[14-08-2007]_[17-08-2007]_[24-08-2007]_[31-08-2007]_[03-09-2007]

Vamos a realizar un proyecto en Lenguaje de programación Java, que servirá para ver como se realiza una conexión a una base de datos MySQL desde una aplicación java, usando en una primera instancia el driver JDBC (MySQL Connector/J).

En un principio vamos a crear una clase visual en Eclipse, que será la ventana inicial de nuestra aplicación.

Iniciar Eclipse 3.2.

File -> **New** -> **Project**

En el asistente seleccionar: **Java Project**

Clickear en: Next
En asistente abierto completar:
Project Name: **Proyecto_ABM**
(*) Create new project in workspace
(*) Use default JRE (Currently 'jdk1.6.0_02')
(*) Use a project folder as root for sources and class files
Clickear en: **Next**
Clickear en: **Finish**

Estos pasos han creado la siguiente estructura en el directorio:

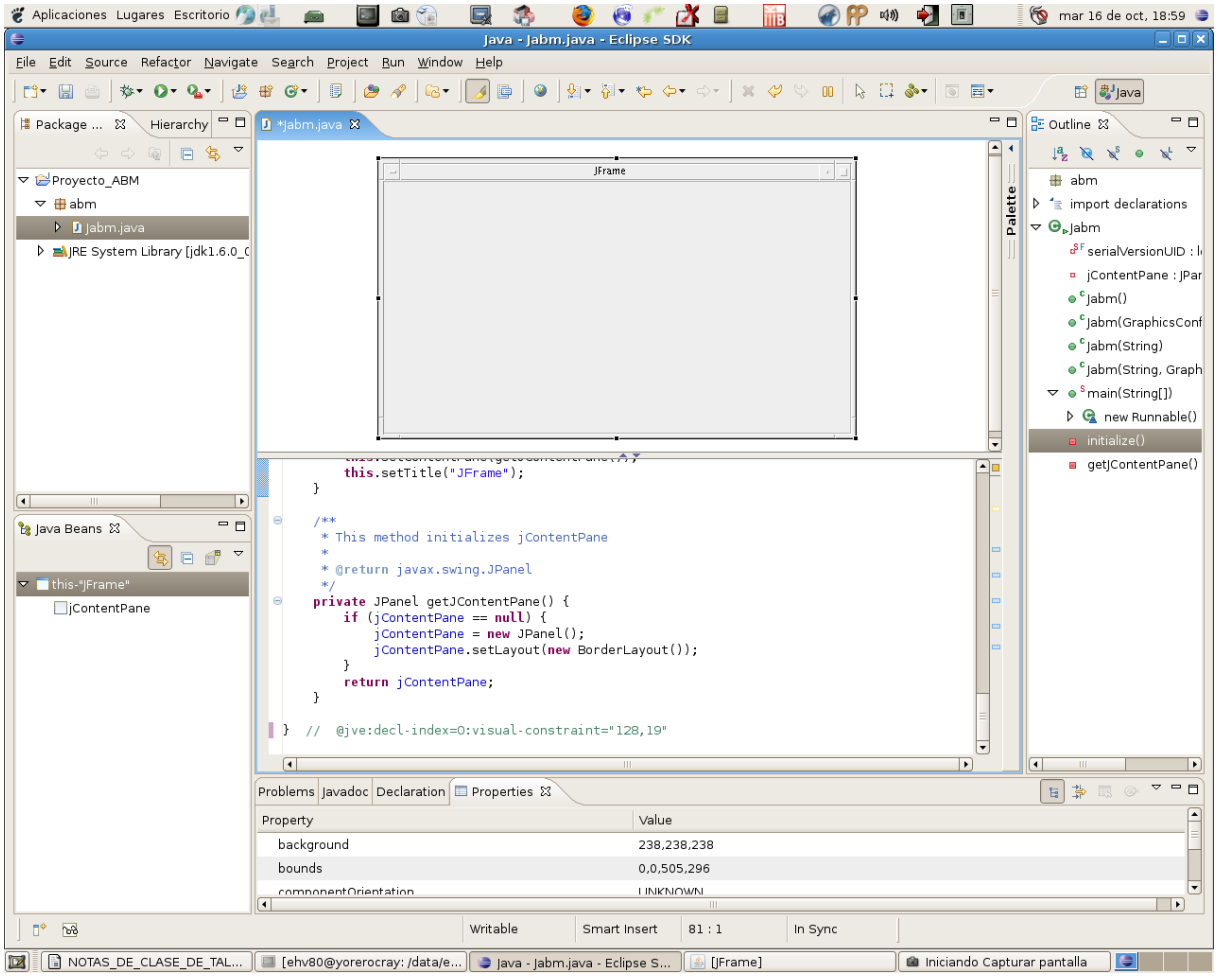
```
/data/eclipse-workspace
        Proyecto_ABM/
            .settings/
                org.eclipse.jdt.core.prefs
            .project
            .classpath
```

La primer clase java que necesitamos será una “clase visual”, “Visual Class”

Para esto vamo al **“Package explorer”**:
Clickear con botón derecho del mouse sobre el proyecto: **Proyecto_ABM**
y
Clickear en: **New** -> **Visual Class**

En el asistente abierto:
Source folder: **Proyecto_ABM**
Package: **abm**
Name: Jabm
Modifiers: **(*) public**
En Style seleccionar: Swing -> Frame
En Superclass dejar: **javax.swing.JFrame**
En Interfaces: (SIN COMPLETAR)
Tildar: **public static void main(String[] args)**
Tildar: **Constructors from superclass**
Tildar: **Inherited abstract methods**
Tildar: **Generate comments**

Se obtiene en Eclipse 3.2 con el plugin: VE-runtime-1.2.1 instalado en las carpetas correspondientes (“features” y “plugins”) algo similar a lo siguiente:



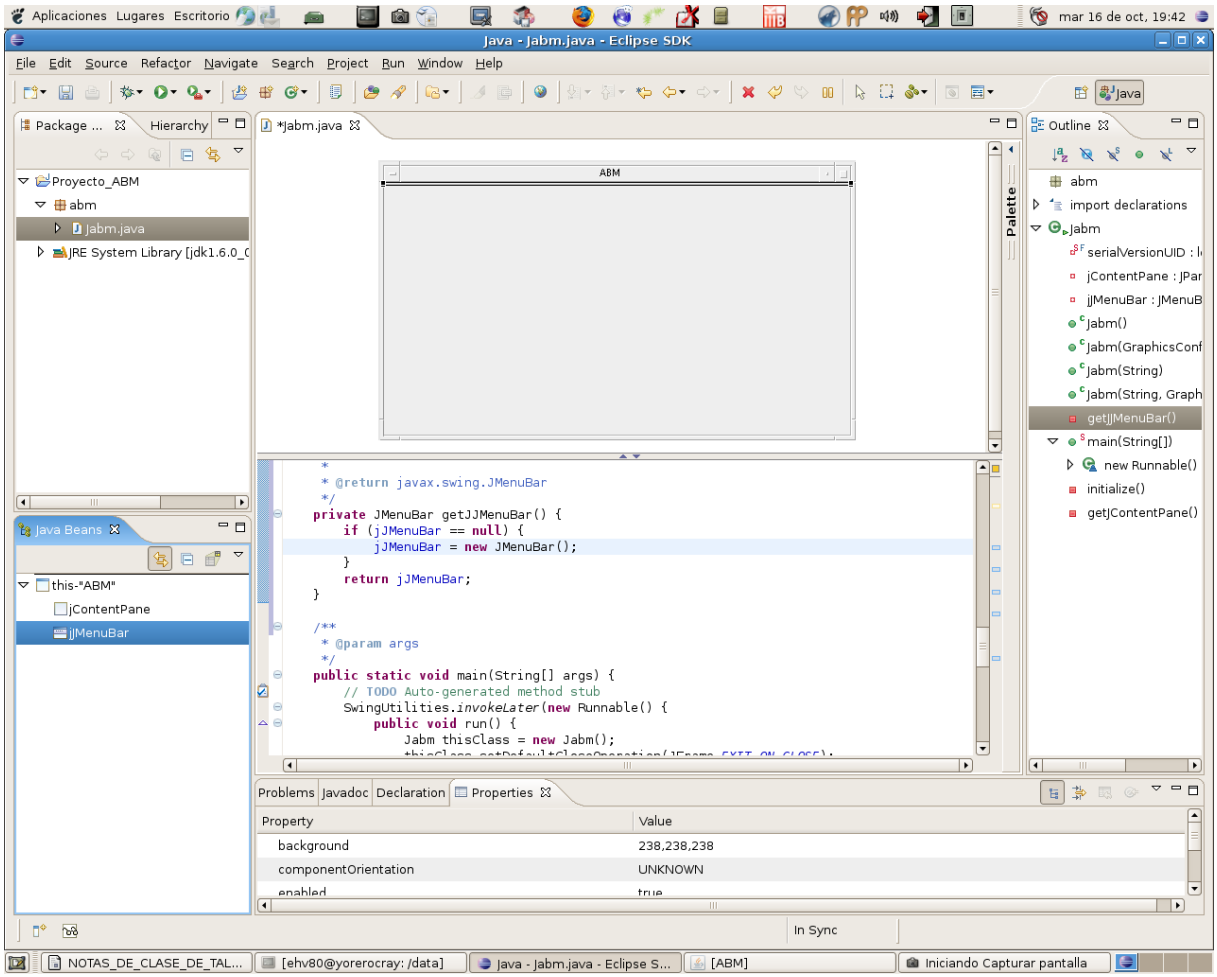
Vamos a cambiar el título de la ventana, **“JFrame”** por **“ABM”**, para esto:

En la solapa “Java Beans”:
Clickear con botón derecho del mouse sobre: **this.”JFrame”**
En el menú desplegado elegir: **Set Title**
En el campo de texto introducir: **ABM**

Lo que sigue es dotar a la **“ventana” (JFrame)** ABM de una **“bara de menú” (JMenuBar)** que contiene tantos **“menús desplegables” (JMenu)** como se necesiten, en nuestro sencillo ejemplo añadiremos uno solo. Dentro del **“menú desplegable” (JMenu)** se listan los **“items de menú” (JMenuItem)** que contiene nuestra aplicación, estos serán tres: Altas, Bajas, Modificaciones.

Para agregar la **“bara de menú” (JMenuBar)**:
Ubicar el mouse en el área del Visual Editor y clickear en: **Palette**
Clickear en: **Swing Menus**
Clickear en: **JMenuBar**
Dirigir el mouse hacia la solapa **“Java Beans”**
y clickear en: **this-”ABM”**

De esta manera obtenemos algo como la siguiente figura:



Para agregar el **“menú desplegable” (JMenu)**:
Ubicar el mouse en el área del Visual Editor y clickear en: **Palette**
Clickear en: **Swing Menus**
Clickear en: **JMenu**
Dirigir el mouse hacia la solapa **“Java Beans”**
y clickear en: **JMenuBar**

Para agregar el primer **“ítem de menú” (JMenuItem)**:
Ubicar el mouse en el área del Visual Editor y clickear en: **Palette**
Clickear en: **Swing Menus**
Clickear en: **JMenuItem**
Dirigir el mouse hacia la solapa **“Java Beans”**
y clickear en: **JMenu**

De esta misma forma agregamos 2 **“ítems de menú”** más.

Ahora vamos a configurar algunas de las propiedades de estos elementos **“Swing Menus”**.

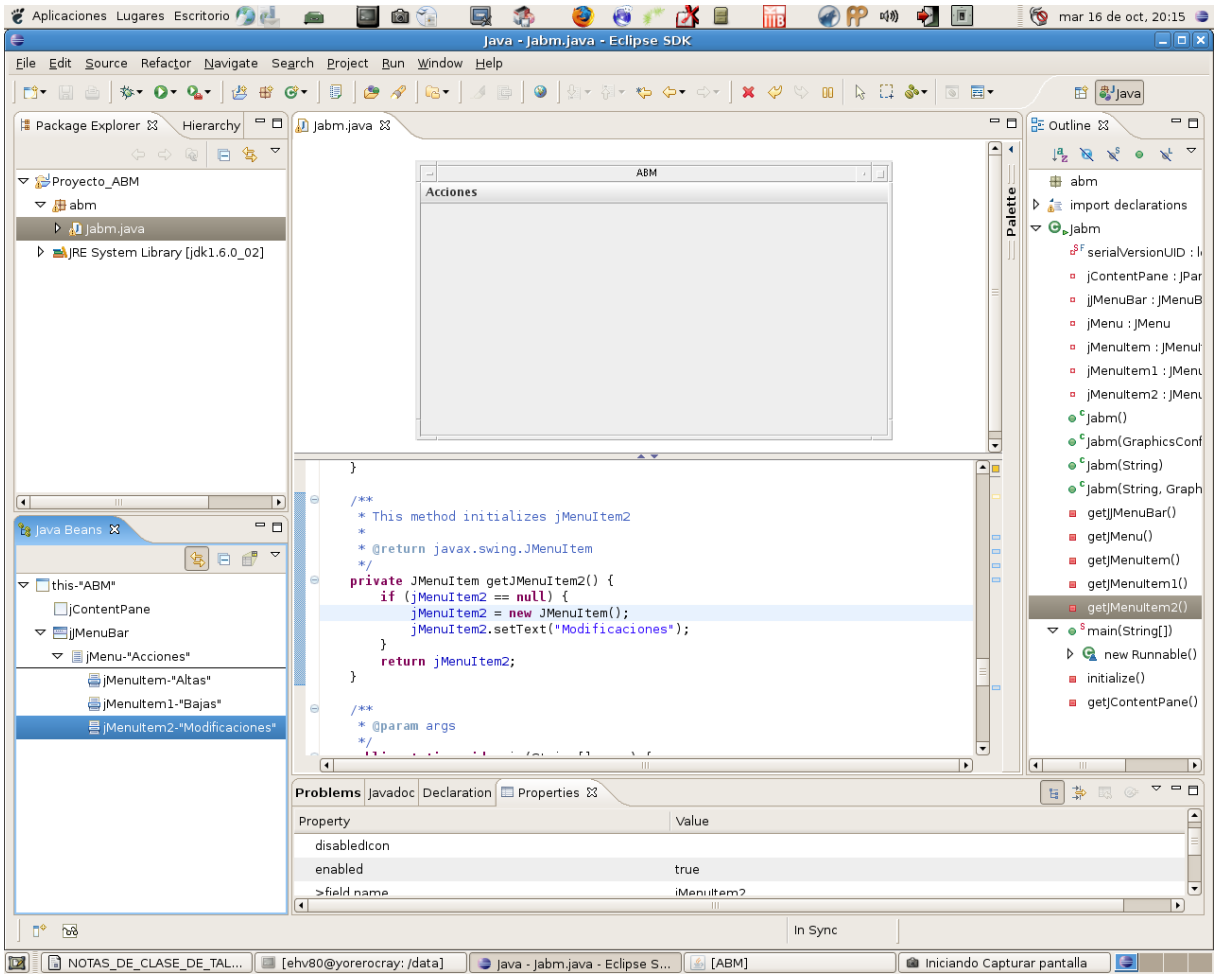
Para cambiar el texto mostrado por el **“menú desplegable” (JMenu)**:
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear con botón derecho del mouse sobre: **JMenu**
Clickear en: **Set Text**
Introducir en el campo de texto mostrado: **Acciones**

Para cambiar el texto mostrado por el **“ítem de menú” (JMenuItem)**:
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear con botón derecho del mouse sobre: **JMenuItem**
Clickear en: **Set Text**
Introducir en el campo de texto mostrado: **Altas**

Para cambiar el texto mostrado por el **“ítem de menú” (JMenuItem)**:
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear con botón derecho del mouse sobre: **JMenuItem1**
Clickear en: **Set Text**
Introducir en el campo de texto mostrado: **Bajas**

Para cambiar el texto mostrado por el **“ítem de menú” (JMenuItem)**:
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear con botón derecho del mouse sobre: **JMenuItem2**
Clickear en: **Set Text**
Introducir en el campo de texto mostrado: **Modificaciones**

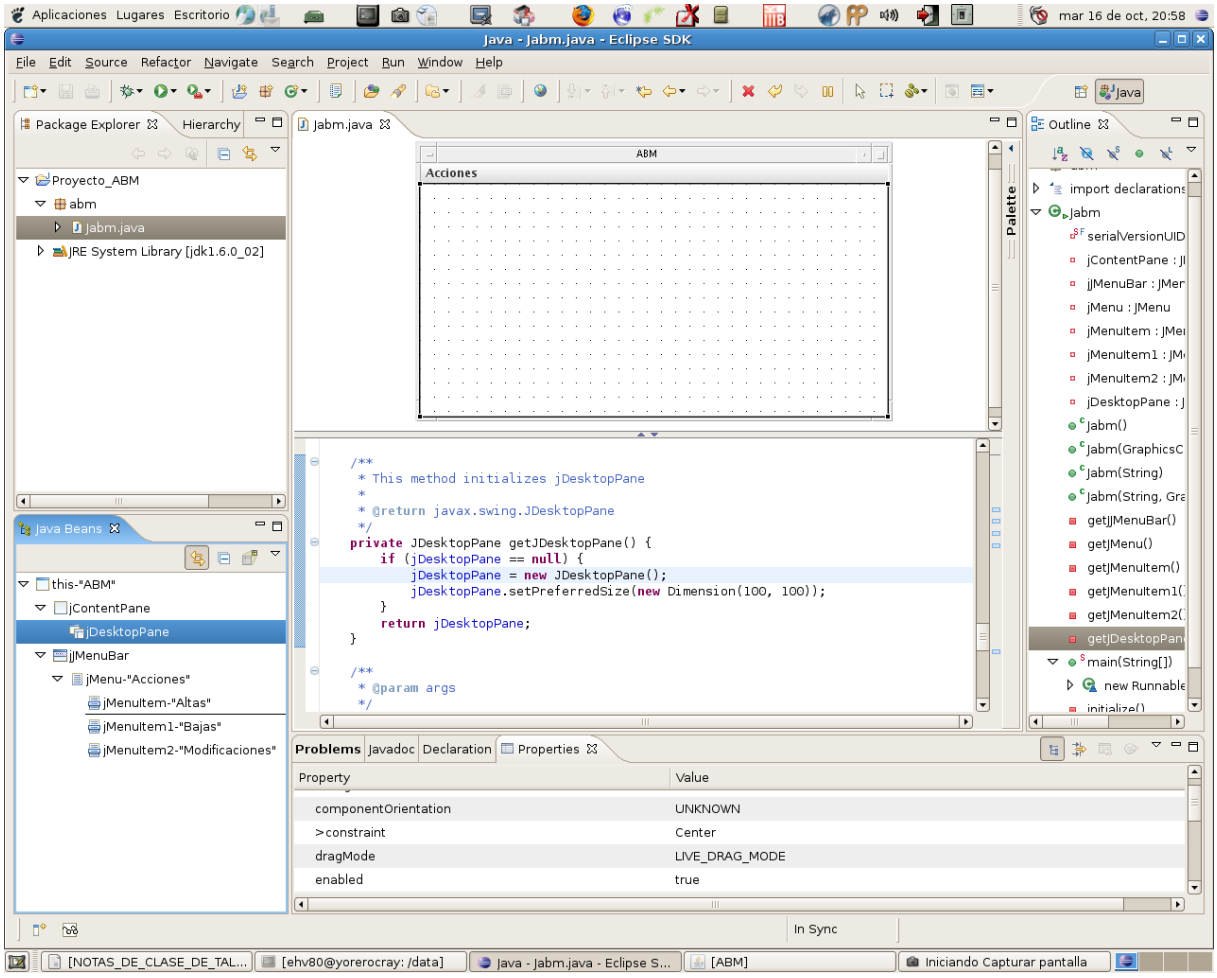
Una vez realizados todos estos pasos, obtendremos en Eclipse 3.2 algo similar a la siguiente imagen:



Vamos a agregar a la clase: **Jabm.java** un **“contenedor intermedio”** (de entre ellos elegimos: **JDesktopPane**):
Ubicar el mouse en el área del Visual Editor y clickear en: **Palette**
Clickear en: **Swing Containers**
Clickear en: **JDesktopPane**
Dirigir el mouse hacia la solapa **“Java Beans”**
y clickear en: **jContentPane**

Vamos a configurar algunas de las propiedades de este **JDesktopPane**
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear dos veces sobre: **jDesktopPane**
Dirigir el mouse hacia la solapa **“Properties”**
Clickear en: >preferredSize y poner: 100,100
Clickear en: >constraint y elegir: Center

Luego de estos deberíamos ver en Eclipse 3.2 algo simlar al siguiente esquema:



Nos resta manejar lo que sucede con los **eventos** que un usuario de la aplicación produce al utilizar el mouse sobre los **“items del menú”**. Para esto vamos a configurar los elementos denominados **“actionPerfomed”** para cada uno de nuestros tres **“JMenuItem”**.

Para configurar el **evento de clickear sobre el "JMenuItem Altas"**:
Dirigir el mouse hacia la solapa **"Java Beans"**
Clickear con el botón derecho del mouse en: **jMenuItem-"Altas"**
Clickear en: **Events**
Clickear en: **actionPerformed**
Dirigir el mouse hacia el área del código fuente:
Editar el método actionPerformed para que contenga:

```
public void actionPerformed(java.awt.event.ActionEvent e) {
	//System.out.println("actionPerformed()");
	// TODO Auto-generated Event stub actionPerformed()
	// arregloJIF tendrá los componentes del JDesktopPane
	JInternalFrame[] arregloJIF = jDesktopPane.getAllFrames();
	boolean existeVentana = false;
	for(int i = 0 ; i < arregloJIF.length ; i++)
	{
		if( arregloJIF[i] instanceof JAltas){
			arregloJIF[i].toFront();
			existeVentana = true;
		}
	}
	if(!existeVentana) {
		JAltas altas = new JAltas();
		jDesktopPane.add(altas);
		altas.setVisible(true);
	}
}
```

De manera similar configuraremos los **eventos de clickear** en los **"JMenuItem Bajas"** y **"JMenuItem Modificaciones"**.

Hay que notar que deberemos **crear nuevas clases "Visuales"** en nuestro "**Proyecto_ABM**" para que funcione la aplicación, entre ellas:

- **JAltas.java**
- **JBajas.java**
- **JModificaciones.java**
- y otras que veremos más adelante

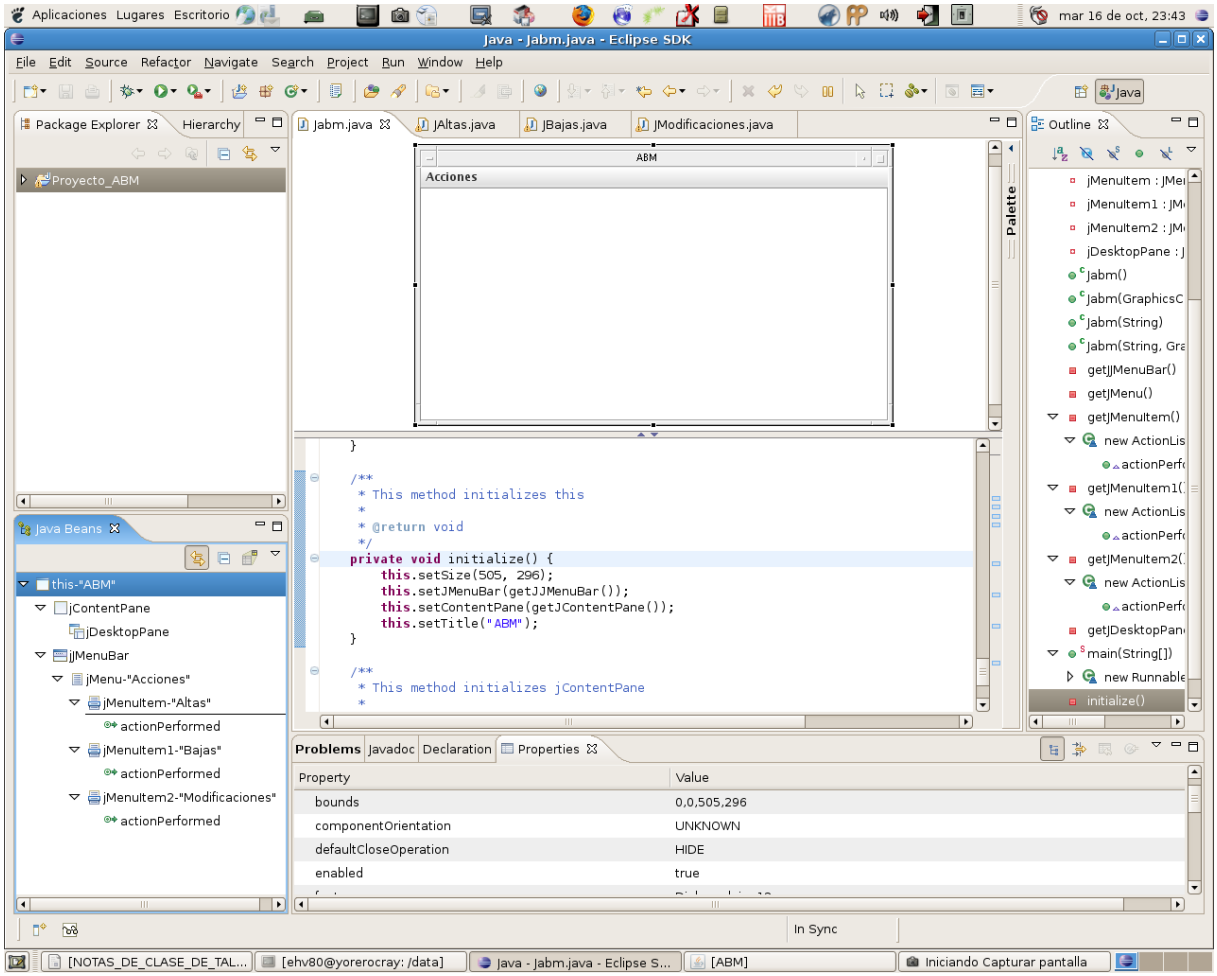
En este punto debemos crear las clases "Visuales" que necesitamos:

Para agregar la clase "Visual": **JAltas.java** al proyecto: **Proyecto_ABM**
Dirigir el mouse hacia la solapa **"Package Explorer"**
Clickear con botón derecho del mouse sobre: **Proyecto_ABM**
Clickear en: **New** -> **Visual Class**
Introducir en "Source Folder": **Proyecto_ABM**
Introducir en "Package": **abm**
Introducir en "Name": JAltas
Elegir en "Modifiers": **(*) public**
Elegir en "Style": Swing -> InternalFrame
Introducir en "Superclass": **javax.swing.JInternalFrame**
Tildar: **public static void main(String[] args)**
Tildar: **Constructors from superclass**
Tildar: **Inherited abstract methods**
Tildar: **Generate comments**
Clickear en: **Finish**

Siguiendo esta forma de proceder creamos también las clases “Visuales”:

- **JBajas.java**
- **JModificaciones.java**

En este punto podemos observar en Eclipse 3.2 algo como lo siguiente:



A continuación, dejamos momentáneamente la clase “Visual”: **Jabm.java** para editar la clase “Visual”: **JAltas.java**

Para hacer esto basta con dirigir el mouse hacia el área del Visual Editor y clickear en la solapa “**JAltas.java**”.

Esta clase contendrá el formulario que el usuario debe completar con los datos que necesite ingresar, para nuestro ejemplo usaremos:

- **Nombre**
- **Apellido**
- **DNI ó LE ó LC**
- **Edad**

Antes que nada debemos agregar **un contenedor Swing**:
Dirigir el mouse hacia el área del Visual Editor
Clickear en: **Palette**
Clickear en: **Swing Containers**
Clickear en: **JDesktopPane**
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear en: **jContentPane**

Vamos a configurar algunas de las propiedades de este contenedor Swing.
Dirigir el mouse hacia la solapa **“Properties”**
Elegir en “>constraint”: Center
Elegir en “preferredSize 100,100

Lo que sigue es agregar las **“etiquetas de los datos” (JLabel)** que se solicitarán, en nuestra aplicación serán tres: “Nombre”, “Apellido”, “Edad”

Para agregar la **“etiqueta de datos” (JLabel) “Nombre”**:
Dirigir el mouse hacia el área del Visual Editor
Clickear en: **Palette**
Clickear en: **Swing Components**
Clickear en: **JLabel**
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear en: **jDesktopPane**

Para configurar el texto que mostrará la **“etiqueta” (JLabel) “Nombre”**:
Dirigir el mouse hacia la solapa “Java Beans”
Clickear con botón derecho del mouse en: **Jlabel-”JLabel”**
Clickear en: **Set Text**
Introducir: **Nombre**

Para configurar algunas propiedades de la **“etiqueta” (JLabel) “Nombre**”:
Dirigir el mouse hacia la solapa **“Properties”**
Clickear en “location” e introducir: 25,25
Clickear en “size” e introducir: 60,20
Clickear en “>horizontalAlignment” y elegir: CENTER
Clickear en “>horizontalTextPosition” CENTER

De la misma forma se deben agregar las **“etiquetas de datos” (JLabel) “Apellido”** y **“Edad”**, teniendo en cuenta otros valores para sus propiedades.

Continuando con la edición de la clase Visual: **JAltas.java**
Para agregar una “casilla de selección mútliple” (JComboBox) que contiene las opciones: “DNI”, “LE”, “LC”:
Dirigir el mouse hacia el área del Visual Editor
Clickear en: **Palette**
Clickear en: **Swing Components**
Clickear en: **JComboBox**
Dirigir el mouse hacia la solapa “Java Beans”
Clickear en: **jDesktopPane**

Para configurar algunas de las propiedades de esta **“casilla de selección múltiple” (JComboBox)**:
Dirigir el mouse hacia la solapa **“Properties”**
Clickear en “location” e introducir: 25,85
Clickear en “>preferredSize” e introducir: 60,20
Clickear en “size” e introducir: 60,20
Clickear en “name”: Tipo de Documento

Para agregar los **“items”** a la **“casilla de selección mútltiple”(JComboBox)**:

Dirigir el mouse hacia el área del código fuente:

Editar la sección referida al **JComboBox** para que luzca como a continuación:

```
private JComboBox getJComboBox() {
          if (jComboBox == null) {
                jComboBox = new JComboBox();
                jComboBox.setSize(new Dimension(60, 20));
                jComboBox.setPreferredSize(new Dimension(60, 20));
                jComboBox.setName("Tipo de Documento");
                jComboBox.setLocation(new Point(25, 90));
                jComboBox.addItem("DNI");
                jComboBox.addItem("LC");
                jComboBox.addItem("LE");
          }
```

Siguiendo con la edición de la clase Visual: **JAltas.java**

Para agregar las **“cajas de texto” (JTextField)** que sirven para que el usuario de la aplicación pueda introducir los datos requeridos:

Dirigir el mouse hacia el área del Visual Editor

Clickear en: **Palette**

Clickear en: **Swing Componets**

Clickear en: **JTextField**

Dirigir el mouse hacia la solapa **“Java Beans”**

Clickear en: **jDesktopPane**

Para configurar algunas propiedades del JTextField:

Dirigir el mouse hacia la solapa “Properties”

Clickear en “>location” e introducir:	125,25
Clickear en “name” e introducir:	Caja_Nombre
Clickear en “>preferredSize” e introducir:	160,20
Clickear en “>size” e introducir:	160,20
Clickear en “text” e introducir:	Ingrese el Nombre...

Estos mismos pasos se deben seguir para agregar las otras **“cajas de texto” (JTextField)** correspondientes a las **“etiquetas de datos” (JLabel) “Apellido”** y **“Edad”**, aunque con otros valores en sus propiedades.

Vamos a agregar dos “botones” (JButton) a nuestra clase Visual **JAltas.java**
uno se llamará **“Grabar”** y el otro será nombrado **“Salir”**:

Para agregar el **“botón” (JButton)** a la clase Visual **JAltas.java**:

Dirigir el mouse hacia el área del Visual Editor

Clickear en: **Palette**

Clickear en: **Swing Components**

Clickear en: **JButton**

Dirigir el mouse hacia la solapa **“Java Beans”**

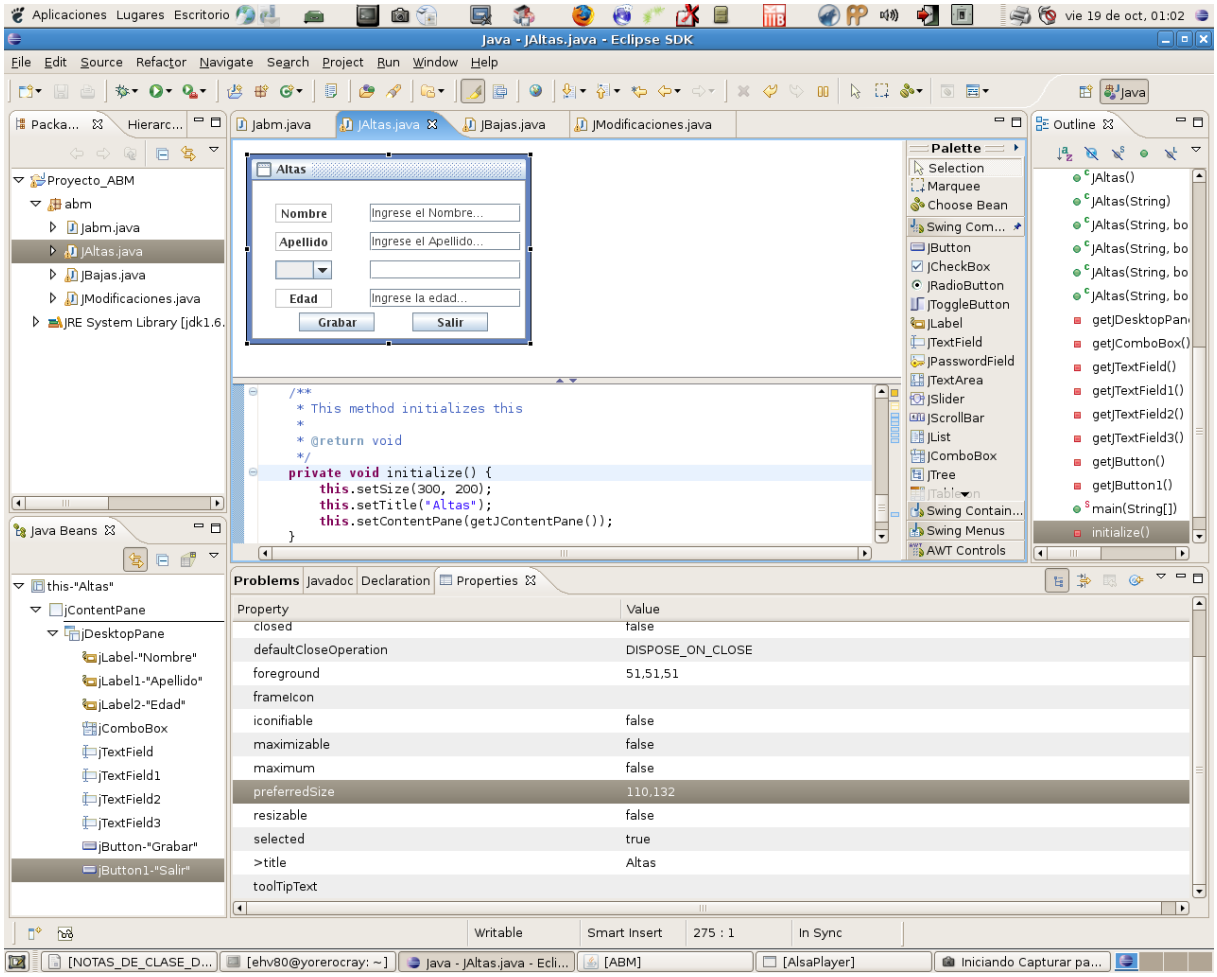
Clickear en: **jDesktopPane**

Para configurar algunas de las propiedades del **JButton "Grabar"**
Dirigir el mouse hacia la solapa "Properties"
Clickear en ">location" e introducir: 50,140
Clickear en ">preferredSize"e introducir: 80,20
Clickear en "size" e introducir: 80,20
Clickear en "text" e introducir: Grabar
Clickear en "horizontalTextPosition" y elegir: CENTER

De la misma forma agregamos el otro **"botón" (JButton) "Salir"** considerando que tendrá necesariamente otros valores en sus propiedades.
Una propiedad importante con respecto a la clase Visual que estamos editando, **JAltas.java** es el Título que mostrará al ejecutarse.

Para configurar el título de la clase Visual **JAltas.java**:
Dirigir el mouse hacia el área del Visual Editor:
Clickear en la barra de Título de la clase Visual **JAltas.java**
Dirigir el mouse hacia la solapa **"Properties"**
Clickear en "title" e introducir: **Altas**

A esta altura del desarrollo la clase Visual debería lucir como se muestra en la siguiente figura:



Dejaremos por un momento la edición de la clase Visual **JAltas.java** para editar la clase Visual **JModificaciones.java**.

La clase Visual **JModificaciones.java** presentará una ventana similar a la que muestra la clase Visual **JAltas.java**, pero contendrá un **"botón"(JButton)** adicional que lo llamaremos **"Buscar"**.

Para obtener esta clase en forma rápida vamos a copiar el contenido del código fuente de la clase Visual **JAltas.java** y ponerlo en el código fuente de la clase Visual **JModificaciones.java**, y luego configurar las propiedades de cada uno de sus componentes visuales.

