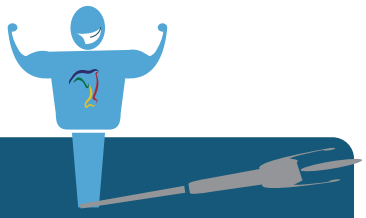
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

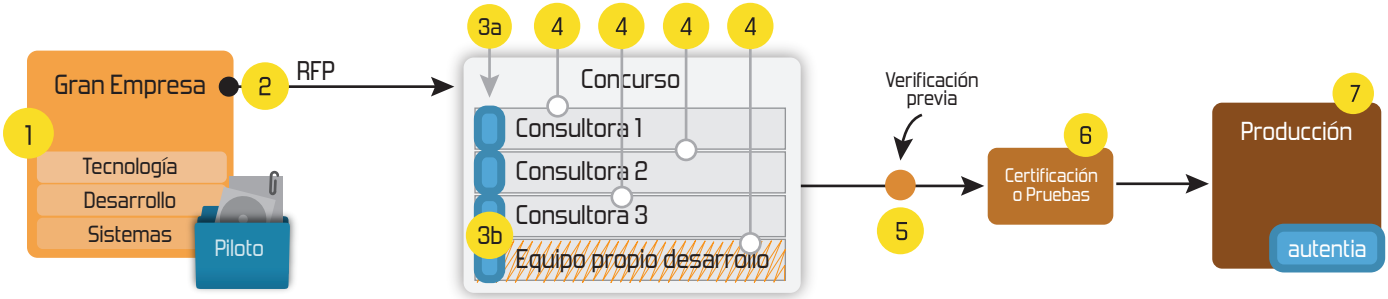
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces, HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Control de autenticación y acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)

Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

Hosting Patrocinado por enREDados.com

Home | Quienes Somos | Empleo | Foros | Tutoriales | Servicios Gratuitos | Contacte



Tutorial desarrollado por:
Roberto Canales Mora 2003-2005
Creador de AdictosAlTrabajo.com y

Director General de Autentia S.L.

Recuerda que me puedes contratar para echarte una mano:

Desarrollo y arquitectura Java/J2EE
Asesoramiento tecnológico Web
Formación / consultoría integrados en tu proyecto

No te cortes y contacta: 655 99 11 72rcanales@autentia.com.

autentia
real business solutions

Descargar este documento en formato PDF jspbasico.pdf

Curso Web J2EE
Curso Avanzado en Desarrollo Web con J2EE

JSP Tutorials
JSP Made Easy With XMLSpy. Syntax & Editing Help, Free D/L.

IntelliJ IDEA
Professional Java IDE for professional developers. Get Trial!

Formación Empresas
Consultoría de Formación Tecnologías Web

Anuncios Goooooogle Anunciarse en este sitio

Proyectos con JSPs

Todo el que haya desarrollado alguna vez páginas HTML verá lo sencillo que es construir aplicaciones en Java utilizando JSPs.

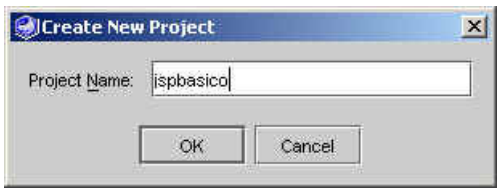
Vamos a empezar con ejemplos sencillos que iremos complicando ...

Uno de los elementos claves en el desarrollo de aplicaciones es tener claro el objetivo. El desarrollo de JSPs es un medio que vamos a utilizar con el fin de crear un buscador de tutoriales.

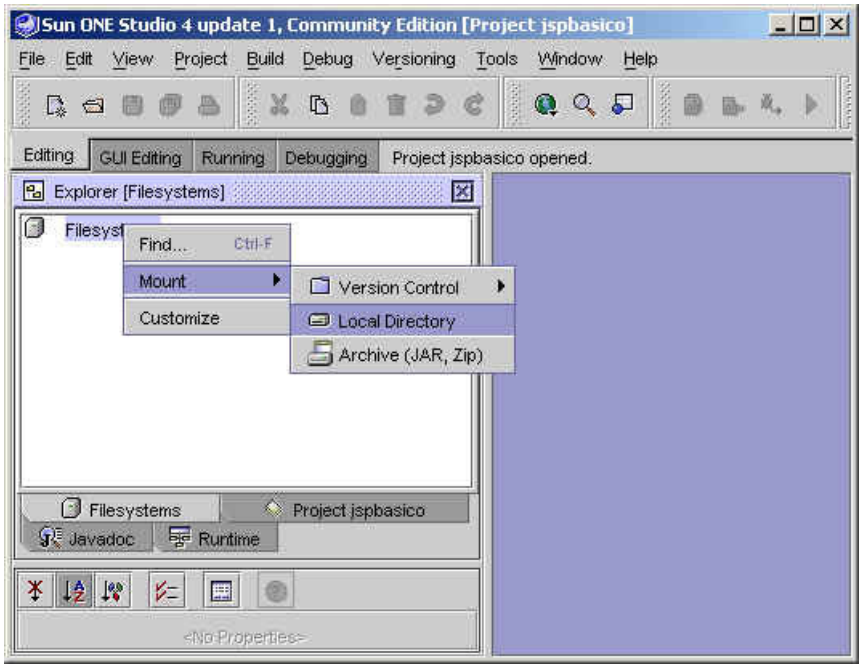
Preparación del entorno

Vamos a usar Forte 4 (la versión gratuita ... que esta basada en NetBeans y que deja de estar soportada por SUN) para montar nuestro ejemplo .

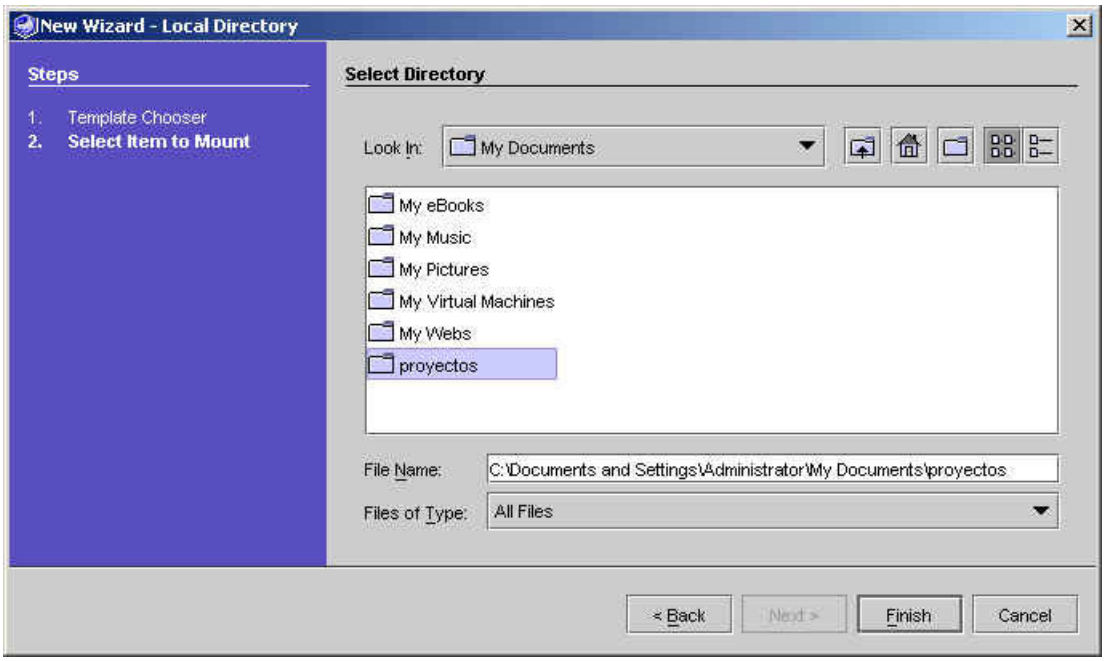
Creamos el proyecto



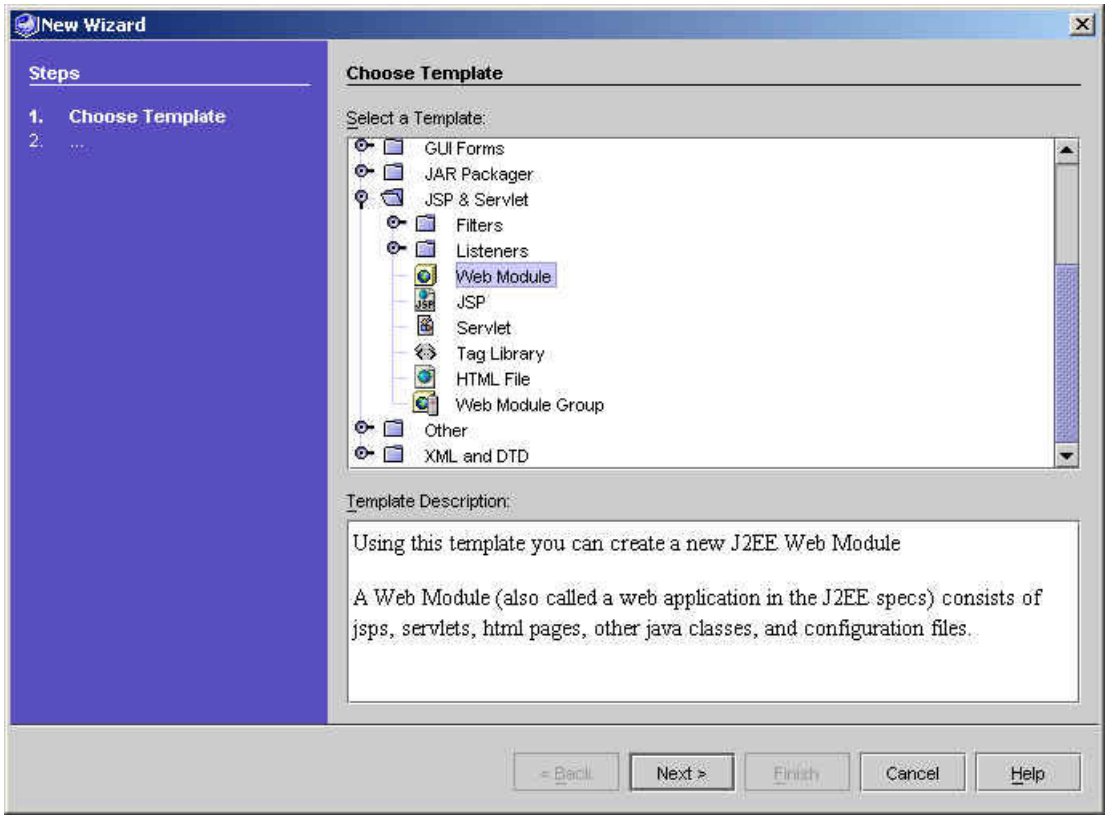
Seleccionamos el directorio de trabajo.



