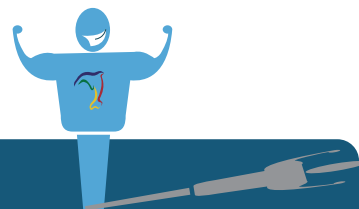


# ¿Qué ofrece Autentia Real Business Solutions S.L?

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

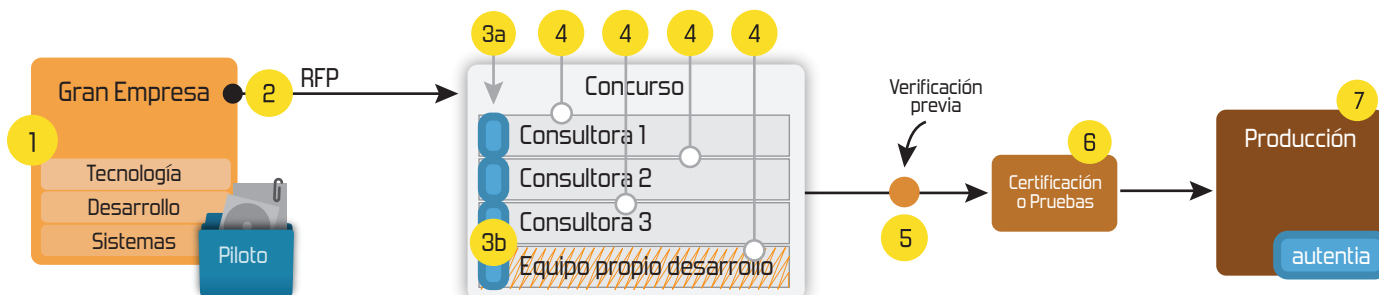
## 1. Desarrollo de componentes y proyectos a medida



## 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)



[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Tutoriales](#) | [Contacte](#)



CoNcept

Lanzado

## TNTConcept versión 0.4.1 ( 04/06/2007)

Desde [Autentia](#) ponemos a vuestra disposición el software que hemos construido (100% gratuito 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: <a href="#">Alejandro Perez García 2003-2007</a></b><br/> <b>Alejandro es Socio fundador de Autentia y nuestro experto en J2EE, Linux y optimización de aplicaciones empresariales.</b></p> <p>Si te gusta lo que ves, <b>puedes contratarle</b> para impartir <b>cursos presenciales</b> en tu empresa o para ayudarte en proyectos (Madrid).</p> <p>Contacta: <a href="mailto:alejandropg@autentia.com">alejandropg@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><br/>real business solutions</p> <p><a href="#">Catálogo de cursos</a></p> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

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

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

### AltioLive

Real-time data visualisation, interaction and integration.  
[www.altio.com](http://www.altio.com)

### Eclipse Hibernate Tools

Java Data-object Hibernate mapping Comprehensive J2EE IDE & Support  
[www.myeclipseide.com](http://www.myeclipseide.com)

### RUP implementation

Shorten time to value with agent-assisted tools for UML and RUP  
[www.jaczone.com](http://www.jaczone.com)

### Free UML 2.1 Design Tool

Visually develop applications with Roundtrip model to code, ERD & DB  
[www.visual-paradigm.com](http://www.visual-paradigm.com)

**Fecha de creación del tutorial: 2007-06-13**

# Hibernate Tools y la generación de código

Creación: 14-04-2007

## Índice de contenidos

- [1. Introducción](#)
- [2. Entorno](#)
- [3. Instalación](#)
- [4. Creación de la configuración de Hibernate](#)
- [5. Hibernate Console](#)
- [6. Generando código a partir de la base de datos](#)
- [7. revenge.xml el fichero de la "venganza" ;\)](#)
- [8. Conclusiones](#)
- [9. Sobre el autor](#)

## 1. Introducción

Hibernate (<http://www.hibernate.org/>) es una potente herramienta de persistencia que nos permite mapear clases en una base de datos relacional.

Hibernate Tools (<http://tools.hibernate.org/>) es un conjunto de utilidades para Ant (<http://ant.apache.org/>) y para Eclipse (<http://www.eclipse.org/>), que nos facilitan el uso de Hibernate.

En este tutorial vamos a ver como usar estas herramientas para hacer el esqueleto de una pequeña aplicación, de manera muy sencilla, generando código a partir de las tablas creadas en la base de datos. Gracias a esto podremos ahorrar bastante tiempo de desarrollo :D

## 2. Entorno

El tutorial está escrito usando el siguiente entorno:

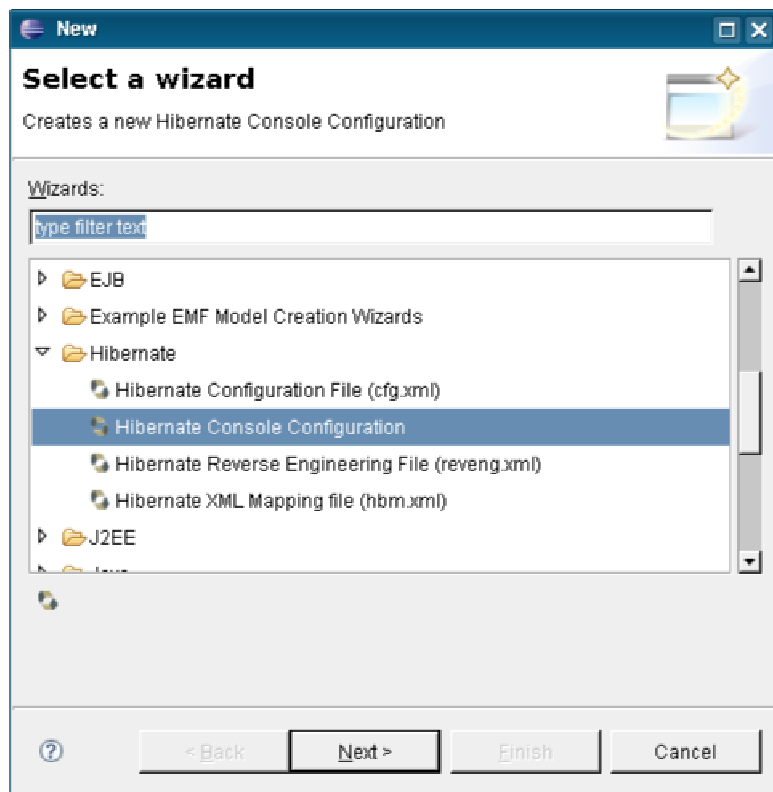
- Hardware: Portátil Asus G1 (Core 2 Duo a 2.1 GHz, 2048 MB RAM, 120 GB HD).
- Sistema Operativo: GNU / Linux, Debian (unstable), Kernel 2.6.21, KDE 3.5
- Máquina Virtual Java: JDK 1.6.0-b105 de Sun Microsystems
- Eclipse 3.2.2
- Hibernate 3.2.2.ga
- Hibernate Tools 3.2.0 Beta9
- MySql 5.0.41-2

## 3. Instalación

Para instalar las Hibernate Tools basta con ir a la página de descargas: <http://www.hibernate.org/6.html> y pinchar sobre el enlace "Download" de "Hibernate Tools"

Una vez descargado basta con descomprimirlo en el directorio del Eclipse (el mismo directorio donde se encuentra el ejecutable de Eclipse, por ejemplo, en Windows, eclipse.exe).

Si ahora arrancamos el Eclipse podemos comprobar, por ejemplo, que tenemos nuevas opciones en el asistente de creación (File -> New -> Other...)

