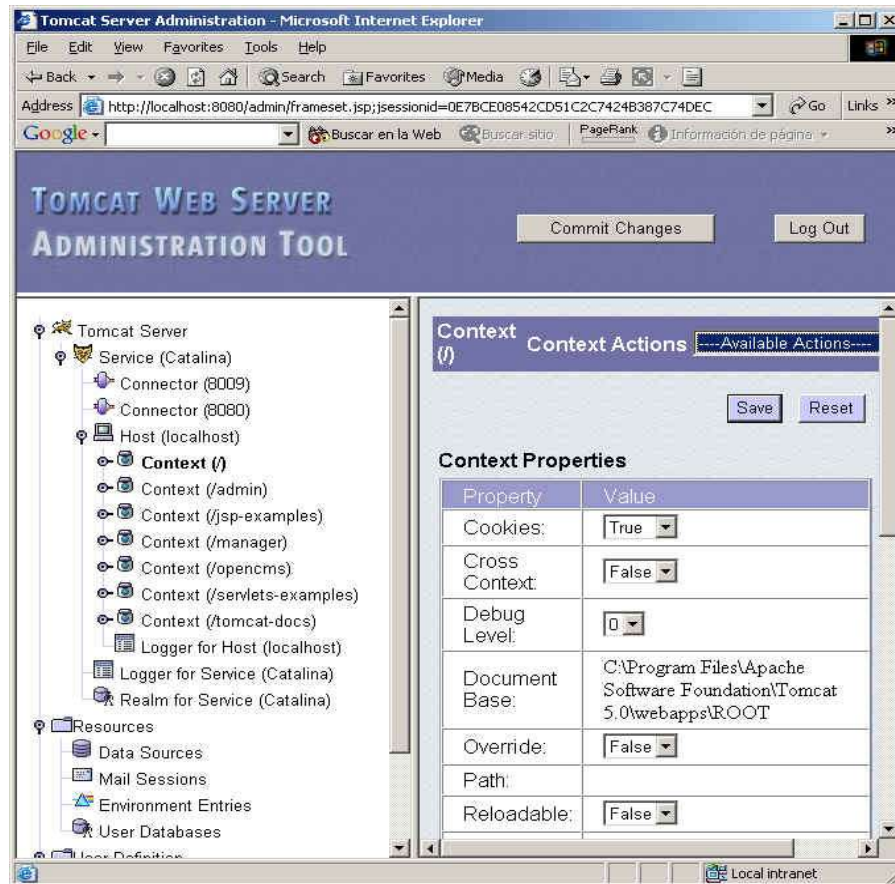
<http://localhost:8080/admin>



Desde aquí podremos administrar los ficheros de configuración de un modo gráfico.

Deberemos introducir el usuario administrador y la password que hemos configurado cuando instalamos Tomcat 5.

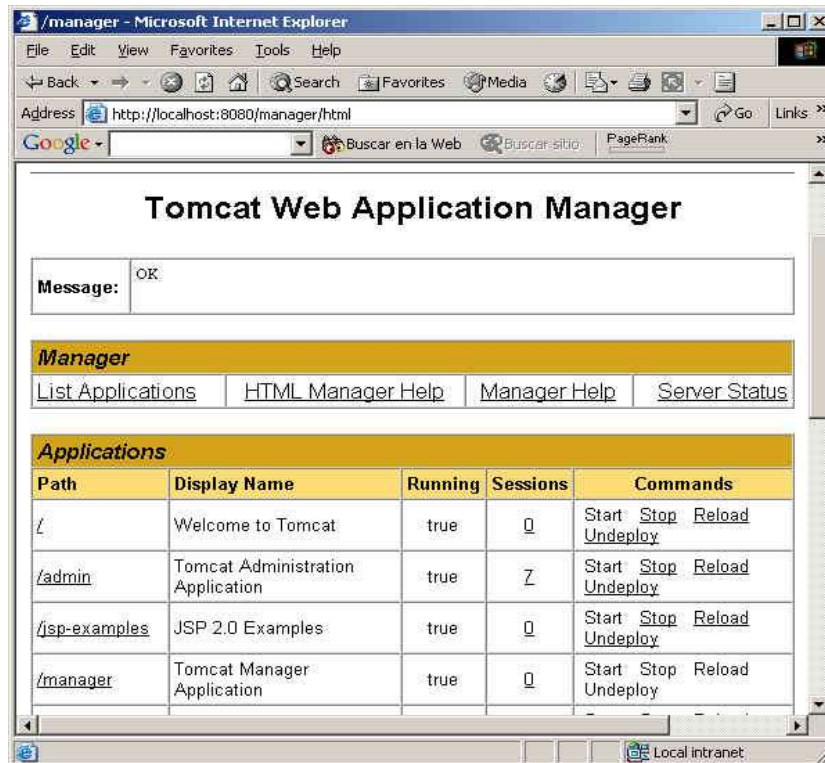
Una vez introducido el usuario y su contraseña, TomCat 5 nos habilitará la siguiente pantalla para trabajar con la administración de los archivos :



3.4.3 Manager

Desde aquí podemos desplegar, detener y recargar las aplicaciones Web, además de obtener información sobre su estado actual, como podemos observar en la pantalla siguiente :





3.5 Pool De Conexiones

Una vez realizada la configuración de Tomcat 5 como hemos visto en los pasos anteriores, realizaremos la configuración del Pool de conexiones a la base de datos de MySQL.

