

# ¿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

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

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

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

**CoNcept** Lanzado TNTConcept versión 0.8 ( 10/12/2007)

¿Gestionas tu empresa con hojas de cálculo? ¿No crees que puede haber un modo mejor?

Desde [Autentia](#) ponemos a vuestra disposición el software que hemos construido (100% gratuito, con código fuente disponible y sin restricciones funcionales) para nuestra gestión interna, llamado TNTConcept (auTeNTia). Construida con las últimas tecnologías de desarrollo Java/J2EE (Spring, JSF, Acegi, Hibernate, Maven, Subversion, etc.) y disponible en licencia GPL, seguro que a muchos profesionales independientes y PYMES os ayudará a organizar mejor vuestra operativa.

Las cosas grandes empiezan siendo algo pequeño ..... Saber más en:

<http://tntconcept.sourceforge.net/>

<p><b>Tutorial desarrollado por:</b> <a href="#">Jose Manuel Sánchez Suárez</a></p> <p>Puedes encontrarme en <a href="#">Autentia</a> Somos expertos en Java/J2EE Contacta en: <a href="mailto:jmsanchez@autentia.com">jmsanchez@autentia.com</a></p>	<p><b>NUEVO CATÁLOGO DE SERVICIOS DE AUTENTIA (PDF 6,2MB)</b></p> <p><a href="http://www.adictosaltrabajo.com">www.adictosaltrabajo.com</a> es el Web de difusión de conocimiento de <a href="http://www.autentia.com">www.autentia.com</a></p> <p> <b>autentia</b> real business solutions</p> <p><a href="#">Catálogo de cursos</a></p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

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

[Firma en nuestro libro de Visitas](#) <-----> [Asociarme al grupo AdictosAlTrabajo en eConozco](#)

**KOTASOFT Software Factory**

Desarrollo software CMMI nivel 3. Soluciones J2EE - Struts, Spring, ...  
[www.kotasoft.com](http://www.kotasoft.com)

**Master Experto Java**

100% alumnos se colocan. Incluye Struts, Hibernate, Ajax  
[www.grupoatrium.com](http://www.grupoatrium.com)

**Fecha de creación del tutorial: 2007-12-12**

**Dali: JPA tools.****0. Índice de contenidos.**

- [1. Introducción](#)
- [2. Entorno.](#)
- [3. Creación de un proyecto con soporte para JPA.](#)
- [4. Generación de entidades en función del modelo de datos.](#)
- [5. Perspectiva JPA y ventanas de propiedades.](#)
- [6. Conclusiones.](#)

**1. Introducción**

[Dali JPA tools](#) es un subproyecto de Eclipse Web Tools Platform 2.0.

Ya hemos visto en adictos [Eclipse Web Tools Platform 2.0](#) de la mano de [Alejandro Perez](#) y éste tutorial se basa en el mismo.

El objetivo del proyecto Dali JPA tools es facilitar el mapeo de objetos con la fuente de datos (O/R mappings) para EJB 3.0. Para ello, nos proporciona dos utilidades:

- la creación de las clases (los fuentes), que contienen las entidades de ejb3, en función de las tablas en base de datos,

- el acceso a la definición de las propiedades de mapeo desde una ventana de detalle que nos ayuda bastante en la definición.

Quizás lo más curioso del proyecto sea su nombre [Why Dali?](#).

## 2. Entorno.

El tutorial está escrito usando el siguiente entorno:

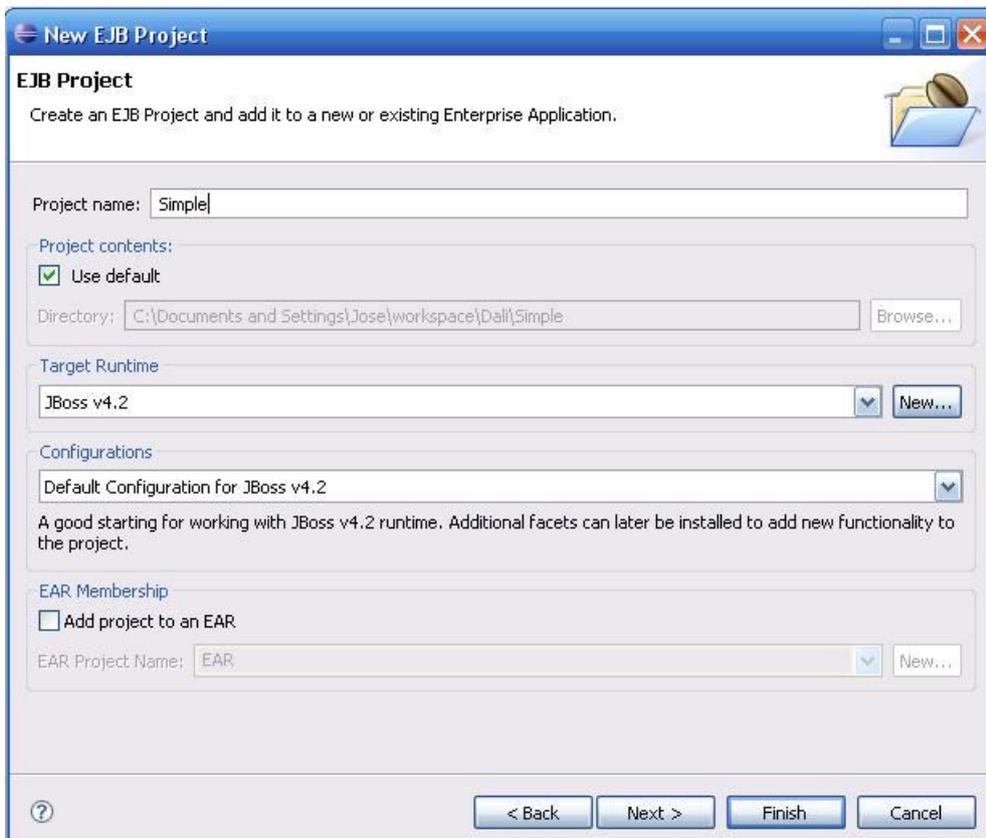
- Hardware: Sobremesa Dell Dimension 6400, 2.13 Ghz, 2 Gb RAM
- Sistema operativo: Windows XP Media center Edition
- JDK 1.6.0\_2
- Eclipse 3.3.
- MySQL 5.0

