

# Creación de un JTree con datos de una base de datos HSQL con JPA

**Autor:** Jorge A. López Vargas (jorgaf@gmail.com)(cafelojano@wordpress.com)

Éste documento pretende ser una pequeña guía de como construir un árbol (JTree) que presente datos obtenidos desde una base de datos, en éste caso HSQL, utilizando como mecanismo de persistencia JPA. De ninguna manera se constituye en un tutor sobre Swing y mucho menos de JPA.

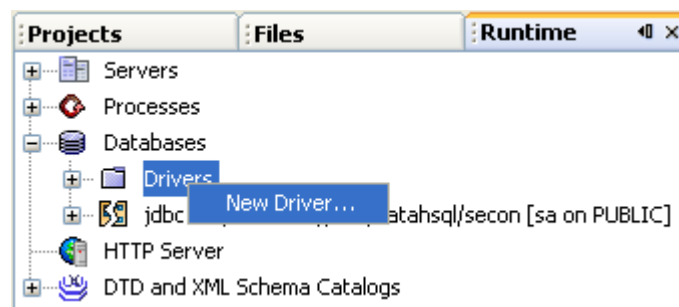
## Herramientas:

Para la construcción de ésta aplicación se emplearon las siguientes herramientas:

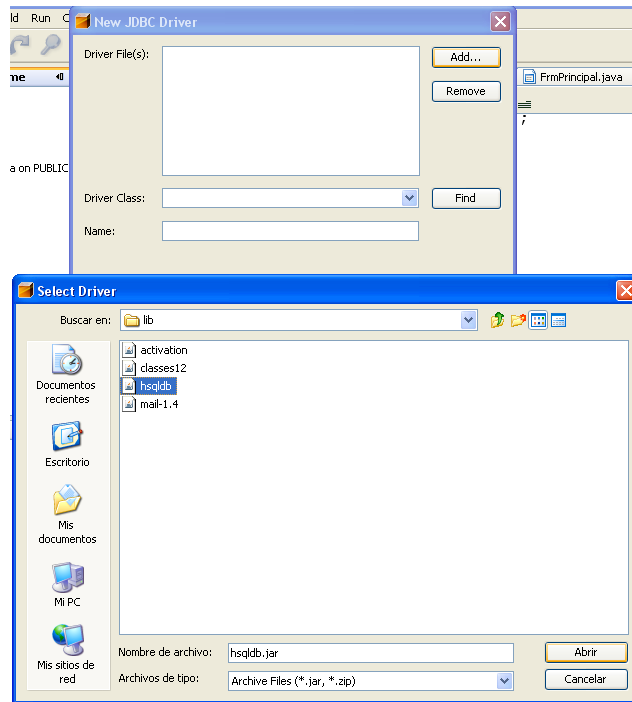
- Java Developer Kit 6.
- NetBeans 5.5
- HSQL 1.8.0.7

## Configuración de NetBeans

La primera tarea que se debe realizar es la configuración de NetBeans, concretamente agregar el driver para las conexiones con HSQL, el mismo que se incluye dentro del archivo .zip que obtiene cuando descarga HSQL. El driver es el archivo hsqldb.jar, debe copiarlo en algún directorio de su disco duro. Para agregar el driver debe ir a la pestaña Runtime, bajo la opción Databases encontrar el nodo Drivers clic derecho y seleccionar New Driver...



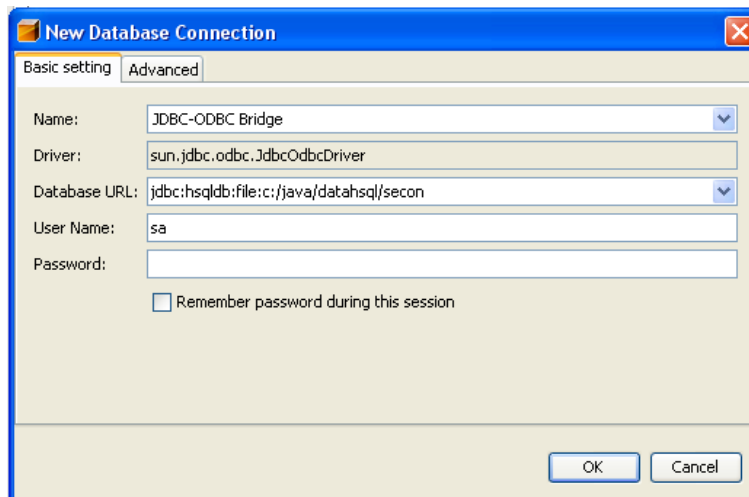
Se presenta una pantalla en donde debemos buscar el driver, haciendo clic en Add... y buscar el archivo .jar en el directorio donde lo copio anteriormente.