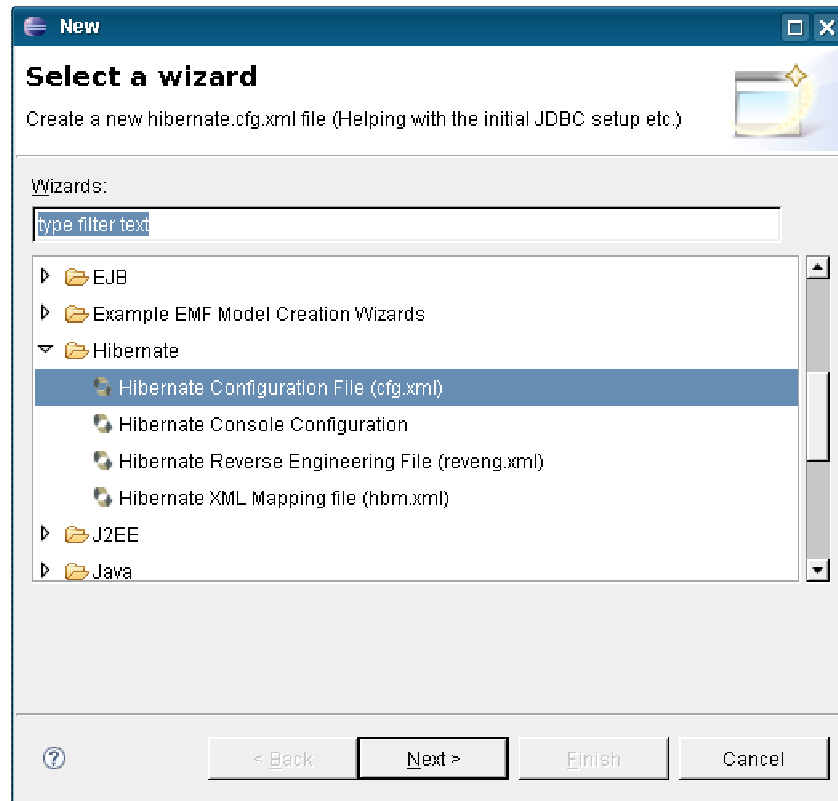


## 4. Creación de la configuración de Hibernate

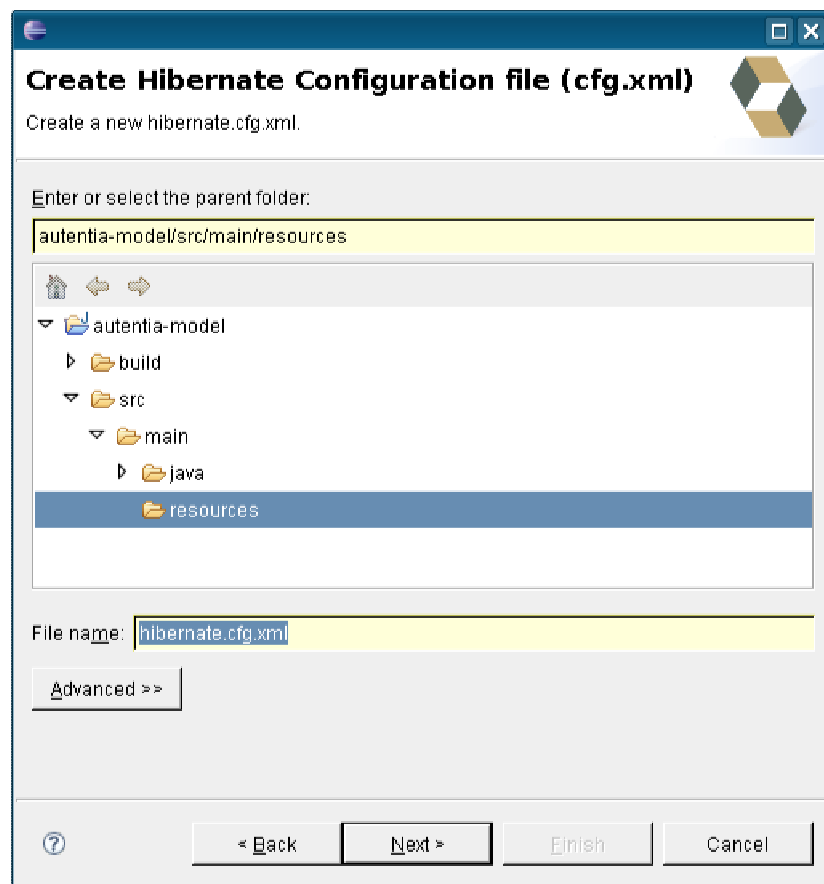
Las Hibernate Tools nos proporcionan un asistente para crear el fichero de configuración de Hibernate (normalmente `hibernate.cfg.xml`).

En este fichero es donde describiremos como se debe conectar Hibernate a la base de datos, cuales son los ficheros xml que describen los mapeos entre las clases y las tablas de la base de datos, ...

Para crearlo haremos: File -> New -> Other... -> Hibernate -> Hibernate Configuration File (cfg.xml)



Le damos el nombre al fichero de configuración (normalmente `hibernate.cfg.xml`), e indicamos donde debe guardarlo. Deberá ser un directorio que en ejecución forme parte del classpath, para que la aplicación lo pueda localizar (si usamos Maven será el directorio `src/main/resources`).



Ahora indicamos el dialecto que debe usar Hibernate. El dialecto, básicamente, es el idioma que ha de hablar Hibernate con nuestra base de datos. También indicamos la clase del driver de acceso a la base de datos, la URL de conexión, el usuario y la password, ... y en definitiva toda la información para que Hibernate se pueda concatenar correctamente a nuestra base de datos.

**Hibernate Configuration File (cfg.xml)**  
This wizard creates a new configuration file to use with Hibernate.

Container: /autentia-model/src/main/resources

File name: hibernate.cfg.xml

Session factory name:

Database dialect: MySQLInnoDB

Driver class: com.mysql.jdbc.Driver

Connection URL: jdbc:mysql://localhost:3306/AUTENTIA

Default Schema:

Default Catalog:

Username: autentia

Password: autentia

☐ Create a console configuration

< Back Next > Finish Cancel

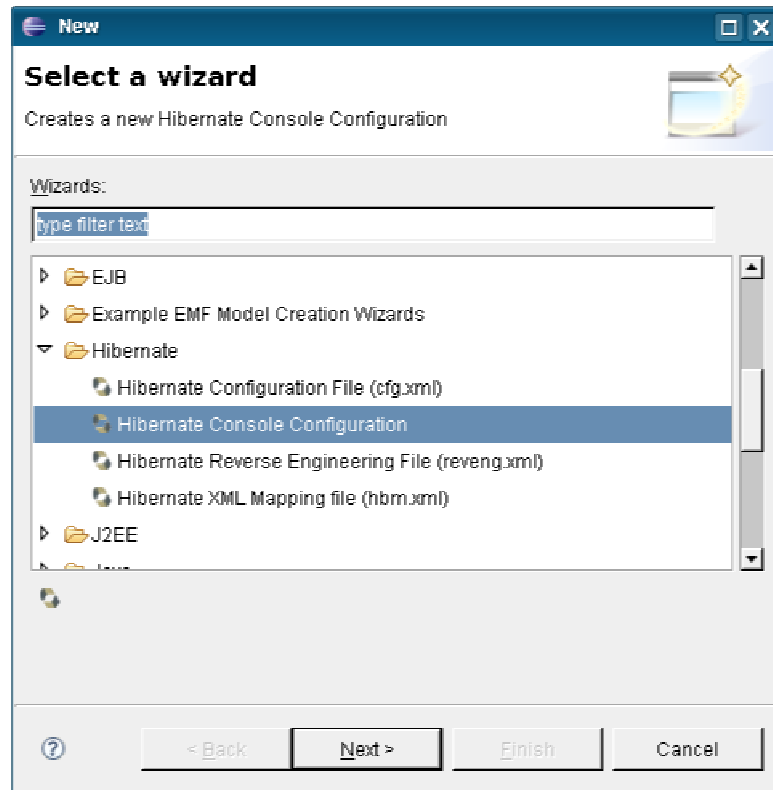
Cuando demos a "Finish" se creará el fichero, que tendrá un aspecto similar (según los datos introducidos en la pantalla anterior) a:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory>
<property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
<property name="hibernate.connection.url">jdbc:mysql://localhost:3306/AUTENTIA</property>
<property name="hibernate.connection.username">autentia</property>
<property name="hibernate.connection.password">autentia</property>
<property name="hibernate.dialect">org.hibernate.dialect.MySQLInnoDBDialect</property>
</session-factory>
</hibernate-configuration>
```

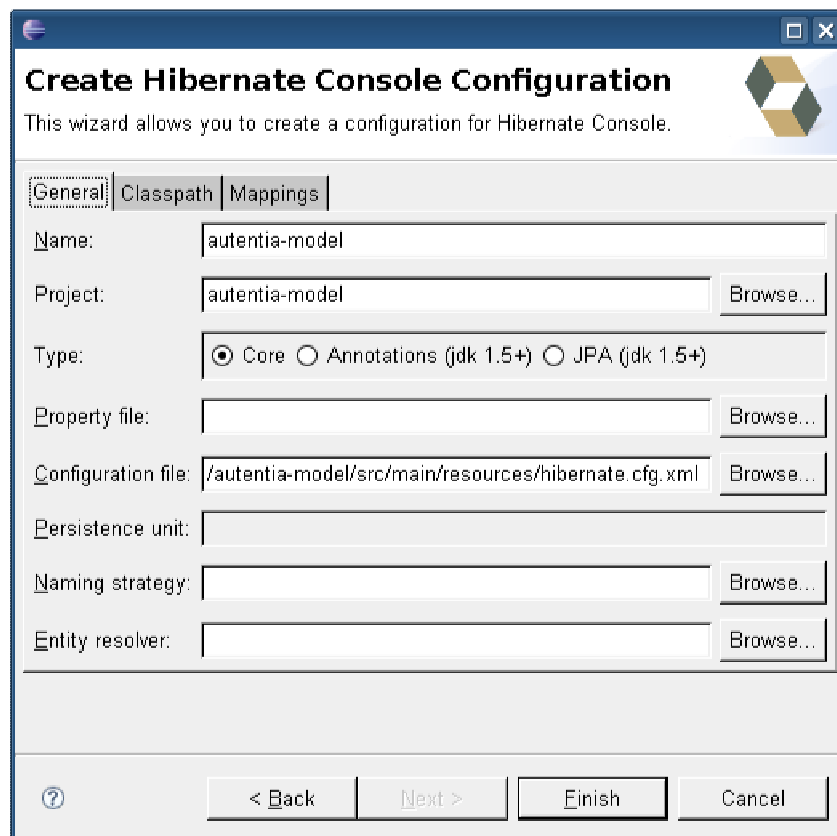
## 5. Hibernate Console

Ahora vamos a crear la consola de Hibernate. La consola es el eje central de las Hibernate Tools, ya que cualquier otra operación que queramos hacer (generar código, lanzar sentencias HQL, ...) dependerán de la configuración de la consola.

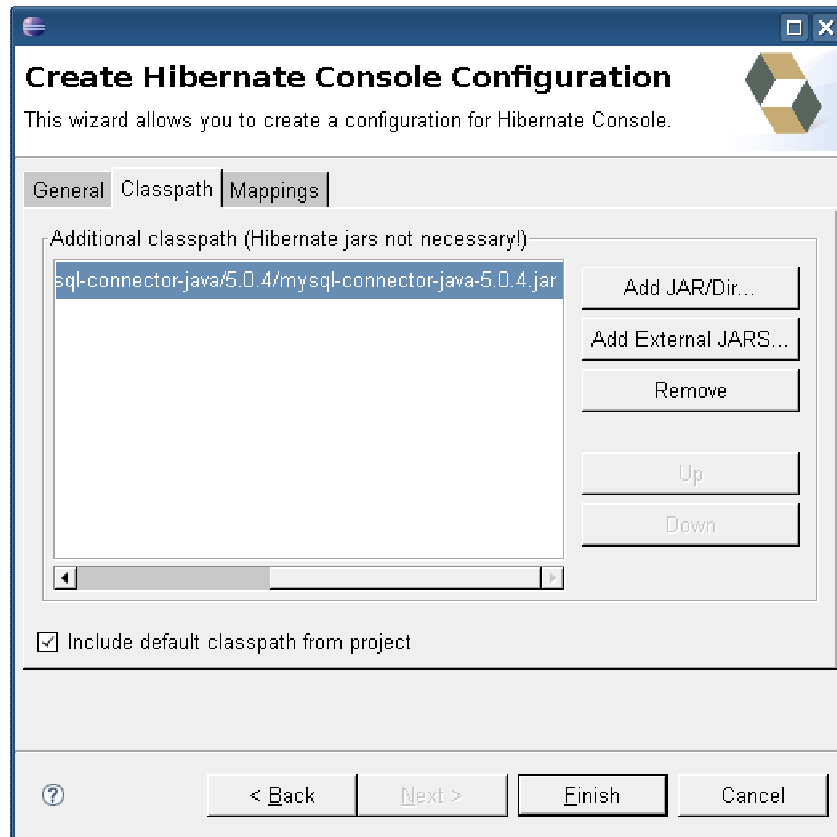
Para crear una nueva configuración de la consola de Hibernate hacemos: File -> New -> Other... -> Hibernate -> Hibernate Console



Indicamos el nombre que le damos a esta configuración, el proyecto asociado, y el fichero de configuración donde está configurada nuestra conexión. Este fichero es el típico fichero "hibernate.cfg.xml" de configuración de Hibernate. Indicaremos el fichero de configuración que hemos creado en el punto anterior:



Antes de pulsar el botón "Finish", pincharemos sobre la pestaña "Classpath". Aquí vamos a indicar donde se encuentra el driver de la base de datos. Esto es muy importante, ya que de lo contrario las Hibernate Tools serán incapaces de conectarse con la base de datos.

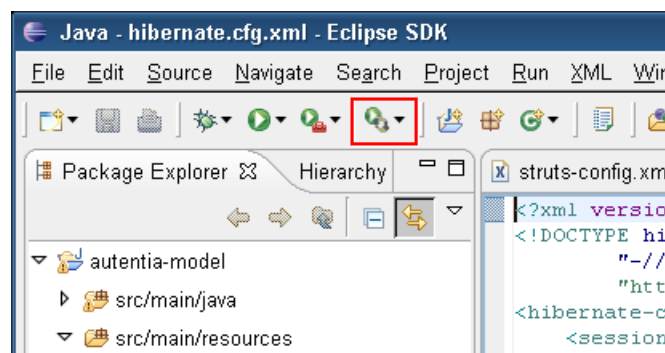


Nótese que también tenemos activado "Include default classpath project". Esto es necesario para que las Hibernate Tools sean capaces de encontrar nuestros .class, por ejemplo para poder lanzar sentencias HQL.

