

Ingeniería del Software II.

Curso 2003/2004.

Enunciado de la práctica del Segundo parcial.

Para este parcial, debe modificarse la práctica del primer parcial de la siguiente manera:

**Servidor:** la aplicación que actúa de **servidor** deberá ser capaz de generar diferentes tipos de código; así, el cliente le enviará un parámetro de tipo *int* que, en caso de valer cero, devolverá las clases Java con el mismo formato que en la práctica del primer parcial; si el parámetro vale uno, devolverá las clases Java con parecido formato, pero incluyendo algunas diferencias (véase Anexo I al final de este documento). El formato del código XML recibido es el mismo que en la práctica anterior.

**Cliente:** la aplicación **cliente** deberá ser capaz de generar la estructura XML a partir de una base de datos de SQL Server como de Access. Véase Anexo II al final de este documento para obtener una pista de cómo recuperar la estructura de tablas y relaciones de una base de datos Access. El formato del código XML que envía al servidor es el mismo que en la práctica anterior.

Consideraciones:

**Servidor:** la generación del código deberá utilizar obligatoriamente el patrón Abstract Factory o alguna variante que deberá representarse en la documentación mediante algún diagrama de clases construido con una herramienta. Considérese que los productos que genera el servidor son clases, imports y métodos, habiendo dos tipos de cada producto (los de tipo cero y los de tipo uno).

**Cliente:** la generación del código deberá utilizar obligatoriamente el patrón Abstract Factory o alguna variante que deberá representarse en la documentación mediante algún diagrama de clases construido con una herramienta. Habrá por tanto dos fábricas o similares: una para Access y otra para SQL Server. Sólo se creará un tipo de producto (el fichero XML).

**Para ambas aplicaciones:** básicamente, la documentación que debe entregarse consistirá en casos de prueba de JUnit, siendo obligatorio para el desarrollo de esta práctica utilizar desarrollo dirigido por las pruebas.

Cualquier duda se atiende en horario de tutorías.

Macario Polo Usaola

Despacho 3.21

Teléfono 926.295300 ext. 3730

[macario.polo@uclm.es](mailto:macario.polo@uclm.es)

## Anexo I. Aspecto de los dos tipos de ficheros Java generados.

```
// Resultado con el parámetro valiendo cero
package dominio;

import java.sql.*;

public class Alumno
{
    protected String mPKNIF;
    protected String mNombre;
    protected String mApellidos;
    protected String mDomicilio;
    protected String mLocalidad;
    protected String mCodigoPostal;
    protected String mTelefono;
    protected String mLogin;
    Matricula[] mMatriculas;

    public Alumno()
    {
        mPKNIF=null;
        mNombre=null;
        mApellidos=null;
        mDomicilio=null;
        mLocalidad=null;
        mCodigoPostal=null;
        mTelefono=null;
        mLogin=null;
    }

    public Alumno(Connection bd, String nif) throws SQLException
    {
        String SQL="Select * from Alumno where NIF=?";
        PreparedStatement p=bd.prepareStatement(SQL);
        p.setString(1, nif);
        ResultSet r=p.executeQuery();
        if (r.next())
        {
            mPKNIF=r.getString(1);
            mNombre=r.getString(2);
            mApellidos=r.getString(3);
            mDomicilio=r.getString(4);
            mLocalidad=r.getString(5);
            mCodigoPostal=r.getString(6);
            mTelefono=r.getString(7);
            mLogin=r.getString(8);
        }
        p.close();
    }

    public void insert(Connection bd) throws SQLException
    {
        String SQL="Insert into Alumno values (?, ?, ?, ?, ?, ?, ?, ?)";
        PreparedStatement p=bd.prepareStatement(SQL);
        p.setString(1, this.mPKNIF);
        p.setString(2, this.mNombre);
        p.setString(3, this.mApellidos);
        p.setString(4, this.mDomicilio);
        p.setString(5, this.mLocalidad);
        p.setString(6, this.mCodigoPostal);
    }
}
```

```

        p.setString(7, this.mTelefono);
        p.setString(8, this.mLogin);
        p.executeUpdate();
        p.close();
    }

    public void delete(Connection bd) throws SQLException
    {
        String SQL="Delete from Alumno where NIF=?";
        PreparedStatement p=bd.prepareStatement(SQL);
        p.setString(1, this.mPKNIF);
        p.executeUpdate();
        p.close();
    }
}

// Resultado con el parámetro valiendo uno (en negro se han resaltado las diferencias)
package dominio;

import java.sql.*;
import persistencia.Broker;

public class Alumno
{
    protected String mPKNIF;
    protected String mNombre;
    protected String mApellidos;
    protected String mDomicilio;
    protected String mLocalidad;
    protected String mCodigoPostal;
    protected String mTelefono;
    protected String mLogin;
    Matricula[] mMatriculas;

    public Alumno()
    {
        mPKNIF=null;
        mNombre=null;
        mApellidos=null;
        mDomicilio=null;
        mLocalidad=null;
        mCodigoPostal=null;
        mTelefono=null;
        mLogin=null;
    }

    public Alumno(String nif) throws SQLException
    {
        String SQL="Select * from Alumno where NIF=?";
        PreparedStatement p=Broker.prepareStatement(SQL);
        p.setString(1, nif);
        ResultSet r=p.executeQuery();
        if (r.next())
        {
            mPKNIF=r.getString(1);
            mNombre=r.getString(2);
            mApellidos=r.getString(3);
            mDomicilio=r.getString(4);
            mLocalidad=r.getString(5);
            mCodigoPostal=r.getString(6);
            mTelefono=r.getString(7);
        }
    }
}

```

```

        mLogin=r.getString(8);
    }
    p.close();
}

public void insert() throws SQLException
{
    String SQL="Insert into Alumno values (?, ?, ?, ?, ?, ?, ?, ?)";
    PreparedStatement p= Broker.prepareStatement(SQL);
    p.setString(1, this.mPKNIF);
    p.setString(2, this.mNombre);
    p.setString(3, this.mApellidos);
    p.setString(4, this.mDomicilio);
    p.setString(5, this.mLocalidad);
    p.setString(6, this.mCodigoPostal);
    p.setString(7, this.mTelefono);
    p.setString(8, this.mLogin);
    p.executeUpdate();
    p.close();
}

public void delete() throws SQLException
{
    String SQL="Delete from Alumno where NIF=?";
    PreparedStatement p= Broker.prepareStatement(SQL);
    p.setString(1, this.mPKNIF);
    p.executeUpdate();
    p.close();
}
}

```

## Anexo II. Recuperación de la estructura de una base de datos Access.

Microsoft Access no responde a muchas de las operaciones contenidas en las interfaces ResultSetMetaData y DatabaseMetaData que sí funcionan en SQL Server.

De este modo, los nombres de las tablas de una base de datos Access se obtienen de la columna *Name* de la tabla *MSysObjects* con el valor del atributo *Type=1*. Por ejemplo:

```

private void loadTables() throws SQLException
{
    String SQL="Select Name from MSysObjects where Type=1 and left(Name, 4)<>'MSys'";
    PreparedStatement p=mBD.prepareStatement(SQL); // mBD es una Connection
    ResultSet r=p.executeQuery();
    while (r.next())
    {
        String tableName=r.getString(1); // El nombre de la tabla viene en la 1ª columna
    }
    p.close();
}

```

Las relaciones entre tablas se obtienen consultando las columnas *szRelationship* (nombre de la relación, formado por las dos tablas que intervienen), *szReferencedObject*

(nombre de la tabla principal de la relación), *szObject* (nombre de la tabla secundaria), *szReferencedColumn* (nombre de la columna principal) y *szColumn* (nombre de la columna secundaria) de la tabla *MSysRelationships*.

Las columnas que forman la clave principal de una tabla se pueden obtener de la siguiente forma:

```
private void loadPKs(String tableName) throws SQLException
{
    DatabaseMetaData meta=mBD.getMetaData();
    ResultSet r=meta.getIndexInfo(null, null, tableName, true, true);
    while (r.next())
    {
        String idx=r.getString(6);
        if (idx!=null && idx.equals("PrimaryKey"))
        {
            String colName=r.getString(9); // La 9ª columna guarda el nombre que buscamos
        }
    }
    r.close();
}
```

Por último, hay que señalar que las tablas *MSys...* son lo que Access llama "objetos del sistema"; para mostrarlos se marca la opción "Objetos del sistema" en el menú Herramientas→Opciones→Ver. Para tener acceso a estas tablas debemos dar en Access el correspondiente permiso mediante Herramientas→Seguridad→Permisos de usuario y de grupo:

