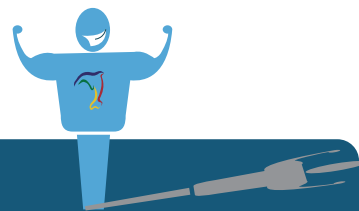


¿Qué ofrece Autentia Real Business Solutions S.L?

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

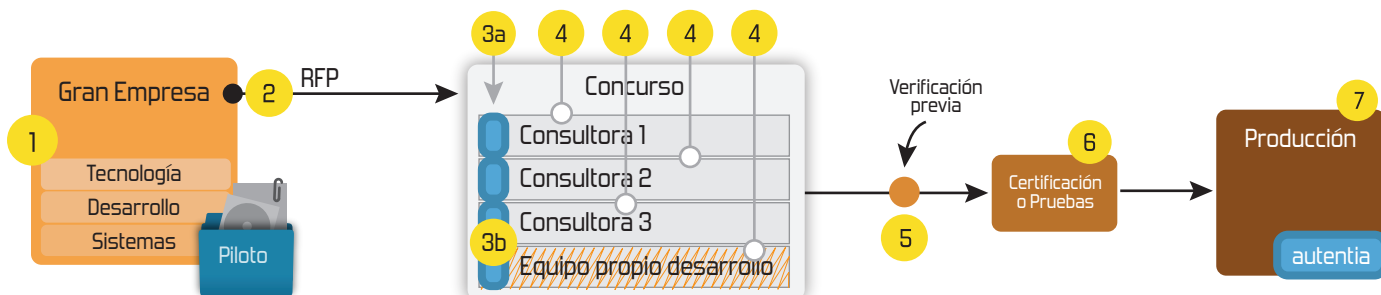
1. Desarrollo de componentes y proyectos a medida



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](#) | [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 echarle una mano: Desarrollo y arquitectura Java/J2EE Asesoramiento tecnológico Web Formación / consultoría integrados en tu proyecto No te cortes y contacta: 655 99 11 72 rcanales@autentia.com.	
---	--	---

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

[Kodo JDO from SolarMetric](#)

Proven object relational mapping using Java Data Objects
www.solarmetric.com

[Download Java Report Tool](#)

Drag & drop Report creation for Java applications. Powerful charts!
www.IntelliVIEW.com

[NetBeans IDE 4.1](#)

Create EJBs & Web Services, ant-based project system and more.
www.netbeans.org

Anuncios Goooooogle

Anunciarse en este sitio

JDO con TJDO

Hoy vamos a ver otra implementación gratuita de JDO llamada TJDO.

