

Rosario A. Alvarez Rodriguez

E.F.P. Ingeniería de Sistemas de la Universidad Nacional de  
San Cristóbal de Huamanga

## Contenido

<b>ADMINISTRACIÓN DEL PROYECTO .....</b>	<b>3</b>
<b>ANÁLISIS.....</b>	<b>3</b>
<b>PROTOTIPOS UI.....</b>	<b>4</b>
<b>DISEÑO .....</b>	<b>7</b>
<b>Base de datos.....</b>	<b>7</b>
<b>Diagramas UML .....</b>	<b>8</b>
<b>IMPLEMENTACIÓN .....</b>	<b>10</b>
<b>Base de datos.....</b>	<b>10</b>
<b>Arquitectura.....</b>	<b>11</b>
<b>Entidades .....</b>	<b>13</b>
<b>Controladoras .....</b>	<b>14</b>
<b>Reporte con IReport .....</b>	<b>15</b>
<b>Creando variable.....</b>	<b>18</b>
<b>Creando grupos.....</b>	<b>19</b>
<b>Conexión con Access.....</b>	<b>21</b>
<b>PATRONES DE DISEÑO .....</b>	<b>22</b>
<b>Patrón Singleton .....</b>	<b>22</b>
<b>Patrón Abstract Factory.....</b>	<b>23</b>
<b>PRUEBAS .....</b>	<b>25</b>
<b>DESPLIEGUE .....</b>	<b>25</b>
<b>Creando ejecutable .....</b>	<b>25</b>
<b>Creando el .exe .....</b>	<b>26</b>
<b>RETO.....</b>	<b>30</b>

## SOFTWARE ESCRITORIO DE GESTIÓN DE ACTIVIDAD ECONÓMICA

Sólo describí todo lo que “pensé” para poder realizar este software.

### ADMINISTRACIÓN DEL PROYECTO

Esta fase es importante porque nos permite conocer “qué hacer”, “cuánto cobrar”, “en qué tiempo” (Project charter), “quiénes harán cada actividad” (cronograma), “qué entregar al cliente y cuándo” (entregables), entre otras cositas que realizan la alta gerencia.

Qué haré: un software escritorio que me permita administrar información referente a la actividad económica “GRAN POLLADA PRO SALUD – ROSARIO” que se realizó el 30 de noviembre 2014.

### ANÁLISIS

Aquí analizaremos todo lo necesario para entender el problema.

Se realiza entrevistando al cliente; revisando otro software parecido, leyendo los documentos de la empresa, etc.

Como el sistema era para mí, sólo me propuse lo siguiente:

- “Quiero saber” a quiénes se les repartió y cuántas tarjetas.
- “Quiero anotar” a los asistentes con o sin tarjeta y cuánto pagan. Quizás su nombre más y, algún campo que me permita registrar alguna observación.
- “Quiero un informe” de las tarjetas asistidas y el monto pagado.
- “Quiero un informe” del monto que deben los que asistieron con o sin tarjeta.
- “Quiero un informe” que me muestre cuánto se vendió libremente.
- Ah, también me gustaría (como en ese momento no podía restar bien) que me ayude a calcular automáticamente el vuelto.

En un inicio todas las ideas no están claras. Es necesario manejar el mismo vocabulario y revisar el catálogo de requerimientos una y otra vez (iterativo), porque no siempre los clientes saben exactamente qué es lo que quieren. Y con los requisitos que nos menciona nosotros podemos aumentar más y de esta manera superamos las expectativas de nuestros clientes (así también evitamos realizar código basura o continuos mantenimientos).

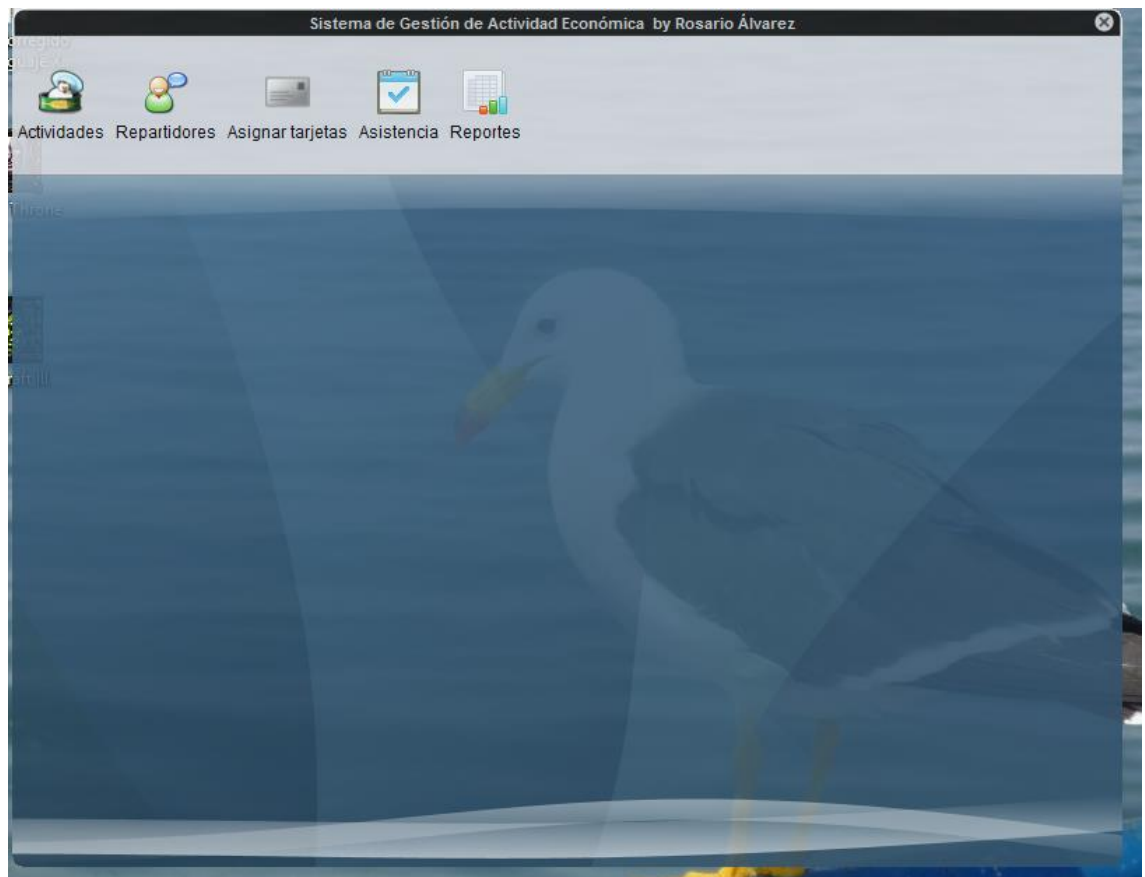
A veces mi cliente lee el documento y dice “ya está bien” y al final (en el despliegue) me salen con “esto no quería” “¿eso para qué?” [...]

Así que mejor entrego un documento que contenga los “pantallazos” o un video simulando la secuencia de ellas. Así evito perder mi tiempo en código que al final no querrá pagar (y no es su culpa)

En lo personal no me gusta invertir mi tiempo en mucho “maquillaje” pero sí considero criterios de funcionalidad del sistema como:

- **Amigable:** facilidad de interacción del sistema con el usuario sin tener que consultar el manual.
- **Legibilidad:** color de textos, contraste de los textos con el fondo y el tamaño de la fuente, que debe ser adecuado para su legibilidad por la mayoría de los usuarios.

- **Eficaz:** una tarea se realiza sin complicación.
- **Eficiencia:** las tareas se realizan rápida y fácilmente.
- **Interfaz gráfica:** que tan placentero resulta la navegación en el sistema gracias a la interfaz gráfica. Incluye imágenes, colores y posición de los elementos que conforma el sistema.
- [...]



Así medio transparente me pareció súper cool; quizás porque era la primera vez que veía interfaces “bonitas” en java...

## PROTOTIPOS UI

**Actividades**

Actividad	Organizado por	Lugar	Fecha	Valor	Total tarjetas

Repartidores

Nombre

+

-

**Repartidor:** persona que distribuye tarjetas.

Asignar Tarjetas

Actividad

Repartidor

Tarjetas del  al

O agregar Tarjeta

Asistencia de colaboradores

Núm tarjeta

☐ Libre

Total Tarjts.

Total a pagar

Monto pagado

Vuelto

Colaborador

Monto por Tarjeta

Nota

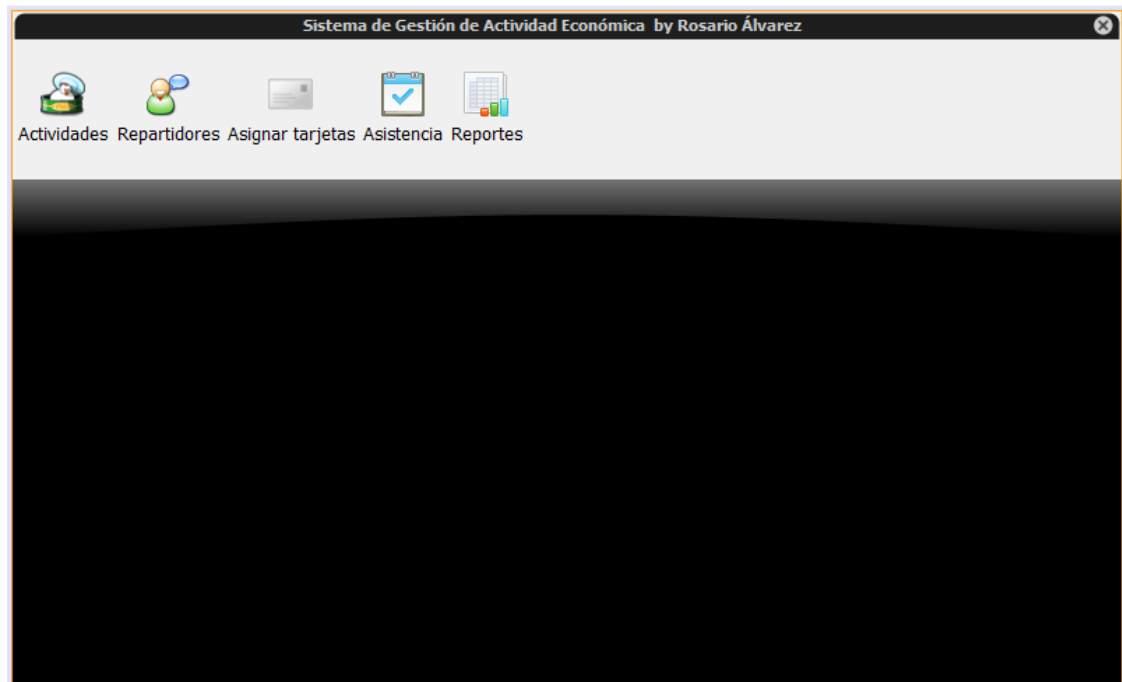
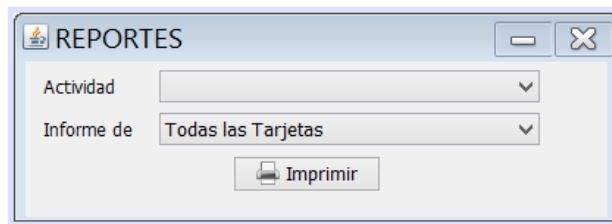
Repartidor

[REPARTIDOR]

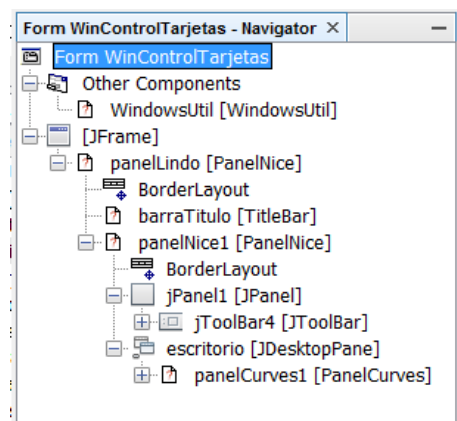
☒ Limpiar datos

En esta interfaz quise que cuando se tipee el número de tarjeta, automáticamente me diga si ya fue asistida o no y el nombre del repartidor. Así también el monto por tarjeta es de acuerdo a la actividad seleccionada y se puede editar (Rafitas me preguntó si podía pagar por partes ¡claro, el sistema lo tiene que hacer!). También quise la opción de limpiar o no datos; puede que haya alguien que compre 10 tarjetas, entonces necesito mantener el nombre del

colaborador para registrarlo. Ah también la opción de “Libre” que indica si la tarjeta fue adquirida libremente.



Esta interfaz contiene:

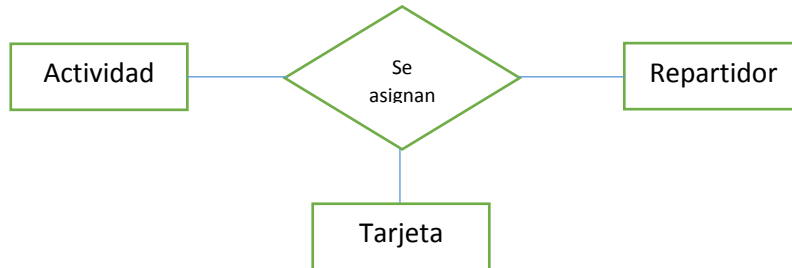


Además de seleccionar la propiedad **undecorated** del JFrame.

# DISEÑO

## Base de datos

El esquema ER me apoyo de la metodología mixta que combina estrategias de diseño ascendente y descendente - “divides y vencerás”.

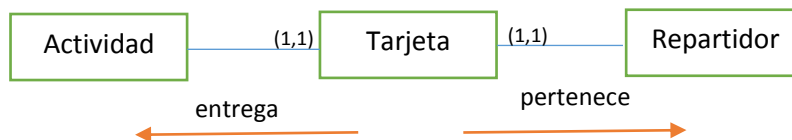


En una actividad económica se asignan (o distribuyen) 1 o muchas tarjetas a uno o muchos repartidores.

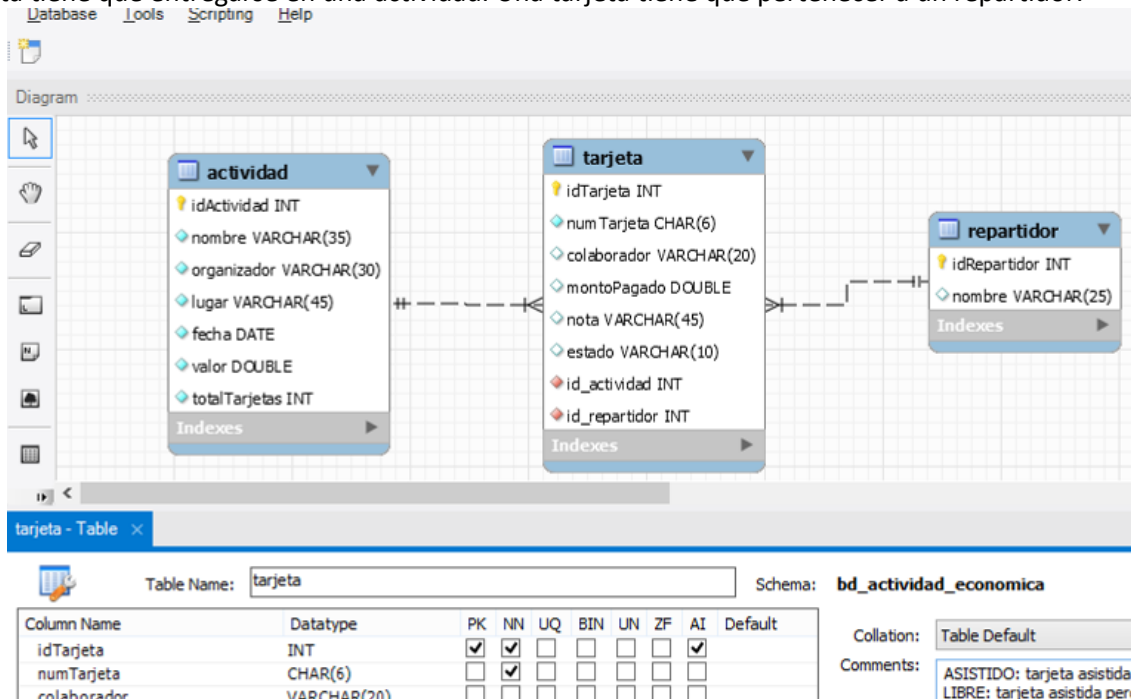
El diagrama ER depende mucho de los requerimientos del software.

Este diagrama ER mostrado funciona muy bien para un usuario que siempre realiza actividades económicas y en cada uno de ellas participan diferentes repartidores.

Pero en mi caso sólo lo hice una vez, luego de haber realizado la actividad fácilmente emito los reportes y los guardo y luego limpio la base de datos. Por eso al final me quedé con este modelo ER:



Una tarjeta tiene que entregarse en una actividad. Una tarjeta tiene que pertenecer a un repartidor.



No olvidar que cuando se elimina una actividad o un repartidor, debe eliminarse en cascada todos los registros en la tabla tarjeta.

## Diagramas UML

Que si es necesario los diagramas UML, ¡claro! Salvo que seas un tromezas@ y lo tengas todo en tu mente (y el software a implementar sea pequeño). Pero por muy pequeño que sea el software a implementar siempre uso papel y lápiz, es mejor invertir tiempo en la fase de análisis y diseño que decir “uy esta base de datos está mal modelada, borrar código oh!” – perder tiempo = perder dinero.

Desde luego hice sólo algunos diagramas “lo necesario” pues hay diagramas que en verdad “me estorban” (es que yo misma lo implementaré y sé cómo hacerlo) o utilizo otros modelos (como el de ER para base de datos, ya no hago diagrama de clases) y sólo tenía una noche para implementarlo (no lo acabé, me faltó reportes); el sistema era para mí :P

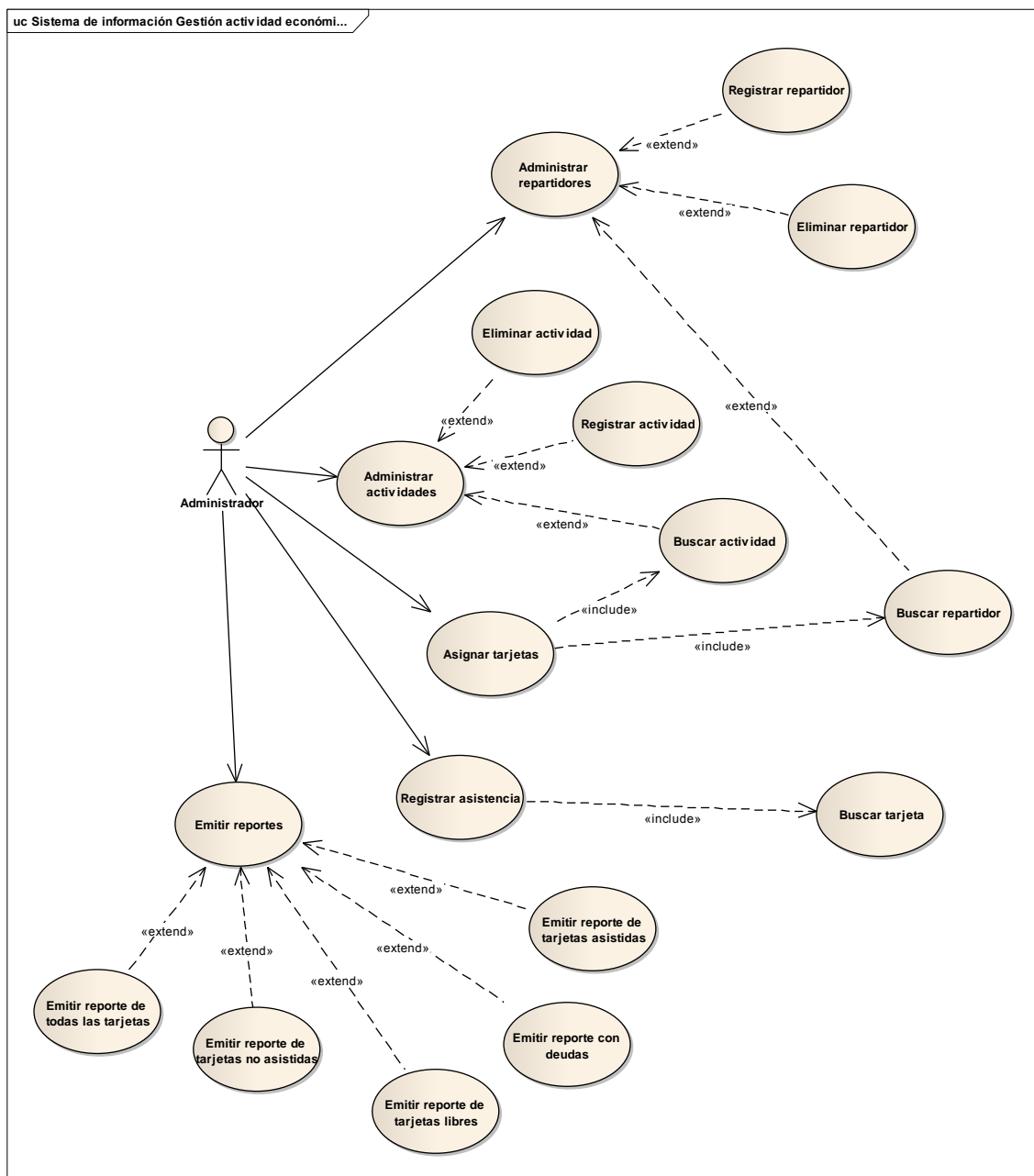


Diagrama de casos de uso



En cuanto al diagrama de secuencia hice sólo para los casos de uso críticos como el de Registrar asistencia. Se dice que por cada diagrama de secuencia sólo va un caso de uso; a veces quisiera que estén los casos de usos relacionados (o sea, por requerimiento); cuando es simple lo hago:

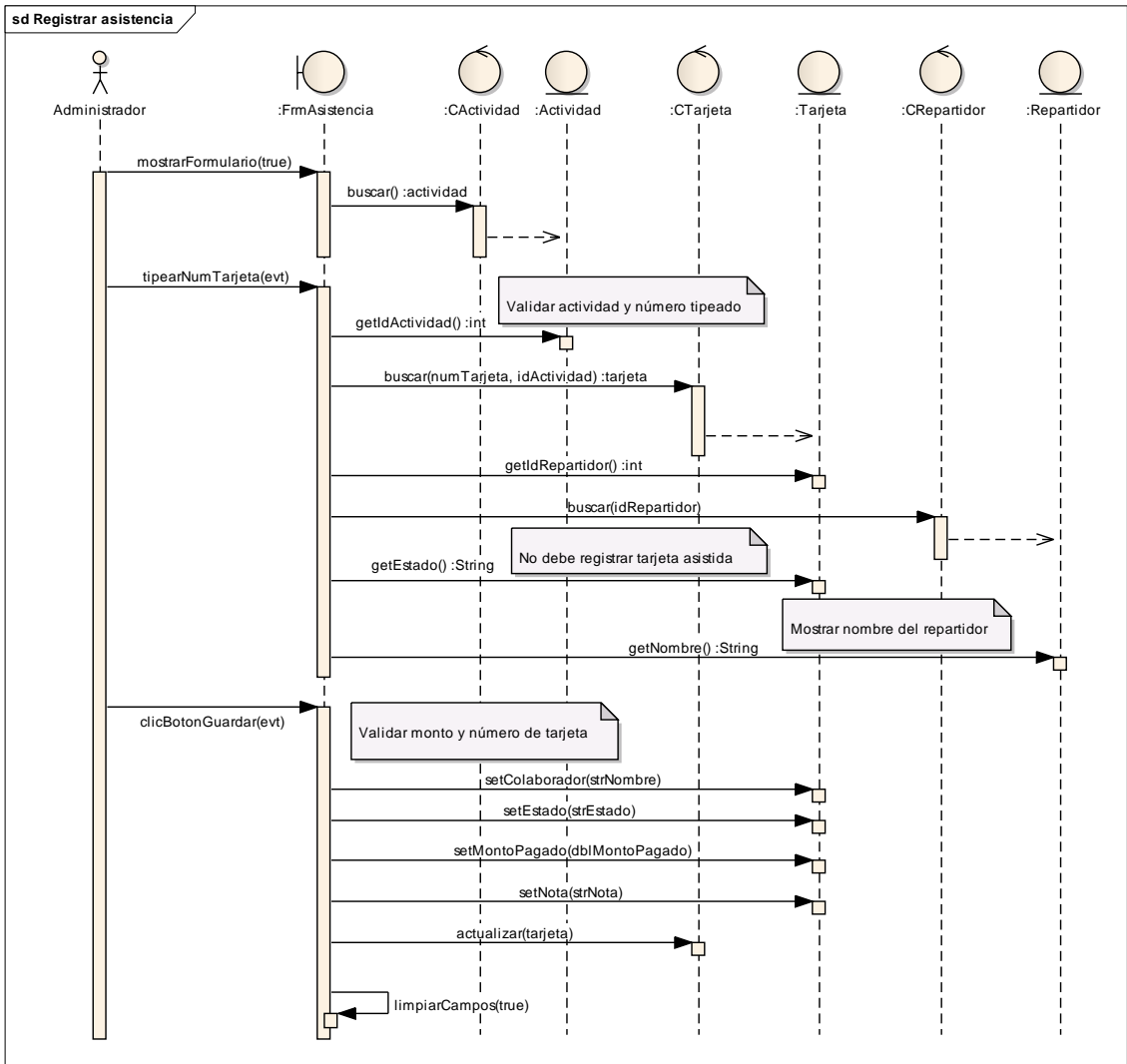
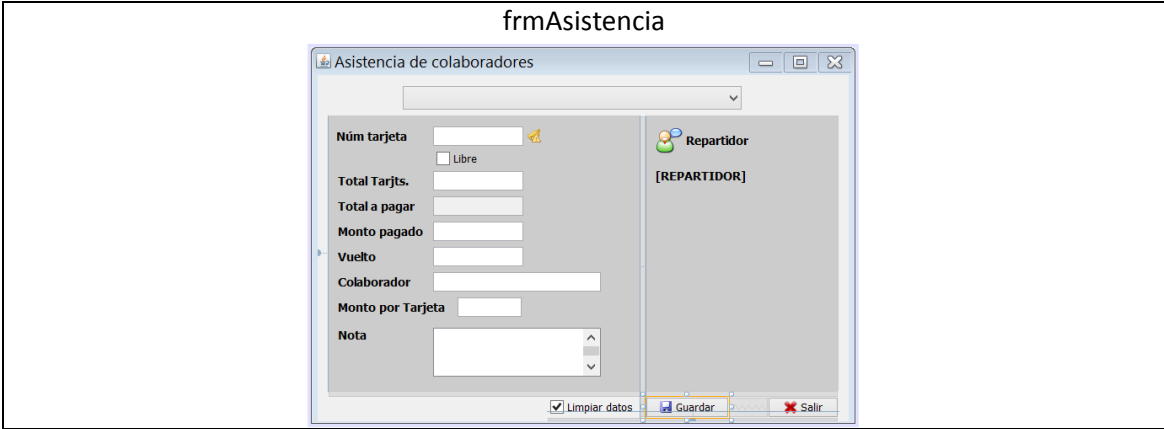


Diagrama de secuencia para el caso de uso: **Registrar asistencia**. También está el caso de uso buscar actividad, buscar tarjeta y buscar repartidor.

Así es mucho más entendible.

En código se refleja algo así (sólo para el caso de uso registrar asistencia):



Clic en el botón guardar

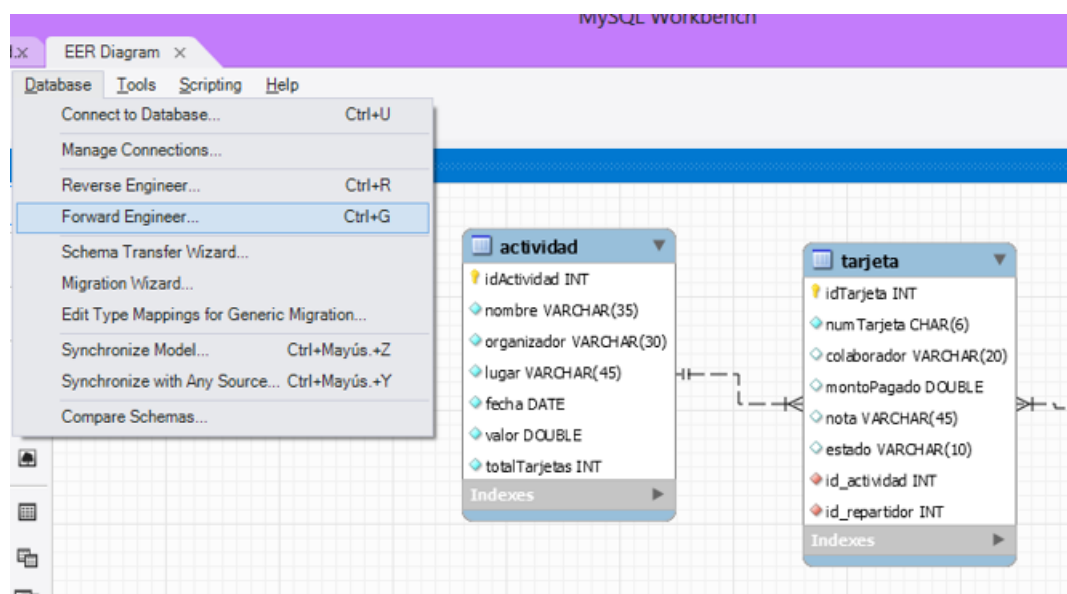
```
private void btnGuardarActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    if (tarjeta == null || txtNumTarjeta.getText().isEmpty())  
    {  
        JOptionPane.showMessageDialog(this, "Indique tarjeta", "Atención", JOptionPane.WARNING_MESSAGE);  
        return;  
    }  
    double montoPagadoPorTarjeta;  
    try {  
        montoPagadoPorTarjeta = Double.parseDouble(txtMontoPagadoPorTarjeta.getText());  
    } catch (NumberFormatException ex) {  
        JOptionPane.showMessageDialog(this, "Monto pagado por tarjeta inválido", "Atención", JOptionPane.WARNING_MESSAGE);  
        return;  
    }  
  
    tarjeta.setColaborador(txtColaborador.getText());  
    tarjeta.setEstado(chkLibre.isSelected() ? "LIBRE" : "ASISTIDO");  
    tarjeta.setMontoPagado(montoPagadoPorTarjeta);  
    tarjeta.setNota(txtNota.getText());  
  
    Resultado resultado = cTarjeta.actualizar(tarjeta);  
    if (resultado.isCorrecto()) {  
        JOptionPane.showMessageDialog(this, "Tarjeta asistida exitosamente.", "Éxito", JOptionPane.INFORMATION_MESSAGE);  
        if (chkLimpiar.isSelected()) {  
            limpiarCampos();  
        }  
    } else {  
        Formateador.mostrarMensajeError(resultado, this);  
    }  
}
```

Si se documenta bien, el programador programa feliz :D

## IMPLEMENTACIÓN

### Base de datos

El MySQLWorkbench me permite generar el script sql a partir de un diagrama ER y crear su base de datos “forward Engineer”



Selecciona según sus necesidades

y... listo

This script will now be executed on the DB server to create your databases.  
You may make changes before executing.

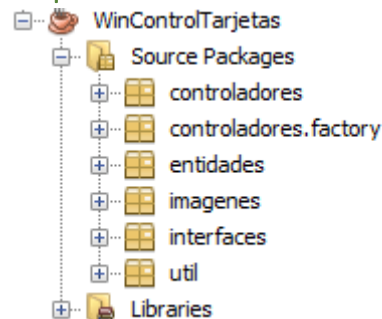
```
1  -- MySQL Workbench Forward Engineering
2
3  SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
4  SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
5  SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_TERNARY_VALUES';
6
7  -- Schema bd_actividad_economica
8
9  DROP SCHEMA IF EXISTS `bd_actividad_economica` ;
10
11  -- Schema bd_actividad_economica
12
13  CREATE SCHEMA IF NOT EXISTS `bd_actividad_economica` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
14
15  -- Table `bd_actividad_economica`.`actividad`
16
17  DROP TABLE IF EXISTS `bd_actividad_economica`.`actividad` ;
18
19  CREATE TABLE IF NOT EXISTS `bd_actividad_economica`.`actividad` (
20    `idActividad` INT NOT NULL AUTO_INCREMENT,
21    `nombre` VARCHAR(35) NOT NULL,
22    `organizador` VARCHAR(30) NOT NULL,
23    `lugar` VARCHAR(45) NOT NULL,
24    `fecha` DATE NOT NULL,
25    `valor` DOUBLE NOT NULL,
26    `totalTarjetas` INT NOT NULL,
27    PRIMARY KEY (`idActividad`))
28    ENGINE = InnoDB;
```

Save to File... Copy to Clipboard

Back Next Cancel

Se puede guardar el script y/o ejecutarlo.