El resto de campos se dejan sin alterar y se presiona Ok, con ésto hemos configurado un nuevo driver para crear una conexión con HSQL.

### **Creación de una nueva Conexión**

Dentro de la pestaña de Runtime de NetBeans, bajo el nodo DataBases clic derecho y seleccionar New Connection... Seleccionar del combo Name HSQLDB, en Database URL debe ingresar lo siguiente: "jdbc:hsqldb:file:c:/java/datahsql/secon". Lo que implica que nuestra base de datos se encuentra en la ruta que se señala luego de file. Dentro de User Name colocar "sa" y Password dejarlo en blanco.



### **Creación de la base de datos**

Crear una ventana de comandos SQL para la conexión recién creada, haciendo clic derecho sobre la conexión y seleccionando Execute command... y copiando el contenido del archivo SeconHSQL.sql

```

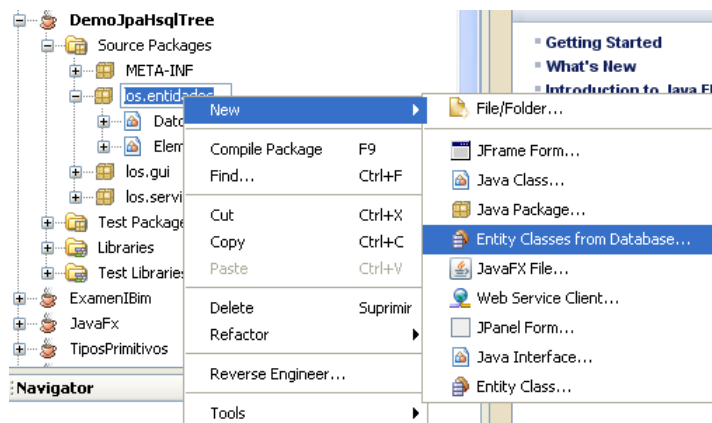
1 /*=====*/
2 /* Table: DATOSCONJUNTO */
3 /*=====*/
4 create cached table DATOSCONJUNTO (
5     DCJ_COD            INTEGER identity            not null,
6     DCJ_DES            VARCHAR(100)                not null,
7     constraint PK_DATOSCONJUNTO primary key (DCJ_COD)
8 );
9
10 /*=====*/
11 /* Table: ELEMENTOS_CONJUNTOS */
12 /*=====*/
13 create cached table ELEMENTOS_CONJUNTOS (
14     ELT_DCJ_COD        INTEGER identity            not null,
15     DCJ_COD            INTEGER                    not null,
16     ELE_ELT_DCJ_COD    INTEGER,
17     ELT_DCJ_NOM        VARCHAR(100)                not null,
18     constraint PK_ELEMENTOS_CONJUNTOS primary key (ELT_DCJ_COD)
19 );
20
21 alter table ELEMENTOS_CONJUNTOS
22     add constraint FK_ELEMENTO_CONJUNTO__DATOSCON foreign key (DCJ_COD)
23     references DATOSCONJUNTO (DCJ_COD);

```

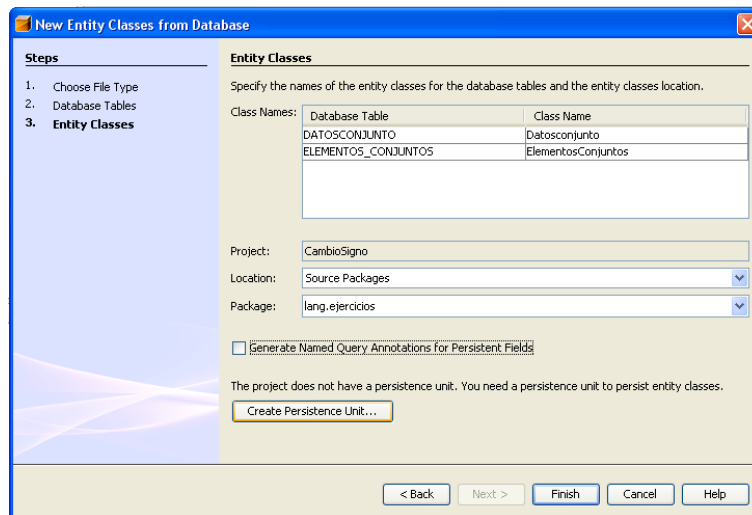
Y finalmente hacer clic en el icono que dice Run SQL (la primera de la derecha, luego del combo Connection)

### Creación de la capa de persistencia con JPA

En la pestaña Projects de NetBeans crear un nuevo proyecto, como cualquier otro proyecto, crear un nuevo paquete dentro de Source Packages que se llame entidades, sobre el paquete recién creado clic derecho y seleccionar New, luego Entity Classes from Database...



Lo anterior muestra una nueva ventana en donde se nos pide seleccionar una Database Connection seleccionar la empieza con jdbc:hsqldb:..., seleccionamos todas las tablas disponibles y podemos ver como la tabla maestra (DATOSCONJUNTOS) se muestra de otro color diferente a la tabla detalle (ELEMENTOS\_CONJUNTOS). Clic en Next, en la pantalla desmarcar Generate Named Query... y hacer clic en el botón Create Persistence Unit... en la nueva pantalla dejar todos los valores por defecto y dar clic en Create y luego en Finish



## Creación de NamedQuery y modificación método toString()

Abrir la clase Datosconjunto y bajo la anotación @Table ubicar el siguiente código:

```
@NamedQuery(name = "obtenerRaiz",
    query = "SELECT CONJUNTO FROM Datosconjunto CONJUNTO WHERE CONJUNTO.dcjCod = :idProvincia")
```

Abrir la clase ElementosConjuntos y bajo la anotación @Table ubicar el siguiente código:

```
@NamedQuery(name = "buscarElementos",
    query = "SELECT ELEM FROM ElementosConjuntos ELEM WHERE ELEM.eltDcjCod = :idPadre")
```

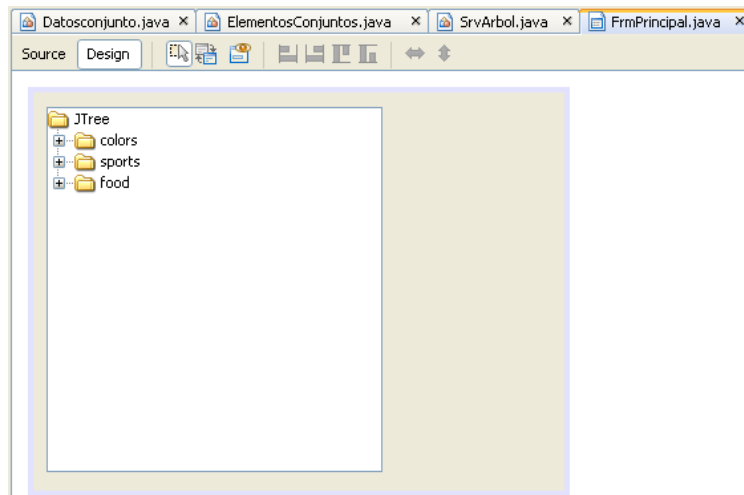
**En éste punto algo importante que destacar si se pudieron percatar los nombres de la tablas son que se utilizan en las Named Query (Datosconjunto y ElementosConjuntos) son diferentes a las tablas de la base de datos. Los nombres de las Named Query son los nombres de las clases que representan a esas tablas. Igual sucede con los campos de las sentencias WHERE, se utilizan los nombres de las propiedades de las clases y no de las columnas de la base de datos.**

Buscar el método toString() de ambas clases y modificarlo de tal manera que únicamente devuelva el nombre y descripción.

## Creación de la clase de Servicio

Ésta clase nos permite crear todos los objetos necesarios para la realizar la consulta con la base de datos y poder armar el árbol. Para la creación de ésta clase copiar el contenido de la clase SrvArbol.java. En próximas entregas haremos la explicación de la misma.

