

Somos su empresa de Soporte a Desarrollo Informático

Ese apoyo que siempre quiso tener

- Desarrollo de componentes y proyectos a medida.
- Auditoría de código y recomendaciones de mejora.
- Arranque de proyectos basados en nuevas tecnologías.
- Curso de Formación
 - Dirección de Proyectos Informáticos.
 - Gestión eficaz del Tiempo.
 - Arquitecturas de desarrollo Web: Web, J2EE, SOA, WebServices, BPM, etc.
 - Java/ J2EE a todos los niveles: JSPs, Servlets, EJBs, JMS, JNI, etc.
 - Análisis y diseño orientado a objeto.
 - UML y patrones de diseño.
 - Buenas prácticas en el desarrollo de aplicaciones
 - Técnicas avanzadas: Lucene, Hibernate, Spring, JSF, Struts, etc.

*Nuestra mejor referencia son los conocimientos que
compartimos en nuestro web*

www.adictosaltrabajo.com

Decenas de entidades cuentan ya con nosotros

Para más información visítenos en www.autentia.com

Tel. 91 675 33 06 - info@autentia.com

Empieza el jaleo: publicamos el libro

INFORMÁTICA PROFESIONAL

LAS REGLAS NO ESCRITAS PARA TRIUNFAR EN LA EMPRESA

-¿Crees que la informática es sólo programar?

-¿Sabrías organizar eficientemente un equipo?

-¿Quién te resuelve las dudas sobre esta profesión?..... o te siembra más?

Hosting patrocinado por

Inicio

Quienes somos

Tutoriales

Formación

Comparador de salarios

Comentar libro

Charlas

Más

Estas en: Inicio » Tutoriales » Realización de in informe con JasperReports-Ireport

Ultimas Noticias

- » Formación por calendario
- » El próximo día 22 a las 19:30 vamos a celebrar el primer Coding Dojo organizado por agilismo.es
- » Comentando el libro: Eneagrama y éxito personal de Ginger Lapid-Bogda
- » "Informática profesional: Las reglas no escritas para triunfar en la empresa." Pincha para comprar..
- » Disponible la primera versión de los plugins para integrar Maven y Bugzilla.
- » Lanzamiento del nuevo Web de Autentia

+ Noticias Destacadas

- » "Informática profesional: Las reglas no escritas para triunfar en la empresa." Pincha para comprar..
- » Autentia patrocina el primer Agile Open Spain
- » Lanzamiento del nuevo Web de Autentia
- » Autentia cumple 6 años

+Comentarios Cómic

+Enlaces

Catálogo de servicios Autentia (PDF 6,2MB)



En formato comic...

Tutorial desarrollado por



Saúl García Díaz

Consultor tecnológico de desarrollo de proyectos informáticos.

Puedes encontrarme en Autentia

Somos expertos en Java/JEE

Catálogo de servicios de Autentia

Descargar (6,2 MB)

Descargar en versión comic (17 MB)

AdictosAlTrabajo.com es el Web de difusión de conocimiento de Autentia.



Catálogo de cursos

Descargar este documento en formato PDF: [javaBeanDataSource.pdf](#)

Acceso de usuarios registrados:

E-mail:

Contraseña:

[Deseo registrarme](#)

[He olvidado mis datos de acceso](#)

Registra tu empresa:

Descubre las ventajas de registrar tu empresa en AdictosAlTrabajo...

[Registrar mi empresa](#)

[Listado de empresas ya registradas](#)



Web

www.adictosaltrabajo.com

Últimos tutoriales

2009-12-14
[Realización de in informe con JasperReports-Ireport](#)

2009-12-11
[Contract-First web services con Visual Studio 2008](#)

2009-12-09
[Integrando Sonar con Hudson](#)

2009-12-09
[Apache + Tomcat: Balanceo de carga y alta disponibilidad](#)

2009-12-08
[MySQL: Replicación de bases de datos en MySQL](#)

Fecha de creación del tutorial: 2009-12-14

JavaBean DataSource Ireport

0. Índice de contenidos.

- 1. Introducción.
- 2. Entorno
- 3. Importando a Eclipse
- 4. La Aplicación
- 5. El informe
- 6. Comprobando el resultado
- 7. Conclusiones

1. Introducción

En esta ocasión vuelvo manos a la obra para explicar cómo realizar un informe con JasperReports-Ireport. Como ya habréis visto en adictos existen muy buenos tutoriales al respecto, así que espero aportar mi granito de arena al maravilloso mundo de generación de informes.

La particularidad del caso que nos ocupa, es conseguir que la fuente de datos del informe sea una lista de JavaBeans y no una consulta definida previamente en el informe. Esta última opción es una manera muy habitual de definir la fuente de datos, sin embargo, en ocasiones puede que no sea la opción que maás se adapte a nuestras necesidades.

Por ejemplo, imaginemos que nuestra aplicación gestiona la generación de informes mediante un servlet que tiene establecido un datasource en concreto pero uno de los informes no recupera la información que debe mostrar de ese datasource. Una solución podría ser crear un servlet para cada datasource y que el informe fuese el encargado de ejecutar la consulta. Pero seguramente no merezca la pena desarrollar un nuevo servlet solo para un informe.

Otra solución es la que nos ocupa, es decir, recuperamos toda la información desde la aplicación, conformamos una lista de objetos con toda esta información e indicamos al informe que ese será su datasource. Esto es un ejemplo pero seguro que podéis aplicar esta forma de definir el datasource de un informe en más de una ocasión.

A continuación vemos cual será cada uno de los pasos a seguir.

2. Entorno

El tutorial está escrito usando el siguiente entorno:

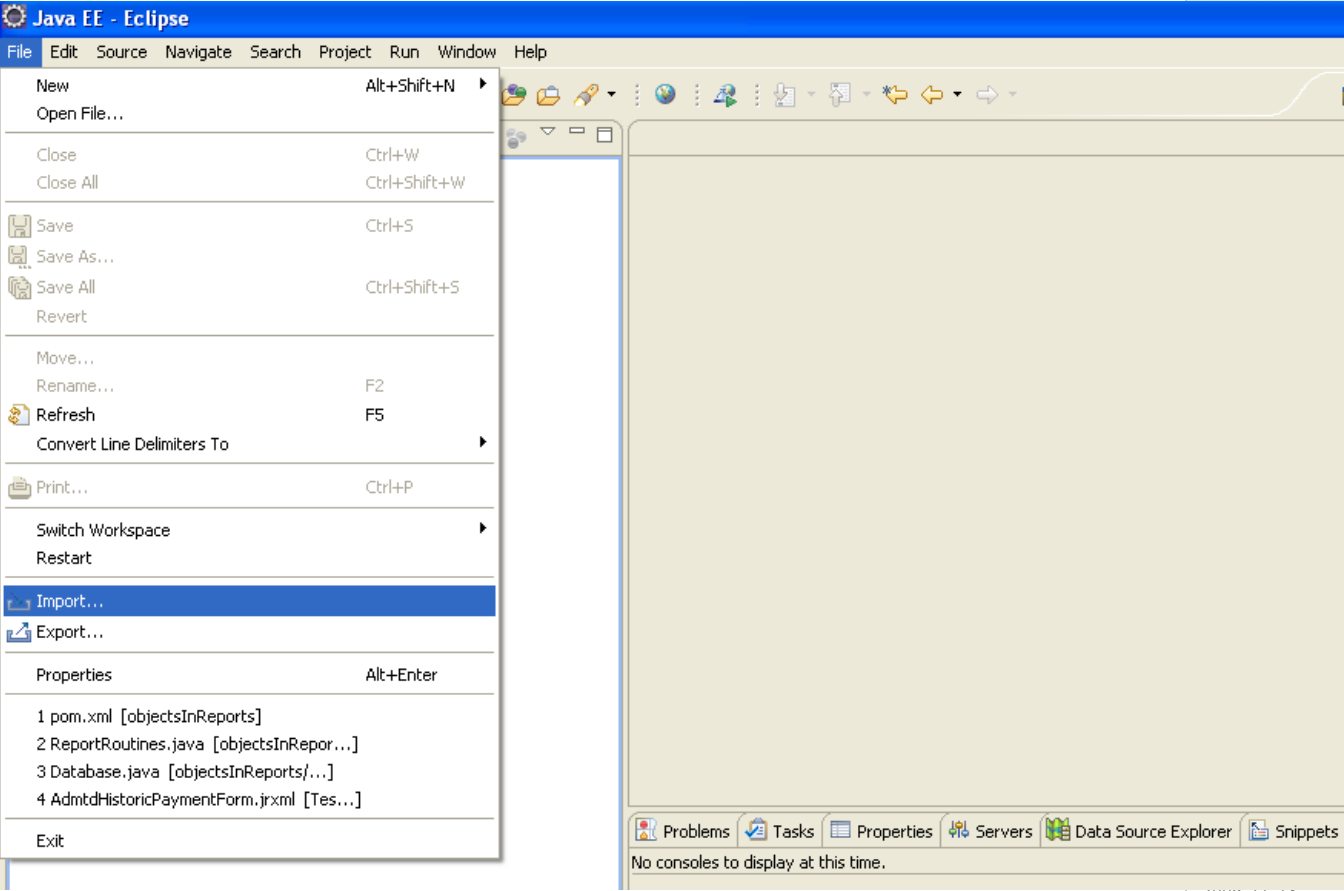
- Hardware: Portátil Dell Latitude E5500(Core Duo T9550 2.66GHz, 4GB RAM, 340 GB HD)
- Sistema operativo: Windows XP
- JDK 1.6.0_1
- Eclipse Galileo
- Maven 2.1.0
- MySQL 5.1.40-Community