Elegimos la carpeta deseada



Ahora creamos un modulo Web .... pulsando en **File->new**

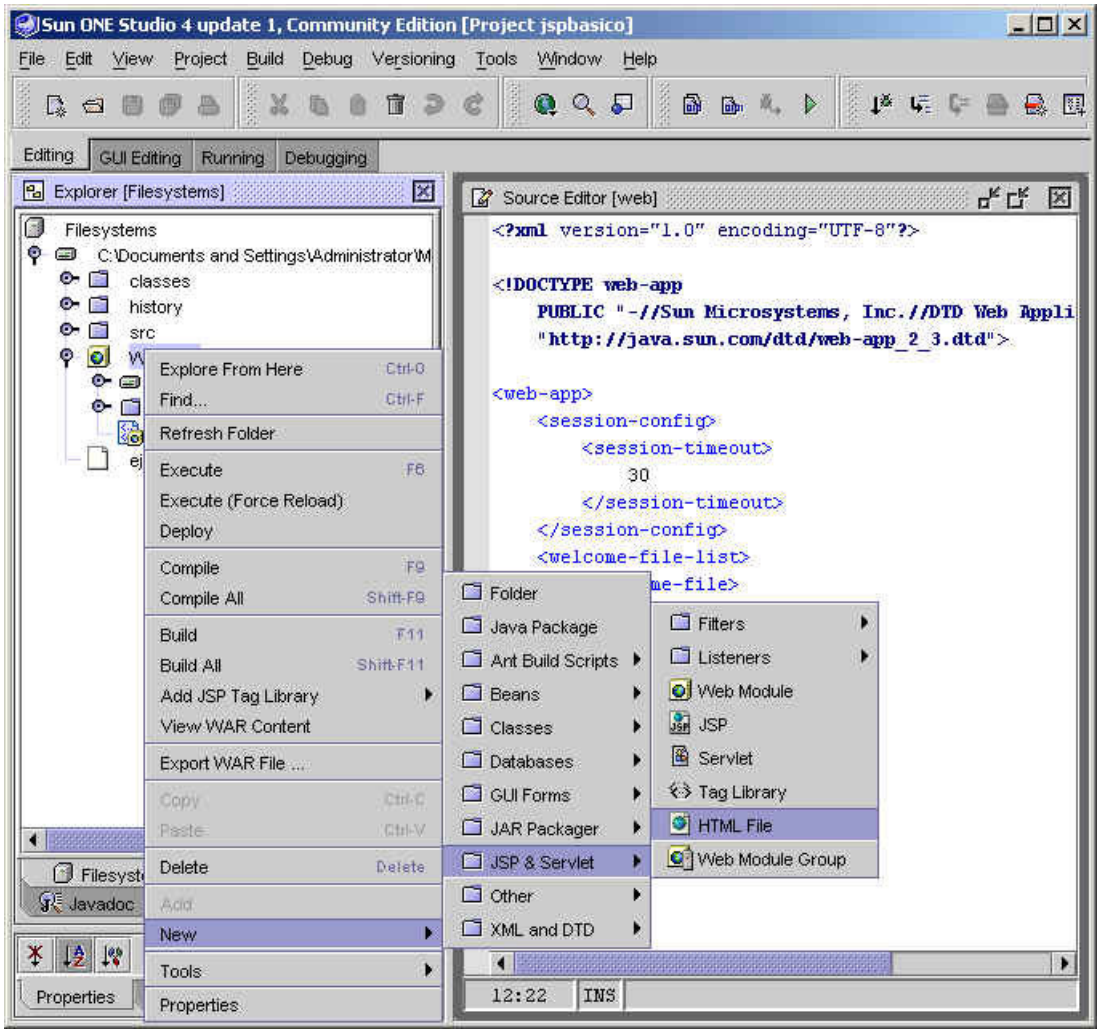


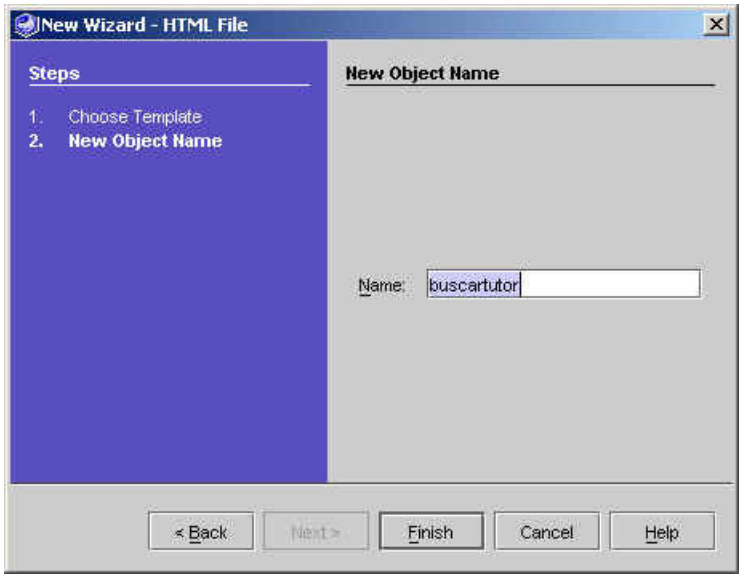
Y convertimos la carpeta en una carpeta Web



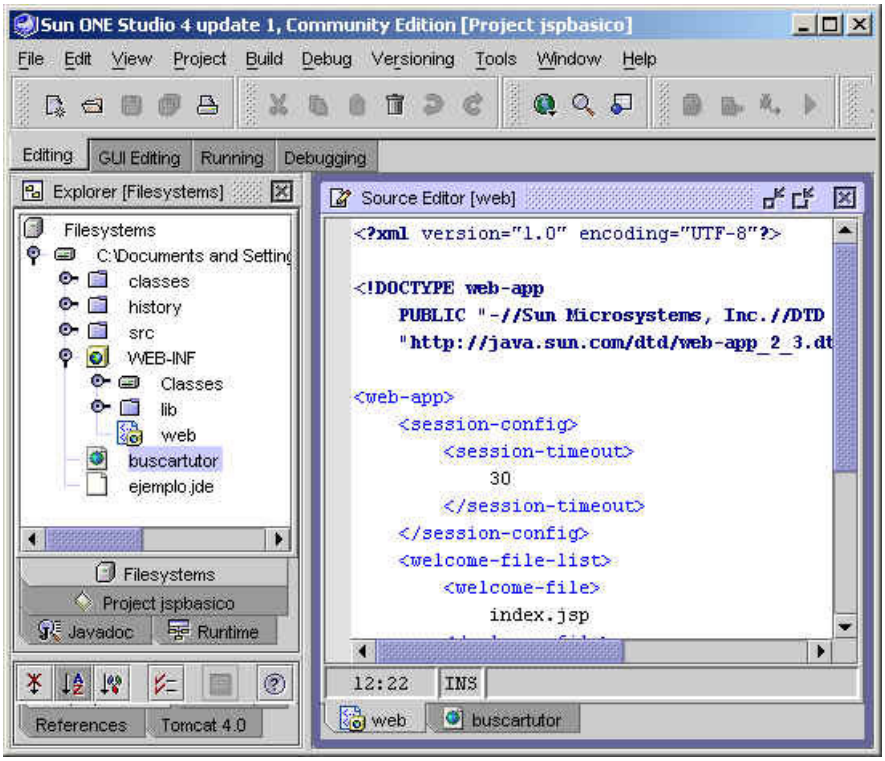
Lectura de parámetros

Una página dinámica mal va a poder hacer su trabajo si no es capaz de leer los datos que recibe de un formulario.

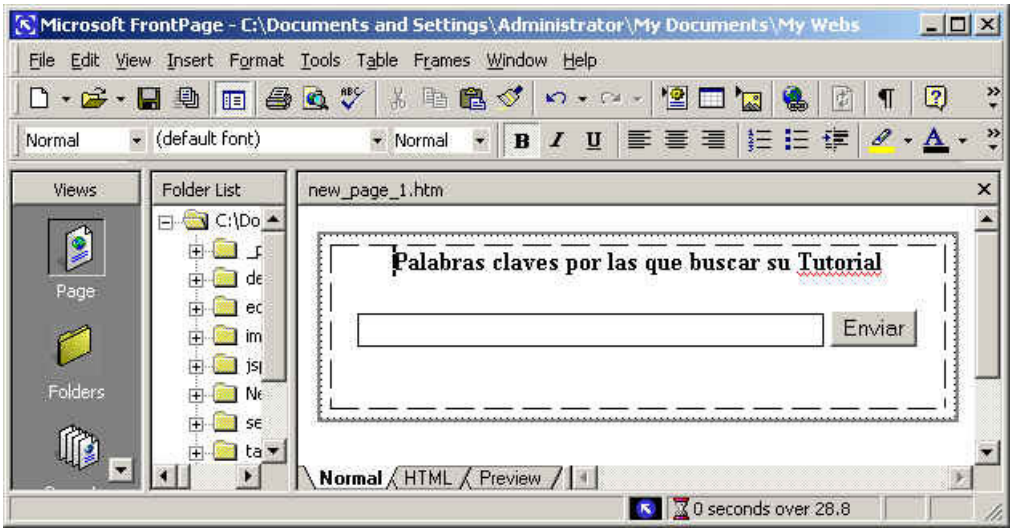


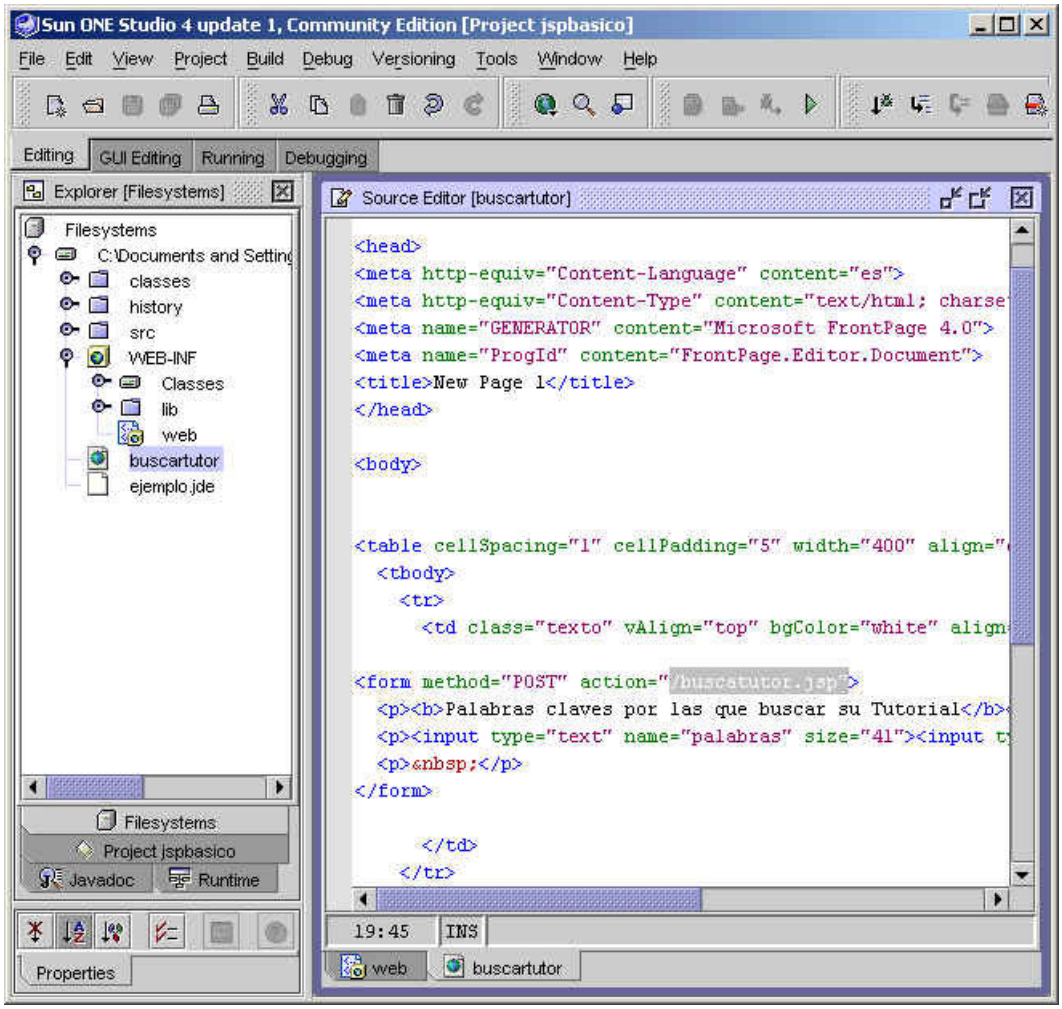


Vemos como queda nuestra página

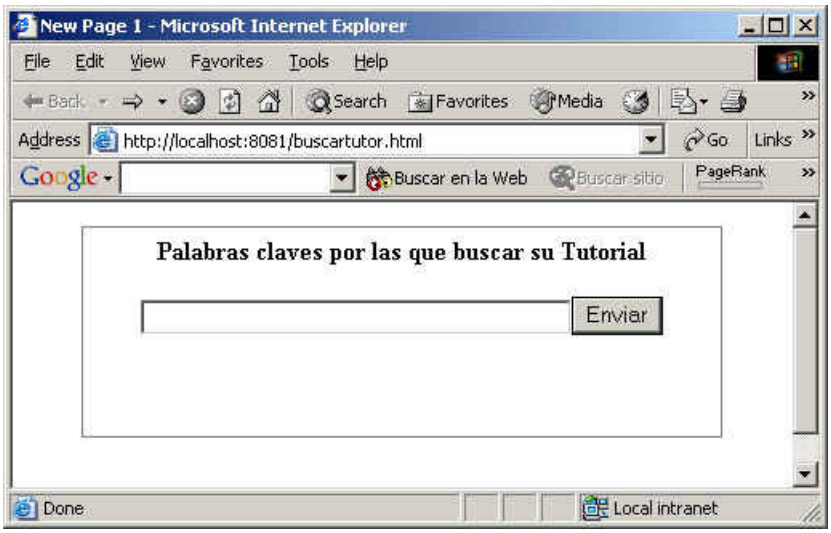


Vamos a crear un formulario en HTML. Para ello nos podríamos ayudar del FrontPage..... luego pegaremos el código.

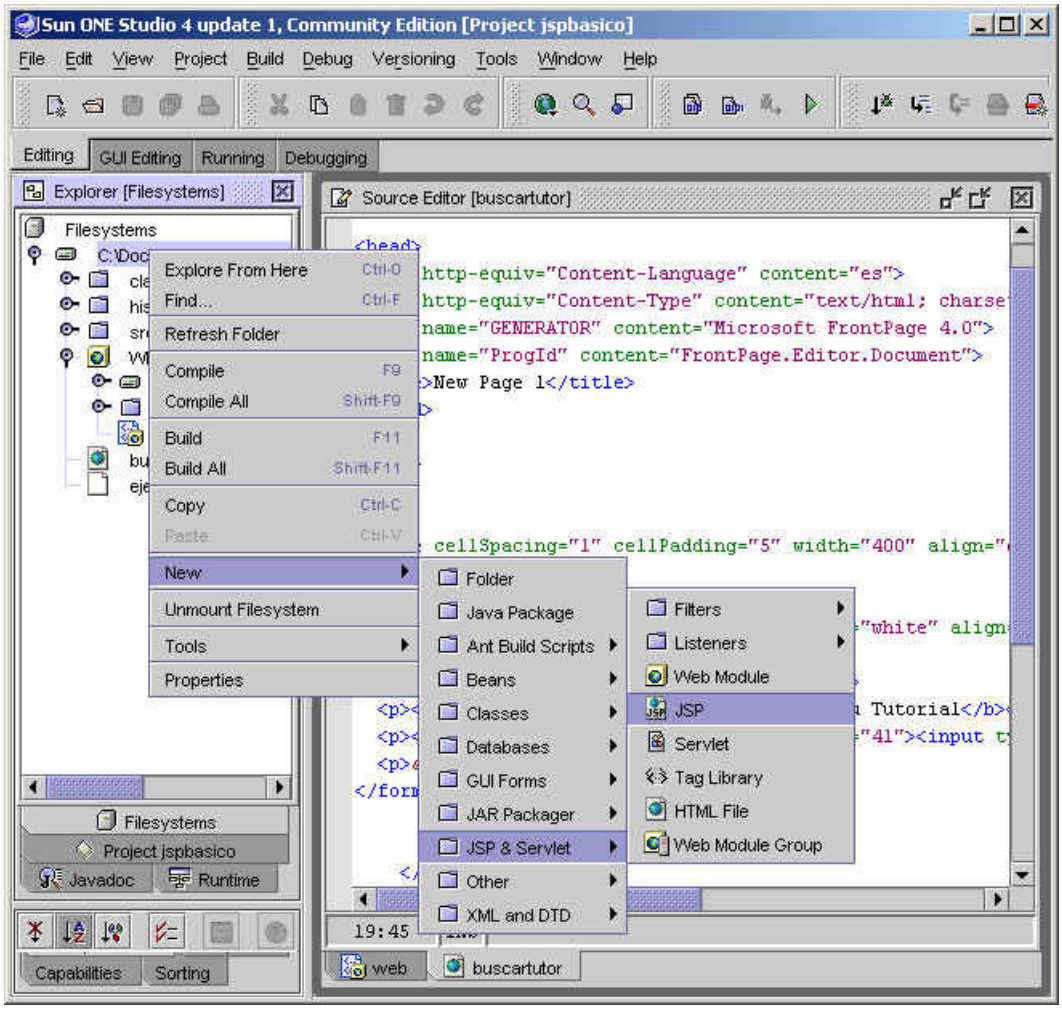




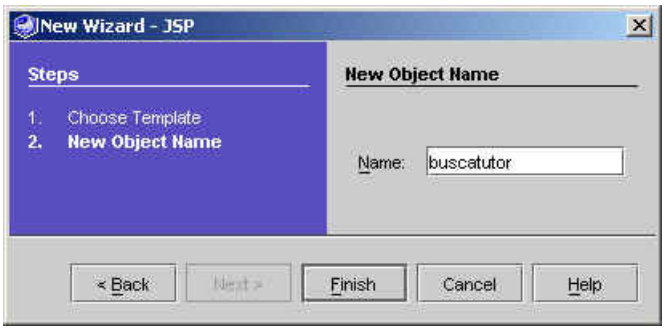
Si pulsamos el botón de previsualización .... ya vemos nuestro formulario



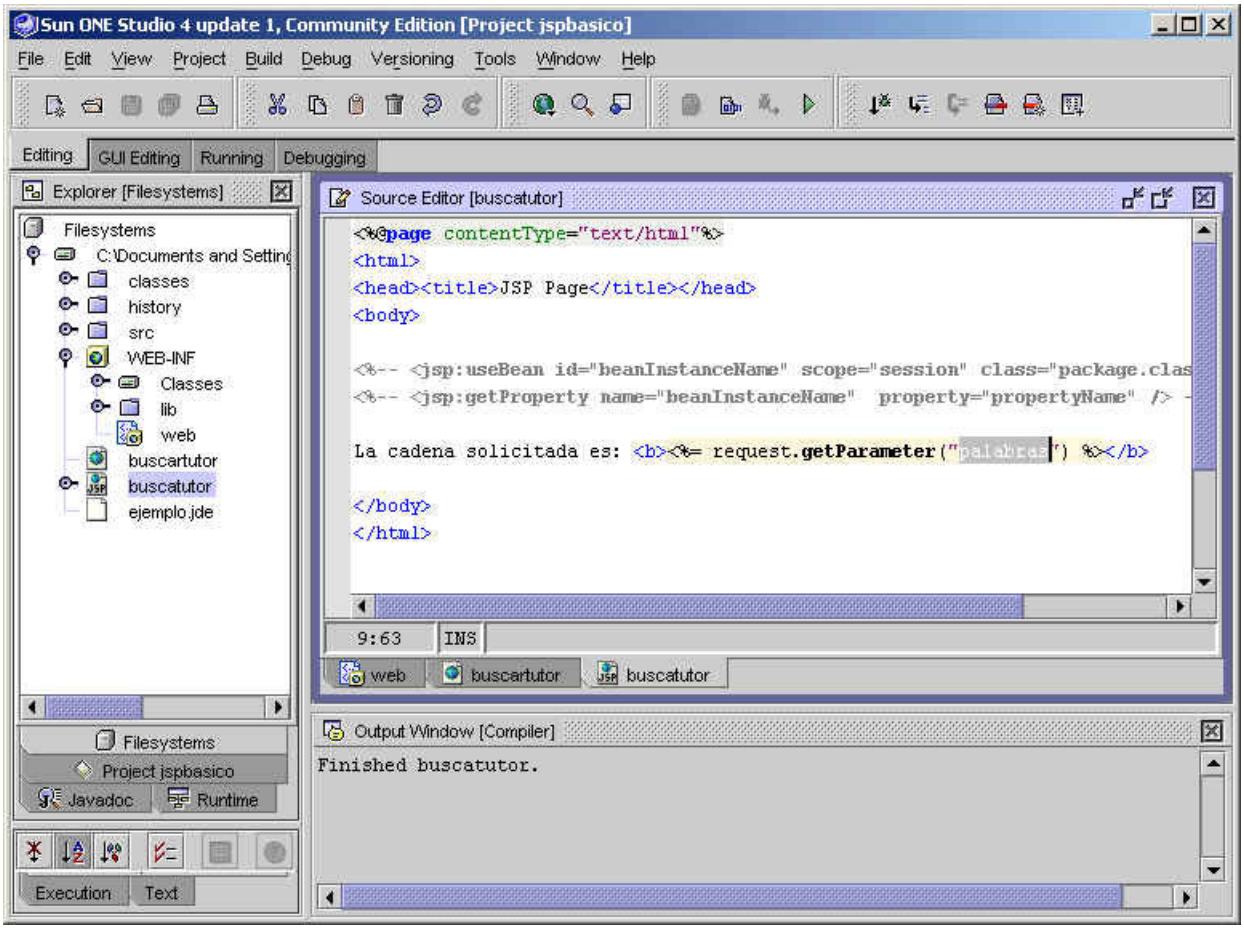
Ahora debemos construir el JSP.



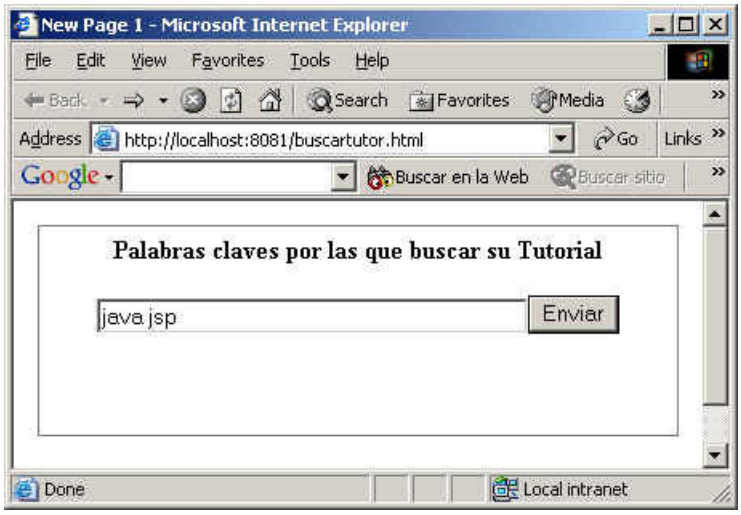
Le asignamos un nombre



Y vamos a leer el parámetro y mostrarlo en pantalla .. luego ya haremos cosas un poco más complicadas.



Si volvemos a ver la página HTML

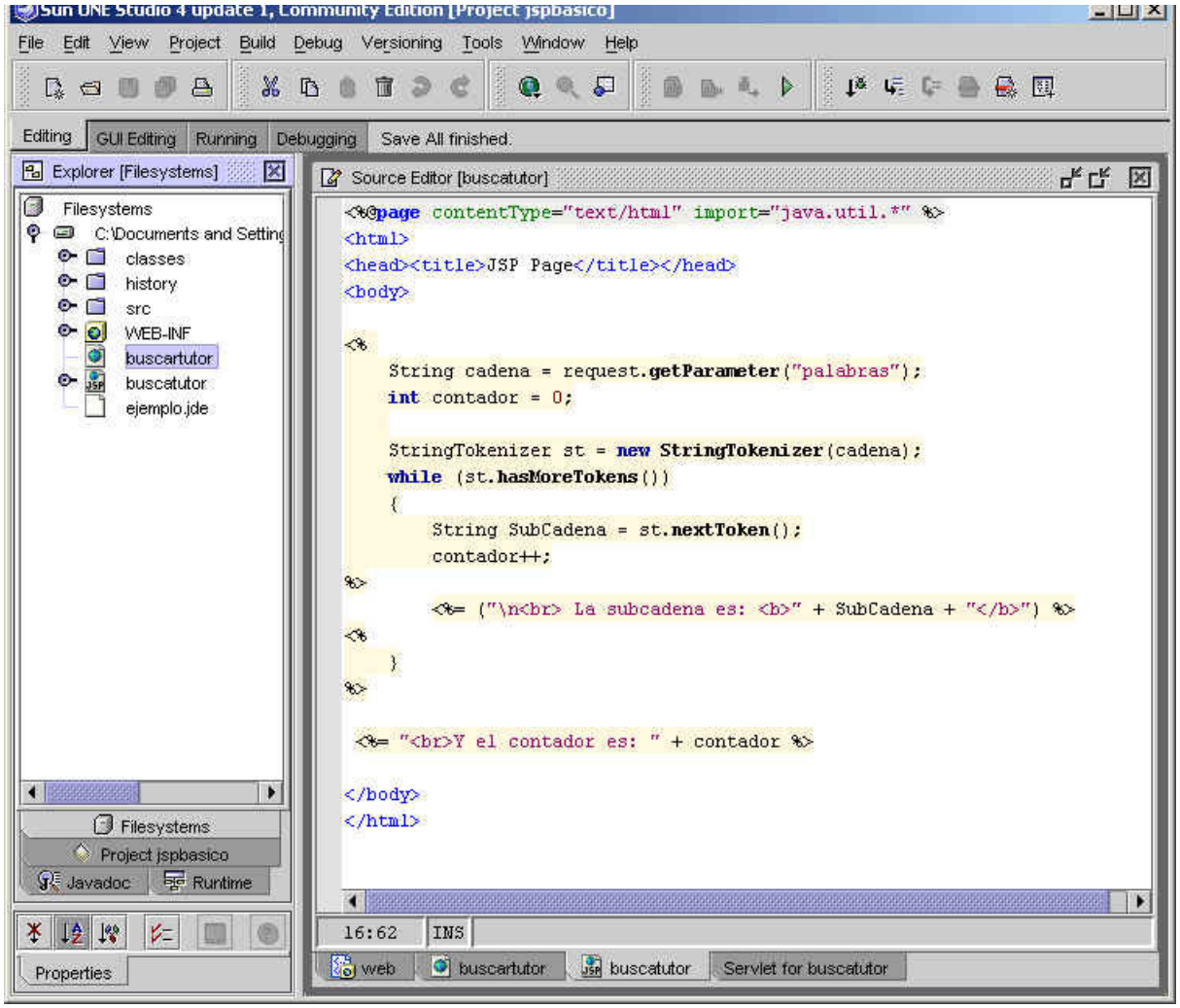


Y pulsamos el botón vemos que todo a funcionado.



Como hemos podido ver, la etiqueta **<%= %>** permite mostrar el valor de una variable.

Hay otra etiqueta dentro de las páginas JSP, todavía más poderosa <% %>, ya que nos permite, de un modo sencillo, incluir cualquier código Java dentro de nuestra página.




```
<%@page contentType="text/html" import="java.util.*" %>
<html>
<head><title>JSP Page</title></head>
<body>

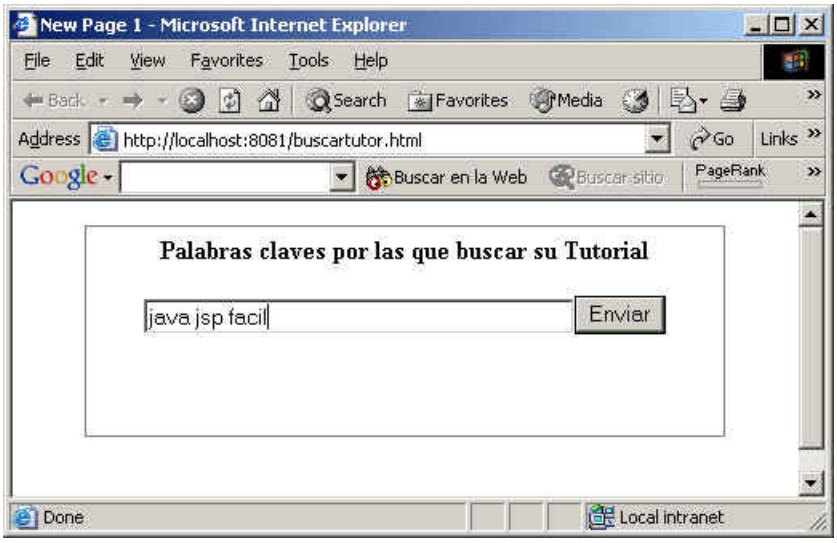

<%
String cadena = request.getParameter("palabras");
int contador = 0;

StringTokenizer st = new StringTokenizer(cadena);
while (st.hasMoreTokens())
{
String SubCadena = st.nextToken();
contador++;
%>
<%= ("\n<br> La subcadena es: <b>" + SubCadena + "</b>") %>
<%
}
%>

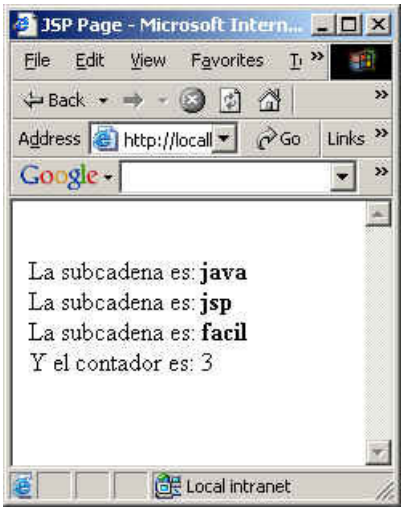
<%= "<br>Y el contador es: " + contador %>

</body>
</html>
```

Probamos nuestra página.



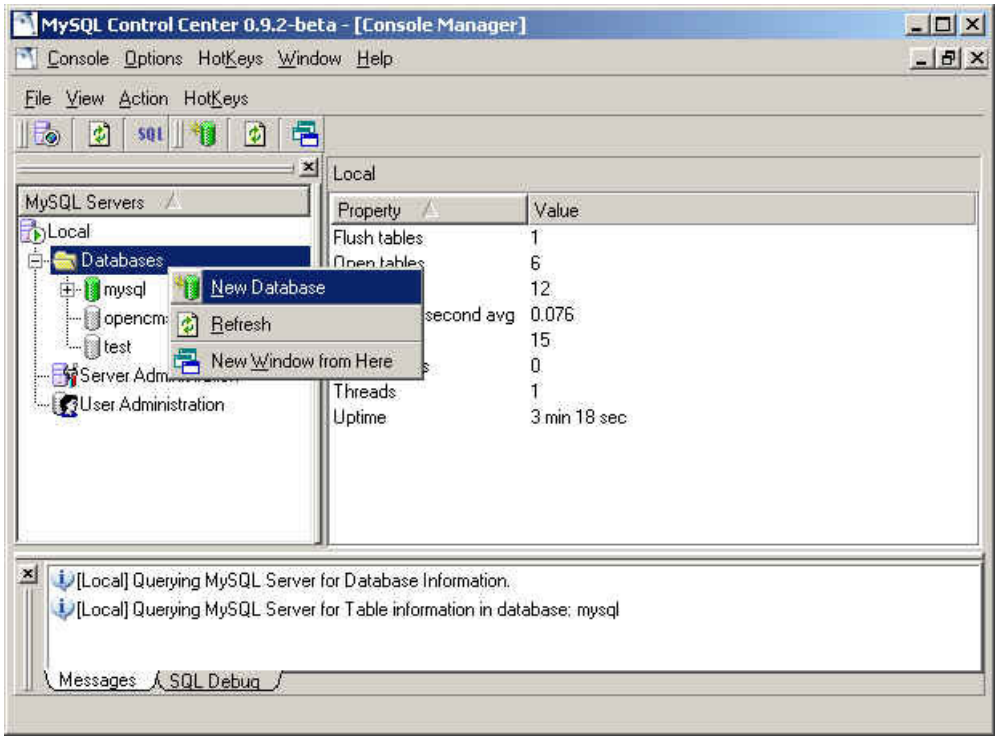
Y el resultado es



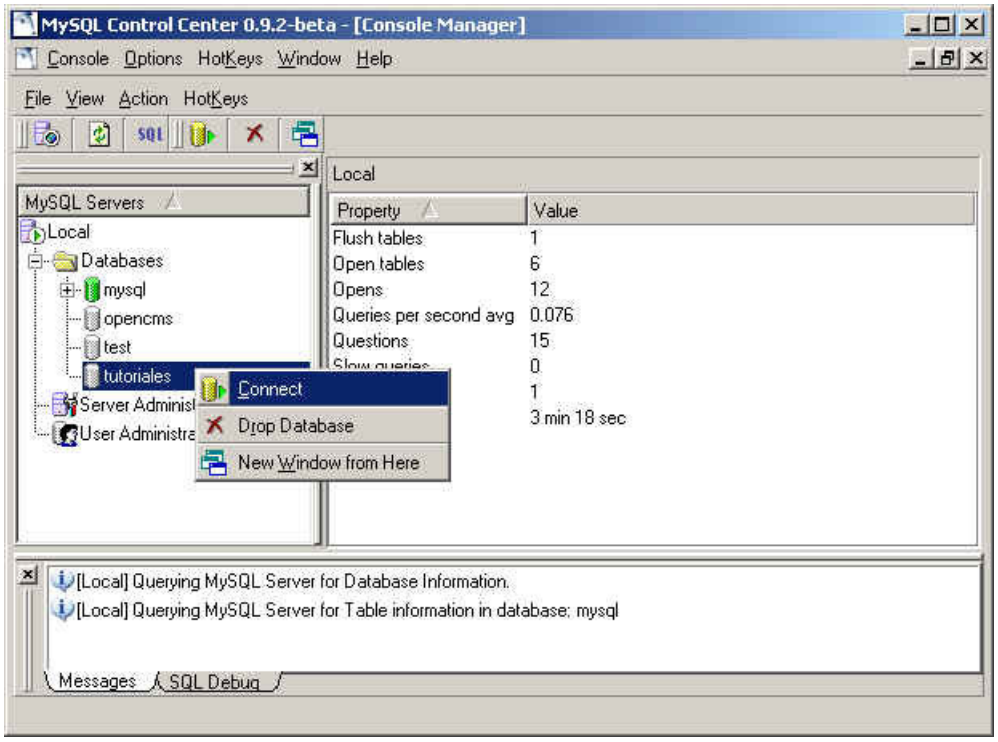
Creación de la base de datos

Ahora crearemos la base de datos donde almacenar la información sobre los tutoriales.

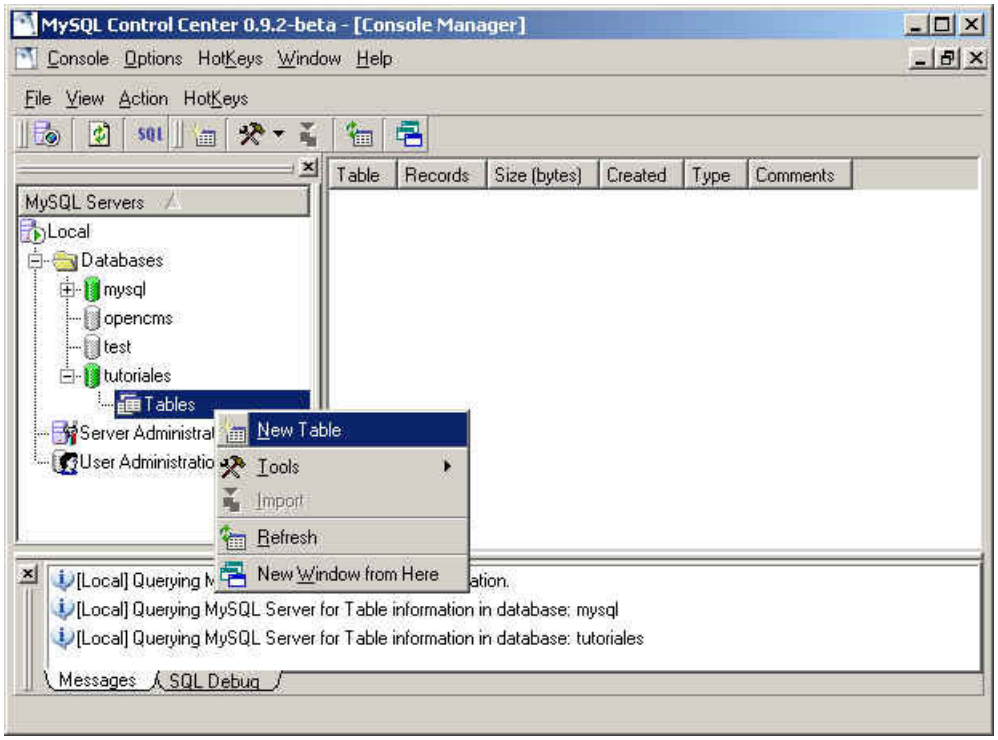
Nos vamos al Control Center de MySQL (ver tutorial asociado)



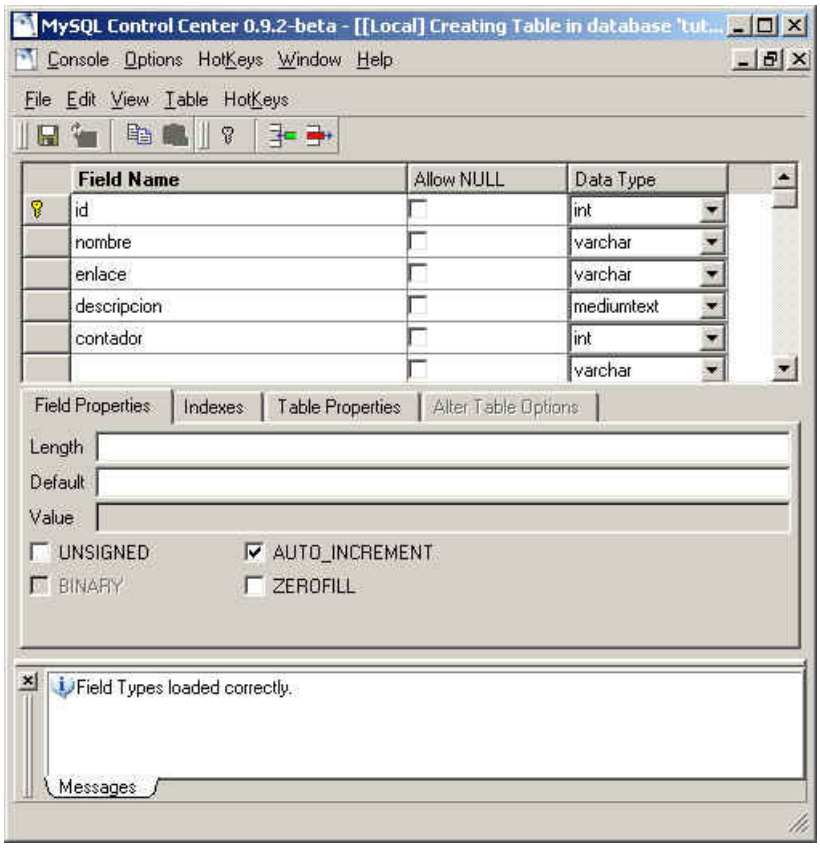
Conectamos a ella



Creamos las tablas en la base de datos



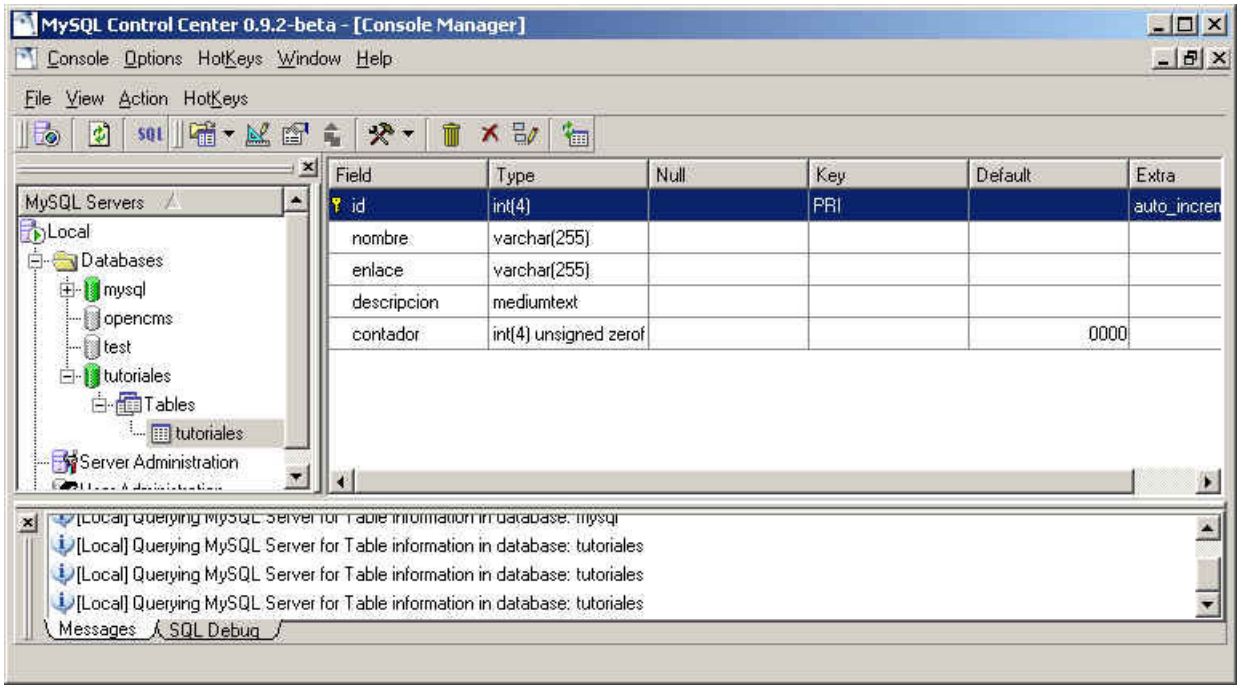
Introducimos todos los campos



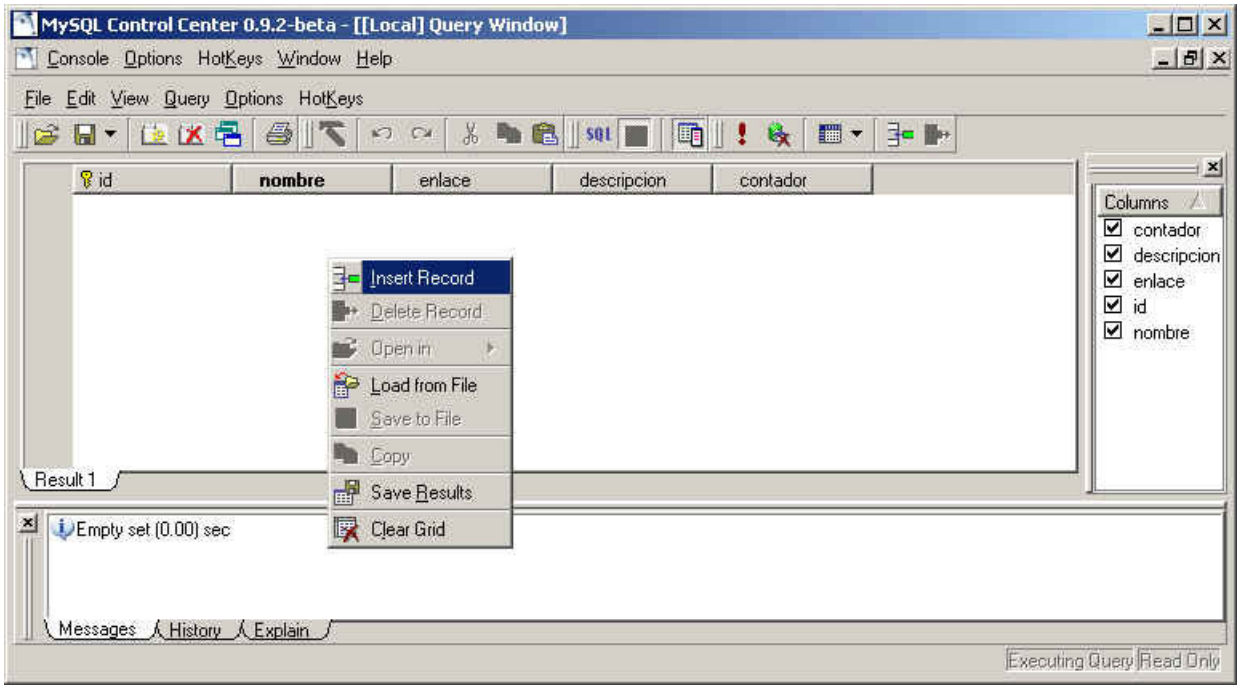
Guardamos y damos nombre a la tabla

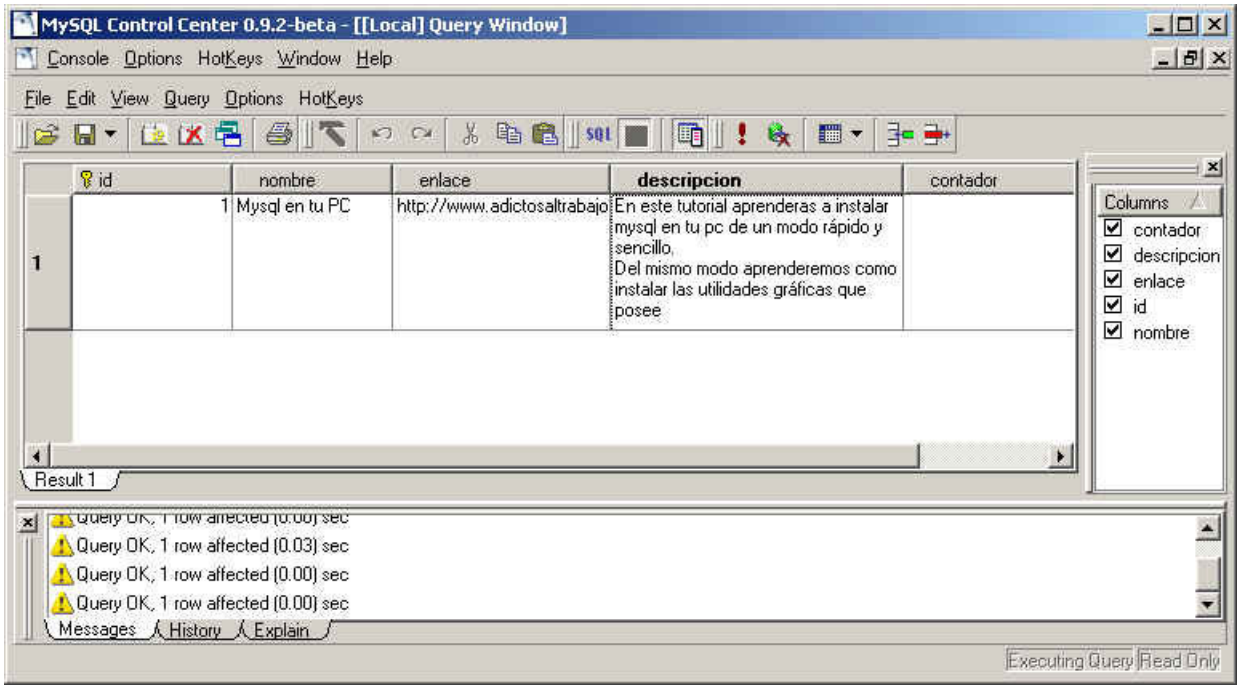


Vemos el resultado generado



Ahora hacemos doble click sobre el nombre de la tabla ... e insertamos registros





Uso de JavaBeans

Ahora, podríamos tratar de insertar el código en nuestro JSP para ir a la base de datos y buscar los tutoriales que tengan dentro de su descripción una de las palabras clave.

Obviamente, no deberíamos mezclar la lógica de presentación y la lógica de negocio por lo que vamos a meter el acceso a la base de datos en un componente: Un JavaBean.

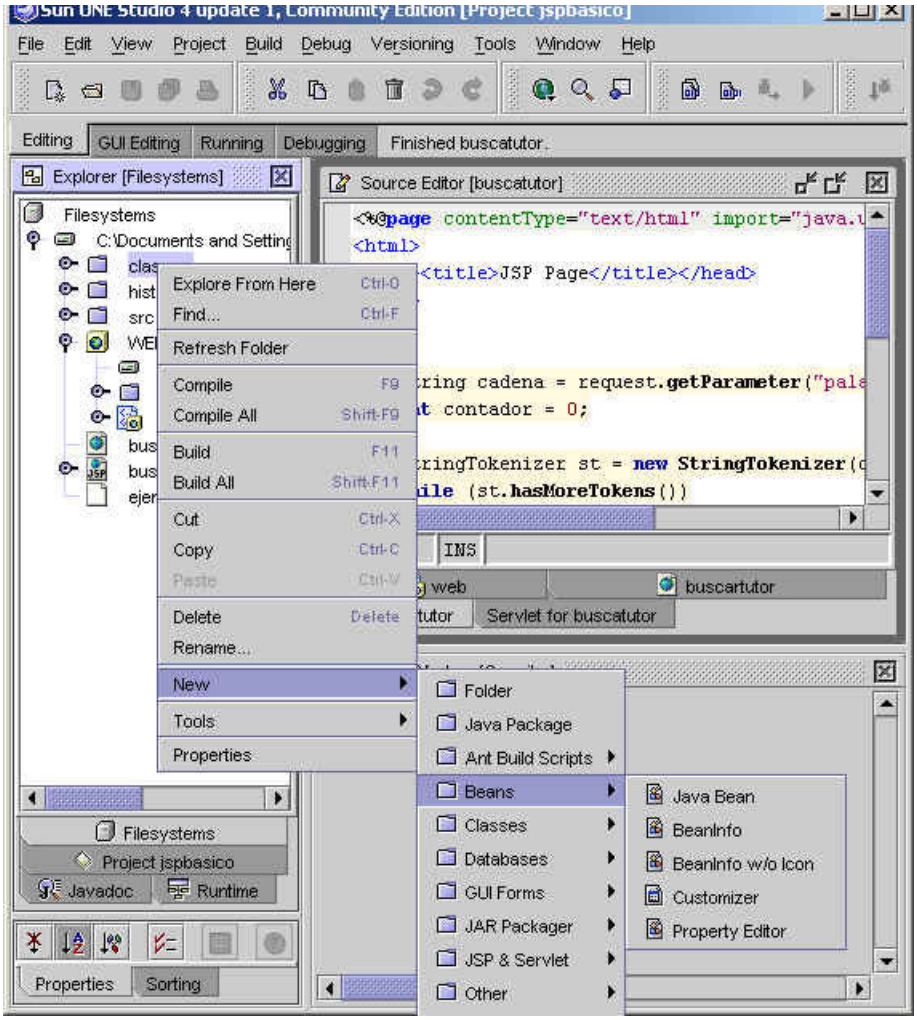
Un JavaBean es una clase Java donde se deben cumplir algunas reglas básicas (que veremos en otro tutorial). Lo mejor es que creemos uno básico.

El funcionamiento será simple ... insertaremos un JavaBean en nuestro JSP. Le pasaremos como parámetro la variable que nos pasen desde el formulario y es Bean recuperará los datos de la base de datos y nos proporcionará métodos para saber los que hay e iterar por ellos.

