6. Construcción de una base de datos a partir de esquemas XML

En este capítulo[5] se muestra el desarrollo que se siguió para elaborar el constructor automático de la base de datos. Apareciendo al principio de este capítulo el algoritmo en el cual se muestra la forma en que debe de comportarse el constructor automático de la base de datos de una forma general, y poco a poco, se vuelve más complejo hasta dar como resultado la obtención de los comandos SQL, necesarios para la creación de la base de datos.

Algoritmo para el constructor automático de la base de datos

Después de que ya se tenía un modelado del constructor automático de bases de datos y se identificaron las características principales que deberían de tener, el siguiente paso fue analizar los posibles casos que se podían presentar al momento de analizar las etiquetas, para este fin se formuló el algoritmo mostrado en la figura 6.1. El algoritmo indica de una manera muy superficial, cómo se debe comportar el constructor automático de la base de datos, al momento de analizar el esquema XML.

Dándonos una idea general del comportamiento general del constructor automático de la base de datos, para posteriormente lograr la identificación de los

[5] La información presentada en este capítulo se obtuvo de revisar la siguiente fuente bibliográfica: MacDonald, Matthew, *Office 2003 XML for Power Users*, Apress, 2004 y la página http://www.elserver.com/wiki/SQL.

diferentes tipos de etiquetas y cómo clasificar cada una de éstas dependiendo de la información que contengan.

Algoritmo para analizar esquemas XML para construir la bases de datos

1. Recibir los datos de entrada necesarios, para que el constructor automático de la base de datos pueda ser ejecutado correctamente.

2. Preparar las etiquetas para que puedan se leídas correctamente. Quitando comentarios <!---- >, tabulaciones, espacios en blanco, etc. Dejando únicamente la información útil.

3. Al momento de ir leyendo el esquema XML, separar el documento por etiquetas <> y posteriormente almacenarlas en una lista o un arreglo.

4. La etiqueta a partir de la cual se debe iniciar la construcción de la base de datos debe de agregarse a una *lista* que busca las tablas y columnas de la base de datos.

5. iniciar el contador con i=0

6. Tomar el elemento 'i' de *lista*, para analizar

7. Buscar la etiqueta del elemento 'i' con el cual se quiere crear una tabla de la base de datos.

8. El elemento ‘i’, es el nombre de tabla que se desea crear.

9. Buscar cuáles de las etiquetas contenidas dentro de este elemento, son consideradas como columnas de la tabla y cuáles se deben agregar a *lista*, para analizarse posteriormente y obtener una nueva tabla

10. i=i+1

11. repetir los pasos del 6 al 10 hasta que los elementos contenidos en *lista*, se acaben.

12. Las instrucciones obtenidas producto de analizar el esquema, enviarlas a un archivo de texto.

Figura 6.1. Algoritmo para analizar esquemas XML

Es importante mencionar que el algoritmo mostrado en la figura 6.1 no analiza todas las posibles estructuras en un esquema XML. Se limita, como se especificó arriba, a procesar las pertinentes para la construcción de la base de datos bibliográfica del Corpus Histórico del Español en México.

En términos generales lo que debe hacer el constructor automático de la base de datos, es agrupar la información del esquema XML para que pueda ser procesada, quitando toda información innecesaria, como pueden ser los comentarios o espacios en blanco. Para analizar dicho esquema, las etiquetas se agrupan en diferentes conjuntos, el *conjunto1* contiene todas las etiquetas del esquema XML, dicho conjunto contiene todas las etiquetas del esquema XML, listadas de la misma forma en que aparecen originalmente en el esquema.

Se busca en el *conjunto1*, si contiene la etiqueta de la forma:

```
<xsd:element name=”elemento_a_buscar”>
```

En caso de encontrar la etiqueta con esas características, el nombre =**”elemento_a_buscar”** se utiliza para nombrar la tabla inicial de la base de datos. A partir de esta y las siguientes etiquetas contenidas dentro de *conjunto1*, se obtiene la información para la estructura de la tabla de la base de datos. Desde el elemento `<xsd:element name=”elemento_a_buscar”>` hasta su etiqueta de cierre `</xsd:element>`.