## Arquitectura



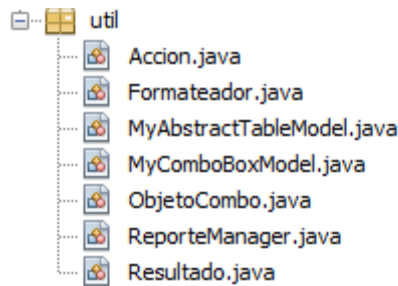
**Controladoras:** clases que acceden a la base de datos

**Entidades:** clases

**Interfaces:** clases de interfaz gráfica

Una vez utilicé JPA (e ingeniería inversa) y me dio la misma arquitectura (controllers, entities), me acostumbré a ella, me gusta.

Explicaré sólo algunos códigos:



Librería que me ofrece esos **controles bonitos**: edisoncorSX.jar

Código que me permite ver la **pantalla transparente y poder arrastrarla**:

```
public class WinControlTarjetas extends javax.swing.JFrame {

    private FrmActividades frmActividades;
    private FrmAsignarTarjetas frmAsignarTarjetas;
    private FrmAsistencia frmAsistencia;
    private FrmRepartidores frmRepartidores;
    private FrmReportes frmReportes;

    public WinControlTarjetas() {

        initComponents();
        setLocationRelativeTo(null);
        WindowsUtil.makeWindowsOpacity(this, .9f);
        WindowsUtil.makeWindowsShape(this, panellindo.getShape());
        new WindowDragger(this, panellindo);
        barraTitulo.addAction(new ActionListener() {
            public void actionPerformed(ActionEvent ae) {
                System.exit(0);
            }
        });
    }
}
```

La clase **Formateador** proporciona métodos estáticos, útiles

```
public class Formateador {

    /**
     * Establece la data y las cabeceras de la tabla. También les asigna e
     * ancho de las columnas.
     *
     * @param tabla la tabla a formatear
     * @param cabeceras nombres de las columnas
     * @param anchos los anchos de las columnas.
     */
    public static void redimensionar(JTable tabla, String cabeceras, Strin

    /**
     * devuelve el item seleccionado de la tabla
     * @param tabla
     * @return
     */
    public static int getItemSeleccionado(JTable tabla) {...10 lines }

    /**
     * Establece el máximo tamaño del JTextField y lo que se tipea lo con
     * a mayúscula.
     *
     * @param caja es el JTextField a formatear
     * @param tamaño es el maximo tamaño del texto que se puede tipear
     * @param evt el evento que produce al tipear
     */
    public static void setMaxTamaño(JTextField caja, int tamaño, KeyEven
```

Siempre acostumbro a documentar los métodos porque siempre va haber alguien que lo va a mantener y necesita entender el código. (un / luego \*\* y un enter)

**MyAbstractTableModel**, **MyComboBoxModel**, **ObjetoCombo** son clases que me permitirán acceder ciertos valores de una tabla y un combo (como mostrar los datos en un JTable desde la base de datos; acceder al id de un ítem seleccionado de un JComboBox, etc).

La clase **Resultado**, es una clase que creé para devolver ciertos valores desde la base de datos. Cuando hago una consulta a la base de datos, por ejemplo, no quisiera que sólo me diga si hubo error o no en mi consulta sql, sino también quería el resultado (si es que no hubo error claro está) en una matriz de String o de Object (para mostrarlos en un JTable) o

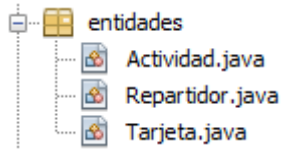
un Vector del ObjetoCombo (para mostrarlos en un JComboBox)

```
public class Resultado {  
  
    private String mensajeError;  
    private boolean correcto;  
    private Vector cache;  
    private Object[][] data;  
    private Vector<ObjetoCombo> itemsCombo ;  
  
    public Resultado() {...4 lines }  
  
    public String getMensajeError() {...3 lines }  
  
    public void setMensajeError(String mensajeError) {...3 lines }  
  
    public boolean isCorrecto() {...3 lines }  
  
    public void setCorrecto(boolean correcto) {...3 lines }  
  
    public Vector getCache() {...3 lines }  
  
    public Vector<ObjetoCombo> getItemsCombo() {...3 lines }  
  
    public Object[][] getData() {...3 lines }  
  
    public void setData(Object[][] data) {...3 lines }  
  
    public void setCache(Vector cache) {...3 lines }  
}
```

La clase **ReporteManager** me ayudará a mostrar los reportes de IReport

## Entidades

Las entidades son las Clases y contienen el mismo nombre de las tablas creadas en el base de datos; asimismo, sus atributos. Estas entidades nos permitirán almacenar datos recogidos desde las interfaces gráficas (vistas) y acceder a ellas en las clases controladoras para registrar, eliminar, modificar o leer registros de la base de datos.

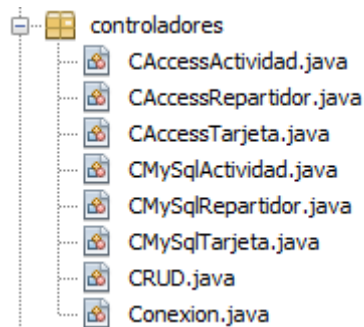


```
public class Repartidor {  
  
    private int idRepartidor;  
    private String nombre;  
  
    public int getIdRepartidor() {  
        return idRepartidor;  
    }  
  
    public void setIdRepartidor(int idRepartidor) {  
        this.idRepartidor = idRepartidor;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
}
```

Clase Repartidor. Note que no tiene constructor

## Controladoras

Las clases controladoras nos permitirán conectarnos a la base de datos a través del API JDBC.



Como las clases controladoras tienen siempre la clase Connection, un Statement, método para cerrar la conexión, y el mismo código para realizar una transacción; creé la clase CRUD (create, read, update, delete) para que después las demás clases puedan heredar de ella.

```
public class CRUD {  
  
    protected Connection cn;  
    protected Statement st;  
  
    public CRUD() {...3 lines }  
  
    protected void closeConnection() {...11 lines }  
  
    protected Resultado insert(String sql) {...3 lines }  
  
    protected Resultado update(String sql) {...3 lines }  
  
    protected Resultado delete (String sql) {...3 lines }  
  
    private Resultado transact(String sql) {...15 lines }  
}
```

Así tenemos por ejemplo a la clase controladora **CMySQLTarjeta**

```
* @author Rosary  
*/  
  
public class CMySQLTarjeta extends CRUD implements ITarjeta{  
  
    @Override  
    public Resultado actualizar(Tarjeta tarjeta) {  
        String sql = "update tarjeta set "  
            + "colaborador = '" + tarjeta.getColaborador() + "' , "  
            + "montoPagado = " + tarjeta.getMontoPagado() + " , "  
            + "nota = '" + tarjeta.getNota() + "' , "  
            + "estado = '" + tarjeta.getEstado() + "' , "  
            + "id_actividad= " + tarjeta.getId_actividad() + " , "  
            + "id_repartidor= " + tarjeta.getId_repartidor() + " "  
            + " where idTarjeta = " + tarjeta.getIdTarjeta();  
        return update(sql);  
    }  
  
    @Override  
    public Resultado registrar(Tarjeta tarjeta) {  
        String sql = "insert into tarjeta (estado,montoPagado,nota, "  
            + "numTarjeta,id_actividad,id_repartidor) "  
            + "values ('" + tarjeta.getEstado() + "' , "  
            + tarjeta.getMontoPagado() + " , "  
            + tarjeta.getNumTarjeta() + " , "  
            + tarjeta.getId_actividad() + " , "  
            + tarjeta.getId_repartidor() + ")";  
        return insert(sql);  
    }  
}
```

El método **update** e **insert** lo heredan de CRUD:

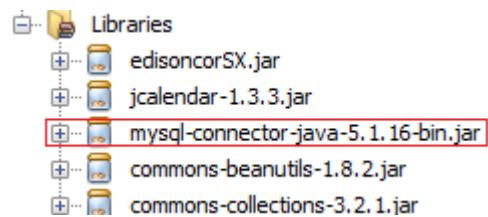
```
protected Resultado insert(String sql){
    return transact(sql);
}

protected Resultado update(String sql){
    return transact(sql);
}

protected Resultado delete (String sql){
    return transact (sql);
}

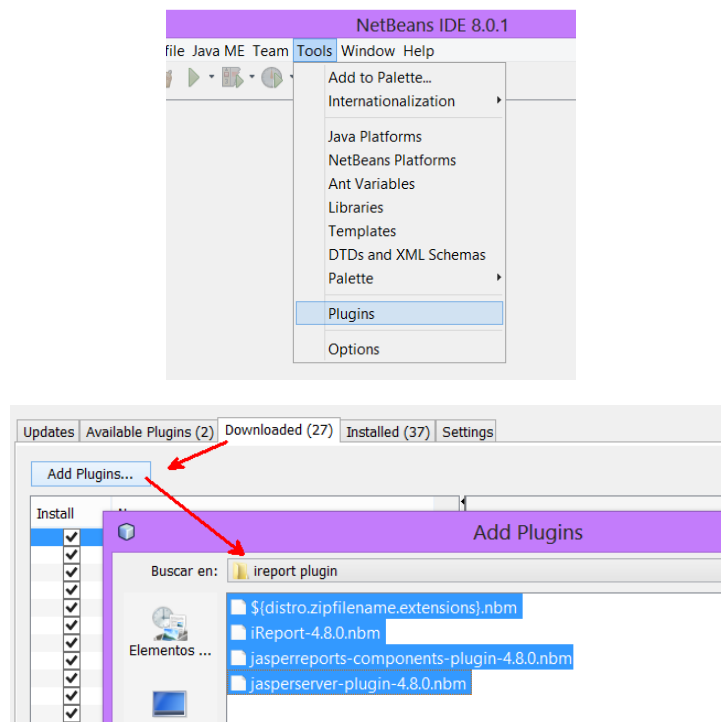
private Resultado transact(String sql){
    Resultado resultado = new Resultado();
    cn = Conexion.getConnection();
    try{
        st = cn.createStatement();
        st.executeUpdate(sql);
        resultado.setCorrecto(true);
    }catch(SQLException ex){
        resultado.setCorrecto(false);
        resultado.setMensajeError(ex.getMessage());
    }finally{
        closeConnection();
    }
    return resultado;
}
```