Ahora ya podemos pulsar el botón de "Finish".

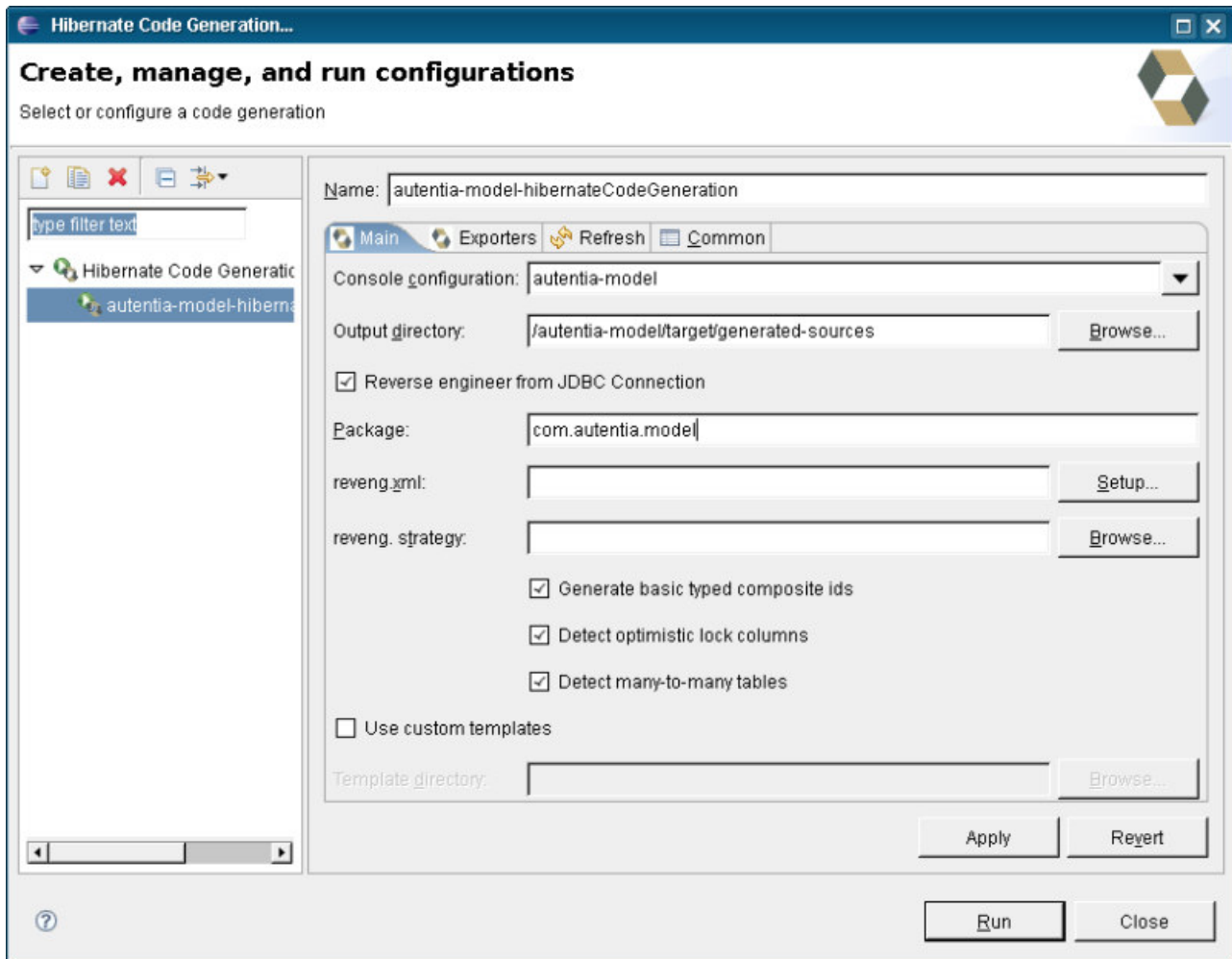
## 6. Generando código a partir de la base de datos

Ya estamos preparados para generar código a partir de las tablas creadas en nuestra base de datos. Para ello pulsamos sobre el nuevo icono (apareció al instalar las Hibernate Tools) que tenemos en la barra de herramientas (en la imagen aparece enmarcado en un rectángulo rojo):



Al pulsar sobre el icono, deberemos seleccionar la opción "Hibernate Code Generation....."

Nos aparece una ventana, donde en su lado izquierda aparece una zona en blanco. Sobre ella hay una barra de herramientas, pulsamos sobre el primer icono "New launch configuration". Y ahora rellenamos los datos de la zona de la derecha:

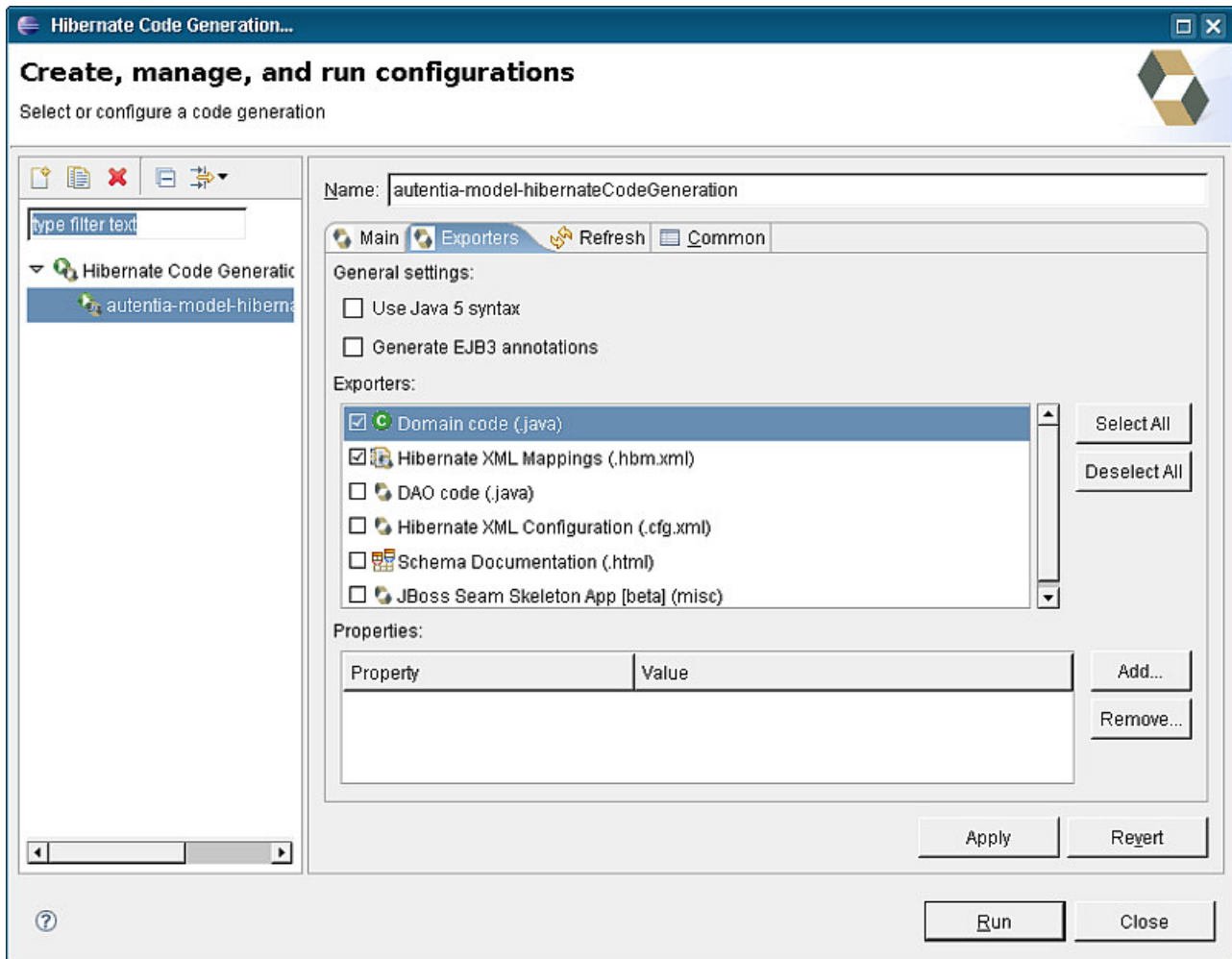


Lo que hemos hecho sobre la pantalla anterior:

- Le hemos puesto un nombre a esta nueva "launch configuration".
- Hemos indicado el nombre de la configuración de consola que se debe utilizar (la que hemos creado en el punto anterior).
- Hemos indicado el directorio de salida para el código generado. Os recomiendo usar un directorio fuera de vuestro directorio de fuentes, para que no perdamos código por error (sobreescribamos algún fichero).
- Hemos marcado la casilla "Reverse engineer from JDBC Connection". Esto es para generar las clases a partir de la información que tenemos en la base de datos. Ojo, si queremos que genera las relaciones entre las clases es imprescindible que la base de datos esté configurada con integridad referencial. Otro ojo, porque si usamos MySQL las tablas deben estar creadas con InnoDB, si están creadas como MyISAM la información de integridad referencial entre las tablas no se tiene en cuenta.
- Y en general marcamos los check que nos interesan.

