

¿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](#)

	<p>Tutorial desarrollado por: Roberto Canales Mora 2003-2005 Creador de AdictosAlTrabajo.com y Director General de Autentia S.L.</p> <p>Recuerda que me puedes contratar para echarle una mano:</p> <p>Desarrollo y arquitectura Java/J2EE Asesoramiento tecnológico Web Formación / consultoría integrados en tu proyecto</p> <p>No te cortes y contacta: 655 99 11 72 rcanales@autentia.com.</p>	 autentia real business solutions
---	--	---

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

Curso Web J2EE

Curso Avanzado en Desarrollo Web con J2EE
www.eps.mondragon.edu/caj2ee

Kodo JDO from SolarMetric

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

Anuncios Goooooogle

Anunciarse en este sitio

Druid y JDO

En los dos anteriores [tutoriales](#), hemos visto como comenzar a trabajar, de un modo práctico, con JDO.

[En el último](#), vimos como incluso se podrían crear las tablas del sistema sin realizar ningún esfuerzo adicional.

Pero la realidad normalmente es bien distinta..... No estamos siempre construyendo nuevas aplicaciones sino que éstas ya existen y los modelos de datos están consolidados (que es lo más importante de un sistema). Yo diría más, sin un buen modelo de datos, y un buen administrador de esa base de datos, va a ser imposible que nuestros desarrollos sean eficientes y extensibles.

Si ya existe la base de datos, necesitamos mapear (si me permitís la expresión) las tablas a nuestros objetos (y crear nuestras clases, por supuesto)

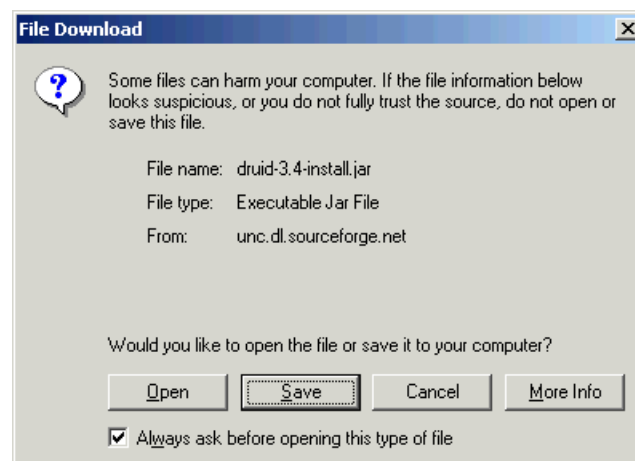
Vamos a utilizar un producto gratuito llamado Druid (que vale para muchas cosas más) para crear nuestras clases de un modo automático. También nos crea el xml descriptor de JDO y el fichero **ant** para compilar.

Descarga

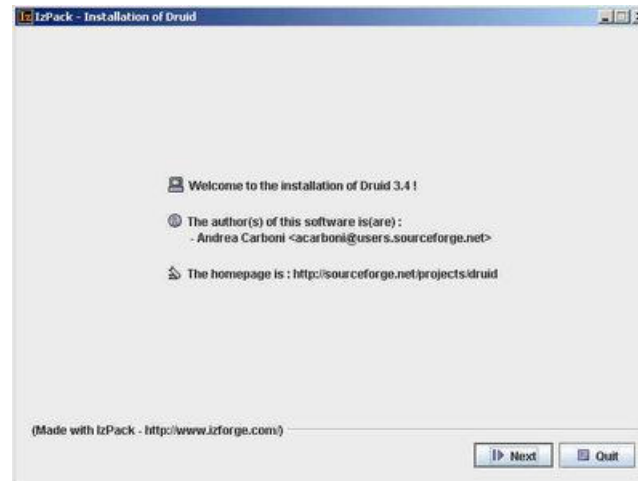
Vamos a la página principal

<http://prdownloads.sourceforge.net/druid/druid-3.4-install.jar?download>

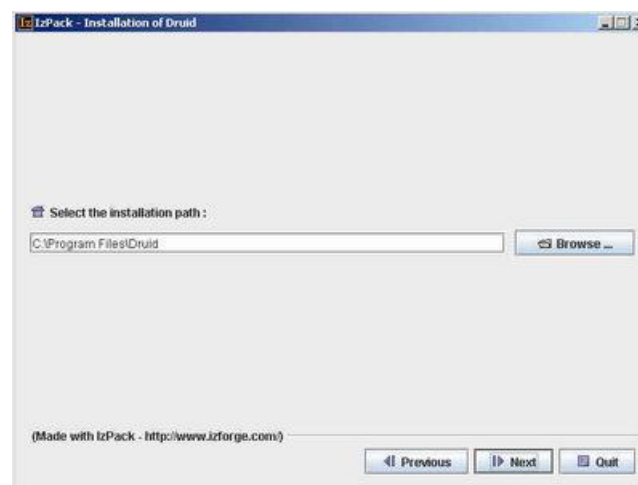
Y nos traemos el fichero



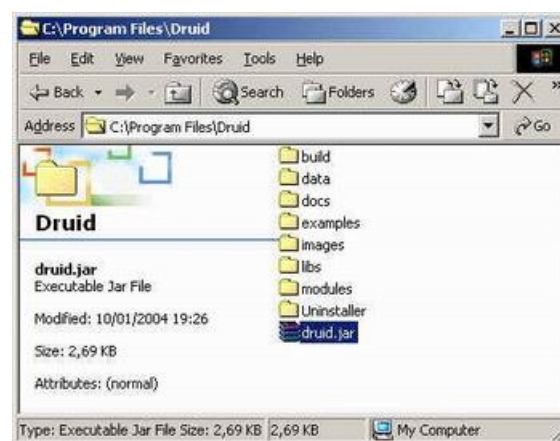
Hacemos doble-click sobre el **jar** y nos arranca el instalador (también podemos arrancarlo a mano)



Elegimos el trayecto de instalación ...