## Creación de la interfaz gráfica de usuario

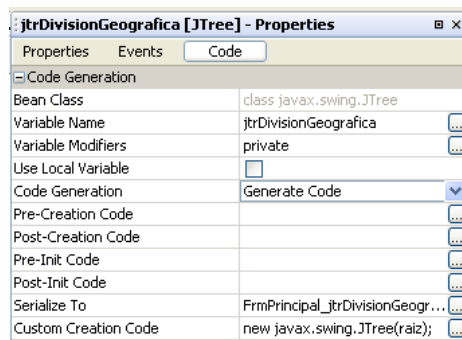


Crear un nuevo paquete con nombre gui, agregar en ella un JFrame, luego agregar un JTree y ubicarlo en algún lugar dentro del JFrame. Ver el código fuente y agregar una propiedad a la clase que represente la raíz del árbol así:

```
private DefaultMutableTreeNode raiz;
```

Vuelva a la pantalla de diseño, seleccione el JTree vaya a la paleta de propiedades seleccione la opción Code y en Custom Creation Code digitar:

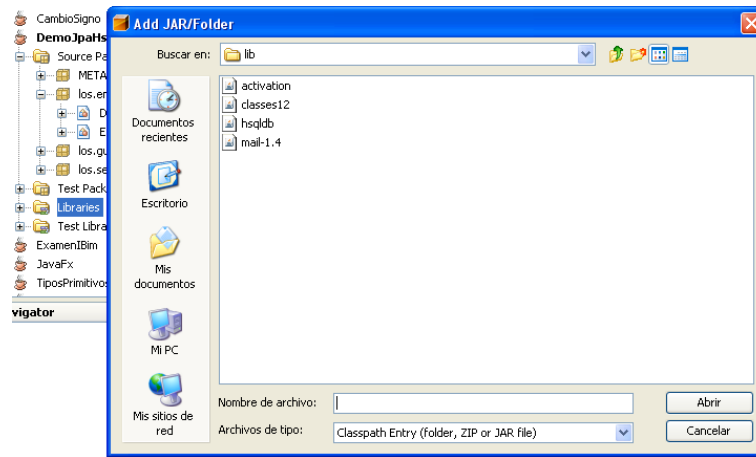
```
new javax.swing.JTree(raiz);
```



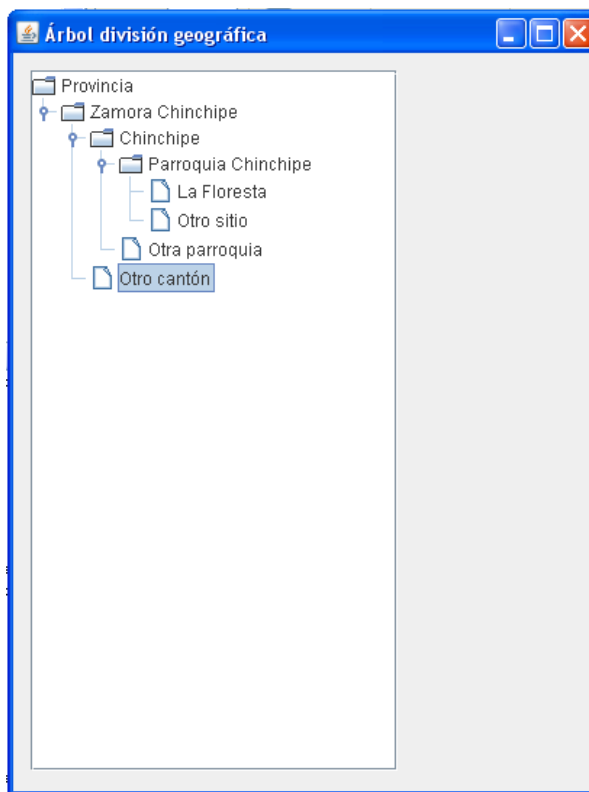
Agregar los métodos cargarArbol y obtenerHijos, según el archivo FrmPrincipal.java. Éstos métodos sirven para crear los nodos de los árboles de una manera recursiva.

### **Agregar Driver al proyecto**

Antes de poder ejecutar el proyecto debemos agregar el driver al mismo, para que puede conectarse con la base de datos nuestra aplicación. Para lo cual se debe hacer lo siguiente: dentro del proyecto seleccionar Libraries, clic derecho Add JAR/Folder... ubicar el archivo hsqldb.jar y listo



Si se ha realizado todo lo anterior se puede ejecutar la aplicación y el resultado será lo siguiente:



Puede cargar el proyecto para ir controlando el código que usted desarrolla y el código empleado.

Las tablas que se utilizaron en el proyecto se pueden ver en el siguiente diagrama:

