

INSTALACION Y CONFIGURACION DE APACHE TOMCAT Y MYSQL

Teniendo conciencia de la dificultad para instalar y configurar las herramientas que necesitamos para el desarrollo de nuestro proyecto "AMSTROG CARGO", lo cual ha sido un obstáculo para el progreso de la implementación; nos hemos permitido subir este tutorial lo mas explicito posible (ya que en Internet la información que se encuentra es muy diversa y los métodos que allí explican funcionaron pero solo combinando varios de esos pasos).

CONTENIDO:

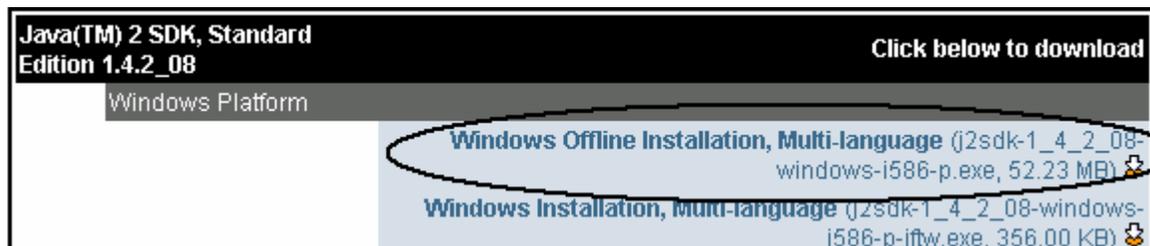
1.INSTALAR JAVA SDK.....	0
2.INSTALAR JAVA WEB SERVICES DEVELOPER PACK.....	3
3.INSTALAR APACHE TOMCAT 4.....	4
4.INSTALAR ENTORNO DE DESARROLLO (NETBEANS IDE).....	8
5.INSTALAR MYSQL PARA LA BASE DE DATOS	12
6.INSTALAR DRIVER MYSQL JDBC PARA JAVA	14
7.CONFIGURACION DE APACHE TOMCAT PARA EJECUTAR LOS SERVLETS. 17	

Este es el proceso que nos funcionó:

1.INSTALAR JAVA SDK

Descarga:

Procederemos a la pagina de java para descargar el jsdk siguiendo todas las instrucciones por defecto



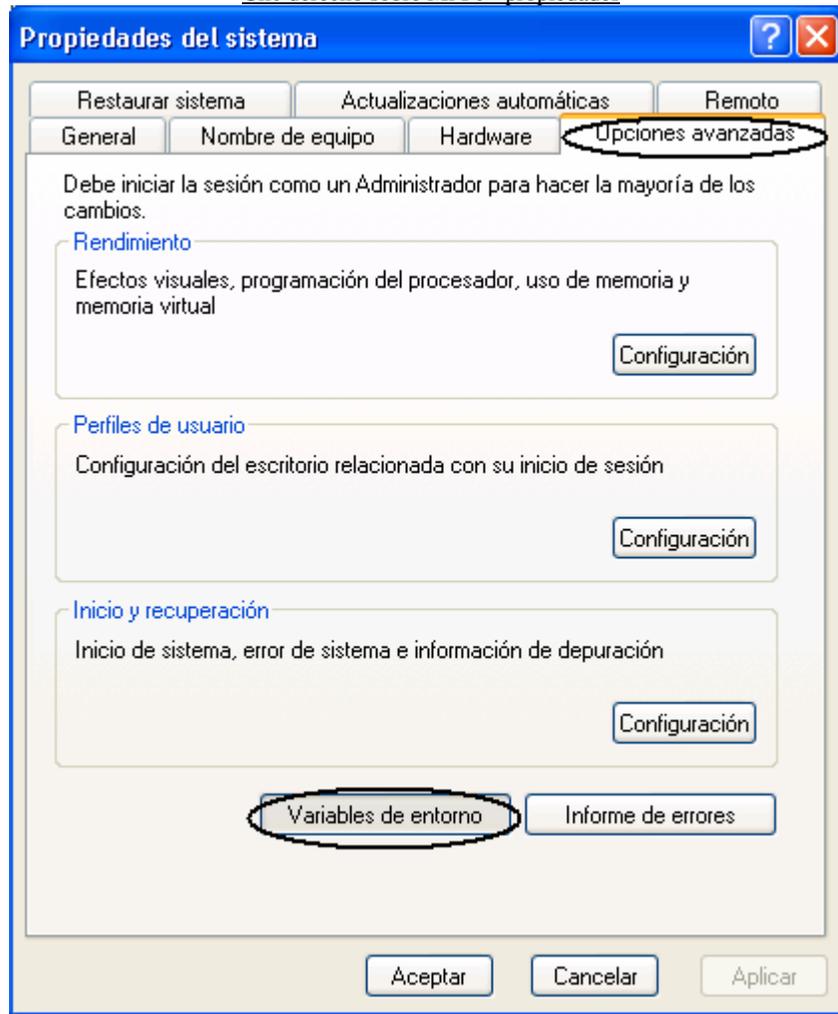
Instalacion:

Luego de haber descargado el archivo de jsdk lo ejecutamos y seguimos los pasos para la instalación por defecto.

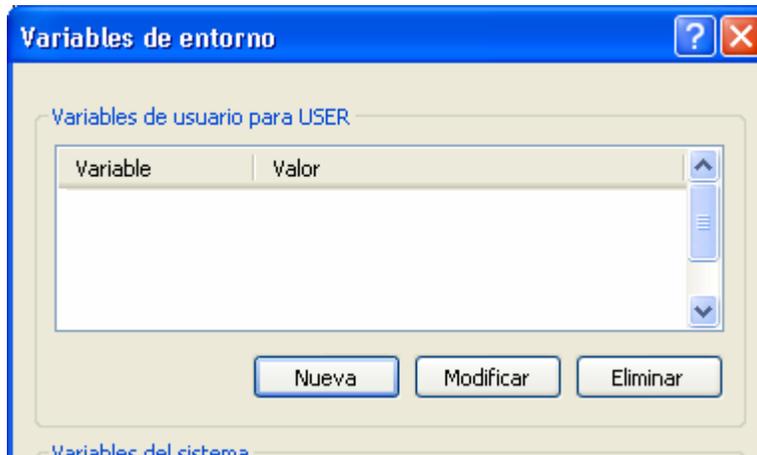
Después de haber terminado la instalación de java debemos establecer manualmente ciertas configuraciones



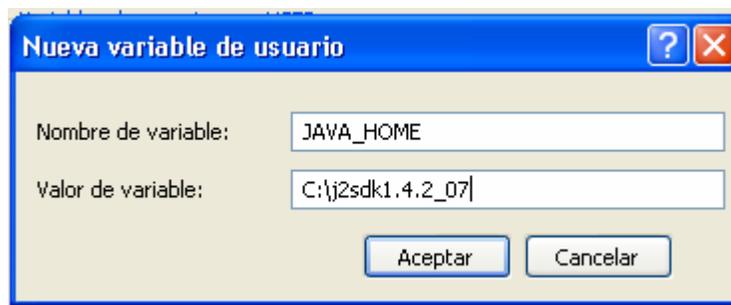
Clic derecho sobre Mi Pc - propiedades



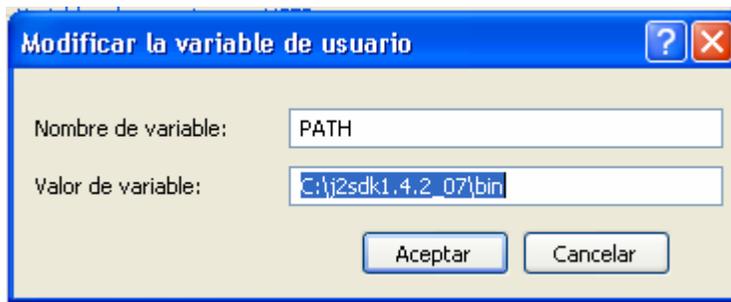
Opciones Avanzadas – Variables de Entorno



Clic en el botón nueva (si no existe la variable) o en modificar.

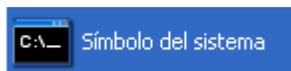


Si esta variable ya existe; no borraremos el valor de esta, por el contrario agregaremos este campo después del campo existente, separándolos por un ; (punto y coma)



Pruebas:

Esta es la manera de probar el funcionamiento de la maquina virtual JAVA



Abrir símbolo del Sistema.

```
C:\Documents and Settings\USER>JAVA -version
java version "1.4.2_07"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.2_07-b05)
Java HotSpot(TM) Client VM (build 1.4.2_07-b05, mixed mode)
```

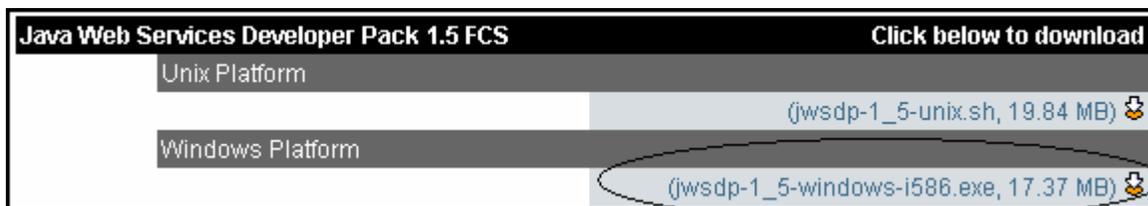
Digitar JAVA -version

2.INSTALAR JAVA WEB SERVICES DEVELOPER PACK

Este es un conjunto de librerías que requiere el jsdk para poder compilar las aplicaciones Web