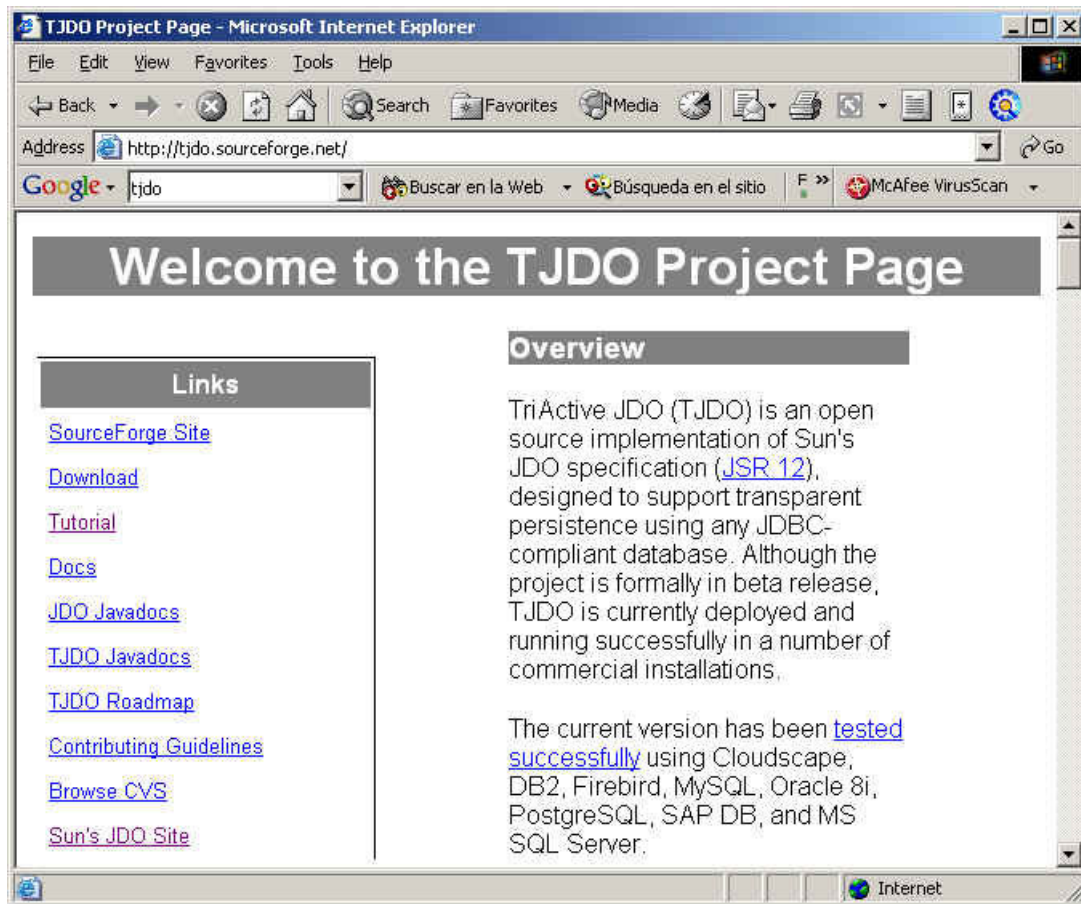
En este caso, se parece más al modo estándar de realizar las implementaciones JDO, las clases se ven transformadas a partir de sí binario, para adaptarse a la especificación JDO.

Los pasos que vamos a realizar son:

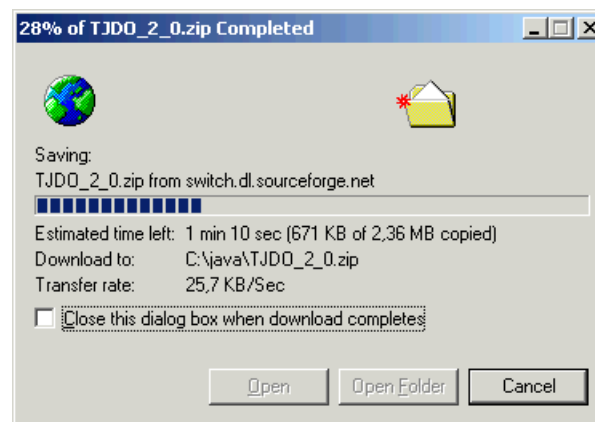
- Descargarnos el entorno
- Descargar los jar que necesita para que funcione
- Construir el código de la clase persistencia.
- Construir la clase cliente
- Crear el fichero descriptor JDO
- Cambiamos el script ant de construcción.
- Ampliar nuestra clase para añadirles los métodos necesarios para que funciones JDO
- Ejecutar el ejemplo, creando las tablas si no existen.

Descargamos el código

Vamos a la página principal <http://tjdo.sourceforge.net/>



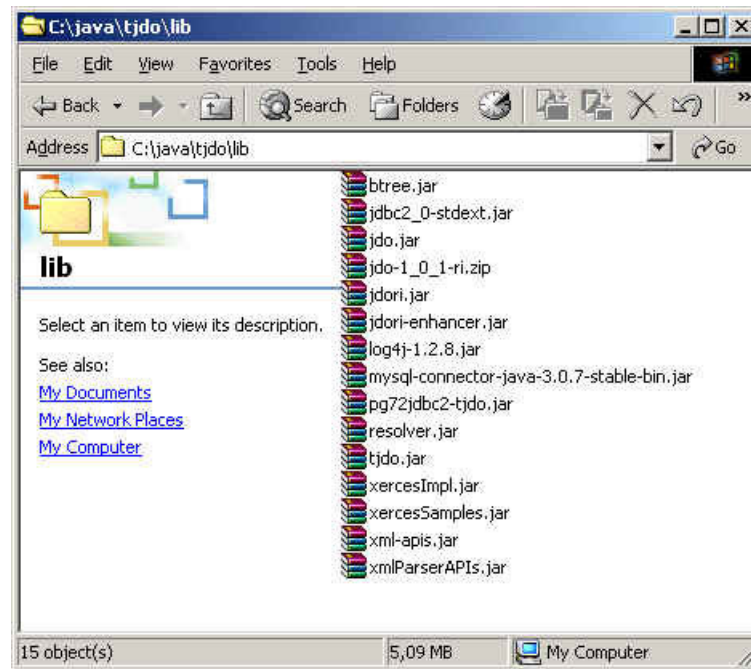
Y nos [descargaremos el Software](#).



Siguiendo el [tutorial oficial](#) podemos ver los pasos a seguir.

Debemos descargarnos algunos ficheros para que nos funcione correctamente (Xerces, extensiones de JDBC, JDO de referencia, driver JDBC de MySQL, Log4J, etc)

Con estos ficheros funciona



Nuestras clases

Creamos nuestra clase persistente

```
package roberto;

/**
 * Tutoriales.java
 *
 * Created on February 19, 2004, 1:16 PM
 * @author Roberto Canales
 */
public class Tutoriales {

    private int id;
    private String titulo;
    private String descripcion;
    private String enlace;

    public Tutoriales() {
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public java.lang.String getTitulo() {
        return titulo;
    }

    public void setTitulo(java.lang.String titulo) {
        this.titulo = titulo;
    }

    public java.lang.String getDescripcion() {
        return descripcion;
    }

    public void setDescripcion(java.lang.String descripcion) {
        this.descripcion = descripcion;
    }

    public java.lang.String getEnlace() {
        return enlace;
    }

    public void setEnlace(java.lang.String enlace) {
        this.enlace = enlace;
    }
}
```

Y la clase cliente

```
package roberto;
```

```

import java.util.*;
import javax.jdo.*;

public class Test
{
    public static void main(String[] args)
    {
        Properties props = new Properties();

        // Set the PersistenceManagerFactoryClass to the TJDO class.
        props.setProperty("javax.jdo.PersistenceManagerFactoryClass",
            "com.triactive.jdo.PersistenceManagerFactoryImpl");

        props.setProperty("javax.jdo.option.ConnectionDriverName", "com.mysql.jdbc.Driver");
        props.setProperty("javax.jdo.option.ConnectionURL", "jdbc:mysql://localhost/tutoriales");
        props.setProperty("javax.jdo.option.ConnectionUserName", "root");
        props.setProperty("javax.jdo.option.ConnectionPassword", "");

        PersistenceManagerFactory pmf = JDOHelper.getPersistenceManagerFactory(props);
        PersistenceManager pm = pmf.getPersistenceManager();

        Transaction tx = pm.currentTransaction();
        try
        {
            tx.begin();

            Tutoriales nuevo = new Tutoriales();
            nuevo.setTitulo("Utilidades JDO");
            nuevo.setEnlace("jdotools");
            nuevo.setDescripcion("Os mostramos como utilizar herramientas automáticas con JDO");

            pm.makePersistent(nuevo);
            tx.commit();
        }
        finally
        {
            if (tx.isActive()) tx.rollback();
            pm.close();
        }

        return;
    }
}

```

Fichero JDO

Creemos el fichero JDO. Lo único especial son las extensiones de fabricante

```

<?xml version="1.0"?>
<!DOCTYPE jdo PUBLIC "-//Sun Microsystems, Inc.//DTD Java Data Objects Metadata 1.0//EN"

"http://java.sun.com/dtd/jdo_1_0.dtd">

<jdo>
  <package name="roberto">
    <class name="Tutoriales">
      <field name="id"/>
      <field name="enlace">
        <extension vendor-name="triactive" key="length" value="max 255"/>
      </field>
      <field name="descripcion">
        <extension vendor-name="triactive" key="length" value="max 255"/>
      </field>
      <field name="titulo">
        <extension vendor-name="triactive" key="length" value="max 255"/>
      </field>
    </class>
  </package>
</jdo>

```

Ojo... debemos copiar este fichero al directorio donde están las clases (.class)

Creación del Script ant

Reutilizando conocimientos, automatizamos la construcción, transformación y ejecución con ANT.

