

<b>Entrega:</b>	<b>1</b>
<b>Revisión:</b>	<b>4</b>
<b>Fecha:</b>	<b>12 / Abril / 2007</b>
<b>Nivel:</b>	<b>Principiante</b>

# Configurar OpenXava 2.1 para trabajar con Postgres.

## Índice de contenido

0: Prólogo.....	1
1.- Nociones sobre lo que vamos a hacer.....	1
2.- Comprobar el funcionamiento de la BD Postgres.....	2
3.- Realizar una copia de seguridad de OpenXava.....	4
4.- Modificar el fichero “build.xml” del proyecto OpenXavaPlantilla.....	4
5.- Crear el fichero de configuración “tomcat-postgres.properties” de la carpeta OpenXavaPlantilla.....	5
6.- Modificar el fichero “hibernate/hibernate-cfg.xml” de la carpeta OpenXavaPlantilla .....	5
7.- Copiar los drives JDBC (jar) de Postgres.....	6
8.- Generamos nuestro “primer proyecto” según la guía de referencia.....	7
9.- Conclusión.....	9

## 0: Prólogo

Espero que esta pequeña guía sirva para los pobres principiantes y carrozas que nos atrevemos a meternos en proyectos que tenemos la intuición que son buenos, pero que nos suenan a “chino”. Está pensada para todos los que venimos de entornos *RAD* (desarrollo rápido de aplicaciones) y que nos cuesta cambiar el “chip”.

Por desgracia, esta introducción está realizada en entorno Windows XP, pido disculpas a los “valientes” que estáis en Linux.

Estamos ante una era de cambios constantes y profundos, y los proyectos de código abierto son un tren donde hay una infinidad de locomotoras que tiramos de él, y somos nosotros los que decidimos cuales van a ser sus derroteros. En el foro hay algunas cuestiones sobre la configuración del *OpenXava* para otras bases de datos que no sean *HSQLDB*. Esta base de datos hasta ahora está sirviendo para motivos académicos, pero no me extrañaría que en pocos años la tengamos como referencia.

En el foro, surgen algunas preguntas sobre la configuración de OpenXava sobre Postgres. Espero que esta pequeña guía sirva para ayudaros un poco.

## 1.- Nociones sobre lo que vamos a hacer

Debemos tener una mínima idea de Postgres. En concreto, debemos tener instalada y configurada dicha base de datos; tiene que estar accesible y debemos saber como conectar con ella (debemos saber su URL, usuario y contraseña). Debemos tener en cuenta que Postgres diferencia entre mayúsculas y minúscula.

Tenemos que decidir si vamos a trabajar en Postgres de forma esporádica o por el contrario vamos a trabajar siempre.

La carpeta *OpenXavaPlantilla*, es como su nombre indica una plantilla, por tanto no es muy conveniente que toquemos cosas de dicha plantilla sin saberlo. Por tanto, si vamos a trabajar “para siempre”, con la BD Postgres, podemos hacer los siguientes cambios en dicho proyecto:

- Crear un fichero de configuración nuevo *tomcat-postgres.properties*
- Modificar el fichero *hibernate/hibernate-cfg.xml* para que apunte a nuestra BD Postgres
- Modificar el fichero *build.xml* para que apunte a nuestro fichero de propiedades.

Por tanto, cada vez que generemos un proyecto nuevo (Paso 8 i punto 2,1 de la guía de referencia), ya habrá:

- creado el *build.xml* de nuestro proyecto igual que el de la plantilla (por tanto no hay que volver a hacer los cambios anteriores)
- tendremos el fichero *hibernate/hibernate-cfg.xml* actualizado a nuestra configuración.
- creado el fichero *tomcat-postgres.properties* en nuestro proyecto (por tanto no hay que crearlo)

Si no se hubiera planteado hacer estos cambios en la plantilla (*OpenXavaPlantilla*), cada vez que generásemos un proyecto, tendríamos que hacer estos cambios a mano. Esta opción es interesante si no vamos a trabajar mucho con Postgres, o si vamos a realizar otros proyectos con acceso a otras bases de datos.