## 3. Creación de un proyecto con soporte para JPA.

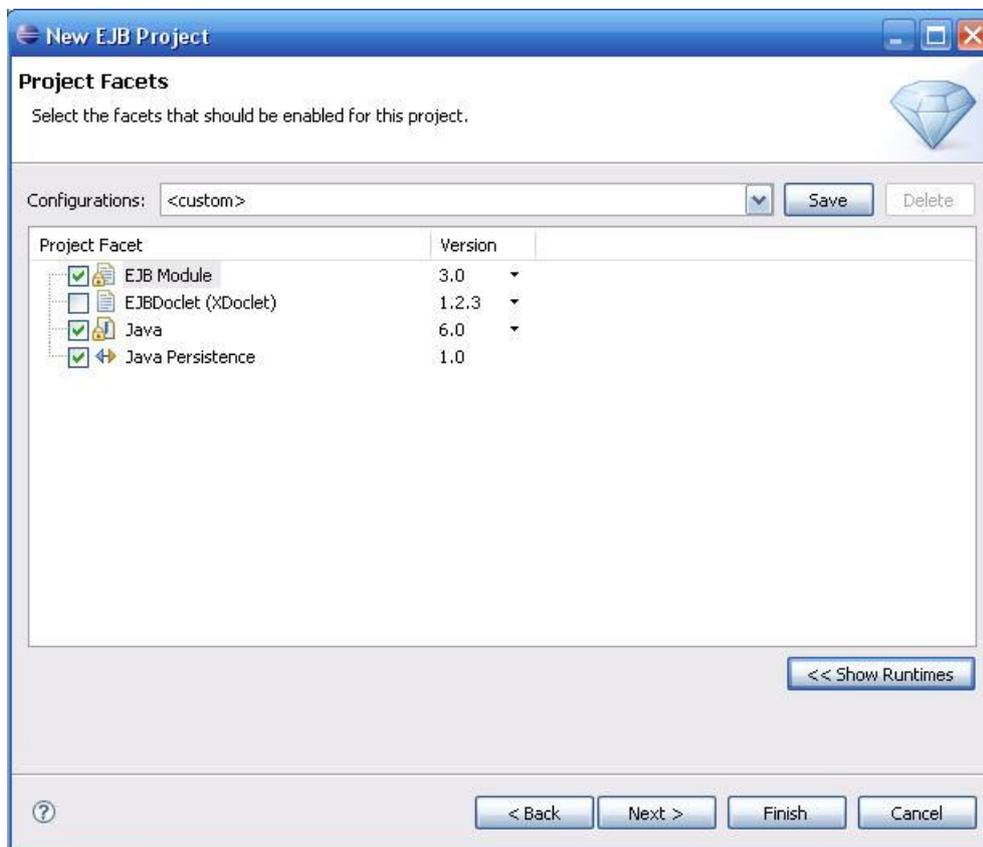
Lo primero que haremos será crear un proyecto de tipo ejb, seleccionar el tipo y Next >



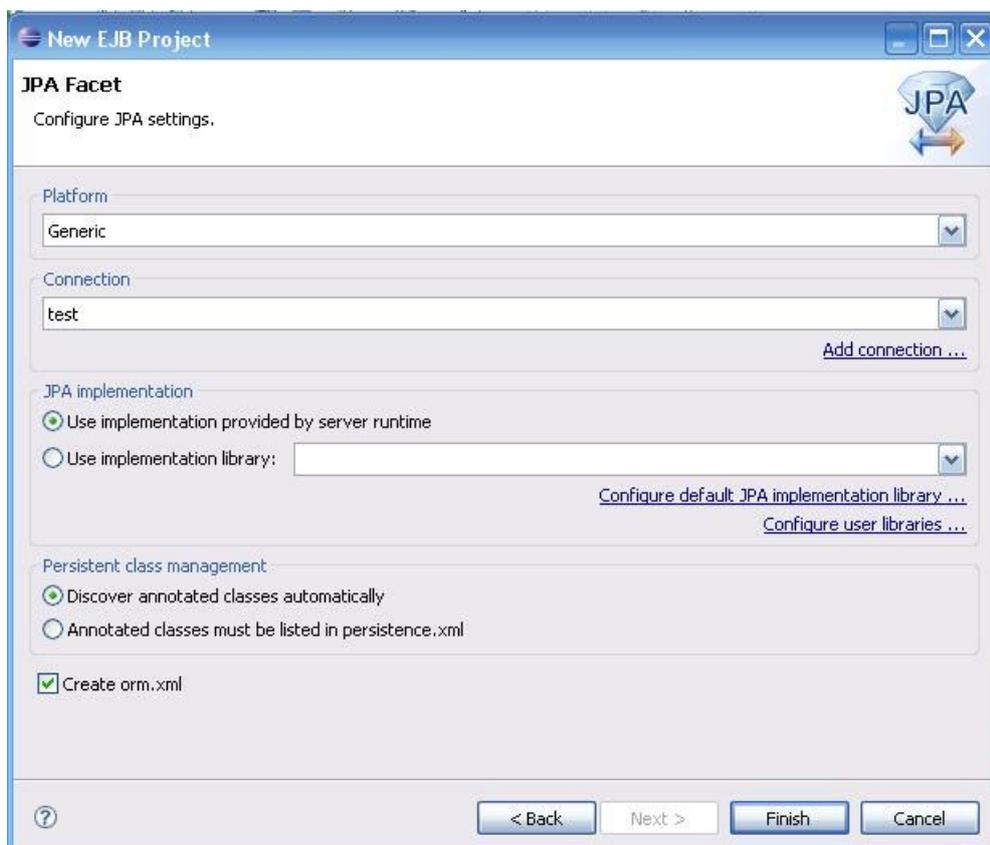
Asignamos un nombre para el proyecto (en nuestro caso Simple) y un "target runtime", un servidor de aplicaciones, en el que correr el proyecto (en nuestro caso Jboss v 4.2), Next >