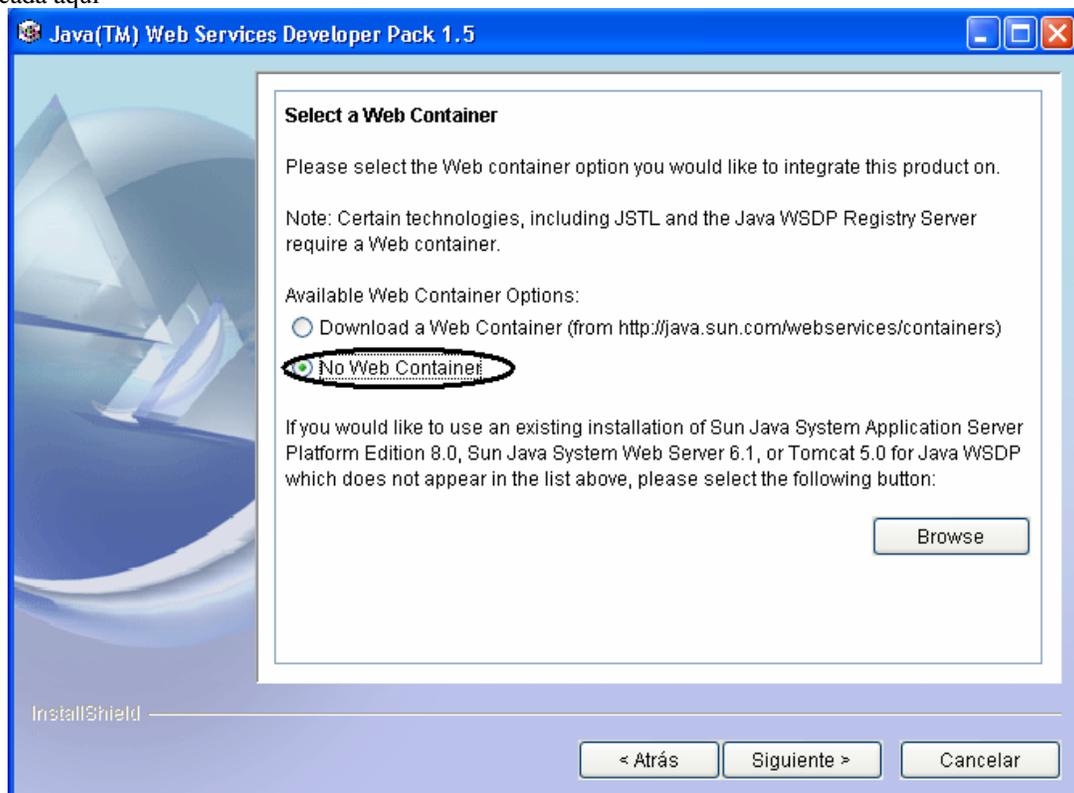
Descarga:

Nos dirigimos a la dirección y seguimos los pasos por defecto para descargar el archivo



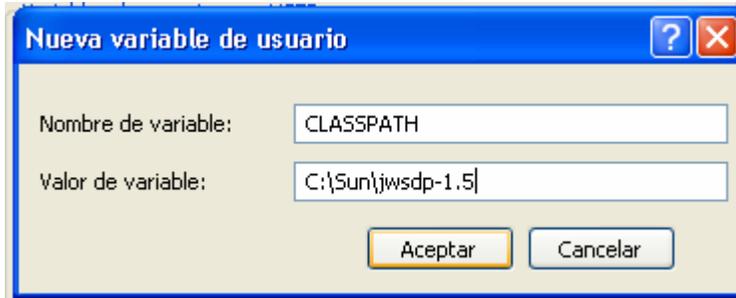
Instalación:

Después de descargar el archivo lo instalaremos normalmente; excepto que cuando aparezca esta pantalla marcaremos la opción indicada aquí



El contenedor web sera instalado posteriormente (Apache Tomcat)

Creamos una nueva variable de entorno llamada CLASSPATH (si ya existe agregaremos esta línea separándola del anterior con ;)

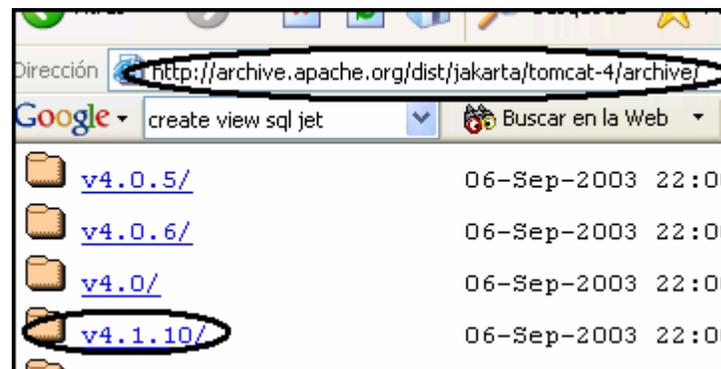


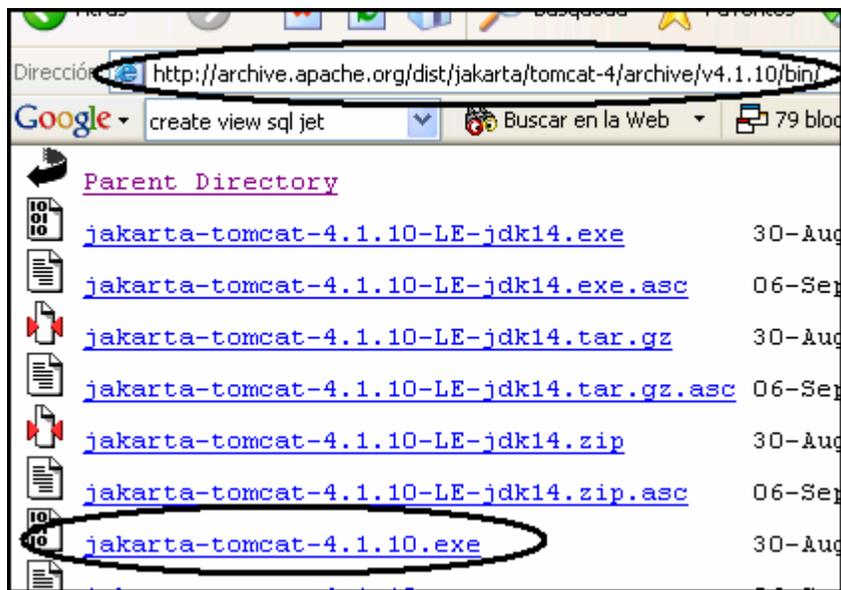
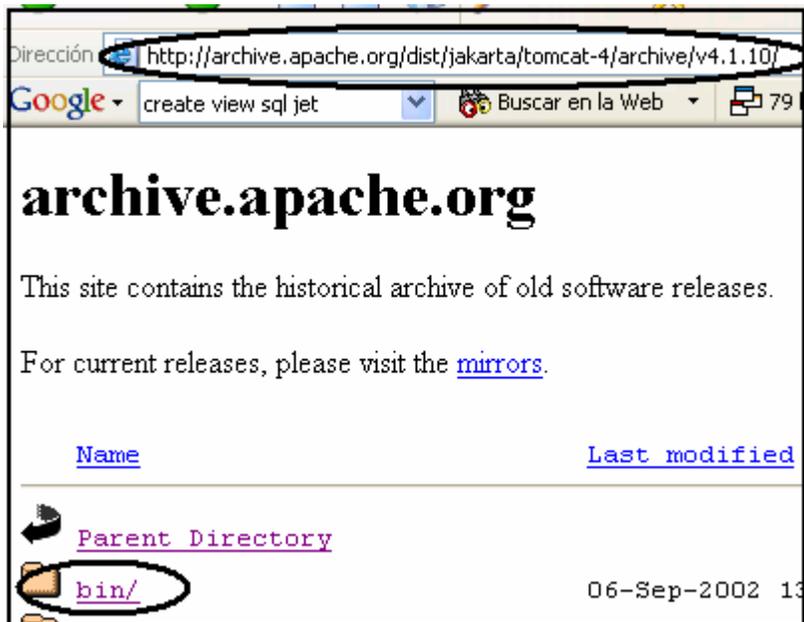
3.INSTALAR APACHE TOMCAT 4

Ahora debemos instalar el servidor de aplicaciones Web Apache tomcat; la versión 4.1.10 no presenta problemas para ejecutar los servlets.

Descarga:

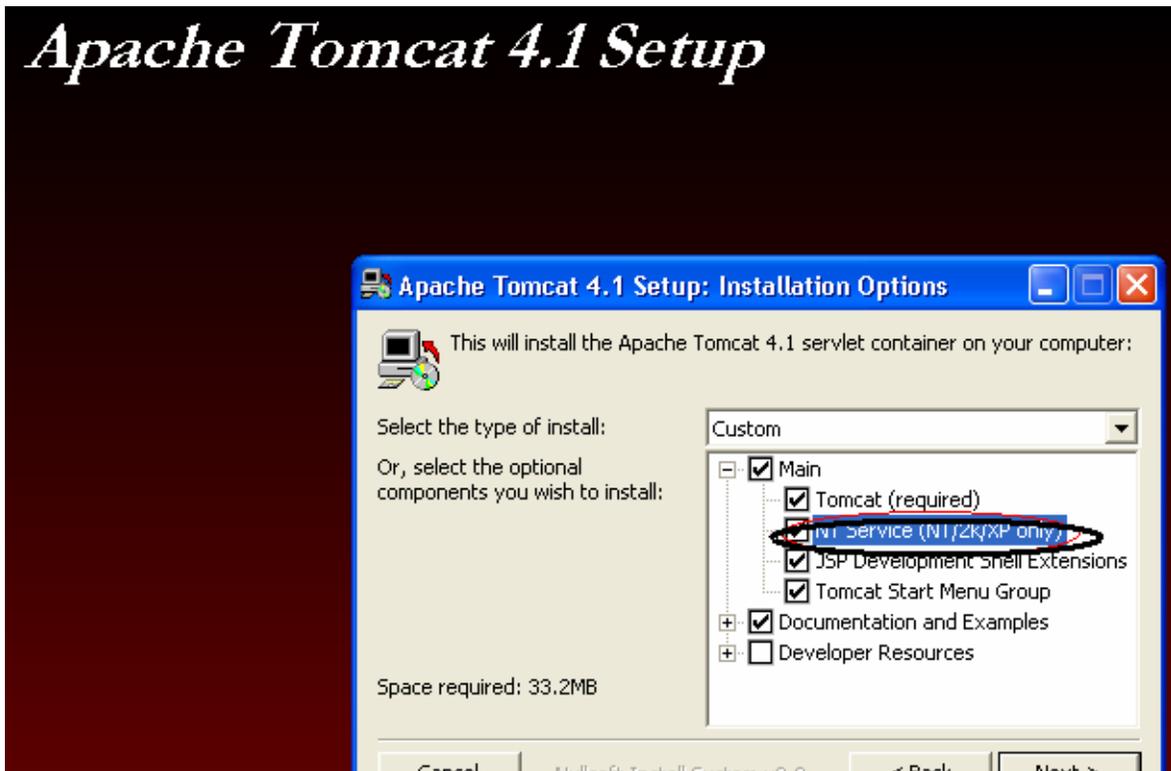
Nos dirigimos a la Web de Apache Tomcat y descargamos el archivo