La primer propiedad que vamos a modificar en la clase Visual **JModificaciones.java** será el nombre de la clase, pues debe ser: JModificaciones, para ello usaremos el Editor de código fuente.
A continuación se listan los cambios que hay que hacer al código fuente de la clase Visual **JModificaciones.java**:

- Seleccionar el atibuto `jContentPane`
  Dirigir el mouse hacia la "barra de Menú de Eclipse"
  Clickear en "Refactor" -> "Rename..."
  En el asistente que aparece completar:

New Name:	jContentPaneModificaciones
Tildar:	Update references
Tildar:	Update textual occurrences in comments and strings (forces preview)
Tildar:	Rename getter: 'getjContentPane' to 'getjContentPaneModificaciones'

  En cada "vista previa" verificar como quedaría el cambio y clickear en "OK"

- Utilizar el mismo procedimiento anterior para hacer un "Refactor" del atributo "`jDesktopPane`" por "`jDesktopPaneModificaciones`"

- Utilizar el mismo procedimiento anterior para hacer un "Refactor" del atributo "`jLabel`" por "`jLabelModificacionesNombre`"

- Utilizar el mismo procedimiento anterior para hacer un "Refactor" del atributo "`jLabel1`" por "`jLabelModificacionesApellido`"

- Utilizar el mismo procedimiento anterior para hacer un "Refactor" del atributo "`jLabel2`" por "`jLabelModificacionesEdad`"

- Utilizar el mismo procedimiento anterior para hacer un "Refactor" del atributo "`jComboBox`" por "`jComboBoxModificaciones`"

- Utilizar el mismo procedimiento anterior para hacer un "Refactor" del atributo "`jTextField`" por "`jTextFieldModificacionesNombre`"

- Utilizar el mismo procedimiento anterior para hacer un "Refactor" del atributo "`jTextField1`" por "`jTextFieldModificacionesApellido`" o

- Utilizar el mismo procedimiento anterior para hacer un "Refactor" del atributo "`jTextField2`" por "`jTextFieldModificacionesDocumento`"

- Utilizar el mismo procedimiento anterior para hacer un “Refactor” del atributo “jTextField3” por “jTextFieldModificacionesEdad”

- Utilizar el mismo procedimiento anterior para hacer un “Refactor” del atributo “jButton” por “jButtonModificacionesGrabar”

- Utilizar el mismo procedimiento anterior para hacer un “Refactor” del atributo “jButton1” por “jButtonModificacionesSalir”

- Agregar el atributo: `private JButton jButton2 = null;`

- En el método: `private JDesktopPane getJDesktopPaneModificaciones() {`

  agregar en la última del cuerpo del método:
  `jDesktopPaneModificaciones.add(getJButton2(), null);`

  Tener en cuenta que como aquí se invoca al método “getJButton2()” que todavía no existe en el código fuente, el editor nos avisará de que hay un error, por ello lo agregaremos a continuación ya que es necesario:

```
/**
     * This method initializes jButton2
     *
     * @return javax.swing.JButton
     */
    private JButton getJButton2() {
          if (jButton2 == null) {
                jButton2 = new JButton();
                jButton2.setSize(new Dimension(80, 20));
                jButton2.setPreferredSize(new Dimension(80, 20));
                jButton2.setText("Buscar");
                jButton2.setHorizontalTextPosition(SwingConstants.CENTER);
                jButton2.setLocation(new Point(0,0));
          }
          return jButton2;
    }
```

- Utilizar el procedimiento anterior para hacer un “Refactor” del atributo “jButton2” por “jButtonModificacionesBuscar”

- Utilizar el mismo procedimiento anterior para hacer un “Refactor” del nombre de la clase Visual que actualmente es “JAltas” pero debe ser “JModificaciones”

Como puede observarse debemos cambiar las posiciones y tamaños de los tres **JButton**: **“Grabar”**, **“Buscar”** y **“Salir”**.

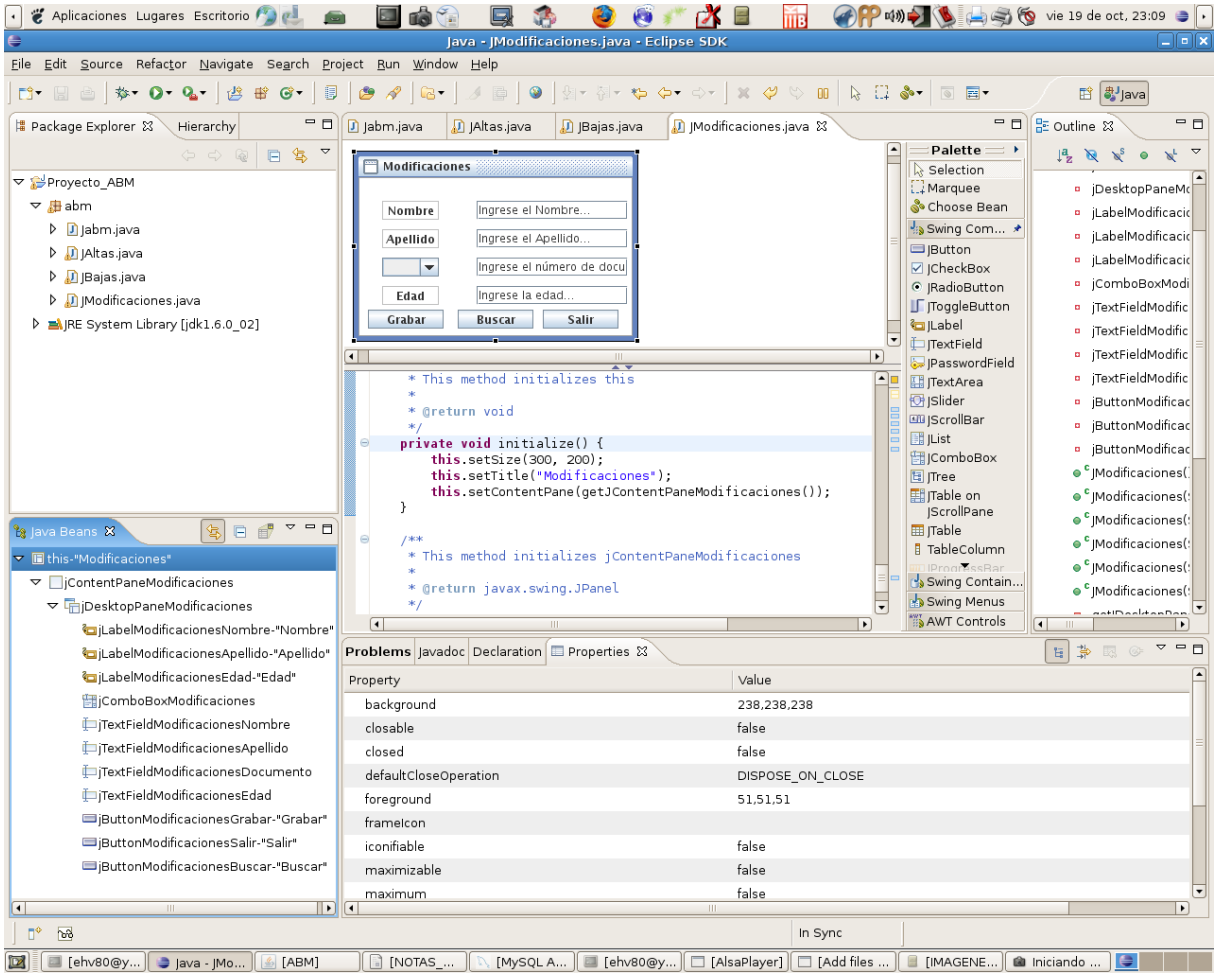
Para configurar las propiedades del “botón” (JButton) “Grabar”:
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear en: **JButtonModificacionesGrabar-”Grabar”**
Dirigir el mouse hacia la solapa **“Properties”**
Clickear en “>location” e introducir: 10,140

Para configurar las propiedades del “botón” (JButton) “Buscar”:
Dirigir el mouse hacia la solapa “Java Beans”
Clickear en: JButtonModificacionesBuscar-”Buscar”
Dirigir el mouse hacia la solapa “Properties”
Clickear en “>location” e introducir: 105,140

Para configurar las propiedades del **“botón” (JButton) “Salir”**:
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear en: **JButtonModificacionesSalir-”Salir”**
Dirigir el mouse hacia la solapa **“Properties”**
Clickear en “>location” e introducir: 195,140

Algo muy importante que falta modificar es el título de la vetana:
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear con botón derecho del mouse sobre: **this-”Altas”**
Clickear en “Set Title” e introducir: **Modificaciones**

Luego de estos cambios la ventana que muestra la clase Visual **JModificaciones.java** debería lucir como se muestra a continuación:



Ahora dejaremos un poco la clase Visual **JModificaciones.java** y volveremos sobre la clase Visual **JAltas.java** para hacer otras modificaciones que nos sirven para lograr un mismo estilo uniforme en todos los archivos de código fuente.

Nuevamente utilizaremos la opción de la “barra de menú” de Eclipse denominada **“Refactor” -> “Rename...”** para renombrar los siguientes atributos:

- “`jLabel`” renombrar a “`jLabelAltasNombre`”
- “`jLabel1`” renombrar a “`jLabelAltasApellido`”
- “`jLabel2`” renombrar a “`jLabelAltasEdad`”
- “`jComboBox`” renombrar a “`jComboBoxAltas`”
- “`jTextField`” renombrar a “`jTextFieldAltasNombre`”
- “`jTextField1`” renombrar a “`jTextFieldAltasApellido`”
- “`jTextField2`” renombrar a “`jTextFieldAltasDocumento`”
- “`jTextField3`” renombrar a “`jTextFieldAltasEdad`”
- “`jButton`” renombrar a “`jButtonAltasGrabar`”
- “`jButton1`” renombrar a “`jButtonAltasSalir`”

Ahora vamos a pasar a configurar la clase Visual **JBajas.java**.

La **“ventana” (JInterFrame)** que presentará esta clase será similar a las otras dos, pero la diferencia está en que uno de sus “botones” (JButton) se llamará **“Borrar”**, éste servirá para eliminar un registro de la Base de Datos que definiremos en el **Gestor de Base de Datos MySQL** con el nombre: **Taller3**, la cuál contendrá una **Tabla** llamada: **ABM** que constará de los siguientes campos:

- **VARCHAR : nombre**
- **VARCHAR : apellido**
- **VARCHAR : tpdoc**
- **INTEGER : ndoc**
- **SMALLINT : edad**

Para **obtener rápidamente esta clase Visual JBajas.java**, nuevamente, **copiaremos** el código fuente de la clase Visual **JModificaciones.java** y lo colocaremos dentro del código fuente de **JBajas.java** y luego **realizaremos las modificaciones necesarias**.

A continuación se listan las modificaciones que se deben realizar mediante el editor de código fuente en la clase Visual **JBajas.java**:

- En la solapa “Java Beans” clickear con botón derecho del mouse en **this-”Modificaciones”**, luego clickear en **“Set Title”** e introducir: Bajas

- En la solapa “Java Beans” clickear con botón derecho del mouse en **jContentPaneModificaciones**, luego clickear en **“Rename Field”** e introducir: **jContentPaneBajas**

- En la solapa “Java Beans” clickear con botón derecho del mouse en **jDesktopPaneModificaciones**, luego clickear en **“Rename Field”** e introducir: **jDesktopPaneBajas**

- En la solapa “Java Beans” clickear con botón derecho del mouse en **jLabelModificacionesNombre**, luego clickear en **“Rename Field”** e introducir: **jLabelBajasNombre**

- En la solapa “Java Beans” clickear con botón derecho del mouse en **jLabelModificacionesApellido**, luego clickear en **“Rename Field”** e introducir: **jLabelBajasApellido**

- En la solapa “Java Beans” clickear con botón derecho del mouse en **jLabelModificacionesEdad**, luego clickear en **“Rename Field”** e introducir: **jLabelBajasEdad**

- En la solapa “Java Beans” clickear con botón derecho del mouse en **jComboBoxModificaciones**, luego clickear en **“Rename Field”** e introducir: **jComboBoxBajas**

- En la solapa “Java Beans” clickear con botón derecho del mouse en **jTextFieldModificacionesNombre**, luego clickear en **“Rename Field”** e introducir: **jTextFieldBajasNombre**
- En la solapa “Java Beans” clickear con botón derecho del mouse en **jTextFieldModificacionesApellido**, luego clickear en **“Rename Field”** e introducir: **jTextFieldBajasApellido**

- En la solapa “Java Beans” clickear con botón derecho del mouse en **jTextFieldModificacionesDocumento**, luego clickear en **“Rename Field”** e introducir: **jTextFieldBajasDocumento**

- En la solapa “Java Beans” clickear con botón derecho del mouse en **jTextFieldModificacionesEdad**, luego clickear en **“Rename Field”** e introducir: **jTextFieldBajasEdad**

- En la solapa “Java Beans” clickear con botón derecho del mouse en **jButtonModificacionesBorrar**, luego clickear en **“Rename Field”** e introducir: **jButtonBajasBorrar**
  Nuevamente clickear con derecho del mouse en **jButtonBajasBorrar**, luego clickear en **“Set Text”** e introducir: **Borrar**

- En la solapa “Java Beans” clickear con botón derecho del mouse en **jButtonModificacionesSalir**, luego clickear en **“Rename Field”** e introducir: **jButtonBajasSalir**

- En la solapa “Java Beans” clickear con botón derecho del mouse en **jButtonModificacionesBuscar**, luego clickear en **“Rename Field”** e introducir: **jButtonBajasBuscar**

- En el editor de código fuente seleccionar el nombre de la clase, que hasta es **“JModificaciones”** y renonbrarla (Refactor) **“JBajas”**

En este instante del desarrollo de nuestra aplicación tenemos las tres ventanas necesarias mínimamente configuradas para obtener una buena visualización de las mismas, pero nos falta explicitar que reacción tendrá cada uno de los componentes visuales de estas ventanas a la interacción que el usuario pueda provocar cuando las use.

También vamos a tener que agregar una clase que se encargará de tratar todo lo concerniente con la **conexión** a la **Base de datos (Taller3)**, la **inserción**, **modificación**, y **eliminación** de los datos que albergue nuestra **Tabla (ABM)**, la cuál deberá ser configurada en el **Sistema Gestor de Base de Datos: MySQL.**

Para agregar la clase, no visual, **JConexion.java**
Dirigir el mouse hacia la solapa **“Package Explorer”**
Clickear con el botón derecho del mouse en: **Proyecto_ABM**
Clickear en: **New -> Class**
En “Source folder” introducir: **Proyecto_ABM**
En “Package” introducir: **abm**
En “Name” introducir: **JConexion**
En “Modifiers” tildar: **(*) public**
En “Superclass” verificar: **java.lang.Object**
Tildar: **public static void main(String[] args)**
Tildar: **Constructors from superclass**
Tildar: **Inherited abstract methods**
Tildar: **Generate comments**
Clickear en: **Finish**

Editaremos el código fuente **JConexion.java** mediante el Editor para que luzca como se muestra a continuación:

```
/**
 *
 */
package abm;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

/**
 * @author ehv80
 *
 */
public class JConexion {

	/**
	 * ATRIBUTOS DE CLASE: static
	 * Uno para todos los objetos que se instancien
	 * de esta Clase.
	 */
	private static Connection conexion = null;
	private static Statement sentencia = null;
	private static String nombreClaseDriverJDBC = "com.mysql.jdbc.Driver";
	private static String urlBaseDeDatos =
	                                    "jdbc:mysql://127.0.0.1/Taller3";
	private static String usuario = "mysql";
	private static String password = "4dm1nsql";

	/* ATRIBUTOS DE CADA INSTANCIA DE LA CLASE */
	/* ? */

	/* MÉTODOS DE CLASE: static
	 * Uno para todos los objetos que se instancien
	 * de esta Clase.
	 */
```

```
/** Clase JConexion */
/**
 * Método de Clase: conectar
 * Sirve para conectar a la Base de Datos.
 *
 * @param String usuario
 * @param String contrasenia
 * @return void
 * @author ehv80
 */
public static void conectar()
{
	if( conexion == null )
	{
		try {
			Class.forName(nombreClaseDriverJDBC);
			conexion =
	DriverManager.getConnection(urlBaseDeDatos, usuario, password);
			sentencia = conexion.createStatement();
		}
		catch (SQLException exSQL) {
			System.out.println(
				"No se puede conectar a la Base de Datos..!");
			System.out.println("SQLException: " +
							exSQL.getSQLState());
			System.out.println("SQLState: " +
							exSQL.getSQLState());
			System.out.println("VendorError: " +
							exSQL.getErrorCode());

		}
		catch (ClassNotFoundException exClassNotFound) {
			// TODO Auto-generated catch block
			System.out.println(
				"No se encuentra el driver JDBC..!");
			exClassNotFound.printStackTrace();
		}
	}
}
```

```
/* Clase JConexion */
/**
 * Método de Clase: desconectar
 * Sirve para desconectar de la Base de Datos.
 *
 * @return void
 */
public static void desconectar()
{
	if( conexion != null )
	{
		try {
			sentencia.close();
		}
		catch (SQLException exSQL) {
			// TODO Auto-generated catch block
			System.out.println(
				"No se puede terminar la sentencia SQL..!");
			System.out.println("SQLException: " +
							exSQL.getMessage());
			System.out.println("SQLState: " +
							exSQL.getSQLState());
			System.out.println("VendorError: " +
							exSQL.getErrorCode());
			exSQL.printStackTrace();
		}
		try {
			conexion.close();
		} catch (SQLException exSQL) {
			// TODO Auto-generated catch block
			System.out.println(
	"No se puede cerrar la conexiÃ³n con la Base de Datos..!");
			System.out.println("SQLException: " +
							exSQL.getMessage());
			System.out.println("SQLState: " +
							exSQL.getSQLState());
			System.out.println("VendorError: " +
							exSQL.getErrorCode());
			exSQL.printStackTrace();
		}
		sentencia = null;
		conexion = null;
	}
}
```

```
/* Clase JConexion */
/**
 * Método de Clase: ejecutarSentencia
 * Ejecuta la sentencia en lenguaje SQL
 * que se pasa como argumento: operacion
 *
 * @param String operacion
 * @return void
 */
public static void ejecutarSentencia(String operacion)
{
      if(sentencia != null)
      {
            try {
                  sentencia.executeUpdate(operacion);
            } catch (SQLException exSQL) {
                  // TODO Auto-generated catch block
                  System.out.println(
            "No se puede ejecutar la sentencia SQL: " + operacion);
                  System.out.println("SQLException: " +
                                          exSQL.getMessage());
                  System.out.println("SQLState: " +
                                          exSQL.getSQLState());
                  System.out.println("VendorError: " +
                                          exSQL.getErrorCode());
                  exSQL.printStackTrace();
            }
      }
}

/**
 * Método de Clase: consulta
 * Sirve para obtener un objeto de la clase JResultadoConsulta
 * que tiene el resultado arrojado por una consulta en Lenguaje
 * SQL hecho sobre la Base de Datos.
 *
 * @param String consultaSQL
 * @return JResultadoConsulta
 */
public static JResultadoConsulta consulta(String consultaSQL)
{
      if(sentencia != null)
      {
            try {
                  return new JResultadoConsulta(
                        sentencia.executeQuery(consultaSQL) );
            }
            catch (SQLException exSQL) {
                  // TODO Auto-generated catch block
                  System.out.println("SQLException: " +
                                                exSQL.getMessage());
                  System.out.println("SQLState: " +
                                          exSQL.getSQLState());
                  System.out.println("VendorError: " +
                                          exSQL.getErrorCode());
                  exSQL.printStackTrace();
            }
      }
      //else
      return null;
}
```

```
	/* Clase JConexion */
	/**
	 * Constructor
	 */
	public JConexion() {
		// TODO Auto-generated constructor stub
		conexion = null;
		sentencia = null;
		usuario = "mysql";
		nombreClaseDriverJDBC = "com.mysql.jdbc.Driver";
		urlBaseDeDatos = "jdbc:mysql://127.0.0.1/Taller3";
		password = "4dm1nsql";
	}

	/**
	 * @param args
	 */
	public static void main(String[] args) {
		// TODO Auto-generated method stub

	}

}
```

Como puede observarse en el código fuente de **JConexion.java** hay una referencia a una clase que aún no hemos creado en nuestro proyecto, por lo tanto vamos a crearla a continuación:

Para agregar la clase, no Visual, **JresultadoConsulta.java**
Dirigir el mouse hacia la solapa **"Package Explorer"**

Clickear con botón derecho del mouse en:	**New -> Class**
En "Source folder" verificar que muestre:	**Proyecto_ABM**
En "Package" introducir:	**abm**
En "Name" introducir.	**JResultadoConsulta**
En "Modifiers" tildar:	**(*) public**
En "Superclass" verificar que muestre:	**java.lang.Object**
Tildar:	**public static void main(String[] args)**
Tildar:	**Constructors from Superclass**
Tildar:	**Inherited abstract methods**
Tildar:	**Generate comments**

Editaremos el código fuente de esta clase **JResultadoConsulta.java** para que quede como se muestra a continuación:

```
/**
 *
 */
package abm;

import java.sql.ResultSet;
import java.util.Vector;
import javax.swing.JTable;
```

```java
/* Clase JResultadoConsulta */
/**
 * @author ehv80
 *
 */
public class JResultadoConsulta {

	/* ATRIBUTOS DE CLASE: static */

	/* ATRIBUTOS DE INSTANCIA */
	// Un Vector de Vector de String
	// es como una matriz de String y redimensionable
	private Vector <Vector <String>> datos = null;
	//aquí tuve que cambiar el "project compliance" a 5.0

	private Vector <String> nombresColumnas = null;

	/* MÉTODO DE CLASE: static */

	/* MÉTODO DE INSTANCIA */

	/**
	 * Método: getTabla
	 *
	 * Sirve para obtener una JTable con el Resultado de la
	 * Consulta SQL.
	 *
	 * @return JTable
	 */
	public JTable getTabla()
	{
		return new JTable(datos, nombresColumnas);
	}

	/**
	 * Método: getFila
	 * Sirve para obtener una fila (Vector de String)
	 * del Vector de Vector de String: datos.
	 * Esto representa un registro de la Tabla de la Base de Datos.
	 *
	 * @param int indice
	 * @return Vector <String>
	 */
	public Vector <String> getFila(int indice)
	{
		return datos.get(indice);
	}

	/** Constructor de la clase JResultadoConsulta
	 * @param ResultSet conjuntoResultado
	 */
	public JResultadoConsulta(ResultSet conjuntoResultado) {
		// TODO Auto-generated constructor stub
		datos = JResultSetToVector.getDatos(conjuntoResultado);
		nombresColumnas =
			JResultSetToVector.getColumnas(conjuntoResultado);
	}

	/** @param args    */
	public static void main(String[] args) {
		// TODO Auto-generated method stub
	}

}
```

Como se pude observar el código fuente de la clase **JResultadoConsulta.java** tiene referencias a una clase llamada **JResultSetToVector** que todavía no hemos creado en nuestro proyecto. Esta clase se encargará de manejar los datos que se obtienen de las consultas en Lenguaje SQL usando la **interface**: **java.sql.ResultSet**

Para agregar la clase **JResultSetToVector** a nuestro proyecto:
Dirigir el mouse hacia la solapa **"Package Explorer"**

Clickear con botón derecho del mouse en:	**New** -> **Class**
En "Source folder" verificar que muestre:	**Proyecto_ABM**
En "Package" introducir:	**abm**
En "Name" introducir.	**JResultSetToVector**
En "Modifiers" tildar:	**(*) public**
En "Superclass" verificar que muestre:	**java.lang.Object**
Tildar:	**public static void main(String[] args)**
Tildar:	**Constructors from Superclass**
Tildar:	**Inherited abstract methods**
Tildar:	**Generate comments**

Editremos el código fuente de esta clase **JResultSetToVector.java** para que contenga las siguientes líneas:

```
/* Clase JResultSetToVector */
/**
 *
 */
package abm;


import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.util.Vector;


/**
 * @author ehv80
 *
 */
public class JResultSetToVector {

	/* ATRIBUTOS DE CLASE: static */
	/* ATRIBUTOS DE INSTANCIA */
	/* MÉTODOS DE CLASE: static */
```

```
/* Clase JResultSetToVector */
      public static Vector <Vector <String>> getDatos(ResultSet
                                                      conjuntoResultado)
      {
            Vector <Vector <String>> datos = new Vector <Vector <String>>();
            try {
                  ResultSetMetaData metadataConjuntoResultado =
                                    conjuntoResultado.getMetaData();
                  Vector <String> vectorTemp = null;
                  int numeroColumnas =
                        metadataConjuntoResultado.getColumnCount();
                  while(conjuntoResultado.next())
                  {
                        vectorTemp = new Vector <String>();
                        for(int i = 1 ; i <= numeroColumnas ; i++ )
                        {
                        vectorTemp.add(conjuntoResultado.getString(i));
                        }
                        datos.add(vectorTemp);
                  }
                  return datos;
            }
            catch (SQLException exSQL)
            {
                  System.out.println(
                  "Error al obtener el resultado de la consulta SQL..!");
                  System.out.println("SQLException: " + exSQL.getMessage());
                  System.out.println("SQLState: " + exSQL.getSQLState());
                  System.out.println("VendorError: " +
                                                exSQL.getErrorCode());
                  return null;
            }
      }

      public static Vector <String> getColumnas(ResultSet conjuntoResultado)
      {
            Vector <String> columnas = new Vector <String>();
            try {
                  ResultSetMetaData metadataConjuntoResultado =
                                          conjuntoResultado.getMetaData();
                  int numeroColumnas =
                              metadataConjuntoResultado.getColumnCount();
                  for(int i = 1 ; i <= numeroColumnas ; i++)
                  {
                  columnas.add(metadataConjuntoResultado.getColumnName(i));
                  }
                  return columnas;
            }
            catch (SQLException exSQL)
            {
                  System.out.println(
                  "Error al obtener el resultado de la consulta SQL..!");
                  System.out.println("SQLException: " + exSQL.getMessage());
                  System.out.println("SQLState: " + exSQL.getSQLState());
                  System.out.println("VendorError: " +
                                                exSQL.getErrorCode());
                  return null;
            }
      }

      /* MÉTODOS DE INSTANCIA */
```

```
/* Clase JResultSetToVector */
	/**
	 * Constructor:
	 */
	public JResultSetToVector() {
		// TODO Auto-generated constructor stub
	}

	/**
	 * @param args
	 */
	public static void main(String[] args) {
		// TODO Auto-generated method stub

	}

}
```

La última clase que tenemos que agregar a nuestro proyecto la llamaremos **JVentanaSeleccion.java**. Será una clase Visual cuya ventana aparecerá cuando el usuario, en frente de la ventana de **Modificaciones** pulse el **"botón" (JButton) "Buscar"**, entonces se hará una consulta SQL **"select * from ABM"** que traerá a una **"Tabla" (JTable)** el contenido de la **Tabla: ABM** de la **Base de Datos: Taller3**, así el usuario puede elegir con un click del mouse un "registro" para modificarlo. Luego de hacer el click esta ventana debería ocultarse y volver a la ventana de Modificaciones.

Para agregar la clase **JVentanaSeleccion.java** a nuestro proyecto:
Dirigir el mouse hacia la solapa **"Package Explorer"**
Clickear con botón derecho del mouse en: **New** -> **Visual Class**
En "Source folder" verificar que muestre: **Proyecto_ABM**
En "Package" introducir: **abm**
En "Name" introducir. **JVentanaSeleccion**
En "Modifiers" tildar: **(*) public**
En "Superclass" verificar: **javax.swing.JInternalFrame**
Tildar: **public static void main(String[] args)**
Tildar: **Constructors from Superclass**
Tildar: **Inherited abstract methods**
Tildar: **Generate comments**

Como en otras oportunidades anteriores vamos a comenzar a configurar los componentes y propiedades que tendrá esta clase Visual **JVentanaSeleccion.java.**

Para configurar el título de la ventana de **JVentanaSeleccion.java:**
Dirigir el mouse hacia la solapa **"Java Beans"**
Clickear con botón derecho del mouse en: **this-""**
Clickear en **"Set Title"** e introducir: **Ventana de Selección**

Para configurar el nombre del **jContentPane:**
Dirigir el mouse hacia la solapa **"Java Beans"**
Clickear con botón derecho del mouse en: **jContentPane**
Clickear en **"Rename Field"** e introducir: **jContentPaneVentanaSeleccion**

Para agregar un **“panel con barras deslizables” (JScrollPane):**
Dirigir el mouse hacia el área del Visual Editor y clickear en: **Palette**
Clickear en: **Swing Containers**
Clickear en: **JscrollPane**
Clickear en: **jContentPaneVentanaSeleccion**

Para cambiar el nombre del **“panel con barras deslizables” (JScrollPane)** recién agregado:
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear con botón derecho del mouse en: **jScrollPane**
Clickear en **“Rename Field”** e introducir: **jScrollPaneVentanaSeleccion**

Para configurar algunas de las propiedades del mismo **“panel con barras deslizables” (JScrollPane):**
Dirigir el mouse hacia la solapa **“Properties”**
Clickear en **“componentOrientation”** y elegir: **LEFT_TO_RIGHT**
Clickear en **“constraint”** y elegir: **Center**
Clickear en **“>preferredSize”** e introducir: **100,100**

Este **“panel con barras deslizables” (JScrollPane) jScrollPaneVentanaSeleccion** va a contener el resultado de la consulta SQL hecha sobre la **Tabla** de la Base de Datos, que será exhibido dentro de una **JTable.**

Para adicionar una **JTable** dentro del **“panel con barras deslizables” (JScrollPane) jScrollPaneVentanaSeleccion:**
Dirigir el mouse hacia el área del Editor Visual y clickear en: **Palette**
Clickear en: **JTable**
Clickear en: **jScrollPaneVentanaSeleccion**

Primero modificaremos el nombre de la **“Tabla” (JTable)**, para ello:
Dirigir el mouse hacia el solapa **“Java Beans”**
Clickear con botón derecho del mouse en: **jTable**
Clickear en **“Rename Field”** e introducir: **jTableVentanaSeleccion**

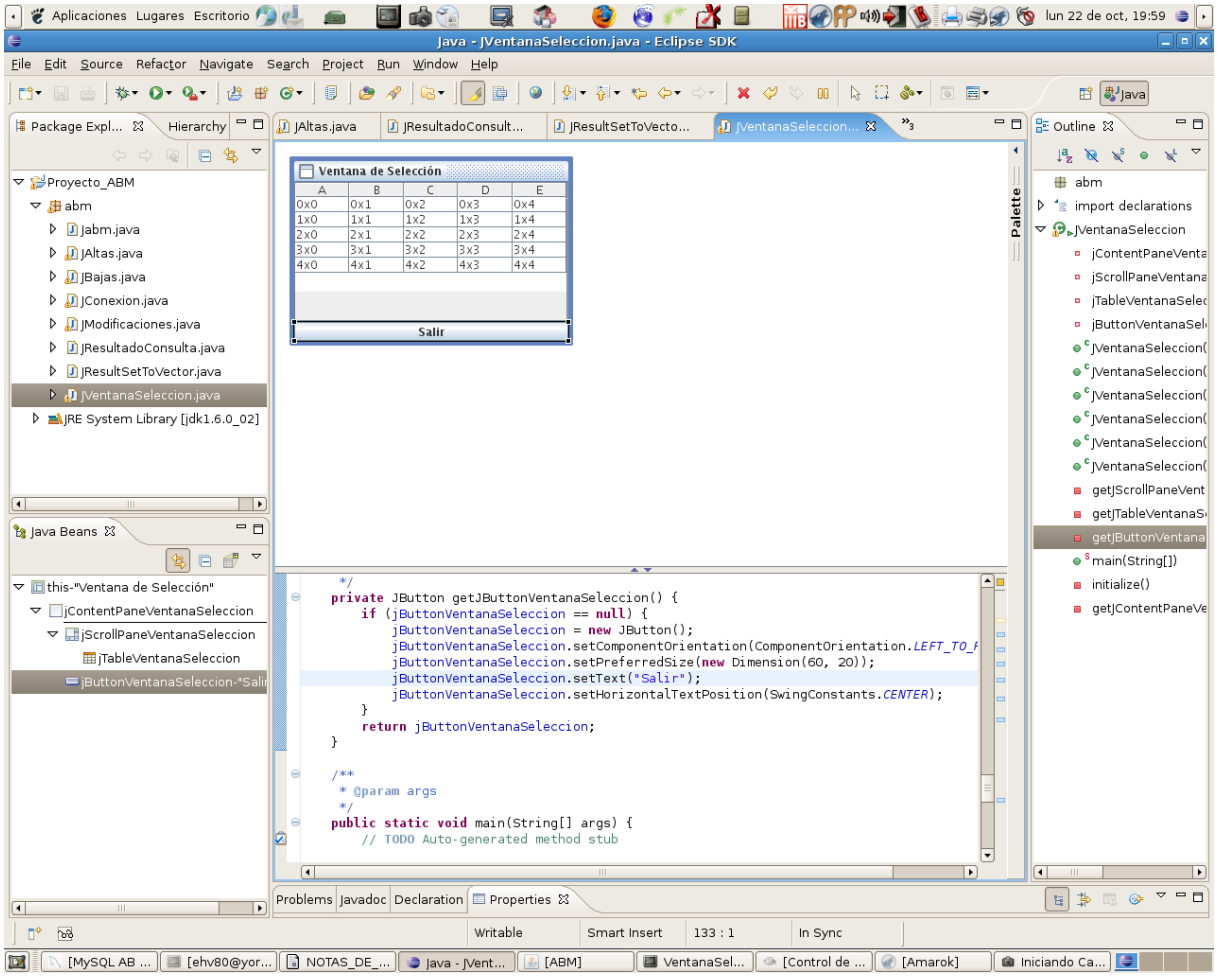
Para configurar algunas de las propiedades de la **“Tabla” (JTable)** recién agregada:
Dirigir el mouse hacia la solapa **“Properties”:**
Clickear en **“componentOrientation”** y elegir: **LEFT_TO_RIGHT**
Clickear en **“constraint”** y elegir: **Center**
Clickear en **“>preferredSize”** e introducir: **100,100**
Clickear en **“size”** e introducir: **100,100**

Ahora agregaremos un **“botón” (JButton)** dentro del **jContentPaneVentanaSeleccion**, para ello:
Dirigir el mouse hacia el área del Editor Visual y clickear en: **Palette**
Clickear en: **Swing Components**
Clickear en: **JButton**
Clickerar en: **jContentPaneVentanaSeleccion**

Para cambiar el nombre del **“botón” (JButton):**
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickearn con botón derecho del mouse sobre: **jButton**
Clickear en **“Rename Field”** e introducir: **jButtonVentanaSeleccion**

Para configurar algunas de las propiedades del **“botón” (JButton)** recién agregado:
Dirigir el mouse hacia la solapa **“Properties”**
Clickear en **“componentOrientation”** y elegir: **LEFT_TO_RIGHT**
Clickear en **“constraint”** y elegir: **South**
Clickear en **“horizontalTextPosition”** y elegir: **CENTER**
Clickear en **“>preferredSize”** e introducir: **60,20**
Clickear en **“text”** e introducir: **Salir**

Con estas propiedades configuradas la ventana que muestra la clase **JventanaSeleccion** debería verse como se muestra a continuación:



Ahora vamos a comenzar a configurar los aspectos referidos al comportamiento de nuestra aplicación con respecto a la interacción del usuario y los posibles “eventos” que éste pueda generar al usar la aplicación.