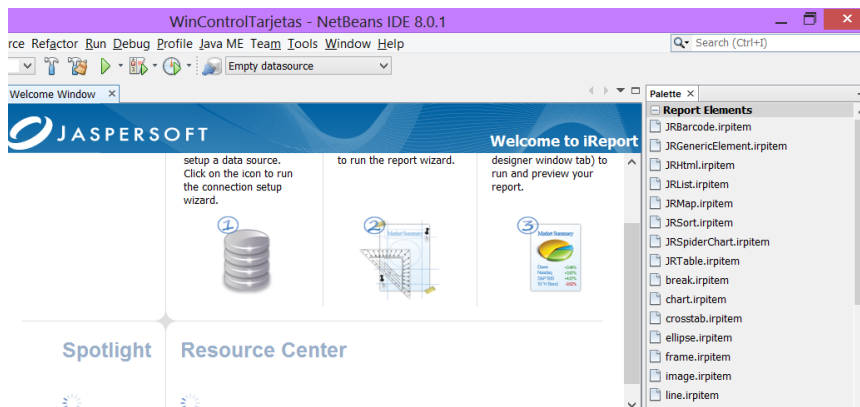
No olvidar cargar el driver de MySql



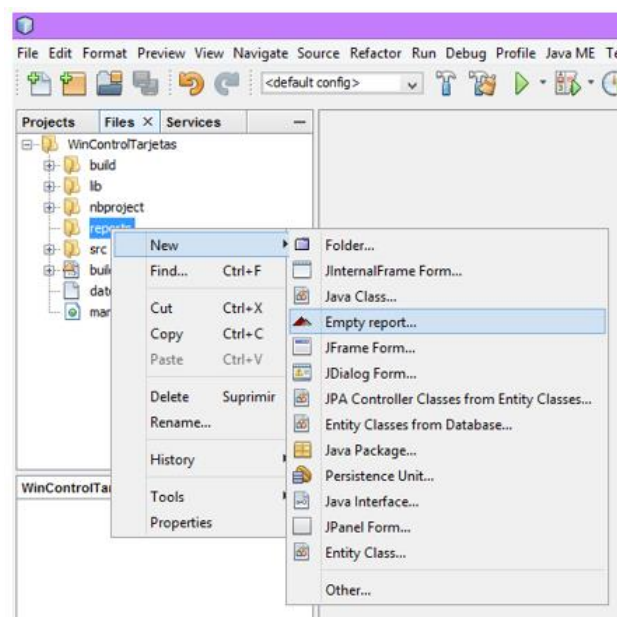
## Reporte con IReport

Veamos cómo se hizo los reportes. En este caso utilicé los pluggins de IReport

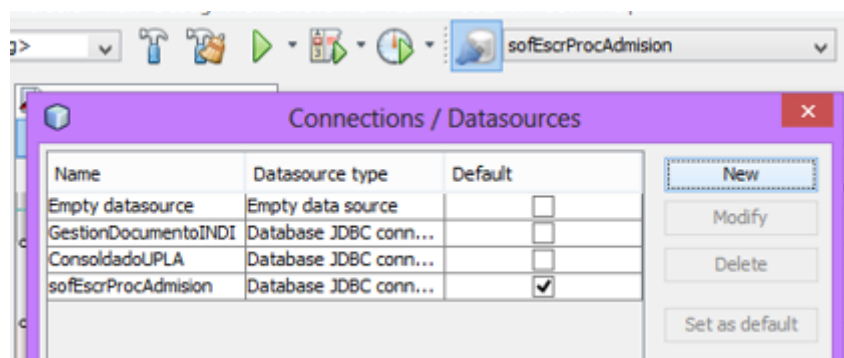




Bien, ahora procedemos a crear nuestro primer reporte (recomiendo reiniciar el NetBeans)



Realizamos nuestra consulta sql (en el IDE que desee) o sino directamente en el Report Query pero para eso debe crearse una conexión:



Configurar parámetros necesarios:

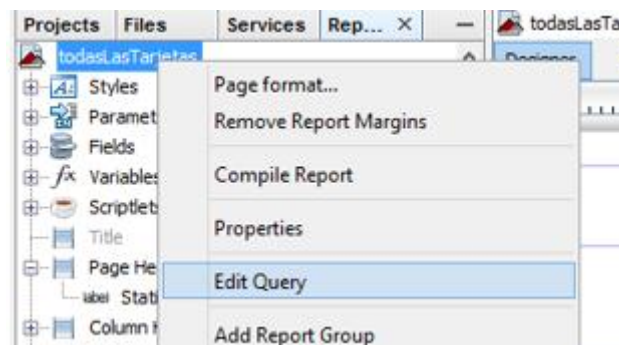


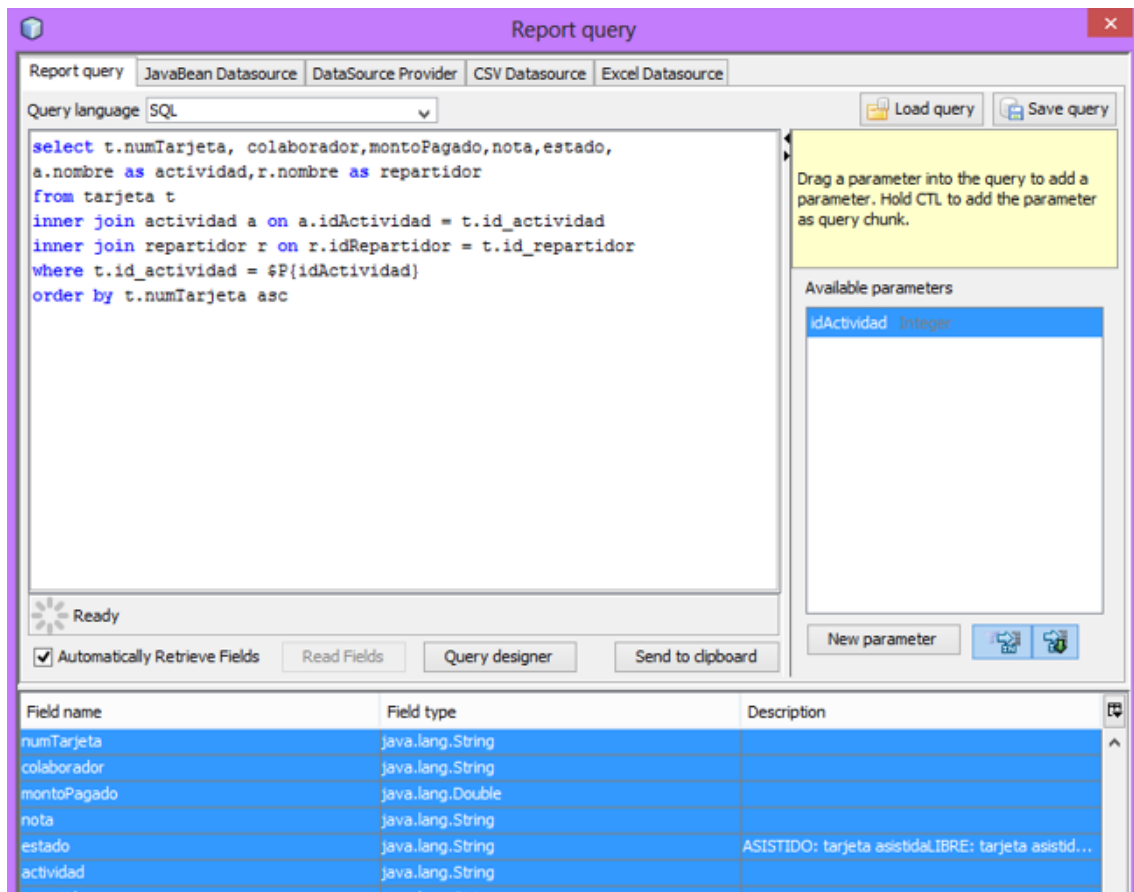


Listo. Copio la consulta

tarjetaasignada	tarjetaasignada	tarjeta	tarjeta	repartidor	tarjetaasignada	tarjeta
<pre> 1 select t.numTarjeta, colaborador, montoPagado, nota, estado, 2 a.nombre as actividad, r.nombre as repartidor 3 from tarjeta t 4 inner join actividad a on a.idActividad = t.id_actividad 5 inner join repartidor r on r.idRepartidor = t.id_repartidor 6 where t.id_actividad = 1 7 order by t.numTarjeta asc; 8 </pre>						
numTarjeta	colaborador	montoPagado	nota	estado	actividad	repartidor
000259	CHARITO	10		ASISTIDO	GRAN POLLADA PRO SALUD - ROSARIO	SABINA RODRIGUEZ
000260	CHARITO	10		ASISTIDO	GRAN POLLADA PRO SALUD - ROSARIO	SABINA RODRIGUEZ
000261	DIEGOMÓN	10		LIBRE	GRAN POLLADA PRO SALUD - ROSARIO	SABINA RODRIGUEZ
000262	DIEGOMÓN	10		LIBRE	GRAN POLLADA PRO SALUD - ROSARIO	SABINA RODRIGUEZ
000263		0		NO ASISTE	GRAN POLLADA PRO SALUD - ROSARIO	SABINA RODRIGUEZ
000264	VECINA GLADYS	0		ASISTIDO	GRAN POLLADA PRO SALUD - ROSARIO	SABINA RODRIGUEZ
000265	VECINA LUZ	0		ASISTIDO	GRAN POLLADA PRO SALUD - ROSARIO	SABINA RODRIGUEZ
000266		10		ASISTIDO	GRAN POLLADA PRO SALUD - ROSARIO	SABINA RODRIGUEZ

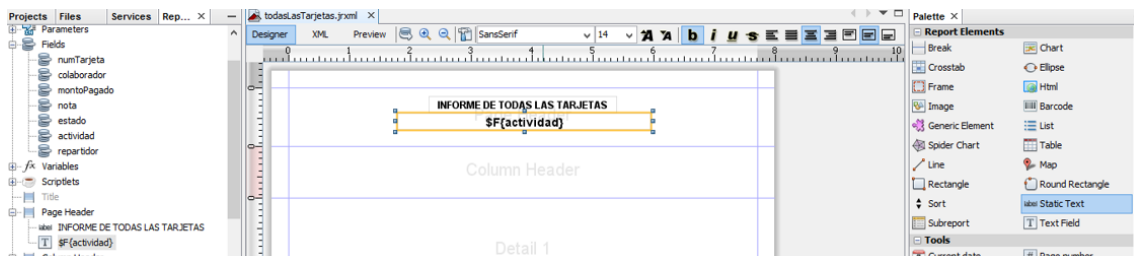
en el Report Query:





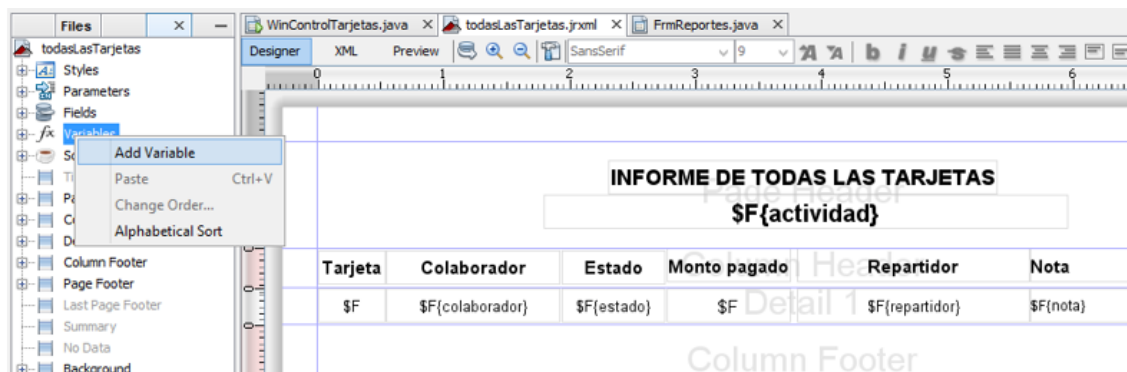
Y le agrego un parámetro (idActividad tipo Integer)

Y ahora solo queda diseñar el reporte.