Después se espera que dentro de la información almacenada en *conjunto1* se encuentren etiquetas del tipo: atributos `<xsd:attribute name="NOMBRE" …>`, y elementos `<xsd:element ref="NOMBRE" …>`. Dichas etiquetas se deben de analizar, principalmente si tales etiquetas son simples o complejas. En caso de ser simples quiere decir que su información se concreta a un tipo de dato (ej. cadena de caracteres, número, un carácter, etc.). Las simples se toman como columnas de la tabla, siempre que el número de apariciones de tal definición sea única. Las complejas, aquellas cuya información está en diversos tipos de datos, corresponden a subtablas, que se van a guardar en una lista, para analizarlas en un proceso recursivo hasta ya no tener más posibles subtablas que analizar.

Una vez que se detectó cuales etiquetas se deben buscar, para iniciar la creación de la base de datos relacional, lo que siguió fue elegir qué estructura de

datos se podría utilizar para almacenar tanto el *conjunto1*, como la *lista* que vaya guardando otras posibles subtablas, y otras que pudieran surgir para guardar nuevas listas.

Se requería, una estructura de datos que fuera dinámica, ya que no se sabe cuál es el número de elementos o etiquetas que se tienen que almacenar y posteriormente ser leídas, y ya que el constructor y alimentador automático de la base de datos se realizaron en el lenguaje de programación de Java. La *clase* que mejor cumplía con las condiciones para realizarlos, son la *clase* tipo *Vector*, que por sus características de crecer de forma automática a medida que se agreguen elementos, y adicionalmente nos brinda diferentes herramientas para agregar, eliminar o insertar elementos en una posición especifica, se decidió utilizar dicha *clase*.

Una vez que se tuvo definido la forma en que debe comportarse el algoritmo y las herramientas básicas que se requerían dentro del lenguaje de programación a utilizar, el siguiente paso, fue realizar un algoritmo mas detallado, que incluyera el comportamiento del constructor automático de la base de datos, dependiendo de los diferentes tipos de etiquetas, y agregando dentro del algoritmo las estructuras tipo *Vector* necesarias para guardar las diferentes listas de etiquetas del esquema XML. Llegando al siguiente algoritmo mas detallado que se presenta a continuación.

Los elementos básicos que el constructor automático de la base de datos debe tener como datos entrada, para poder leer el esquema XML y poder construir la base de datos relacional con extensión XSD son:

- El nombre del archivo que contiene el esquema XML, con terminación XSD.
- El nombre de la etiqueta, a partir de la cual se empieza a realizar la construcción de la base de datos.

También es importante que se especifique cuál atributo o etiqueta se considera que es la llave primaria de la primera tabla que se genere, para que a partir de esta, se puede generar una base de datos relacional.

Una vez que ya se tienen los datos necesarios para que el constructor pueda empezar a leer el esquema XML, se debe de ir seleccionando cuál información contenida dentro del archivo es útil y cuál no es necesario considerar, como espacios en blanco, tabulaciones, comentarios, etc. Quedando únicamente la etiquetas del esquema XML.

Estas etiquetas se almacenan en el orden del archivo original, con la diferencia de que se pueden identificar las etiquetas individualmente, y no existen elementos innecesarios. Esta información puede ser consultada o leída, para obtener partes del documento según convenga. Dichas etiquetas se almacenan en *vector1*.

Adicionalmente se crean otros dos objetos tipo *vector*, llamado *vector4* y *vector5*. *vector4* contiene los elementos que pueden ser considerados como tabla de la base de datos a construir. vector5 va almacenando las instrucciones SQL, producto del recorrido del esquema XML.

El primer elemento que se agrega a *vector4*, es el nombre de la etiqueta por la cual se quiere empezar a realizar la construcción de la base de datos. Para el *vector5*, el primer elemento que se crea es que contiene la instrucción de crear la base de datos (aunque puede omitirse si el dato de entrada de la base de datos es nulo).

Una vez que se introdujo la etiqueta a buscar en el *vector4*, comienza un proceso iterativo, en el que se van extrayendo los elementos contenidos dentro de *vector4*, en orden ascendente. Empezando por el *elemento_a_buscar,* hasta que se agoten los elementos contenidos dentro de dicho *vector4*.

Como ya se comentó se extrae el primer elemento de *vector4* y se busca dentro de las etiquetas contenidas dentro de *vector1*, la etiqueta que contenga la forma siguiente:

:<xsd:element name="***elemento_ a_buscar***">

Una vez encontrada, se busca si la etiqueta contiene en su parte final autocierre, en caso de ser afirmativo, únicamente se almacena dicha etiqueta en el vector2. En caso de que la etiqueta no presente el autocierre, se obtienen las etiquetas siguientes hasta encontrar la etiqueta de cierre de dicha etiqueta. Almacenándose esas etiquetas en *vector2*.

A partir de aquí se trabaja con los elementos contenidos dentro de *vector2*, que contiene la información para crear el nombre de una tabla, las columnas de la tabla de la base de datos, y las posibles nuevas tablas que se pueden generar.

Se pueden obtener 3 casos en el análisis de la información contenida en vector2, los cuales pueden ser que sólo se obtenga una etiqueta, que se obtengan mas de una, y que no se obtengan etiquetas. En caso de no obtener etiquetas, quiere decir que *elemento_a_buscar* no se encuentra dentro del esquema XML, y por lo tanto no se pueden crear las tablas de la base de datos.

En caso de que el resultado de la búsqueda dé sólo una etiqueta, se debe buscar si dicha etiqueta es simple o es compleja. Si es simple significa que la información que pretende delimitar, no contiene dentro de esta mas etiquetas. Si es compleja quiere decir que dentro de esta etiqueta contiene otras etiquetas.

Los casos que se pueden presentar, es que su tipo sea de tipo –entero, texto, char, etc-, o que sea compuesto o sea que su tipo se derive en otra serie de

etiquetas. Para ellas se debe de analizar su tipo, si tiene atributos o esta referenciada hacia otras.

Verificando básicamente 3 aspectos, que tengan terminación de texto simple o enteros**,** el número de ocurrencias de las etiquetas, y si se encuentran dentro de el elemento.

En caso de que alguno de los elementos tenga una ocurrencia mayor a 1, agregar el nombre de la etiqueta a vector4.

Para varias etiquetas se debe de revisar si el elemento es *choice* o no es *choice*, ya que si es *choice*, se debe verificar la máxima ocurrencia de la etiqueta.

Los atributos contenidos dentro de *vector2,* generalmente van a ser, columnas de la base de datos, ya que son elementos simples.

Las etiquetas que tienen alguna referencia hacia algo, se deben de buscar si su tipo es de texto simple o números enteros o es una composición de otras etiquetas, en caso de ser simple, pueden ser columnas de la tabla, siempre y cuando su ocurrencia en la tabla sea igual o menor a 1.

En caso de ser *choice*, se debe de verificar, cuál es el máximo de repeticiones que se puede escoger un elemento. En caso de ser igual a 1, se debe analizar si el número de ocurrencias puede afectar el almacenamiento de los

datos, y si no afecta considerarlo columna de la tabla, en caso contrario agregarlo a *vector4*, para analizarlo como una nueva tabla.

Al finalizar el análisis de todas las posibles tablas que pudieran generarse y que se encuentran almacenadas en *vector4*, el siguiente paso es ejecutar los comandos SQL guardados en *vector5,* resultado de ir recorriendo las diferentes etiquetas del esquema XML.

Comandos SQL para crear la base de datos a partir del algoritmo

Cómo se explicó en la parte de anterior, los comandos SQL necesarios para crear la base de datos se fueron almacenando en *vector5,* inferidos al recorrer la estructura del esquema XML. Una vez que se ha recorrido la estructura del esquema en su totalidad, se ejecutan dichas instrucciones. Esencialmente, el comando que se utiliza para la creación de la base de datos es:

```
CREATE DATABASE <NOMBRE_BD>;
```

Que crea la base de datos, donde `<NOMBRE_BD>`, es el nombre de dicha base. Esta instrucción se inserta en la lista desde el inicio de la ejecución del constructor, ya que se da por hecho que el elemento a buscar se encuentra en el esquema XML y en caso de no encontrarse no ejecutan las instrucciones SQL.

Posteriormente, dependiendo del recorrido del esquema XML, se insertan en la lista comandos de creación de las tablas pertinentes, con los atributos o

elementos que se obtienen del análisis. El tipo de columna puede variar dependiendo de la manera en que esté declarada en el esquema XML (ya sea texto, número entero, etc.).

Así por ejemplo, de la siguiente parte del esquema XML, se obtienen las instrucciones SQL mostradas abajo:

```
<xsd:element name="encabezado">
       <xsd:complexType>
              <xsd:sequence>
                     <xsd:element ref="titulo" minOccurs="1" maxOccurs="1"/>
                     <xsd:element ref="descripcion" minOccurs="0" maxOccurs="1"/>
                     <xsd:element ref="parametros" minOccurs="1" maxOccurs="1"/>
                     <xsd:element ref="autor" minOccurs="0" maxOccurs="unbounded"/>
                     <xsd:element ref="institucion" minOccurs="0" maxOccurs="1"/>
                     <xsd:element ref="referencia" minOccurs="1" maxOccurs="unbounded"/>
                     <xsd:element ref="ciudadPublicacion" minOccurs="1" maxOccurs="1"/>
                     <xsd:element ref="fechaOriginal" minOccurs="0" maxOccurs="1"/>
                     <xsd:element ref="fechaPublicacion" minOccurs="1" maxOccurs="1"/>
                     <xsd:element ref="enlace" minOccurs="0" maxOccurs="1"/>
                     <xsd:element ref="imagen" minOccurs="0" maxOccurs="1"/>
                     <xsd:element ref="responsables" minOccurs="1" maxOccurs="1"/>
              </xsd:sequence>
              <xsd:attribute name="id" use="optional" type="enteroPositivo"/>
              <xsd:attribute name="archivo" use="optional">
                     <xsd:simpleType>
                            <xsd:restriction base="xsd:token">
                                   <xsd:pattern value="[0-9a-záäéëíïñóöúü_-]+.xml"/>
                            </xsd:restriction>
                     </xsd:simpleType>
              </xsd:attribute>
       </xsd:complexType>
</xsd:element>
```

Figura 6.2. Esquema XML

```
CREATE TABLE encabezado (
      id int,
      archivo text,
      titulo text,
      descripcion text,
      institucion text,
      ciudadPublicacion text,
      fechaOriginal int,
      fechaPublicacion int,
      enlace text
);
```

Figura 6.3. Instrucción SQL

Las etiquetas que no aparecen como instrucciones SQL, como *autor, responsables,* entre otras, debido a la características en que están definidas en el esquema XML, no se consideran columnas de la tabla que se está creando y por lo tanto deben agregarse a *vector4*, para posteriormente ser analizadas y así crear las otras tablas pertinentes, hasta que ya no existan otros elementos que deban ser analizados, que se encuentren contenidos en *vector4,* y como ya se dijo anteriormente producto de ir recorriendo el esquema XML.

Así finaliza la explicación del funcionamiento del constructor automático de la base de datos. En el siguiente capítulo se describe el funcionamiento del alimentador de la base de datos.