Instalacion:

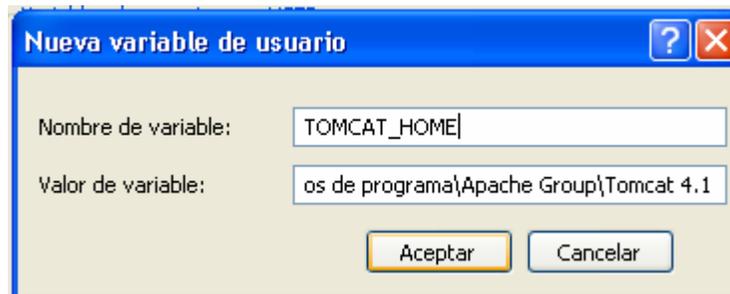
Después de descargar el archivo lo instalaremos normalmente; excepto que cuando aparezca esta pantalla marcaremos la opción indicada aquí



Crearemos 2 Nuevas variables de entorno con la ruta del Tomcat

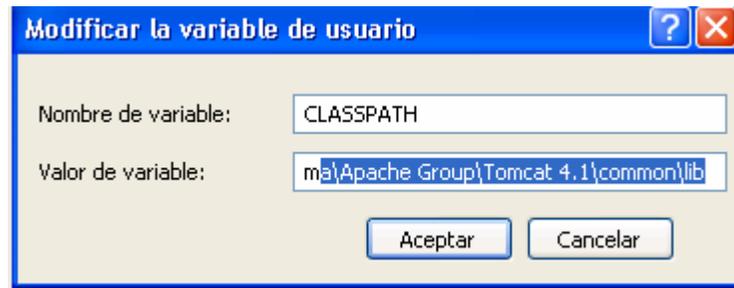


c:\Archivos de Programa\Apache Group\Tomcat 4.1



c:\Archivos de Programa\Apache Group\Tomcat 4.1

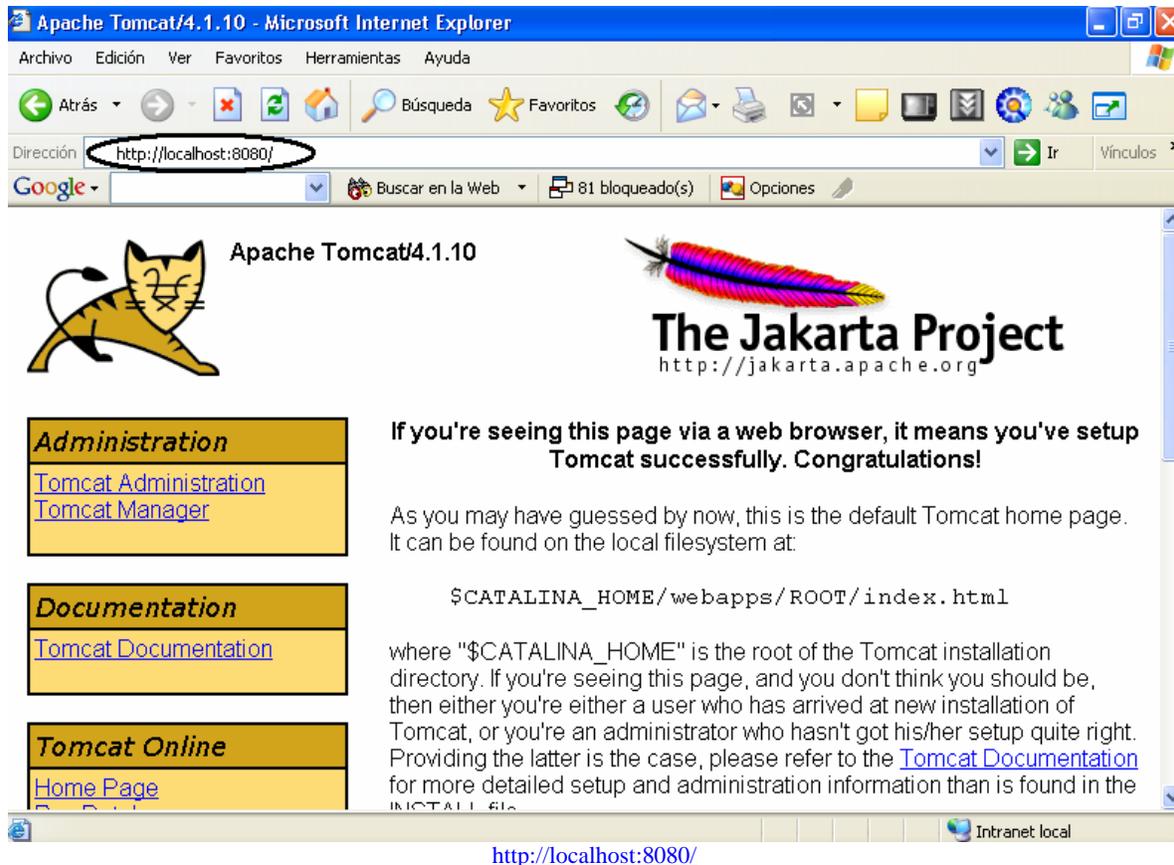
agregamos la ruta: `c:\Archivos de Programa\Apache Group\Tomcat 4.1\common\lib` al CLASSPATH (con el ;)

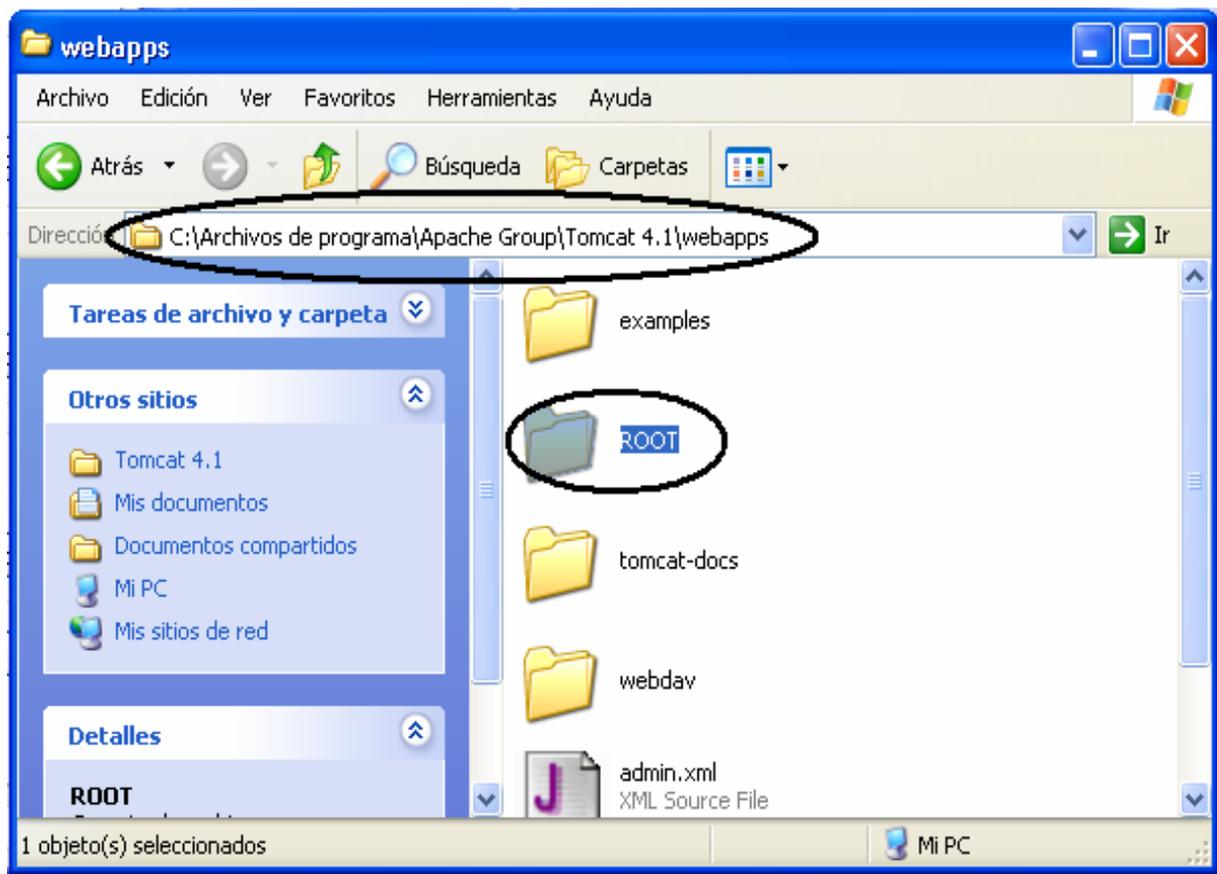


`c:\Archivos de Programa\Apache Group\Tomcat 4.1\common\lib`

Pruebas:

Abrimos el Navegador Web y Digitamos la siguiente dirección: <http://localhost:8080> si todo es correcto aparecerá la siguiente pantalla.





En esta ruta se encontrarán todas las aplicaciones que se ejecutarán el servidor apache
c:\Archivos de Programa\Apache Group\Tomcat 4.1\webapps\ROOT

4.INSTALAR ENTORNO DE DESARROLLO (NETBEANS IDE)

Ahora que tenemos el servidor funcionando podemos comenzar a desarrollar aplicaciones Web, para ello necesitamos un entorno para compilar y desarrollar el código.

Existen varios entornos gratuitos que se pueden utilizar.