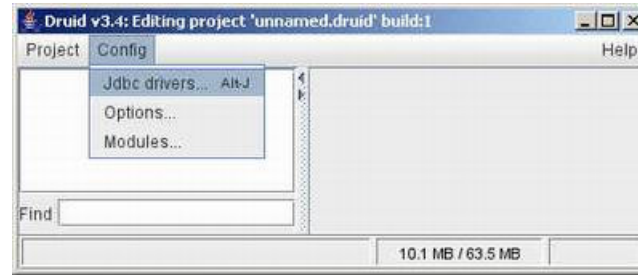
Instalamos el entorno y vemos como queda la estructura de este modo.



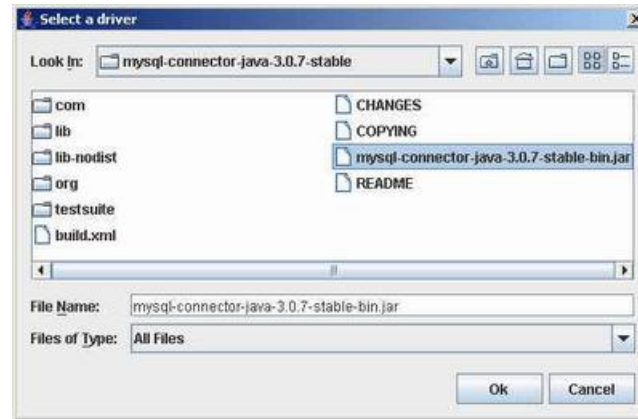
Arrancamos el programa y vemos el aspecto.

Configuración

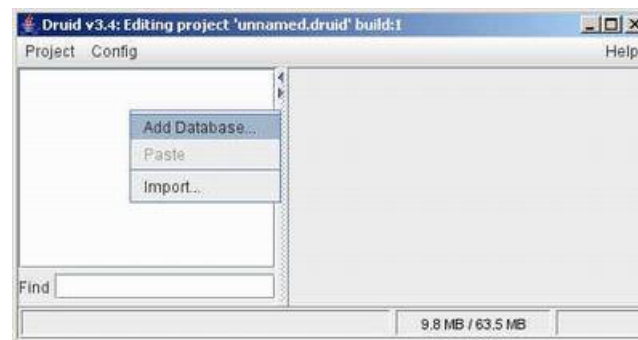
Vamos a seleccionar el driver de la base de datos a utilizar.



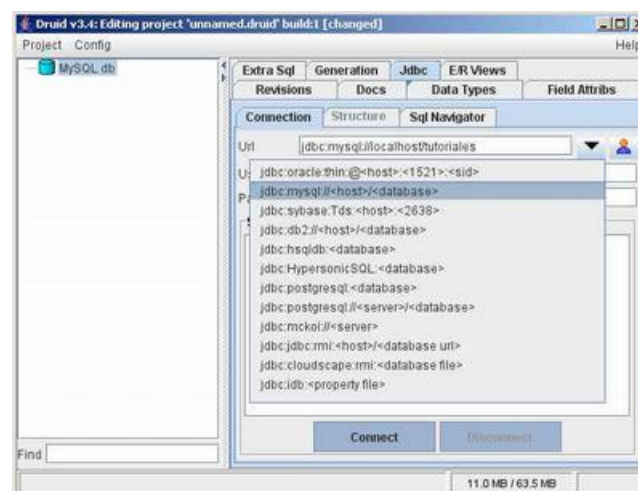
Seleccionamos el Driver de MySQL



Pinchamos el botón derecho del ratón sobre el panel central de **druid** y seleccionamos Añadir Base de Datos (Add Database)

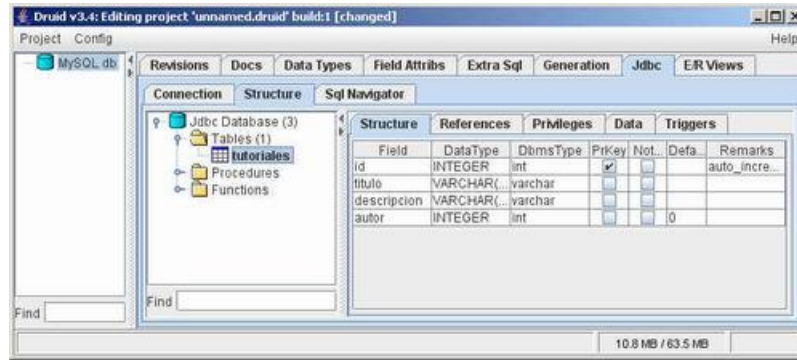


Elegimos el tipo de base de datos



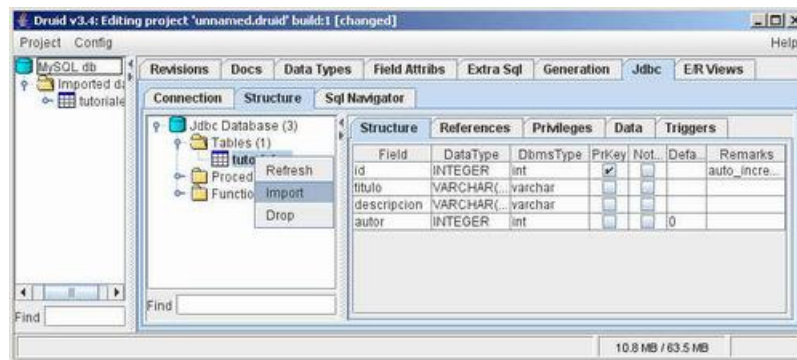
Elegimos la cadena de conexión: **jdbc:mysql://localhost/tutoriales** .