Ahora pasamos a la pestaña de "Exporters". Cada "exporter" es capaz de generar algo diferente (clases java, ficheros de configuración, documentación, ...). Marcaremos los que nos interesen. En la imagen de ejemplo se puede ver que hemos marcado "Domain code (java)" para generar los POJOs e "Hibernate XML Mappings (.hbm.xml)" para generar los ficheros de configuración xml donde se describe el mapeo entre las clases y las tablas de la base de datos.





En esta misma pantalla de los "Exportes" tenemos otra dos opciones:

- Use Java 5 syntax: el código java generado usará la sintaxis List<Clase> para indicar el tipo de las colecciones.
- Generate EJB3 annotations: Genera POJOs anotados según el estándar de EJB3. Esto es una alternativa a los ficheros xml de mapeo, de forma que, mediante estas anotaciones, en el mismo POJO es donde se indica como se debe mapear con la base de datos. Estas anotaciones además de evitarnos mantener esos xml, tienen la ventaja de que son compatibles con las anotaciones de la nueva especificación 3 de EJBs (podríamos convertir nuestros POJOs en EJBs de forma casi directa, o usar nuestros POJOs con la capa de persistencia de EJB3 en vez de con Hibernate).

Sólo podemos usar estas opciones si tenemos una máquina virtual 5 o superior. Si es el caso, os lo recomiendo, la primera para detectar en compilación posibles problemas de tipos, y el segundo sobre todo por escribir y mantener menos ficheros.

Ahora ya podemos al botón "Run" para generar el código.

## 7. revenge.xml el fichero de la "venganza" ;)

Si observamos el código generado en el punto anterior podemos ver dos cosas:

- Si nuestros identificadores en la base de datos son numéricos, los atributos correspondientes de los POJOs son tipos básicos (short, int, long).
- Las relaciones de integridad referencial en la base de datos se han convertido en asociaciones bidireccionales en las clases (es decir, si tengo una relación 1:n tendré una clase "foo" con un atributo que hace referencia a la clase "bar", y en la clase "bar" tendré una lista de objetos de "foo").

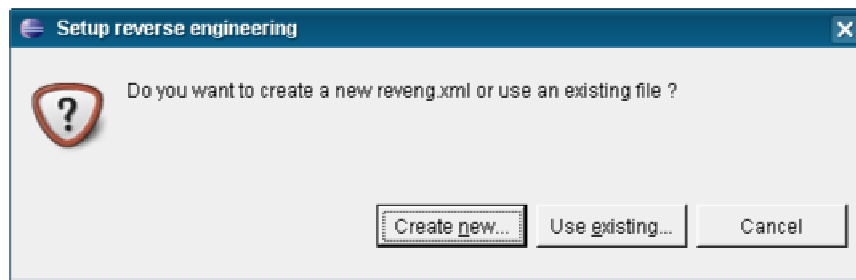
Estas dos situaciones no son siempre recomendables:

- En el caso de los identificadores es recomendable que siempre sean atributos nulos, de esta forma Hibernate es capaz de distinguir si la entidad ya existe en la base de datos o si se trata de una nueva entidad que habrá que añadir.
- En el caso de las relaciones, no siempre es necesaria esa bidireccionalidad, de hecho, estas asociaciones bidireccionales son el caso menos frecuente, ya que solemos hacer la navegación siempre en un sentido (por ejemplo de un pedido saco la lista de productos, pero de un producto no saco la lista de todos los pedidos donde aparece).

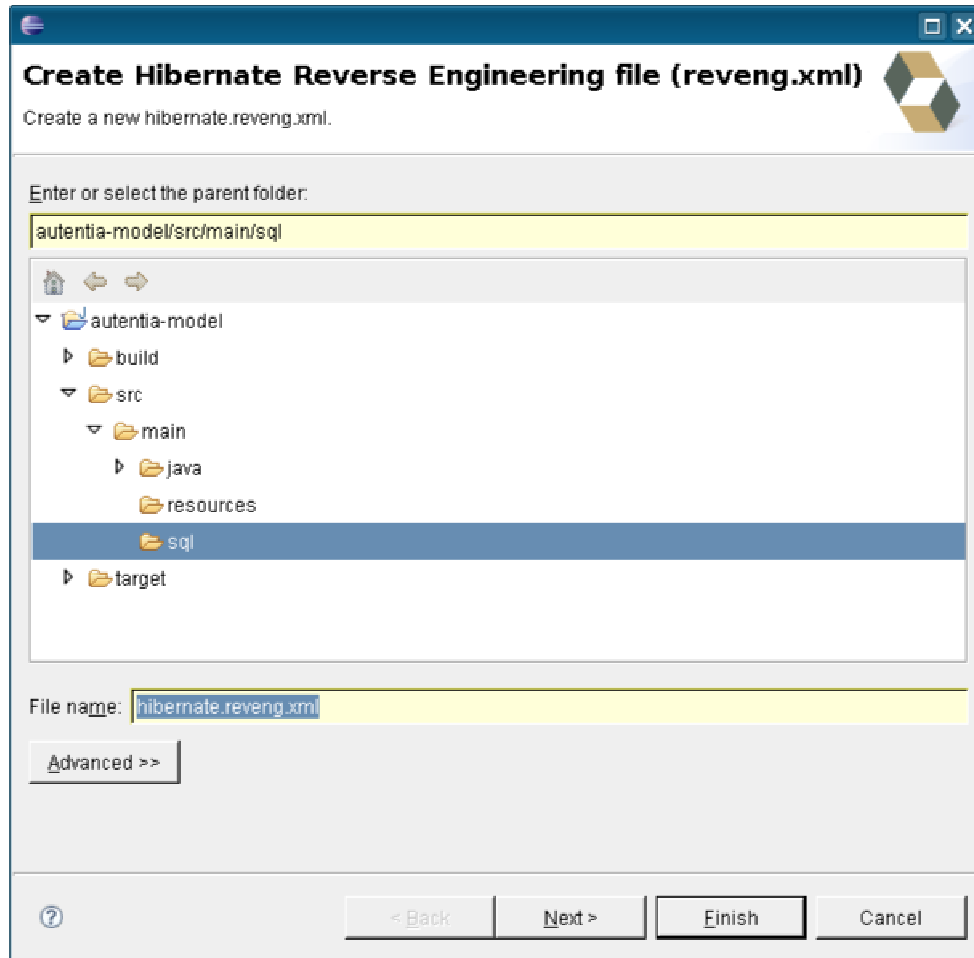
Para refinar este tipo de cosas podemos hacerlo a mano o usar el fichero `revenge.xml`. El uso de este fichero es recomendable ya que nos permite regenerar las clases sin perder los cambios.

Para crear este fichero, sobre la primera pantalla que veíamos al configurar el "launch configuration" vemos que hay un campo "reveng.xml" con un botón "Setup...". Pulsamos este botón.

Le decimos que queremos crear un nuevo fichero "Create new...".

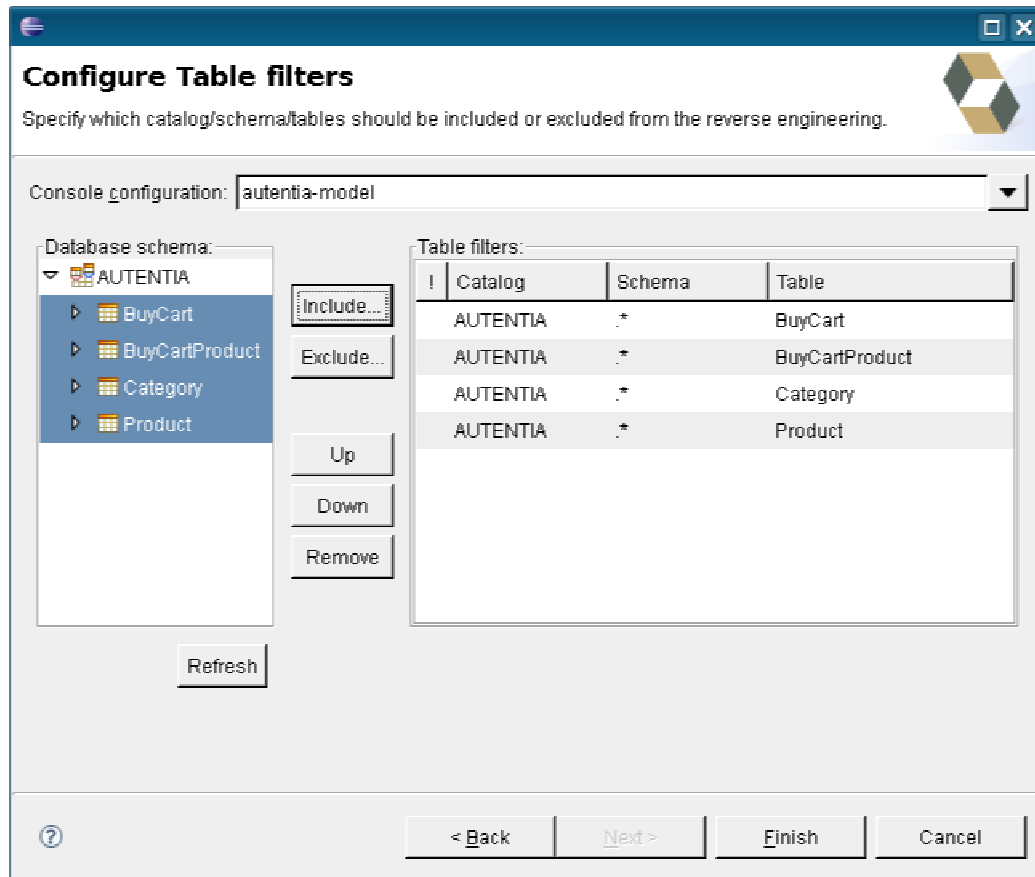


Indicamos donde se debe guardar el fichero (debería ser un directorio que luego quede fuera de nuestra distribución).

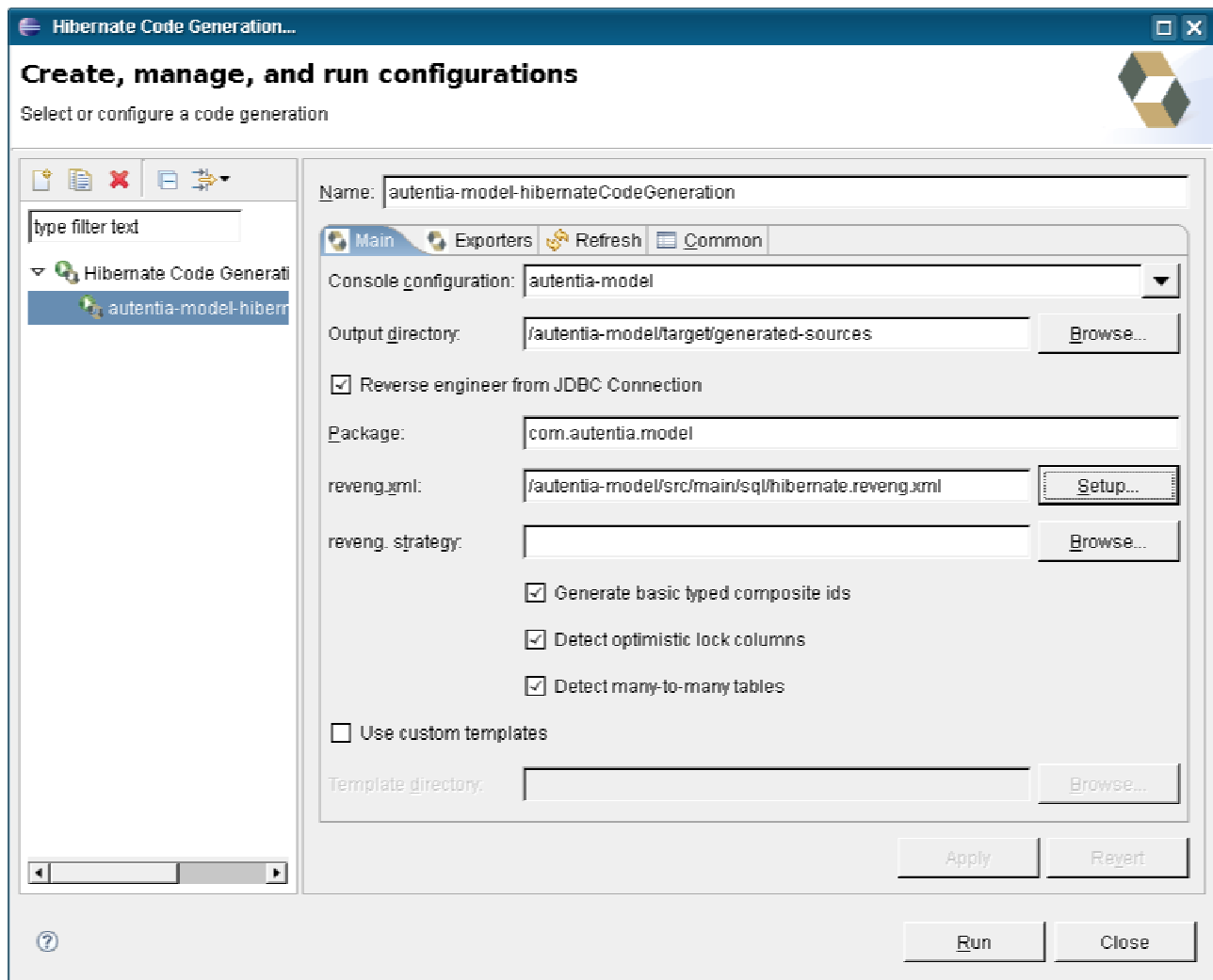


Ahora seleccionamos la configuración de consola que creamos anteriormente y pulsamos sobre el botón "Refresh". Con esto nos aparecerá a la izquierda nuestro esquema de la base de datos con las tablas, esto nos permite seleccionar las tablas de las que queremos generar código (por defecto lo que hicimos en el punto anterior genera código para todas las tablas del esquema).

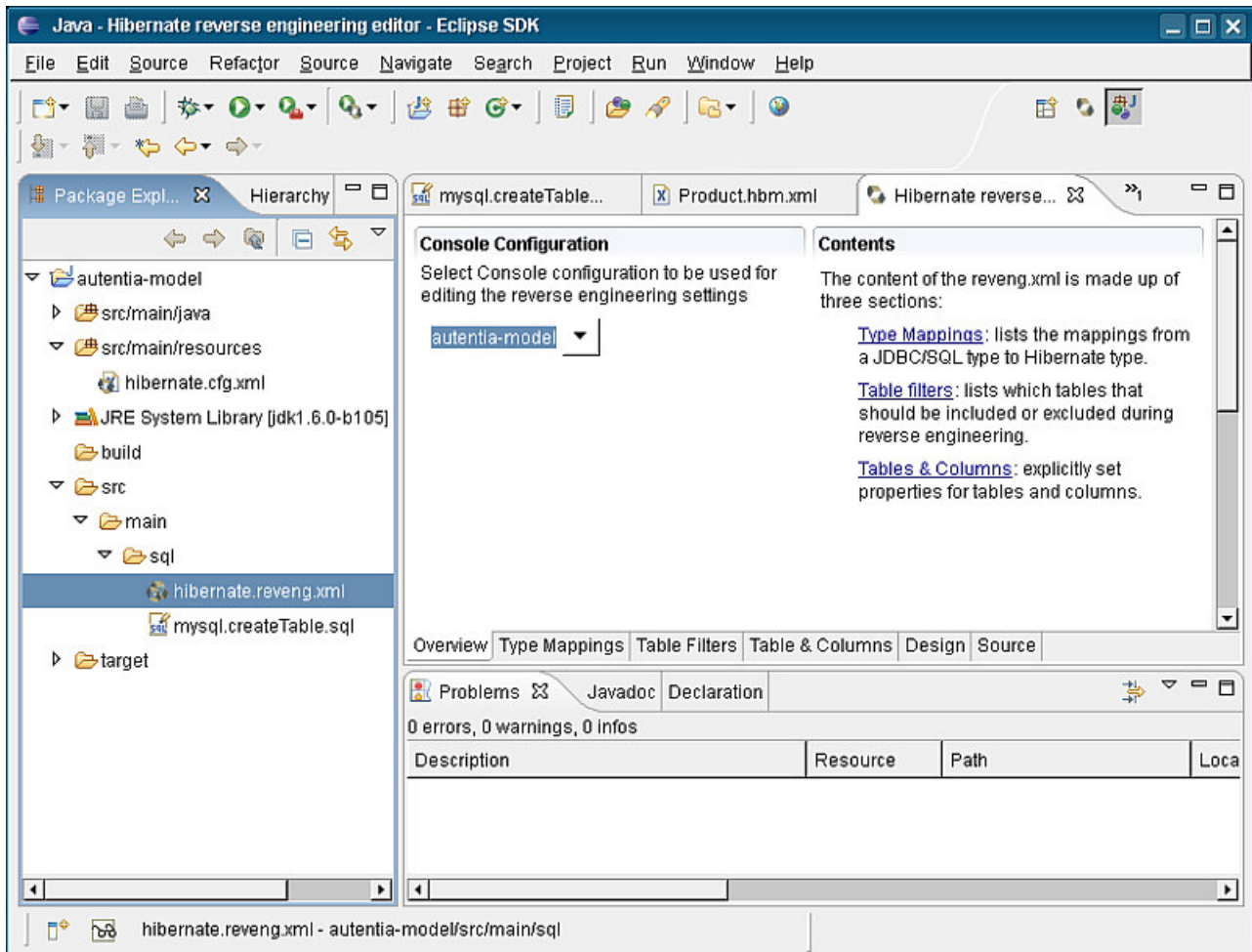
Marcamos las que nos interesan y pulsamos sobre "Include...". Veremos como pasan al lado de la derecha.



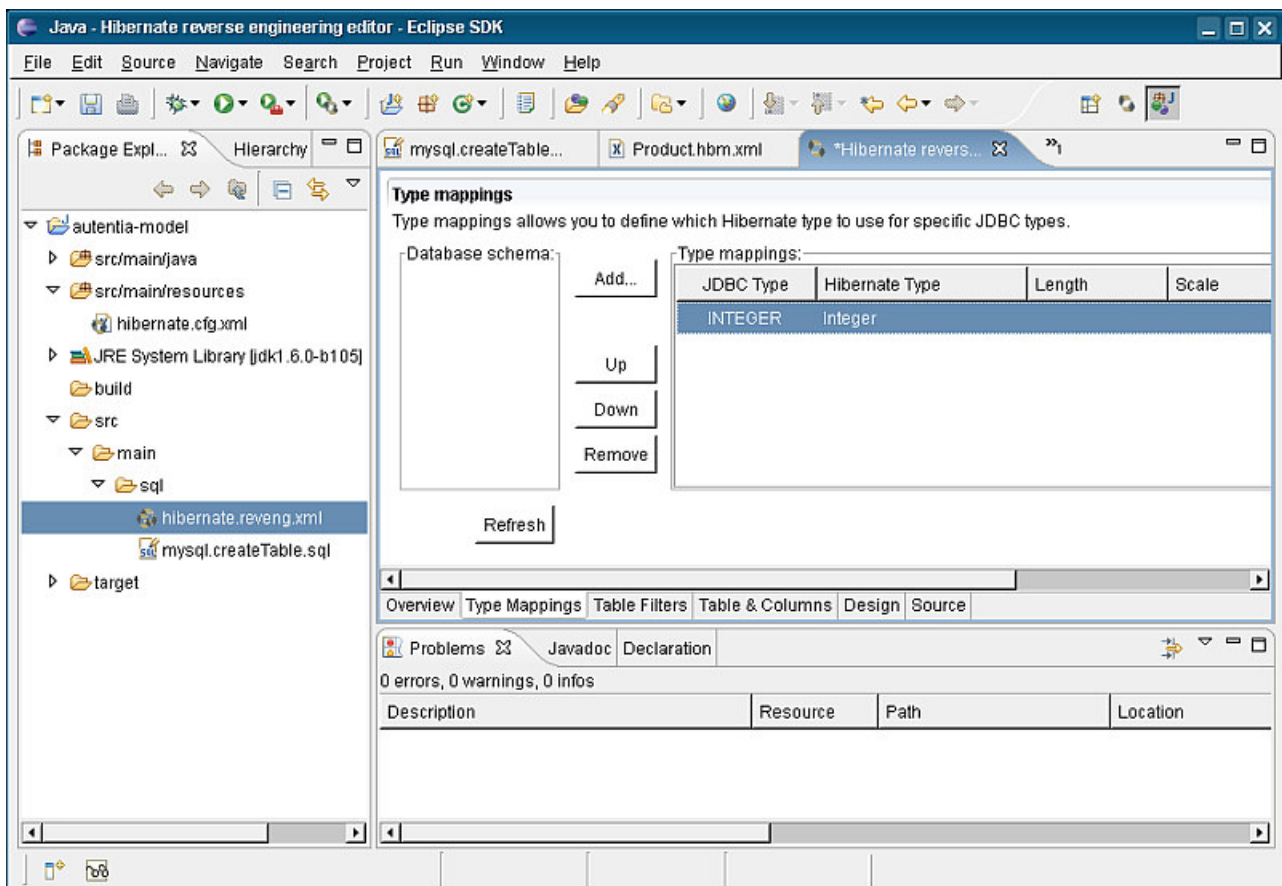