Empezaremos por la primera clase que hemos escrito en nuestro proyecto: **Jabm.java**

Para configurar el **“evento de cerrar la ventana”** que muestra la clase **Jabm**:

Dirigir el mouse hacia la solapa **“Package Explorer”**
Clickear dos veces en **Jabm.java**
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear con botón derecho del mouse sobre: **this-”ABM”**
Clickear en: **Events -> windowClosing**
Clickear en **windowClosing**
Dirigir el mouse hacia el área del Editor de código fuente:
Comentar la línea que contiene:

```
System.out.println("windowClosing()");
```

Agregar en su lugar el siguiente código, que sirve para capturar el evento de clickear en el botón que cierra la ventana “[X]”:

```
dispose();
```

Para configurar el evento de presionar una tecla que causa la apertura del **“menú desplegable” (JMenu) “Acciones”:**
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear en: **JMenu-”Acciones”**
Dirigir el mouse hacia la solapa **“Properties”**
Clickear en **“mnemonic”** y luego en su botón correspondiente: [...]
De la lista que se muestra elegir **“HOME”** y clickear en **“OK”**

Para configurar el evento de presionar una tecla que causa la apertura de la **“ventana interna”** asociada al **“ítem de menú” (JMenuItem) “Altas”:**
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear en: **JMenuItem-”Altas”**
Dirigir el mouse hacia la solapa **“Properties”**
Clickear en **“mnemonic”** y luego en su botón correspondiente: [...]
De la lista que se muestra elegir **“A”** y clickear en **“OK”**

Para configurar el evento de presionar una tecla que causa la apertura de la **“ventana interna”** asociada al **“ítem de menú” (JMenuItem) “Bajas”:**
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear en: **JMenuItem-”Bajas”**
Dirigir el mouse hacia la solapa **“Properties”**
Clickear en **“mnemonic”** y luego en su botón correspondiente: [...]
De la lista que se muestra elegir **“B”** y clickear en **“OK”**

Para configurar el evento de presionar una tecla que causa la apertura de la **“ventana interna”** asociada al **“ítem de menú” (JMenuItem) “Modificaciones”:**
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear en: **JMenuItem-”Modificaciones”**
Dirigir el mouse hacia la solapa **“Properties”**
Clickear en **“mnemonic”** y luego en su botón correspondiente: [...]
De la lista que se muestra elegir **“M”** y clickear en **“OK”**

Continuando con la clase Visual: **Jabm.java**, vamos agregar un **“ítem de menú” (JMenuItem)** al **“menú desplegable” (JMenu)** que ofrezca al usuario la posibilidad de salir de la ventana que muestra la aplicación:

Para agregar un nuevo **“ítem de menú” (JMenuItem)** a la clase Visual **Jabm.java:**
Dirigir el mouse hacia el área del Editor Visual y clickear en: **Palette**
Clickear en: **Swing Menus**
Clickear en: **JMenuItem**
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear en: **jMenu-”Acciones”**

Para configurar algunas de las propiedades del **JMenuItem** recién agregado:
Dirigir el mouse hacia la solapa **“Java Beans”**y clickear en: **jMenuItem3**
Clickear en **“mnemonic”** y luego en su botón correspondiente: [...]
de la lista que se muestra elegir **“S”** y clickear en **“OK”**
Clickear en **“name”** y elegir: **Salir**
Clickear en **“text”** y elegir: **Salir**

Para configurar la captura del evento de cerrar la ventana al clickear sobre el **“ítem de menú” (JMenuItem) jMenuItem3-”Salir”:**
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear con botón derecho del mouse en: **jMenuItem3-”Salir”**
Del menú que aparece clickear en: **Events -> actionPerformed**
y editar el código fuente de la clase visual **Jabm.java** para que quede esa sección de la siguiente manera:

```
jMenuItem3.addActionListener(new java.awt.event.ActionListener() {
  public void actionPerformed(java.awt.event.ActionEvent e) {
        //System.out.println("actionPerformed()");
        // TODO Auto-generated Event stub actionPerformed()
        dispose();
  }
```

Lo que sigue es configurar los aspectos referidos a la captura de eventos con respecto a la ventana que muestra la clase Visual **JAltas.java**, para ello vamos dirigir el mouse hacia el área del editor visual y clickear en la solapa del código fuente **JAltas.java** entonces proseguimos como se detalla a continuación:

Para configurar algunas de las propiedades de la clase visual **JAltas.java**
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear en: **this-”Altas”**
Dirigir el mouse hacia la solapa **“Properties”**
Clickear en **“closable”** y elegir: **true**

Para configurar el evento asociado al **“botón” (JButton) jButtonAltasGrabar-”Grabar”**:

Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear con botón derecho del mouse en:
“**jButtonAltasGrabar-”Grabar”**

Del menú que aparece clickear en: **Events -> actionPerformed**
Comentar la línea de código fuente que contiene:

```
System.out.println("actionPerformed()");
```

y reemplazar por el siguiente código fuente:

```
JConexion.conectar();
Jconexion.ejecutarSentencia(
   "INSERT INTO ABM VALUES('"+
         jTextFieldAltasNombre.getText() +"','"+
         jTextFieldAltasApellido.getText() +"','"+
         (String)jComboBoxAltas.getSelectedItem() + "','"+
         jTextFieldAltasDocumento.getText() +"','"+
         jTextFieldAltasEdad.getText() +"')");
Jconexion.desconectar();
```

Para configurar el evento asociado al **“botón” (JButton) jButtonAltasSalir-”Salir”**:

Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear con botón derecho del mouse en:
jButtonAltasSalir-”Salir”

Del menú que aparece clickear en: **Events -> actionPerformed**
Comentar la línea de código fuente que contiene:

```
System.out.println("actionPerformed()");
```

y reemplazar por el siguiente código fuente:

```
dispose();
```

Ahora continuaremos con otros eventos y propiedades para la clase visual **JBajas.java**:
Dirigir el mouse hacia la solapa **“Java Beans”**
Clickear en: **this-”Bajas”**
Dirigir el mouse hacia la solapa **“Properties”**
Clickear en **“closable”** y elegir: **true**

Para configurar el evento asociado a la acción de presionar el **“botón” (JButton) jButtonBajasBorrar-”Borrar”:**
Dirigir el mouse hacia la solapa **“Java Beans”**

Clickear con botón derecho del mouse en: **jButtonBajasBorrar-”Borrar”**

Del menú que aparece clickear en: **Event -> actionPerformed**
Reemplazar el código por:

```
jButtonBajasBorrar.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent e) {
              //System.out.println("actionPerformed()");
              // TODO Auto-generated Event stub actionPerformed()
              JConexion.conectar();
              Jconexion.ejecutarSentencia(
                    "DELETE FROM ABM where ndoc =" +
                    jTextFieldBajasDocumento.getText());
              JConexion.desconectar();
        }
  });
```

En la clase visual **JBajas.java** para configurar el evento asociado a presionar el **“botón” (JButton) jButtonBajasSalir-”Salir”:**
Dirigir el mouse hacia la solapa **“Java Beans”**

Clickear con botón derecho del mouse en: **jButtonBajasSalir-”Salir”**

Del menú desplegable clickear en: **Events -> actionPerformed**
Reemplazar el código fuente por:

```
jButtonBajasSalir.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent e) {
              //System.out.println("actionPerformed()");
              // TODO Auto-generated Event stub actionPerformed()
              dispose();
        }
  });
```

En la clase visual **JBajas.java** para configurar el evento asociado a presionar el **“botón” (JButton) jButtonBajasBuscar-”Buscar”:**
Dirigir el mouse hacia la solapa **“Java Beans”**

Clickear con botón derecho del mouse en:
(JButton) jButtonBajasBuscar-”Buscar”

Del menú desplegable clickear en: **Events -> actionPerformed**
Reemplazar el código fuente por:

```
jButtonBajasBuscar.addActionListener(
      new java.awt.event.ActionListener() {
            public void actionPerformed(
                  java.awt.event.ActionEvent e) {
                  //System.out.println("actionPerformed()");
                  // TODO Auto-generated Event stub actionPerformed()
                  String consulta = "SELECT * FROM ABM ";
                  if(!(jTextFieldBajasNombre.getText().equals(""))) {
                        consulta += "WHERE nombre='" +
                              jTextFieldBajasNombre.getText()+"'";
                  }
                  else if( !(jTextFieldBajasApellido.getText().equals(""))){
                        consulta += "WHERE apellido='" +
                              jTextFieldBajasApellido.getText()+"'";
                  }
                  else if(!(jTextFieldBajasDocumento.getText().equals(""))){
                        consulta += "WHERE ndoc='" +
                              jTextFieldBajasDocumento.getText()+"'";
                  }
                  else if(!(jTextFieldBajasEdad.getText().equals(""))){
                  consulta += "WHERE edad='" +
                        jTextFieldBajasEdad.getText()+"'";
                  }
                  JConexion.conectar();
                  JResultadoConsulta resultadoConsulta =
                                    JConexion.consulta(consulta);
                  JVentanaSeleccion ventanaSeleccion = new
                    JVentanaSeleccion(resultadoConsulta, getEstaVentana() );
                  jDesktopPaneBajas.add(ventanaSeleccion);
                  ventanaSeleccion.setVisible(true);
                  Jconexion.desconectar();
                  setVisible(true);
            }
      });
```

En la clase Visual **JBajas.java** para agregar el método **“getEstaVentana”** que retorna “**this**”:
Dirigir el mouse hacia el área del editor de código fuente:
Editar el siguiente código fuente:

```
/** Método: getEstaVentana
 * Retorna la ventana que está en ejecución actualmente.
 * @return JBajas this
 */
public JBajas getEstaVentana(){
      return this;
}
```

Para agregar un nuevo método **“cargaDatos”** necesario para la clase visual **JBajas.java**:

Dirigir el mouse hacia el área del editor de código fuente:

Editar el siguiente tramo de código:

```
/** Método: cargaDatos
 * Toma los datos que están almacenados temporalmente
 * en el Vector de String "datos" y los introduce e los
 * campos de texto correspondientes de la interface del
 * objeto instancia de la clase visual JBajas.java.
 *
 * @param Vector <String> datos
 * @return void
 */
public void cargaDatos(Vector <String> datos){
      setVisible(true);
      jTextFieldBajasNombre.setText(datos.get(0));
      jTextFieldBajasApellido.setText(datos.get(1));
      jComboBoxBajas.setSelectedItem(datos.get(2));
      jTextFieldBajasDocumento.setText(datos.get(3));
      jTextFieldBajasEdad.setText(datos.get(4));
}
```

Se observará que el editor de código fuente del Eclipse muestra un error al que en estos momentos ignoraremos, pues se debe a la falta de código fuente en la clase visual **JVentanaSeleccion.java** que modificaremos más adelante.

Ahora pasaremos a configurar algunas propiedades y aspectos referidos a los eventos relacionados con la clase visual **JModificaciones.java.**

Para configurar la captura del evento asociado a presionar el **“botón” (JButton) jButtonModificacionesGrabar-”Grabar”** de la clase visual **JModificaciones.java:**

Dirigir el mouse hacia el área del editor de código fuente y clickear en la solapa **“JModificaciones.java”**
Dirigir el mouse hacia la solpa **“Java Beans”**
Clickear con botón derecho del mouse en:
jButtonModificacionesGrabar-”Grabar”

Del menú que aparece clickear en: **Events -> actionPerformed**

Reemplazar el código fuente que aparece por el siguiente:

```
jButtonModificacionesGrabar.addActionListener(
      new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent e) {
                  //System.out.println("actionPerformed()");
                  // TODO Auto-generated Event stub actionPerformed()
                  JConexion.conectar();
                  JConexion.ejecutarSentencia("UPDATE ABM SET nombre='" +
                        jTextFieldModificacionesNombre.getText() +
                        "' , apellido='" +
                        jTextFieldModificacionesApellido.getText() +
                        "' , tpdoc='" +
                        (String)jComboBoxModificaciones.getSelectedItem() +
                        "' , ndoc='" +
                        jTextFieldModificacionesDocumento.getText() +
                        "' , edad='" +
                        jTextFieldModificacionesEdad.getText() + "'" +
                        "WHERE ndoc=" +
                        jTextFieldModificacionesDocumento.getText() );
                  JConexion.desconectar();
            }
      });
}
```

Para configurar la captura del evento asociado a presionar el **“botón” (JButton) jButtonModificacionesSalir-”Salir”** de la clase visual **JModificaciones.java:**
Dirigir el mouse hacia el área del editor de código fuente y clickear en la solapa **“JModificaciones.java”**

Dirigir el mouse hacia la solpa **“Java Beans”**
Clickear con botón derecho del mouse en:
jButtonModificacionesSalir-”Salir”
Del menú que aparece clickear en: **Events -> actionPerformed**
Reemplazar el código fuente que aparece por el siguiente:

```
jButtonModificacionesSalir.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent e) {
                  //System.out.println("actionPerformed()");
                  // TODO Auto-generated Event stub actionPerformed()
                  dispose();
            }
      });
}
```

Para configurar la captura del evento asociado a presionar el **“botón” (JButton) jButtonModificacionesBuscar-”Buscar”** de la clase visual **JModificaciones.java:**

Dirigir el mouse hacia el área del editor de código fuente y clickear en la solapa **“JModificaciones.java”**

Dirigir el mouse hacia la solpa **“Java Beans”**
Clickear con botón derecho del mouse en:
jButtonModificacionesBuscar-”Buscar”

Del menú que aparece clickear en: **Events -> actionPerformed**
Reemplazar el código fuente que aparece por el siguiente:

```
jButtonModificacionesBuscar.addActionListener(
      new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent e) {
                  //System.out.println("actionPerformed()");
                  // TODO Auto-generated Event stub actionPerformed()
                  String consulta = "SELECT * FROM ABM";
                  if(!(jTextFieldModificacionesNombre.getText().equals("")))
                  {
                        consulta += "WHERE nombre='" +
                        jTextFieldModificacionesNombre.getText() + "'";
                  }
                  else if(!(jTextFieldModificacionesApellido.getText().
                                                      equals("")))
                  {
                        consulta += "WHERE apellido='" +
                        jTextFieldModificacionesApellido.getText() + "'";
                  }
                  else if(!(jTextFieldModificacionesDocumento.getText().
                                                      equals("")))
                  {
                        consulta += "WHERE ndoc='" +
                        jTextFieldModificacionesDocumento.getText()+"'";
                  }
                  else if(!(jTextFieldModificacionesEdad.getText()
                                                      .equals("")))
                  {
                        consulta += "WHERE edad='" +
                              jTextFieldModificacionesEdad.getText() + "'";
                  }
                  JConexion.conectar();
                  JResultadoConsulta resultadoConsulta =
                                          JConexion.consulta(consulta);
                  JVentanaSeleccion ventanaSeleccion =
                              new JventanaSeleccion(
                                    resultadoConsulta,
                                    getEstaVentana()
                              );
                  jDesktopPaneModificaciones.add(ventanaSeleccion);
                  ventanaSeleccion.setVisible(true);
                  JConexion.desconectar();
                  setVisible(true);
            }
      });
}
```

Además deberemos agregar a la clase visual **JModificaciones.java** el siguiente método mediante el editor de código fuente:

```
/** Método: getEstaVentana
 * @return JModificaciones this
 */
public JModificaciones getEstaVentana() {
	return this;
}
```

También tendremos que agregar el siguiente método a la clase visual **JModificaciones.java** mediante el editor de código fuente:

```
/**
 * Método: cargaDatos
 *
 * @param Vector <String> datos
 * @return void
 */
public void cargaDatos(Vector <String> datos) {
	setVisible(true);
	jTextFieldModificacionesNombre.setText(datos.get(0));
	jTextFieldModificacionesApellido.setText(datos.get(1));
	jComboBoxModificaciones.setSelectedItem(datos.get(2));
	jTextFieldModificacionesDocumento.setText(datos.get(3));
	jTextFieldModificacionesEdad.setText(datos.get(4));
}
```

Se observarán errores en los códigos fuentes de los siguientes archivos:

- **JBajas.java**
- **JModificaciones.java**

Para corregirlos deberemos agregar a la clase **JVentanaSeleccion.java** dos constructores adicionales:

Dirigir el mouse hacia la solapa **“Package Explorer”**

Clickear en: **JVentanaSeleccion.java**

Dirigir el mouse hacia el área del editor de código fuente y editar la siguiente porción de código:

```
/** Constructor adicional:
 * @param JResultadoConsulta resultadoConsulta
 * @param JModficaciones modificaciones
 */
public JventanaSeleccion(
            JResultadoConsulta resultadoConsulta,
            JModificaciones modificaciones
      )
{
      super();
      initialize();
      this.resultadoConsulta = resultadoConsulta;
      this.modificaciones = modificaciones;
      jTableVentanaSeleccion = resultadoConsulta.getTabla();
      jTableVentanaSeleccion.addMouseListener(
            new java.awt.event.MouseAdapter() {
                  public void mouseClicked(
                        java.awt.event.MouseEvent e
                  )
                  {
                        getModificaciones().cargaDatos(
                              getResultadoConsulta().
                                    getFila(jTableVentanaSeleccion.
                                          getSelectedRow()
                        ));
                        setVisible(false);
                  }
            });

      jScrollPaneVentanaSeleccion.
            setViewportView(jTableVentanaSeleccion);
}
```

```
/** Constructor adicional:
 * @param JResultadoconsulta resultadoConsulta
 * @param JBajas bajas
 */
public JventanaSeleccion(
      JResultadoConsulta resultadoConsulta,
      JBajas bajas
)
{
      super();
      initialize();
      this.resultadoConsulta = resultadoConsulta;
      this.bajas = bajas;
      jTableVentanaSeleccion = resultadoConsulta.getTabla();
      jTableVentanaSeleccion.addMouseListener(new
      java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent e)
            {
                  getBajas().
                        cargaDatos(getResultadoConsulta().
                              getFila(
                                    jTableVentanaSeleccion.
                                          getSelectedRow()
                              )
                  );
                  setVisible(false);
            }
      });
      jScrollPaneVentanaSeleccion.
            setViewportView(jTableVentanaSeleccion);
}
```

Además se deberán agregar a la clase visual **JVentanaSeleccion.java** los atributos siguientes:

- `private JResultadoConsulta resultadoConsulta;`
- `private JModificaciones modificaciones;`
- `private JBajas bajas;`

También deberemos agregar a la clase visual **JVentanaSeleccion.java** algunos métodos más, con el código fuente que se muestra a continuación:

```
/** Método: getResultadoConsulta
 * @param void
 * @return JResultadoConsulta resultadoConsulta
 */
public JResultadoConsulta getResultadoConsulta() {
	return this.resultadoConsulta;
}

/** Método: getModificaciones
 * @param void
 * @return JModificaciones modificaciones
 */
public JModificaciones getModificaciones() {
	return this.modificaciones;
}

/** Método: getBajas
 * @param void
 * @return JBajas bajas
 */
public JBajas getBajas()
{
	return this.bajas;
}
```

Falta configurar la captura del evento asociado con presionar el "botón" **(JButton) jButtonVentanaSeleccion-"Salir"** de la clase visual **JVentanaSeleccion.java**:

Dirigir el mouse hacia la solapa **"Java Beans"**

Clickear con botón derecho del mouse en: **jButtonVentanaSeleccion-"Salir"**

Del menú que emerge, clickear en: **Events -> actionPerformed**
Reemplazar el código fuente por el siguiente:

```
jButtonVentanaSeleccion.addActionListener(
	new java.awt.event.ActionListener() {
		public void actionPerformed(java.awt.event.ActionEvent e) {
			//System.out.println("actionPerformed()");
			// TODO Auto-generated Event stub actionPerformed()
			dispose();
		}
	}
);
```

Algo que nos quedó en el tintero es configurar la siguiente propiedad en la clase visual **JModificaciones.java:**
Dirigir el mouse hacia la solapa **"Package Explorer"**:
Clickear en: **JModificaciones.java**
Dirigir el mouse hacia la solapa **"Java Beans":**
Clickear en: **this-"Modificaciones"**
Dirigir el mouse hacia la solapa **"Properties"**
Clickear en **"closable"** y elegir: **true**

También para esta clase visual **JModificaciones.java** debemos agregar el siguiente atributo:

```
private static final long serialVersionUID = 1L;
```

Otra cosa que tendremos que agregar, pero a la clase visual **JVentanaSeleccion.java** es el siguiente atributo:

```
private static final long serialVersionUID = 1L;
```

Nuevamente, pero esta vez para la clase visual **JBajas.java** tendremos que agregar también el atributo:

```
private static final long serialVersionUID = 1L;
```

Lo mismo tenemos que hacer para la clase visual **JAltas.java**:

```
private static final long serialVersionUID = 1L;
```

Ahora veremos como incorporar al proyecto: **Proyecto_ABM** el **driver JDBC** llamado **MySQL Connector/J** que servirá **para realizar las conexiones a la base de datos** que debe configurarse en el **Sistema Gestor de Base de Datos MySQL:**

mysql-standard-5.0.24a-linux-i686.

Para incorporar el **driver JDBC** al proyecto: **Proyecto_ABM**
Dirigir el mouse hacia la solapa **“Package Explorer”**
Clickear con botón derecho del mouse en: **Proyecto_ABM**
En el menú clickear en: **Build Path -> Add External Archives...**
En el examinador de archivos elegir la ubicación en donde está el archivo: **mysql-connector-java-5.0.3-bin.jar**
Clickear en: **Aceptar**

Ahora se puede hacer una primera prueba de ejecución mediante:
Run -> Run as -> Java Application
clickear en: **Jabm.java**
y luego en: **OK**

De esta primera prueba de ejecución observamos que nuestra aplicación puede realizar la incorporación de datos a la tabla de la base de datos, la obtención de los datos a través de la **“Ventana de Selección”** y la modificación de los mismos. Esta “**Ventana de Selección”** carece del botón de cerrar la ventana.

Para agregar la propiedad de poder cerrar la ventana a partir del clásico botón cerrar **[X]**:
Dirigir el mouse hacia la solapa **“Package Explorer”**:
Clickear en: **JVentanaSeleccion.java**
Dirigir el mouse hacia la solapa **“Java Beans”**:
Clickear en: **this-”VentanaSelección”**
Dirigir el mouse hacia la solapa **“Properties”**
Clickear en **“closable”** y elegir: **true**

Para mejorar el comportamiento, vamos a tener que modificar el código fuente de las clases visuales:

- **JBajas.java**
- **JModificaciones.java**
- **JVentanaSeleccion.java**

A continuación listaremos los cambios que haremos a la clase visual **JBajas.java**, para esto dirigir el mouse hacia la solapa **"Package Explorer"** y clickear en **JBajas.java:**

- Antes que nada vamos a eliminar el texto sugerido que presenta cada **"campo de texto" (JTextfield)**
  - Dirigir el mouse hacia la solapa **"Java Beans"**
    Clickear en: **jTexFieldBajasNombre**
    Dirigir el mouse hacia la solapa **"Properties"**
    Clickear en **">text"** y borrar su contenido.
  - Dirigir el mouse hacia la solapa **"Java Beans"**
    Clickear en: **jTexFieldBajasApellido**
    Dirigir el mouse hacia la solapa **"Properties"**
    Clickear en **">text"** y borrar su contenido.
  - Dirigir el mouse hacia la solapa **"Java Beans"**
    Clickear en: **jTexFieldBajasDocumento**
    Dirigir el mouse hacia la solapa **"Properties"**
    Clickear en **">text"** y borrar su contenido.
  - Dirigir el mouse hacia la solapa **"Java Beans"**
    Clickear en: **jTexFieldBajasEdad**
    Dirigir el mouse hacia la solapa **"Properties"**
    Clickear en **">text"** y borrar su contenido.
- Para mejorar la visibilidad de los datos al realizar una **"búsqueda"**, vamos a configurar las siguientes propiedades.
  Primero dirigir el mouse hacia la solapa: **"Java Beans"**
  Clickear en: **this-"Bajas"**
  Luego dirigir el mouse hacia la solapa: **"Properties"**:
  - clickear en **"componentOrientation"**
    Elegir: **"LEFT_TO_RIGHT"**
  - clickear en **"maximizable"** y elegir: **"true"**
  - clickear en **"resizable"** y elegir: **"true"**

Ahora listaremos los cambios que haremos a la clase visual **JModificaciones.java**, para esto

Dirigir el mouse hacia la solapa: **"Package Explorer"**
Clickear en: **JModificaciones.java**

- Antes que nada vamos a eliminar el texto sugerido que presenta cada **"campo de texto" (JTextfield)**

  - Dirigir el mouse hacia la solapa: **"Java Beans"**
    Clickear en: **jTexFieldModificacionesNombre**
    Dirigir el mouse hacia la solapa: **"Properties"**
    Clickear en **">text"** y borrar su contenido.

  - Dirigir el mouse hacia la solapa **"Java Beans"**
    Clickear en: **jTexFieldModificacionesApellido**
    Dirigir el mouse hacia la solapa: **"Properties"**
    Clickear en **">text"** y borrar su contenido.

  - Dirigir el mouse hacia la solapa **"Java Beans"**
    Clickear en: **jTexFieldModificacionesDocumento**
    Dirigir el mouse hacia la solapa: **"Properties"**
    Clickear en **">text"** y borrar su contenido.

  - Dirigir el mouse hacia la solapa **"Java Beans"**
    Clickear en: **jTexFieldModificacionesEdad**
    Dirigir el mouse hacia la solapa: **"Properties"**
    Clickear en **">text"** y borrar su contenido.

- Para mejorar la visibilidad de los datos al realizar una **"búsqueda"**, vamos a configurar las siguientes propiedades.

  Primero dirigir el mouse hacia la solapa: **"Java Beans"**

  Clickear en: **this-"Modificaciones"**

  Luego dirigir el mouse hacia la solapa: **"Properties"**:

  - clickear en **"componentOrientation"**
    Elegir: **"LEFT_TO_RIGHT"**

  - clickear en **"maximizable"** y elegir: **"true"**

  - clickear en **"resizable"** y elegir: **"true"**

Ahora listaremos los cambios que haremos a la clase visual **JVentanaSeleccion.java**, para esto:

Dirigir el mouse hacia la solapa: **"Package Explorer"**
Clickear en: **JVentanaSeleccion.java**

- Para configurar algunas de sus propiedades

Dirigir el mouse hacia la solapa: **"Java Beans"**

Clickear en: **this-"Ventana de Selección"**

Luego dirigir el mouse hacia la solapa: **"Properties"**

- clickear en **"componentOrientation"**
  Elegir: **"LEFT_TO_RIGHT"**
- clickear en **"maximizable"** y elegir: **"true"**
- clickear en **"resizable"** y elegir: **"true"**

También agregaremos estas características a la clase visual **JAltas.java**:

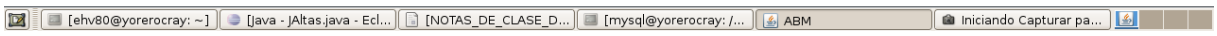
Dirigir el mouse hacia la solapa: **"Package Explorer"**
Clickear en: **JAltas.java**
Luego dirigir el mouse hacia la solapa: **"Java Beans"**
Clickear en: **this-"Altas"**

Dirigir el mouse hacia la solapa **"Properties":**

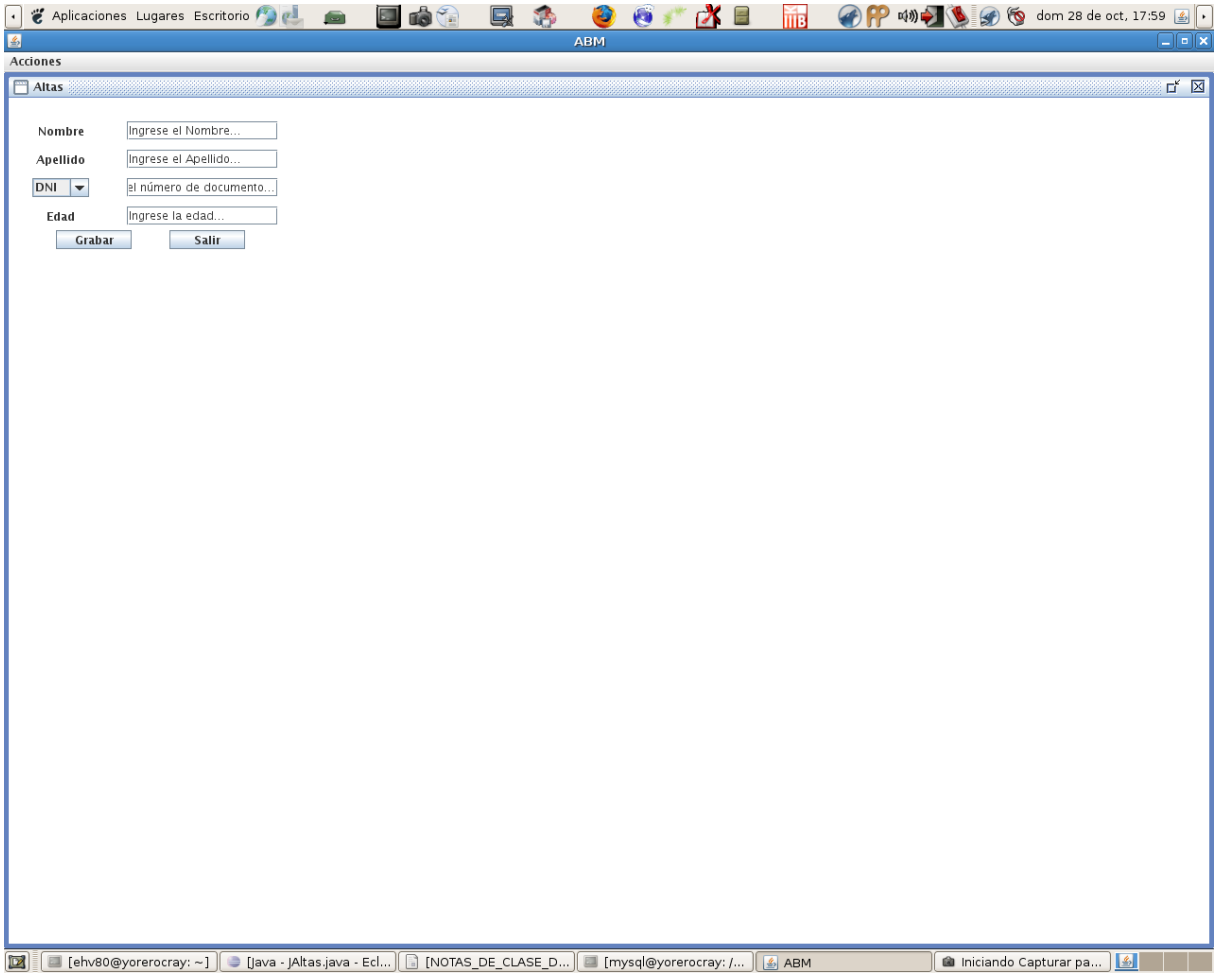
- clickear en **"componentOrientation"**
  Elegir: **"LEFT_TO_RIGHT"**
- clickear en **"maximizable"** y elegir: **"true"**
- clickear en **"resizable"** y elegir: **"true"**

Hemos terminado, por ahora, con todas las modificaciones para este proyecto. Al ejecutarlo se debería ver las respectivas pantallas que se muestran a continuación:

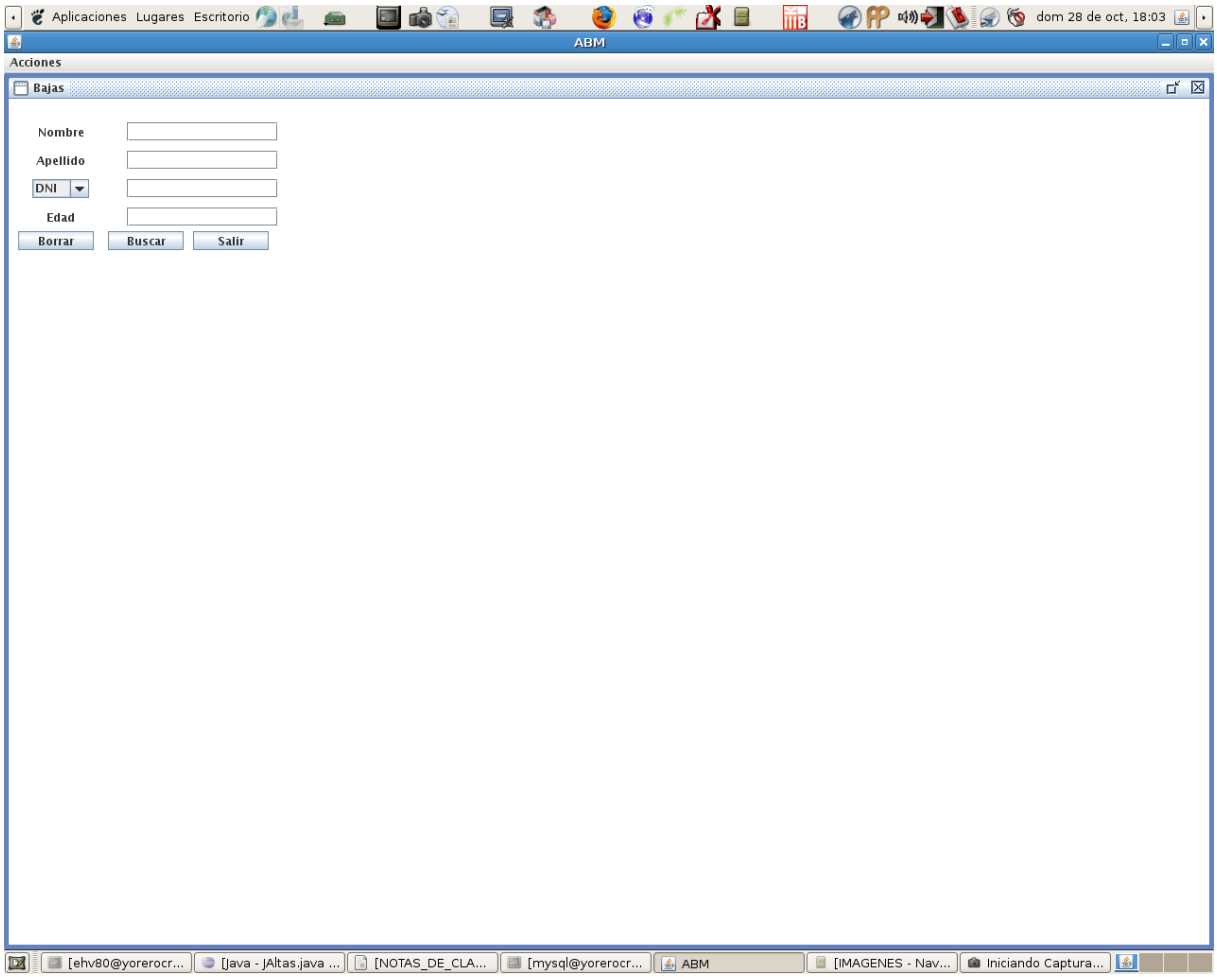
Al iniciar y maximizar la aplicación del **Proyecto_ABM**: **Jabm.java**



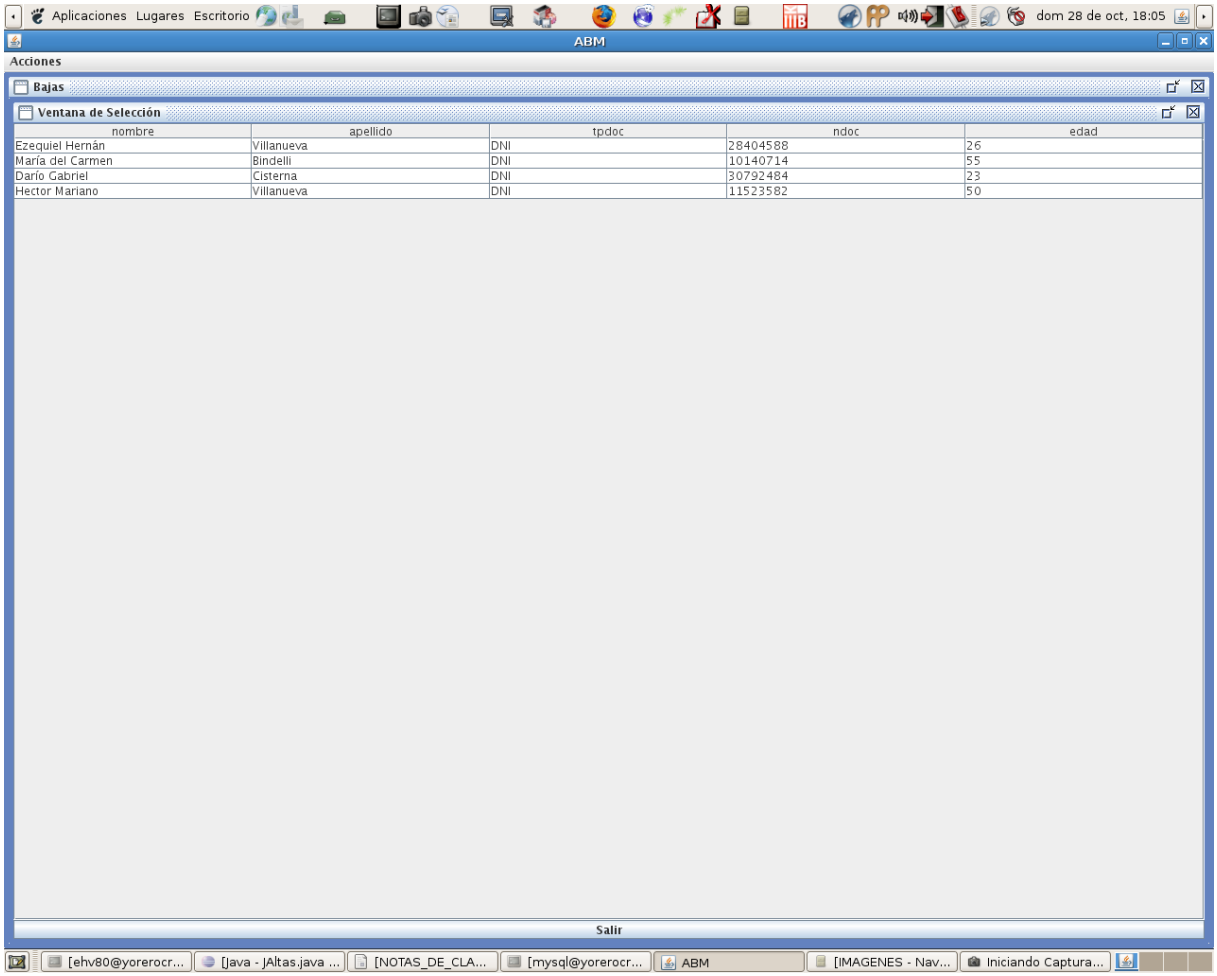
Al clickear, y maximizar, en: **“Acciones” -> “Altas”**



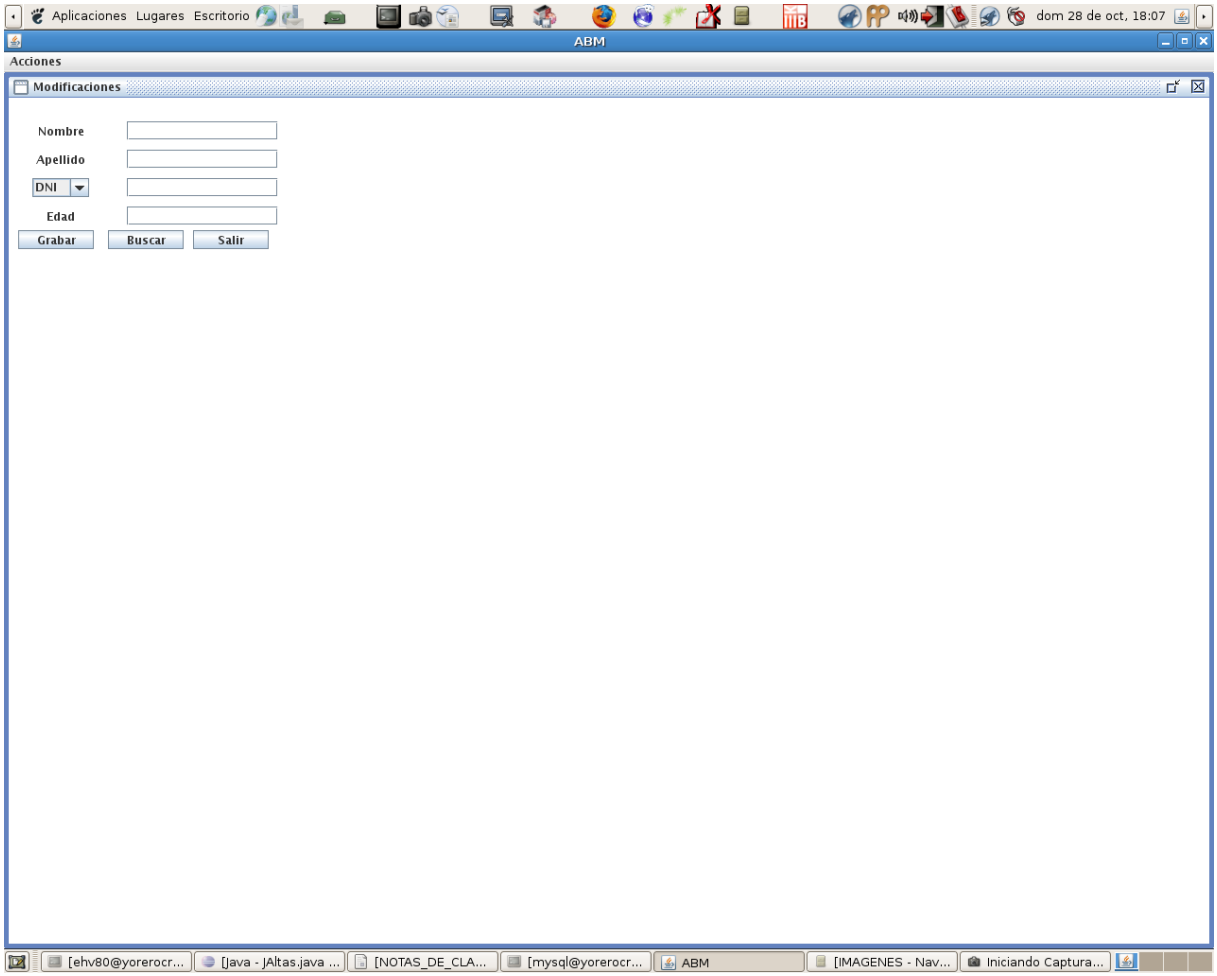
Al clickear, y maximizar, en: **“Acciones” -> “Bajas”**