Ahora podemos navegar por la base de datos.

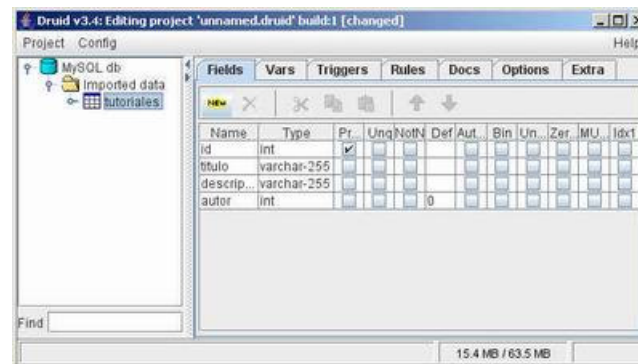


Mapeo

Cuando encontramos la tabla, pinchamos el botón derecho para que nos aparezca en el panel izquierdo



Ahora, pinchando sobre el panel principal, nos aparece en la parte derecha los atributos nos debemos fijar en el tipo (Type).



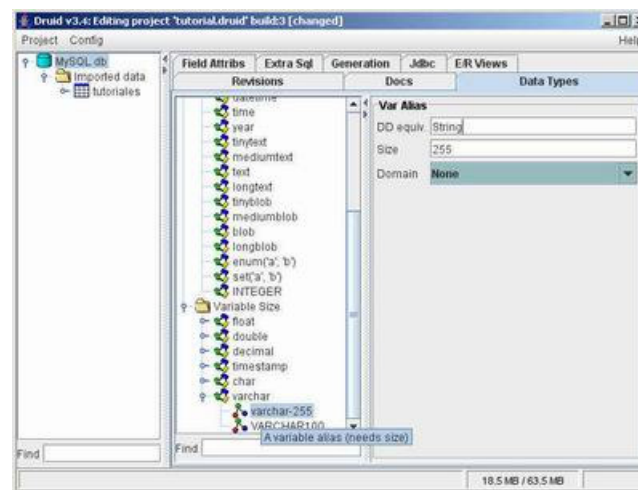
Podemos pinchar en el tipo y seleccionar dentro del siguiente desplegable.



En el panel izquierdo, pinchamos en la base de datos.

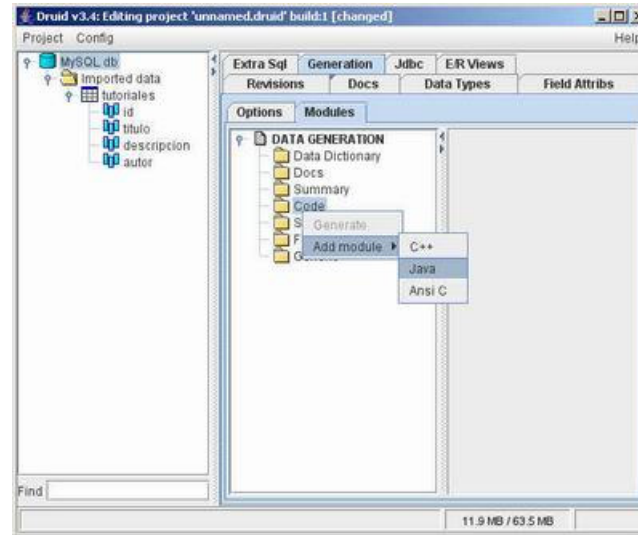
En la parte derecha, elegimos los tipos de datos y donde pone "DD equiv", escribimos el tipo Java al que queremos asociar las variables.

Vemos como hemos asociado el `varchar-255` a una cadenas de caracteres tipo `String`