Pulsamos sobre "Finish" y volvemos a la pantalla de configuración de "launch configuration". Podemos ver como ya aparece el nombre del fichero que acabamos de crear.



Ahora podemos localizar el fichero en nuestro explorador de paquetes de Eclipse y abrirlo. Veremos que nos aparece un editor específico, que nos permite, de forma más menos visual, modificar este fichero.



En la pestaña "Type Mappings" podemos indicar como se deben mapear los tipos de la base de datos con los tipos de Java. Por ejemplo, en esta sección podemos añadir un mapeo del tipo INTEGER de JDBC al tipo `java.lang.Integer` de Java. Con esto solucionamos el tema de los identificadores numéricos, consiguiendo que sean nulos.



Si queremos que las relaciones de la base de datos no se conviertan en asociaciones bidireccionales, tendremos que modificar a mano fuente del fichero `revenge.xml` (lamentablemente las Hibernate Tools todavía no soportan hacer esto de forma visual). Para ello podemos pinchar sobre la pestaña "Source".

Un ejemplo de fichero sería el siguiente:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-reverse-engineering PUBLIC "-//Hibernate/Hibernate Reverse Engineering DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-reverse-engineering-3.0.dtd" >
<hibernate-reverse-engineering>
<type-mapping>
<sql-type jdbc-type="INTEGER" hibernate-type="java.lang.Integer"></sql-type>
</type-mapping>
<table-filter match-catalog="AUTENTIA" match-name="BuyCart"/>
<table-filter match-catalog="AUTENTIA" match-name="BuyCartProduct"/>
<table-filter match-catalog="AUTENTIA" match-name="Category"/>
<table-filter match-catalog="AUTENTIA" match-name="Product"/>

<table name="Product">
<foreign-key constraint-name="fk_category_id">
<set exclude="true"/>
</foreign-key>
</table>
<table name="BuyCartProduct">
<foreign-key constraint-name="fk_product_id">
<set exclude="true"/>
</foreign-key>
<foreign-key constraint-name="fk_buyCart_id">
<many-to-one exclude="true"/>
</foreign-key>
</table>
</hibernate-reverse-engineering>
```

Vemos como tenemos los mapeos de los tipos de datos, las tablas que se tienen que usar al hacer la ingeniería inversa, y luego como se tienen que hacer las asociaciones.

En el ejemplo hay una relación 1:n entre la tabla "Category" y "Product". De forma que por defecto se nos creará un atributo en la clase "Product" que apunta a la categoría correspondiente, y en la clase "Category" tendremos una lista de todos los productos que tienen esa categoría. Lo que estamos haciendo en el ejemplo es que esta asociación sea unidireccional de forma que desde la clase "Product" podremos acceder a su categoría, pero desde la clase "Category" no podremos acceder a todos los productos.

Nótese que "`fk_category_id`" es el nombre de la "constraint" que hay en el campo de la tabla "Product" donde se guarda la clave ajena de la tabla "Category". Mostramos el script de creación para aclarar este párrafo:

```
CREATE TABLE Category (
  id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  name VARCHAR(45) NOT NULL,
  description VARCHAR(255) NULL,
  PRIMARY KEY(id)
)
engine=innodb default charset=utf8 collate=utf8_spanish_ci;
CREATE TABLE Product (
  id INTEGER UNSIGNED NOT NULL AUTO_INCREMENT,
  categoryId INTEGER UNSIGNED NOT NULL,
  name VARCHAR(45) NULL,
  description VARCHAR(255) NULL,
  price INTEGER UNSIGNED NOT NULL,
  PRIMARY KEY(id),
  constraint fk_category_id foreign key (categoryId) references Category (id)
)
engine=innodb default charset=utf8 collate=utf8_spanish_ci;
```

Ahora, gracias al fichero `revenge.xml` podemos volver a generar las clases, pero esta vez el resultado obtenido se ajustará mucho más a nuestras necesidades.

## 8. Conclusiones

En este tutorial hemos visto como podemos ahorrar tiempo de desarrollo gracias a las Hibernate Tools, ya que nos permiten generar gran parte del código necesario para el acceso a datos: los POJOs, los ficheros de configuración, ...

El uso de Hibernate es muy recomendable para aislarnos de la base de datos y facilitarnos el acceso a la misma. Con herramientas como Hibernate Tools conseguimos facilitar mucho más el trabajo.

## 9. Sobre el autor

Alejandro Pérez García, Ingeniero en Informática (especialidad de Ingeniería del Software)

Socio fundador de Autentia (Formación, Consultoría, Desarrollo de sistemas transaccionales)

<mailto:alejandropg@autentia.com>

Autentia Real Business Solutions S.L. - "Soporte a Desarrollo"

<http://www.autentia.com>



This work is licensed under a [Creative Commons Attribution-NonCommercial-No Derivative Works 2.5 License](http://creativecommons.org/licenses/by-nc-nd/2.5/).  
[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**

J2EE, EJBs, Struts...

[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.

|                                 |                                       |
|---------------------------------|---------------------------------------|
| <b>Subscribirse a Novedades</b> |                                       |
| <b>e-mail</b>                   | <input type="text"/>                  |
|                                 | <input type="button" value="Enviar"/> |

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

### Nombre Corto

[Profiling Java con Eclipse Test Performance Tools Platform \(TPTP\)](#)

[Análisis de rendimiento \(Profiling\) de aplicaciones web con eclipse](#)

[Optimización Java con Eclipse Profiler Plugin](#)

[Manejar dos bases de datos distintas con Hibernate](#)

[Mejora de la calidad del código fuente con Eclipse](#)

[Manual Básico de Eclipse JET](#)

[Hibernate 3 y los tipos de datos para cadenas largas](#)

[Optimizando código Java con Eclipse Test Performance Tools Platform](#)

[Hibernate 3.1, Colecciones, Fetch y Lazy](#)

[Creación automática de recursos](#)

### Descripción

En este tutorial, aprenderemos de una manera sencilla como analizar nuestro código visualizando sus estadísticas de ejecución con Eclipse Test Performance Tools Platform (TPTP)

En este tutorial se va a explicar como analizar el rendimiento de nuestras aplicaciones web con una herramienta propia de Eclipse, llamada Eclipse TPTP.

Alejandro Pérez nos enseña como analizar el rendimiento de nuestras aplicaciones con Eclipse Profiler Plugin.

Alejandro Pérez nos enseña como manejar dos bases de datos distintas con Hibernate

En este pequeño tutorial se muestra una de esas multiples opciones de la plataforma de desarrollo Eclipse que muchos de nosotros no vemos y que nos hubieran ahorrado un gran esfuerzo si lo hubieramos conocido en su día.

Este tutorial nos introducirá en Java Emitter Template (JET) que nos sirve para generar automáticamente código fuente a partir de plantillas

En este tutorial se contará la experiencia que hemos tenido a la hora de manejar los diferentes tipos de datos existentes para grandes cadenas de texto, tales como el tipo Clob de Oracle, o los tipos TEXT de MySQL y SQLServer, utilizando la última versión

En este tutorial vamos a aprender como usar Eclipse Test Performance Tools Platform (TPTP), que nos permite analizar nuestro código

En este tutorial vamos a ver cómo se comportan ciertas relaciones, y cómo podemos optimizar las consultas a la base de datos con Hibernate

En este tutorial aprendereis como utilizar la herramienta middlegen para generar distintas capas de persistencia (CMP 2.0, JDO, Hibernate, Torque), a partir de un modelo físico de

[Hibernate con Middlegen](#)

datos, de un modo automático, mediante el uso de la herramienta middlegen

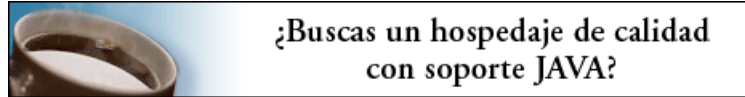
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

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](#)



[www.AdictosAlTrabajo.com](http://www.AdictosAlTrabajo.com) Optimizado 800X600