Seleccionamos la versión de Java (6.0), de EJB (3.0) y la de JPA persistence (1.0), Next >



Pasamos por una ventana intermedia en la que nos pide información sobre el directorio en el que se alojarán los fuentes del proyecto, y llegamos a la siguiente, en la que nos pide información sobre la fuente de datos y la implementación de JPA:



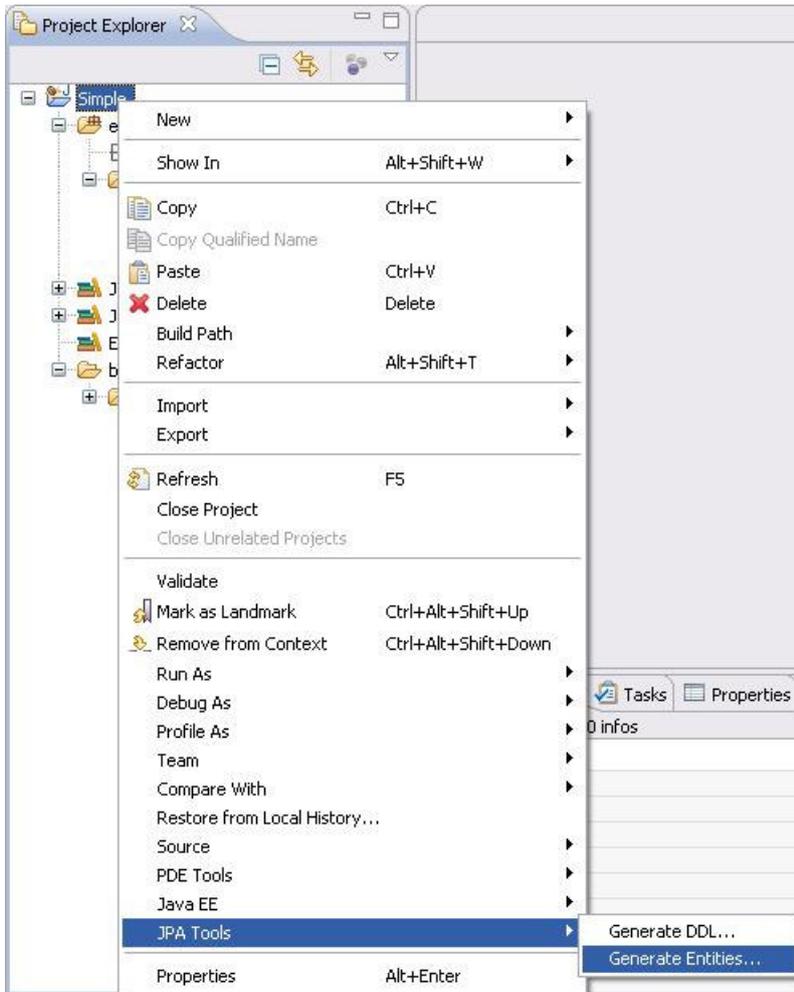
Necesitamos una conexión con base de datos configurada, como se explica en el tutorial [Eclipse Web Tools Platform 2.0](#), sino la tenemos podemos crearla pulsando sobre "Add connection".