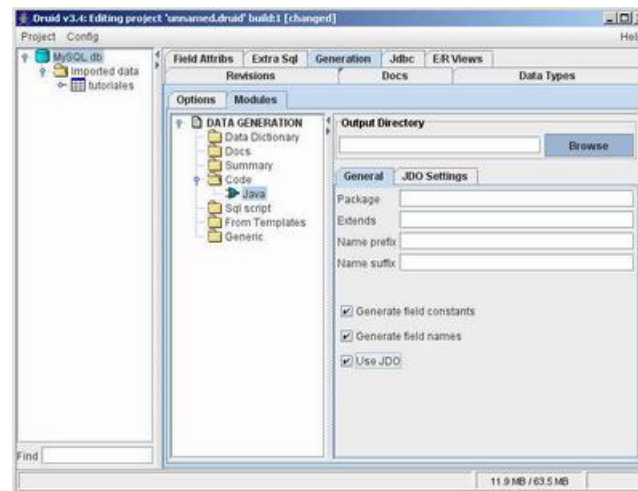


Generación

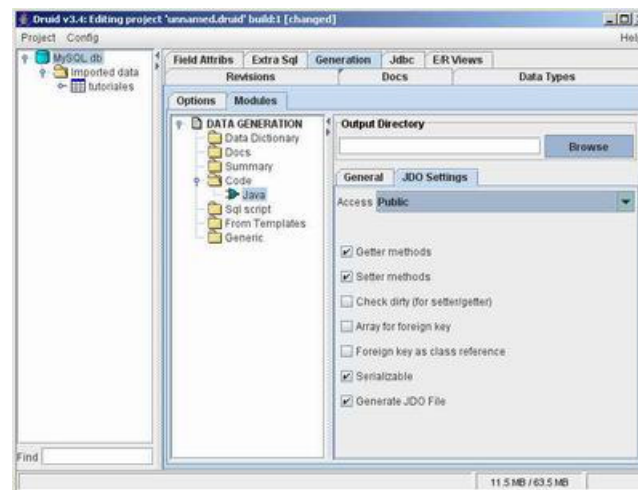
Seleccionamos ahora en el lado derecho la lengüeta de *Generation*. Y pinchando en *Code* (botón derecho) podemos añadir el módulo de Java.



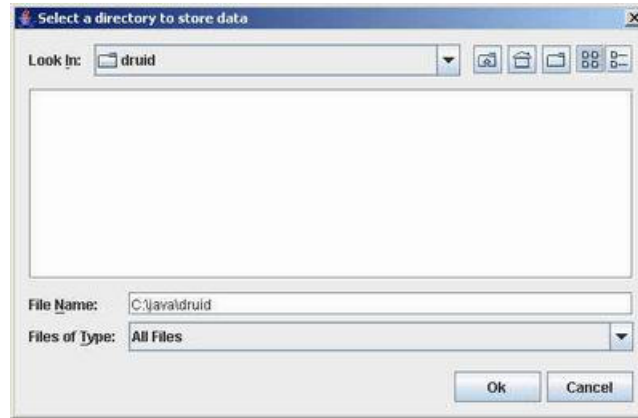
Debemos seleccionar el directorio de salida, nombre del paquete y otros parámetros generales



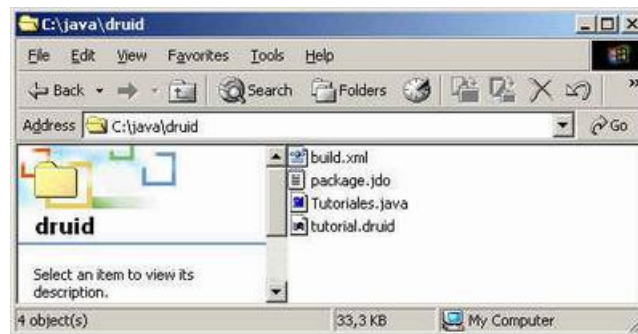
Y también los parámetros JDO.



Elegimos nuestro directorio destino



Y si pulsamos el botón derecho sobre nuestra tabla y pulsamos generar nos aparecen los siguientes elementos ...



Resultado

Esta es la clase Java generada.

```
//=====
//==  tutoriales.java                Build: 1
//=====

package roberto;

import java.util.*;
import java.math.*;

//=====

public class Tutoriales implements java.io.Serializable
{
    /** auto_increment */
    public static final String ID           = "id";
    public static final String TITULO       = "titulo";
    public static final String DESCRIPCION  = "descripcion";
    public static final String AUTOR        = "autor";

    //-----

    public static final int TITULO_SIZE     = 255;
    public static final int DESCRIPCION_SIZE = 255;

    //-----

    public int    id;
    public String titulo;
    public String descripcion;
    public int    autor;

    //-----

    public int getId() { return id; }

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

    //-----

    public String getTitulo() { return titulo; }

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



```
//-----
public String getDescripcion() { return descripcion; }

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

//-----

public int getAutor() { return autor; }

public void setAutor(int autor)
{
    this.autor = autor;
}
}
```

Este es el fichero **jdo**:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE jdo SYSTEM "file:/javax/jdo/jdo.dtd">
<jdo>
  <package name="roberto">
    <class name="Tutoriales" identity-type="datastore">
      <field name="id" persistence-modifier="persistent">
        <extension vendor-name="obj" key="column" value="ID"/>
      </field>
      <field name="titulo" persistence-modifier="persistent">
        <extension vendor-name="obj" key="column" value="TITULO"/>
      </field>
      <field name="descripcion" persistence-modifier="persistent">
        <extension vendor-name="obj" key="column" value="DESCRIPCION"/>
      </field>
      <field name="autor" persistence-modifier="persistent">
        <extension vendor-name="obj" key="column" value="AUTOR"/>
      </field>
    </class>
  </package>
</jdo>
```

Compilación

Nos ha creado hasta en **Script ant** para compilar y ampliar la clase (enhance para que cumpla el estandar JDO).

Vemos que falla porque nos faltan algunos ficheros

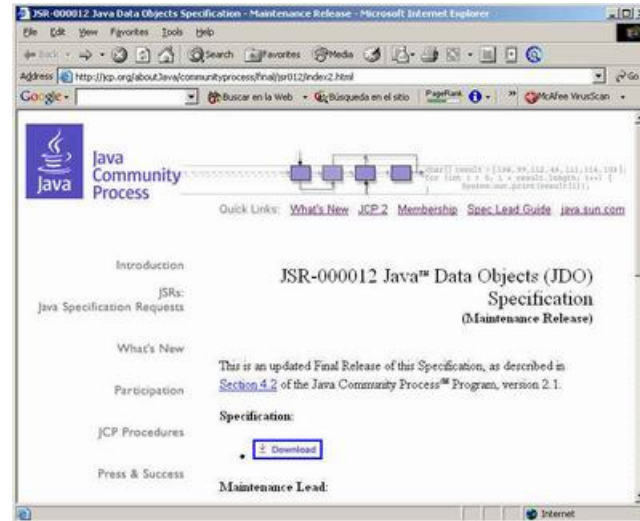
```
C:\WINNT\system32\cmd.exe
make-class:
[mkdir] Created dir: C:\java\druid\src\roberto
[mkdir] Created dir: C:\java\druid\build
[copy] Copying 1 file to C:\java\druid\src\roberto
[javac] Compiling 1 source file to C:\java\druid\build

jdo-enhancer:
[mkdir] Created dir: C:\java\druid\lib
[echo]
[echo]
[echo] *** NOTE: Copy the jdo.jar and the jdori.jar to the lib directory
[echo]
[mkdir] Created dir: C:\java\druid\build\META-INF
[copy] Copying 1 file to C:\java\druid\build\META-INF
[echo] JDO Enhancing: Tutoriales.class
[java] java.lang.NoClassDefFoundError: com/sun/jdori/enhancer/Main
[java] Exception in thread "main"

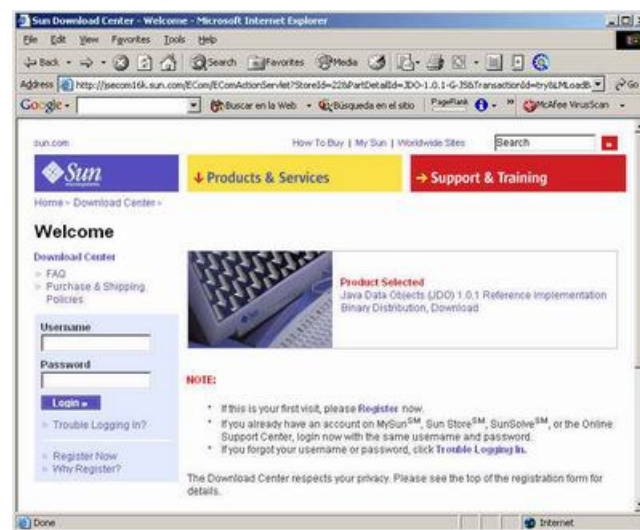
BUILD FAILED
C:\java\druid\build.xml:39: Java returned: 1

Total time: 12 seconds
C:\java\druid>
```

Para que funcione bien podemos descargarnos la implementación de referencia de JDO de SUN.



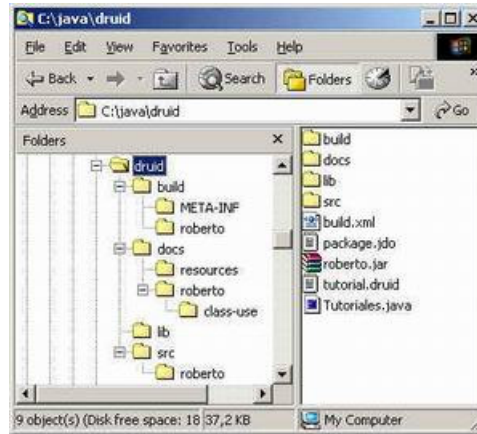
Aunque necesitamos registrarnos para poder descargarla.



Este del el fichero la implementación de referencia.



Si copiamos los ficheros anteriores al directorio lib, compilará correctamente



Y el script ant no dará problemas

```
C:\java\druid>ant
Buildfile: build.xml

make-class:

jdo-enhancer:
    [echo]
    [echo] build
    [echo] *** NOTE: Copy the jdo.jar and the jdori.jar to the lib directory
    [echo]
    [echo] JDO Enhancing: Tutoriales.class
    [java] done.

jar:
    [echo] Creating jar file
    [jar] Building jar: C:\java\druid\roberto.jar

javadocs:
    [echo] Creating javadocs
    [mkdir] Created dir: C:\java\druid\docs
    [javadoc] Generating Javadoc
    [javadoc] Javadoc execution
    [javadoc] Loading source files for package roberto...
    [javadoc] Constructing Javadoc information...
    [javadoc] Standard Doclet version 1.4.2
    [javadoc] Building tree for all the packages and classes...
    [javadoc] Building index for all the packages and classes...
    [javadoc] Building index for all classes...

    [javadoc] Generating C:\java\druid\docs\stylesheet.css...

all:

BUILD SUCCESSFUL
Total time: 16 seconds
C:\java\druid>
```

Comprobación de código generado

Si decompilamos el código, vemos nuestra clase JDO con todos los elementos requeridos

```
// Decompiled by DJ v3.5.5.77 Copyright 2003 Atanas Neshkov Date: 18/02/2004 10:17:58
// Home Page : http://members.fortunecity.com/neshkov/dj.html - Check often for new version!
// Decompiler options: packimports(3)

package roberto;

import java.io.*;
import javax.jdo.PersistenceManager;
import javax.jdo.spi.*;

public class Tutoriales
    implements Serializable, PersistenceCapable
{

    public Tutoriales()
    {
    }

    public int getId()
    {
        return jdoGetid(this);
    }

    public void setId(int i)
    {
    }
}
```

```
        jdoSetid(this, i);
    }

    public String getTitulo()
    {
        return jdoGettitulo(this);
    }

    public void setTitulo(String s)
    {
        jdoSettitulo(this, s);
    }

    public String getDescripcion()
    {
        return jdoGetdescripcion(this);
    }

    public void setDescripcion(String s)
    {
        jdoSetdescripcion(this, s);
    }

    public int getAutor()
    {
        return jdoGetautor(this);
    }

    public void setAutor(int i)
    {
        jdoSetautor(this, i);
    }

    public final void jdoReplaceFlags()
    {
        StateManager statemanager = jdoStateManager;
        if(statemanager != null)
            jdoFlags = statemanager.replacingFlags(this);
    }

    public final boolean jdoIsPersistent()
    {
        StateManager statemanager = jdoStateManager;
        if(statemanager != null)
            return statemanager.isPersistent(this);
        else
            return false;
    }

    public final boolean jdoIsTransactional()
    {
        StateManager statemanager = jdoStateManager;
        if(statemanager != null)
            return statemanager.isTransactional(this);
        else
            return false;
    }

    public final boolean jdoIsNew()
    {
        StateManager statemanager = jdoStateManager;
        if(statemanager != null)
            return statemanager.isNew(this);
        else
            return false;
    }

    public final boolean jdoIsDeleted()
    {
        StateManager statemanager = jdoStateManager;
        if(statemanager != null)
            return statemanager.isDeleted(this);
        else
            return false;
    }

    public final boolean jdoIsDirty()
    {
        StateManager statemanager = jdoStateManager;
        if(statemanager != null)
            return statemanager.isDirty(this);
        else
            return false;
    }

    public final void jdoMakeDirty(String s)
    {
        StateManager statemanager = jdoStateManager;
        if(statemanager != null)
            statemanager.makeDirty(this, s);
    }

    protected final void jdoPreSerialize()
    {
        StateManager statemanager = jdoStateManager;
        if(statemanager != null)
```

```

        statemanager.preSerialize(this);
    }

    public final PersistenceManager jdoGetPersistenceManager()
    {
        StateManager statemanager = jdoStateManager;
        if (statemanager != null)
            return statemanager.getPersistenceManager(this);
        else
            return null;
    }

    public final Object jdoGetObjectId()
    {
        StateManager statemanager = jdoStateManager;
        if (statemanager != null)
            return statemanager.getObjectId(this);
        else
            return null;
    }

    public final Object jdoGetTransactionalObjectId()
    {
        StateManager statemanager = jdoStateManager;
        if (statemanager != null)
            return statemanager.getTransactionalObjectId(this);
        else
            return null;
    }

    public final synchronized void jdoReplaceStateManager(StateManager statemanager)
    {
        StateManager statemanager1 = jdoStateManager;
        if (statemanager1 != null)
        {
            jdoStateManager = statemanager1.replacingStateManager(this, statemanager);
            return;
        } else
        {
            JDOImplHelper.checkAuthorizedStateManager(statemanager);
            jdoStateManager = statemanager;
            jdoFlags = 1;
            return;
        }
    }

    public final void jdoProvideFields(int ai[])
    {
        if (ai == null)
            throw new IllegalArgumentException("arg1");
        int i = ai.length;
        for (int j = 0; j < i; j++)
            jdoProvideField(ai[j]);
    }

    public final void jdoReplaceFields(int ai[])
    {
        if (ai == null)
            throw new IllegalArgumentException("arg1");
        int i = ai.length;
        for (int j = 0; j < i; j++)
            jdoReplaceField(ai[j]);
    }

    protected static final Class sunjdo$classForName$(String s)
    {
        try
        {
            return Class.forName(s);
        }
        catch (ClassNotFoundException classnotfoundexception)
        {
            throw new NoClassDefFoundError(classnotfoundexception.getMessage());
        }
    }

    protected static int jdoGetManagedFieldCount()
    {
        return jdoInheritedFieldCount + 4;
    }

    public PersistenceCapable jdoNewInstance(StateManager statemanager)
    {
        Tutoriales tutoriales = new Tutoriales();
        tutoriales.jdoFlags = 1;
        tutoriales.jdoStateManager = statemanager;
        return tutoriales;
    }

    public PersistenceCapable jdoNewInstance(StateManager statemanager, Object obj)
    {
        Tutoriales tutoriales = new Tutoriales();
        tutoriales.jdoCopyKeyFieldsFromObjectId(obj);
        tutoriales.jdoFlags = 1;
    }