Al clickear, y maximizar, en: **“Acciones” -> “Bajas” -> “Buscar”**

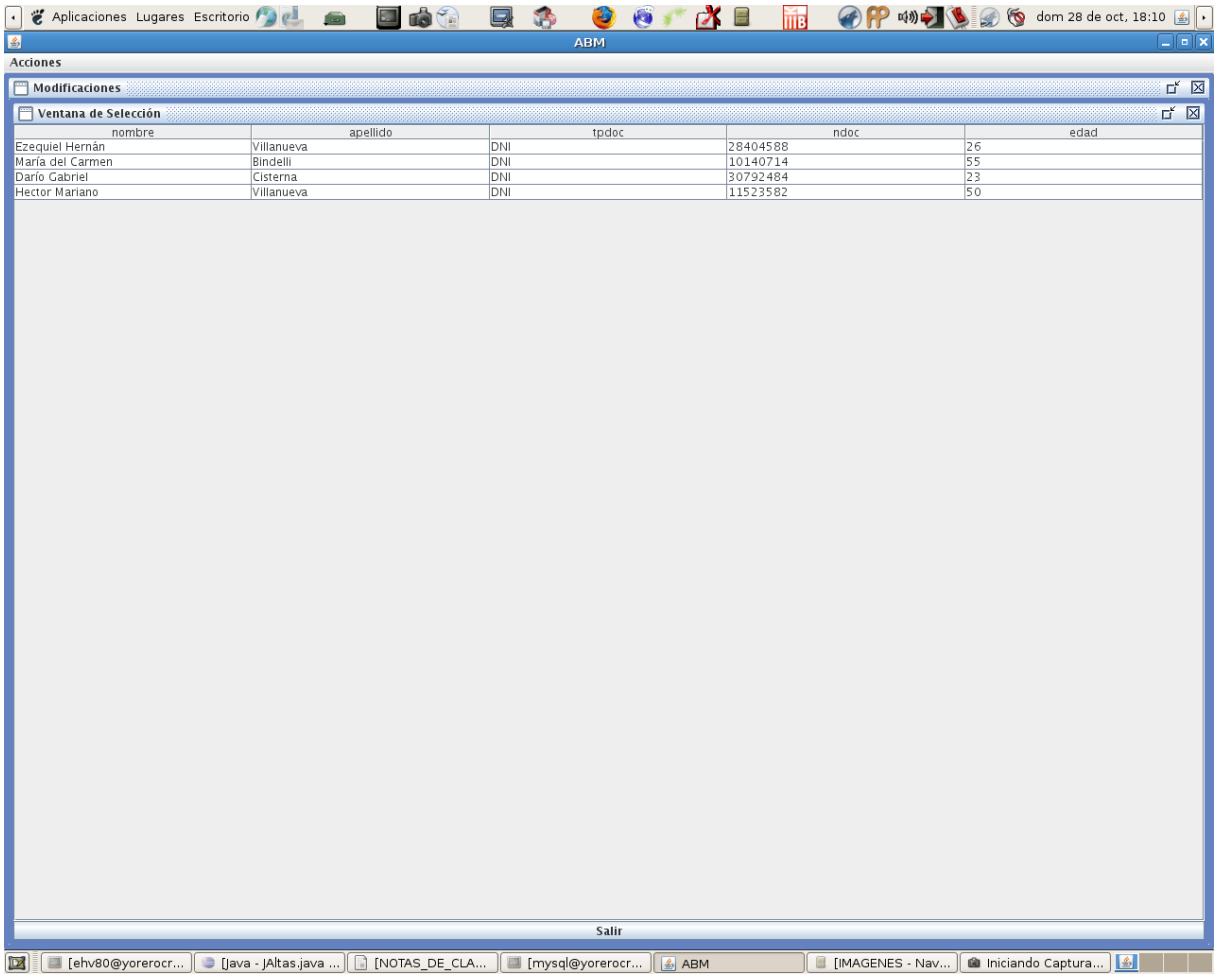


Al clickear, y maximizar, en: **“Acciones” -> “Modificaciones”**



Al clickear, y maximizar, en:

“Acciones” ->“ Modificaciones” -> “Buscar”



[14-08-2007]_[17-08-2007]_[24-08-2007]_[31-08-2007]_[03-09-2007]
[07-09-2007]_[]

Ahora veremos algunos conceptos necesarios para entender cómo es que funcionan las herramientas que proporciona HIBERNATE:

http://www.hibernate.org/

Según el sitio:

“Relational Persistence for Java and .NET

Hibernate is a powerful, high performance object/relational persistence and query service. Hibernate lets you develop persistent classes following object-oriented idiom - including association, inheritance, polymorphism, composition, and collections. Hibernate allows you to express queries in its own portable SQL extension (HQL), as well as in native SQL, or with an object-oriented Criteria and Example API.

Unlike many other persistence solutions, Hibernate does not hide the power of SQL from you and guarantees that your investment in relational technology and knowledge is as valid as always. The LGPL open source license allows the use of Hibernate and NHibernate in open source and commercial projects.

Hibernate is a Professional Open Source project and a critical component of the JBoss Enterprise Middleware System (JEMS) suite of products. JBoss, a division of Red Hat, offers a range of 24x7 Professional Support, Consulting, and Training services to assist you with Hibernate. ”

que en Español es:

“

Persistencia Relacional para Java y .NET

Hibernate es un poderoso, servicio de consultas y persistencia objeto/relacional de alta performance. Hibernate nos permite desarrollar clases persistentes siguiendo un idioma orientado a objetos – incluyendo asociación, herencia, polimorfismo, composición, y colecciones. Hibernate permite expresar las consultas en su propia y portable extensión del lenguaje SQL (HQL), así como también en SQL nativo, o con un API de Ejemplo y Criterio Orientado a Objetos.

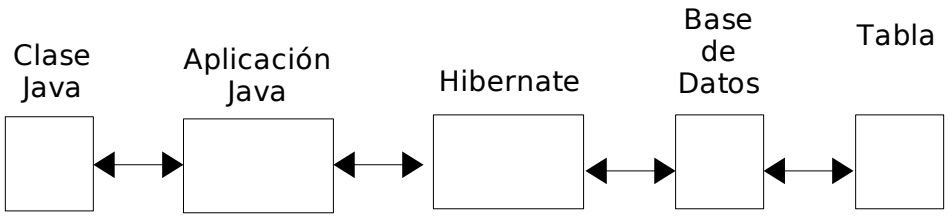
Diferente a muchas otras soluciones de persistencia, Hibernate no oculta el poder de SQL y garantiza que tu inversión en tecnología relacional y conocimiento será tan válido como siempre. La Licencia de código abierto **LGPL** permite usar Hibernate y NHiberbate en proyectos de código abierto y comerciales.

Hibernate es un **Proyecto de Código Abierto Profesional** y un **Componente crítico de** la **suite de productos del Sistema Intermedio Empresarial JBoss (JEMS)**. JBoss, una división de Red Hat, ofrece Servicios de Entrenamiento, Consultoría, y Soporte Profesional en un rango horario 24x7 para asisirte con Hibernate.“

El problema de la persistencia de datos en Java, y en casi todos los lenguajes de programación, es que se concibe a la "BASE DE DATOS" simplemente como un conjunto de tablas y es el programador el encargado de mantener la correspondencia entre **"Atributos"** de una **"instancia de una clase"** y **"campos de una tabla en una base de datos"**.

El **API JDBC** logra una primera forma de persistir los datos haciendo uso de la **"interface" ResultSet,** como vimos en el proyecto: **Proyecto_ABM**.

¿Cómo es el esquema de situación típico del funcionamiento de Hibernate?



Ahora con el uso de las herramientas de **Hibernate**, ya no se requiere que sea el programador quien usa el driver **JDBC MySQL Connector/J**, sino que más bien será el **"SesionFactory" de Hibernate** quien haga uso de esto. **Hibernate** es un **Framework (marco de trabajo).** Veremos cómo hay que **configurar Hibernate** en **Eclipse**. El archivo de configuración mas importante, utiliza el formato **XML,** se trata de un formato de archivo que hace uso de un **"Lenguaje de Marcas Extensible"** muy similar al estilo de los archivos de las páginas web **"HTML"**, con **etiquetas (tags) y directivas.**

El archivo más importante se llama: **hibernate.cfg.xml**

¿Qué hay en el archivo de configuración **hibernate.cfg.xml** ?

Este archivo contiene la información acerca de:

- el **driver** que utilizarán las herramientas de **Hibernate** para conectarse a la **Base de Datos**
- el **nombre de usuario** que se usará para establecer la conexión
- la **contraseña**
- la **dirección IP** del host donde funciona el **Sistema Gestor de Base de Datos**, y **el nombre de la Base de Datos (llamado catalog en las herramientas de Hibernate)**, especificados **como una URL (Uniform Resource Locator)**
- El **dialecto** que se usará **de acuerdo** con el **Sistema Gestor de Base de Datos**

Para realizar la configuración de este archivo: **hibernate.cfg.xml** usaremos un **“plugin” para Eclipse 3.2.1**, que nos proporcionará un **asistente de configuración gráfico** o **“wizard”**, que hace menos tediosa esta tarea, de lo contrario el programador debería saber cuáles son las directivas específicas para poder realizar la configuración mediante la edición manual del archivo, una tarea muy poco agradable.

¿Qué puedo hacer una vez que está listo el archivo de configuración?

Pues una de las ventajas que nos ofrece es poder ver:

- los **datos de la Tabla de una Base de Datos** como si se tratara de un **objeto, instancia de una Clase**;
- también en el sentido contrario: ver un **objeto de una Clase java** como si se tratara de **un Registro (una fila de una Tabla) en una Tabla de una Base de Datos**.

Otra característica importante es que a la hora de realizar una **consulta**, el **resultado** que se obtiene **ya no es un ResultSet**, sino mas bién una **lista de objetos** de una clase. Para lograr esto **Hibernate** tiene un **módulo** que realiza la **Ingeniería Reversa** necesaria para crear los objetos a partir de los campos de las tablas en la base de datos. Pero para realizar esto **hay que indicar** en un **archivo “xml” (hibernate.reveng.xml)**, esta **correspondencia o mapeo**.

La correspondencia entre **“campos de una tabla”** y **“atributos de un objeto”** se especifica mediante el archivo:

[nombre_de_la_clase].hbm.xml

En nuestro ejemplo será: Abm.hbm.xml

Este archivo va a contener la configuración referida a qué **campos de la Tabla en la Base de Datos** se corresponde con qué **variables o atributos de un objeto de una clase y especifica con qué Tipo de dato** se va a llevar a cabo esa correspondencia. Aquí entran en juego los **Tipos de Datos JDBC** (como VARCHAR, INTEGER, etc) y los **Tipos de Datos Definidos por las herramientas de Hibernate** (como string, int, etc. Notar que están en minúscula!)

Este archivo **[nombre_de_la_clase].hbm.xml** estará en el proyecto que se esté desarrollando para utilizar **Hibernate**.

Hay que tener en cuenta que si se usan 2 Tablas en la Base de Datos, se crearán sus 2 clases respectivas, junto con sus 2 archivos **[nombre_de_la_clase].hbm.xml.**

El **plugin de Hibernate para Eclipse** no sólo va a crear las **clases con sus variables**, sino que además incorporará a la clase **los constructores** necesarios, la **implementación del método “equals”**, y si se desea, el asistente de configuración gráfico también puede generar la **documentación “html”**.

Para que todo funcione bién **necesitaremos incorporar a nuestro proyecto 9 archivos de librerías, que tienen la extensión ".jar", que encontrarán en el directorio:**

[ruta_a_eclipse]/
plugins/
org.hibernate.eclipse_3.2.0.beta8/
lib/
hibernate/

estos son:

- **antlr-2.7.6.jar**
- **asm.jar**
- **cglib-2.1.3.jar**
- **commons-collections-2.1.1.jar**
- **commons-logging-1.0.4.jar**
- **dom4j-1.6.1.jar**
- **hibernate3.jar**
- **jta.jar**
- **log4j-1.2.11.jar (es opcional)**

El lenguaje de consulta que se maneja con **Hibernate** se denomina **HQL: Hibernate Query Languaje**. Se trata de una extensión del **Lenguaje de Consutas Estructurado: SQL**, para consultas en **Bases de Datos Objeto/Relacional**.

Por ejemplo, para una **Tabla llamada "ABM"** que **se mapea** a **objetos de una Clase llamada Abm**:

- una **sentencia en SQL**: **select * from ABM;**

  Usando JDBC, retorna un "ResultSet" con todos los registros de la tabla.

- una **consulta en HQL**: **select p from Abm as p;**

  Usando Hibernate Retorna una Lista de objetos de la clase p. Más precisamente indicaremos: List<Abm>

Otros ejemplos de posibles sentencias HQL:

- **select p.nombre from Abm as p**

  Retorna un Lista de String: List <String>

  De la **"tabla ABM" el "campo nombre"** como **"atributo de un objeto de la clase p"**
  Aquí "p" es una "alias" para el nombre de la clase "Abm".

- **select p.nombre,p.apellido from Abm as p**

  Retorna una lista de arreglos de Object: List<Object[]>
  Pues arma un objeto que consta de 2 Strings.
  De la **"Tabla ABM" los "campos nombre y apellido"** como **"atributos de un objeto de la Clase p" que son "parte de" un Object.**
  Aquí se deberá realizar "casting" para controlar los Tipos de Datos.

Todo esto cosume muchísimos recursos de hardware y memoria.

Un **tema clave de Hibernate** es **cómo manejar la Persistencia de los datos**.

Para **modificar (insert, update, delete) datos** en la **Base de Datos** mediante **Hibernate:**

- Hay que **crear <u>una única</u> instancia de la Clase "SessionFactory":**

```
private static SessionFactory sessionFactory;
sessionFactory = new  Configuration().configure().buildSessionFactory();
```

- **A partir de** este objeto de la Clase **"SessionFactory"** podremos crear **<u>una única</u>** instancia de la Clase **"Session":**

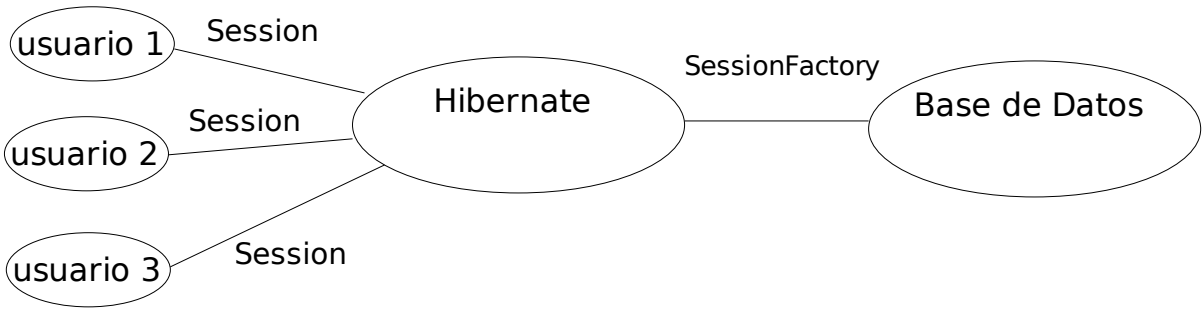
```
private static Session session;
session = sessionFactory.openSession();
```

- Mediante el objeto de la Clase **"Session"** podremos invocar algunos de sus **Métodos**. Las herramientas de **Hibernate** se encargarán de **traducir estos Métodos a la sentencia HQL necesaria** de acuerdo al **dialecto, de la Base de Datos, elegido**. Comúnmente entre ellos se suelen utilizar:
  - **Método: "save(Object arg)"**
    El objeto pasado como argumento, será almacenado en la Base de Datos.
  - **Método: "saveOrUpdate(Object arg)"**
    El objeto pasado como argumento, será actualizado en la Base de Datos.
  - **Método "delete(Object arg)"**
    El objeto pasado como argumento, será eliminado en la Base de Datos.

Supongamos que queremos grabar un objeto nuevo, el código fuente sería similar al siguiente:

```
Abm a = new abm( ... );
sesion.save(a);
```

De esta manera pueden haber **muchos usuarios conectados** a una **"Session" de Hibernate, NÓ A LA BASE DE DATOS**, mientras que **Hibernate hace una sola conexión con la base de datos mediante "SessionFactory"**.



Estas, y otras clases, están definidas dentro de los 8 archivos **".jar"** que deberemos importar.

Un código fuente de ejemplo para crear una instancia de la clase **SessionFactory** sería:

```
SessionFactory sf;
sf = new Configuration().Configure().buildSessionFactory();
/*
 * Crea un SessionFactory. En general se crea uno por
 * aplicación. Se recomienda hacerlo al inicio de la misma
 * pues es un proceso que consume muchos recursos.
 * El driver JDBC que usará Hibernate se le indicará en el
 * asistente de configuración.
 */
Session s = sf.openSession();
/*
 * Hace una conexión entre usuario e Hibernate. Esta
 * sesión proporciona los métodos para modificar los datos
 * de la base de datos
 */
List<abm> lista = (List<Abm>)s.createQuery(
                                "select p from abm as p"
                            ).list();
/*
 * Almacena el resultado de la consulta HQL en una lista de
 * objetos de tipo "Abm".
 */
```

```
/* La clase Session posee los métodos:
 *     Save
 *     SaveOrUpdate
 *     Delete
 */
/* El método "Delete" de la clase "Session" elimina objetos
 * de la sesión con Hibernate. Luego Hibernate borra los
 * registros necesarios de la Tabla en la Base de Datos.
 */
```

Luego de esta breve introducción, nos dispondremos a configurar un **proyecto nuevo en Eclipse** para hacer uso de los **plugins de Hibernate**.

Antes que nada debemos descargar el archivo:

HibernateTools-3.2.0.beta8.zip

Este archivo debemos descomprimirlo en algún directorio del sistema de archivos. Luego copiar:

- todo el contenido del directorio **"features"** en el directorio **"features"** del **Eclipse**

- todo el contenido del directorio **"plugins"** en el directorio **"plugins"** del **Eclipse**.

Para que **Eclipse** tome los **"plugins"** y **"features"** incorporados, hay que eliminar los siguientes directorios:

[ruta_al_eclipse]/configuration/org.eclipse.core.runtime/
[ruta_al_eclipse]/configuration/org.eclipse.osgi/
[ruta_al_eclipse]/configuration/org.eclipse.update/

Luego puede iniciarse **Eclipse** normalmente.
Observará algunas nuevas **opciones**, **botones** y **perspectivas relacionadas a Hibernate.**

Una aclaración **muy importante** es que **antes de realizar** todo este **procedimiento**, se debe **ejecutar el Sistema Gestor de Base de Datos MySQL** y **dejarlo en funcionamiento**.

Para agregar un nuevo proyecto en Eclipse:

File -> **New** -> **Project**

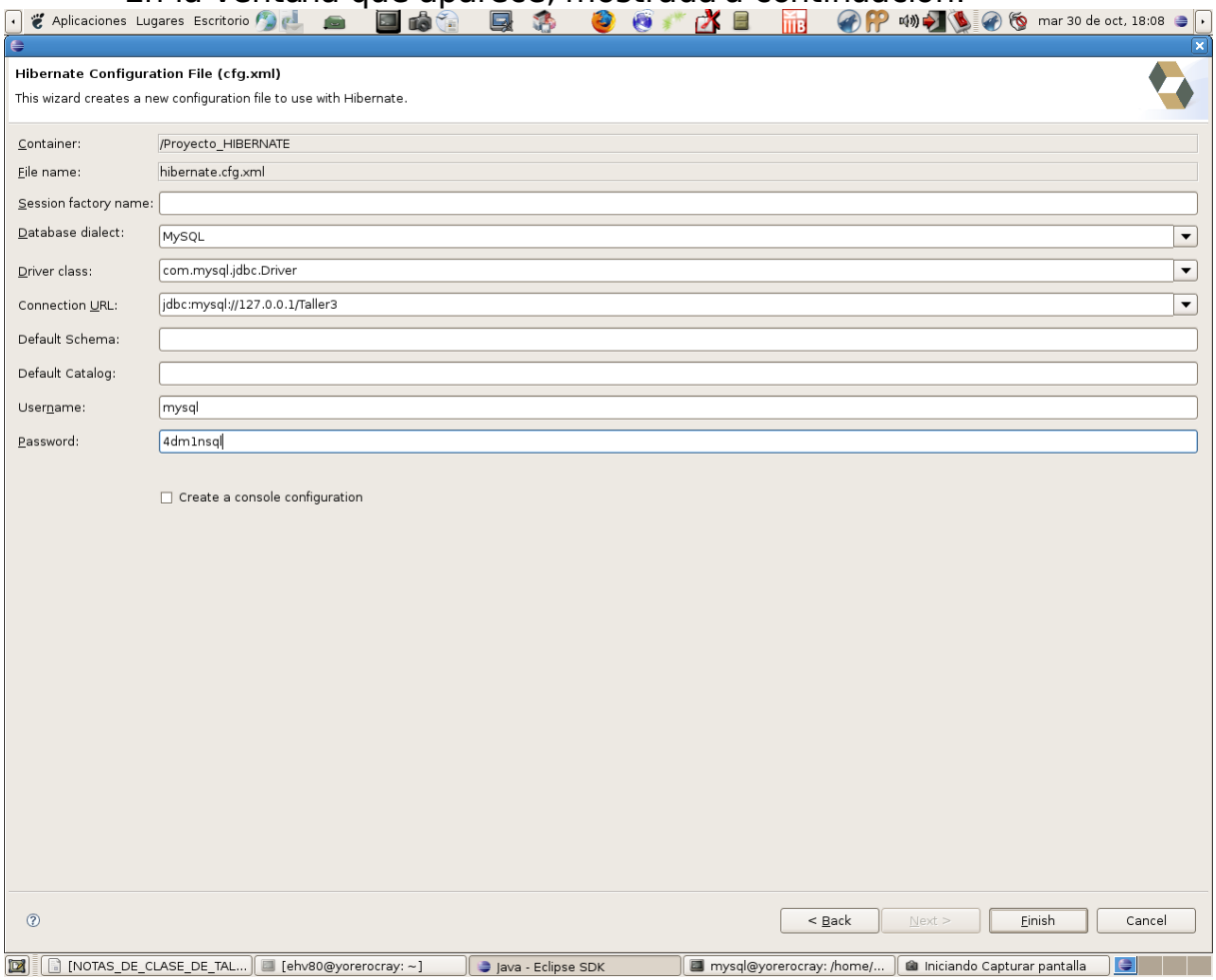
En el asistente seleccionar: **Java Project**

Clickear en: **Next**
En asistente abierto completar:
Project Name: **Proyecto_HIBERNATE**
(*) Create new project in workspace
(*) Use default JRE (Currently 'jdk1.6.0_02')
(*) Use a project folder as root for sources and class files
Clickear en: **Next**
Clickear en: **Finish**

Vamos a **crear el archivo de configuración**: **hibernate.cfg.xml**

Dirigir el mouse hacia la solapa: **"Package Explorer"**
Clickear con botón derecho del mouse sobre: **"Proyecto_HIBERNATE"**
Clickear en: **New -> Other...**
En la ventana que aparece elegir:
Hibernate -> Hibernate Configuration File (cfg.xml)
Clickear en: **Next**
En la ventana que aparece:
"Create Hibernate Configuration file (cfg.xml)"
completar:
File name: **hibernate.cfg.xml**
Clickear en: **Next**

En la ventana que aparece, mostrada a continuación:



"Hibernate Configuration file (cfg.xml)" completar:
Container: **/Proyecto_HIBERNATE**
File Name: **hibernate.cfg.xml**
Session factory name:
Database dialect: **MySQL**
Driver class: **com.mysql.jdbc.Driver**
Connection URL: **jdbc:mysql://127.0.0.1/Taller3**
Default Schema:
Default Catalog:
Username: **mysql**
Password: **4dm1nsql**
Clickear en: **Finish**

Hecho esto, deberemos esperar hasta que termine de generar el archivo de configuración **“xml”**. Una vez que finaliza, va a mostrar, en el área del editor de código fuente, una solapa llamada **“hibernate.cfg.xml”** que muestra las propiedades que hemos definido.

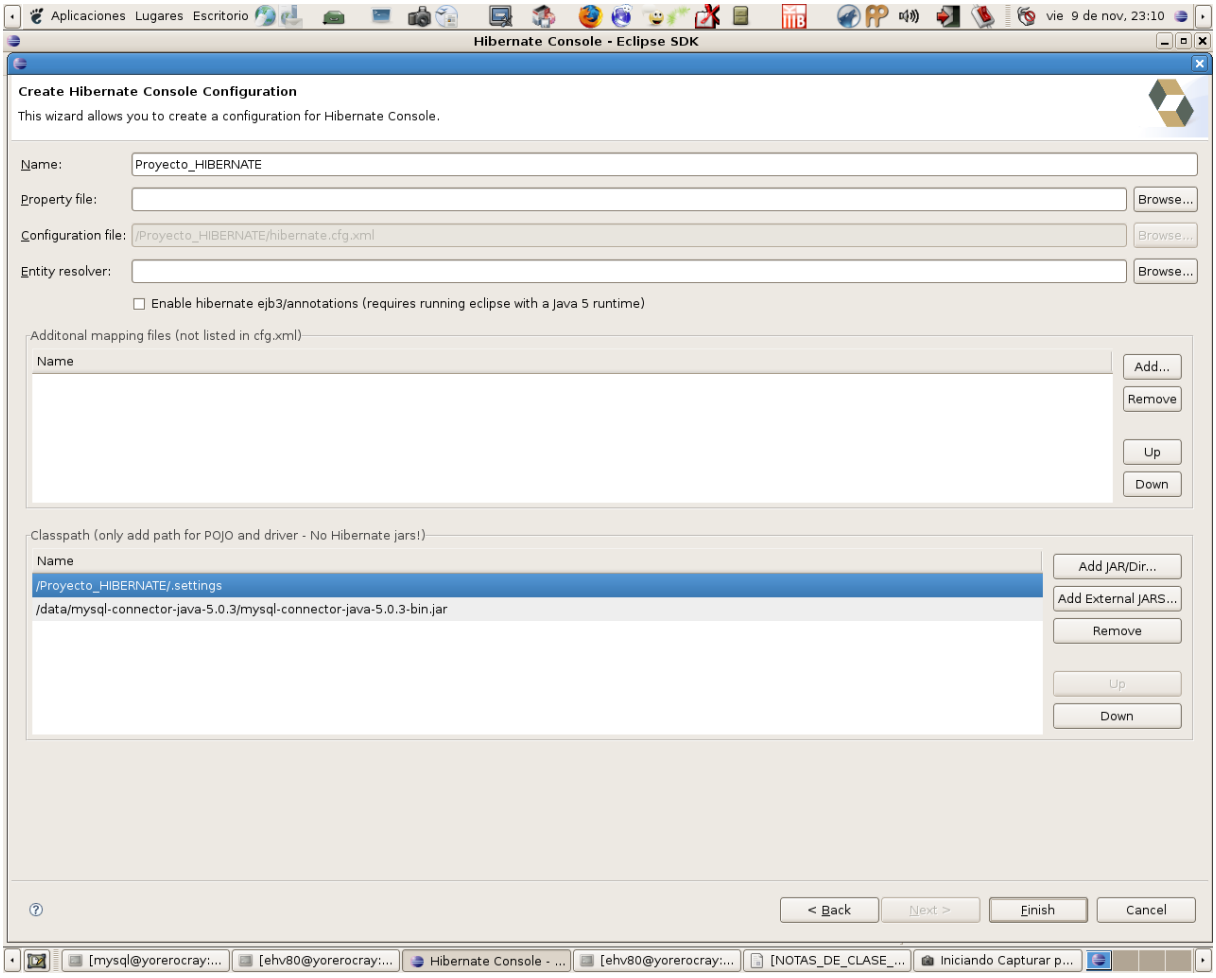
Se observa el siguiente contenido en la solapa **“hibernate.cfg.xml”**, haciendo click en la solapa interna inferior **“Source”:**

```
<?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.bytecode.use_reflection_optimizer">false
        </property>
        <property
           name="hibernate.connection.driver_class">com.mysql.jdbc.Driver
        </property>
        <property
           name="hibernate.connection.password">4dm1nsql
        </property>
        <property
           name="hibernate.connection.url">jdbc:mysql://127.0.0.1/Taller3
        </property>
        <property
           name="hibernate.connection.username">mysql
        </property>
        <property
           name="hibernate.dialect">org.hibernate.dialect.MySQLDialect
        </property>
        <mapping resource="Abm.hbm.xml" />
    </session-factory>
</hibernate-configuration>
```

El siguiente paso es crear una **“Consola de Configuración”**. Por lo general, a cada **“archivo de configuración de Hibernate”** le corresponde una **“Consola de Configuración”**, que sirve para monitorizar el estado de la conexión que realiza el **“SessionFactory”** con la **“Base de Datos”**.

Vamos a configurar la **“Consola de Configuración”**

Dirigir el mouse hacia la solapa: **“Package Explorer”**
Clickear con botón derecho del mouse sobre: **“Proyecto_HIBERNATE”**
Clickear en: **New -> Other...**
En la ventana que aparece elegir:
Hibernate -> Hibernate Console Configuration
Clickear en: **Next**
En la ventana que aparece
“Create Hibernate Console Configuration”, mostrada a continuación:



En esta ventana **"Create Hibernate Console Configuration"**
Verificar que contenga los siguientes datos:
Name: **Proyecto_HIBERNATE**
Property file:
Configuration file: **/Proyecto_HIBERNATE/hibernate.cfg.xml**
Entity resolver:

En **"Additional Mapping files (not listed in cfg.xml)"**:
Dejaremos sin cambios.

En **"Classpath (only add path for POJO and driver –No Hibernate jars!)"**
Clickear en: **"Add External JARS..."**
Examinar el Sistema de Archivos para encontrar el directorio donde se encuentra el **driver JDBC MySQL Connector/J**
En mi caso es:
/data/
mysql-connector-java-5.0.3/
mysql-connector-java-5.0.3-bin.jar
Clickear en: **Aceptar**
Clickear en: **Finish**

Ahora vamos a activar la **"Perspectiva de Hibernate"**, para ello:

Clickear en la Barra de Menú de Eclipe sobre la opción:
Window -> Open Perspective -> Other...
Elegir: **Hibernate Console**
Clickear en: **OK**

También tendremos que activar **las siguientes vistas**:

Clickear en el Menú de Eclipse en: **Window -> Show View -> Other...**
Clickear en: **Hibernate -> Hibernate Configurations**
Clickear en: **OK**

Clickear en el Menú de Eclipse en: **Window -> Show View -> Other...**
Clickear en: **Hibernate -> Hibernate Dynamic SQL Preview**
Clickear en: **OK**

Clickear en el Menú de Eclipse en: **Window -> Show View -> Other...**
Clickear en: **Hibernate -> Hibernate Entity Model**
Clickear en: **OK**

Clickear en el Menú de Eclipse en: **Window -> Show View -> Other...**
Clickear en: **Hibernate -> Hibernate Query Result**
Clickear en: **OK**

Clickear en el Menú de Eclipse en: **Window -> Show View -> Other...**
Clickear en: **Hibernate -> Hibernate Relational Model**
Clickear en: **OK**

EL paso número 3, es configurar los aspectos relacionados a la **ingeniería inversa:**

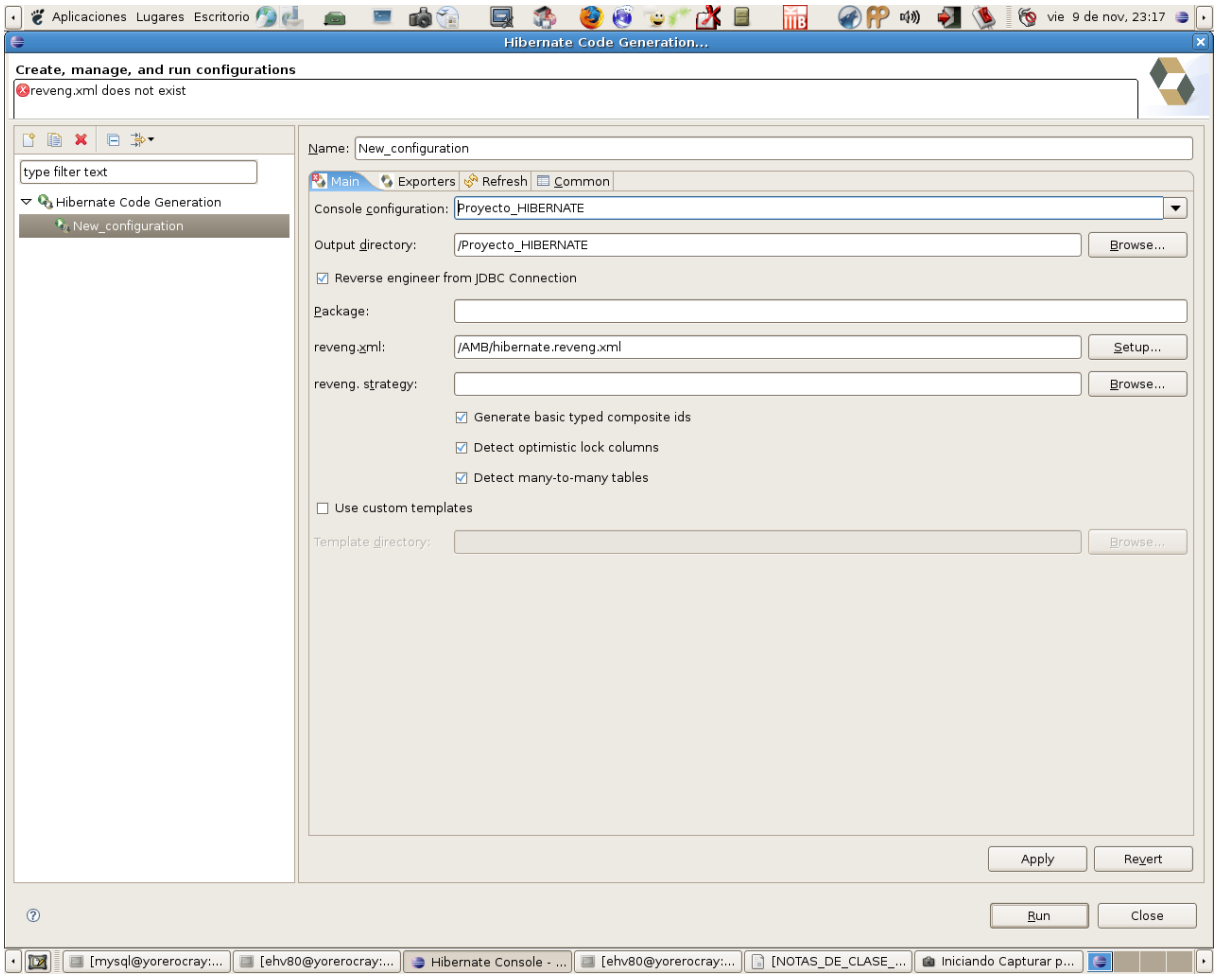
Clickear en el botón específico, que tiene el logotipo de Hibernate:
Hibernate Code Generation… -> Hibernate Code Generation

En la ventana que aparece titulada
“Create, manage and run configurations”:

Clickear con botón derecho del mouse en: **Hibernate Code Generation**
Clickear en: **New**

Debería aparecer en la sección izquierda: **New_Configuration**

Mientras, en la sección derecha se muestran varias solapas:
En **“Name”** verificar que contenga: **New_Configuration**
Esta ventana luce como se muestra a continuación:



En la **solapa "Main":**

Clickear en el botón de la opción **"Console configuration"**
Elegir: **Proyecto_HIBERNATE**

En **"Output directory"** clickear en **"Browse"**
Elegir: **Proyecto_HIBERNATE**
Clickear en: **OK**

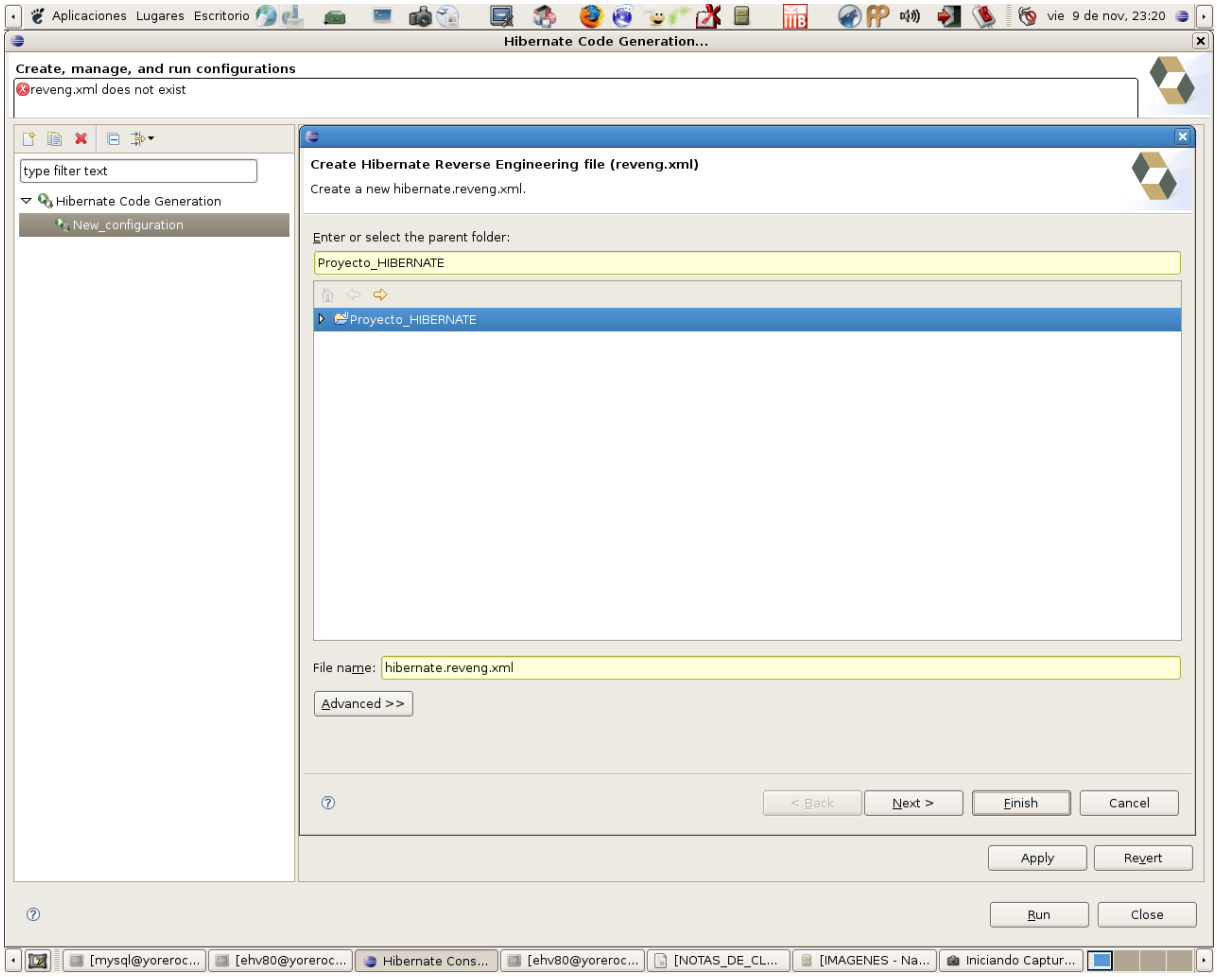
Tildar la opción: **Reverse engineer from JDBC Connection**

En "Package": dejar sin completar

En **"reveng.xml"** clickear en **"Setup..."**

En la ventana que aparece **"Setup reverse engineering"**:
Clickear en: **"Create new..."**

En la ventana que aparece
"Create Hibernate Reverse Engineering file (reveng.xml)"
mostrada a continuación:



Clickear en la carpeta: **"Proyecto_HIBERNATE"**
En **"File name"** va: **hibernate.reveng.xml**
Clickear en: **Next**

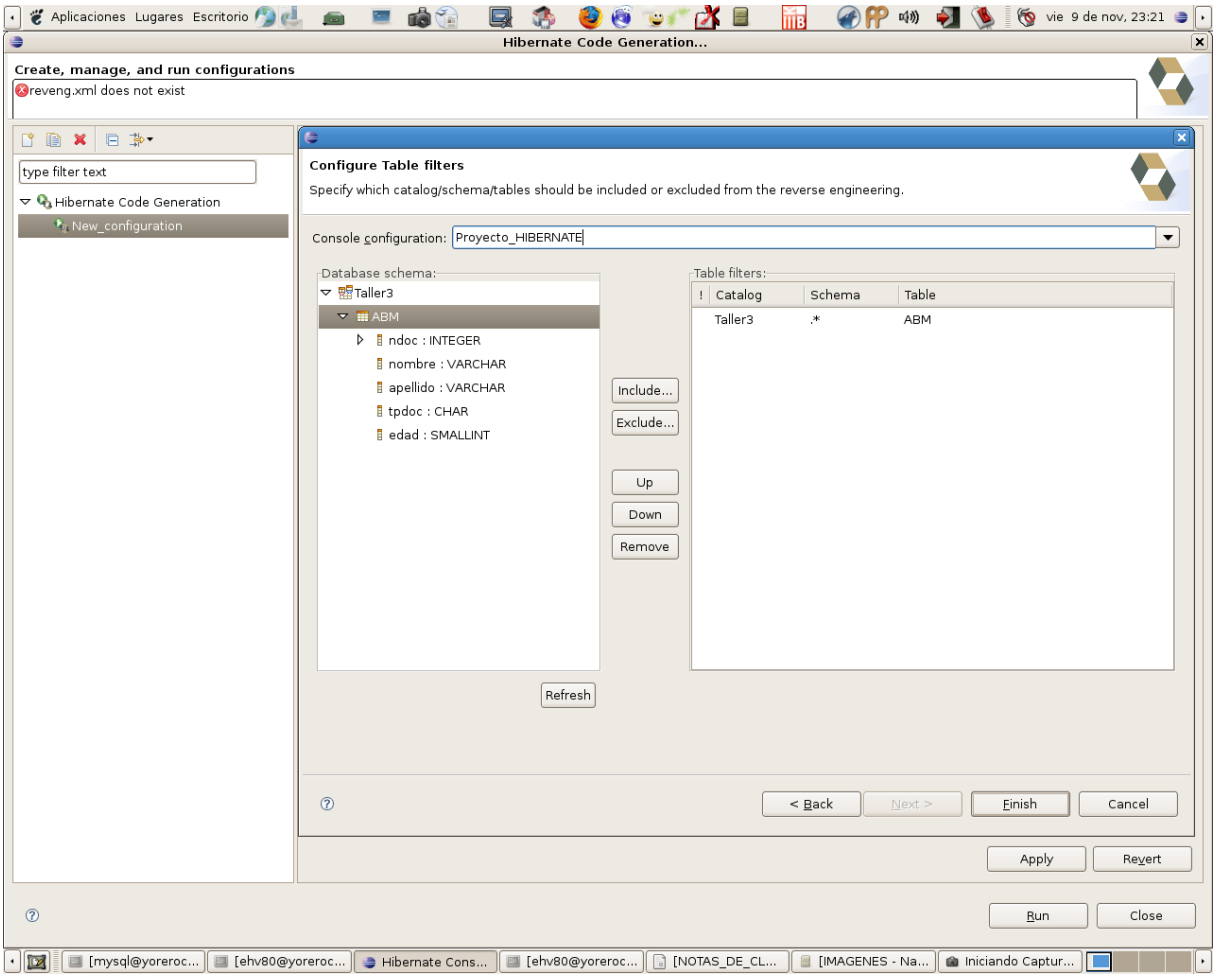
En ventana que aparece “**Configure Table Filters**”:

En **“Console Configuration”:**
Elegir: **Proyecto_HIBERNATE**

Clickear en el botón: **Refresh**

Aparece en la sección **“Database Schema”:** **Taller3**
Clickear para desplegar la Base de Datos: **Taller3**
Clickear para desplegar la Tabla: **ABM**
Clickear para seleccionar a la Tabla: **ABM**
Clickear en el botón: **Include...**
Clickear en: **Finish**

Esta ventana debería lucir como se muestra a continuación:



Vuelve a la ventana **“Create, manage and run configurations”**:

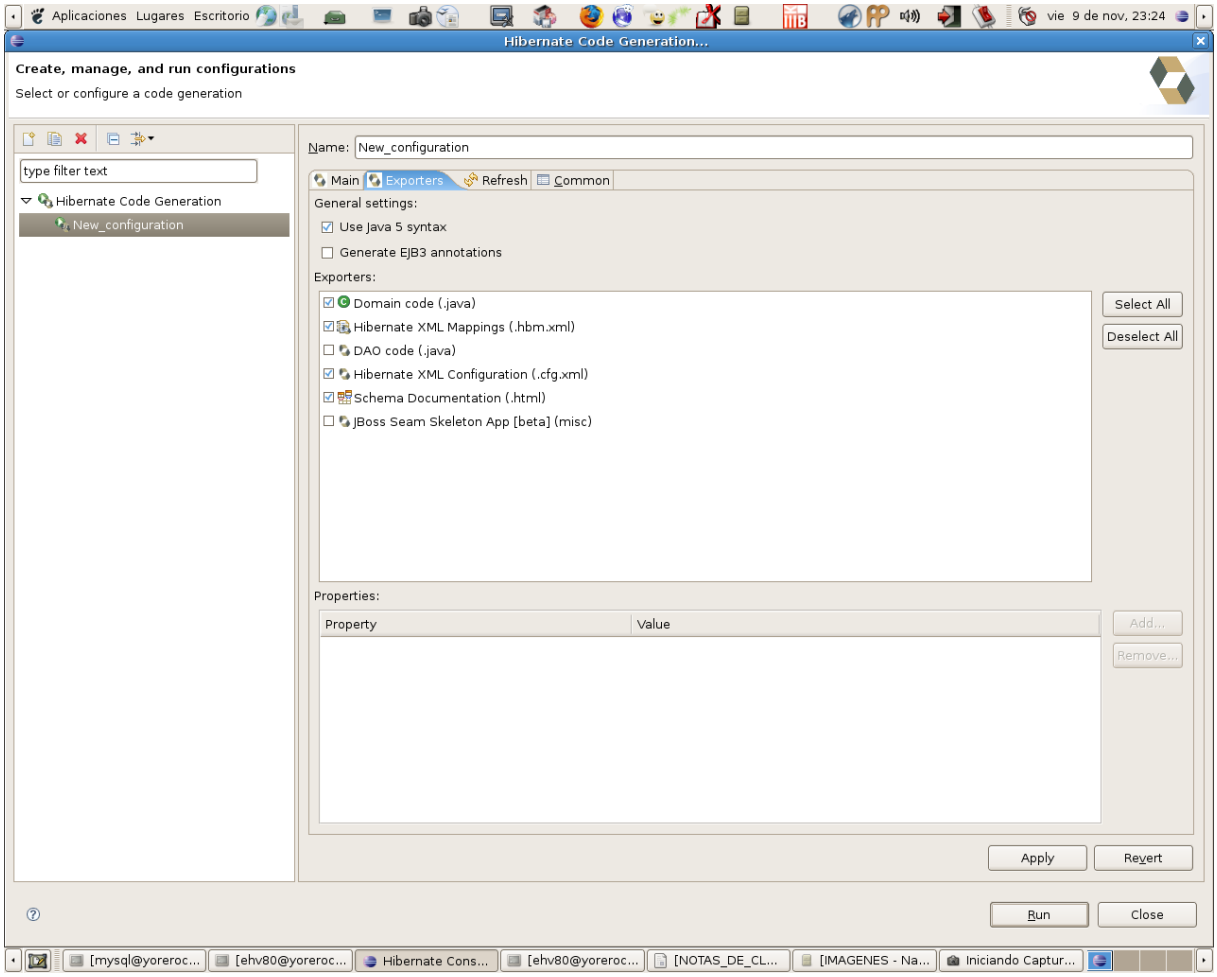
Clickear en la **solapa “Exporters”**

En **“General Settings”** tildamos la opción:
[*] Use Java 5 syntax
Dejamos sin tildar:
[ ] Generate EJB3 annotations

En **“Exporters”** tildamos las opciones:
[*] Domain code (.java)
[*] Hibernate XML Mappings (.hbm.xml)
[*] Hibernate XML Configuration (.cfg.xml)
[*] Schema Documentation (.html)
Dejamos sin tildar:
[ ] DAO Code (.java)
[ ] JBoss Seam Skeleton App [beta] (misc)

Clickear en: **Apply**
Clickear en: **Run**

Esta ventana debería verse como se muestra a continuación:



El asistente realiza todos los procesos necesarios para la creación, **a partir de los campos de la Tabla en la Base de Datos**, de **las Clases java,** y **los archivos de configuración "xml"**, **la Documentación en formato "Html"** como se muestra a continuación:

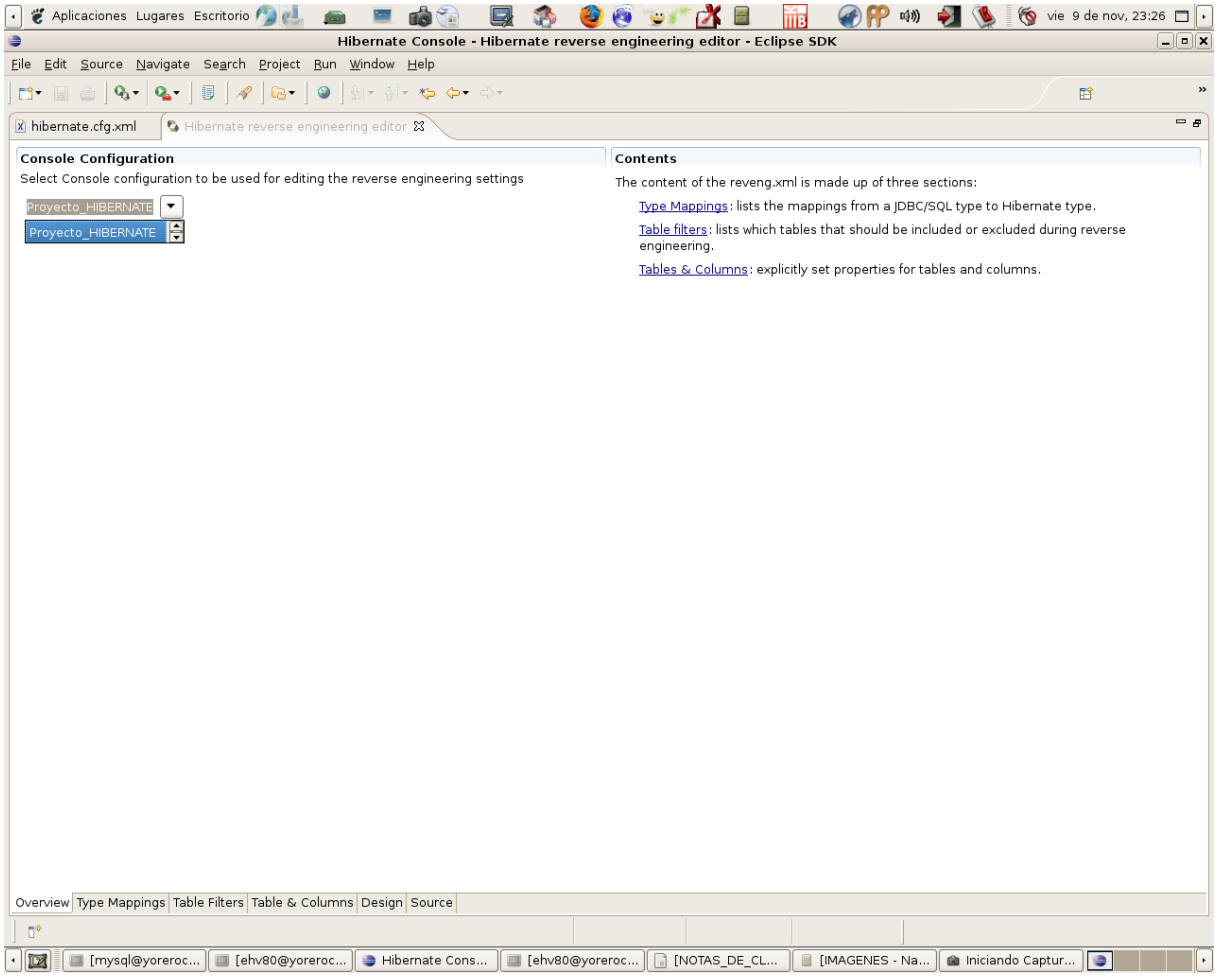
- **assets/**
  - **doc-style.css**
  - **hibernate_logo.gif**
- **entities/**
  - **Abm.html**
  - **allentities.html**
  - **allpackages.html**
  - **index.html**
  - **summary.html**
- **tables/**
  - **Taller3.default/**
    - **ABM.html**
    - **summary.html**
    - **tables.html**
  - **allschemas.html**
  - **alltables.html**
  - **index.html**
  - **summary.html**
- **Abm.class**
- **Abm.hbm.xml**
- **Abm.java**
- **header.html**
- **hibernate.cfg.xml**
- **hibernate.reveng.xml**
- **index.html**
- **AbmHome.class**
- **AbmHome.java**
- **index.html**

En el área del editor de código fuente aparecerá una solapa nueva llamada **“Hibernate reverse engineering editor”**, que posee diversas solapas internas para realizar configuraciones:

Clickear en la **solapa interna inferior: “Overview”**

En la sección izquierda **“Console Configuration”**:
Seleccionar: **Proyecto_HIBERNATE**

Esta **solapa interna inferior “Overview”** debería lucir como mostramos a continuación:

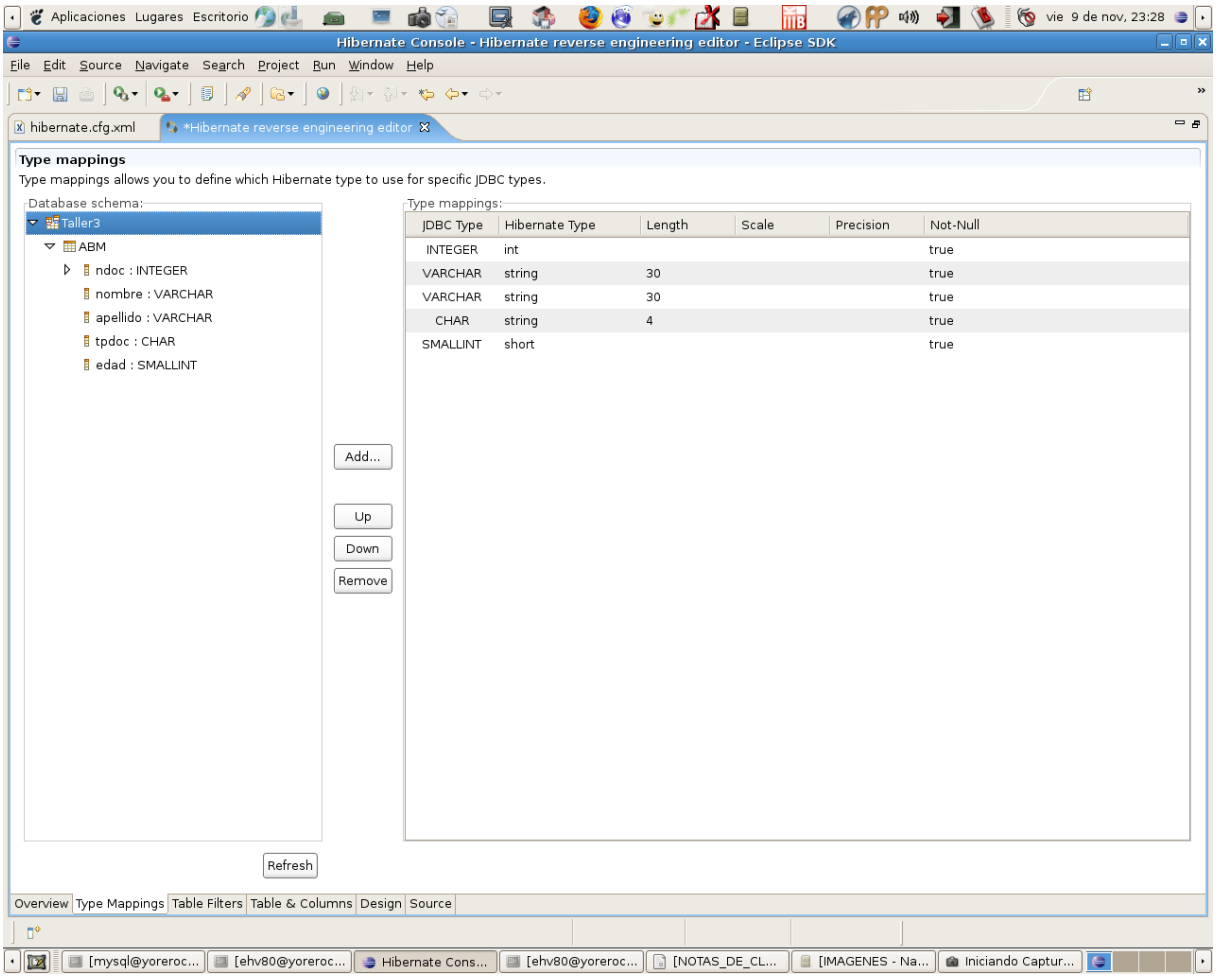


Clickear en la **solapa interna inferior: “Type Mappings”**

Clickear en: **“Refresh”**

En la **sección izquierda**: **“Database schema”**
Clickear en: **Taller3**
Clickear en: **ABM**
Clickear en **“ndoc : INTEGER”** y luego en **“Add...”**
Clickear en **“nombre : VARCHAR”** y luego en **“Add...”**
Clickear en **“apellido : VARCHAR”** y luego en: **“Add...”**
Clickear en **“tpdoc : CHAR”** y luego en: **“Add...”**
Clickear en **“edad : SMALLINT”** y luego en: **“Add...”**

Esta **solapa interna inferior “Type Mappings”** debería lucir como se muestra a continuación:



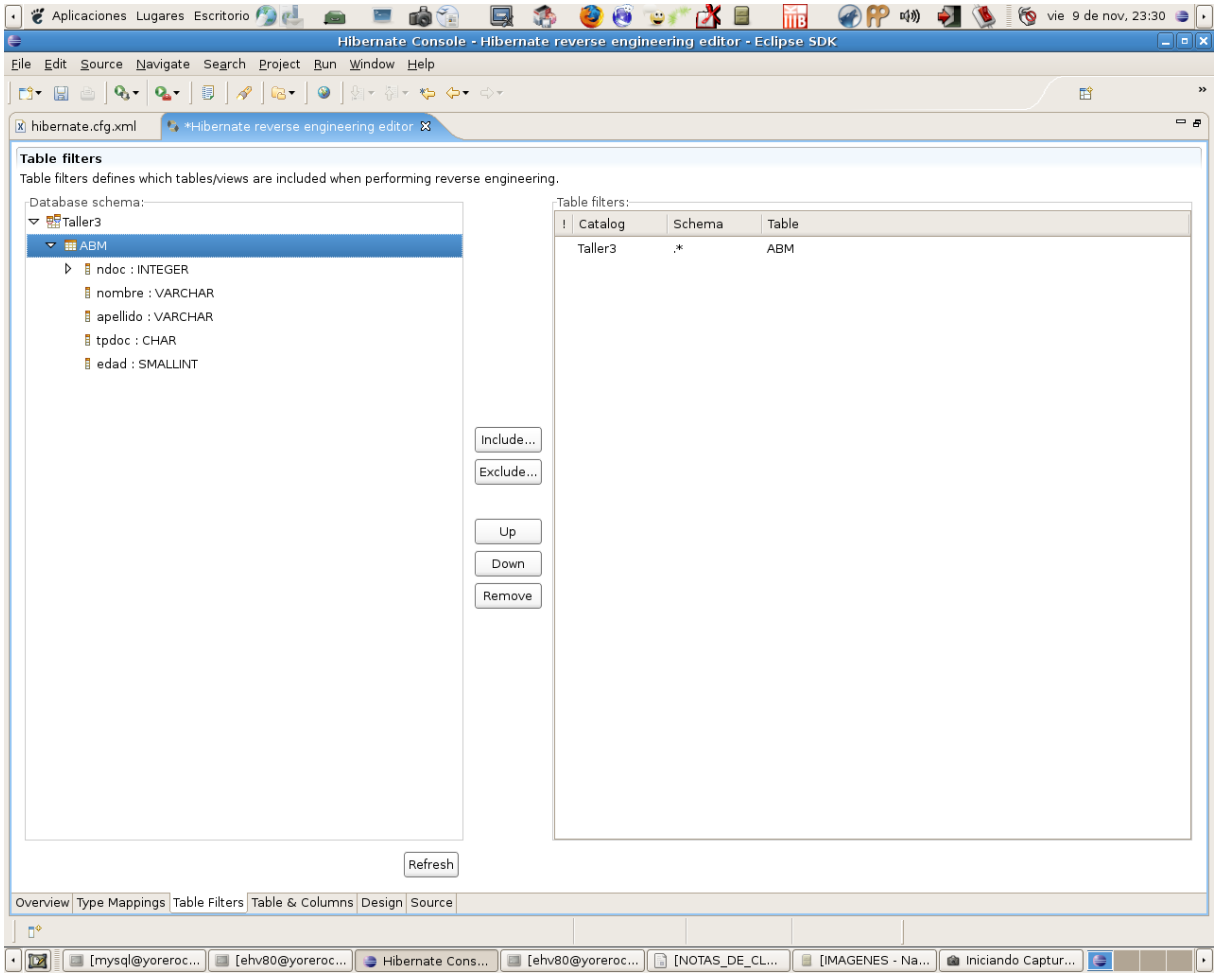
Clickear en la **solapa interna inferior:** **“Table Filters”**

En la **sección izquierda**: **“Database Schema”**
Para actualizar los datos se puede clickear en: **“Refresh”**
Aquí no es necesario hacerlo.

En la **sección derecha**: **“Table Filters”**
Hay que verificar que contenga los siguientes datos:

Catalog	**Schema**	**Table**
Taller3	**.***	**ABM**

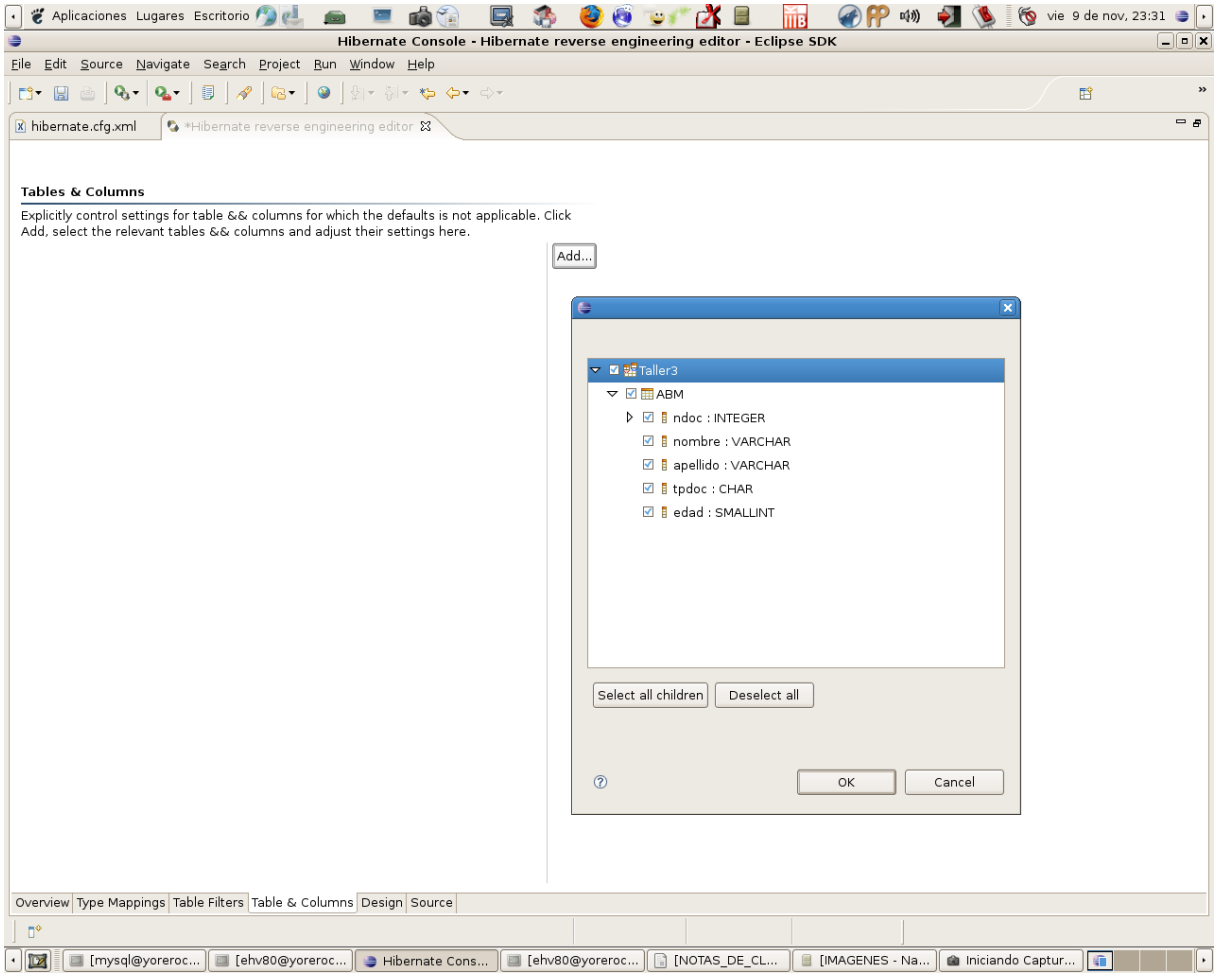
Esta **solapa interna inferior “Table Filters”** debería verse como mostramos a continuación:



Clickear en la **solapa interna inferior: “Tables & Columns”**

Clickear en: **Add...**

En la ventana que aparece, mostrada a continuación:



Clickear, para desplegar la base de datos: **[ ] Taller3**
Clickear, para desplegar la tabla: **[ ] ABM**
Tildar en la casilla de verificación de la tabla: **[*] ABM**
De esta manera deberían tildarse automáticamente:

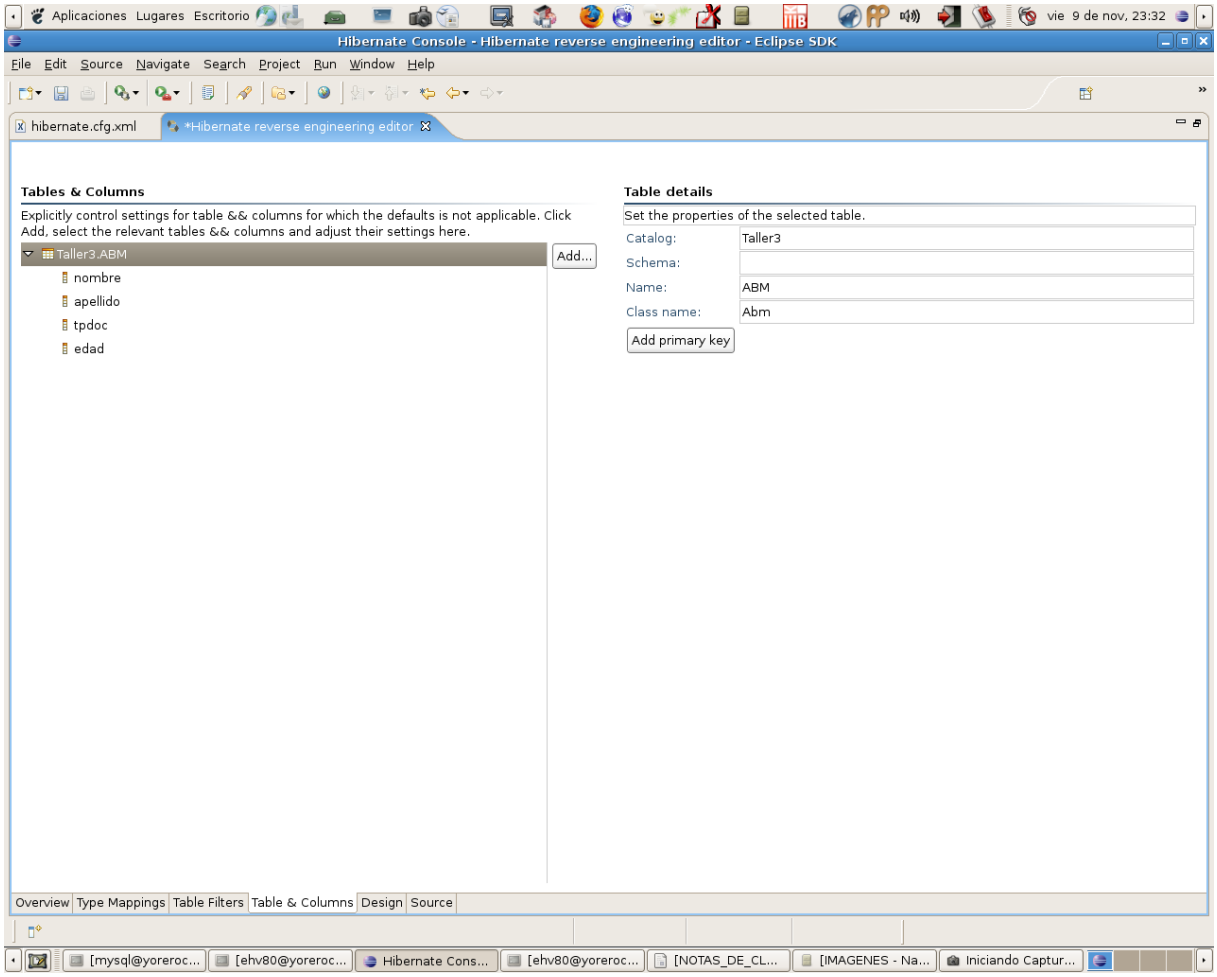
[*] Taller3
[*] ABM
[*] ndoc : INTEGER
[*] nombre : VARCHAR
[*] apellido : VARCHAR
[*] tpdoc : CHAR
[*] edad : SMALLINT

Clickear en: **OK**

En **la sección izquierda**: **“Tables & Columns”**
Clickear para desplegar la tabla: **Taller3.Abm**

En **la sección derecha:** **“Table details”**
En **“Catalog”** verificar que contenga: **Taller3**
En **“Schema”** dejamos como está, en blanco.
En **“Name”** verificar que contenga: **ABM**
En **“Class Name” introducir**: **Abm**

Esta ventana debería lucir como se muestra a continuación:

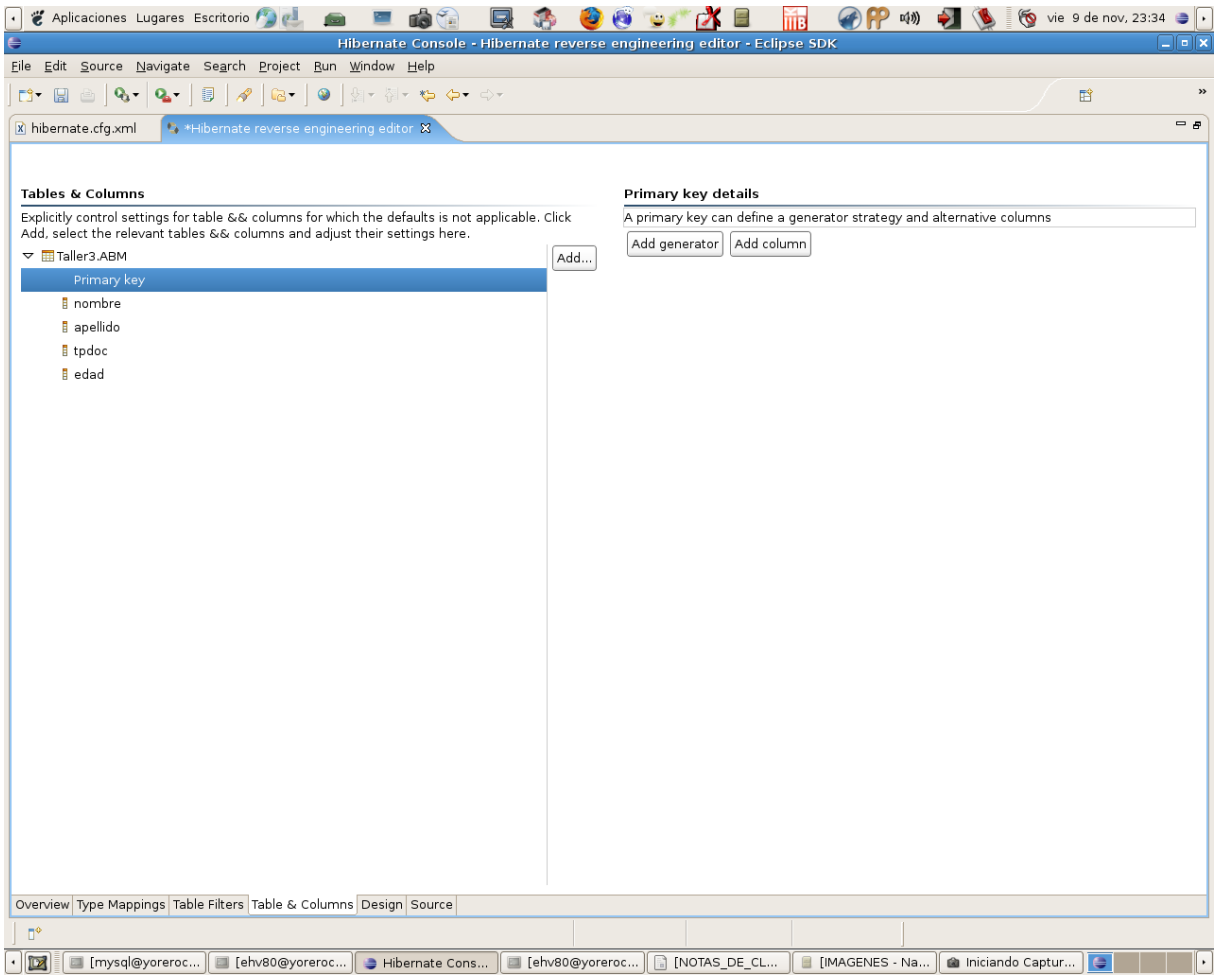


Clickear en el botón: **Add primary key**

En la **sección izquierda:** **“Tables & Columns”**
Clickear en: **Primary key**

Entonces en la **sección derecha**: **“Primary key details”**
Se pueden observar 2 botones:
“Add generator”
“Add column”

Esta ventana debería verse como la que se muestra a continuación:

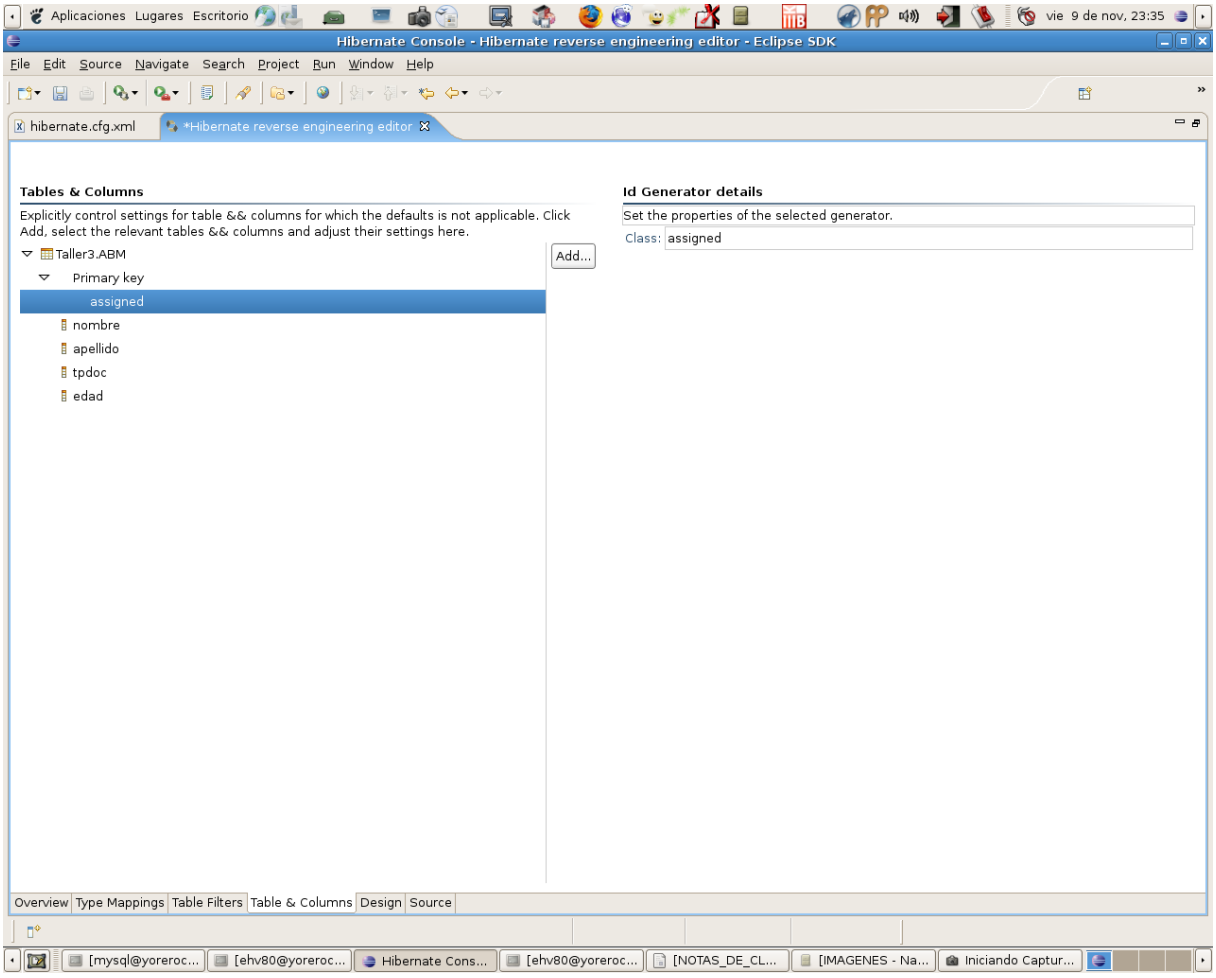


Clickear en el botón: **Add generator**

Ahora en la **sección izquierda**: **“Tables & Columns”**
Clickear para desplegar contenido en: **Primary key**

Entonces en la **sección derecha**: **“Id Generator details”**
En **“Class”** dejamos cómo está: **assigned**

Esta ventana debería lucir como la mostrada a continuación:



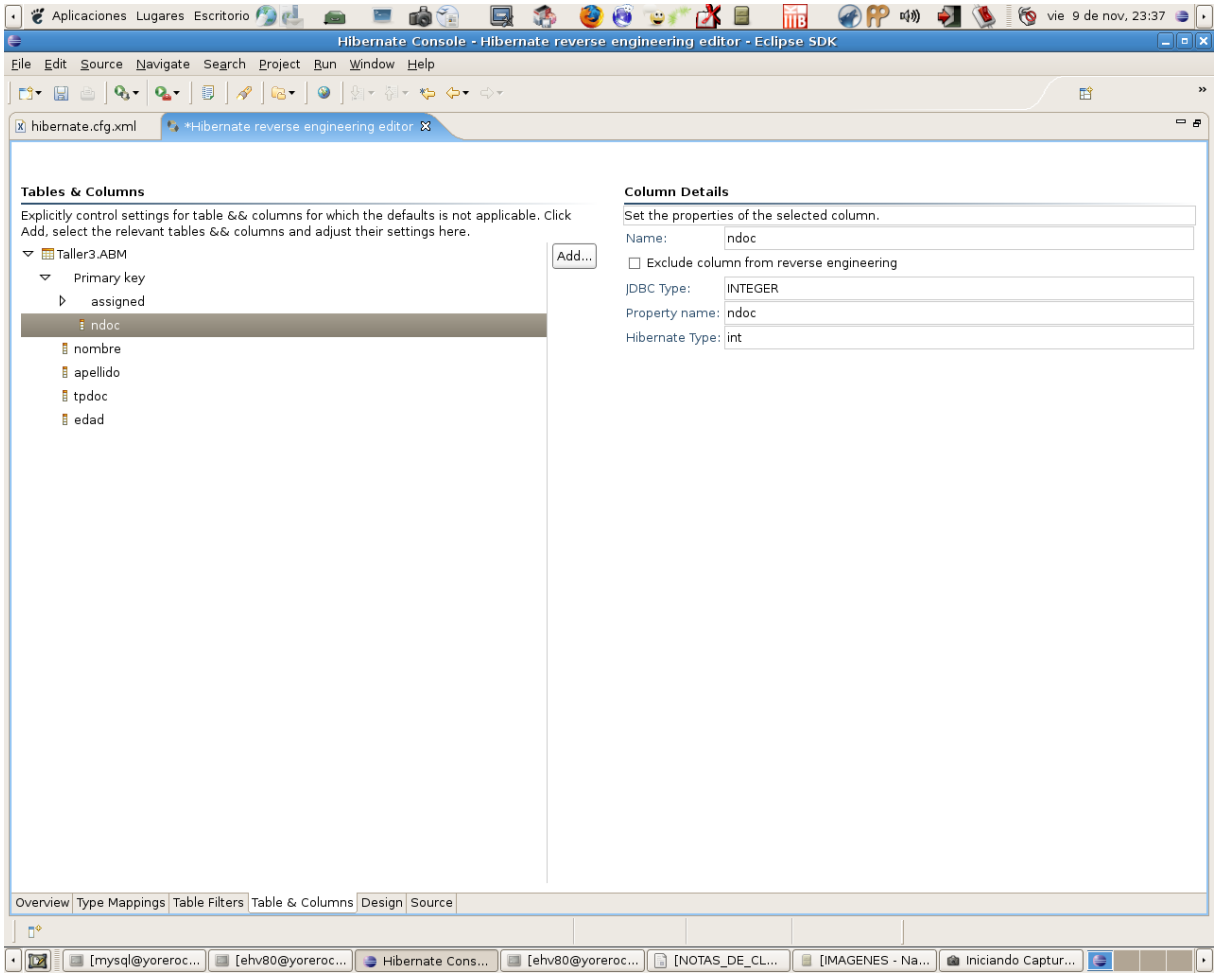
Volvemos a la **sección izquierda:** **“Tables & Columns”**
Para clickear en: **Primary key**

Entonces en la **sección derecha**: **“Primary key details”**
Clickear en el botón: **Add column**

En la **sección izquierda:** **“Tables & Columns”**
Clickear en: **column_1**

Entonces verá en la **sección derecha:** **“Column Details”**
En **“Name”** cambiar el contenido a: **ndoc**
En **“JDBC Type”** introducir: **INTEGER**
En **“Property name”** introducir: **ndoc**
En **“Hibernate Type”** introducir: **int**

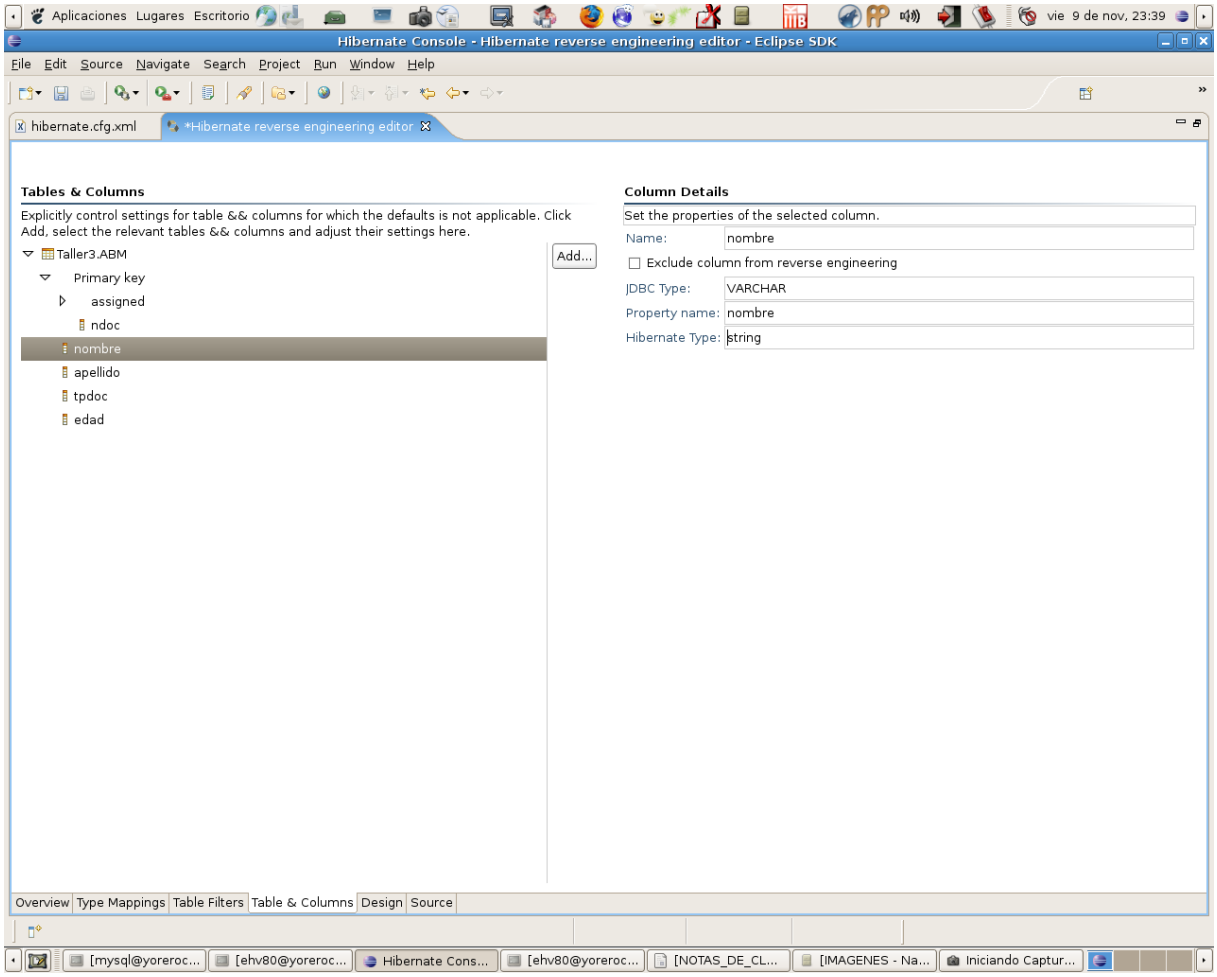
Esta ventana debería mostrar los siguientes detalles:



Luego en el **sección izquierda:** **“Table & Columns”**
Clickear en: **nombre**

Entonces verá en la **sección derecha:** **“Column Details”**
En **“Name”** dejaremos como está: **nombre**
En **“JDBC Type”** introducir: **VARCHAR**
En **“Property Name”** introducir: **nombre**
En **“Hibernate Type”** introducir: **string**

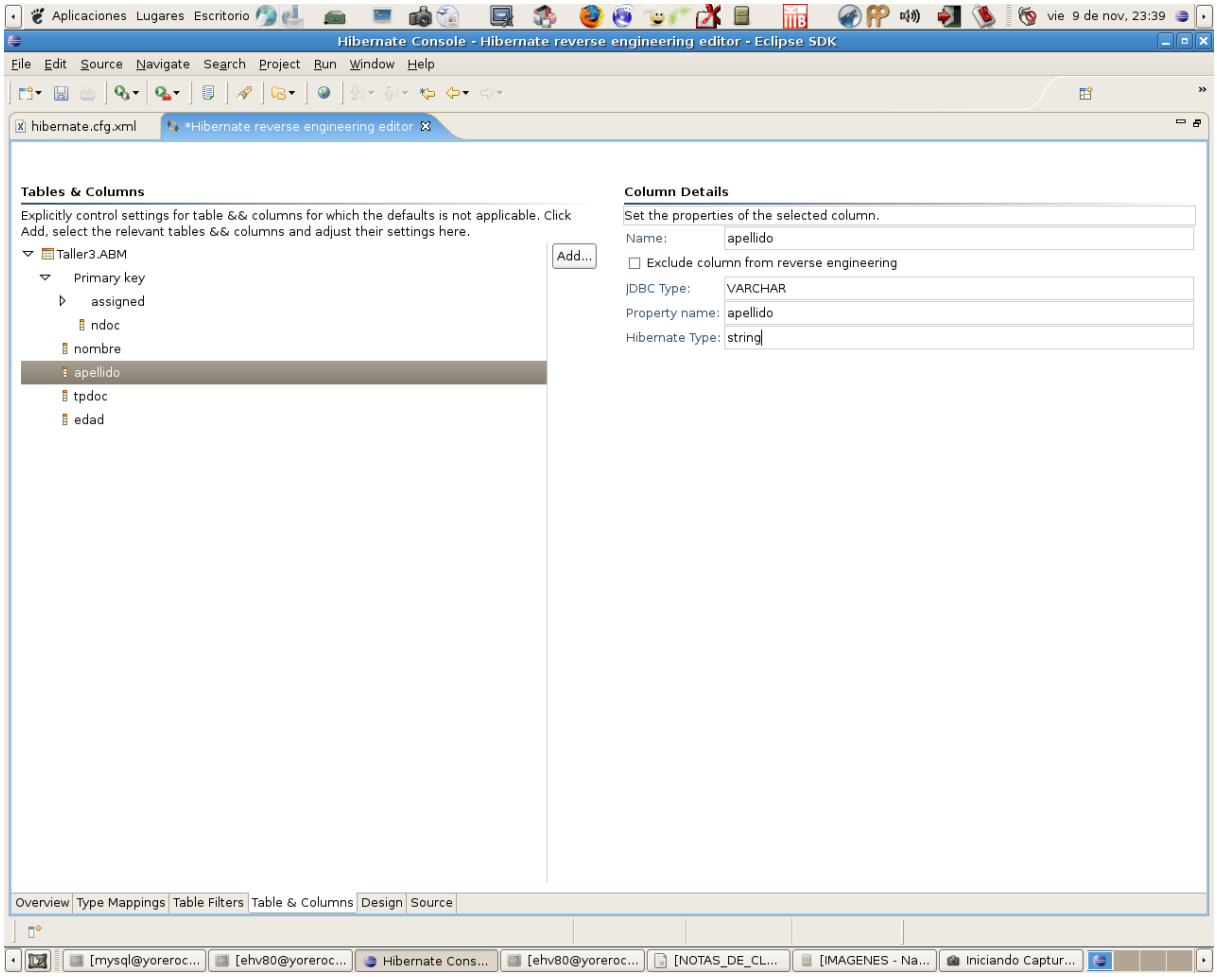
Esta ventana se muestra a continuación:



Luego en la **sección izquierda:** **“Table & Columns”**
Clickear en: **apellido**

La **sección derecha** ahora se llama: **“Column Details”**
En **“Name”** dejaremos como está: **apellido**
En **“JDBC Type”** introducir: **VARCHAR**
En **“Property Name”** introducir: **apellido**
En **“Hibernate Type”** introducir: **string**

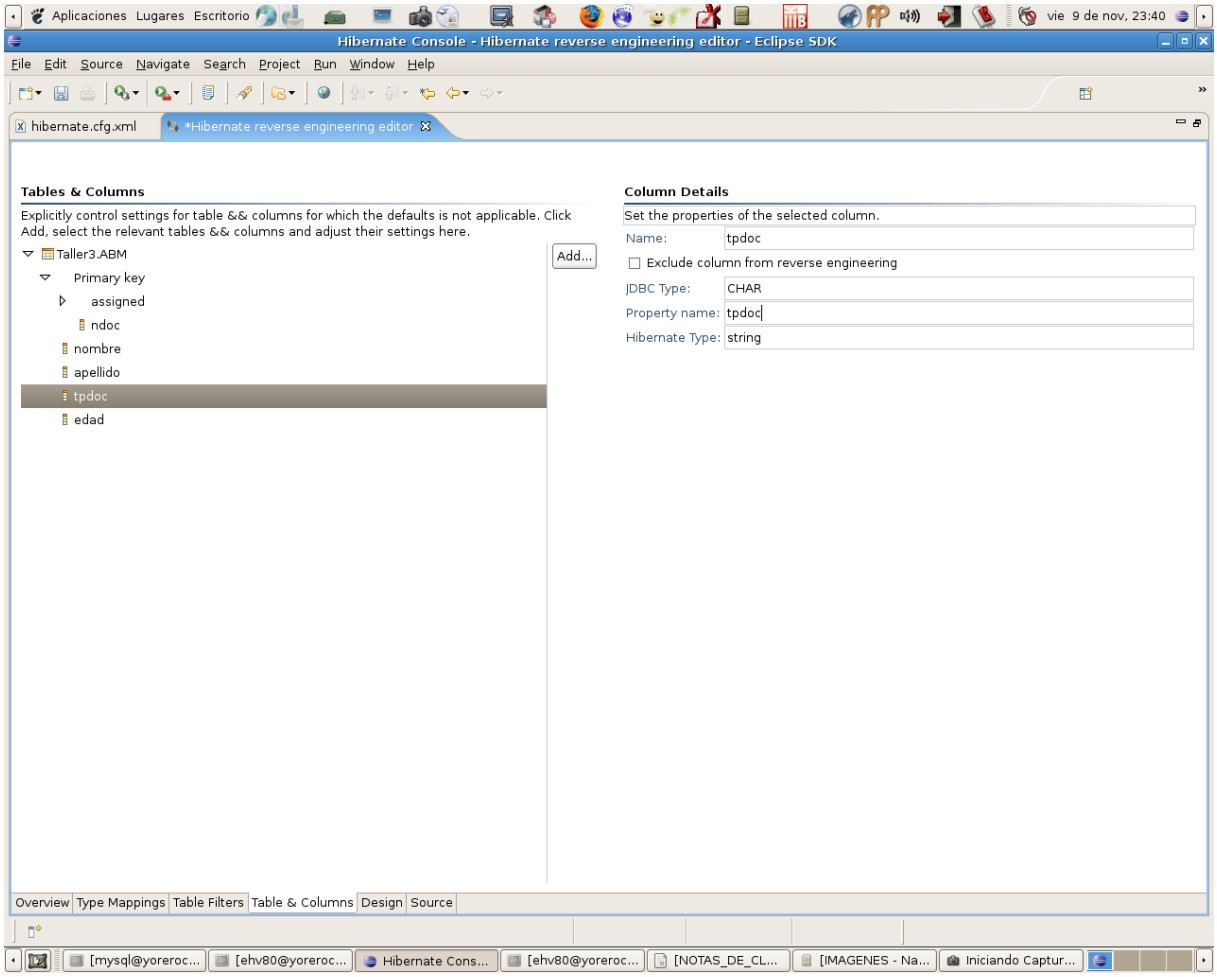
Esta ventana se muestra a continuación:



Luego en la **sección izquierda:** **“Table & Columns”**
Clickear en: **tpdoc**

La **sección derecha** ahora se llama: **“Column Details”**
En **“Name”** dejaremos como está: **tpdoc**
En **“JDBC Type”** introducir: **CHAR**
En **“Property Name”** introducir: **tpdoc**
En **“Hibernate Type”** introducir: **string**

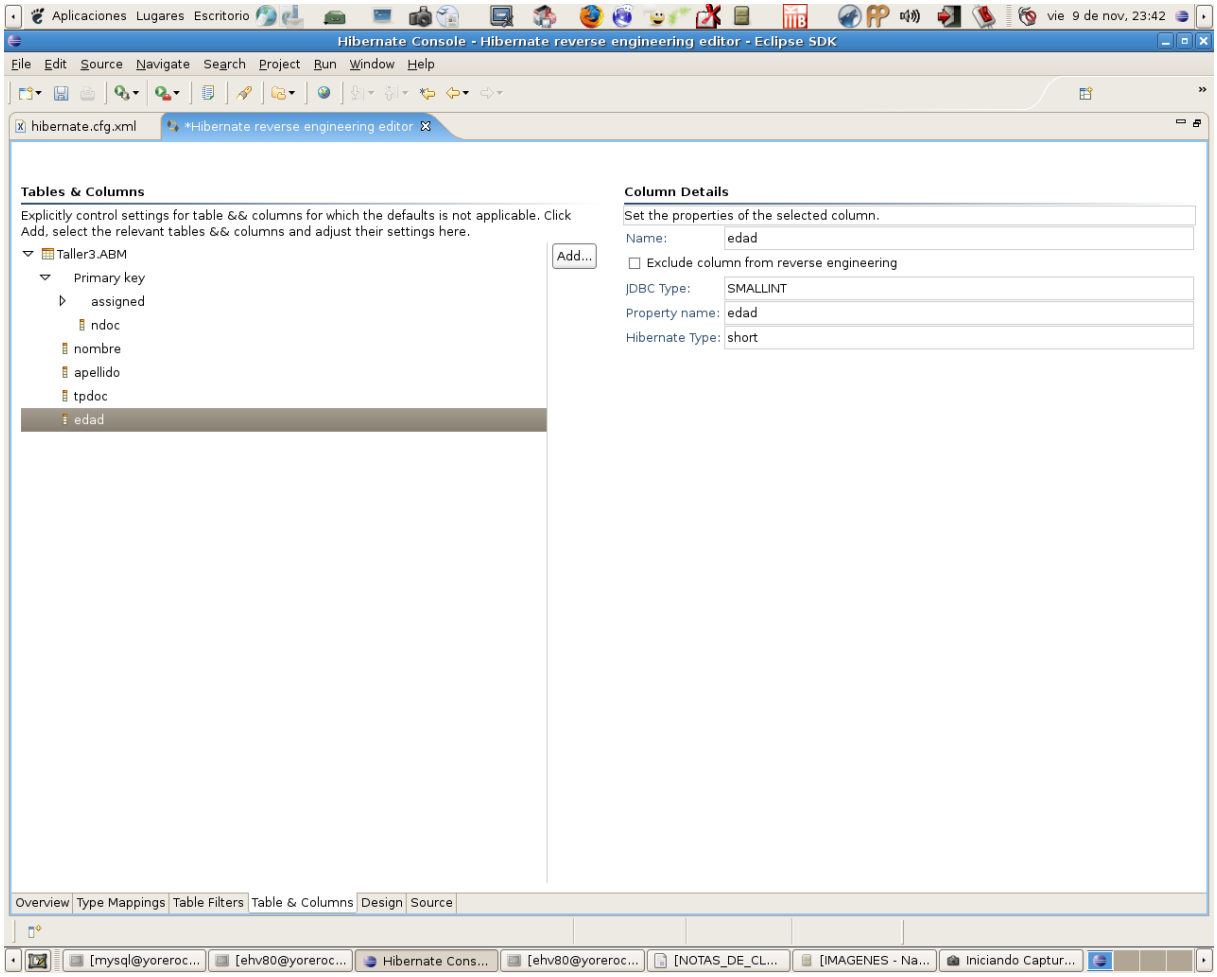
Esta ventana se muestra a continuación:



Luego en la **sección izquierda:** **“Table & Columns”**
Clickear en: **edad**

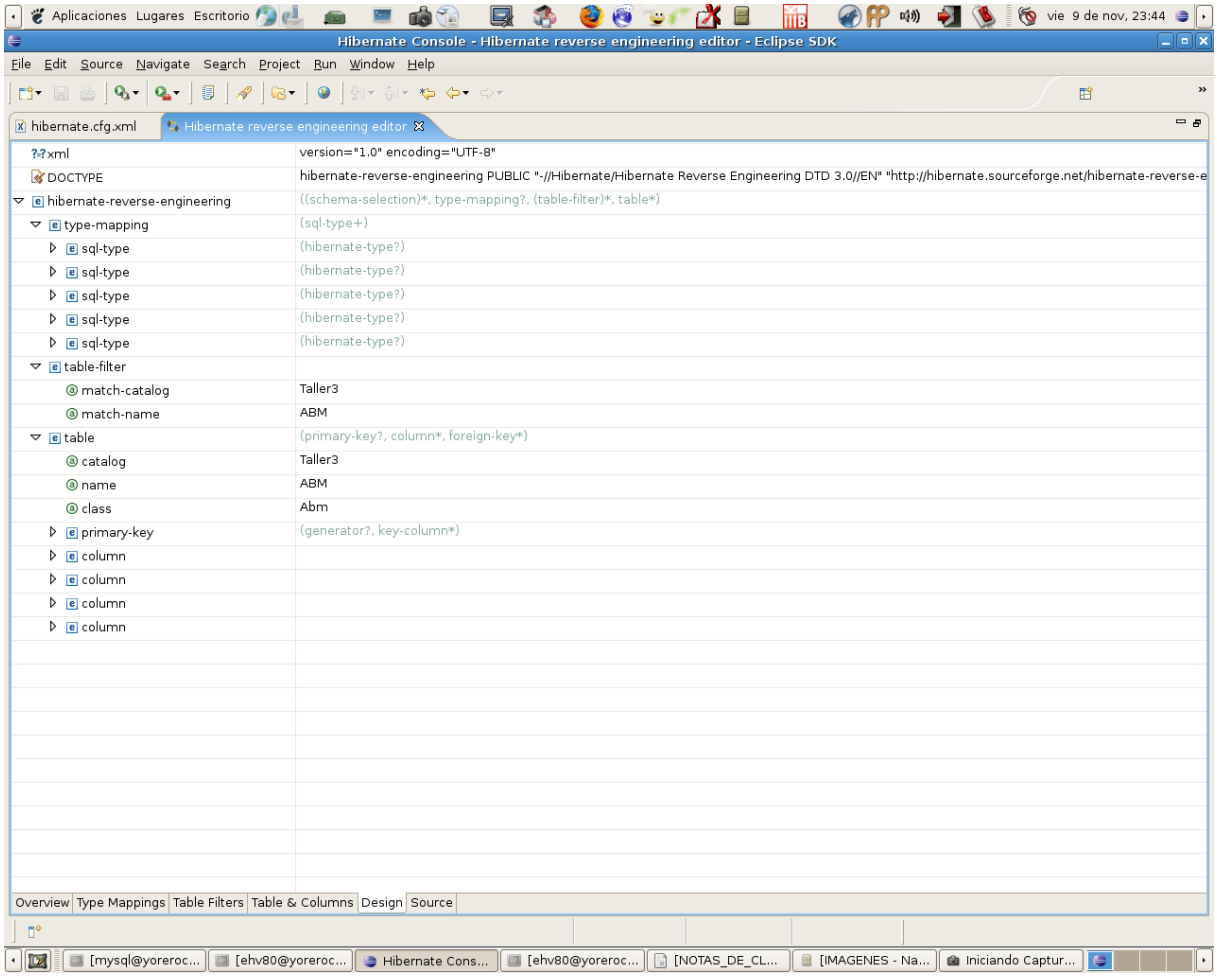
La **sección derecha** ahora se llama: **“Column Details”**
En **“Name”** dejaremos como está: **edad**
En **“JDBC Type”** introducir: **SMALLINT**
En **“Property Name”** introducir: **edad**
En **“Hibernate Type”** introducir: **short**

Esta ventana se muestra a continuación:



En la **solapa interna inferior: “Design”**
Puede verse y editar en forma gráfica la configuración del archivo **“hibernate.reveng.xml”**.

Esta ventana se muestra a continuación:



En la **solapa interna inferior:** **“Source”**

Puede verse y editar el archivo **“hibernate.reveng.xml”**, que luce como el que se muestra a continuación:

```
<?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="int" not-null="true">
            </sql-type>
            <sql-type
                  jdbc-type="VARCHAR" hibernate-type="string"
                  length="30" not-null="true">
            </sql-type>
            <sql-type
                  jdbc-type="VARCHAR" hibernate-type="string"
                  length="30" not-null="true">
            </sql-type>
            <sql-type
                  jdbc-type="CHAR" hibernate-type="string"
                  length="4" not-null="true">
            </sql-type>
            <sql-type
                  jdbc-type="SMALLINT" hibernate-type="short"
                  not-null="true">
            </sql-type>
      </type-mapping>
      <table-filter match-catalog="Taller3" match-name="ABM" />
<table catalog="Taller3" name="ABM" class="Abm">
      <primary-key>
            <generator class="assigned"></generator>
            <key-column
                  name="ndoc" jdbc-type="INTEGER"
                  property="ndoc" type="int"/>
      </primary-key>
      <column
            name="nombre" jdbc-type='VARCHAR'
            property="nombre" type="string"/>
      <column
            name="apellido" jdbc-type='VARCHAR'
            property="apellido" type="string"/>
      <column
            name="tpdoc" jdbc-type="CHAR" type="string"
            property="tpdoc"/>
      <column
            name="edad" jdbc-type='SMALLINT'
            property="edad" type="short"/>
</table>
</hibernate-reverse-engineering>
```

Lo próximo que haremos será guardar todos los cambios hechos al proyecto mediante el menú de **Eclipse**: **File -> Save All**

Al abrir el código fuente **Abm.java** con el editor se observará en la siguiente línea:

```
public class Abm implements java.io.Serializable {
```

la advertencia que se muestra a continuación:

```
The serializable class Abm does not declare a static final
serialVersionUID field of type long
```

Al clickear sobre la misma, se muestran las siguientes alternativas:

- Add default serial version ID
- Add generated serial version ID
- Rename in file (Ctrl+2 R direct acces)
- Change modifiers to final where posible

Aquí **ignoraremos esta advertencia** para evitar problemas con el mapeo entre los campos de los registros de la Base de Datos **“Taller3.ABM”** y los Atributos de los Objetos de la Clase **“Abm.java”.**

De este modo el código fuente de la Clase **Abm.java** queda como se muestra a continuación:

```
      // CLASE Abm.java
import java.util.Vector;

// default package
// Generated 09/11/2007 23:25:04 by Hibernate Tools 3.2.0.beta8

/**
 * Abm generated by hbm2java
 */

public class Abm implements java.io.Serializable {


      // Fields


      /**
       *
       */


      //private static final long serialVersionUID = 1L;

      private int ndoc;

      private String nombre;

      private String apellido;

      private String tpdoc;

      private short edad;
```

```
	// CLASE Abm.java
	// Constructors

	/** default constructor */
	public Abm() {
	}

	/** full constructor */
	public Abm(int ndoc, String nombre, String apellido, String tpdoc,
			short edad) {
		this.ndoc = ndoc;
		this.nombre = nombre;
		this.apellido = apellido;
		this.tpdoc = tpdoc;
		this.edad = edad;
	}

	/**
	 * Constructor adicional. Crea a partir de
	 * un Vector<String> que obtiene de una consulta
	 * HQL a la Base de Datos.
	 * @param datos
	 */
	public Abm( Vector<String> datos ) {
	  this.ndoc = Integer.parseInt(datos.get(0));
	  this.nombre = datos.get(1);
	  this.apellido = datos.get(2);
	  this.tpdoc = datos.get(3);
	  this.edad = Short.parseShort(datos.get(4));
	}

	// Property accessors
	public int getNdoc() {  return this.ndoc; }

	public void setNdoc(int ndoc) {     this.ndoc = ndoc; }

	public String getNombre() {   return this.nombre; }

	public void setNombre(String nombre) {    this.nombre = nombre;   }

	public String getApellido() { return this.apellido;   }

	public void setApellido(String apellido) {this.apellido = apellido;}

	public String getTpdoc() {    return this.tpdoc;      }

	public void setTpdoc(String tpdoc) {      this.tpdoc = tpdoc;     }

	public short getEdad() {      return this.edad; }

	public void setEdad(short edad) {   this.edad = edad; }

}
```

Lo que haremos ahora será guardar todos los cambios mediante las opción del menú de eclipse:

File -> Save All

Vamos a importar los **archivos de librerías:**

- **antlr-2.7.6.jar**
- **asm.jar**
- **cglib-2.1.3.jar**
- **commons-collections-2.1.1.jar**
- **commons-logging-1.0.4.jar**
- **dom4j-1.6.1.jar**
- **hibernate3.jar**
- **jta.jar**

Dirigir el mouse hacia la solapa: **“Package Explorer”**

Clickear con botón derecho del mouse en: **Proyecto_HIBERNATE**

Clickear en: **Build Path -> Add External Archives...**

Examinar el sistema de archivos para indicar las rutas:

-[ruta_a_eclipse]/plugins/org.hibernate.eclipse_3.2.0.beta8/lib/hibernate/
antlr-2.7.6.jar

-[ruta_a_eclipse]/plugins/org.hibernate.eclipse_3.2.0.beta8/lib/hibernate/
asm.jar

-[ruta_a_eclipse]/plugins/org.hibernate.eclipse_3.2.0.beta8/lib/hibernate/
cglib-2.1.3.jar

-[ruta_a_eclipse]/plugins/org.hibernate.eclipse_3.2.0.beta8/lib/hibernate/
commons-collections-2.1.1.jar

-[ruta_a_eclipse]/plugins/org.hibernate.eclipse_3.2.0.beta8/lib/hibernate/
commons-logging-1.0.4.jar

-[ruta_a_eclipse]/plugins/org.hibernate.eclipse_3.2.0.beta8/lib/hibernate/
dom4j-1.6.1.jar

-[ruta_a_eclipse]/plugins/org.hibernate.eclipse_3.2.0.beta8/lib/hibernate/
hibernate3.jar

-[ruta_a_eclipse]/plugins/org.hibernate.eclipse_3.2.0.beta8/lib/hibernate/
jta.jar

Clickear en: **Aceptar**

Luego guardaremos todos los cambios mediante la opción del menú:
File -> Save All

Vamos a crear las clases necesarias, para obtener una aplicación similar a la que obtuvimos con el proyecto: **Proyecto_ABM**, pero usando en este caso las herramientas de **Hibernate**.

Para crear la clase visual: **JABM.java**

Dirigir el mouse hacia la solapa: **"Package Explorer"**

Clickear con botón derecho del mouse sobre: **Proyecto_HIBERNATE**

Clickear en: **New -> Other... -> Visual Class**

En **"Source Folder"** introducir: **Proyecto_HIBERNATE**
En **"Package"** dejar en blanco:
En **"Name"** introducir: **JABM**
En **"modifiers"**tildar : **(*) public**
En **"Style"** elegir: **Swing -> Frame**
En **"Superclass"** verificar. **javax.swing.JFrame**
Tildar en: **[*] public static void main(String[] args)**
Tildar en: **[*] Constructors from superclass**
Tildar en: **[*] inherited abstract methods**
Tildar en: **[*] Generate comments**

Queremos lograr que esta clase visual **JABM** sea idéntica a la clase **Jabm** del proyecto **Proyecto_ABM**. Para ello seguiremos los pasos que vimos en la confección de esta clase visual **Jabm** (**página 1 en adelante**).