También habrá que decirle al sistema donde están las librerías de acceso a la base de datos JDBC, o sea “los .jar”. Una buena opción seria si vamos a trabajar mucho con Postgres el crear una carpeta *lib* en la capeta *OpenXavaPlantilla* y meter los *jars* ahí, para que luego los copie. Si no, podemos crear la carpeta *lib* con los *jars* en nuestro proyecto

Bien vamos a realizar todos los pasos para que nuestro *OpenXava* pueda trabajar en Postgres como base de datos predeterminada cuando generemos los proyectos.

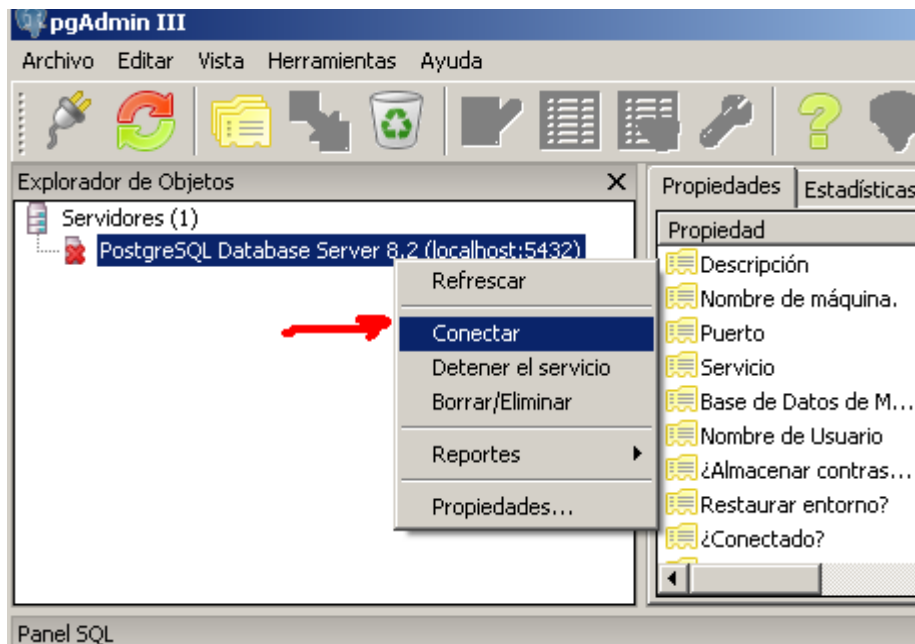
## 2.- Comprobar el funcionamiento de la BD Postgres.

Hemos de comprobar que el gestor de bases de datos *Postgres* esté en marcha y que sea accesible desde nuestra máquina. La mejor forma es entrar en el *pgAdmin*. Si no se tiene, es muy conveniente tenerlo instalado (se obtiene de la Web de *Postgres*)

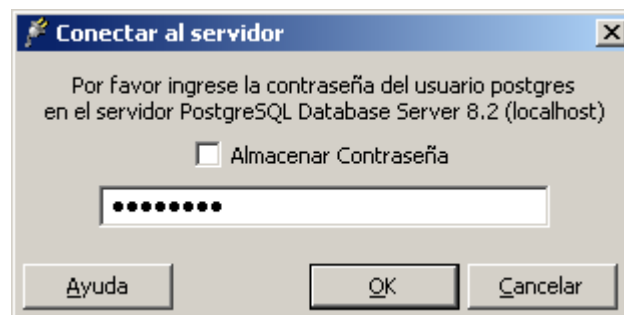


Una vez abierto, se debe intentar conectar (si no nos deja entrar, ya no podemos seguir adelante).

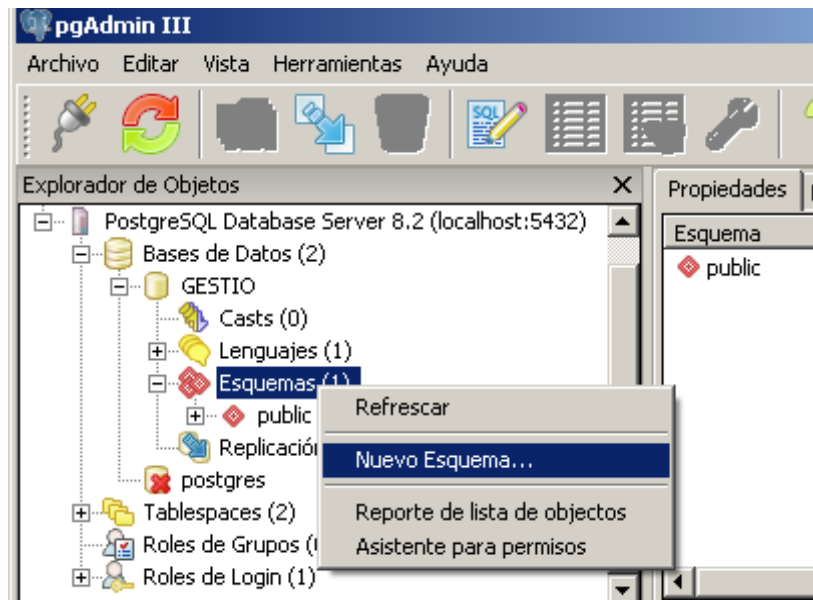
Para ello, con el botón derecho del ratón sobre el servidor seleccionamos conectar.



A continuación le damos la contraseña que hayamos dado para el usuario “*postgres*”.



Una vez dentro, podemos crear un nuevo esquema, dándole al botón derecho en la carpeta esquemas:



Tengo previamente creada una base de datos llamada “*GESTIO*” que voy a aprovechar y sobre la cual creo un esquema que le llamo por ejemplo “*esq\_01*”. Por tanto ahora tendré los esquemas “*public*” y “*esq\_01*”.

#### Observaciones:

**NUNCA** debéis poner en la base de datos Postgres los nombres de esquemas o tablas en **MAYÚSCULAS**. Siempre en **MINÚSCULAS**! ya que postgres sobre hibernate mapea los nombres de tablas y esquemas en minúsculas.

Por tanto hemos salvado este primer requisito.

### 3.- Realizar una copia de seguridad de OpenXava

Vamos a hacer cambios profundos en la configuración de *OpenXava*, y nunca debemos olvidar que algunos somos “novatos”. Cualquier tiempo invertido en hacer copias de seguridad nunca es tiempo perdido!. Por tanto debemos copiar en principio toda la carpeta ***workSpace*** que cuelga del directorio de instalación de *Openxava*.

También tenéis que demostrar que habéis podido llegar hasta el punto 2.6 de la guía de referencia de OpenXava (y habéis podido ejecutar la primera aplicación).

Por tanto sabemos que:

- Tenéis *Postgres* trabajando correctamente.
- Sabéis construir vuestro “*primer proyecto OpenXava*”.
- Habéis hecho una copia de seguridad de vuestro trabajo.

### 4.- Modificar el fichero “build.xml” del proyecto OpenXavaPlantilla

Cambiamos la variable que apunta al fichero de configuración del build.xml que hay en la carpeta

OpenXavaPlantilla. Buscamos una línea que tenga *tomcat-hypersonic* y la cambiamos por otra que hace referencia a un nuevo fichero de configuración *tomcat-postgres* que crearemos en el siguiente apartado.

```
<!--  
<property name="configuration" value="tomcat-hypersonic"/>  
-->  
<property name="configuration" value="tomcat-postgres"/>
```

Guardamos los cambios.

Por tanto un resumen de lo que se tiene hasta ahora:

- Tenéis *Postgres* trabajando correctamente.
- Sabéis construir vuestro “*primer proyecto OpenXava*”.
- Habéis hecho una copia de seguridad de vuestro trabajo.
- Habéis cambiado “HSQL” por “Postgres” en el *build.xml* del proyecto *workspace/Openxava*

## 5.- Crear el fichero de configuración “tomcat-postgres.properties” de la carpeta OpenXavaPlantilla

Copiamos el contenido del fichero *tomcat-hypersonic.properties* en un nuevo fichero *tomcat-postgres.properties*. A continuación cambiamos todo aquello que nos interesa. En este ejemplo modificaremos el **separador** que lo cambiaremos de “\_” a “.”. **OJO: No hay que poner las comillas (“”), debe quedar así:**

También se modificará el dialecto a **PostgreSQLDialect**.

```
separator=.  
organization_es=Gestión cuatrocientos, s.l.  
organization_ca=Gestió quatre-cents, s.l.  
organization_en=Management four hundred, limited  
hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect  
datasource.prefix=java comp/env/jdbc
```

## 6.- Modificar el fichero “hibernate/hibernate-cfg.xml” de la carpeta OpenXavaPlantilla

Esta es la segunda parte siniestra de nuestra introducción. Tenemos que comentar todo lo relativo a HSQLDB (que estaba etiquetado como Hypersonic) y crear nuevas definiciones para Postgres (que se ha etiquetado como Postgres), quedando:

```
<session-factory>  
<!-- Hypersonic  
<property name="hibernate.connection.driver_class">org.hsqldb.jdbcDriver</property>  
<property name="hibernate.dialect">org.hibernate.dialect.HSQLDialect</property>  
<property name="hibernate.connection.url">jdbc:hsqldb:hsq!://localhost:1666</property>  
-->  
  
<!-- Postgres-->
```

```
<property name="hibernate.connection.driver_class">org.postgresql.Driver</property>
<property name="hibernate.dialect">org.hibernate.dialect.PostgreSQLDialect</property>
<property name="hibernate.connection.url">jdbc:postgresql://localhost:5432/GESTIO</property>
```

Donde el servidor nuestro es `//localhost` (pero vosotros tenéis que indicar el nombre de la máquina donde reside el servidor Postgres o la dirección), `5432` es el puerto que normalmente está asociado el servicio de BD Postgres y `GESTIO` es el nombre de nuestra base de datos (pero vosotros tenéis que indicar la vuestra).

Además hay que cambiar todo lo relativo a usuarios y contraseñas. Tenemos que poner comentarios al famoso usuario “SA” sin contraseña, y meter el usuario “*postgres*” (o el que sea) con su contraseña “*mi\_password\_de\_postgres*” (OJO: la que tu hayas definido o conozcas!)

Además debemos meter entre comentarios el acceso a Hypersonic que es lo que había antes e introducir el nuevo Acceso Postgres lo que hemos añadido después.

```
<!-- Acceso Hypersonic
<property name="hibernate.connection.username">sa</property>
<property name="hibernate.connection.password"></property>
<property name="hibernate.show_sql">>false</property>
-->

<!-- Acceso Postgres -->
<property name="hibernate.connection.username">postgres</property>
<property name="hibernate.connection.password">mi_password_de_postgres</property>
<property name="hibernate.show_sql">>false</property>
```

Y ya hemos conseguido terminar la parte tenebrosa de nuestro proceso!

## 7.- Copiar los drivers JDBC (jar) de Postgres.

Hay que crear una carpeta `lib` dentro de la carpeta `OpenXavaPlantilla`.

En nuestro caso, los jdbc se llaman `postgresql-8.2-504.jdbc3.jar`. Y los copiamos a las carpetas:

- `portal/common/lib`
- `OpenXavaPlantilla/lib`

Por tanto un resumen de lo que se tiene hasta ahora:

- Tenéis *Postgres* trabajando correctamente.
- Sabéis construir vuestro “*primer proyecto OpenXava*”.
- Habéis hecho una copia de seguridad de vuestro trabajo.
- Habéis cambiado “HSQL” por “Postgres” en el *build.xml* del proyecto `workspace/Openxava`
- Se ha creado el fichero de configuración *tomcat\_postgres.properties* y cambiado el separador “*un guión subrayado*” por un “*punto*”.
- Habéis cambiado las definiciones de HSQLDB por las de Postgres en el fichero *hibernate/hibernate-cfg.xml* del proyecto *OpenXavaPlantilla*.
- Se han copiado los ficheros “*.jar*” del JDBC de Postgres en sus respectivas carpetas.

## 8.- Generamos nuestro “primer proyecto” según la guía de referencia.

Hay que conseguir la guía de referencia de la web de OpenXava, donde se explica muy claramente y con todo tipo de detalles como crear nuestro primer proyecto.

Seguimos como buenos alumnos el capítulo 2 de la guía de referencia. Hacemos **la parte 2.1** y creamos el proyecto (editando el archivo *CrearNuevoProyecto.xml* y posteriormente ejecutándolo como *Ant Build*).

En el punto 2.2 de la guía, vamos a configurar el acceso a la base de datos. Para ello, modificamos el fichero `context.xml` del directorio del Tomcat. Y le introducimos la siguiente información:

```
< Resource name= "jdbc/MiGestionDS" auth= "Container" type= "javax.sql.DataSource"
    maxActive= "20" maxIdle= "5" maxWait= "10000"
    username= "postgres" password= "mi_password_de_postgres"
    driverClassName= "org.postgresql.Driver"
    url= "jdbc:postgresql://localhost:5432/GESTIO"/>
```

Donde:

- “mi\_password\_de\_postgres” es la contraseña que tengamos de acceso a Postgres.
- `url= "jdbc:postgresql://localhost:5432/GESTIO"` es donde definimos el acceso a la base de datos, siendo “//localhost” la dirección IP o nombre del servidor donde resida el gestor de base de datos y GESTIO es el nombre de nuestra base de datos que hayamos creado en Postgres.

### Observaciones:

Hay que distinguir entre aquellos que trabajen arrancando el Tomcat desde fuera del Eclipse (ejecutando start-portal desde el directorio de instalación del OpenXava) y los que arranquen el Tomcat desde el Eclipse tal como se explica en <http://www.gestion400.com/web/guest/wtp>

Para los primeros el fichero `context.xml` está en “portal/conf”.

Para los segundos en “workspace\Servers\Tomcat v5.5 Server @ localhost-config”

Si el nivel os lo permite, se aconseja trabajar dentro de Eclipse, ya que se consigue una mayor velocidad y menor cambio de pantallas. Cuanto más nos centremos en lo que hacemos, y menos cambios de pantallas tengamos, reduciremos las probabilidades de tener cometer errores!

Ahora realizamos **la parte 2.3 de la guía** y modificamos el componente *Almacen.xml* y lo dejamos así:

```
<!DOCTYPE componente SYSTEM "dtds/componente.dtd">

< componente nombre= "Almacen">

    < entidad>
        < propiedad nombre= "codigo" tipo= "String" clave= "true"
            longitud= "5" requerido= "true"/>
        < propiedad nombre= "nombre" tipo= "String"
            longitud= "40" requerido= "true"/>
    </entidad>
```

```

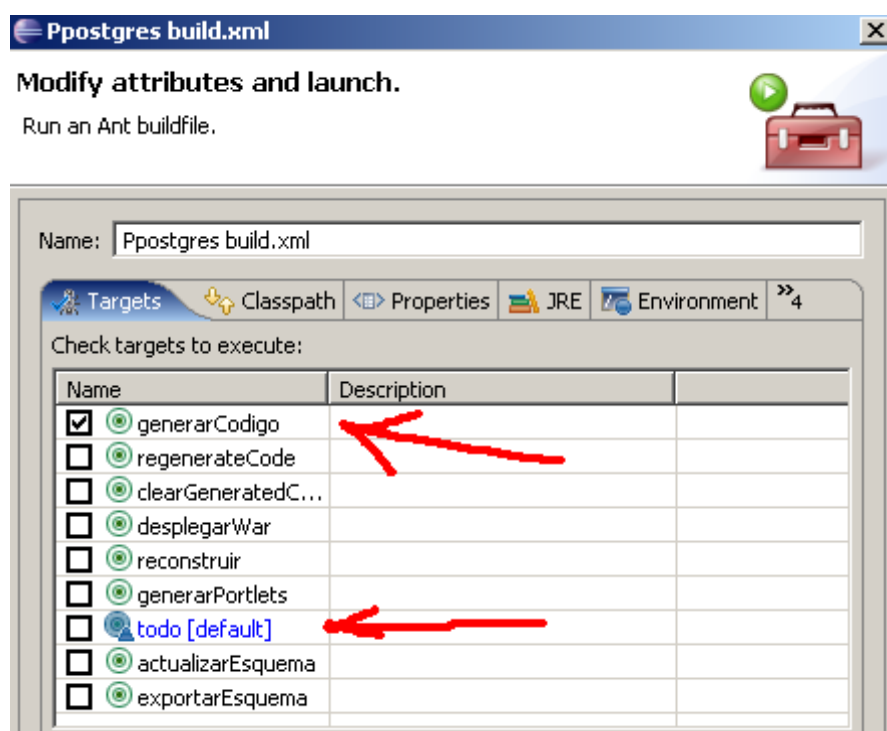
< mapeo-entidad tabla= "ESQ_01 @separator@ALMACENES">
    < mapeo-propiedad propiedad-modelo= "codigo" columna= "CODIGO"/>
    < mapeo-propiedad propiedad-modelo= "nombre" columna= "NOMBRE"/>
</ mapeo-entidad>

</ componente>


```

Donde “ESQ\_01” es el esquema que hemos creado en la base de datos Postgres, y “ALMACENES” la tabla a crear. **OJO: Cuando hibernate haga el mapeo a Postgres los pasará a MINUSCULAS!**

Le damos a las tareas ant “*generarCodigo*” (Nos situamos en eclipse sobre el fichero *build.xml* de nuestro proyecto, con el botón derecho del ratón seleccionamos “Run As” - “3 Ant build ..” y deseccionamos “**todo**” y seleccionamos “generarCodigo”.



Mas tarde nos colocamos sobre la carpeta del proyecto creado en Eclipse y le damos a la **tecla F5** para que refresque los ficheros generados.

No estaría mal darle al icono  para que compile nuestro proyecto cuando generamos o regeneramos código.

No estaría mal comprobar que los cambios que hemos hecho en los ficheros de OpenXavaPlantilla, se han incorporado en nuestro proyecto, es decir:

- build.xml
- hibernate/hibernate-cfg.xml
- tomcat\_postgres.properties
- carpeta lib con el jar de Postgres.(librerías JDBC)

Ya que estos cambios en nuestro proyecto son los que hacen funcionar sobre Postgres. Tal como ya se ha dicho, si se hubieran realizado estos cambios directamente sobre nuestro proyecto, no hubiera



hecho falta tocar la carpeta OpenXavaPlantilla, pero nos interesa que Postgres sea nuestra BD.

Ahora tal como dice la guía de referencia en el **punto 2.5**:

- editamos el fichero “Gestion/build.xml” y buscamos la tarea ant *actualizarEsquema*.
- Ponemos el valor correcto para *schema.path* “`"/lib/postgresql-8.2-504.jdbc3.jar"`
- Ejecutamos la tarea ant *actualizarEsquema* tal como hemos hecho en el apartado anterior
- Observamos que está creando nuestra tabla ALMACENES
- Arrancamos el Tomcat ( y para los que trabajen con el Tomcat desde Eclipse (en la pestaña de servidores), acordaros de añadir vuestro proyecto al Tomcat ) y ya está todo listo.
- Abrimos el navegador de internet y ejecutamos nuestra aplicación tal como dice la guía de referencia en el **punto 2.6**.

Parece ser que ya funciona todo a la perfección!

## 9.- Conclusión

Con esta pequeña guía hemos configurado este entorno vivo de desarrollo de aplicaciones llamado OpenXava.

Espero que hayáis tenido éxito en la configuración de OpenXava sobre Postgres y que nos veamos en próximas guías que quiero seguir haciendo a medida que aprenda más.