```

```

        tutoriales.jdoStateManager = statemanager;
        return tutoriales;
    }

    public void jdoProvideField(int i)
    {
        StateManager statemanager = jdoStateManager;
        switch(i - jdoInheritedFieldCount)
        {
            case 0: // '\0'
                if(statemanager == null)
                {
                    throw new IllegalStateException("arg0.jdoStateManager");
                } else
                {
                    statemanager.providedIntField(this, i, autor);
                    return;
                }

            case 1: // '\001'
                if(statemanager == null)
                {
                    throw new IllegalStateException("arg0.jdoStateManager");
                } else
                {
                    statemanager.providedStringField(this, i, descripcion);
                    return;
                }

            case 2: // '\002'
                if(statemanager == null)
                {
                    throw new IllegalStateException("arg0.jdoStateManager");
                } else
                {
                    statemanager.providedIntField(this, i, id);
                    return;
                }

            case 3: // '\003'
                if(statemanager == null)
                {
                    throw new IllegalStateException("arg0.jdoStateManager");
                } else
                {
                    statemanager.providedStringField(this, i, titulo);
                    return;
                }
        }
        throw new IllegalArgumentException("arg1");
    }

    public void jdoReplaceField(int i)
    {
        StateManager statemanager = jdoStateManager;
        switch(i - jdoInheritedFieldCount)
        {
            case 0: // '\0'
                if(statemanager == null)
                {
                    throw new IllegalStateException("arg0.jdoStateManager");
                } else
                {
                    autor = statemanager.replacingIntField(this, i);
                    return;
                }

            case 1: // '\001'
                if(statemanager == null)
                {
                    throw new IllegalStateException("arg0.jdoStateManager");
                } else
                {
                    descripcion = statemanager.replacingStringField(this, i);
                    return;
                }

            case 2: // '\002'
                if(statemanager == null)
                {
                    throw new IllegalStateException("arg0.jdoStateManager");
                } else
                {
                    id = statemanager.replacingIntField(this, i);
                    return;
                }

            case 3: // '\003'
                if(statemanager == null)
                {
                    throw new IllegalStateException("arg0.jdoStateManager");
                } else
                {
                    titulo = statemanager.replacingStringField(this, i);
                    return;
                }
        }
    }

```

```

    }
    throw new IllegalArgumentException("arg1");
}

protected final void jdoCopyField(Tutoriales tutoriales, int i)
{
    switch(i - jdoInheritedFieldCount)
    {
        case 0: // '\0'
            if(tutoriales == null)
            {
                throw new IllegalArgumentException("arg1");
            } else
            {
                autor = tutoriales.autor;
                return;
            }

        case 1: // '\001'
            if(tutoriales == null)
            {
                throw new IllegalArgumentException("arg1");
            } else
            {
                descripcion = tutoriales.descripcion;
                return;
            }

        case 2: // '\002'
            if(tutoriales == null)
            {
                throw new IllegalArgumentException("arg1");
            } else
            {
                id = tutoriales.id;
                return;
            }

        case 3: // '\003'
            if(tutoriales == null)
            {
                throw new IllegalArgumentException("arg1");
            } else
            {
                titulo = tutoriales.titulo;
                return;
            }
    }
    throw new IllegalArgumentException("arg2");
}

public void jdoCopyFields(Object obj, int ai[])
{
    if(jdoStateManager == null)
        throw new IllegalStateException("arg0.jdoStateManager");
    if(!(obj instanceof Tutoriales))
        throw new IllegalArgumentException("arg1");
    if(ai == null)
        throw new IllegalArgumentException("arg2");
    Tutoriales tutoriales = (Tutoriales)obj;
    if(tutoriales.jdoStateManager != jdoStateManager)
        throw new IllegalArgumentException("arg1.jdoStateManager");
    int i = ai.length;
    for(int j = 0; j < i; j++)
        jdoCopyField(tutoriales, ai[j]);
}

public Object jdoNewObjectIdInstance()
{
    return null;
}

public Object jdoNewObjectIdInstance(String s)
{
    return null;
}

public void jdoCopyKeyFieldsToObjectId(Object obj)
{
}

protected void jdoCopyKeyFieldsFromObjectId(Object obj)
{
}

public void jdoCopyKeyFieldsToObjectId(javax.jdo.spi.PersistenceCapable.ObjectIdFieldSupplier objectidfieldsupplier, Object obj)
{
}

public void jdoCopyKeyFieldsFromObjectId(javax.jdo.spi.PersistenceCapable.ObjectIdFieldConsumer objectidfieldconsumer, Object obj)
{
}

public static final int jdoGetautor(Tutoriales tutoriales)
{
}

```



```

        if(tutoriales.jdoFlags <= 0)
            return tutoriales.autor;
        StateManager statemanager = tutoriales.jdoStateManager;
        if(statemanager == null)
            return tutoriales.autor;
        if(statemanager.isLoaded(tutoriales, jdoInheritedFieldCount + 0))
            return tutoriales.autor;
        else
            return statemanager.getIntField(tutoriales, jdoInheritedFieldCount + 0, tutoriales.autor);
    }

    public static final void jdoSetautor(Tutoriales tutoriales, int i)
    {
        if(tutoriales.jdoFlags == 0)
        {
            tutoriales.autor = i;
            return;
        }
        StateManager statemanager = tutoriales.jdoStateManager;
        if(statemanager == null)
        {
            tutoriales.autor = i;
            return;
        }
        else
        {
            statemanager.setIntField(tutoriales, jdoInheritedFieldCount + 0, tutoriales.autor, i);
            return;
        }
    }

    public static final String jdoGetdescripcion(Tutoriales tutoriales)
    {
        if(tutoriales.jdoFlags <= 0)
            return tutoriales.descripcion;
        StateManager statemanager = tutoriales.jdoStateManager;
        if(statemanager == null)
            return tutoriales.descripcion;
        if(statemanager.isLoaded(tutoriales, jdoInheritedFieldCount + 1))
            return tutoriales.descripcion;
        else
            return statemanager.getStringField(tutoriales, jdoInheritedFieldCount + 1, tutoriales.descripcion);
    }

    public static final void jdoSetdescripcion(Tutoriales tutoriales, String s)
    {
        if(tutoriales.jdoFlags == 0)
        {
            tutoriales.descripcion = s;
            return;
        }
        StateManager statemanager = tutoriales.jdoStateManager;
        if(statemanager == null)
        {
            tutoriales.descripcion = s;
            return;
        }
        else
        {
            statemanager.setStringField(tutoriales, jdoInheritedFieldCount + 1, tutoriales.descripcion, s);
            return;
        }
    }

    public static final int jdoGetid(Tutoriales tutoriales)
    {
        if(tutoriales.jdoFlags <= 0)
            return tutoriales.id;
        StateManager statemanager = tutoriales.jdoStateManager;
        if(statemanager == null)
            return tutoriales.id;
        if(statemanager.isLoaded(tutoriales, jdoInheritedFieldCount + 2))
            return tutoriales.id;
        else
            return statemanager.getIntField(tutoriales, jdoInheritedFieldCount + 2, tutoriales.id);
    }

    public static final void jdoSetid(Tutoriales tutoriales, int i)
    {
        if(tutoriales.jdoFlags == 0)
        {
            tutoriales.id = i;
            return;
        }
        StateManager statemanager = tutoriales.jdoStateManager;
        if(statemanager == null)
        {
            tutoriales.id = i;
            return;
        }
        else
        {
            statemanager.setIntField(tutoriales, jdoInheritedFieldCount + 2, tutoriales.id, i);
            return;
        }
    }

    public static final String jdoGettitulo(Tutoriales tutoriales)
    {

```

```

        if(tutoriales.jdoFlags <= 0)
            return tutoriales.titulo;
        StateManager statemanager = tutoriales.jdoStateManager;
        if(statemanager == null)
            return tutoriales.titulo;
        if(statemanager.isLoaded(tutoriales, jdoInheritedFieldCount + 3))
            return tutoriales.titulo;
        else
            return statemanager.getStringField(tutoriales, jdoInheritedFieldCount + 3, tutoriales.titulo);
    }

    public static final void jdoSettitulo(Tutoriales tutoriales, String s)
    {
        if(tutoriales.jdoFlags == 0)
        {
            tutoriales.titulo = s;
            return;
        }
        StateManager statemanager = tutoriales.jdoStateManager;
        if(statemanager == null)
        {
            tutoriales.titulo = s;
            return;
        }
        else
        {
            statemanager.setStringField(tutoriales, jdoInheritedFieldCount + 3, tutoriales.titulo, s);
            return;
        }
    }

    private void writeObject(ObjectOutputStream objectoutputstream)
        throws IOException
    {
        jdoPreSerialize();
        objectoutputstream.defaultWriteObject();
    }

    public static final String ID = "id";
    public static final String TITULO = "titulo";
    public static final String DESCRIPCION = "descripcion";
    public static final String AUTOR = "autor";
    public static final int TITULO_SIZE = 255;
    public static final int DESCRIPCION_SIZE = 255;
    public int id;
    public String titulo;
    public String descripcion;
    public int autor;
    protected transient StateManager jdoStateManager; /* synthetic field */
    protected transient byte jdoFlags; /* synthetic field */
    private static final int jdoInheritedFieldCount = 0; /* synthetic field */
    private static final String jdoFieldNames[] = {
        "autor", "descripcion", "id", "titulo"
    }; /* synthetic field */
    private static final Class jdoFieldTypes[]; /* synthetic field */
    private static final byte jdoFieldFlags[] = {
        21, 21, 21, 21
    }; /* synthetic field */
    private static final Class jdoPersistenceCapableSuperclass; /* synthetic field */

    static
    {
        jdoFieldTypes = (new Class[] {
            Integer.TYPE, sunjdo$classForName$("java.lang.String"), Integer.TYPE, sunjdo$classForName$("java.lang.String")
        });
        jdoPersistenceCapableSuperclass = null;
        JDOImplHelper.registerClass(sunjdo$classForName$("roberto.Tutoriales"), jdoFieldNames,
jdoFieldTypes, jdoFieldFlags, jdoPersistenceCapableSuperclass, new Tutoriales());
    }
}

```

Conclusión

Estas herramientas nos pueden ayudar a realizar el trabajo cuando no tenemos demasiada experiencia.

Para mi gusto, la usabilidad de ésta en concreto hay que mejorarla un poco pero ... no hay que quejarse y siempre hay que dar gracias a quien trabaja gratis para nosotros

[Sobre el Autor ..](#)

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

Creatividad Internet

[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	<input type="text"/>
	<input type="button" value="Enviar"/>

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

Nombre Corto	Descripción
Creación automática de recursos Hibernate con Middlegen	En este tutorial aprenderéis como utilizar la herramienta middlegen para generar distintas capas de persistencia (CMP 2.0, JDO, Hibernate, Torque), a partir de un modelo físico de datos, de un modo automático, mediante el uso de la herramienta middlegen
Generación automática de código JDBC	En este tutorial os enseñamos como, sin conocimiento de JDBC, crear vuestro programas en Java, gracias a JDBCTest.
CMP Entity Beans y MySql	Os mostramos como crear un Entity Bean con persistencia controlada por el servidor, configurado para usar MySql
JDBC y 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

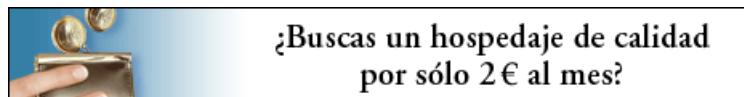
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

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