Creamos un paquete donde meter nuestros Beans



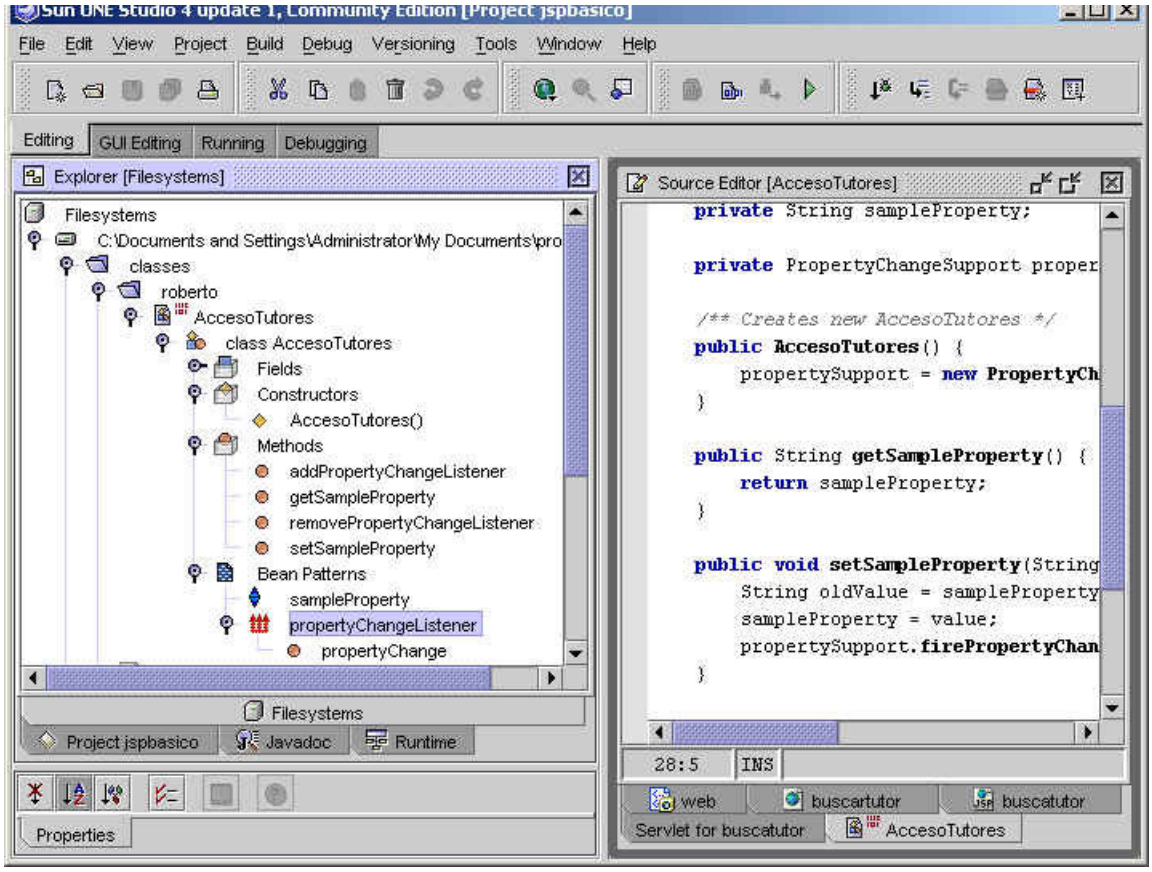
Creamos el Bean



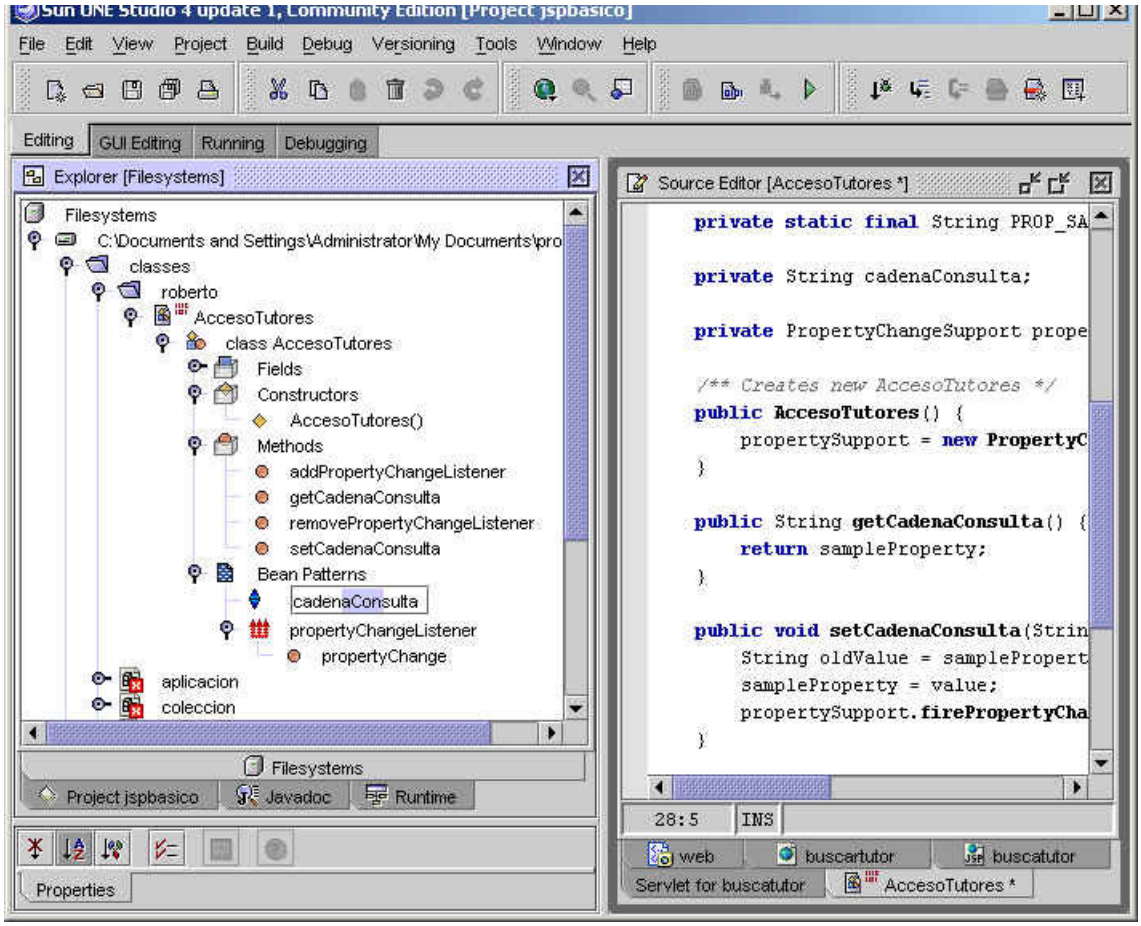
Le asignamos un nombre



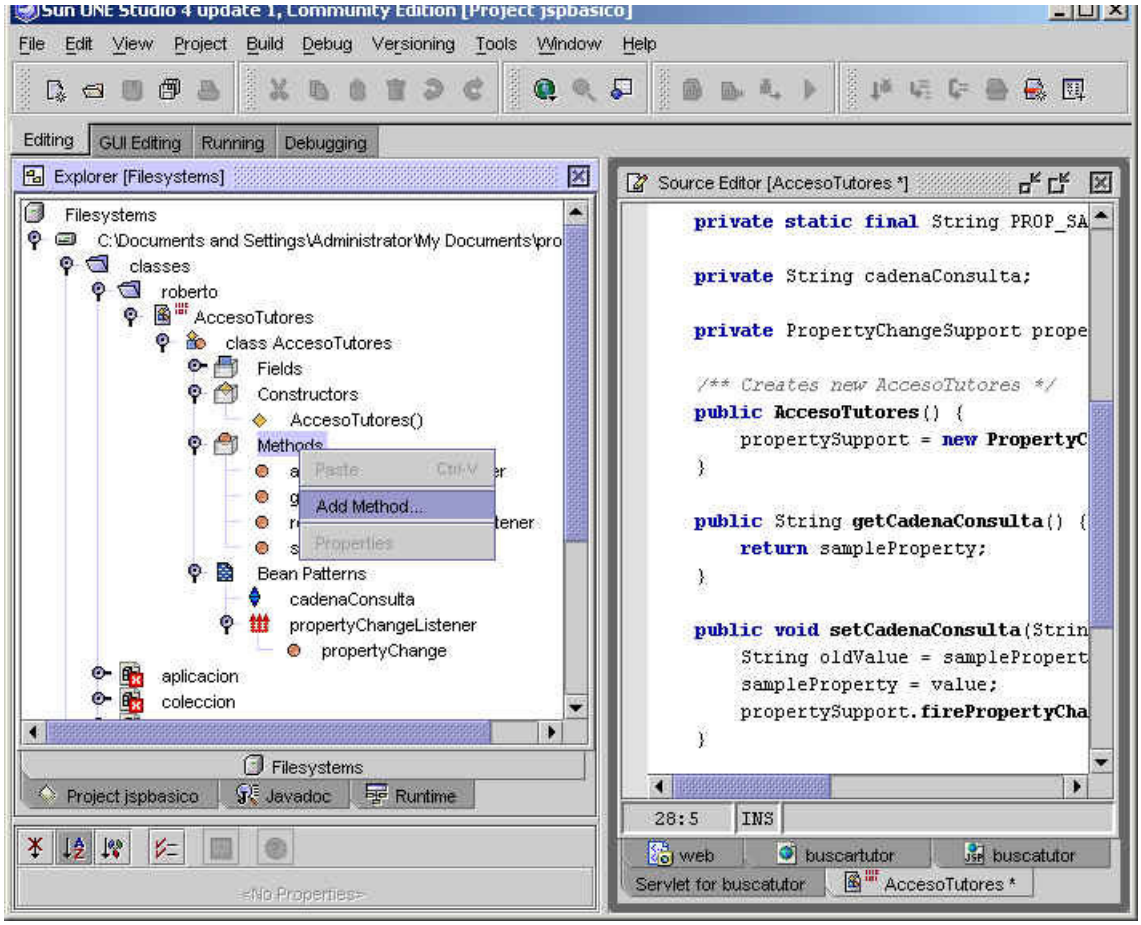
Y vemos que se ha creado un monton de código de ejemplo ....



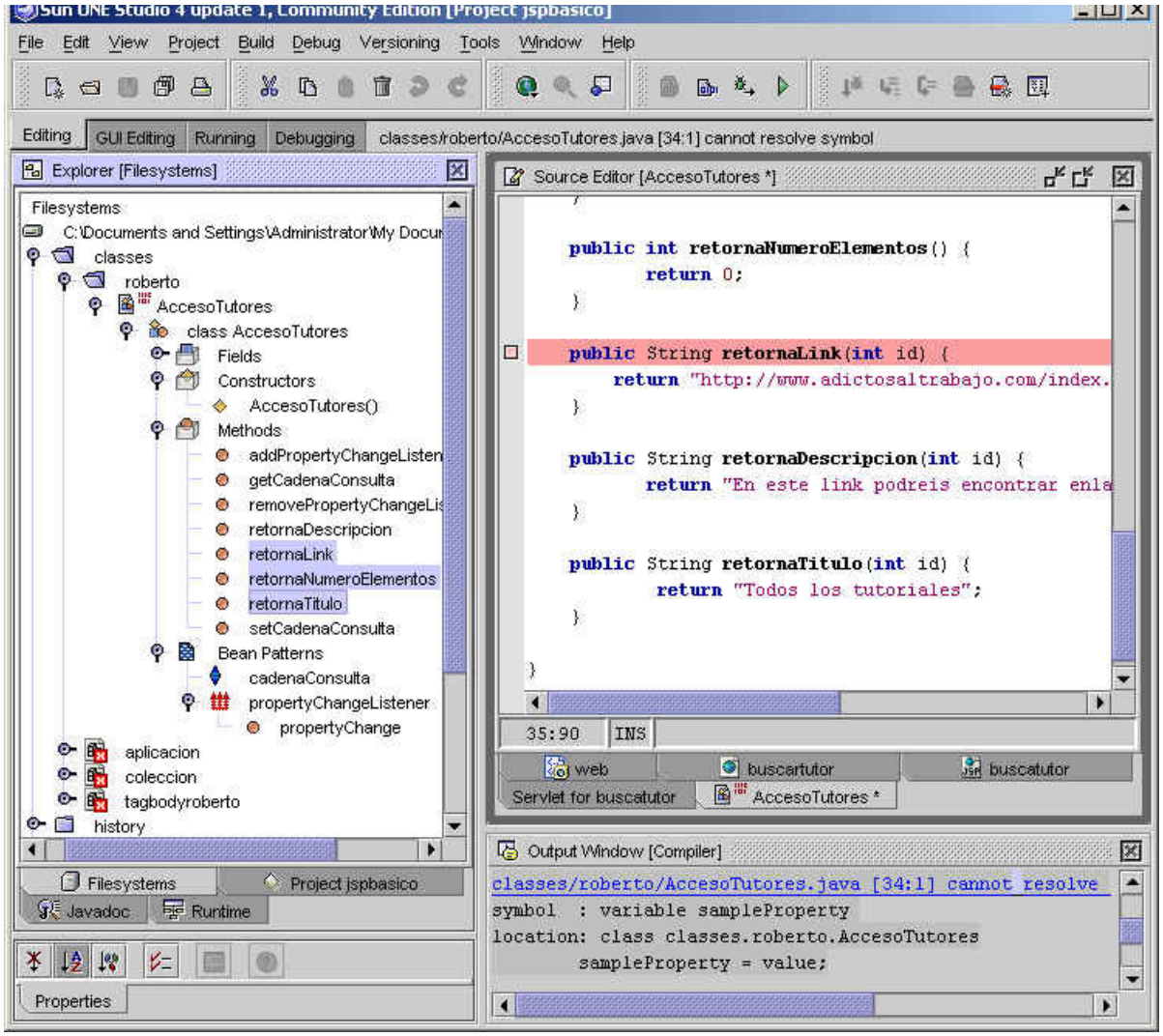
Si cambiamos el nombre de la propiedad .... todo cambiara automáticamente (bueno ... no es perfecto y algunas cosas se le olvidan y hay que cambiarlas a mano). Ya tenemos una propiedad que se llama Cadena de Consulta.



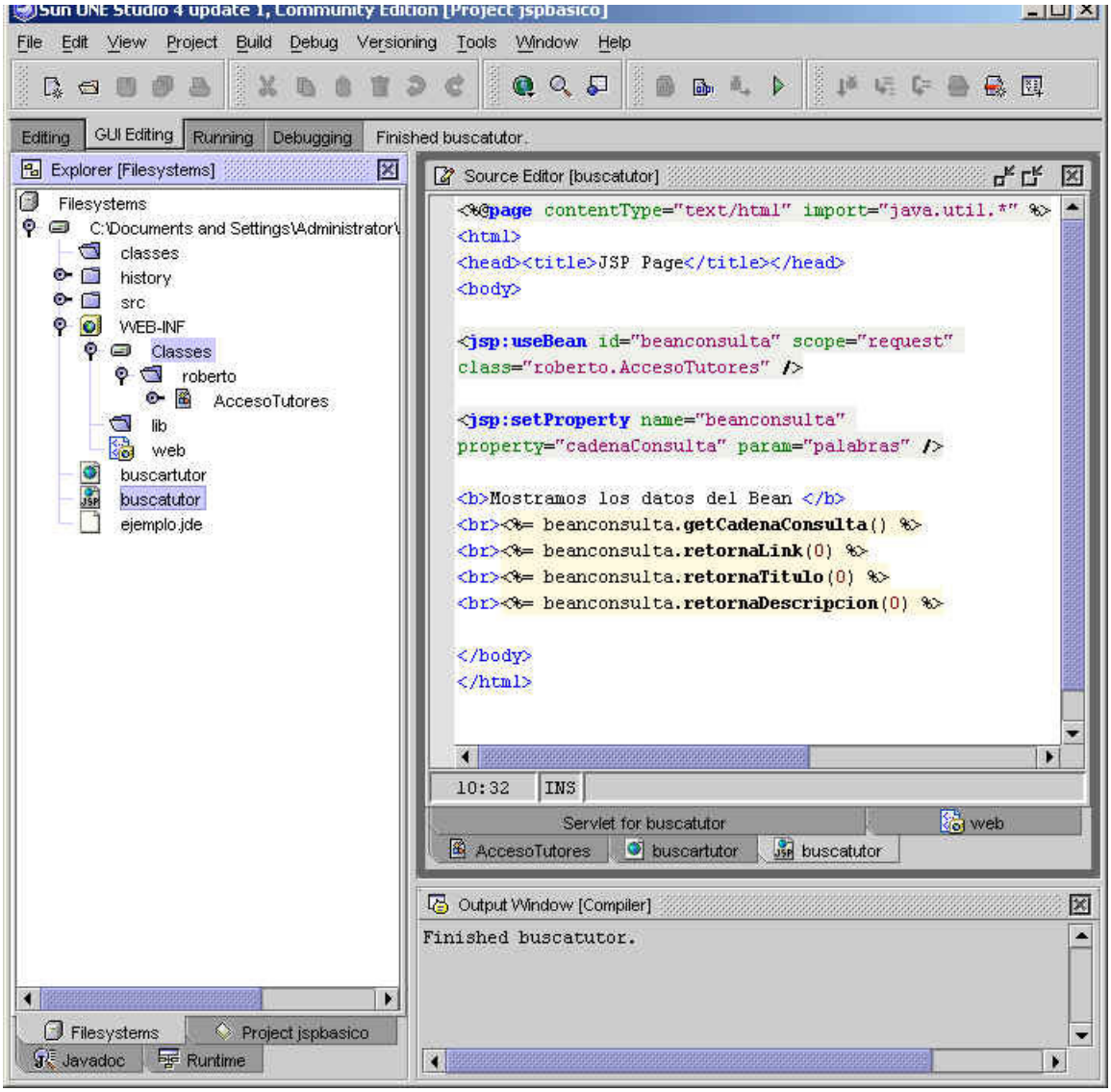
Ahora añadimos el método que nos retorne la cantidad de elementos



Insertamos cada uno de los métodos con datos de pruebas

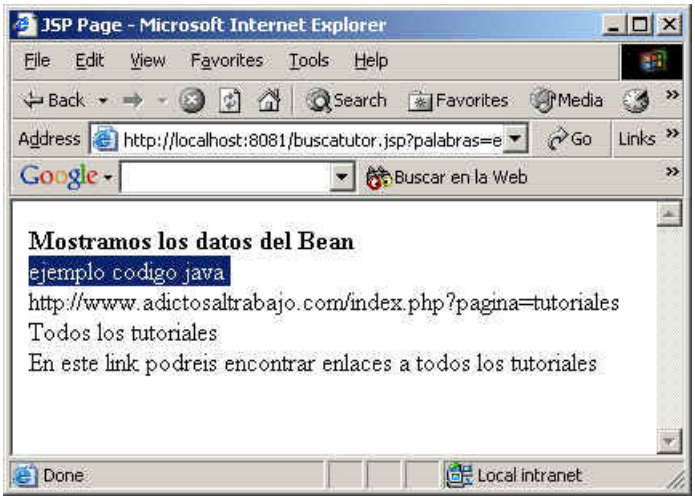


Ahora modificamos nuestro JSP para incluir el Bean .... tambien debemos darnos cuenta que hemos cambiado de sitio nuestro Bean en el proyecto para que cuelgue de la carpeta WEB_INF.



Si reejecutamos el código .... veremos que ya lo tenemos .....

Vemos que se pasa bien el parámetro .... y que somos capaces de recuperar las cadenas de vuelta .....

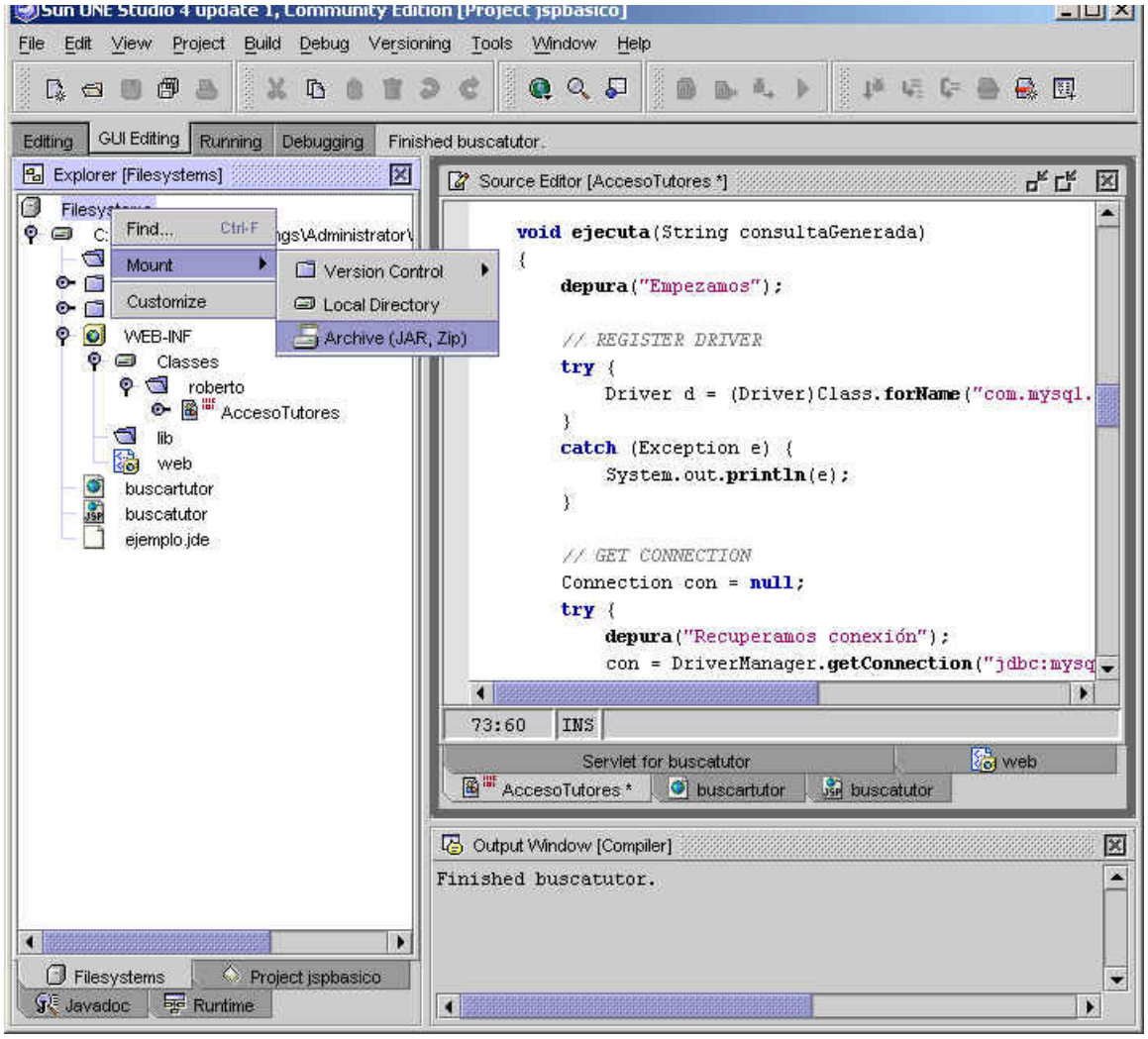


Ahora solo tenemos que ligar nuestro Bean con la base de datos ..... Esto ya es programación tradicional.

Podeir revisar algunos de los tutoriales que tenemos que os ensaña com trabajar con Java y MySql

De todos modos .. vamos a completar de un modo rápido la aplicación.

Incluimos el path donde se encuentra el driver JDBC de mysql



Una vez que nos aparece en pantalla... lo copiamos y pegamos sobre el directorio WEB-INF/Classes

Hemos insertado el unas nuevas funciones para recuperar los datos de la consulta.

ESTE NO ES EL MODO OPTIMO DE HACERLO... ES SOLO UN EJEMPLO

```
public String retornaConsulta() {
   return ejecuta("select * from tutoriales");
}


void depura(String mensaje) {
   System.out.println(mensaje);
}

String ejecuta(String consultaGenerada)
{
   depura("Empezamos");

   // REGISTER DRIVER
   try
   {
      Driver d = (Driver)Class.forName("com.mysql.jdbc.Driver").newInstance();
   }
   catch (Exception e)
   {
      System.out.println(e);
   }

   // GET CONNECTION
   Connection con = null;

   try
   {
      depura("Recuperamos conexión");
      con = DriverManager.getConnection("jdbc:mysql://localhost/tutoriales","","");
   }
   catch(Exception e)
   {
      System.out.println(e);
   }

   // GET CONNECTION WARNINGS
```

```
    SQLWarning warning = null;
    try
    {
       warning = con.getWarnings();

       if (warning == null)
       {
          System.out.println("No Warnings");
          // return;
       }

       while (warning != null)
       {
          System.out.println("Warning: "+warning);
          warning = warning.getNextWarning();
       }
    }
    catch (Exception e)
    {
       System.out.println(e);
    }

    // CREATE STATEMENT
    java.sql.Statement stmt = null;

    try
    {
       stmt = con.createStatement();
    }
    catch (Exception e)
    {
       System.out.println(e);
    }

    // EXECUTE QUERY
    ResultSet results = null;
    try
    {

       results = stmt.executeQuery(consultaGenerada);
    }
    catch (Exception e)
    {
       System.out.println(e);
    }

    // GET ALL RESULTS
    StringBuffer buf = new StringBuffer();

    try
    {
       ResultSetMetaData rsmd = results.getMetaData();
       int numCols = rsmd.getColumnCount();
       int i, rowcount = 0;

       // get column header info
       for (i=1; i <= numCols; i++) {
          if (i > 1) buf.append(",");
          buf.append(rsmd.getColumnLabel(i));
       }
       buf.append("\n");

       // break it off at 100 rows max
       while (results.next() && rowcount < 100) {
          // Loop through each column, getting the column
          // data and displaying

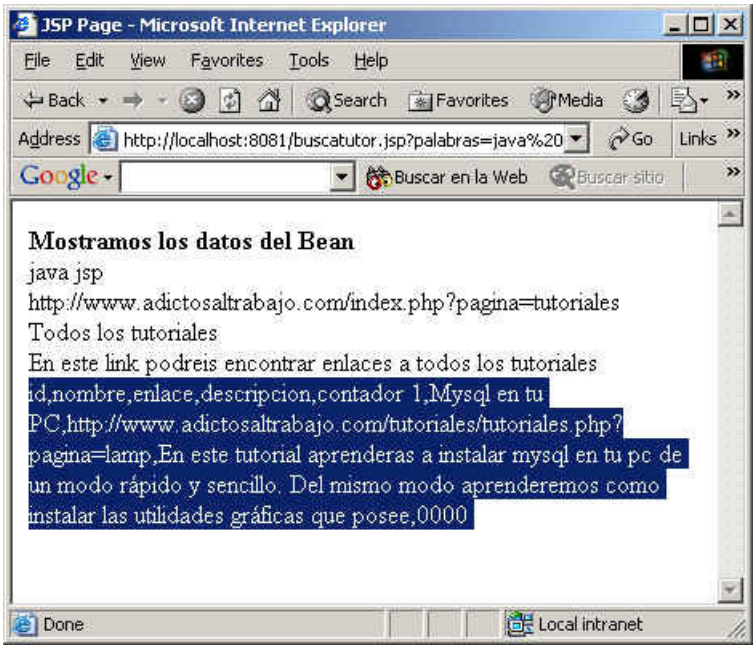
          for (i=1; i <= numCols; i++) {
             if (i > 1) buf.append(",");
             buf.append(results.getString(i));
          }
          buf.append("\n");
          rowcount++;
       }

       System.out.println(buf);
       results.close();

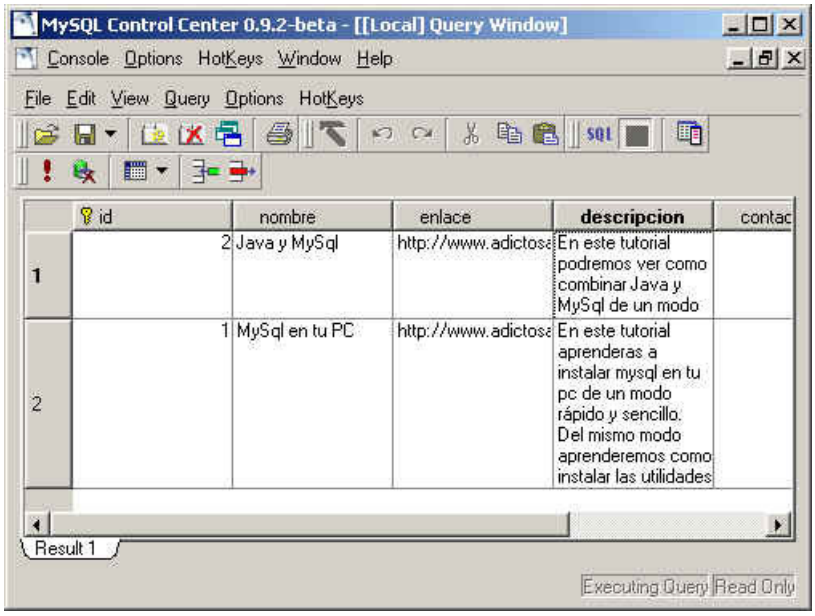
    }
    catch (Exception e)
    {
       System.out.println(e);
       return e.getMessage();
    }

    depura("Finalizamos");
    return buf.toString();
 }
```

Con esto podemos ver en pantalla lo que se retorna .. que no es poco .... para ver como generar este código de modo automático podemos visitar otro tutorial ..... JDBC automático



Insertamos algún registro más en la base de datos



Cambiando el código de nuestro Bean

Y nuestro JSP se vería como

```
<%@page contentType="text/html" import="java.util.*" %>
<html>
<head><title>JSP Page</title></head>
<body>

<jsp:useBean id="beanconsulta" scope="request"
class="roberto.AccesoTutores" />

<jsp:setProperty name="beanconsulta"
property="cadenaConsulta" param="palabras" />

Tutoriales disponibles para la consulta:<b><br> <%= beanconsulta.retornaConsulta() %> </b>
<br>

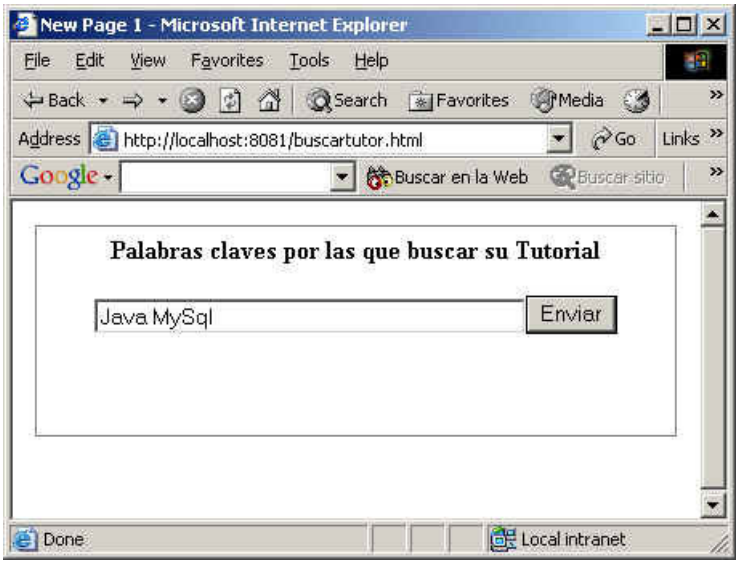
<%
int numeroElementos = beanconsulta.retornaNumeroElementos();

for (int i=0; i < numeroElementos; i++)
{
%>
<br><%= beanconsulta.retornaLink(i) %>
<br><%= beanconsulta.retornaTitulo(i) %>
<br><%= beanconsulta.retornaDescripcion(i) %>
<br><br>
<%
```

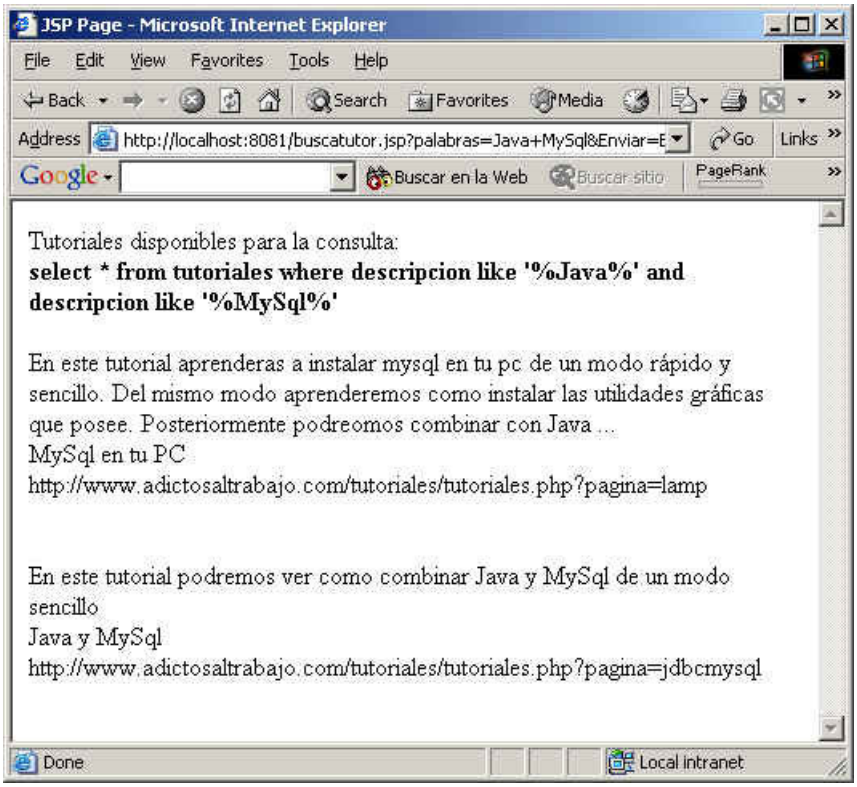
```
}
%>

</body>
</html>
```

Realizamos la petición:



Y obtenemos la respuesta



Ahora solo nos queda darle un toque para que el diseño gráfico sea un poco más agradable.

```
<%@page contentType="text/html" import="java.util.*" %>
<html>
<head><title>JSP Page</title></head>
<body>

<jsp:useBean id="beanconsulta" scope="request"
class="roberto.AccesoTutores" />

<jsp:setProperty name="beanconsulta" property="cadenaConsulta" param="palabras" />

<% beanconsulta.retornaConsulta(); %>

<p><b>Resultado de la Consulta:  </b>Encontrados <b> <%= beanconsulta.retornaNumeroElementos() %></b>
resultados.</p>
```

```
<br>
<table border="1" width="90%" bordercolor="#FFFFFF" cellspacing="0" cellpadding="0" height="41"
bgcolor="#800000">
<tr>
<td width="33%" align="center" height="18" bordercolor="#800000"><b><font
color="#FFFFFF">Título</font></b></td>
<td width="67%" align="center" height="18" bordercolor="#800000"><b><font
color="#FFFFFF">Descripción</font></b></td>
</tr>

<%
int numeroElementos = beanconsulta.retornaNumeroElementos();

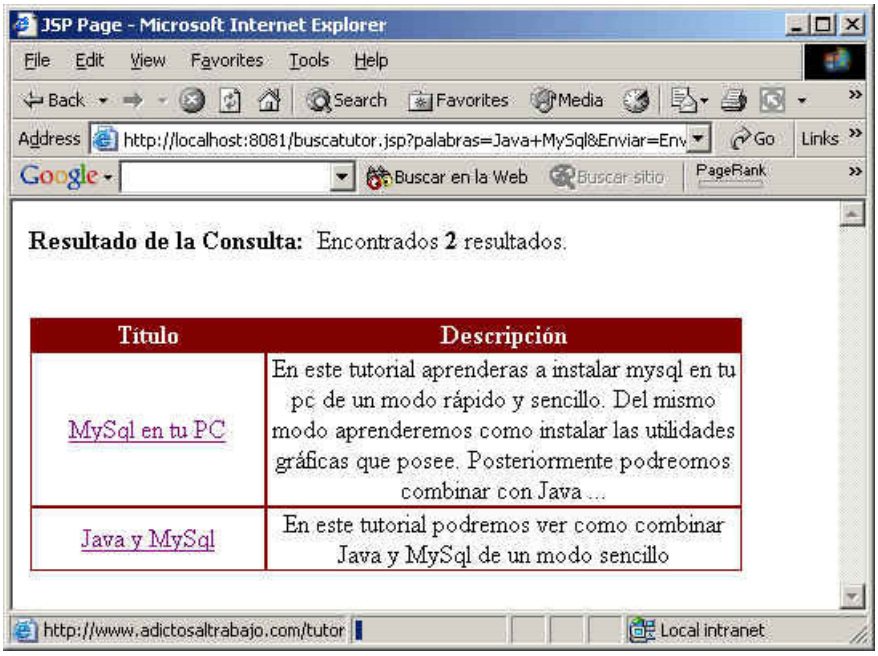
for (int i=0; i < numeroElementos; i++)
{
%>
<tr>
<td width="33%" align="center" height="18" bordercolor="#800000" bgcolor="#FFFFFF">

<a href="<%= beanconsulta.retornaLink(i) %>"><%= beanconsulta.retornaTitulo(i) %></a>

</td>
<td width="67%" align="center" height="18" bordercolor="#800000" bgcolor="#FFFFFF"><%=
beanconsulta.retornaDescripcion(i) %>

</td>
</tr>
<%
}
%>

</table>
</body>
</html>
```



Bueno ..... creo que para ser un primer ejemplo ... hemos hecho muchas cosas .... en próximos tutoriales veremos más ejemplo del desarrollo con JSPs y la combinación con otros componentes.

Ultimo cambio 06/25/2003

Sobre el Autor ..

Si desea contratar formación, consultoria o desarrollo de piezas a medida puede contactar con

J2EE, EJBs, Struts...

Autentia S.L. Somos expertos en:
J2EE, C++, OOP, UML, Vignette, Creatividad ..
y muchas otras cosas

Nuevo servicio de notificaciones

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

Subscribirse a Novedades	
e-mail	
	Enviar

Otros Tutoriales Recomendados (También ver todos)

Nombre Corto	Descripción
Ejecutar JSPs almacenados en Base de Datos	Atendiendo una pregunta de nuestro foro, os mostramos como, con unos sencillos pasos, podemos ejecutar JSPs almacenandos en la base de datos. Esto puede ser una idea base para un gestor de contenidos construido en Java.
Control navegación en Servlets	Os mostramos como construir el esqueleto de una aplicación basada en Servlets y JSP, con control de navegación.
Control navegación en Servlets	Os mostramos como construir el esqueleto de una aplicación basada en Servlets y JSP, con control de navegación.
Cachear porciones de JSPs	En este tutorial os ensañamos como incrementar increiblemente el rendimiento de vuestro Web basado en tecnología JSP con el FrameWork de cache OSCACHE
JSP´s y Modelo-Vista-Controlador	En este tutorial os enseñamos como crear un JSP, su relación con los servlets y como crear un ejemplo MVC en Tomcat
Repositorio CVS en Windows	Os mostramos como montar un servidor para el control de versiones CVS en Windows asi como acceder a él a través de WinCVS
Forzar diálogo Guardar Como en JSPs	Os mostramos como afrontar tareas comunes en JSP: Forzar el diálogo Guardar Como al generar dinámicamente un fichero desde un JSP y asegurarnos que no se cachean nuestros ficheros (probado en IExplorer 6)
Gestión errores en JSPs	Os mostramos como realizar ciertas labores intermedias en JSPs: Comentarios, gestión de errores, formateo de fechas y precompilación de ficheros
Generar imagenes desde Servlets	Os mostramos como generar ficheros GIF desde un servlet java. Util para generear gráficas dinámicas, contadores, etc
JSP 2.0, JSTL y Lenguaje de expresiones	Os mostramos las novedades de JSP 2.0: Nuevas librerías estandar de etiquetas y el lenguaje de expresiones con ejemplos de acceso a base de datos, XML y XSL en JSP

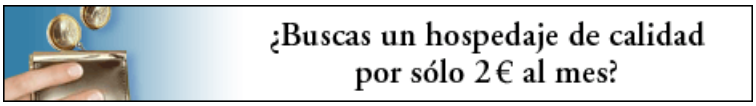
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores.

En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo.

Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.

Patrocinados por enredados.com .... Hosting en Castellano con soporte Java/J2EE



www.AdictosAlTrabajo.com Opimizado 800X600