- JasperReport 3.0.0
- Ireport 3.5.2

3. Importando a Eclipse

Lo primero de todo para poder probar la aplicación es descargarse los fuentes de [aquí](#). Hay que tener en cuenta que para el ejemplo he usado MySQL, por tanto si utilizáis cualquier otra base de datos será necesario modificar el fichero de dependencias y la clase Database.java para establecer correctamente la conexión. Una vez descargados los fuentes:

- Importamos el proyecto desde eclipse



- Seleccionamos Maven 2 Project y pulsamos Next

2009-12-07
[Analizando la calidad del código Java con Sonar](#)

2009-12-03
[Instalar OpenESB 2.1 e Introducción](#)

[File Forms](#)

[Cando y enido desde nes Java](#)

[Cando nuestras a](#)

[nar Google ating System xp](#)

[l tamaño de VirtualBox](#)

[esentaciones. da Zen.](#)

[e google](#)

2009-11-13
[Introducción a Escritorios Animado \(Winamp y MilkWave \)](#)

2009-11-12
[Maven Assembly Plugin: empaquetando aplicaciones con Maven para la ejecución de procesos batch.](#)

2009-11-12
[Tutorial básico sobre Google Application Engine](#)

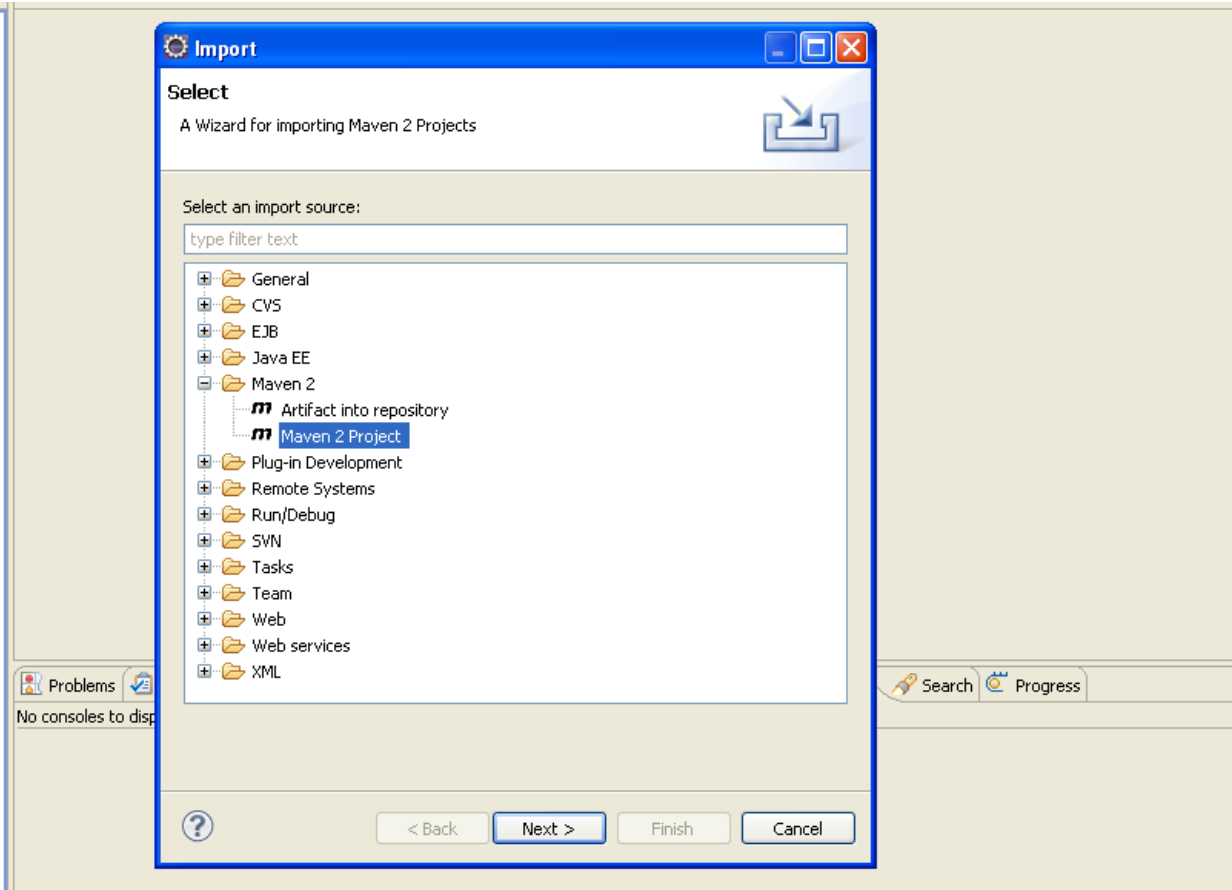
2009-11-11
[Instalación de Glassfish 2.1](#)

2009-11-26
[Rational Software Architect y Rational RequisitePro](#)

2009-10-27
[Cómo conseguir que Subversion avise a Hudson para lanzar una build](#)

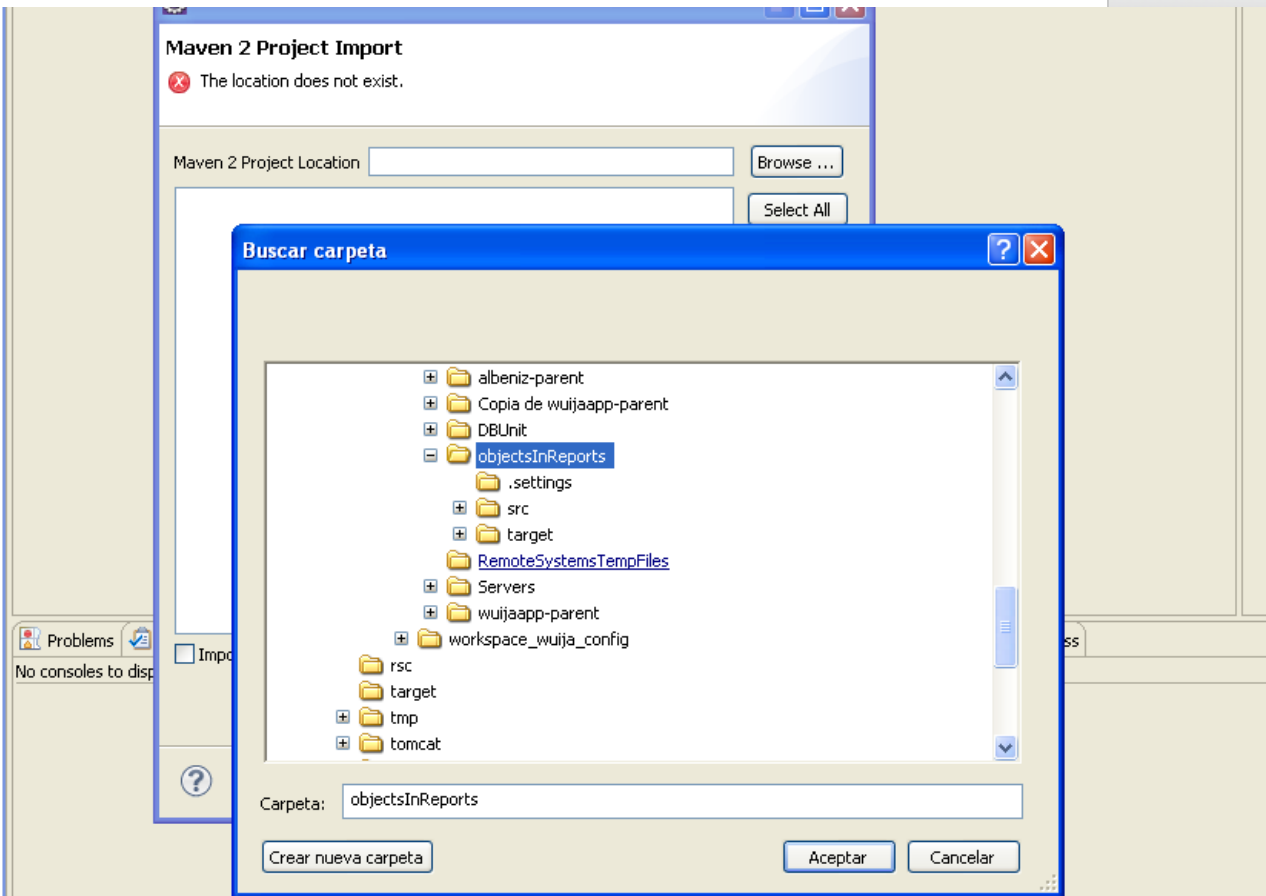
2009-10-26
[Cómo instalar Hudson en Apache Tomcat](#)

2009-10-26
[Estilos externos en iReport](#)



- A continuación, buscamos el proyecto dentro del workspace en que tengamos descomprimido los fuentes

2009-10-14
[Echo en MySQL](#)

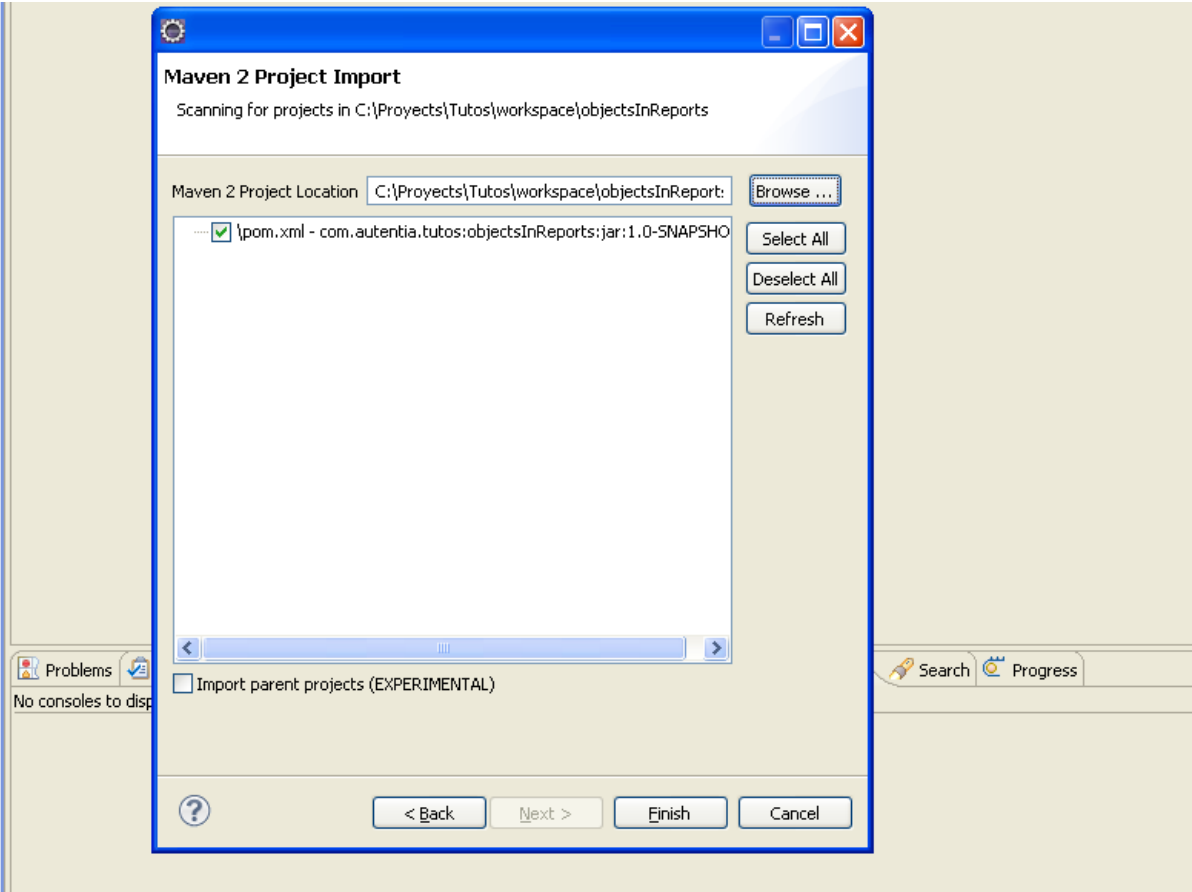


- Por último pulsamos finish.

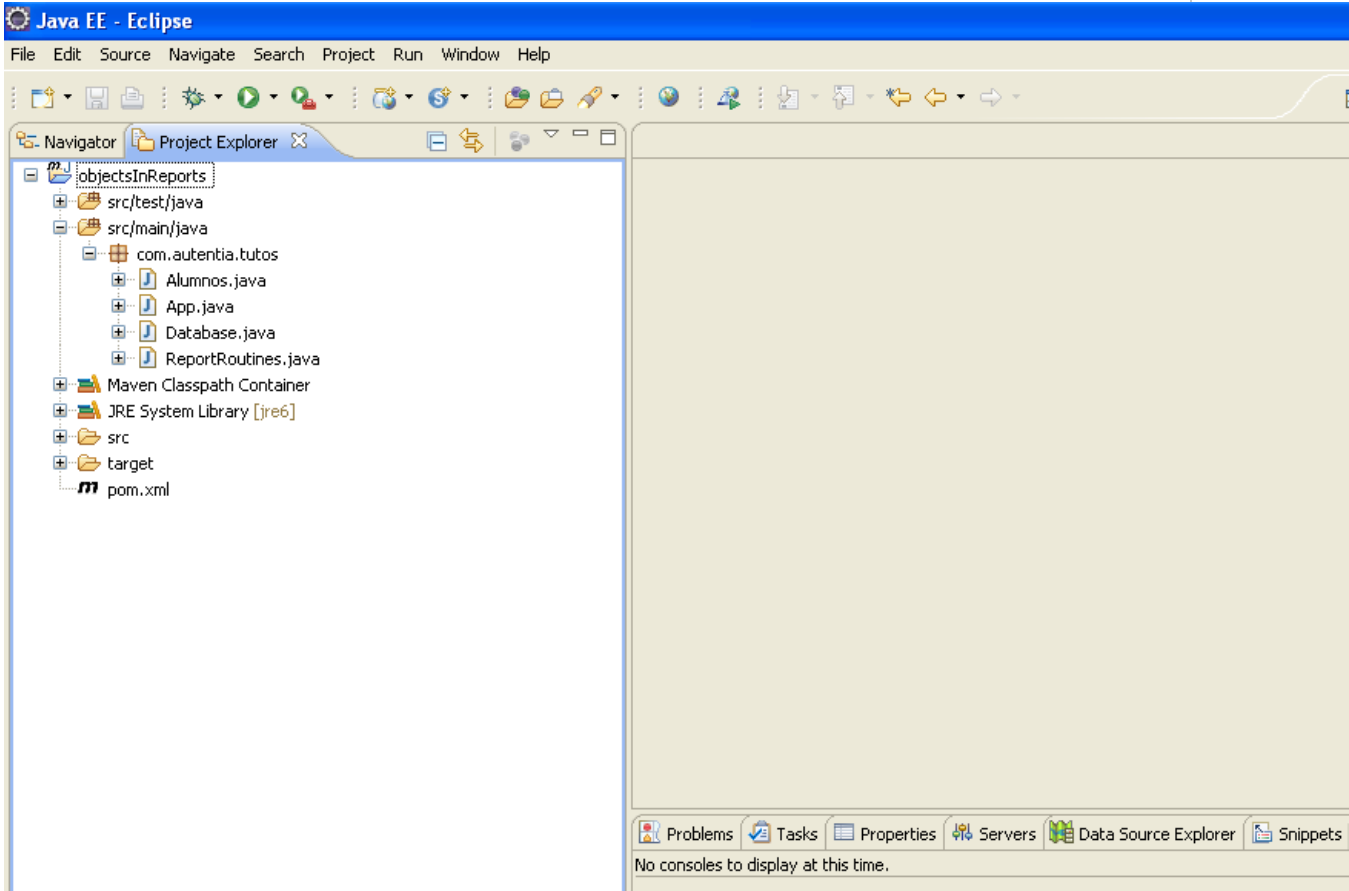
2009-09-12
[Instalación de Liferay en Tomcat existente](#)

2009-09-11
[Release Bugzilla Maven Plugin](#)

2009-09-11



- Podemos observar como el proyecto se ha importado correctamente.



Una vez tenemos el proyecto en eclipse será necesario realizar un **mvn install** desde línea de comandos para que se descarguen las dependencias necesarias en caso de que todavía no las tengamos en nuestro repositorio local

4. La Aplicación

La aplicación del ejemplo es muy sencilla y se define en la clase App.java que vemos a continuación:

```
01. public class App
```

```
02. {
03.
04.
05.
06.
07. public static void main( String[] args ) throws SQLException
08. {
09.     Connection con=null;
10.     //Map con Los posibles otros parametros del informe
11.     HashMap map=new HashMap();
12.
13.     try {
14.         // Recumeramos La conxion a base de datos
15.         con =Database.getConnection();
16.         // Recuperamos La lista ojetos que mostraremos en el informe
17.         List alumnos=Database.getAlumnos(con);
18.         // Generamos el informe
19.         ReportRoutines.exportPDF(alumnos, map);
20.
21.
22.
23.
24.
25.     } catch (SQLException e) {
26.
27.         e.printStackTrace();
28.
29.     } catch (ClassNotFoundException e) {
30.
31.         e.printStackTrace();
32.
33.     } catch (JRException e) {
34.
35.         e.printStackTrace();
36.
37.     }finally{
38.         //Cerramos conexion con base de datos
39.         con.close();
40.     }
41. }
42.
43. }
```

Como podemos ver nuestra aplicación recupera la conexión con base de datos, lanza la consulta correspondiente y devuelve una lista de objetos (Alumnos.java). Finalmente se llama al método encargado de generar el informe pasándole como parámetros la lista de objetos y un map con el resto de posibles parámetros que necesitase el informe.

De las acciones realizadas, la que verdaderamente nos importa para el caso que nos ocupa es la acción de generar el informe. A continuación vemos la clase ReportRoutines.java, encargada de gestionar la generación de nuestro informe.

```
view plain print ?
01. public class ReportRoutines {
02.
03.
04. private static final String REPORT_PATH = "C:\\tutos\\reports\\test.jrxml";
05. private static final String REPORT_EXPORT_PATH = "C:\\tutos\\reports\\test.pdf";
06.
07. /**
08.  * Genera el pdf para el informe
09.  * @param aux = Lista de objetos que mostraremos en el informe
10.  * @param map = Resto de parametros que viajan al informe
11.  * @throws JRException
12.  */
13.
14. public static void exportPDF (List aux, HashMap map) throws JRException
15. {
16.     // Definimos cual sera nuestra fuente de datos
17.     JRBeanCollectionDataSource ds =new JRBeanCollectionDataSource(aux);
18.     // Recuperamos el fichero fuente
19.     JasperDesign jd=JRXmlLoader.load(REPORT_PATH);
20.     // Compilamos el informe jrxml
21.     JasperReport report = JasperCompileManager.compileReport(jd);
22.     // Rellenamos el informe con la conexion creada y sus parametros establecidos
23.     JasperPrint print = JasperFillManager.fillReport(report,map, ds);
24.
25.     // Exportamos el informe a formato PDF
26.     JasperExportManager.exportReportToPdfFile(print, REPORT_EXPORT_PATH);
27.
28. }
29.
30.
31. }
```

Como he dicho anteriormente aquí es donde esta lo verdaderamente importante para que nuestro informe funcione con el datasource formado por la lista de objetos recuperados en el paso anterior. Hacemos uso de la clase **JRBeanCollectionDataSource** para definir el datasource a partir de la lista de objetos, después basta con pasar ese datasource al método **JasperFillManager.fillReport**

5. El informe

Una vez que ya sabemos cómo definir el datasource con una lista de objetos nos dispondremos a crear el informe propiamente dicho. En este punto y como veremos a continuación indicaremos al informe que tipo de objeto será el que conforme la fuente de

2009-07-30
Funciones esenciales para crear un juego.

2009-07-30
2º tutorial TNT Concept versión 1.16.1

Últimas ofertas de empleo

2009-07-31
T. Información - Operador (día / noche) - BARCELONA.

2009-06-25
Atención a cliente - Call Center - BARCELONA.

2009-06-19
Otras - Ingeniería (minas, puentes y puertos) - VALENCIA.

2009-06-17
Comercial - Ventas - ALICANTE.

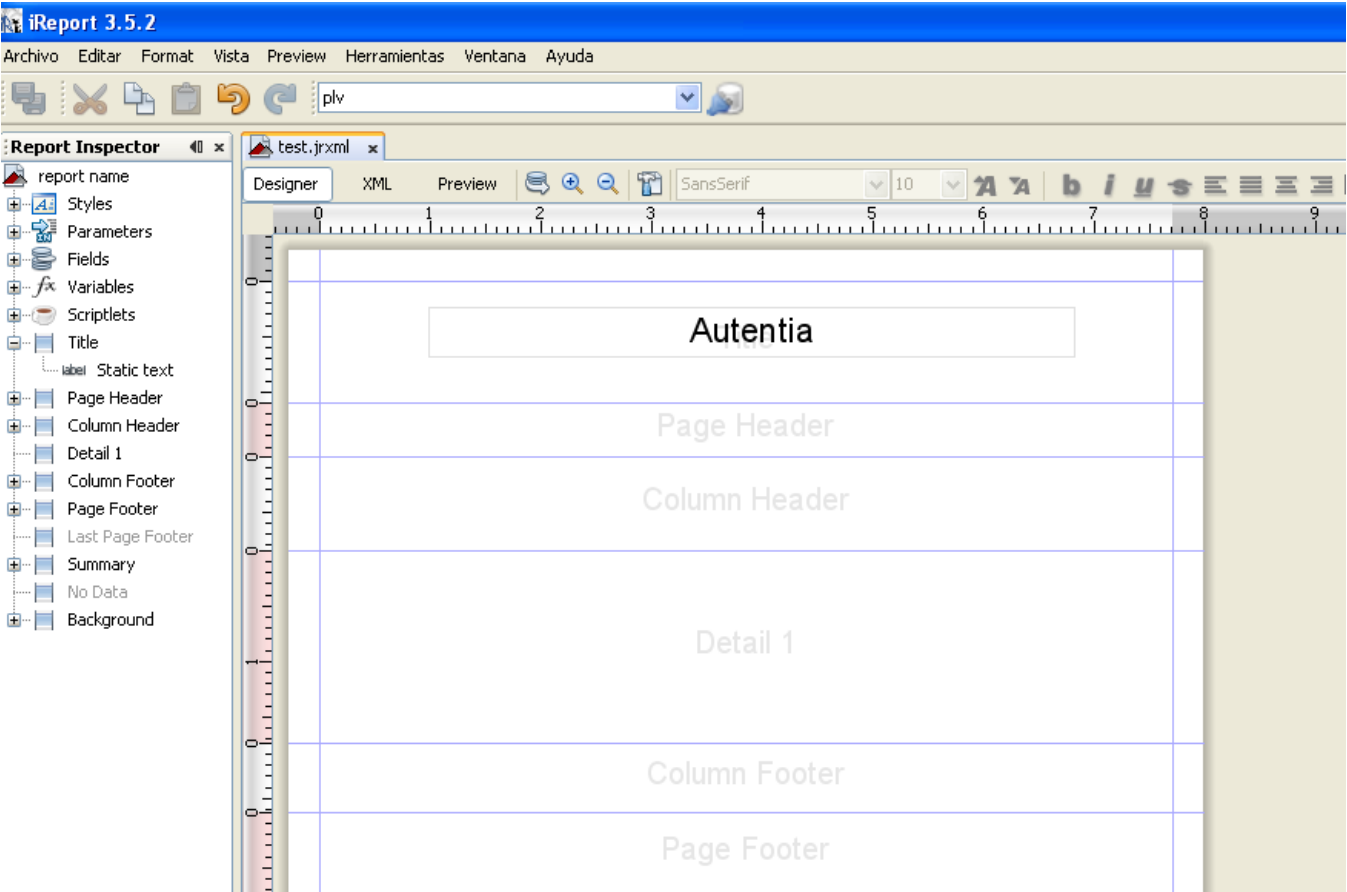
2009-06-03
Comercial - Ventas - VIZCAYA.

Ads by Google

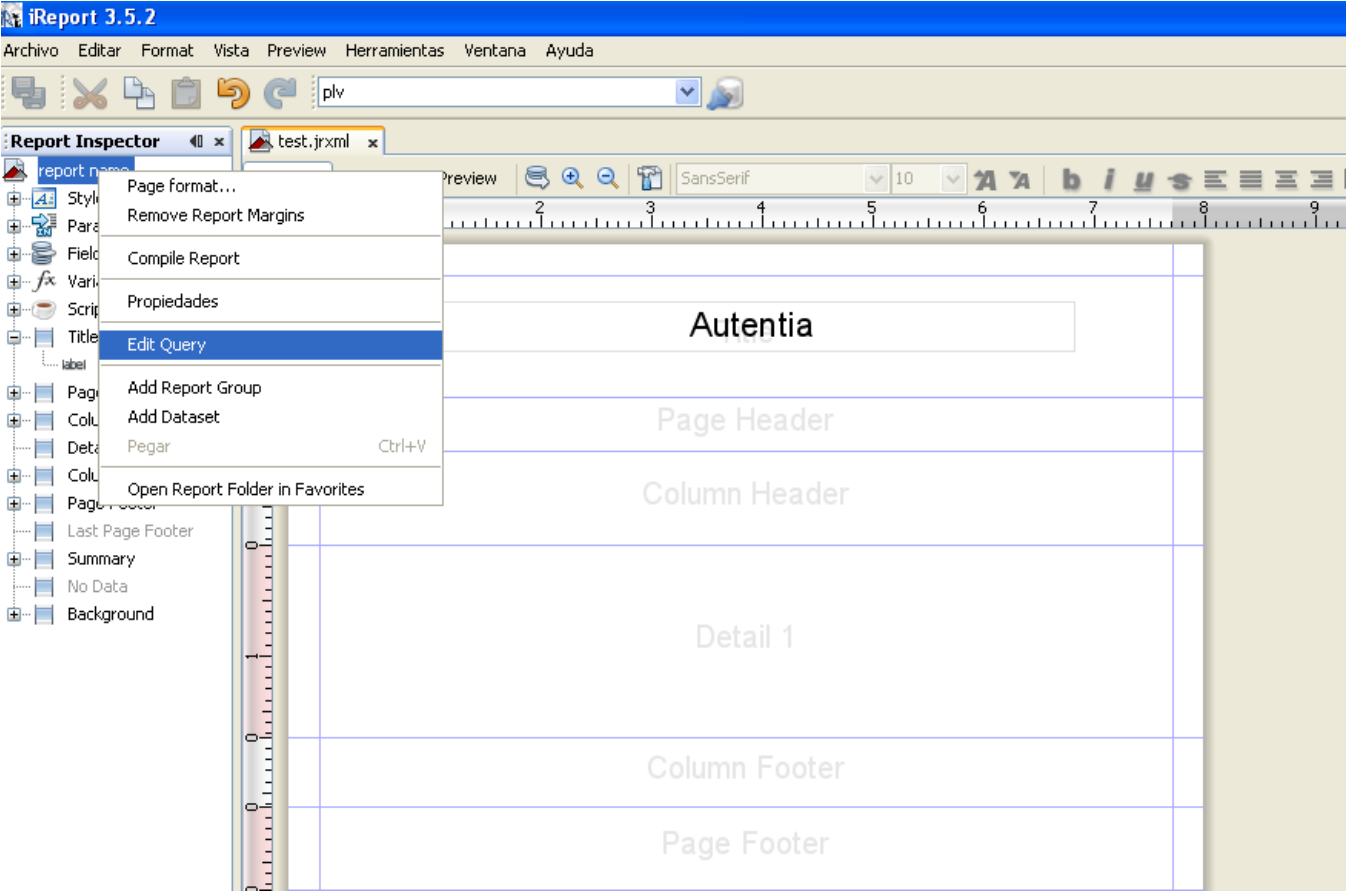
[JSP Examples](#)
[JSP JDBC](#)
[Java JDBC Tutorial](#)
[Java Servlet](#)

datos. De esta manera IReports es capaz de leer los fileds del objeto y podremos utilizar cada uno de estos field en el diseño de nuestro informe.

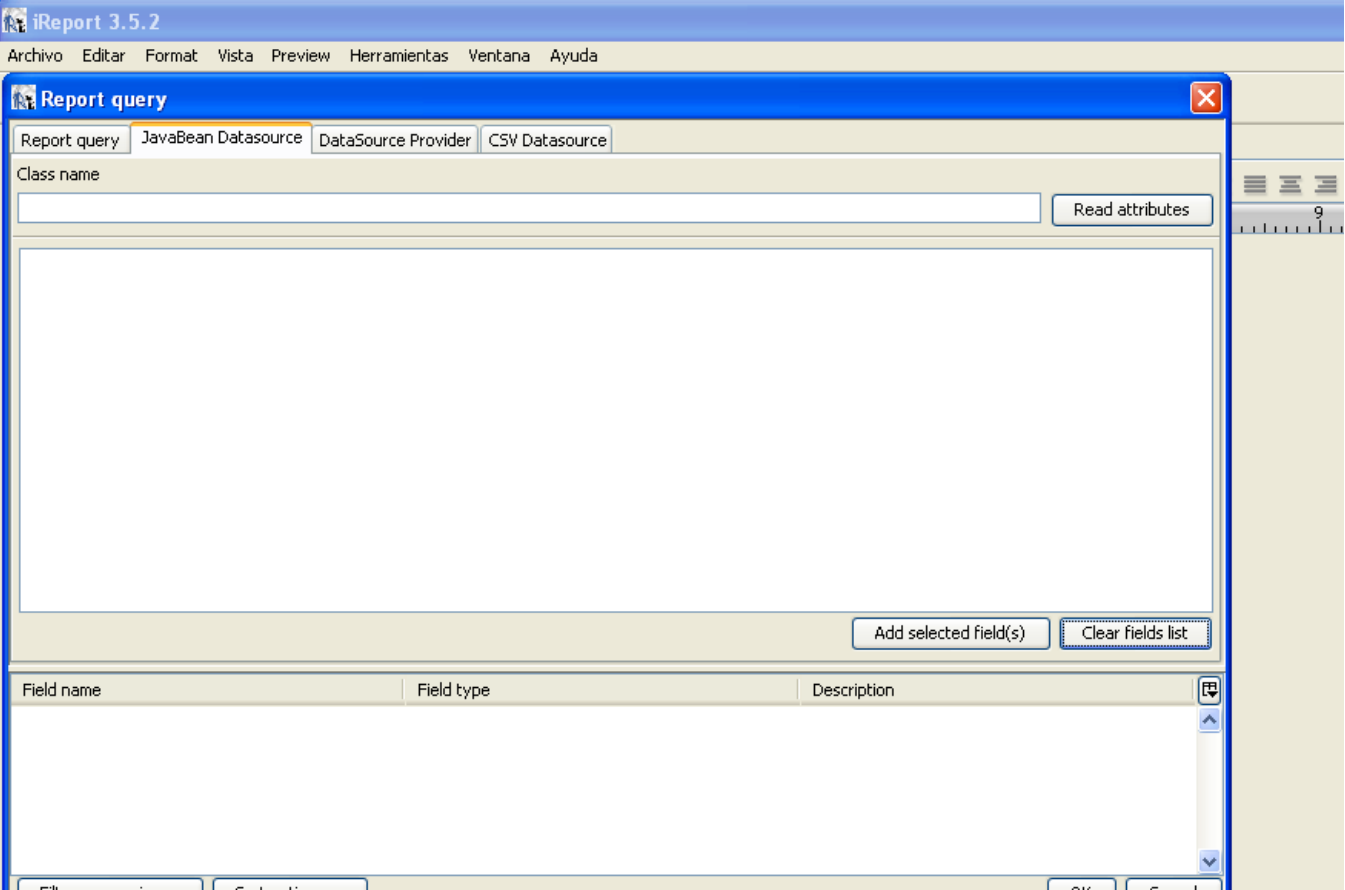
- Creamos un informe en blanco



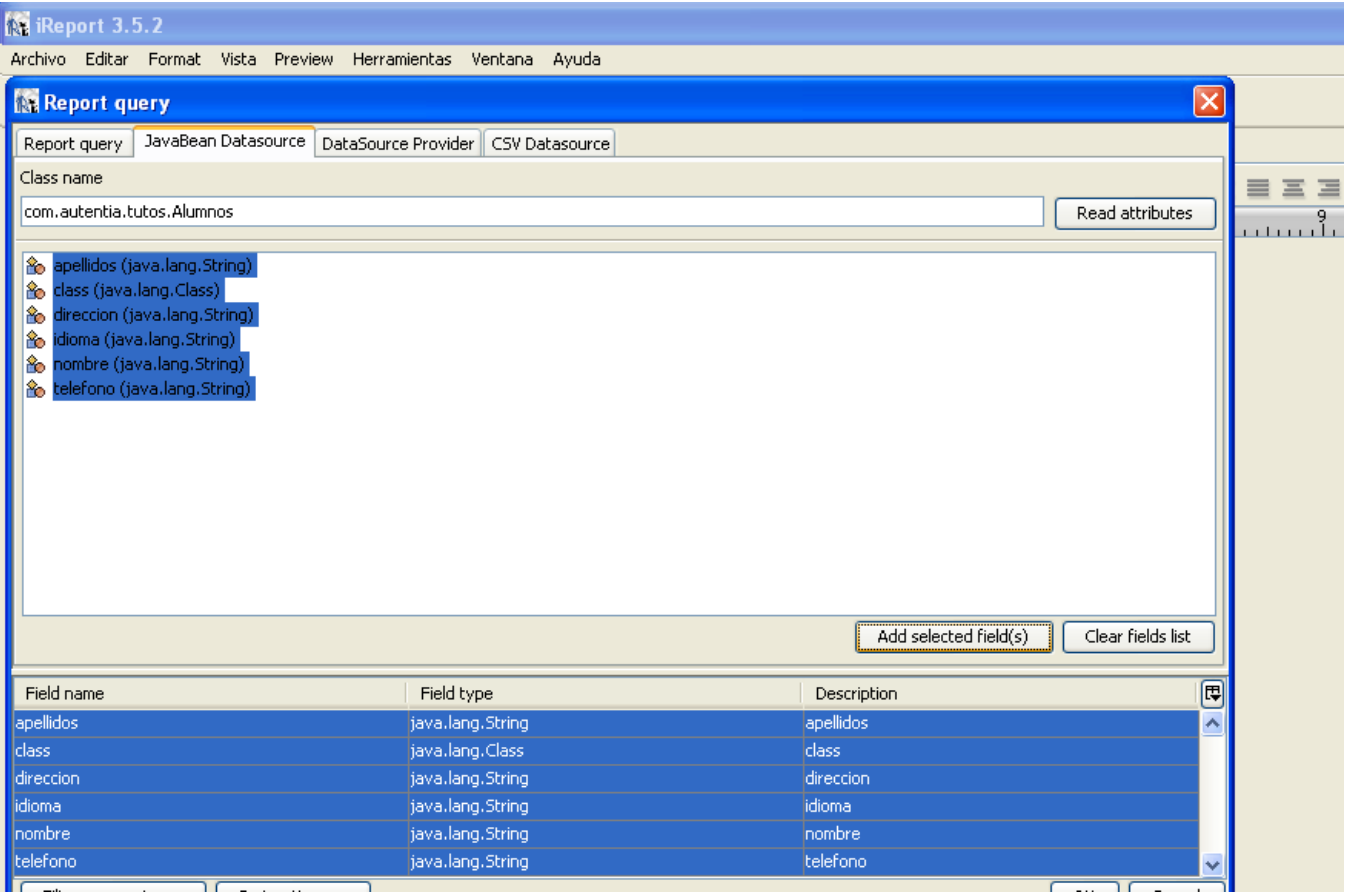
- Ahora, en el menú lateral izquierdo pulsamo el botón derecho sobre el nombre del informe y seleccioname Edit Query



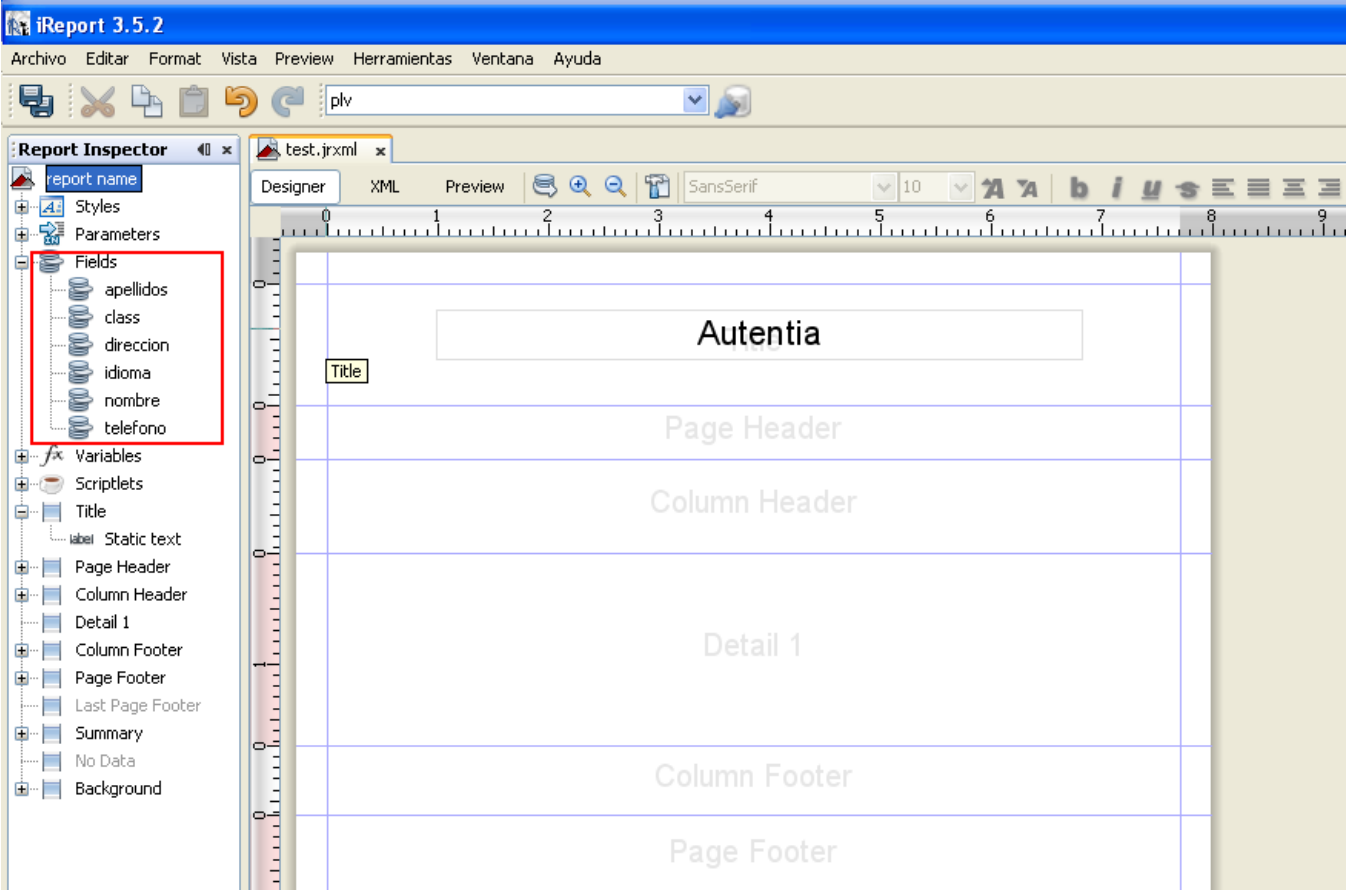
- A continuación aparece una ventana emergente con cuatro pestañas. Nos situamos sobre la segunda pestaña, JavaBean DataSource.



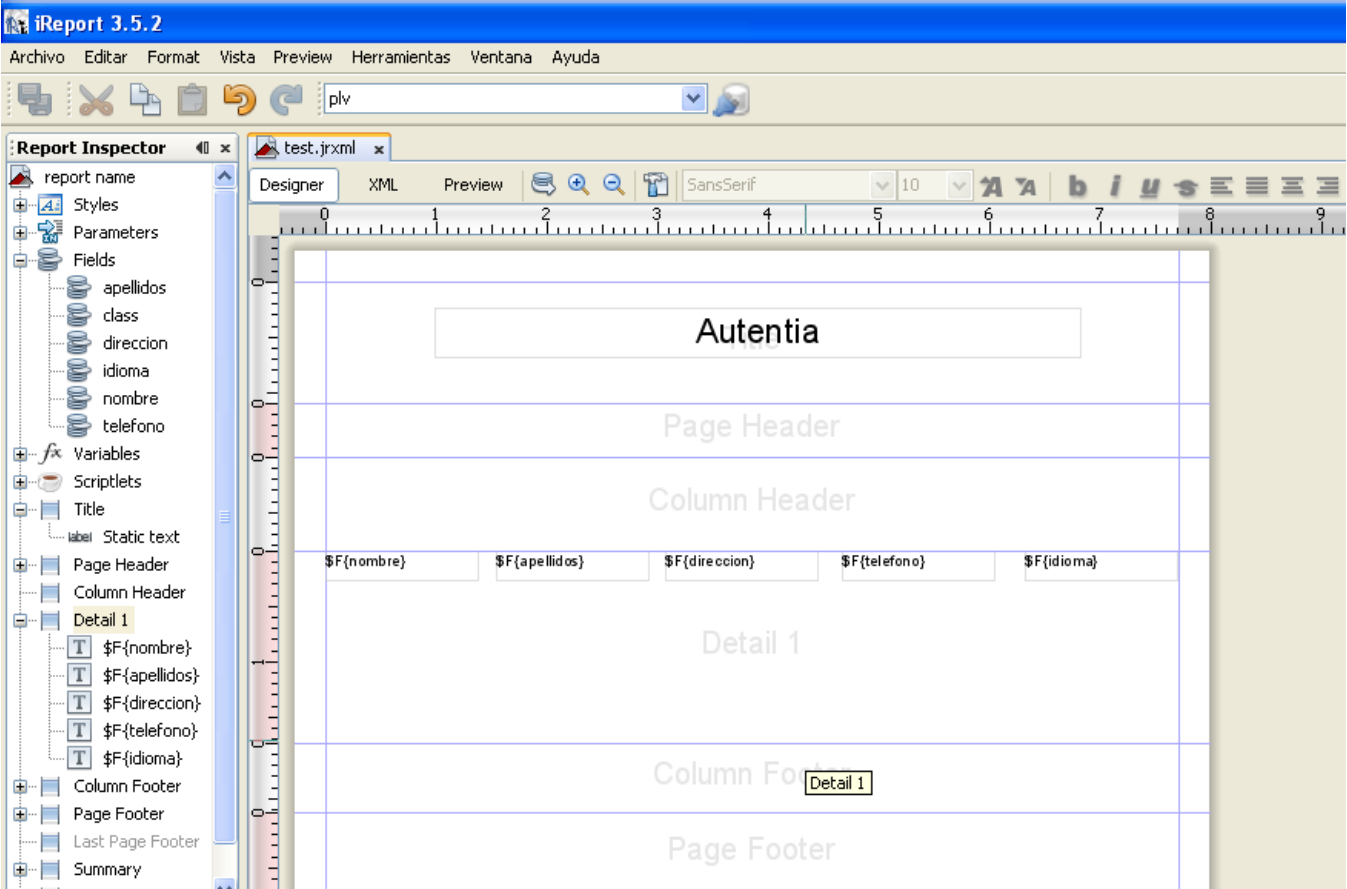
- Indicamos en nombre de la clase con el paquete completo y pulsamos en boton Read attributes
- Apareceran en la misma venta todos los campos de la clase que le indiquemos. De estos campos seleccionamos aquellos que queremos que se incluir en el informe y pulsamos el botón Add selected field
- En este punto es importante tener en cuenta que para que Ireport reconozca nuestra clase es necesario añadirle al classpath el jar donde se encuentra dicha clase



- Comprobamos que tenemos todos los fields disponible para incluir en el informe

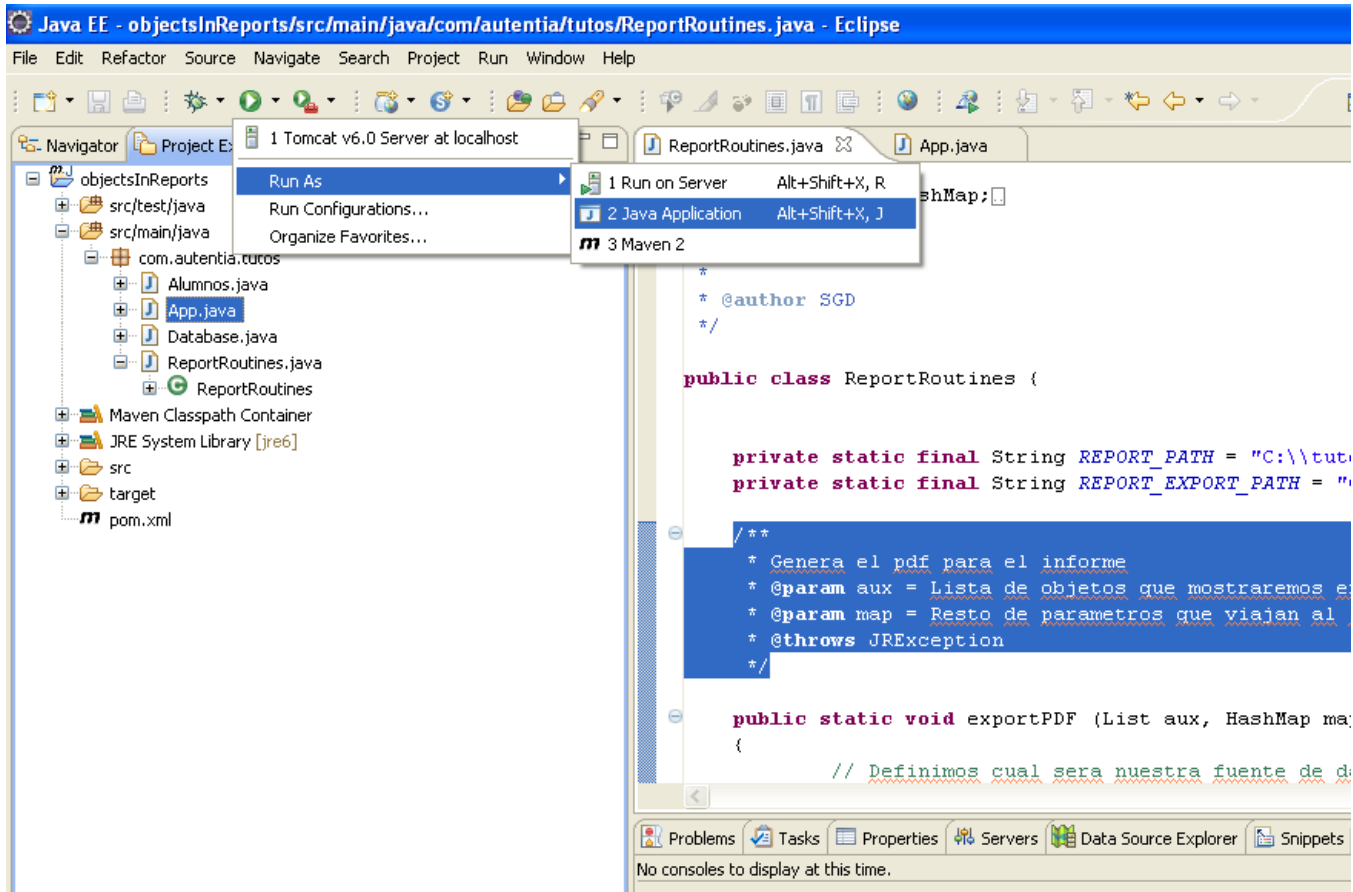


• Por último añadimos los campos recuperados en el informe.



6. Comprobando el resultado.

Para poder ver el resultado basta con situarnos encima de la clase App.java y :



Si nos vamos al directorio de salida definido en la clase ReportRoutines podemos ver que nos ha generado correctamente el informe en PDF.

7. Conclusiones.

Como vemos JasperReports-Ireport nos permite de una manera sencilla tener a mano distintas opciones de definir una fuente de datos y creedme, ésto es una ventaja ya que en ocasiones y debido a distintos motivos, pero sobre todo a la complejidad de los modelos de datos puede hacerse muy complicado sacar toda la información a mostrar en el informe en una sola consulta. Como siempre conocer la herramienta que con la que estamos trabajando supondrá una gran diferencia en nuestra productividad así que con este tutorial espero que hayáis aprendido un poquito mas acerca de JasperReports-Ireport.

Un saludo.

Saúl

Correo a: sgdiaz@autentia.com

¿Qué te ha parecido el tutorial? Déjanos saber tu opinión y ¡vota!

Muy malo Malo Regular Bueno Muy bueno



(Sólo para usuarios registrados)

» [Regístrate y accede a esta y otras ventajas](#) «

Anímate y coméntanos lo que pienses sobre este tutorial

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

(Sólo para usuarios registrados)

» [Regístrate y accede a esta y otras ventajas](#) «

Autor

Mensaje de usuario registrado

- Puedes inscribirte en nuestro servicio de notificaciones haciendo clic aquí.
- Puedes firmar en nuestro libro de visitas haciendo clic aquí.
- Puedes asociarte al grupo AdictosAlTrabajo en XING haciendo clic aquí.
- Añadir a favoritos Technorati.



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Recuerda

Autentia te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#)). Somos expertos en: J2EE, Struts, JSF, C+ +, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	Valoración	Votos	Pdf
Realización de in informe con JasperReports-Ireport	La particularidad del caso que nos ocupa, es conseguir que la fuente de datos del informe sea una lista de JavaBeans y no una consulta definida previamente en el informe.	2009-12-14	5	-	-	
Instalar OpenESB 2.1 e Introducción	En este tutorial veremos como descargar e instalar OpenESB y explicaremos sus funcionalidades	2009-12-03	462	-	-	
Tutorial básico sobre Google Application Engine	En este tutorial veremos cómo ejecutar una simple aplicación web en la infraestructura de Google	2009-11-12	906	Bueno	1	
Maven Assembly Plugin: empaquetando aplicaciones con Maven para la ejecución de procesos batch.	Examinamos cómo llevar a cabo el empaquetado de nuestras utilidades batch, basadas en una capa de servicios y cómo hacer la distribución independiente del entorno.	2009-11-12	619	-	-	
Instalación de Glassfish 2.1	En este tutorial nos veremos cómo instalar el servidor de aplicaciones GlassFish. Además veremos los primeros pasos, como entrar en la consola de administración del servidor, y desplegar una aplicación EAR (Enterprise Application)	2009-11-11	1136	Bueno	2	
Cómo conseguir que Subversion avise a Hudson para lanzar una build	En este tutorial vamos a ver como configurar Subversion para que sea este el que avise a Hudson cada vez que hay un commit, y así se lance la build.	2009-10-27	1299	-	-	
Cómo instalar Hudson en Apache Tomcat	Instalar Hudson en Apache Tomcat	2009-10-26	1602	-	-	
EJB 3.0 y pruebas unitarias con Maven, JUnit 4 y Apache Open EJB.	Continuamos buscando una buena solución para llevar a cabo test de EJBs, ahora con OpenEJB.	2009-09-23	1673	-	-	
EJB 3.0 y pruebas de persistencia con Maven, JUnit 4 y Embedded JBoss sobre Java 6.	Damos continuidad al tutorial EJB 3.0 y pruebas unitarias con Maven, JUnit 4 y Embedded JBoss Java 6, probando el llevar a cabo un test de persistencia de un EJB de entidad, bajo soporte de EJB de servicio que implementa el patrón dao, en el mismo entorno	2009-09-14	1950	Muy bueno	2	
Enlazar Bugzilla con MavenChangesPlugin	En este tutorial veremos como enlazar Bugzilla con MavenChangesPlugin	2009-09-11	867	-	-	

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.

[Ads by Google](#) [Certificaciones Java](#) [JDBC Connection](#) [DB Java](#) [JDBC SQL](#) [JDBC Example](#)

Copyright 2003-2009 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacte](#)

[W3C XHTML 1.0](#) [W3C CSS](#) [XML RSS](#) [XML RDFM](#)