Para realizar dicha tarea, debemos realizar los siguientes pasos :

- 1 - Registrar el Pool en el fichero *server.xml*
- 2 - Dejar a mano las clases del Driver
- 3 - Crear nuestro servlet
- 4 - Hacer accesible el Pool al servlet
- 5 - Crear la tabla en base de datos

Ahora veremos paso a paso como configurar cada uno de estos servicios :

1 - Registrar el pool en el archivo server.xml

Lo primero que hacemos, es modificar el fichero *server.xml* como se muestra en el ejemplo. El archivo de configuración se encuentra en *C:\Archivos de programa\Apache Software Foundation\Tomcat 5.0\conf\server.xml*

```
<Context path="/tomcat5" docBase="tomcat5" debug="0" reloadable="true" >
<ResourceParams name="jdbc/tutoriales">
<parameter><name>username</name>
<value>root</value>
</parameter>

<parameter>
<name>password</name>
<value></value>
</parameter>

<parameter>
<name>driverClassName</name>
<value>com.mysql.jdbc.Driver</value>
</parameter>

<parameter>
<name>url</name>
<value>jdbc:mysql://localhost/tutoriales</value>
</parameter>

</ResourceParams>
</Context>
```

2 - Fichero .Jar del Driver JDBC

Ahora introducimos el fichero .Jar del Driver Jdbc de MySQL en el directorio adecuado. *C:\Archivos de Programa\Apache Software Foundation\Tomcat 5.0\common\lib\mysql-connector-java-3.0.7-stable-bin.jar*

3- Crear el Servlet

Posteriormente debemos crear nuestra aplicación. Creamos el Servlet (*poolBaseDatos.java*). Para ello hemos creado un servlet java.

El código anexo al temario es básico, solo faltaría realizar un tratamiento mejorado de la gestión de errores.





Se incluye el extracto del código como documentación anexa al tema

4 - Hacer accesible el Pool al servlet

Ahora, necesitamos un fichero descriptor: *web.xml*, que deberá configurarse como se muestra a continuación :

```
<?xml version="1.0" encoding="UTF-8"?>

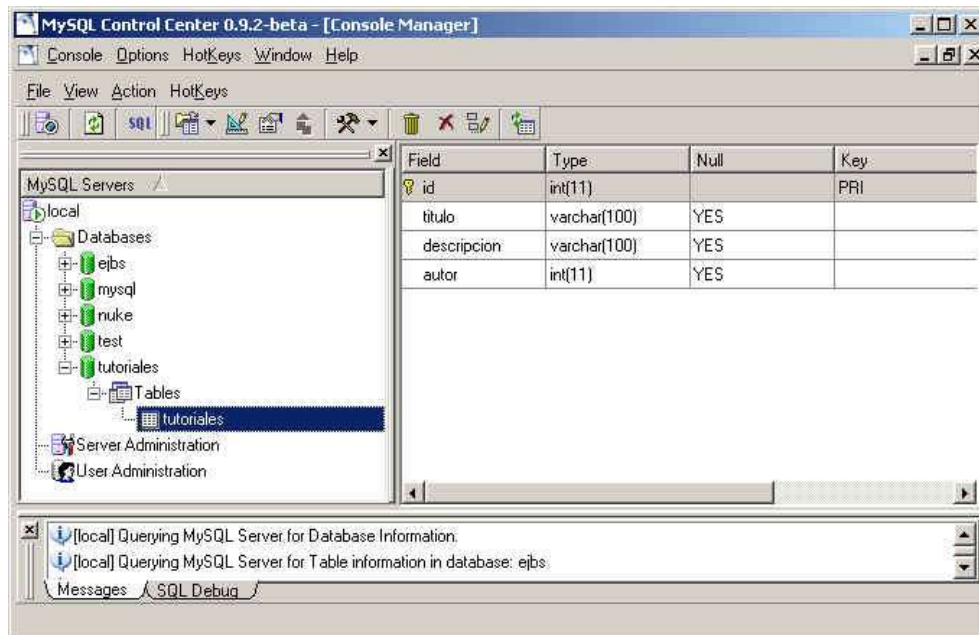
<!DOCTYPE web-app
PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>

<servlet>
  <servlet-name>poolBaseDatos</servlet-name>
  <servlet-class>poolBaseDatos</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>poolBaseDatos</servlet-name>
  <url-pattern>/servlet/poolBaseDatos</url-pattern>
</servlet-mapping>
<session-config>
  <session-timeout>30</session-timeout>
</session-config>
<welcome-file-list>
  <welcome-file>index.jsp</welcome-file>
  <welcome-file>index.html</welcome-file>
  <welcome-file>index.htm</welcome-file>
</welcome-file-list>
<resource-ref>
  <res-ref-name>jdbc/tutoriales</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
</web-app>
```


5 - Crear la tabla en base de datos

A continuación creamos una tabla en la base de datos de MySQL que vamos a utilizar como ejemplo, en la cual damos de alta unos cuantos registros.



A continuación sólo nos hace falta desplegar la aplicación.

Para tal fin creamos nuestro directorio y copiamos los ficheros en *C:\Archivos de Programa\Apache Software Foundation\Tomcat 5.0\webapps\tomcat5*

```

WEB
| web.xml
|----classes
|     poolBaseDatos.class
|     poolBaseDatos.java
|----lib
    
```

Otra solución consiste en comprimir la estructura anterior en un fichero WAR situándolo en la siguiente ruta, destinada a albergar nuestras aplicaciones: *C:\Archivos de Programa\Apache Software Foundation\Tomcat 5.0\webapps*

Por último solo resta introducir la URL de nuestra aplicación en el navegador y comprobar el resultado final.