El **build.properties**

```

#####
#
source.dir=src
source.java.dir=${source.dir}

build.dir=.
build.classes.dir=${build.dir}/

lib.dir=lib

```

```
target.dir=target
```

Y el **build.xml**

```
<project name="construir" default="compilar" basedir=".">

  <property file="build.properties"/>

  <path id="ejecucion-classpath">
    <fileset dir=".">
      <include name="**/*.jar"/>
      <include name="**/*.zip"/>
    </fileset>
    <pathelement path="."/>
  </path>

  <target name="compilar">
    <timestamp/>
    <mkdir dir="${build.dir}"/>
    <mkdir dir="${build.classes.dir}"/>
    <javac srcdir="${source.java.dir}" destdir="${build.classes.dir}">
      <classpath refid="ejecucion-classpath"/>
    </javac>
  </target>

  <target name="jar" depends="compilar">
    <mkdir dir="${target.dir}"/>
    <jar index="true" jarfile="${target.dir}/${jar.name}">
      <fileset dir="${build.classes.dir}">
        <include name="**/*.class"/>
      </fileset>
      <fileset dir="${build.resource.dir}">
        <include name="**/*"/>
      </fileset>
    </jar>
  </target>

  <target name="ejecuta" depends="">
    <java fork="no"
      classname="roberto.Test"
      failonerror="true"
      classpathref="ejecucion-classpath">

      <sysproperty key="com.triactive.jdo.autoCreateTables" value="true"/>

    </java>
  </target>

  <target name="enhance" depends="">
    <java fork="no"
      classname="com.triactive.jdo.enhance.SunReferenceEnhancer"
      failonerror="true"
      classpathref="ejecucion-classpath">

      <arg value="src/Tutoriales.jdo"/>

    </java>
  </target>

  <target name="enhanceR" depends="">
    <java fork="yes" failonerror="yes" classname="com.sun.jdori.enhancer.Main"

    classpathref="ejecucion-classpath">
      <arg line="-v -f -d . src/Tutoriales.jdo roberto/Tutoriales.class"/>
    </java>
  </target>
</project>
```

Tenemos ya los comandos para compilar, empaquetar, expandir las clases persistentes (enhance, por dos sistemas distintos) y ejecutar.

La estructura queda tal como (sin el directorio de documentación):

```
C:.\
|  build.properties
|  build.xml
|  jdo.log
|  LICENSE.txt
|  log4j.properties
|  README.txt
|
|  --- lib
|      btree.jar
|      jdbc2_0-stdext.jar
|      jdo-1_0_1-ri.zip
|      jdo.jar
```

```

jdori-enhancer.jar
jdori.jar
log4j-1.2.8.jar
mysql-connector-java-3.0.7-stable-bin.jar
pg72jdbc2-tjdo.jar
resolver.jar
tjdo.jar
xercesImpl.jar
xercesSamples.jar
xml-apis.jar
xmlParserAPIs.jar

```

```

roberto
Test.class
Tutoriales.class
Tutoriales.jad
Tutoriales.jdo

```

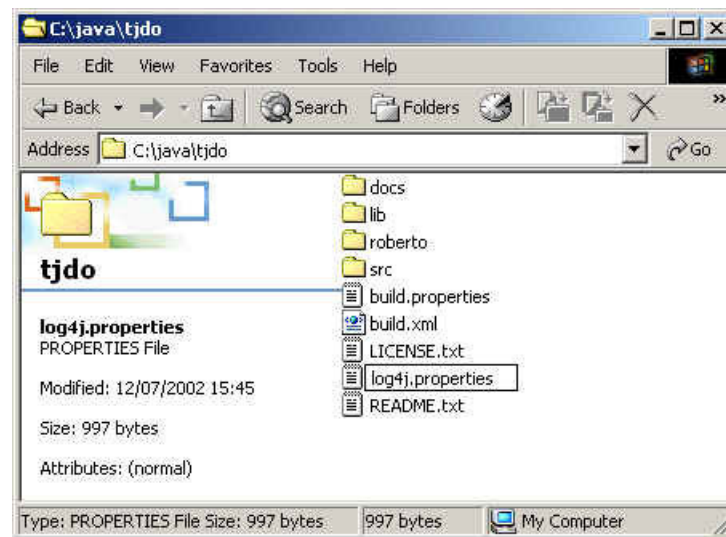
```

src
Test.java
Tutoriales.java
Tutoriales.jdo

```

Configuramos LOG4J

Solo tenemos que renombrar el fichero de configuración ...



Compilamos, y ampliamos las clases

Lo hacemos desde la línea de comando, utilizando ANT.

```

C:\java\tjdo>ant
Buildfile: build.xml

compilar:
[javac] Compiling 2 source files to C:\java\tjdo

BUILD SUCCESSFUL
Total time: 2 seconds

C:\java\tjdo>ant enhance
Buildfile: build.xml

enhance:
[java] Enhancing class roberto.Tutoriales
[java] done.

BUILD SUCCESSFUL
Total time: 2 seconds

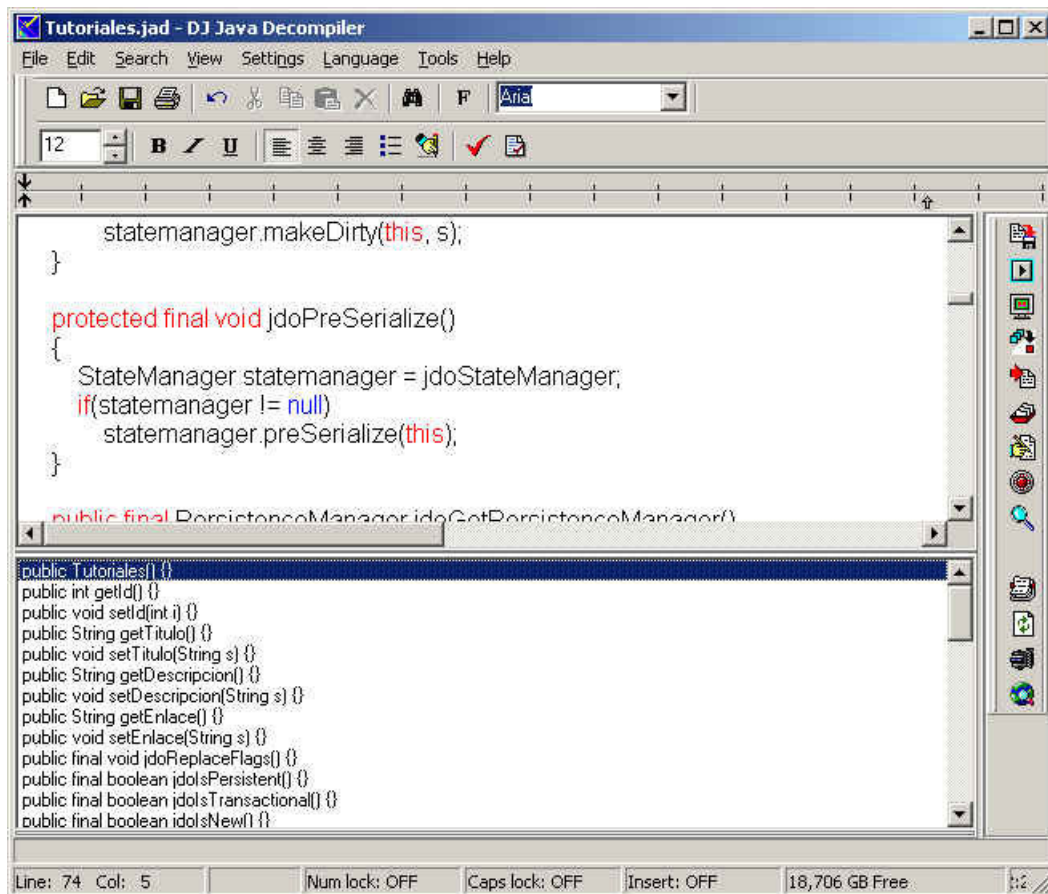
C:\java\tjdo>ant ejecuta
Buildfile: build.xml

ejecuta:

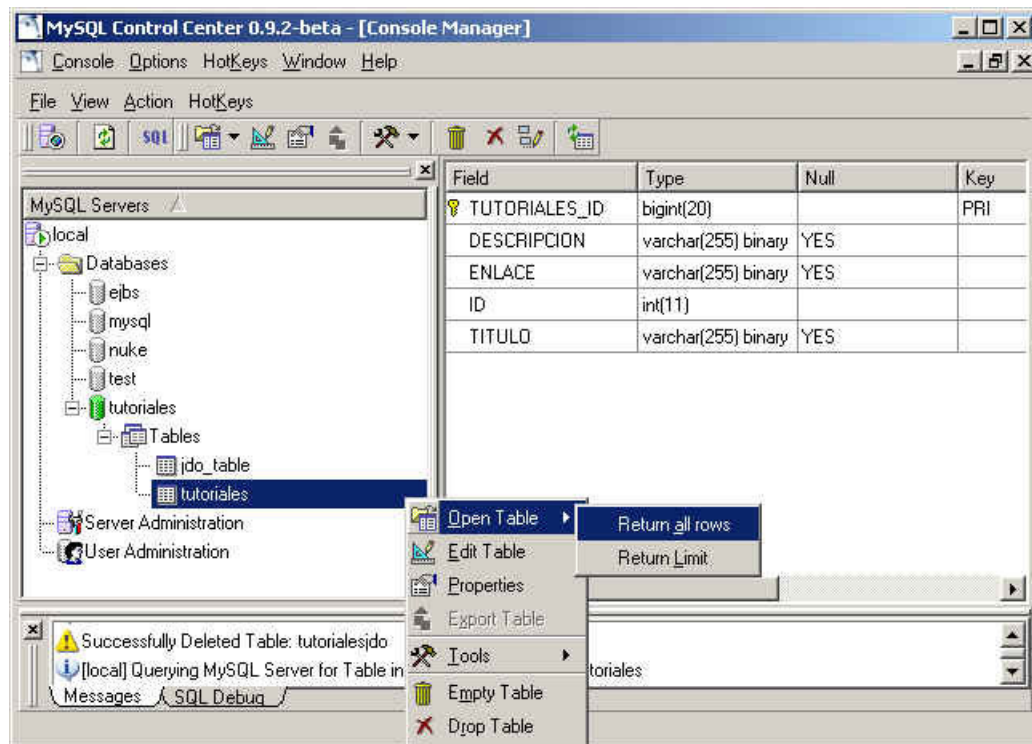
BUILD SUCCESSFUL
Total time: 3 seconds

```

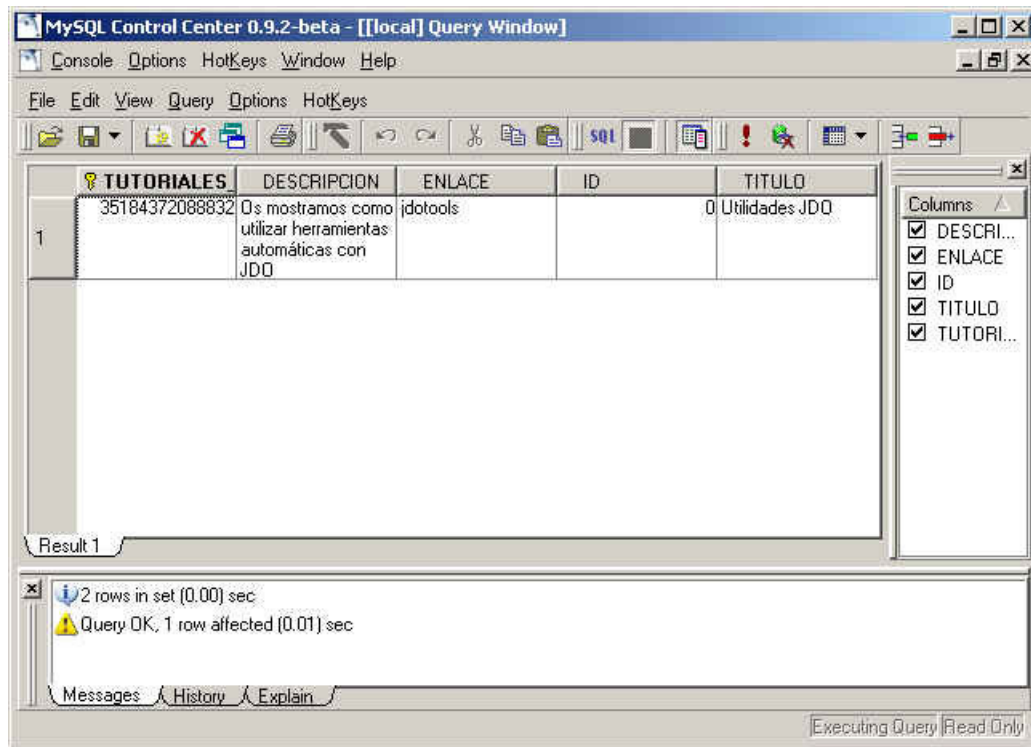

Y podemos ver, decompilando la clase generada, como se ha ampliado el código



Y también podemos ver como se han creado las tablas en la base de datos e insertado el registro



Podemos ver nuestro registro.



Conclusión

Ya podemos entender más fácilmente concepto de JDO.... aunque tenemos que utilizar herramientas donde todo esto esté integrado

Pronto os mostraremos pluggins de NetBeans y Eclipse... con lo que nuestra vida es más sencilla.

[Sobre el Autor..](#)

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

soluciones reales para su negocio

[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

[Reingeniería JDO con Druid](#)

[Generación automática de código JDBC](#)

[CMP Entity Beans y MySql](#)

[JDBC y MySql](#)

Descripción

Os mostramos como crear vuestras clases y descriptores JDO, de tablas existentes, con la herramienta gratuita Druid.

En este tutorial os enseñamos como, sin conocimiento de JDBC, crear vuestro programas en Java, gracias a JDBCtest.

Os mostramos como crear un Entity Bean con persistencia controlada por el servidor, configurado para usar MySql

En el tutorial anterior vimos como instalar MySQL en Windows, ahora vamos a ver como acceder desde una aplicación Java.

[Desarrollo de Entity Beans](#)

Os mostramos como construir un Entity Bean básico y desplegarlo en el servidor J2EE de referencia. Lo usaremos como base de buenas prácticas J2EE

[Web Services en tu IPAQ](#)

Cesar Crespo nos enseña como programar accesos Web Services desde tu IPAQ en Visual C++ con PocketSOAP, Apache SOAP y Axis

[AspectJ, Programación con Aspectos](#)

Os mostramos como configurar AspectJ (extensión Java para la programación basada en aspectos) y un pequeño ejemplo para medir la velocidad de una función sin alterar su código.

[Soporte de Asserts en Java 1.4.x](#)

Os mostramos como utilizar los asserts en Java (disponibles a partir de la versión 1.4)

[Novedades en Java 1.5](#)

Ya está disponible la versión Beta del J2SDK 1.5. Os mostramos algunas de las nuevas características introducidas en el lenguaje Java: Clases genéricas, enumeraciones, bucles simplificados, etc.

[JDO con OJB](#)

Os mostramos como configurar el entorno OJB de apache para construir la primera aplicación JDO

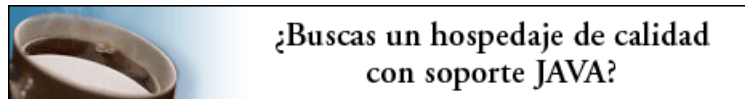
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

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 Optimizado 800X600