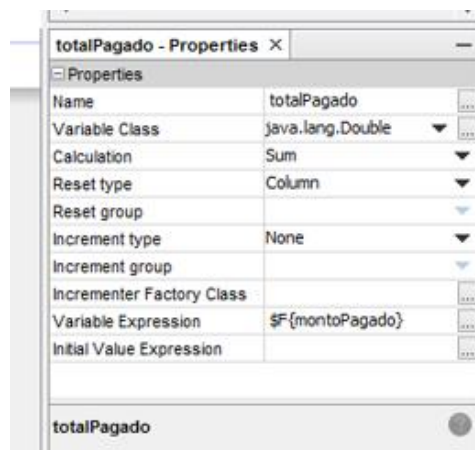


Creando variable

Necesito una variable que me calcule el monto total pagado.



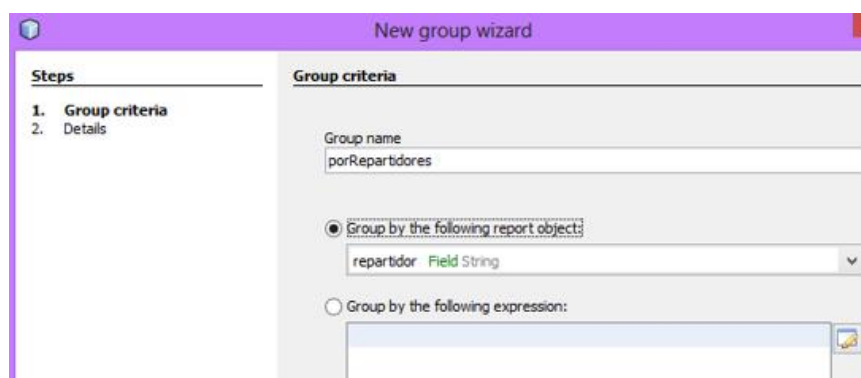
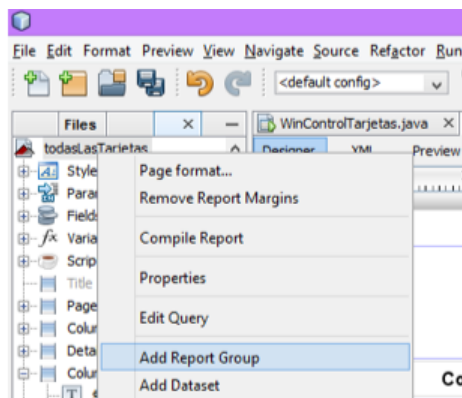
Establecemos los siguientes valores en sus propiedades



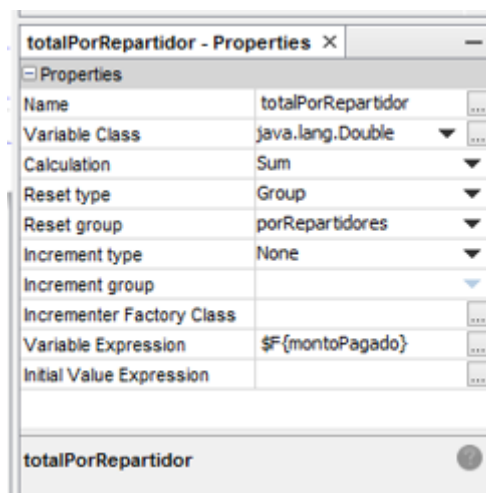
INFORME DE TODAS LAS TARJETAS					
\$F{actividad}					
Tarjeta	Colaborador	Estado	Monto pagado	Repartidor	Nota
\$F	\$F{colaborador}	\$F{estado}	\$F	\$F{repartidor}	\$F{nota}
Total \$/. \$V{totalPagado}					

### Creando grupos

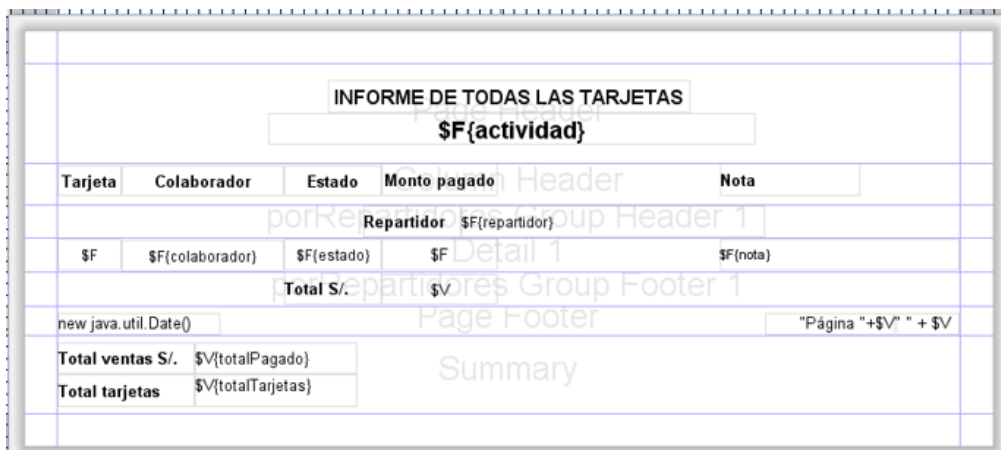
Quisiera conocer el monto total pagado por repartidor.



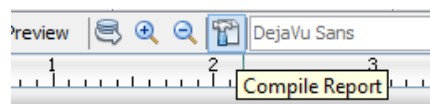
Configuramos algunas de sus propiedades



Y rediseñamos el reporte

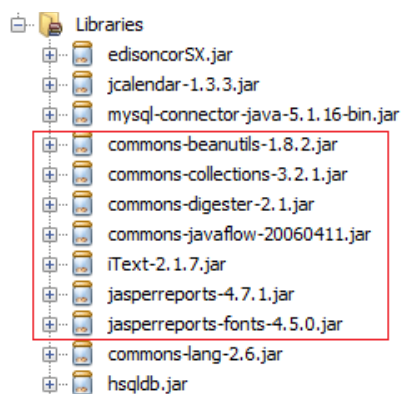


Ahora a compilarlo



INFORME DE

No olvidar cargar las librerías necesarias.



Sistema Actividad Económica by Rosario Álvarez

Actividades Repartidores Asignar tarjetas Asistencia Reportes

Vista previa

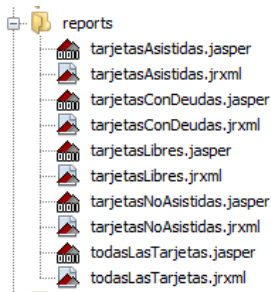
000245		NO ASISTE	0.0	
000246		NO ASISTE	0.0	
000247		NO ASISTE	0.0	
000248		NO ASISTE	0.0	
000249		NO ASISTE	0.0	
000250		NO ASISTE	0.0	
Total \$/.			0.0	
Repartidor SABINA RODRIGUEZ				
000251	CHARITO	ASISTIDO	10.0	
000252	CHARITO	ASISTIDO	10.0	
000253	CHARITO	ASISTIDO	10.0	
000254	CHARITO	ASISTIDO	10.0	
000255	CHARITO	ASISTIDO	10.0	

14/02/2015 4:46 AM

Página 4 de 8

Página 4 de 8

Para el resto de los reportes, copié el archivo tarjetasAsistidas.jrxml y le modifiqué la consulta, el diseño y el nombre del archivo.

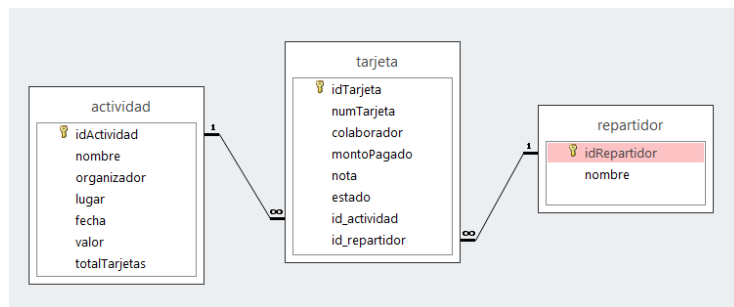


## Conexión con Access

Pues bien, habrán notado en el paquete controladoras tengo 6 clases controladoras; esto es, 3 para que se conecten a MySQL y las otras, a Access.

Es que se me ocurrió compartir el software con alguien que no sabe nada de programación, ni de base de datos; ¿cómo podría utilizarlo? Elegí MySQL porque es mi favorito, pero ahora no puedo decirle “anda al mercado en camión a comprar un caramelo”, hay que saber seleccionar un SGBD. Así que necesito que se conecte a Access (pues considero que la gran mayoría tiene SO Microsfot, y en ella, el paquete Office) pero no quiero borrar el código realizado para conectarme a MySQL (para realizar backup o por seguridad).

Así que cree una base de datos en **Access 2013** con los mismos nombres de tablas y sus atributos y, relaciones.

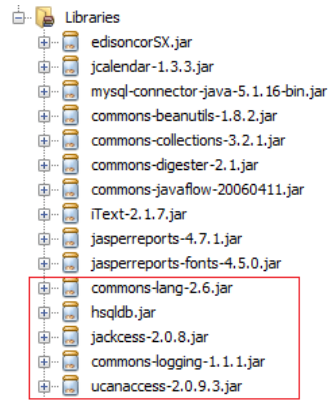


Para conectarnos a la base de datos Access (accdb o mdb) en Java 8 ya no es necesario utilizar el puente JDBC-ODBC pues fue removido.

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver"); // for MS Access ... MS access
driver loading
```

```
String conURL = "jdbc:odbc:sampleDNS";
Connection con = DriverManager.getConnection(conURL);
Statement statement = con.createStatement();
```

Sino, se utilizará 5 jar extras:



Y la url de conexión

```
Connection conn=DriverManager.getConnection("jdbc:ucanaccess://<Path to your
database i.e. MS Access DB>");
Statement s = conn.createStatement();
```

OK, ahora necesito que mis interfaces gráficas funcionen sin importar a qué base de datos se conecten, que haya una bandera que diga si es Access o MySql ¿cómo lo soluciono?

Afortunadamente existen los patrones de diseño uf. Es mágico.

## PATRONES DE DISEÑO

Son esquemas que se usan para solucionar un problema. Los esquemas han sido probados extensivamente, y han funcionado. Se tiene experiencia sobre su uso. En verdad funciona.

### Patrón Singleton

El patrón Singleton limita el número de instancias de un objeto a uno. Los clientes que quieren usar dicho objeto compartirán la única instancia existente.

Puesto que la clase Connection consume recursos, es necesario utilizar el patrón Singleton.

```
public class Conexion {
    //se hace privado el constructor para evitar que el cliente lo instancie
    private Conexion() {
    }

    /**
     * Obtiene la instancia a través de un método estático que devuelve siempre
     * la misma instancia.
     * @return objeto Connection
     */
    public static Connection getConnection() {
        Connection connection = null;
        try {
            connection = DriverManager.getConnection(DatosIni.getUrl(),
                DatosIni.getUsuario(), DatosIni.getPassword());
        } catch (SQLException e) {
            System.out.println("Error:" + e.getMessage());
        }
        return connection;
    }
}
```

DatosIni es una clase que accede permanente a los datos proporcionados en el archivo datos.ini

## Patrón Abstract Factory

Proporciona una interfaz para crear familias de objetos relacionados o que dependen entre sí, sin especificar sus clases concretas.

O sea, desde la interfaz gráfica FrmAsistencia podemos crear así:

```
public class FrmAsistencia extends javax.swing.JInternalFrame {

    private final IActividad cActividad;
    private final IRepartidor cRepartidor;
    private final ITarjeta cTarjeta;
    private final MyComboBoxModel cbmActividad = new MyComboBoxModel();
    private double valor;
    private Repartidor repartidor;
    private Tarjeta tarjeta;

    public FrmAsistencia() {
        initComponents();
        int bd = DatosIni.getBd().equals("MySQL") ? CFactory.MYSQL : CFactory.ACCESS;
        cActividad = CFactory.getCFactory(bd).getCActividad();
        cRepartidor = CFactory.getCFactory(bd).getCRepartidor();
        cTarjeta = CFactory.getCFactory(bd).getCTarjeta();
        error.setVisible(false);
        lblRepartidor.setText("");
    }
}
```

No sabemos si cActividad, cRepartidor y cTarjeta se conecta a MySQL o a Access; CFactory se encarga de darnos la clase que queremos según el archivo de configuración (datos.ini)

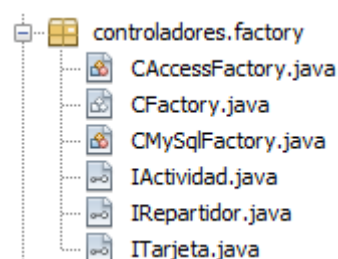
datos.ini	
1	BD = Access
2	URL = jdbc:ucanaccess://F:\\bd_actividad_economica.accdb
3	USER =
4	PASSWORD =

Configuración del archivo datos.ini para conectarse a Access.

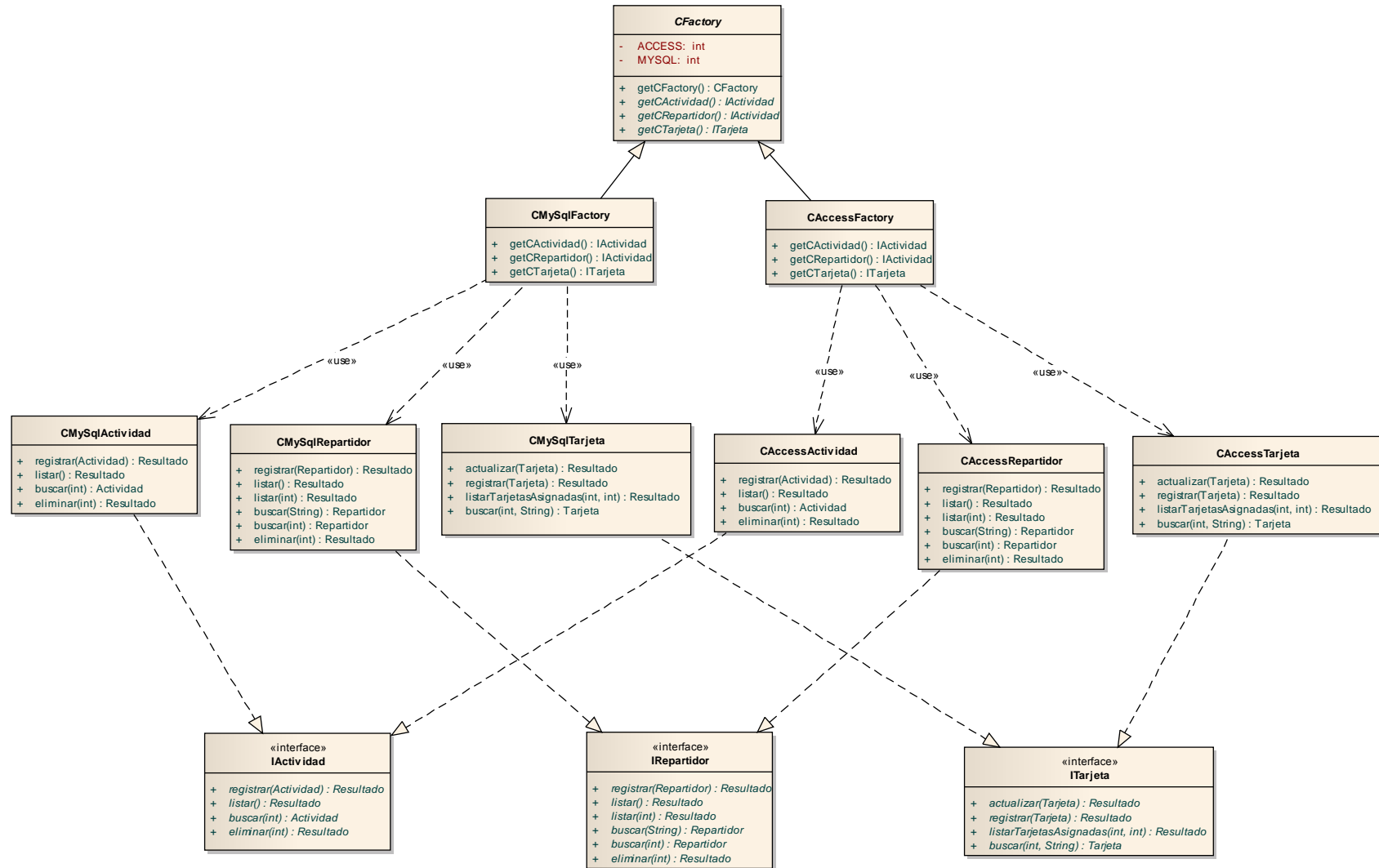
datos.ini	
1	BD = MySQL
2	URL = jdbc:mysql://localhost/bd_actividad_economica
3	USER = root
4	PASSWORD =

Configuración del archivo datos.ini para conectarse a MySQL

## Clases del patrón Abstract Factory



## Diagrama UML del patrón Abstract Factory del sistema de información Gestión de actividad económica





Revisen el código, sólo modifiqué el constructor de cada clase de interfaz gráfica.

Si aún no comprenden el patrón Abstract Factory, les invito a revisar en la web, ahí hay ejemplos con carritos, animalitos, je je ;)

Para mi suerte los reportes funcionaron perfectamente; las consultas sql de mysql y Access son parecidas (excepto la de seleccionar fecha “#”).

## PRUEBAS

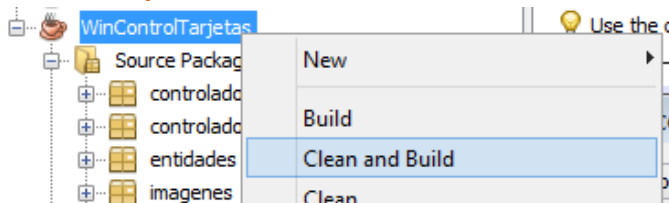
Existen sistemas de pruebas como la caja blanca, negra, de funcionalidad, de usabilidad, entre otros; y lo realizan diferentes usuarios (Empresas, usuarios comunes, desarrolladores).

No olvidar las normas de calidad.

Este software sólo lo probé yo, es muy probable que algún usuario común encuentre un pequeño error... :/

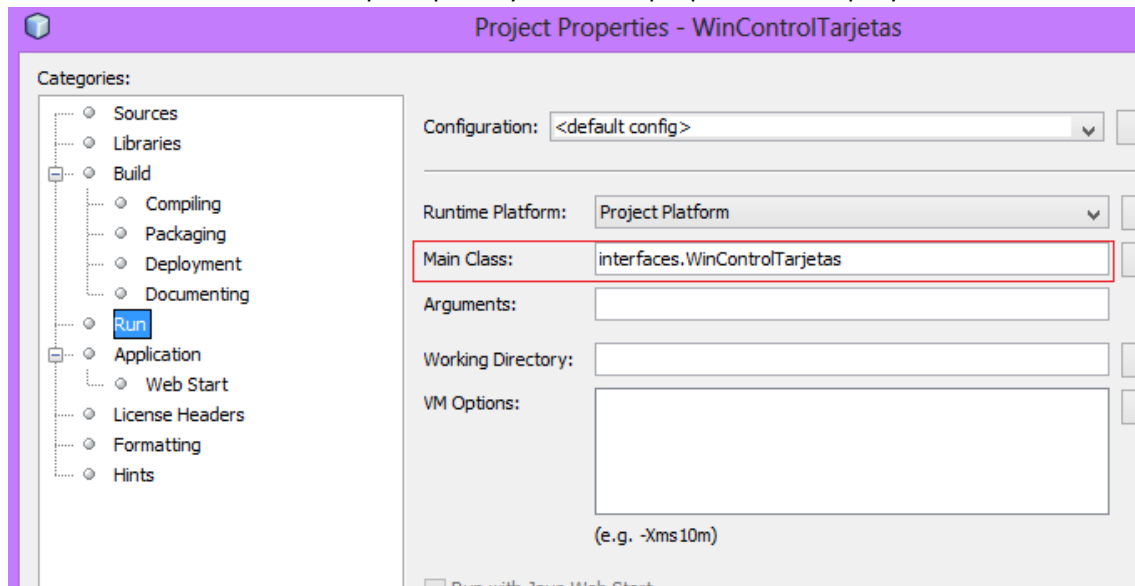
## DESPLIEGUE

### Creando ejecutable



Les pedirá seleccionar la clase principal, sabemos que para correr un proyecto escrito en Java se inicia por la clase que contenga el método main.

Otra forma de indicar la clase principal es yendo a las propiedades del proyecto:



Bien, en el directorio donde fue creado el proyecto se habrá creado una carpeta “dist” ahí está nuestro jar.

Equipo ▸ ARCHIVOS (F:) ▸ sistemas ▸ SISTEMA CONTROL TARJETAS ▸ WinControlTarjetas ▸ dist ▸				
Nombre	Fecha de modifica...	Tipo	Tamaño	
lib	24/02/2015 08:09 a...	Carpeta de archivos		
README.TXT	24/02/2015 08:09 a...	Documento de tex...	2 KB	
WinControlTarjetas.jar	24/02/2015 08:09 a...	Executable Jar File	215 KB	

Creamos cualquier carpeta “SIGEAE” y ahí pondremos nuestro jar, la carpeta lib, la carpeta reports (sólo los archivos jasper), y el archivo datos.ini

equipo ▸ ARCHIVOS (F:) ▸ SIGEAE	
Nombre	Fecha
lib	24/02/
reports	24/02/
datos.ini	22/02/
WinControlTarjetas.jar	24/02/

Listo, sólo crear un acceso directo y ya.

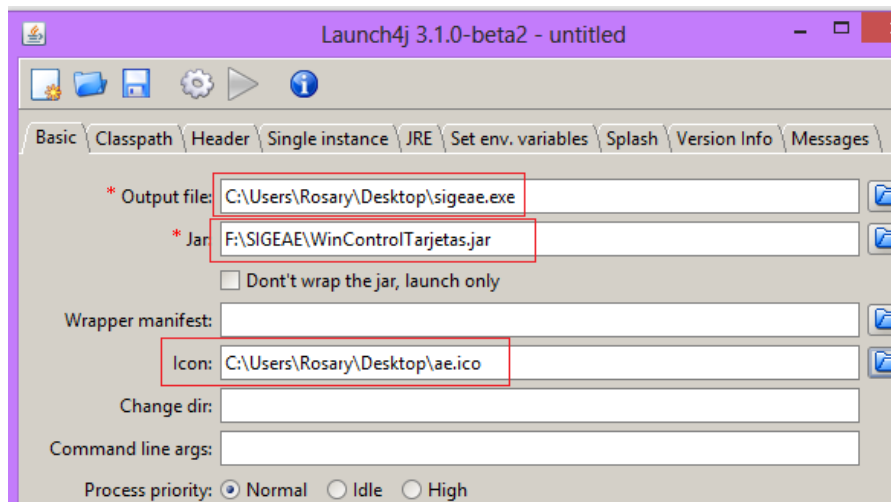
Pero un usuario común quiere “next, next y finish”; hay muchos programas que nos ayudan a crear su ejecutable .exe de un programa escrito en java. En mi caso utilicé 2 programas: JLaunch4j e Inno Setup Compiler.

### Creando el .exe

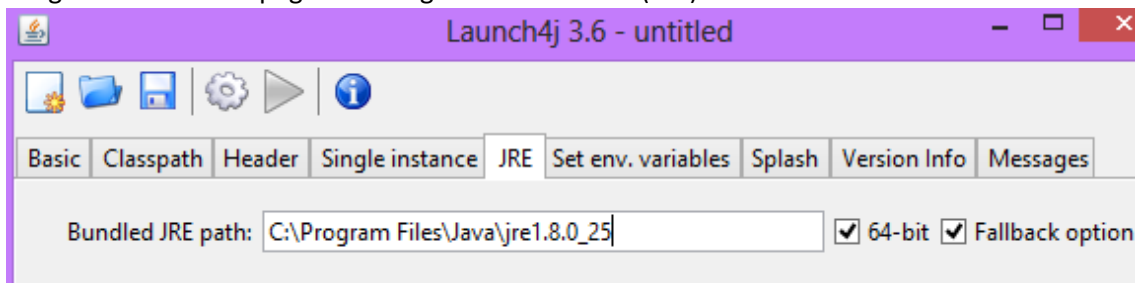
Copiar todas las librerías necesarias en el siguiente directorio

Java ▸ jdk1.8.0_25 ▸ jre ▸ lib ▸ ext				
Buscar en ext				
Nombre	Fecha de modifica...	Tipo	Tamaño	
access-bridge-64.jar	14/10/2014 09:35 ...	Executable Jar File	184 KB	
cldrdata.jar	14/10/2014 09:35 ...	Executable Jar File	3,771 KB	
commons-beanutils-1.8.2.jar	24/02/2015 08:09 a...	Executable Jar File	227 KB	
commons-collections-3.2.1.jar	24/02/2015 08:09 a...	Executable Jar File	562 KB	
commons-digester-2.1.jar	24/02/2015 08:09 a...	Executable Jar File	193 KB	
commons-javaflow-20060411.jar	24/02/2015 08:09 a...	Executable Jar File	95 KB	
commons-lang-2.6.jar	24/02/2015 08:09 a...	Executable Jar File	278 KB	
commons-logging-1.1.1.jar	24/02/2015 08:09 a...	Executable Jar File	59 KB	
dnsns.jar	14/10/2014 09:35 ...	Executable Jar File	9 KB	
edisoncorSX.jar	24/02/2015 08:09 a...	Executable Jar File	3,980 KB	
hsqldb.jar	24/02/2015 08:09 a...	Executable Jar File	1,363 KB	
iText-2.1.7.jar	24/02/2015 08:09 a...	Executable Jar File	1,104 KB	
jaccess.jar	14/10/2014 09:35 ...	Executable Jar File	44 KB	
jackcess-2.0.8.jar	24/02/2015 08:09 a...	Executable Jar File	788 KB	
jasperreports-4.7.1.jar	24/02/2015 08:09 a...	Executable Jar File	3,908 KB	
jasperreports-fonts-4.5.0.jar	24/02/2015 08:09 a...	Executable Jar File	2,422 KB	

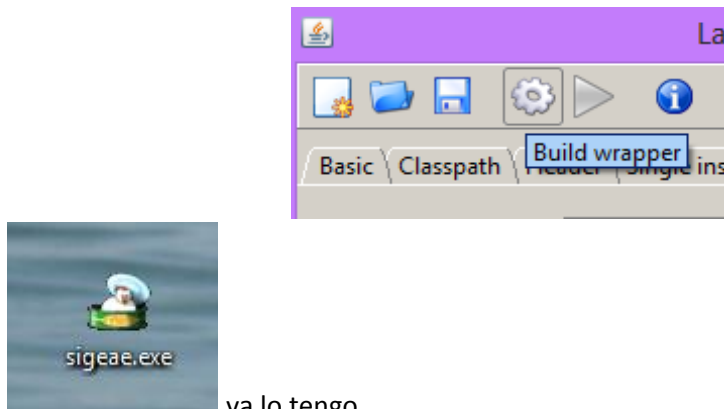
Ejecutamos el Launch4j y establecemos las siguientes propiedades: el archivo de salida, el jar compilado y un ícono de la aplicación.



Luego en la ficha JRE pegamos el siguiente directorio (JRE)



Y finalmente



Listo ya lo tengo.

Ahora ejecutamos el **Inno Setup Compiler** y creamos un nuevo archivo

Sólo es configurar los parámetros necesarios

Inno Setup Script Wizard

**Application Information**  
Please specify some basic information about your application.

**Application name:**  
Sigeae

**Application version:**  
1.0

**Application publisher:**  
Javashi

**Application website:**  
<http://www.javashi.wordpress.com/>

bold = required

< Back Next > Cancel

Inno Setup Script Wizard

**Application Folder**  
Please specify folder information about your application.

**Application destination base folder:**  
(Custom)

D:\Javashi

**Application folder name:**  
Sigeae

☐ Allow user to change the application folder

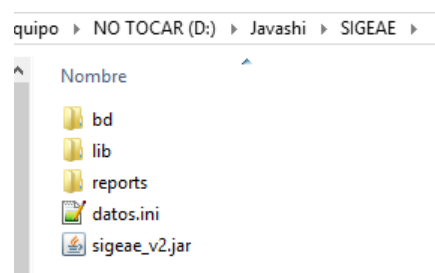
Other:

☐ The application doesn't need a folder

bold = required

< Back Next > Cancel

Aquí hacemos algo, en nuestra carpeta en donde se encuentra todo el ejecutable creamos la carpeta bd y en ella colocamos la base de datos en Access y el script de MySQL (es decir, todos los recursos necesarios).

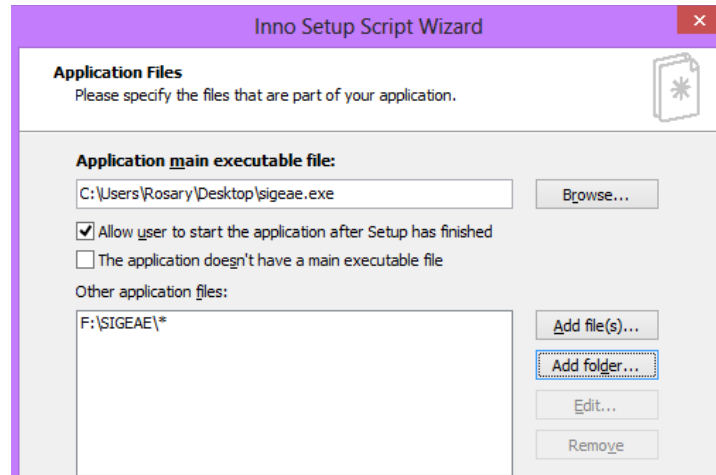


También se configura la nueva ubicación de la base de datos en el archivo datos.ini

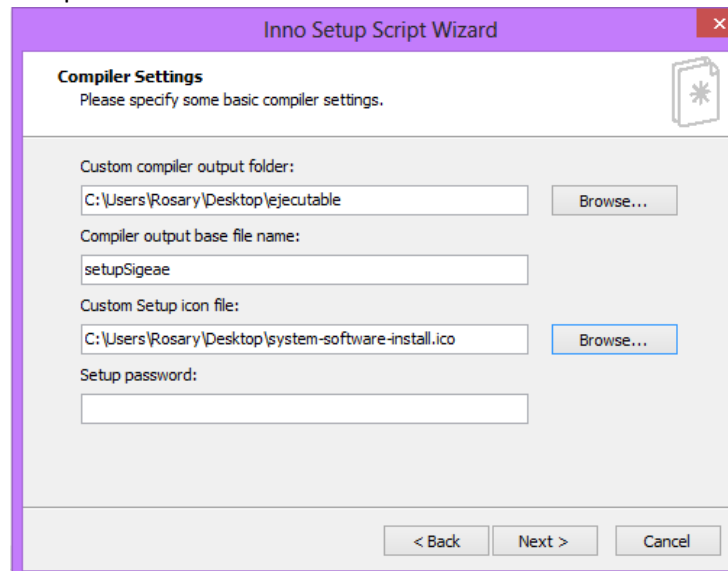
datos.ini			
1	BD	=	Access
2	URL	=	jdbc:ucanaccess://D:\\Javashi\\SIGEAE\\bd\\bd_actividad_economica.accdb
3	USER	=	
4	PASSWORD	=	

Obs: indiqué la unidad D puesto que en el SO Windows 8 me deniega el acceso a ciertos archivos alojados en la unidad C (o al menos no sé cómo resolverlo).

Ahora continuamos con la configuración del instalador. Indicamos el archivo .exe creado; y también la carpeta en donde pusimos nuestros archivos.



De ahí el resto de los pasos se resuelve intuitivamente



Al final se compila el script y si gustan lo guardan.

## RETO

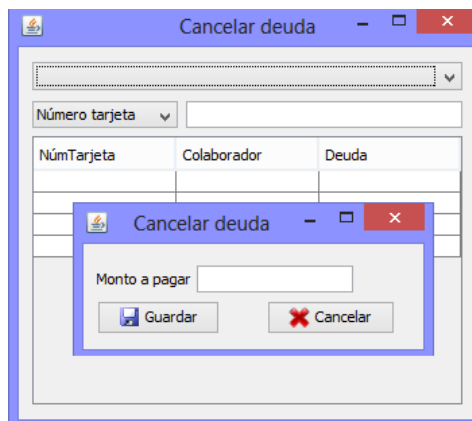
Rafitas de seguro, al iniciar las clases, me pagará lo que debe quizás todo o quizás una parte. Necesito que el sistema actualice el monto pagado de cada tarjeta.

Si tu objetivo es aprender te reto a implementar el siguiente requerimiento

- El administrador podrá ser capaz de cancelar la deuda indicando la actividad; número de tarjeta o nombre del colaborador, y, registrar el monto a cancelar.

Descripción:

- El administrador selecciona la actividad económica y el sistema muestra la lista de deudores ordenados por número de tarjeta.
- El administrador selecciona la opción de búsqueda, por número de tarjeta o por nombre del colaborador; ingresa el texto a buscar y el sistema filtra la lista de resultado según el texto tipeado.
- El administrador da doble clic en el deudor y el sistema le muestra el cuadro de diálogo para ingresar el monto a pagar de la deuda.
- El administrador ingresa el monto a pagar y el sistema le muestra la lista de deudores actualizada.
- Si el monto ingresado es inválido el sistema muestra un mensaje de atención indicando que ingrese un monto válido.



“El analista pide deseos y el programador hace **magia**”

*Rosario Alvarez*