Esta clase **JABM** debería tener el siguiente código fuente:

```
import javax.swing.SwingUtilities;
import java.awt.BorderLayout;
import javax.swing.JPanel;
import java.awt.GraphicsConfiguration;
import java.awt.HeadlessException;
import javax.swing.JFrame;
import java.awt.ComponentOrientation;
import java.awt.Dimension;
import javax.swing.JInternalFrame;
import javax.swing.JMenuBar;
import javax.swing.JMenu;
import javax.swing.JMenuItem;
import javax.swing.JDesktopPane;
import javax.swing.SwingConstants;

/** @author ehv80 */
public class JABM extends JFrame {

      private JPanel jContentPaneABM = null;

      private JMenuBar jJMenuBarABM = null;

      private JMenu jMenuABM = null;

      private JMenuItem jMenuItemABMAltas = null;

      private JMenuItem jMenuItemABMBajas = null;

      private JMenuItem jMenuItemABMModificaciones = null;
```

```java
// CLASE JABM.java
private JMenuItem jMenuItemABMSalir = null;

private JDesktopPane jDesktopPaneABM = null;

/** @throws HeadlessException  */
public JABM() throws HeadlessException {
	super();
	initialize();
}

/**	 @param gc  */
public JABM(GraphicsConfiguration gc) {
	super(gc);
	// TODO Auto-generated constructor stub
	initialize();
}

/** @param title	@throws HeadlessException	*/
public JABM(String title) throws HeadlessException {
	super(title);
	initialize();
}

/**	@param title	@param gc	*/
public JABM(String title, GraphicsConfiguration gc) {
	super(title, gc);
	// TODO Auto-generated constructor stub
	initialize();
}

/**	This method initializes jJMenuBarABM
 *	@return javax.swing.JMenuBar
 **/
private JMenuBar getJJMenuBarABM() {
	if (jJMenuBarABM == null) {
		jJMenuBarABM = new JMenuBar();
		jJMenuBarABM.setComponentOrientation(
			ComponentOrientation.LEFT_TO_RIGHT);
		jJMenuBarABM.add(getJMenuABM());
	}
	return jJMenuBarABM;
}

/**	This method initializes jMenuABM
 *	@return javax.swing.JMenu
 */
private JMenu getJMenuABM() {
	if (jMenuABM == null) {
		jMenuABM = new JMenu();
		jMenuABM.setComponentOrientation(
			ComponentOrientation.LEFT_TO_RIGHT);
		jMenuABM.setText("ACCIONES");
		jMenuABM.add(getJMenuItemABMAltas());
		jMenuABM.add(getJMenuItemABMBajas());
		jMenuABM.add(getJMenuItemABMModificaciones());
		jMenuABM.add(getJMenuItemABMSalir());
	}
	return jMenuABM;
}
```

```
// CLASE JABM.java
/**    This method initializes jMenuItemABMAltas
 *     @return javax.swing.JMenuItem
 */
private JMenuItem getJMenuItemABMAltas() {
      if (jMenuItemABMAltas == null)
      {
            jMenuItemABMAltas = new JMenuItem();
            jMenuItemABMAltas.setText("Altas");
            jMenuItemABMAltas.setHorizontalAlignment(
                                    SwingConstants.LEADING);
            jMenuItemABMAltas.setHorizontalTextPosition(
                                    SwingConstants.TRAILING);
            jMenuItemABMAltas.setComponentOrientation(
                                  ComponentOrientation.UNKNOWN);
            jMenuItemABMAltas.addActionListener(
                  new java.awt.event.ActionListener()
                  {
                     public void actionPerformed(
                           java.awt.event.ActionEvent e)
                     {
                        //System.out.println("actionPerformed()");
                        // TODO Auto-generated Event stub
                        // actionPerformed()
                        JInternalFrame[] arregloJIF =
                              jDesktopPaneABM.getAllFrames();
                        boolean existeVentana = false;
                        for(int i = 0 ; i < arregloJIF.length ; i++)
                        {
                              if( arregloJIF[i] instanceof Jaltas)
                              {
                                    arregloJIF[i].toFront();
                                    existeVentana = true;
                              }
                        }
                        if(!existeVentana)
                        {
                              JAltas altas = new JAltas();
                              jDesktopPaneABM.add(altas);
                              altas.setVisible(true);
                        }
                     }
                  });
      }
      return jMenuItemABMAltas;
}
```

```
// CLASE JABM.java
/**    This method initializes jMenuItemABMBajas
 *    @return javax.swing.JMenuItem
 */
private JMenuItem getJMenuItemABMBajas()
{
      if (jMenuItemABMBajas == null)
      {
            jMenuItemABMBajas = new JMenuItem();
            jMenuItemABMBajas.setText("Bajas");
            jMenuItemABMBajas.setComponentOrientation(
                  ComponentOrientation.UNKNOWN);
            jMenuItemABMBajas.addActionListener(
                  new java.awt.event.ActionListener()
                  {
                        public void actionPerformed(
                              java.awt.event.ActionEvent e)
                        {
                              //System.out.println(
                              //"actionPerformed()");
                              // TODO Auto-generated Event stub
                              //actionPerformed()
                              JInternalFrame[] arregloJIF =
                                    jDesktopPaneABM.getAllFrames();
                              boolean existeVentana = false;
                              for(int i=0; i < arregloJIF.length; i++)
                              {
                                 if( arregloJIF[i] instanceof JBajas)
                                 {
                                    arregloJIF[i].toFront();
                                    existeVentana = true;
                                 }
                              }
                              if(!existeVentana)
                              {
                                    JBajas bajas = new JBajas();
                                    jDesktopPaneABM.add(bajas);
                                    bajas.setVisible(true);
                              }
                        }
                  });
      }
      return jMenuItemABMBajas;
}
```

```
// CLASE JABM.java
/**    This method initializes jMenuItemABMModificaciones
 *    @return javax.swing.JMenuItem
 */
private JMenuItem getJMenuItemABMModificaciones()
{
      if (jMenuItemABMModificaciones == null)
      {
            jMenuItemABMModificaciones = new JMenuItem();
            jMenuItemABMModificaciones.setText("Modificaciones");
            jMenuItemABMModificaciones.setComponentOrientation(
                              ComponentOrientation.UNKNOWN);
            jMenuItemABMModificaciones.addActionListener(
                  new java.awt.event.ActionListener()
                  {
                        public void actionPerformed(
                              java.awt.event.ActionEvent e)
                        {
                              //System.out.println
                              //("actionPerformed()");
                              // TODO Auto-generated
                              //Event stub actionPerformed()
                              JInternalFrame[] arregloJIF =
                                    jDesktopPaneABM.getAllFrames();
                              boolean existeVentana = false;
                              for(int i=0; i < arregloJIF.length; i++)
                              {
                                    if( arregloJIF[i] instanceof
                                                      JModificaciones)
                                    {
                                          arregloJIF[i].toFront();
                                          existeVentana = true;
                                    }
                              }
                              if(!existeVentana)
                              {
                                 JModificaciones modificaciones =
                                    new JModificaciones();
                                 jDesktopPaneABM.add(modificaciones);
                                 modificaciones.setVisible(true);
                              }
                        }
                  });
      }
      return jMenuItemABMModificaciones;
}
```

```java
// CLASE JABM.java
/**    This method initializes jMenuItemABMSalir
 *    @return javax.swing.JMenuItem
 */
private JMenuItem getJMenuItemABMSalir()
{
	if (jMenuItemABMSalir == null)
	{
		jMenuItemABMSalir = new JMenuItem();
		jMenuItemABMSalir.setText("Salir");
		jMenuItemABMSalir.setComponentOrientation(
				ComponentOrientation.UNKNOWN);
		jMenuItemABMSalir.addActionListener(
			new java.awt.event.ActionListener()
			{
				public void actionPerformed(
					java.awt.event.ActionEvent e)
				{
				   //System.out.println("actionPerformed()");
				   // TODO Auto-generated Event stub
				   //actionPerformed()
				   dispose();
				}
			}
		);
	}
	return jMenuItemABMSalir;
}

/**   This method initializes jDesktopPaneABM
 *    @return javax.swing.JDesktopPane
 */
private JDesktopPane getJDesktopPaneABM()
{
	if (jDesktopPaneABM == null)
	{
		jDesktopPaneABM = new JDesktopPane();
		jDesktopPaneABM.setComponentOrientation(
				ComponentOrientation.LEFT_TO_RIGHT);
		jDesktopPaneABM.setPreferredSize(new Dimension(200, 100));
	}
	return jDesktopPaneABM;
}

/**     @param args      */
public static void main(String[] args)
{
	// TODO Auto-generated method stub
	SwingUtilities.invokeLater(new Runnable()
	{
		public void run()
		{
			JABM thisClass = new JABM();
			thisClass.setDefaultCloseOperation(
					JFrame.EXIT_ON_CLOSE);
			thisClass.setVisible(true);
		}
	});
}
```

```
	// CLASE JABM.java
	/**	This method initializes this
	 *	@return void
	 */
	private void initialize()
	{
		this.setSize(400, 300);
		this.setPreferredSize(new Dimension(400, 300));
		this.setJMenuBar(getJJMenuBarABM());
		this.setContentPane(getJContentPaneABM());
		this.setTitle("HIBERNATE ABM");
	}

	/**	This method initializes jContentPaneABM
	 *	@return javax.swing.JPanel
	 */
	private JPanel getJContentPaneABM() {
		if (jContentPaneABM == null)
		{
			jContentPaneABM = new JPanel();
			jContentPaneABM.setLayout(new BorderLayout());
			jContentPaneABM.setComponentOrientation(
					ComponentOrientation.LEFT_TO_RIGHT);
			jContentPaneABM.setPreferredSize(new Dimension(200, 100));
			jContentPaneABM.add(getJDesktopPaneABM(),
						BorderLayout.CENTER);
		}
		return jContentPaneABM;
	}

}
// FIN DE LA CLASE: JABM.java
```

Lo que sigue a continuación es agregar la clase encargada de la entrada de datos en la Base de Datos: **JAltas.java**

Para crear La clase visual: **JAltas.java**

Dirigir el mouse hacia la solapa: **"Package Explorer"**

Clickear con botón derecho del mouse sobre: **Proyecto_HIBERNATE**

Clickear en: **New -> Other... -> Visual Class**

En **"Source Folder"** introducir: **Proyecto_HIBERNATE**
En **"Package"** dejaremos en blanco:
En **"Name"** introducir: **JAltas**
En **"modifiers"**tildar : **(*) public**
En **"Style"** elegir: **Swing -> InternalFrame**
En **"Superclass"** verificar. **javax.swing.JInternalFrame**
Tildar en: **[*] public static void main(String[] args)**
Tildar en: **[*] Constructors from superclass**
Tildar en: **[*] inherited abstract methods**
Tildar en: **[*] Generate comments**

Queremos lograr que esta clase visual **JAltas** sea idéntica a la clase **JAltas** del proyecto **Proyecto_ABM**. Para ello seguiremos los pasos que vimos en la confección de esta clase visual **JAltas** (**página 6 en adelante**).

Esta clase **JAltas** debería tener el siguiente código fuente:

```
//CLASE JAltas.java
import java.awt.BorderLayout;
import javax.swing.JPanel;
import javax.swing.JInternalFrame;
import javax.swing.BorderFactory;
import javax.swing.border.EtchedBorder;
import javax.swing.JDesktopPane;
import javax.swing.JLabel;
import java.awt.Dimension;
import java.awt.Point;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.SwingConstants;
import javax.swing.JComboBox;
import javax.swing.JTextField;
import javax.swing.JButton;

/**  @author ehv80 */
public class JAltas extends JinternalFrame
{
      private JPanel jContentPaneAltas = null;
      private JDesktopPane jDesktopPaneAltas = null;
      private JLabel jLabelAltasNombre = null;
      private JLabel jLabelAltasApellido = null;
      private JComboBox jComboBoxAltasTipoDocumento = null;
      private JLabel jLabelAltasEdad = null;
      private JTextField jTextFieldAltasNombre = null;
      private JTextField jTextFieldAltasApellido = null;
      private JTextField jTextFieldAltasDocumento = null;
```

```
	private JTextField jTextFieldAltasEdad = null;
	//CLASE JAltas.java
	private JButton jButtonAltasGrabar = null;
	private JButton jButtonAltasSalir = null;

	public Jaltas()
	{
		super();
		initialize();
	}

	/**			@param title	*/
	public JAltas(String title)
	{
		super(title);
		initialize();
	}

	/**		@param title
	 *		@param resizable
	 */
	public JAltas(String title, boolean resizable)
	{
		super(title, resizable);
		initialize();
	}

	/**	@param title
	 *	@param resizable
	 *	@param closable
	 */
	public JAltas(String title, boolean resizable, boolean closable)
	{
		super(title, resizable, closable);
		initialize();
	}

	/**	@param title
	 *	@param resizable
	 *	@param closable
	 *	@param maximizable
	 */
	public JAltas(String title, boolean resizable,
			boolean closable, boolean maximizable)
	{
		super(title, resizable, closable, maximizable);
		initialize();
	}

	/**	@param title
	 *	@param resizable
	 *	@param closable
	 *	@param maximizable
	 *	@param iconifiable
	 */
	public JAltas(String title, boolean resizable, boolean closable,
				boolean maximizable, boolean iconifiable)
	{
		super(title, resizable, closable, maximizable, iconifiable);
		initialize();
	}
```

```
//CLASE JAltas.java
/**    This method initializes jDesktopPaneAltas
 *    @return javax.swing.JDesktopPane
 */
private JDesktopPane getJDesktopPaneAltas()
{
      if (jDesktopPaneAltas == null)
      {
            jLabelAltasEdad = new JLabel();
            jLabelAltasEdad.setText("EDAD");
            jLabelAltasEdad.setLocation(new Point(25, 105));
            jLabelAltasEdad.setPreferredSize(new Dimension(80, 20));
            jLabelAltasEdad.setHorizontalTextPosition(
                                    SwingConstants.CENTER);
            jLabelAltasEdad.setHorizontalAlignment(
                                    SwingConstants.CENTER);
            jLabelAltasEdad.setSize(new Dimension(80, 20));
            jLabelAltasApellido = new JLabel();
            jLabelAltasApellido.setHorizontalAlignment(
                                    SwingConstants.CENTER);
            jLabelAltasApellido.setHorizontalTextPosition(
                                          SwingConstants.CENTER);
            jLabelAltasApellido.setSize(new Dimension(80, 20));
            jLabelAltasApellido.setLocation(new Point(25, 45));
            jLabelAltasApellido.setText("APELLIDO");
            jLabelAltasNombre = new JLabel();
            jLabelAltasNombre.setText("NOMBRE");
            jLabelAltasNombre.setLocation(new Point(25, 15));
            jLabelAltasNombre.setToolTipText("");
            jLabelAltasNombre.setHorizontalTextPosition(
                                    SwingConstants.CENTER);
            jLabelAltasNombre.setHorizontalAlignment(
                                    SwingConstants.CENTER);
            jLabelAltasNombre.setPreferredSize(new Dimension(80, 20));
            jLabelAltasNombre.setSize(new Dimension(80, 20));
            jDesktopPaneAltas = new JDesktopPane();
            jDesktopPaneAltas.setPreferredSize(
                                          new Dimension(300, 200));
            jDesktopPaneAltas.add(jLabelAltasNombre, null);
            jDesktopPaneAltas.add(jLabelAltasApellido, null);
            jDesktopPaneAltas.add(
                  getJComboBoxAltasTipoDocumento(), null);
            jDesktopPaneAltas.add(jLabelAltasEdad, null);
            jDesktopPaneAltas.add(getJTextFieldAltasNombre(), null);
            jDesktopPaneAltas.add(getJTextFieldAltasApellido(), null);
            jDesktopPaneAltas.add(
                        getJTextFieldAltasDocumento(), null);
            jDesktopPaneAltas.add(getJTextFieldAltasEdad(), null);
            jDesktopPaneAltas.add(getJButtonAltasGrabar(), null);
            jDesktopPaneAltas.add(getJButtonAltasSalir(), null);
      }
      return jDesktopPaneAltas;
}
```

```java
//CLASE JAltas.java
/**    This method initializes jComboBoxAltasTipoDocumento
 *     @return javax.swing.JComboBox
 */
private JComboBox getJComboBoxAltasTipoDocumento()
{
      if (jComboBoxAltasTipoDocumento == null)
      {
            jComboBoxAltasTipoDocumento = new JComboBox();
            jComboBoxAltasTipoDocumento.setSize(
                                    new Dimension(80, 20));
            jComboBoxAltasTipoDocumento.setPreferredSize(
                                    new Dimension(80, 20));
            jComboBoxAltasTipoDocumento.setLocation(
                                    new Point(25, 75));
            jComboBoxAltasTipoDocumento.addItem("DNI");
            jComboBoxAltasTipoDocumento.addItem("LC");
            jComboBoxAltasTipoDocumento.addItem("LE");
      }
      return jComboBoxAltasTipoDocumento;
}

/**    This method initializes jTextFieldAltasNombre
 *     @return javax.swing.JTextField
 */
private JTextField getJTextFieldAltasNombre()
{
      if (jTextFieldAltasNombre == null)
      {
            jTextFieldAltasNombre = new JTextField();
            jTextFieldAltasNombre.setSize(new Dimension(140, 20));
            jTextFieldAltasNombre.setPreferredSize(
                              new Dimension(140, 20));
            jTextFieldAltasNombre.setToolTipText(
                              "Ingrese el Nombre...");
            jTextFieldAltasNombre.setLocation(new Point(115, 15));
      }
      return jTextFieldAltasNombre;
}

/**    This method initializes jTextFieldAltasApellido
 *     @return javax.swing.JTextField
 */
private JTextField getJTextFieldAltasApellido()
{
      if (jTextFieldAltasApellido == null)
      {
            jTextFieldAltasApellido = new JTextField();
            jTextFieldAltasApellido.setSize(new Dimension(140, 20));
            jTextFieldAltasApellido.setPreferredSize(
                                    new Dimension(140, 20));
            jTextFieldAltasApellido.setToolTipText(
                                    "Ingrese el Apellido...");
            jTextFieldAltasApellido.setLocation(new Point(115, 45));
      }
      return jTextFieldAltasApellido;
}
```

```java
//CLASE JAltas.java
/**   This method initializes jTextFieldAltasDocumento
 *    @return javax.swing.JTextField
 */
private JTextField getJTextFieldAltasDocumento()
{
      if (jTextFieldAltasDocumento == null)
      {
            jTextFieldAltasDocumento = new JTextField();
            jTextFieldAltasDocumento.setSize(new Dimension(140, 20));
            jTextFieldAltasDocumento.setPreferredSize(
                                    new Dimension(140, 20));
            jTextFieldAltasDocumento.setToolTipText(
               "Introduzca el número de documento (sin puntos)...");
            jTextFieldAltasDocumento.setLocation(new Point(115, 75));
      }
      return jTextFieldAltasDocumento;
}

/**   This method initializes jTextFieldAltasEdad
 *    @return javax.swing.JTextField
 */
private JTextField getJTextFieldAltasEdad()
{
      if (jTextFieldAltasEdad == null)
      {
            jTextFieldAltasEdad = new JTextField();
            jTextFieldAltasEdad.setSize(new Dimension(140, 20));
            jTextFieldAltasEdad.setPreferredSize(
                                 new Dimension(140, 20));
            jTextFieldAltasEdad.setToolTipText("Ingrese la edad...");
            jTextFieldAltasEdad.setLocation(new Point(115, 105));
      }
      return jTextFieldAltasEdad;
}
```

```
//CLASE JAltas.java
/**    This method initializes jButtonAltasGrabar
 *    @return javax.swing.JButton
 */
private JButton getJButtonAltasGrabar()
{
   if (jButtonAltasGrabar == null)
   {
      jButtonAltasGrabar = new JButton();
      jButtonAltasGrabar.setSize(new Dimension(90, 20));
      jButtonAltasGrabar.setPreferredSize(
            new Dimension(90, 20));
      jButtonAltasGrabar.setText("GRABAR");
      jButtonAltasGrabar.setHorizontalTextPosition(
                              SwingConstants.CENTER);
      jButtonAltasGrabar.setLocation(new Point(2, 135));
      jButtonAltasGrabar.addActionListener(
         new java.awt.event.ActionListener()
         {
            public void actionPerformed(
                  java.awt.event.ActionEvent e)
            {
            //System.out.println("actionPerformed()");
            // TODO Auto-generated Event stubactionPerformed()
               if( jTextFieldAltasNombre.getText().equals("") ||
                  jTextFieldAltasApellido.getText().equals("") ||
                  jComboBoxAltasTipoDocumento.getSelectedItem().equals("")||
                  jTextFieldAltasDocumento.getText().equals("") ||
                  jTextFieldAltasEdad.getText().equals("")
                 )
               {
                  JOptionPane.showMessageDialog (new JFrame(),
                        "Por favor verifique si ha ingresado
                        correctamente todos los datos...",
                        "ADVERTENCIA..!",
                        JoptionPane.INFORMATION_MESSAGE
                  );
               }
               else
               {
                  Abm abmAlta;
                  try
                  {
                     abmAlta = new Abm(
                     Integer.parseInt(jTextFieldAltasDocumento.getText()),
                        jTextFieldAltasNombre.getText() ,
                        jTextFieldAltasApellido.getText() ,
                      (String)jComboBoxAltasTipoDocumento.getSelectedItem(),
                        Short.parseShort( jTextFieldAltasEdad.getText() )
                     );
                     HibernateUtil.conectar();//?
                     HibernateUtil.guardarAbm(abmAlta);//?
                  }
                  catch(Exception ex){
                     System.err.println(
                        "Ha ocurrido una Excepción al instanciar un objeto
                           Abm " + ex);
                     JOptionPane.showMessageDialog(
                        new JFrame(),
                        "Un error muy extraÃ±o ha ocurrido... \nMejor,
                           cierre la aplicación...!",
                        "A la perinola... !!!",
                     JOptionPane.INFORMATION_MESSAGE  );
```

```
                }
        //CLASE JAltas.java
                finally{
                        //TODO
                        jTextFieldAltasNombre.setText("");
                        jTextFieldAltasApellido.setText("");
                        jTextFieldAltasDocumento.setText("");
                        jTextFieldAltasEdad.setText("");
                        abmAlta = null;
                }
            }
        }
      });
   }
        return jButtonAltasGrabar;
}

/**   This method initializes jButtonAltasSalir
 *    @return javax.swing.JButton
 */
private JButton getJButtonAltasSalir() {
        if (jButtonAltasSalir == null) {
                jButtonAltasSalir = new JButton();
                jButtonAltasSalir.setSize(new Dimension(90, 20));
                jButtonAltasSalir.setHorizontalTextPosition(
                                        SwingConstants.CENTER);
                jButtonAltasSalir.setText("SALIR");
                jButtonAltasSalir.setPreferredSize(new Dimension(90, 20));
                jButtonAltasSalir.setLocation(new Point(193, 135));
                jButtonAltasSalir.addActionListener(
                   new java.awt.event.ActionListener()
                   {
                      public void actionPerformed(java.awt.event.ActionEvent e)
                      {
                            //System.out.println("actionPerformed()");
                            // TODO Auto-generated Event stub actionPerformed()
                            if(HibernateUtil.getSession() != null &&
                                  HibernateUtil.getSessionFactory() != null)
                            {
                                  HibernateUtil.desconectar();//?
                            }
                            dispose();
                      }
                   }
                );
        }
        return jButtonAltasSalir;
}

/**   @param args  */
public static void main(String[] args)
{
        // TODO Auto-generated method stub
}
```

```
//CLASE JAltas.java
/**   This method initializes this
 *    @return void
 */
private void initialize()
{
      this.setSize(300, 200);
      this.setResizable(true);
      this.setMaximizable(true);
      this.setClosable(true);
      this.setTitle("HIBERNATE ALTAS");
      this.setContentPane(getJContentPaneAltas());
}

/**   This method initializes jContentPaneAltas
 *    @return javax.swing.JPanel
 */
private JPanel getJContentPaneAltas()
{
      if (jContentPaneAltas == null)
      {
            jContentPaneAltas = new JPanel();
            jContentPaneAltas.setLayout(new BorderLayout());
            jContentPaneAltas.setBorder(
                  BorderFactory.createEtchedBorder(EtchedBorder.LOWERED));
            jContentPaneAltas.setSize(new Dimension(300, 200));
            jContentPaneAltas.setPreferredSize(new Dimension(300, 200));
            jContentPaneAltas.add(getJDesktopPaneAltas(),
                                          BorderLayout.CENTER);
      }
      return jContentPaneAltas;
   }

}
//FIN DE LA CLASE JAltas.java
```

Vamos a crear otra de las clases necesarias: **JModificaciones.java**

Dirigir el mouse hacia la solapa: **"Package Explorer"**

Clickear con botón derecho del mouse sobre: **Proyecto_HIBERNATE**

Clickear en: **New -> Other... -> Visual Class**

En **"Source Folder"** introducir: **Proyecto_HIBERNATE**
En **"Package"** dejar en blanco:
En **"Name"** introducir: **JModificaciones**
En **"modifiers"**tildar : **(*) public**
En **"Style"** elegir: **Swing -> InternalFrame**
En **"Superclass"** verificar. **javax.swing.JInternalFrame**
Tildar en: **[*] public static void main(String[] args)**
Tildar en: **[*] Constructors from superclass**
Tildar en: **[*] inherited abstract methods**
Tildar en: **[*] Generate comments**

Queremos lograr que esta clase visual **JModificaciones** sea idéntica a la clase **JModificaciones** del proyecto **Proyecto_ABM**. Para ello seguiremos los pasos que vimos en la confección de esta clase visual **JModificaciones** (**página 11 en adelante**).

Esta clase **JModificaciones** debería tener el siguiente código fuente:

```
// CLASE JModificaciones.java
import java.awt.BorderLayout;
import javax.swing.JPanel;
import javax.swing.JInternalFrame;
import javax.swing.BorderFactory;
import javax.swing.border.EtchedBorder;
import javax.swing.JDesktopPane;
import java.awt.Dimension;
import java.awt.Point;
import java.util.Vector;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.SwingConstants;
import javax.swing.JComboBox;
import javax.swing.JTextField;
import javax.swing.JButton;

/**   @author ehv80     */
public class JModificaciones extends JInternalFrame {
      private JPanel jContentPaneModificaciones = null;
      private JDesktopPane jDesktopPaneModificaciones = null;
      private JLabel jLabelModificacionesNombre = null;
      private JLabel jLabelModificacionesApellido = null;
      private JComboBox jComboBoxModificacionesTipoDocumento = null;
      private JLabel jLabelModificacionesEdad = null;
      private JTextField jTextFieldModificacionesNombre = null;
      private JTextField jTextFieldModificacionesApellido = null;
      private JTextField jTextFieldModificacionesDocumento = null;
      private JTextField jTextFieldModificacionesEdad = null;
      private JButton jButtonModificacionesGrabar = null;
      private JButton jButtonModificacionesBuscar = null;
      private JButton jButtonModificacionesSalir = null;
```

```java
// CLASE JModificaciones.java
	public JModificaciones() {
		super();
		initialize();
	}
	/** @param title   */
	public JModificaciones(String title) {
		super(title);
		initialize();
	}
	/**   @param title
	 *    @param resizable
	 */
	public JModificaciones(String title, boolean resizable) {
		super(title, resizable);
		initialize();
	}
	/**   @param title
	 *    @param resizable
	 *    @param closable
	 */
	public JModificaciones(String title, boolean resizable,
				boolean closable) {
		super(title, resizable, closable);
		initialize();
	}
	/**   @param title
	 *    @param resizable
	 *    @param closable
	 *    @param maximizable
	 */
	public JModificaciones(String title, boolean resizable,
				boolean closable, boolean maximizable) {
		super(title, resizable, closable, maximizable);
		initialize();
	}
	/**   @param title
	 *    @param resizable
	 *    @param closable
	 *    @param maximizable
	 *    @param iconifiable
	 */
	public JModificaciones(String title, boolean resizable,
		boolean closable, boolean maximizable, boolean iconifiable) {
		super(title, resizable, closable, maximizable, iconifiable);
		initialize();
	}
	/**   Método: cargaDatos
	*     @param Vector <String> datos
	*     @return void                          */
	public void cargaDatos(Vector <String> datos)
	{
	   setVisible(true);
	   // Orden según Abm.hbm.xml
	   jTextFieldModificacionesDocumento.setText(datos.get(0));
	   jTextFieldModificacionesNombre.setText(datos.get(1));
	   jTextFieldModificacionesApellido.setText(datos.get(2));
	   jComboBoxModificacionesTipoDocumento.setSelectedItem(datos.get(3));
	   jTextFieldModificacionesEdad.setText(datos.get(4));
	}
```

```
// CLASE JModificaciones.java
      /**   This method initializes jDesktopPaneModificaciones
       *    @return javax.swing.JDesktopPane           */
   private JDesktopPane getJDesktopPaneModificaciones() {
      if (jDesktopPaneModificaciones == null)
      {
         jLabelModificacionesEdad = new JLabel();
         jLabelModificacionesEdad.setText("EDAD");
         jLabelModificacionesEdad.setLocation(new Point(25, 105));
         jLabelModificacionesEdad.setHorizontalAlignment(
            SwingConstants.CENTER);
         jLabelModificacionesEdad.setHorizontalTextPosition(
            SwingConstants.CENTER);
         jLabelModificacionesEdad.setPreferredSize(new Dimension(80, 20));
         jLabelModificacionesEdad.setSize(new Dimension(80, 20));
         jLabelModificacionesApellido = new JLabel();
         jLabelModificacionesApellido.setText("APELLIDO");
         jLabelModificacionesApellido.setSize(new Dimension(80, 20));
        jLabelModificacionesApellido.setPreferredSize(new Dimension(80,20));
         jLabelModificacionesApellido.setHorizontalTextPosition(
            SwingConstants.CENTER);
         jLabelModificacionesApellido.setHorizontalAlignment(
            SwingConstants.CENTER);
         jLabelModificacionesApellido.setLocation(new Point(25, 45));
         jLabelModificacionesNombre = new JLabel();
         jLabelModificacionesNombre.setText("NOMBRE");
         jLabelModificacionesNombre.setLocation(new Point(25, 15));
         jLabelModificacionesNombre.setPreferredSize(new Dimension(80, 20));
         jLabelModificacionesNombre.setHorizontalTextPosition(
            SwingConstants.CENTER);
         jLabelModificacionesNombre.setHorizontalAlignment(
            SwingConstants.CENTER);
         jLabelModificacionesNombre.setSize(new Dimension(80, 20));
         jDesktopPaneModificaciones = new JDesktopPane();
         jDesktopPaneModificaciones.add(jLabelModificacionesNombre, null);
         jDesktopPaneModificaciones.add(jLabelModificacionesApellido, null);
         jDesktopPaneModificaciones.add(
            getJComboBoxModificacionesTipoDocumento(), null);
         jDesktopPaneModificaciones.add(jLabelModificacionesEdad, null);
         jDesktopPaneModificaciones.add(
            getJTextFieldModificacionesNombre(), null);
         jDesktopPaneModificaciones.add(
            getJTextFieldModificacionesApellido(), null);
         jDesktopPaneModificaciones.add(
            getJTextFieldModificacionesDocumento(), null);
         jDesktopPaneModificaciones.add(
            getJTextFieldModificacionesEdad(), null);
         jDesktopPaneModificaciones.add(
            getJButtonModificacionesGrabar(), null);
         jDesktopPaneModificaciones.add(
            getJButtonModificacionesBuscar(), null);
         jDesktopPaneModificaciones.add(
            getJButtonModificacionesSalir(), null);
         }
      return jDesktopPaneModificaciones;
   }
```

```
// CLASE JModificaciones.java
      /**   This method initializes jComboBoxModificacionesTipoDocumento
       *    @return javax.swing.JComboBox        */
      private JComboBox getJComboBoxModificacionesTipoDocumento() {
         if (jComboBoxModificacionesTipoDocumento == null)
         {
            jComboBoxModificacionesTipoDocumento = new JComboBox();
            jComboBoxModificacionesTipoDocumento.setSize(
                  new Dimension(80, 20));
            jComboBoxModificacionesTipoDocumento.setPreferredSize(
                  new Dimension(80, 20));
            jComboBoxModificacionesTipoDocumento.setLocation(
                  new Point(25, 75));
            jComboBoxModificacionesTipoDocumento.addItem("DNI");
            jComboBoxModificacionesTipoDocumento.addItem("LC");
            jComboBoxModificacionesTipoDocumento.addItem("LE");
         }
         return jComboBoxModificacionesTipoDocumento;
      }
      /**   This method initializes jTextFieldModificacionesNombre
       *    @return javax.swing.JTextField             */
      private JTextField getJTextFieldModificacionesNombre() {
         if (jTextFieldModificacionesNombre == null)
         {
            jTextFieldModificacionesNombre = new JTextField();
            jTextFieldModificacionesNombre.setLocation(
                  new Point(115, 15));
            jTextFieldModificacionesNombre.setPreferredSize(
                  new Dimension(140, 20));
            jTextFieldModificacionesNombre.setSize(new Dimension(140, 20));
         }
         return jTextFieldModificacionesNombre;
      }
      /**    This method initializes jTextFieldModificacionesApellido
       *     @return javax.swing.JTextField            */
      private JTextField getJTextFieldModificacionesApellido() {
         if (jTextFieldModificacionesApellido == null)
         {
            jTextFieldModificacionesApellido = new JTextField();
            jTextFieldModificacionesApellido.setSize(
                  new Dimension(140, 20));
            jTextFieldModificacionesApellido.setPreferredSize(
                  new Dimension(140, 20));
            jTextFieldModificacionesApellido.setLocation(
                  new Point(115, 45));
         }
         return jTextFieldModificacionesApellido;
      }
      /**    This method initializes jTextFieldModificacionesDocumento
       *    @return javax.swing.JTextField             */
      private JTextField getJTextFieldModificacionesDocumento() {
         if (jTextFieldModificacionesDocumento == null)
         {
            jTextFieldModificacionesDocumento = new JTextField();
            jTextFieldModificacionesDocumento.setSize(
                  new Dimension(140, 20));
            jTextFieldModificacionesDocumento.setPreferredSize(
                  new Dimension(140, 20));
            jTextFieldModificacionesDocumento.setLocation(
                  new Point(115, 75));
         }
         return jTextFieldModificacionesDocumento;
      }
```

```java
// CLASE JModificaciones.java
/**   This method initializes jTextFieldModificacionesEdad
 *    @return javax.swing.JTextField             */
private JTextField getJTextFieldModificacionesEdad()
{
   if (jTextFieldModificacionesEdad == null)
   {
      jTextFieldModificacionesEdad = new JTextField();
      jTextFieldModificacionesEdad.setSize(new Dimension(140, 20));
      jTextFieldModificacionesEdad.setPreferredSize(
            new Dimension(140, 20));
      jTextFieldModificacionesEdad.setLocation(new Point(115, 105));
   }
   return jTextFieldModificacionesEdad;
}
```

```
// CLASE JModificaciones.java
      /**   This method initializes jButtonModificacionesGrabar
       *    @return javax.swing.JButton          */
      private JButton getJButtonModificacionesGrabar()
      {
         if (jButtonModificacionesGrabar == null)
         {
            jButtonModificacionesGrabar = new JButton();
            jButtonModificacionesGrabar.setText("GRABAR");
            jButtonModificacionesGrabar.setSize(new Dimension(90, 20));
            jButtonModificacionesGrabar.setHorizontalTextPosition(
                  SwingConstants.CENTER);
            jButtonModificacionesGrabar.setLocation(new Point(2, 135));
            jButtonModificacionesGrabar.addActionListener(
               new java.awt.event.ActionListener()
               {
                  public void actionPerformed(java.awt.event.ActionEvent e)
                  {
                     //System.out.println("actionPerformed()");
                     // TODO Auto-generated Event stub actionPerformed()

                    if(jTextFieldModificacionesNombre.getText().equals("")||
                     jTextFieldModificacionesApellido.getText().equals("")||
                     ((String)(jComboBoxModificacionesTipoDocumento.
                                          getSelectedItem())).equals("")||
                    jTextFieldModificacionesDocumento.getText().equals("")||
                     jTextFieldModificacionesEdad.getText().equals("")
                    )
                    {
                        JOptionPane.showMessageDialog(
                              new JFrame(),
                              "Asegúrese de introducir todos los datos
                               correctamente, o presione en \"BUSCAR\"..!",
                              "ADVERTENCIA..!",
                              JOptionPane.INFORMATION_MESSAGE     );
                    }
                    else
                    {
                       Abm abmModificable;
                       try
                       {
                          HibernateUtil.conectar(); //?
                          abmModificable = new Abm(
                             Integer.parseInt(
                               jTextFieldModificacionesDocumento.getText()),
                             jTextFieldModificacionesNombre.getText() ,
                             jTextFieldModificacionesApellido.getText() ,
                             (String)(jComboBoxModificacionesTipoDocumento.
                                getSelectedItem()) ,
                             Short.parseShort(jTextFieldModificacionesEdad.
                                getText())
                          );
                          HibernateUtil.actualizarAbm(abmModificable);
                       }
                       catch(Exception ex){
                          System.err.println("Ha ocurrido una Excepción al
                             actualizar un objeto Abm " + ex);
                          JoptionPane.showMessageDialog( new JFrame(),
                             "Error al actualizar la información de la Base
                                 de Datos...!",
                             "Advertencia...!",
                             JOptionPane.ERROR_MESSAGE
                          );
                       }
```

```
// CLASE JModificaciones.java
                        finally
                        {
                                jTextFieldModificacionesNombre.setText("");
                                jTextFieldModificacionesApellido.setText("");
                                jTextFieldModificacionesDocumento.setText("");
                                jTextFieldModificacionesEdad.setText("");
                                abmModificable = null;
                        }
                      }
                    }
                  }
              );
           }
           return jButtonModificacionesGrabar;
        }
        /**   This method initializes jButtonModificacionesBuscar
         *    @return javax.swing.JButton          */
        private JButton getJButtonModificacionesBuscar()
        {
           if (jButtonModificacionesBuscar == null)
           {
              jButtonModificacionesBuscar = new JButton();
              jButtonModificacionesBuscar.setHorizontalTextPosition(
                    SwingConstants.CENTER);
              jButtonModificacionesBuscar.setLocation(new Point(97, 135));
              jButtonModificacionesBuscar.setPreferredSize(
                    new Dimension(90, 20));
              jButtonModificacionesBuscar.setText("BUSCAR");
              jButtonModificacionesBuscar.setSize(new Dimension(90, 20));
              jButtonModificacionesBuscar.addActionListener(
               new java.awt.event.ActionListener()
               {
                public void actionPerformed(java.awt.event.ActionEvent e)
                {
                  //System.out.println("actionPerformed()");
                  // TODO Auto-generated Event stub actionPerformed()
                  String sentenciaHQL="select p from Abm as p where 1=1";
                  if(!jTextFieldModificacionesNombre.getText().equals(""))
                    sentenciaHQL += " and p.nombre='" +
                    jTextFieldModificacionesNombre.getText() + "' ";
                  if( !jTextFieldModificacionesApellido.getText().equals("") )
                    sentenciaHQL += " and p.apellido='" +
                    jTextFieldModificacionesApellido.getText() + "' ";
                  if(!jTextFieldModificacionesDocumento.getText().equals("") )
                    sentenciaHQL += " and p.ndoc='" +
                    jTextFieldModificacionesDocumento.getText() + "' ";
                  if( !jTextFieldModificacionesEdad.getText().equals("") )
                    sentenciaHQL += " and p.edad='" +
                    jTextFieldModificacionesEdad.getText() + "' ";
                  //?
                  JResultadoConsulta resultadoConsulta;
                  JSeleccion ventanaSeleccion;
                  try
                  {
                    HibernateUtil.conectar();
                    resultadoConsulta = HibernateUtil.consulta(sentenciaHQL);
                    ventanaSeleccion = new JSeleccion(resultadoConsulta,
                       getEstaVentana() );
                    jDesktopPaneModificaciones.add(ventanaSeleccion);
                    ventanaSeleccion.setVisible(true);
                    //HibernateUtil.desconectar(); //?
                    //setVisible(true);
                  }
```

```
// CLASE JModificaciones.java
                catch(Exception ex)
                {
                  System.err.println("Ha ocurrido una Excepción al obtener
                     la información desde la Base de Datos " + ex);
                  JOptionPane.showMessageDialog(   new JFrame(),
                   "Error al obtener la información de la Base de Datos..!",
                   "ADVERTENCIA..!",
                   JOptionPane.ERROR_MESSAGE
                  );
                }
                finally
                {
                  //TODO
                }
              }
             }
            );
         }
         return jButtonModificacionesBuscar;
      }
      public JModificaciones getEstaVentana() {  return this;     }
      /**    This method initializes jButtonModificacionesSalir
       *     @return javax.swing.JButton         */
      private JButton getJButtonModificacionesSalir()
      {
         if (jButtonModificacionesSalir == null)
         {
            jButtonModificacionesSalir = new JButton();
            jButtonModificacionesSalir.setSize(new Dimension(90, 20));
            jButtonModificacionesSalir.setText("SALIR");
            jButtonModificacionesSalir.setLocation(new Point(193, 135));
            jButtonModificacionesSalir.addActionListener(
               new java.awt.event.ActionListener()
               {
                  public void actionPerformed(java.awt.event.ActionEvent e)
                  {
                     //System.out.println("actionPerformed()");
                     // TODO Auto-generated Event stub actionPerformed()
                     if( HibernateUtil.getSession() != null &&
                        HibernateUtil.getSessionFactory() != null )
                     {
                        HibernateUtil.desconectar();
                     }
                     dispose();
                  }
               }
            );
         }
         return jButtonModificacionesSalir;
      }
      /**   @param args        */
      public static void main(String[] args) {}
      /**    This method initializes this
       *     @return void            */
      private void initialize() {
            this.setSize(300, 200);
            this.setClosable(true);
            this.setMaximizable(true);
            this.setResizable(true);
            this.setTitle("HIBERNATE MODIFICACIONES");
            this.setContentPane(getJContentPaneModificaciones());
      }
```

```
// CLASE JModificaciones.java
      /**    This method initializes jContentPaneModificaciones
       *     @return javax.swing.JPanel          */
      private JPanel getJContentPaneModificaciones()
      {
         if (jContentPaneModificaciones == null)
         {
            jContentPaneModificaciones = new JPanel();
            jContentPaneModificaciones.setLayout(new BorderLayout());
            jContentPaneModificaciones.setBorder(
               BorderFactory.createEtchedBorder(EtchedBorder.LOWERED));
            jContentPaneModificaciones.add(getJDesktopPaneModificaciones(),
               BorderLayout.CENTER);
         }
         return jContentPaneModificaciones;
      }
}
// FIN DE LA CLASE JModificaciones.java
```

Para crear otra clase visual: **JBajas.java**

Dirigir el mouse hacia la solapa: **"Package Explorer"**

Clickear con botón derecho del mouse sobre: **Proyecto_HIBERNATE**

Clickear en: **New -> Other... -> Visual Class**

En **"Source Folder"** introducir: **Proyecto_HIBERNATE**
En **"Package"** introducir:
En **"Name"** introducir: **JBajas**
En **"modifiers"**tildar : **(*) public**
En **"Style"** elegir: **Swing ->InternalFrame**
En **"Superclass"** verificar. **javax.swing.JInternalFrame**
Tildar en: **[*] public static void main(String[] args)**
Tildar en: **[*] Constructors from superclass**
Tildar en: **[*] inherited abstract methods**
Tildar en: **[*] Generate comments**

Queremos lograr que esta clase visual **JBajas** sea idéntica a la clase **JBajas** del proyecto **Proyecto_ABM**. Para ello seguiremos los pasos que vimos en la confección de esta clase visual **JBajas** (**página 14 en adelante**).