NetBeans IDE <http://www.netbeans.org>

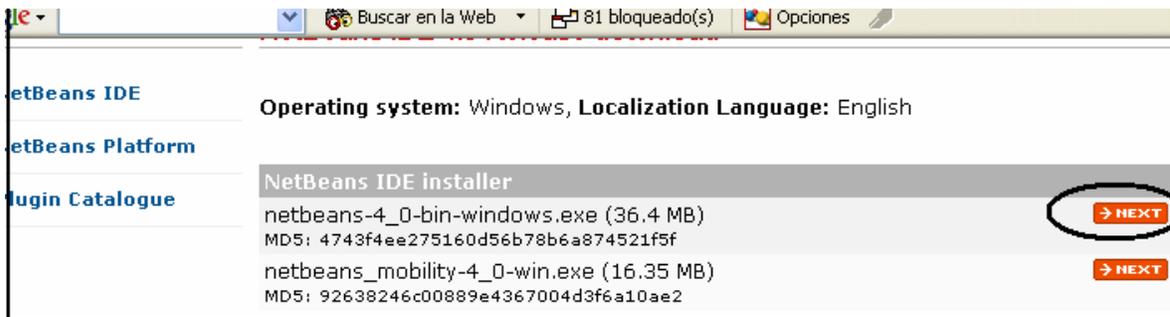
JCreator LE : <http://www.JCreator.com>

Eclipse IDE : <http://www.eclipse.org>

En nuestro caso elegimos en Netbeans IDE pues no requiere instalación de componentes adicionales para su funcionamiento.

Descarga:

Nos dirigimos a la Web de Netbeans y descargamos el archivo siguiendo las instrucciones

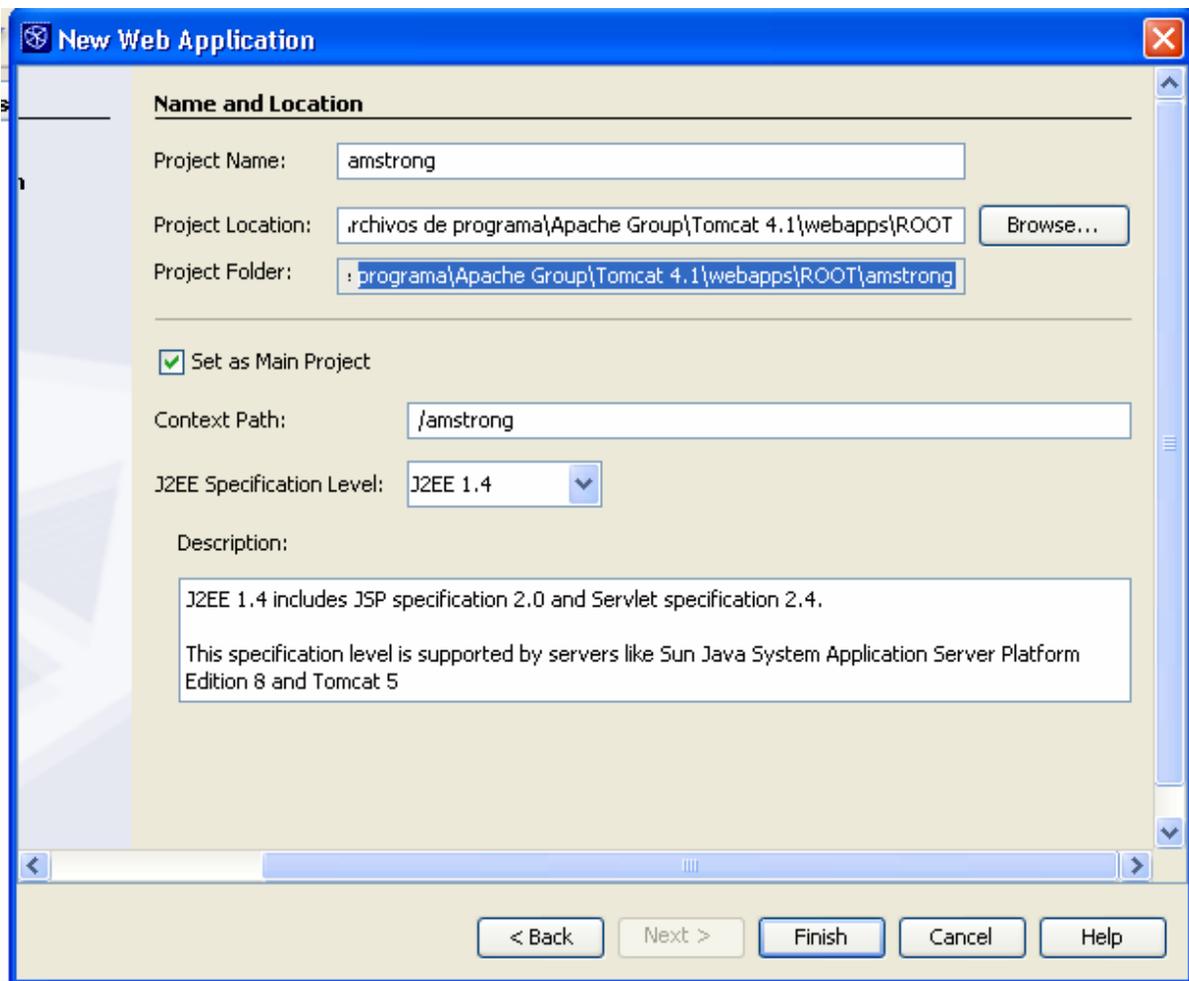
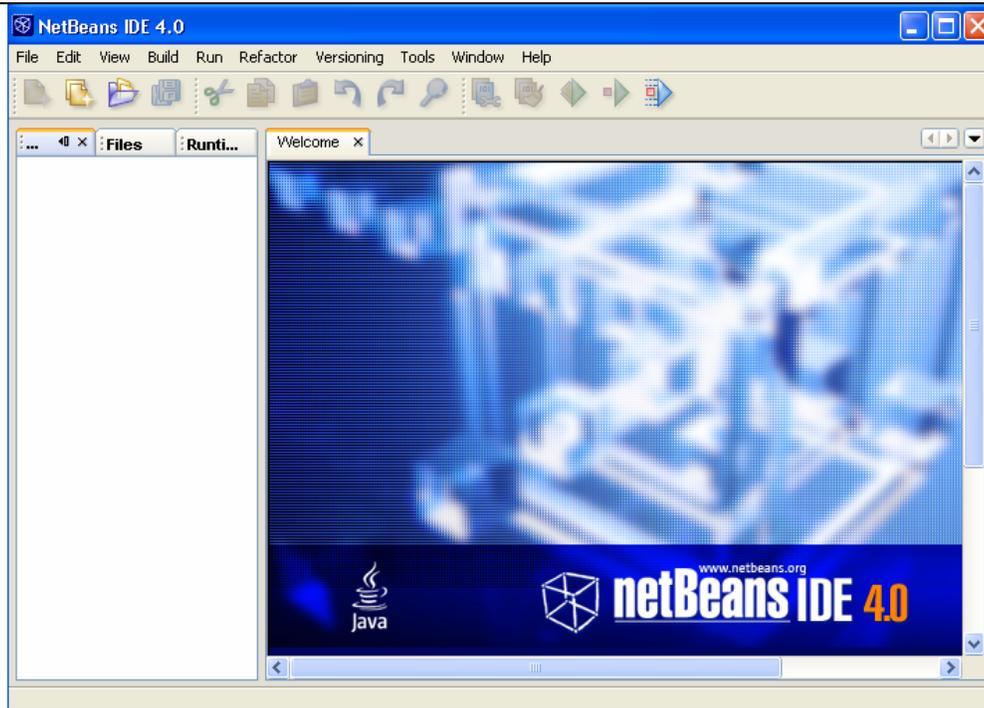


Instalacion:

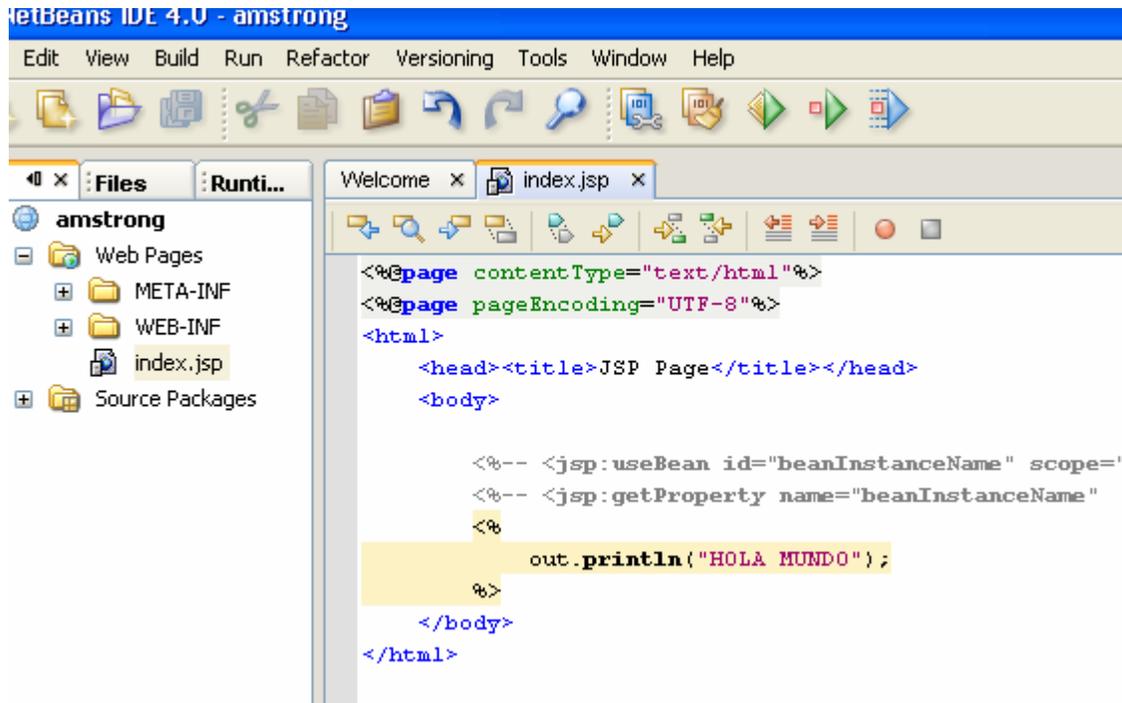
Realizamos la instalación por defecto

Pruebas:

Vamos a probar nuestro entorno de desarrollo con el servidor Tomcat.



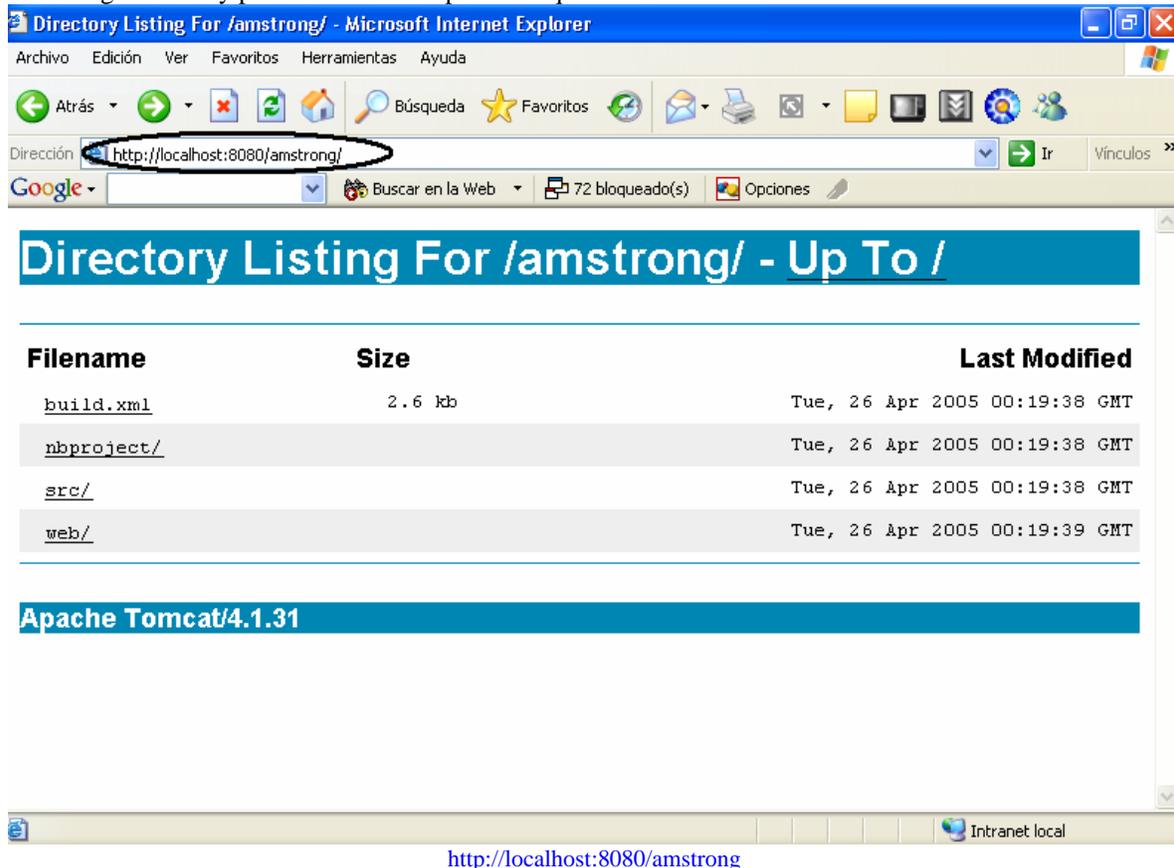
New proyect – Web Aplicacion

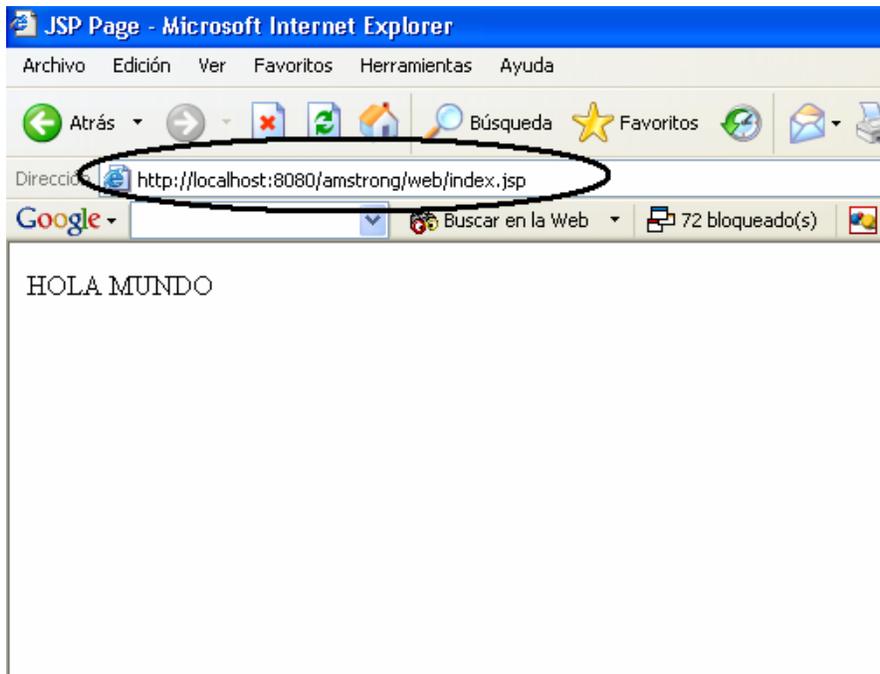


Código de prueba

Prueba con Tomcat:

Abrimos el navegador web y probamos la mini-aplicación que creamos.





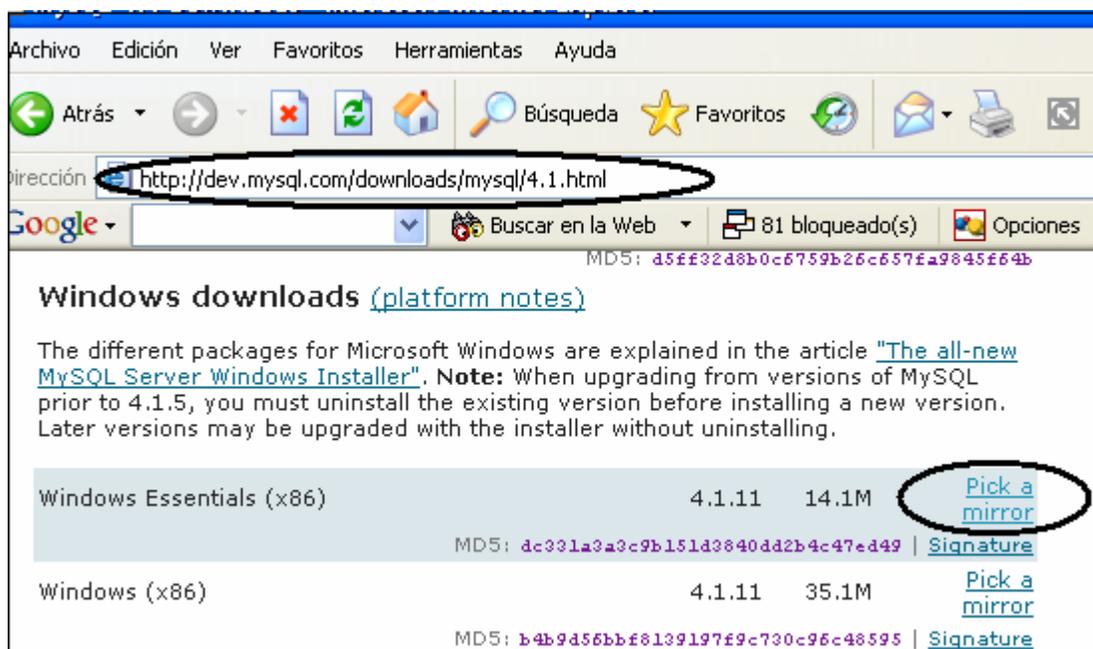
<http://localhost:8080/amstrong/web/index.jsp>

5. INSTALAR MYSQL PARA LA BASE DE DATOS

Ya que hemos testeado el servidor apache y el entorno de desarrollo, necesitamos un manejador para bases de datos; según los requisitos de nuestro proyecto el manejador a utilizarse sera MySQL.

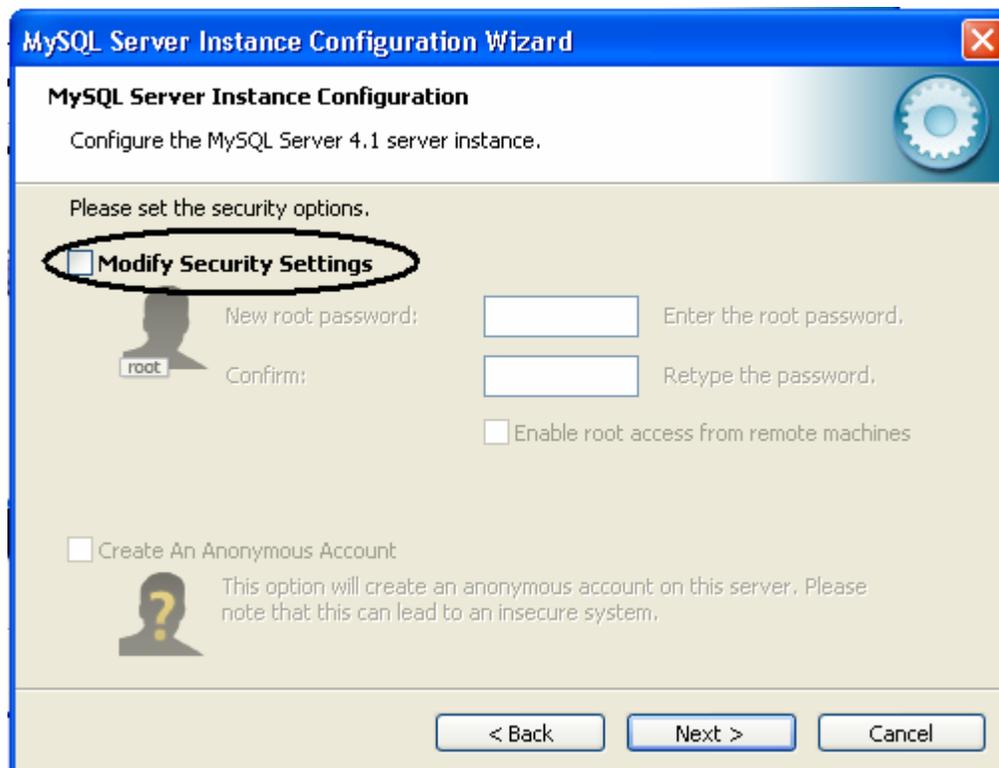
Descarga:

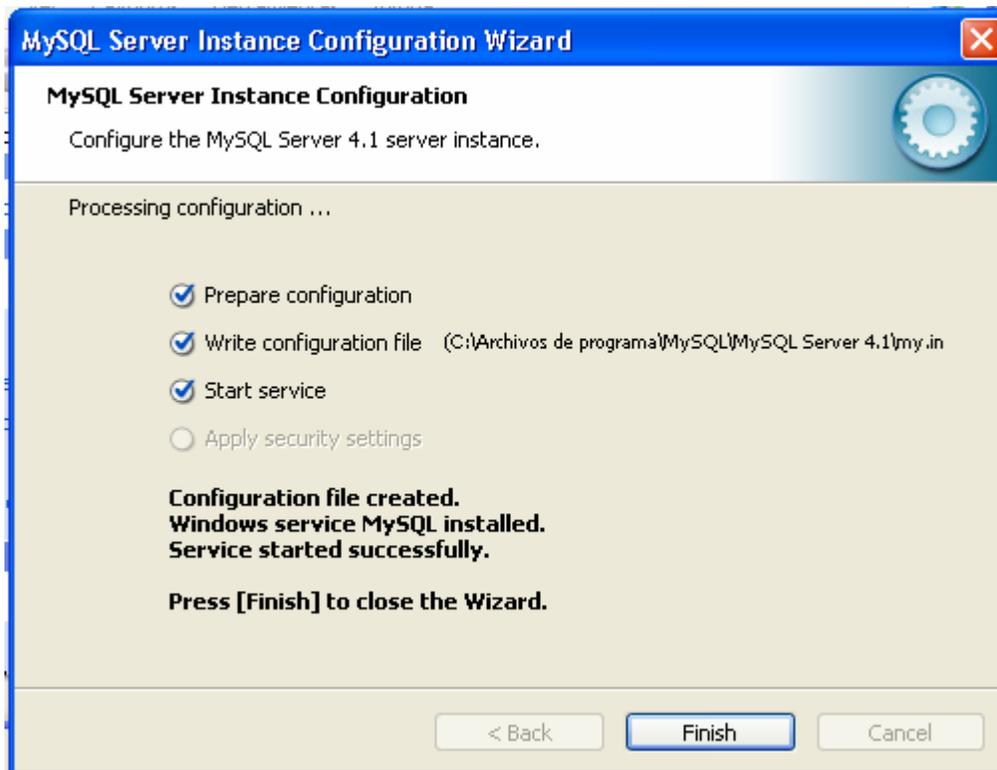
Nos dirigimos a la Web de MySQL y descargamos el instalador según las instrucciones



Instalacion y Prueba:

Después de descargar, instalaremos según los pasos, cuando encontremos estas pantallas marcaremos las opciones indicadas.





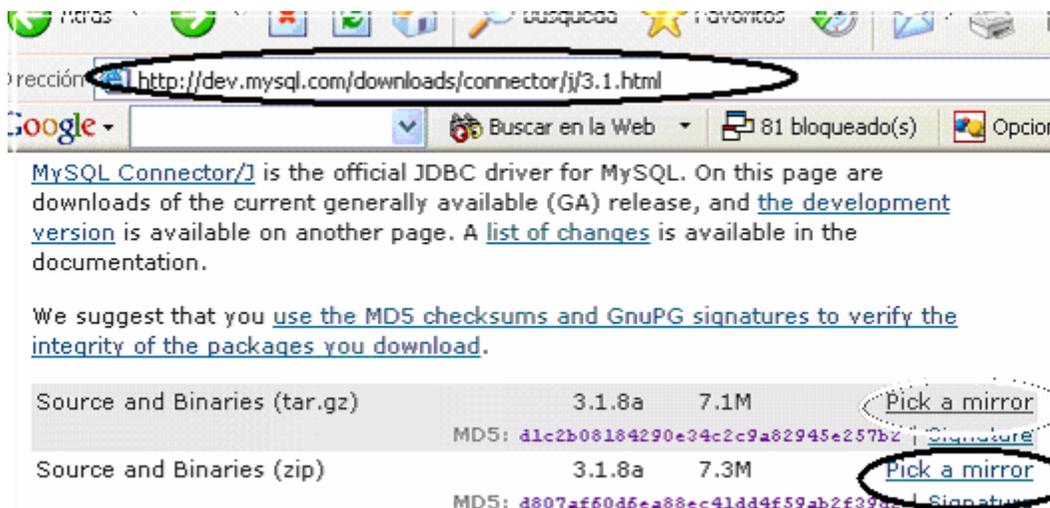
Windows service MySQL installed
Service started successfully
Estos mensajes indican la instalación exitosa

6.INSTALAR DRIVER MYSQL JDBC PARA JAVA

Como vamos a manejar conectividad entre MySQL – Java –JSP y Servlets necesitamos un controlador para que nos de acceso a la base de Datos.

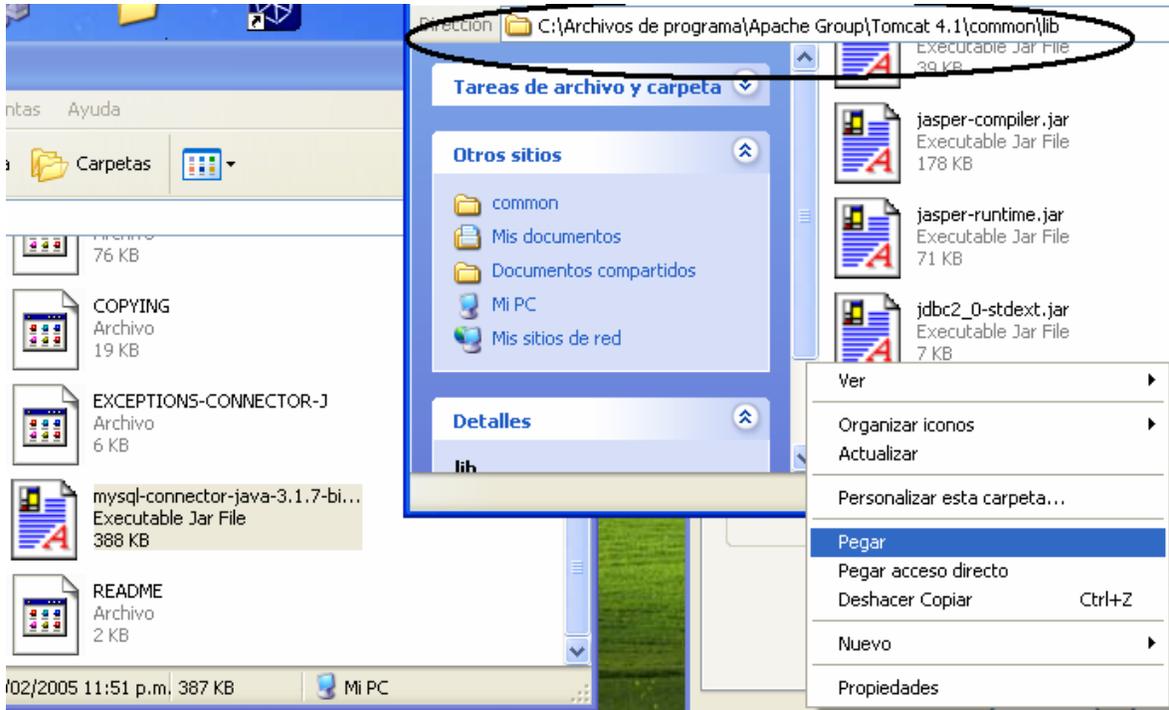
Descarga:

Descargar según las instrucciones.



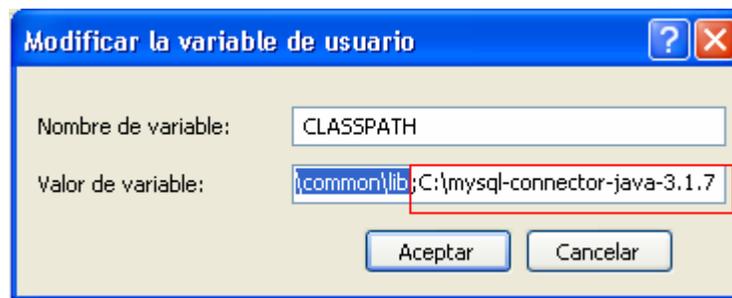
Instalacion:

Descomprimir el archivo en el directorio por default `c:\mysql-connector-java-3.1.7` y copiar al directorio `c:\Archivos de Programa\Apache Group\Tomcat 4.1\common\lib` el archivo `mysql-connector-java-3.1.7-bi.jar`



Copiar el archivo...

Agregamos al classpath la ruta del conector.



`c:\mysql-connector-java-3.1.7`

Y por ultimo reiniciamos el equipo.

Pruebas:

Con este código se puede testear el funcionamiento del driver de MySQL.
Creamos un archivo de prueba en NetBeans

```
<%@ page contentType="text/html" language="java" import="java.sql.*"%>

<head>
  <title>Test de Conexion</title>
</head>

<body>

<%
  //Definir variables de conexion
  Connection con;
  Statement stm;
  |
  try {
    Class.forName("com.mysql.jdbc.Driver");
    //Establecer conexion en la base de datos: bd 'test' ,usuario 'root' ,clave ''
    con = DriverManager.getConnection("jdbc:mysql://localhost:3306/test", "root", "");
    stm = con.createStatement();
    out.println("Conexion efectuada correctamente");
  } catch (Exception e) {
    out.println("No fue posible Conectarse" + e.getMessage());
  }
%>
</body>
```

Y Probamos el funcionamiento del Conector.



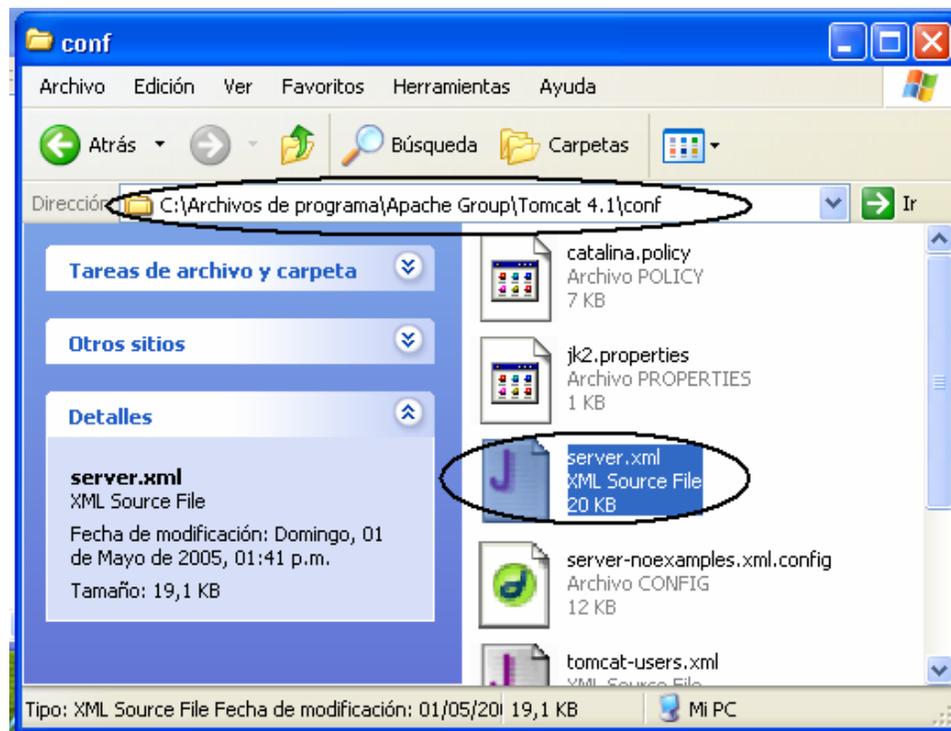
7.CONFIGURACION DE APACHE TOMCAT PARA EJECUTAR LOS SERVLETS

El último paso para empezar a desarrollar con las herramientas instaladas es configurar el Apache para ejecutar los servlets.

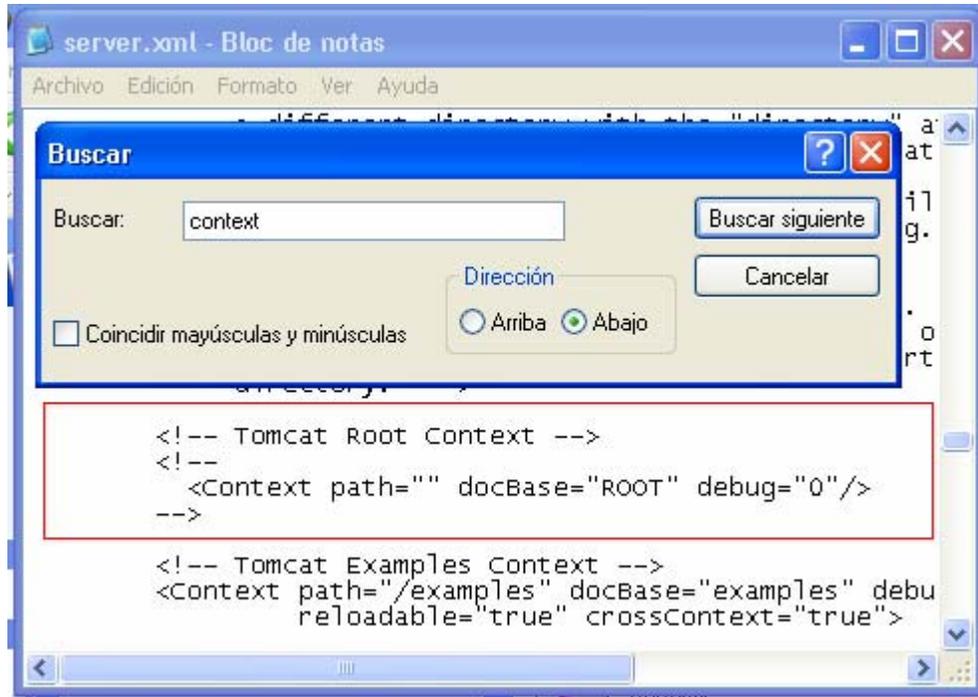
Instalacion:

Debemos editar el archivo de configuración de Tomcat: *server.xml* ubicado en

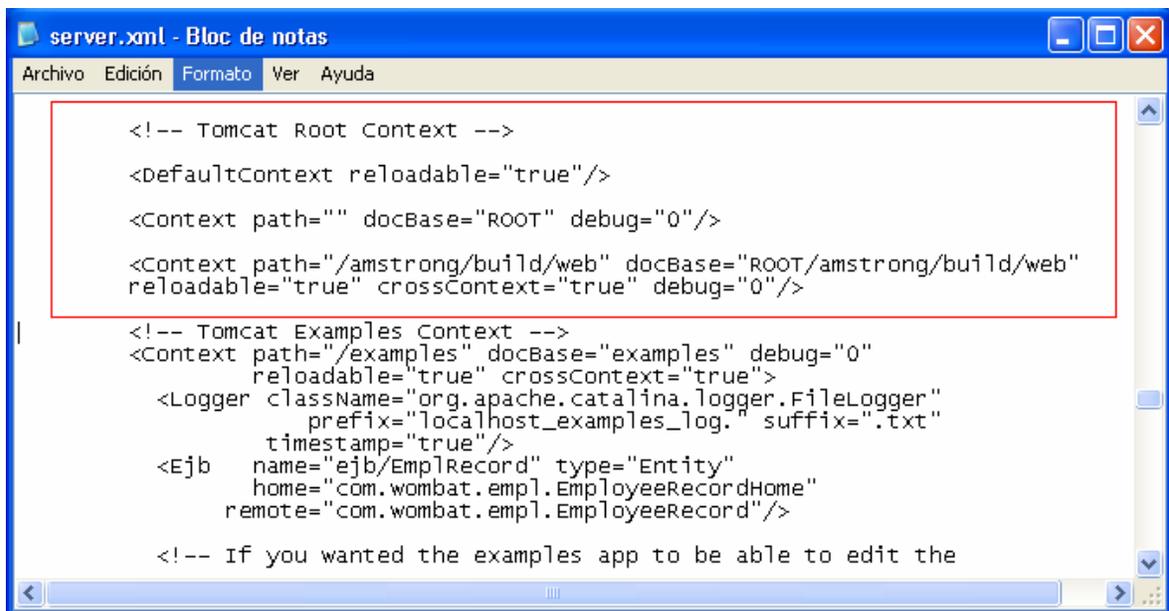
c:\Archivos de Programa\Apache Group\Tomcat 4.1\conf



Buscaremos la siguientes líneas en el archivo:

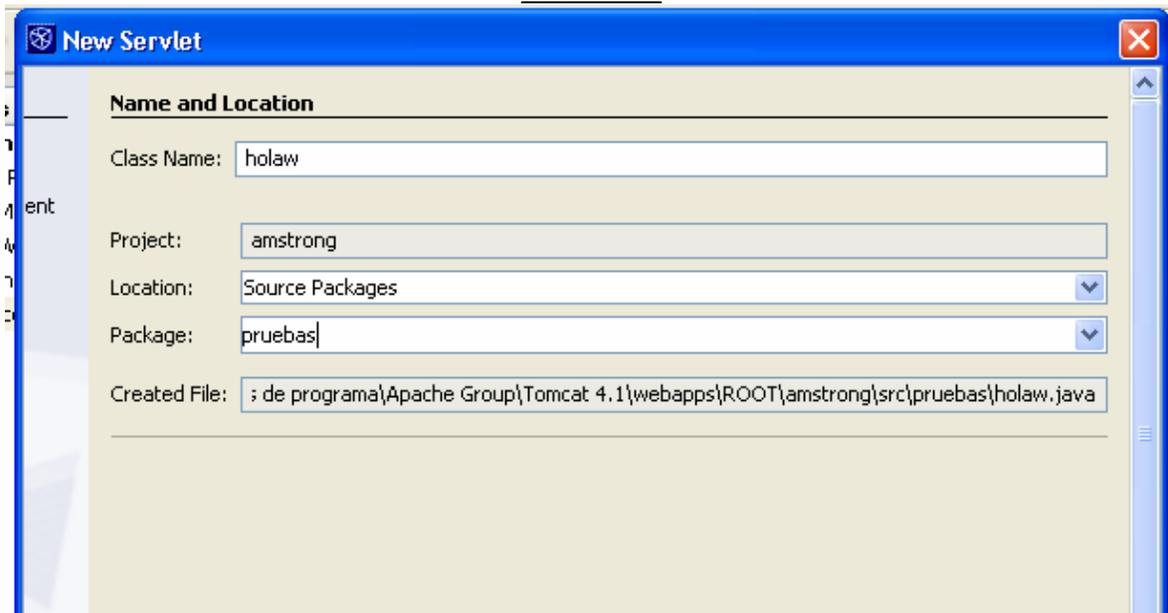
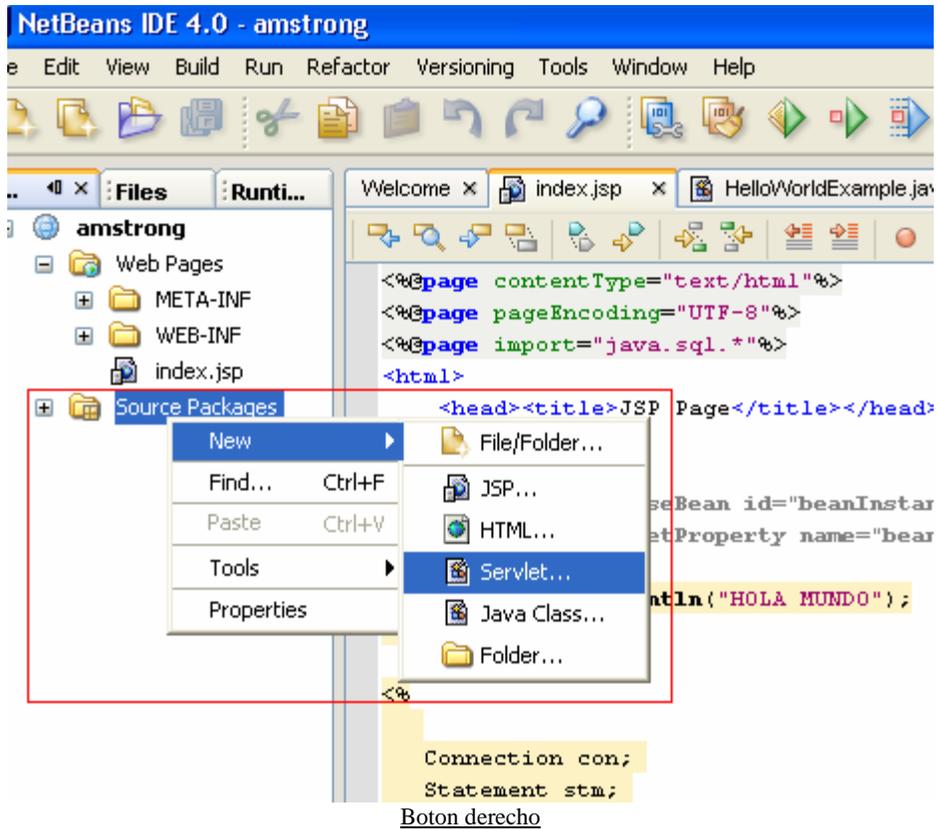


Y las modificaremos por esto:



Pruebas:

Ahora procederemos a probar la configuración con un sencillo servlet



Utilizando este código podremos probar los servlets.

```

package pruebas;

import java.io.*;
import java.net.*;

import javax.servlet.*;
import javax.servlet.http.*;

public class holaw extends HttpServlet {

    public void init(ServletConfig config) throws ServletException {
        super.init(config);
    }

    public void destroy() {
    }

    //Process Request Genera codigo html
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        //Aqui se GeneraCodigo de Salida HTML

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        //Aqui se GeneraCodigo de Salida HTML
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("HOLA A TODOS");
        out.println("</body>");
        out.println("</html>");

        out.close();
    }

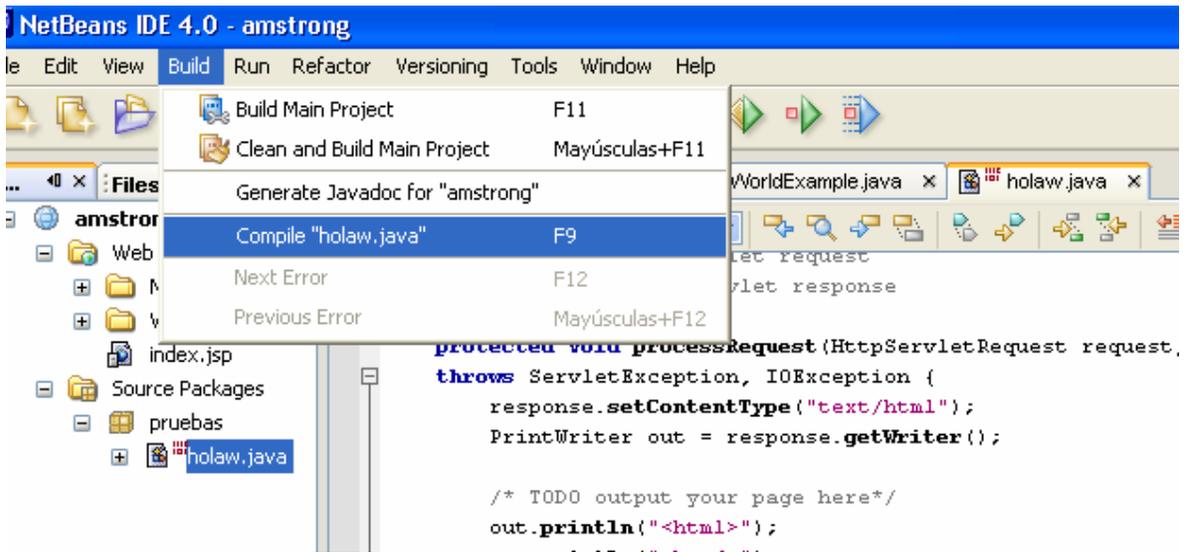
    //Los Metodos DoGet y Do Post Capturan objetos de Formularios y pueden generar html
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        processRequest(request, response);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        processRequest(request, response);
    }

    public String getServletInfo() {
        return "Descripcion";
    }
}

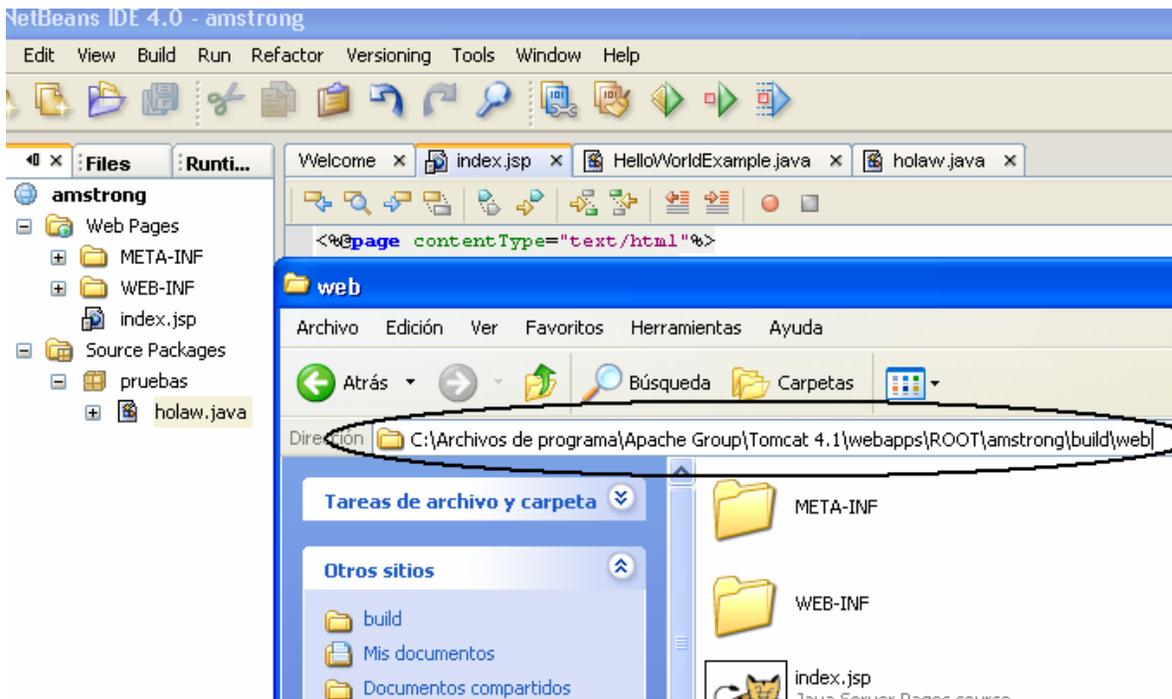
```

Compilamos el servlet

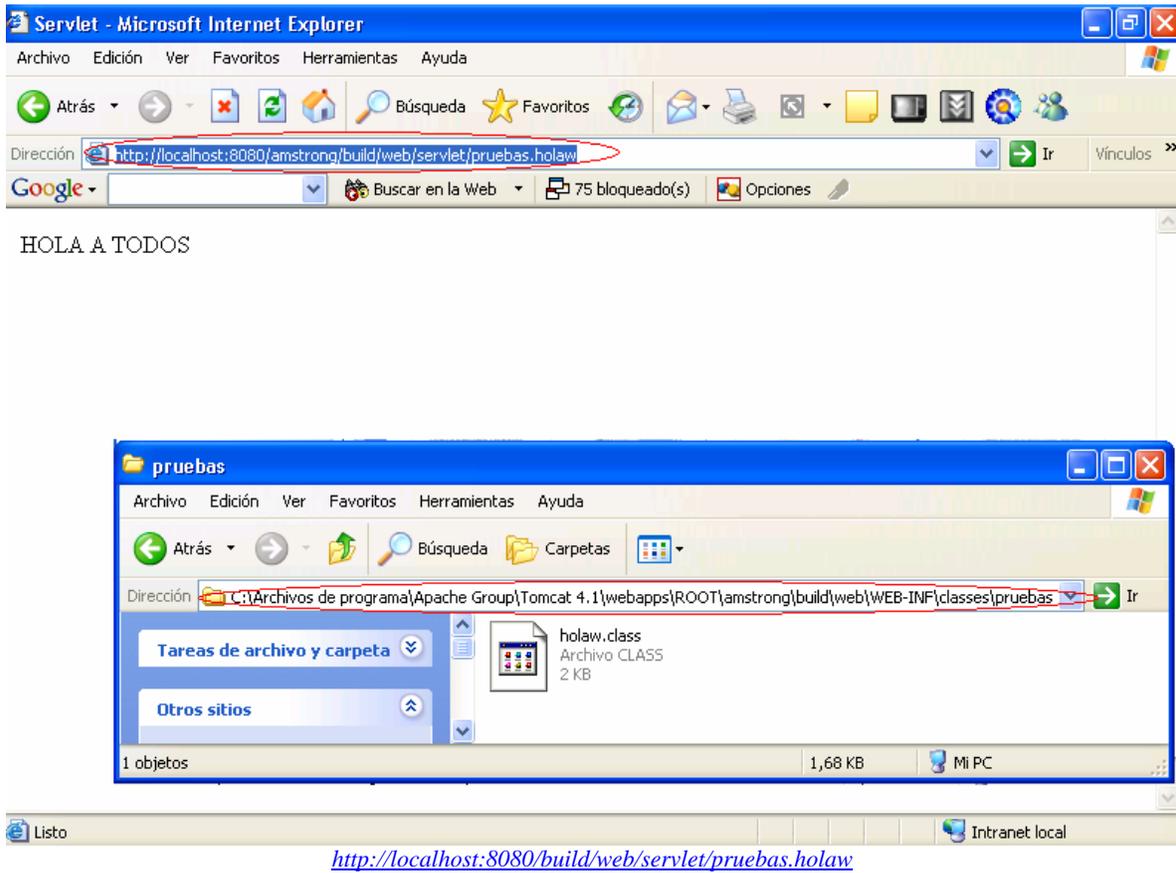


Como acabamos de compilar de ahora en adelante trabajaremos nuestra aplicación Web
En esta ruta

c:\Archivos de Programa\Apache Group\Tomcat 4.1\Webapps\ROOT\amstrong\build\web



Esta es la prueba de el servidor apache con los servlets.



Ahora el Entorno de Desarrollo está listo para comenzar el trabajo