La implementación de JPA será la que proporcione el servidor de aplicaciones "server runtime".

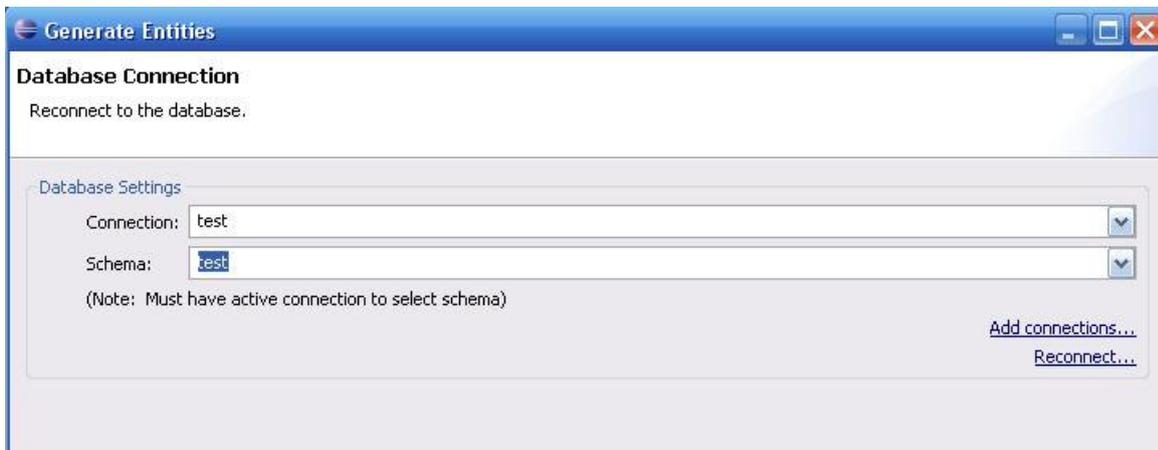
Pulsamos sobre Finish > y se creará la estructura del proyecto.

#### 4. Generación de entidades en función del modelo de datos.

Sobre el proyecto, botón derecho > JPA Tools > Generate Entities...



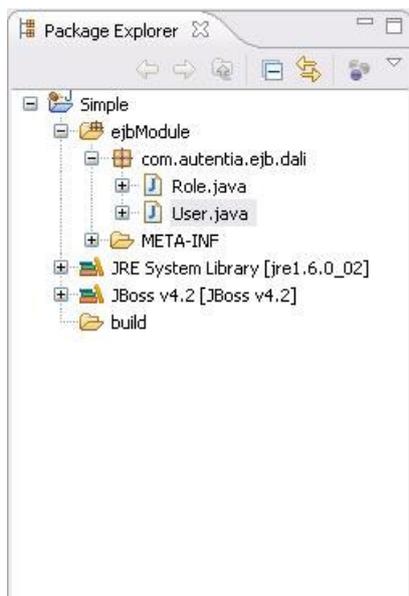
Seleccionamos la conexión, definida cuando creamos el proyecto, y el esquema sobre el que vamos a generar las entidades, Next >



Asignamos un paquete para las clases que contendrán las entidades y seleccionamos qué tablas queremos importar, Finish >



En este punto del tutorial, nuestro proyecto debería tener un aspecto como el que sigue:



Bajo el paquete seleccionado se han creado tantas clases como tablas hemos seleccionado, con todo lo necesario para utilizarse como ejb3 the entidad: sus anotaciones correspondientes, la definición de atributos y sus setters y sus getters.

```

view plain print ?
01. package com.autentia.ejb.dali;
02.
03. import java.io.Serializable;
04. import javax.persistence.Entity;
05. import javax.persistence.Id;
06. import javax.persistence.JoinColumn;
07. import javax.persistence.ManyToOne;
08. import javax.persistence.Table;
09.
10. @Entity
11. @Table(name="User", schema="test") // no es necesario si está bien definida la conexión
12. public class User implements Serializable
13. {
14.
15.     @Id
16.     private Object id;
17.
18.     private String login;
19.
20.     private String password;
21.
22.     @ManyToOne
23.     @JoinColumn(name="roleId")
24.     private Role roleid;
25.
26.     private static final long serialVersionUID = 1L;
27.
28.     public User() {
29.         super();
30.     }
31.
32.     public Object getId() {
33.         return this.id;
34.     }
35.
36.     public void setId(Object id) {
37.         this.id = id;
38.     }
39.
40.     public String getLogin() {
41.         return this.login;
42.     }
43.
44.     public void setLogin(String login) {
45.         this.login = login;
46.     }
47.
48.     public String getPassword() {
49.         return this.password;
50.     }
51.
52.     public void setPassword(String password) {
53.         this.password = password;
54.     }
55.
56.     public Role getRoleid() {
57.         return this.roleid;
58.     }
59.
60.     public void setRoleid(Role roleid) {
61.         this.roleid = roleid;
62.     }
63.
64. }

```

Pueden aparecer warnings o errores porque los nombres de las tablas en base de datos no sean nombres válidos para un atributo de una clase java, lo solucionamos a namo, o ayuándonos como se muestra en el siguiente punto.

Si la base de datos no está bien mapeada puede que aparezca el siguiente error: Schema "root" cannot be resolved for table... En tal caso, o mapeamos bien contra la base de datos o añadimos la siguiente anotación a nuestra Entity: @Table(name="NOMBRE\_DE\_LA\_TABLA", schema="NOMBRE\_DEL\_ESQUEMA").

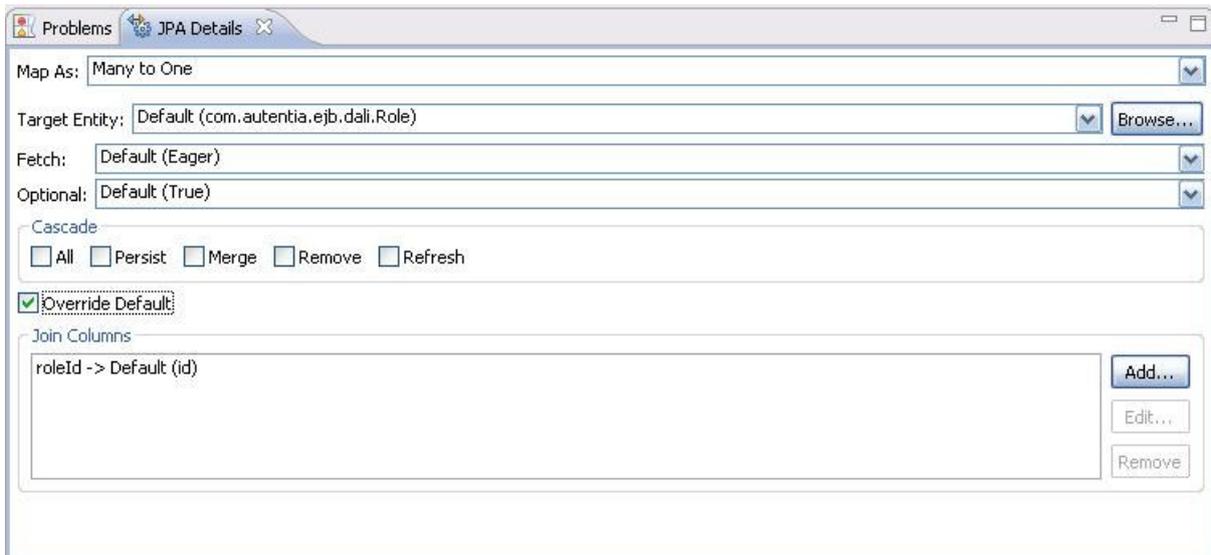
## 5. Perspectiva JPA y ventanas de propiedades.

Si seleccionamos la perspectiva JPA Development, veremos que al acceder a una clase marcada como @Entity se muestra una ventana como la que sigue, en la que se muestran los atributos de la clase marcados con un

icono en función del tipo



Si pulsamos sobre ellos, aparece una ventana de propiedades como la que sigue, en la que podremos modificar todo lo referente a sus anotaciones:



Todas las anotaciones que se refieren al mapeo de objetos y sus relaciones, que ya veíamos en el tutorial [Anotaciones en EJB 3.0](#) podemos modificarlas a través de ésta ventana de propiedades sin necesidad de realizarlo a mano.

## 6. Conclusiones.

La generación de las clases a partir del modelo de datos es un ahorro importante de carga de trabajo y la ventana asistida en la que modificar las propiedades de los campos ayudará a no cometer errores.

Aunque como con todo, siempre habrá quién prefiera hacerlo "a mano".

Un saludo.

[Jose Manuel.](#)

[Autentia.](#)



This work is licensed under a [Creative Commons Attribution-NonCommercial-No Derivative Works](#)



[2.5 License.](#)

[Puedes opinar sobre este tutorial aquí](#)

## Recuerda

que el personal de [Autentia](#) te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#))

¿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?

[info@autentia.com](mailto:info@autentia.com)

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos .....  
**Autentia = Soporte a Desarrollo & Formación**

[Autentia S.L.](#) Somos expertos en:  
**J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ..**  
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	<input type="text"/>
	<input type="button" value="Enviar"/>

---

## Otros Tutoriales Recomendados ([También ver todos](#))

### Nombre Corto

### Descripción

[Interceptando un EJB en JBoss](#)

En este tutorial os vamos a enseñar la arquitectura de EJBs en JBoss y a como modificarla, insertando un interceptor propio dentro de la cadena de interceptores del Proxy Cliente.

[Anotaciones en EJB 3.0](#)

Este tutorial nos va a enseñar algunas características del API de EJB 3.0 y las mejoras introducidas en la nueva versión 3.0

[Hibernate y las anotaciones de EJB 3.0](#)

En este tutorial Alejandro Pérez nos muestra las ventajas que nos aporta Hibernate y las anotaciones de EJB 3.0

[EJB 3.0: Resurrection](#)

Este tutorial nos va a presentar las nuevas funcionalidades que nos aportan los EJB 3.0.

[Comparativa entre EJB3 y Spring](#)

En este tutorial os mostramos una comparativa entre EJB3 y Spring esperando que os ayude a decidir qué tecnología utilizar.

[Comparativa entre Hibernate y EJB3 en la Capa de Persistencia](#)

El presente documento pretende dar algunas luces a la comparativa entre la opción de usar Hibernate y/o EJB3 para la capa de persistencia

[EJB 3.0 y pruebas unitarias con Maven, JUnit y Embedded JBoss](#)

En este tutorial Alejandro Pérez nos enseña como realizar test unitarios sobre EJB 3.0. Para ello se usará Maven, JUnit y Embedded JBoss

[EJB 3.0, un ejemplo práctico con Maven y JBoss](#)

Este tutorial presenta un ejemplo sencillo donde se verá como desarrollar EJBs de sesión y de entidad, inyección de dependencias, llamar a los EJBs desde una aplicación Web, definición de un DataSource, y como configurarlo y hacerlo funcionar en JBoss, y

[Despliegue gráfico de EJBs](#)

Os mostramos como crear y desplegar de un modo gráfico un EJB de sesión en el servidor de aplicaciones de referencia de Sun

[EJB's y Orion](#)

Recreación de la guía paso a paso de como crear una aplicación Web con EJB's y Servlets y su despliegue con ANT sobre Orion

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](mailto:rcanales@adictosaltrabajo.com) para su resolución.

[Patrocinados por enredados.com .... Hosting en Castellano con soporte Java/J2EE](#)