Esta clase **JBajas** debería contener el siguiente código fuente:

```
      //CLASE JBajas.java
import java.awt.BorderLayout;
import javax.swing.JPanel;
import javax.swing.JInternalFrame;
import javax.swing.JDesktopPane;
import javax.swing.JLabel;
import java.awt.Dimension;
import java.awt.Point;
import java.util.Vector;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.SwingConstants;
import javax.swing.JComboBox;
import javax.swing.JTextField;
import javax.swing.JButton;
import javax.swing.BorderFactory;
import javax.swing.border.EtchedBorder;
/**   @author ehv80      */
public class JBajas extends JInternalFrame {
      private JPanel jContentPaneBajas = null;
      private JDesktopPane jDesktopPaneBajas = null;
      private JLabel jLabelBajasNombre = null;
      private JLabel jLabelBajasApellido = null;
      private JComboBox jComboBoxBajasTipoDocumento = null;
      private JLabel jLabelBajasEdad = null;
      private JTextField jTextFieldBajasNombre = null;
      private JTextField jTextFieldBajasApellido = null;
      private JTextField jTextFieldBajasDocumento = null;
      private JTextField jTextFieldBajasEdad = null;
      private JButton jButtonBajasBorrar = null;
      private JButton jButtonBajasBuscar = null;
      private JButton jButtonBajasSalir = null;
      public JBajas() {
            super();
            initialize();
      }
      /**   @param title       */
      public JBajas(String title) {
            super(title);
            initialize();
      }
      /**   @param title
       *    @param resizable         */
      public JBajas(String title, boolean resizable) {
            super(title, resizable);
            initialize();
      }
      /**   @param title
       *    @param resizable
       *    @param closable    */
      public JBajas(String title, boolean resizable, boolean closable) {
            super(title, resizable, closable);
            initialize();
      }
      /**   @param title
       *    @param resizable
       *    @param closable
       *    @param maximizable       */
      public JBajas(String title, boolean resizable,
                  boolean closable, boolean maximizable) {
            super(title, resizable, closable, maximizable);
            initialize();
      }
```

```
    //CLASE JBajas.java
    /**   @param title
     *    @param resizable
     *    @param closable
     *    @param maximizable
     *    @param iconifiable       */
    public JBajas(String title, boolean resizable, boolean closable,
          boolean maximizable, boolean iconifiable) {
          super(title, resizable, closable, maximizable, iconifiable);
          initialize();
    }
    /** Método:     cargaDatos
   * Toma los datos que estÃin almacenados temporalmente
   * en el Vector de String "datos" y los introduce e los
   * campos de texto correspondientes de la interface del
   * objeto instancia de la clase visual JBajas.java.
   *
   * @param Vector <String> datos
   * @return void
   */
  public void cargaDatos(Vector <String> datos){
          setVisible(true);
          //El órden está dado por Abm.hbm.xml
          jTextFieldBajasDocumento.setText(datos.get(0));
          jTextFieldBajasNombre.setText(datos.get(1));
          jTextFieldBajasApellido.setText(datos.get(2));
          jComboBoxBajasTipoDocumento.setSelectedItem(datos.get(3));
          jTextFieldBajasEdad.setText(datos.get(4));
  }
    /**   This method initializes jDesktopPaneBajas
     *    @return javax.swing.JDesktopPane           */
    private JDesktopPane getJDesktopPaneBajas()
    {
       if (jDesktopPaneBajas == null) {
          jLabelBajasEdad = new JLabel();
          jLabelBajasEdad.setText("EDAD");
          jLabelBajasEdad.setLocation(new Point(25, 105));
          jLabelBajasEdad.setPreferredSize(new Dimension(80, 20));
          jLabelBajasEdad.setHorizontalAlignment(SwingConstants.CENTER);
          jLabelBajasEdad.setHorizontalTextPosition(
             SwingConstants.CENTER);
          jLabelBajasEdad.setSize(new Dimension(80, 20));
          jLabelBajasApellido = new JLabel();
          jLabelBajasApellido.setText("APELLIDO");
          jLabelBajasApellido.setLocation(new Point(25, 45));
          jLabelBajasApellido.setPreferredSize(new Dimension(80, 20));
          jLabelBajasApellido.setHorizontalAlignment(
             SwingConstants.CENTER);
          jLabelBajasApellido.setHorizontalTextPosition(
             SwingConstants.CENTER);
          jLabelBajasApellido.setSize(new Dimension(80, 20));
          jLabelBajasNombre = new JLabel();
          jLabelBajasNombre.setText("NOMBRE");
          jLabelBajasNombre.setLocation(new Point(25, 15));
          jLabelBajasNombre.setHorizontalAlignment(
             SwingConstants.CENTER);
          jLabelBajasNombre.setHorizontalTextPosition(
             SwingConstants.CENTER);
          jLabelBajasNombre.setPreferredSize(new Dimension(80, 20));
          jLabelBajasNombre.setSize(new Dimension(80, 20));
          jDesktopPaneBajas = new JDesktopPane();
          jDesktopPaneBajas.add(jLabelBajasNombre, null);
          jDesktopPaneBajas.add(jLabelBajasApellido, null);
          jDesktopPaneBajas.add(getJComboBoxBajasTipoDocumento(), null);
```

```
//CLASE JBajas.java
      jDesktopPaneBajas.add(jLabelBajasEdad, null);
      jDesktopPaneBajas.add(getJTextFieldBajasNombre(), null);
      jDesktopPaneBajas.add(getJTextFieldBajasApellido(), null);
      jDesktopPaneBajas.add(getJTextFieldBajasDocumento(), null);
      jDesktopPaneBajas.add(getJTextFieldBajasEdad(), null);
      jDesktopPaneBajas.add(getJButtonBajasBorrar(), null);
      jDesktopPaneBajas.add(getJButtonBajasBuscar(), null);
      jDesktopPaneBajas.add(getJButtonBajasSalir(), null);
   }
   return jDesktopPaneBajas;
}
/**    This method initializes jComboBoxBajasTipoDocumento
 *    @return javax.swing.JComboBox        */
private JComboBox getJComboBoxBajasTipoDocumento()
{
   if (jComboBoxBajasTipoDocumento == null)
   {
      jComboBoxBajasTipoDocumento = new JComboBox();
      jComboBoxBajasTipoDocumento.setSize(new Dimension(80, 20));
      jComboBoxBajasTipoDocumento.setLocation(new Point(25, 75));
      jComboBoxBajasTipoDocumento.addItem("DNI");
      jComboBoxBajasTipoDocumento.addItem("LC");
      jComboBoxBajasTipoDocumento.addItem("LE");
   }
   return jComboBoxBajasTipoDocumento;
}
/**   This method initializes jTextFieldBajasNombre
 *    @return javax.swing.JTextField              */
private JTextField getJTextFieldBajasNombre() {
      if (jTextFieldBajasNombre == null) {
            jTextFieldBajasNombre = new JTextField();
            jTextFieldBajasNombre.setSize(new Dimension(140, 20));
            jTextFieldBajasNombre.setLocation(new Point(115, 15));
      }
      return jTextFieldBajasNombre;
}
/**   This method initializes jTextFieldBajasApellido
 *    @return javax.swing.JTextField              */
private JTextField getJTextFieldBajasApellido()
{
   if (jTextFieldBajasApellido == null)
   {
      jTextFieldBajasApellido = new JTextField();
      jTextFieldBajasApellido.setSize(new Dimension(140, 20));
      jTextFieldBajasApellido.setPreferredSize(
         new Dimension(140, 20));
      jTextFieldBajasApellido.setLocation(new Point(115, 45));
   }
   return jTextFieldBajasApellido;
}
/**   This method initializes jTextFieldBajasDocumento
 *    @return javax.swing.JTextField              */
private JTextField getJTextFieldBajasDocumento()
{
   if (jTextFieldBajasDocumento == null)
   {
      jTextFieldBajasDocumento = new JTextField();
      jTextFieldBajasDocumento.setSize(new Dimension(140, 20));
      jTextFieldBajasDocumento.setLocation(new Point(115, 75));
   }
   return jTextFieldBajasDocumento;
}
```

```
//CLASE JBajas.java
/**    This method initializes jTextFieldBajasEdad
 *    @return javax.swing.JTextField             */
private JTextField getJTextFieldBajasEdad()
{
   if (jTextFieldBajasEdad == null)
   {
      jTextFieldBajasEdad = new JTextField();
      jTextFieldBajasEdad.setSize(new Dimension(140, 20));
      jTextFieldBajasEdad.setPreferredSize(new Dimension(140, 20));
      jTextFieldBajasEdad.setLocation(new Point(115, 105));
   }
   return jTextFieldBajasEdad;
}
/**   This method initializes jButtonBajasBorrar
 *    @return javax.swing.JButton          */
private JButton getJButtonBajasBorrar()
{
   if (jButtonBajasBorrar == null)
   {
      jButtonBajasBorrar = new JButton();
      jButtonBajasBorrar.setSize(new Dimension(90, 20));
      jButtonBajasBorrar.setPreferredSize(new Dimension(90, 20));
      jButtonBajasBorrar.setText("BORRAR");
      jButtonBajasBorrar.setHorizontalTextPosition(
         SwingConstants.CENTER);
      jButtonBajasBorrar.setLocation(new Point(2, 135));
      jButtonBajasBorrar.addActionListener(
         new java.awt.event.ActionListener()
         {
            public void actionPerformed(java.awt.event.ActionEvent e)
            {
               //System.out.println("actionPerformed()");
               // TODO Auto-generated Event stub actionPerformed()
               if(jTextFieldBajasNombre.getText().equals("") ||
                  jTextFieldBajasApellido.getText().equals("") ||
                  ((String)
                     (jComboBoxBajasTipoDocumento.getSelectedItem()))
                        .equals("") ||
                  jTextFieldBajasDocumento.getText().equals("") ||
                  jTextFieldBajasEdad.getText().equals("")
               )
               {
                  JoptionPane.showMessageDialog(   new JFrame(),
                        "Asegúrese de introducir todos los datos
                        correctamente, o presione en \"BUSCAR\"..!",
                        "ADVERTENCIA..!",
                        JoptionPane.INFORMATION_MESSAGE   );
               }
               else{
                  Abm abmBaja;
                  try{
                     HibernateUtil.conectar(); //?
                     abmBaja = new Abm(   Integer.parseInt(
                           jTextFieldBajasDocumento.getText()) ,
                        jTextFieldBajasNombre.getText() ,
                        jTextFieldBajasApellido.getText() ,
                        (String)(jComboBoxBajasTipoDocumento.
                                          getSelectedItem() ) ,
                        Short.parseShort(jTextFieldBajasEdad.
                                                getText())
                     );
                     HibernateUtil.eliminarAbm(abmBaja);
                  }
```

```
//CLASE JBajas.java
                    catch(Exception ex){
                       System.err.println("Ha ocurrido una Excepción al
                          eliminar un objeto Abm " + ex);
                       JOptionPane.showMessageDialog(   new JFrame(),
                          "Error al eliminar la información de la Base
                          de Datos...!",
                          "Advertencia...!",
                          JOptionPane.ERROR_MESSAGE
                       );
                    }
                    finally{
                          jTextFieldBajasNombre.setText("");
                          jTextFieldBajasApellido.setText("");
                          jTextFieldBajasDocumento.setText("");
                          jTextFieldBajasEdad.setText("");
                          abmBaja = null;
                    }
                 }
              }
           }
        );
     }
     return jButtonBajasBorrar;
  }
  /**   This method initializes jButtonBajasBuscar
   *    @return javax.swing.JButton          */
  private JButton getJButtonBajasBuscar(){
     if (jButtonBajasBuscar == null) {
        jButtonBajasBuscar = new JButton();
        jButtonBajasBuscar.setSize(new Dimension(90, 20));
        jButtonBajasBuscar.setHorizontalTextPosition(
           SwingConstants.CENTER);
        jButtonBajasBuscar.setText("BUSCAR");
        jButtonBajasBuscar.setLocation(new Point(97, 135));
        jButtonBajasBuscar.addActionListener(
           new java.awt.event.ActionListener()
           {
              public void actionPerformed(java.awt.event.ActionEvent e)
              {
                 String sentenciaHQL="select p from Abm as p where 1=1";
                 if( !jTextFieldBajasNombre.getText().equals("") )
                    sentenciaHQL += " and p.nombre='" +
                    jTextFieldBajasNombre.getText() + "' ";
                 if( !jTextFieldBajasApellido.getText().equals("") )
                    sentenciaHQL += " and p.apellido='" +
                    jTextFieldBajasApellido.getText() + "' ";
                 if( !jTextFieldBajasDocumento.getText().equals("") )
                    sentenciaHQL += " and p.ndoc='" +
                    jTextFieldBajasDocumento.getText() + "' ";
                 if( !jTextFieldBajasEdad.getText().equals("") )
                    sentenciaHQL += " and p.edad='" +
                    jTextFieldBajasEdad.getText() + "' ";
                 JResultadoConsulta resultadoConsulta;
                 JSeleccion ventanaSeleccion;
                 try{
                  HibernateUtil.conectar();
                  resultadoConsulta = HibernateUtil.consulta(sentenciaHQL);
                  ventanaSeleccion = new JSeleccion(resultadoConsulta,
                    getEstaVentana() );
                  jDesktopPaneBajas.add(ventanaSeleccion);
                  ventanaSeleccion.setVisible(true);
                 }
```

```
//CLASE JBajas.java
            catch(Exception ex){
             System.err.println("Ha ocurrido una Excepción al obtener
                la información desde la Base de Datos " + ex);
             JOptionPane.showMessageDialog(     new JFrame(),
                "Error al obtener la información de la Base
                   deDatos..!",
                "ADVERTENCIA..!",
                JOptionPane.ERROR_MESSAGE );
             }
             finally{
                //TODO
             }
            }
         }
      );
   }
   return jButtonBajasBuscar;
}
/**   This method initializes jButtonBajasSalir
 *    @return javax.swing.JButton          */
private JButton getJButtonBajasSalir() {
   if (jButtonBajasSalir == null) {
      jButtonBajasSalir = new JButton();
      jButtonBajasSalir.setSize(new Dimension(90, 20));
      jButtonBajasSalir.setPreferredSize(new Dimension(90, 20));
      jButtonBajasSalir.setHorizontalTextPosition(
         SwingConstants.CENTER);
      jButtonBajasSalir.setText("SALIR");
      jButtonBajasSalir.setLocation(new Point(193, 135));
      jButtonBajasSalir.addActionListener(
         new java.awt.event.ActionListener()
         {
            public void actionPerformed(java.awt.event.ActionEvent e)
            {
               //System.out.println("actionPerformed()");
               // TODO Auto-generated Event stub actionPerformed()
               if( HibernateUtil.getSession() != null &&
                  HibernateUtil.getSessionFactory() != null )
               {
                   HibernateUtil.desconectar();
               }
               dispose();
            }
         }
      );
   }
   return jButtonBajasSalir;
}
/**   @param args        */
public static void main(String[] args) {  }
/**   This method initializes this
 *    @return void             */
private void initialize() {
      this.setSize(300, 200);
      this.setMaximizable(true);
      this.setClosable(true);
      this.setResizable(true);
      this.setTitle("HIBERNATE BAJAS");
      this.setContentPane(getJContentPaneBajas());
}
```

```
        //CLASE JBajas.java
        /**   This method initializes jContentPaneBajas
         *    @return javax.swing.JPanel           */
        private JPanel getJContentPaneBajas() {
           if (jContentPaneBajas == null) {
              jContentPaneBajas = new JPanel();
              jContentPaneBajas.setLayout(new BorderLayout());
              jContentPaneBajas.setBorder(
                 BorderFactory.createEtchedBorder(EtchedBorder.LOWERED));
              jContentPaneBajas.add(getJDesktopPaneBajas(),
              BorderLayout.CENTER);
           }
           return jContentPaneBajas;
        }
        /**   Retorna la Ventana Actual JBajas.
         *    @return JBajas           */
        public JBajas getEstaVentana(){     return this;      }

}
//FIN DE LA CLASE: JBajas.java
```

Vamos a agregar otra nueva clase al proyecto: **Proyecto_Hibernate.** Tendrá la misma función que la Clase **JConexión.java** del proyecto **Proyecto_ABM**, la llamaremos: **HibernateUtil.java**. Esta clase nos permitirá manipular la información que hay en la Base de Datos mediante unos métodos estáticos que encapsularán algunos de los métodos propios de la Clase **"Session".**

Para agregar la clase, no visual, **HibernateUtil.java**
Dirigir el mouse hacia la solapa: **"Package Explorer"**
Clickear con el botón derecho del mouse en: **Proyecto_Hibernate**
Clickear en: **New -> Class**
En "Source folder" introducir: **Proyecto_Hibernate**
En "Package" dejar en blanco:
En "Name" introducir: **HibernateUtil**
En "Modifiers" tildar: **(*) public**
En "Superclass" verificar: **java.lang.Object**
Tildar: **public static void main(String[] args)**
Tildar: **Constructors from superclass**
Tildar: **Inherited abstract methods**
Tildar: **Generate comments**
Clickear en: **Finish**

Editaremos el código fuente **HibernateUtil.java** mediante el Editor para que luzca como se muestra a continuación:

```
//CLASE HibernateUtil.java
import java.util.List;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import org.hibernate.HibernateException;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
/**   @author ehv80     */
public class HibernateUtil {
      private static SessionFactory sessionFactory;
      private static Session session;
```

```
//CLASE HibernateUtil.java
      /** Inicializa el único SessionFactory para toda la aplicación.
       * @param void
       * @return void          */
      public static void createSessionFactory()
      {
         try{
            sessionFactory =
            new Configuration().configure().buildSessionFactory();
         }
         catch(HibernateException ex){
            System.err.println("Ha ocurrido una Excepción al inicializar el
               SessionFactory "+ ex);
         }
      }
      /**   Finaliza el único SessionFactory para toda la aplicación.
       * @param void
       * @return void          */
      public static void closeSessionFactory()
      {
            try{  sessionFactory.close(); }
            catch(HibernateException ex)
            {
               System.err.println("Ha ocurrido una Excepción al cerrar el
                  SessionFactory " + ex);
            }
      }
      /**   @param void
       *    @return SessionFactory         */
      public static SessionFactory getSessionFactory()
      {           return sessionFactory;
      }
      /**   Inicializa estáticamente la Session.
       * @param void
       * @return void          */
      public static void createSession()
      {
            try{
                  if(sessionFactory != null )
                  {     session = sessionFactory.openSession();
                  }
            }
            catch(HibernateException ex)
            {     System.err.println("Ha ocurrido una ExcepciÃ³n al
                     inicializar estáticamente la Session " + ex);
            }
      }
      /**   Cierra estáticamente la Session.
       * @param void
       * @return void          */
      public static void closeSession()
      {
            try{  session.close();  }
            catch(HibernateException ex)
            {
                  System.err.println("Ha ocurrido una Excepción al cerrar
                     estáticamente la Session " + ex);
            }
      }
      /**   Retorna la Session.
       *    @return Session.   */
      public static Session getSession() {      return session;   }
```

```
//CLASE HibernateUtil.java

	/** Conecta mediante createSessionFactory y createSession.   */
	public static void conectar()
	{
		createSessionFactory();
		createSession();
	}

	/** Desconecta mediante  closeSession y closeSessionFactory.  */
	public static void desconectar()
	{
		closeSession();
		closeSessionFactory();
	}

	/** Almacena objeto Abm en la Base de Datos: Taller3
	 *  haciendo uso del método "save" de Session.
	 *  @param AbmAlta       */
	public static void guardarAbm(Abm abmAlta )
	{
		try
		{
			//Inicia una TransacciÃ³n
			Transaction transaccion = session.beginTransaction();
			session.save(abmAlta);
			//indica fin y aplicación de la transacción
			transaccion.commit();
		}
		catch(HibernateException ex)
		{
			System.err.println("Ha ocurrido una Excepción al almacenar
			   un objeto Abm "+ ex);
			JOptionPane.showMessageDialog(new JFrame(),
			   "Error al almacenar los datos en la Base de Datos...!",
			   "ERROR INESPERADO...!",
			   JOptionPane.ERROR_MESSAGE);
		}
		finally{
			//closeSession();//Para que pueda seguir agregando nuevos
			//objetos Abm
		}
	}
```

```
//CLASE HibernateUtil.java
      /**Elimina objeto Abm de la Base de Datos: Taller3
       * haciendo uso del método "delete" de Session.
       * @param AbmBaja              */
      public static void eliminarAbm(Abm abmBaja )
      {
            if(sessionFactory == null)
            {
                  createSessionFactory();
            }
            if( session == null)
            {
                  createSession();
            }
            try{
                  //Inicia una Transacción
                  Transaction transaccion = session.beginTransaction();
                  session.delete(abmBaja);
                  //indica fin y aplicación de la transacción
                  transaccion.commit();
            }
            catch(HibernateException ex){
                  System.err.println("Ha ocurrido una Excepción al eliminar
                     un objeto Abm "+ ex);
                  JOptionPane.showMessageDialog(new JFrame(),
                     "Error al eliminar los datos en la Base de Datos...!",
                     "ERROR INESPERADO...!",
                     JOptionPane.ERROR_MESSAGE);
            }
            finally {   closeSession();//?      }
      }

      /**Actualiza objeto Abm de la Base de Datos: Taller3
       * haciendo uso del método "saveOrUpdate" de Session.
       * @param AbmBaja                                    */
      public static void actualizarAbm(Abm abm )
      {
            if(sessionFactory == null)
            {
                  createSessionFactory();
            }
            if( session == null)
            {
                  createSession();
            }
            try{
                  //Inicia una TransacciÃ³n
                  Transaction transaccion = session.beginTransaction();
                  session.saveOrUpdate(abm);
                  //indica fin y aplicación de la transacción
                  transaccion.commit();
            }
            catch(HibernateException ex){
                  System.err.println("Ha ocurrido una Excepción al
                     actualizar un objeto Abm "+ ex);
                  JOptionPane.showMessageDialog(new JFrame(),
                     “Error al actualizar los datos en la Base de Datos..!",
                     "ERROR INESPERADO...!",
                     JOptionPane.ERROR_MESSAGE);
            }
            finally {   closeSession();//?      }
      }
```

```
//CLASE HibernateUtil.java
      public static JResultadoConsulta consulta(String consultaHQL)
      {
            if( session == null )
            {
                  if( sessionFactory == null )
                  {
                        createSessionFactory();
                  }
                  createSession();
            }
            JResultadoConsulta resultadoConsulta;
            List <Abm> listaAbm;
            try
            {     listaAbm =
                     (List<Abm>)(session.createQuery(consultaHQL).list());
                  resultadoConsulta = new JResultadoConsulta(listaAbm);
                  return resultadoConsulta;
            }
            catch(Exception ex)
            {
                  System.err.println("Ha ocurrido una Excepción al ejecutar
                     la consulta HQL " + ex );
                  JoptionPane.showMessageDialog( new JFrame(),
                     "Error al realizar la consulta a la Base de Datos...!",
                     "ERROR INESPERADO..!",
                     JOptionPane.ERROR_MESSAGE
                  );
                  return (JResultadoConsulta)null;
            }
            finally
            {
                  //TODO
            }
      }
}
//FIN DE LA CLASE: HibernateUtil.java
```

Igual que en el proyecto anterior vamos a necesitar una clase que se encargue de almacenar temporalmente los resultados de las consultas que se ejecuten, la llamaremos **JResultadoConsulta.java**, a continuación mostramos su código fuente:

```
//CLASE JResultadoConsulta.java
import java.lang.reflect.Field;
import java.util.List;
import java.util.Vector;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JTable;
/**   @author ehv80     */
public class JResultadoConsulta {

      // Un Vector de Vector de String
      // es como una matriz de String y redimensionable
      private Vector <Vector <String>> datos = null;
      //aquí tuve que cambiar el "project compliance" a la versión 5.0

      private Vector <String> nombresColumnas = null;
```

```java
//CLASE JResultadoConsulta.java
	/**Método: getFila
	 * Sirve para obtener una fila (Vector de String)
	 * del Vector de Vector de String: datos.
	 * Esto representa un registro de la Tabla de la Base de Datos.
	 *
	 * @param int indice
	 * @return Vector <String>                  */
	public Vector <String> getFila(int indice)
	{
		return datos.get(indice);
	}
	/**   Constructor de la clase JResultadoConsulta
	 *    @param ResultSet conjuntoResultado         */
	public JResultadoConsulta(List<Abm> conjuntoResultado) {
	   datos = new Vector <Vector <String>>();
	   Field[] camposDeclarados;
	   try
	   {
	      for(Abm abm : conjuntoResultado )
	      {
			Vector<String> temp = new Vector<String>();
				temp.add( String.valueOf( abm.getNdoc() ) );
				temp.add( abm.getNombre() );
				temp.add( abm.getApellido() );
				temp.add( abm.getTpdoc() );
				temp.add( String.valueOf( abm.getEdad() ));
			//datos.add( abm.getDatos );
			datos.add( temp );
	      }
	      camposDeclarados = Abm.class.getDeclaredFields();
	      nombresColumnas = new Vector <String>();
	      for(Field campo : camposDeclarados)
	      {
			nombresColumnas.add(campo.getName());
	      }
	   }
	   catch(Exception ex){
	      System.err.println("Ha ocurrido una Excepción al inicializar un
	         objeto JResultadoConsulta " + ex);
	      JoptionPane.showMessageDialog(      new JFrame(),
	         "Error al inicializar el conjunto resultado de la consulta!",
	         "ERROR INESPERADO...!",
	         JOptionPane.ERROR_MESSAGE
	      );
	   }
	   finally{
	      //TODO
	   }
	}
	/**Método: getTabla
	 * Sirve para obtener una JTable con el Resueltado de la
	 * Consulta SQL.
	 *  @return JTable             */
	public JTable getTabla()
	{
		return new JTable(datos, nombresColumnas);
	}
	public Abm getAbm(int i)
	{
		return new Abm(datos.get(i));
	}
}
```

```
//FIN DE LA CLASE JResultadoConsulta.java
```

Se debe recordar que como **en el Proyecto_HIBERNATE estamos haciendo uso de las herramientas de HIBERNATE, no necesitaremos aquí una Clase** como **JResultSetToVector.java.**

Otra Clase que necesitamos agregar será: **JSeleccion.java**, que de la misma manera que en el **Proyecto_ABM**, se encargará de mostrar los resultados de las consultas ejecutadas en la Base de Datos al usuario en una **JTable dentro de un JScrollPane.** Su código fuente se muestra a continuación:

```
//CLASE JSeleccion.java
import java.awt.BorderLayout;
import javax.swing.JPanel;
import javax.swing.JInternalFrame;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JButton;
import java.awt.Dimension;
/**   @author ehv80      */
public class JSeleccion extends JInternalFrame {
      private JPanel jContentPaneSeleccion = null;
      private JScrollPane jScrollPaneSeleccion = null;
      private JTable jTableSeleccion = null;
      private JButton jButtonSeleccionSalir = null;
      private JResultadoConsulta resultadoConsulta = null;
      private JModificaciones modificaciones;
      private JBajas bajas;

      public JSeleccion() {
            super();
            initialize();
      }
      /**   @param title       */
      public JSeleccion(String title) {
            super(title);
            initialize();
      }
      /**   @param title
       *    @param resizable         */
      public JSeleccion(String title, boolean resizable) {
            super(title, resizable);
            initialize();
      }
      /**   @param title
       *    @param resizable
       *    @param closable                */
      public JSeleccion(String title, boolean resizable,
                  boolean closable) {
            super(title, resizable, closable);
            initialize();
      }
      /**   @param title
       *    @param resizable
       *    @param closable
       *    @param maximizable             */
      public JSeleccion(String title, boolean resizable,
                  boolean closable, boolean maximizable) {
            super(title, resizable, closable, maximizable);
            initialize();
      }
```

```
//CLASE JSeleccion.java
      /**   @param title
       *    @param resizable
       *    @param closable
       *    @param maximizable
       *    @param iconifiable       */
      public JSeleccion(String title, boolean resizable, boolean closable,
            boolean maximizable, boolean iconifiable)
      {
            super(title, resizable, closable, maximizable, iconifiable);
            initialize();
      }
      /** Constructor adicional:
     * @param JResultadoConsulta resultadoConsulta
     * @param JModficaciones modificaciones          */
    public JSeleccion(JResultadoConsulta resultadoConsulta,
      JModificaciones modificaciones)
    {
      super();
      initialize();
      this.resultadoConsulta = resultadoConsulta;
      this.modificaciones = modificaciones;
      jTableSeleccion = resultadoConsulta.getTabla();
      jTableSeleccion.addMouseListener(new java.awt.event.MouseAdapter()
      {
         public void mouseClicked( java.awt.event.MouseEvent e)
         {
            getModificaciones().cargaDatos(getResultadoConsulta().
               getFila(jTableSeleccion.getSelectedRow()));
            setVisible(false);
         }
      });
      jScrollPaneSeleccion.setViewportView(jTableSeleccion);
    }
    /** Constructor adicional:
     * @param JResultadoconsulta resultadoConsulta
     * @param JBajas bajas                          */
    public JSeleccion(JResultadoConsulta resultadoConsulta, JBajas bajas)
    {
      super();
      initialize();
      this.resultadoConsulta = resultadoConsulta;
      this.bajas = bajas;
      jTableSeleccion = resultadoConsulta.getTabla();
      jTableSeleccion.addMouseListener(new java.awt.event.MouseAdapter(){
         public void mouseClicked(java.awt.event.MouseEvent e)
         {
            getBajas().cargaDatos(getResultadoConsulta().
               getFila(jTableSeleccion.getSelectedRow()));
            setVisible(false);
         }
      });
      jScrollPaneSeleccion.setViewportView(jTableSeleccion);
    }
    /** Método: getModificaciones
     * @param void
     * @return JBajas bajas        */
      public JBajas getBajas() {
            return this.bajas;
      }
```

```
//CLASE JSeleccion.java
      /** Método: getModificaciones
       *      @param void
       *      @return JModificaciones modificaciones         */
    public JModificaciones getModificaciones() {
            return this.modificaciones;
    }
      /** @param List<Abm>
       * @return void          */
      public void setResultadoConsulta(JResultadoConsulta otraListaDeAbms )
      {
            this.resultadoConsulta = otraListaDeAbms;
      }
      /**   @param void
       * @return List<Abm>           */
      public JResultadoConsulta getResultadoConsulta()
      {
            return this.resultadoConsulta;
      }
      /**   This method initializes jScrollPaneSeleccion
       *    @return javax.swing.JScrollPane                  */
      private JScrollPane getJScrollPaneSeleccion() {
            if (jScrollPaneSeleccion == null) {
                  jScrollPaneSeleccion = new JScrollPane();
                  jScrollPaneSeleccion.setViewportView(
                     getJTableSeleccion());
            }
            return jScrollPaneSeleccion;
      }
      /**   This method initializes jTableSeleccion
       *    @return javax.swing.JTable                 */
      private JTable getJTableSeleccion() {
            if (jTableSeleccion == null) {
                  jTableSeleccion = new JTable();
            }
            return jTableSeleccion;
      }
      /**   This method initializes jButtonSeleccionSalir
       *    @return javax.swing.JButton                */
      private JButton getJButtonSeleccionSalir()
      {
         if (jButtonSeleccionSalir == null)
         {
            jButtonSeleccionSalir = new JButton();
            jButtonSeleccionSalir.setText("SALIR");
            jButtonSeleccionSalir.addActionListener(
               new java.awt.event.ActionListener()
               {
                  public void actionPerformed(
                     java.awt.event.ActionEvent e)
                  {
                     if(HibernateUtil.getSession() != null &&
                        HibernateUtil.getSession() != null)
                     {
                        HibernateUtil.desconectar();
                     }
                     dispose();
                  }
               }
            );
         }
         return jButtonSeleccionSalir;
      }
```

```
//CLASE JSeleccion.java
      /**    This method initializes this
       *    @return void                    */
      private void initialize() {
            this.setSize(300, 200);
            this.setPreferredSize(new Dimension(300, 200));
            this.setResizable(true);
            this.setMaximizable(true);
            this.setClosable(true);
            this.setTitle("BUSCA CON HIBERNATE");
            this.setContentPane(getJContentPaneSeleccion());
      }
      /**   This method initializes jContentPaneSeleccion
       *    @return javax.swing.JPanel                  */
      private JPanel getJContentPaneSeleccion()
      {
         if (jContentPaneSeleccion == null)
         {
            jContentPaneSeleccion = new JPanel();
            jContentPaneSeleccion.setLayout(new BorderLayout());
            jContentPaneSeleccion.add(getJScrollPaneSeleccion(),
               BorderLayout.CENTER);
            jContentPaneSeleccion.add(getJButtonSeleccionSalir(),
               BorderLayout.SOUTH);
            }
            return jContentPaneSeleccion;
      }
}
//FIN DE LA CLASE: JSeleccion.java
```