

# **Implementación y pruebas**

---

## **CAPÍTULO 6**

## CAPÍTULO 6 “Implementación y pruebas”

En este capítulo se va a mostrar el trabajo de programación que se realizó en OGGDB y las pruebas que se realizaron para calificar su desempeño. Todas las clases fueron programadas específicamente para este proyecto excepto el paquete de WKB (dentro del paquete Guardar), el cuál fue tomado sin hacerle modificaciones del proyecto de Manuel Pech [Pech, 2002]. También se programaron de manera tal que pudieran ser reutilizadas y/o modificadas con facilidad.

### 6.1 Implementación

#### 6.1.1 Paquetes del sistema OGGDB

El sistema se encuentra dividido en cuatro paquetes, los cuales se muestran en la figura 6.1:

- Paquete de Lectura de documento GML y almacenamiento en la base de datos (Guardar) Este paquete contiene el paquete para la codificación de las geometrías (WKB).
- Paquete de interfaces (Interfaces)
- Paquete de *features* (*features*)
- Paquete de consultas y generación de documento GML (Recuperar)

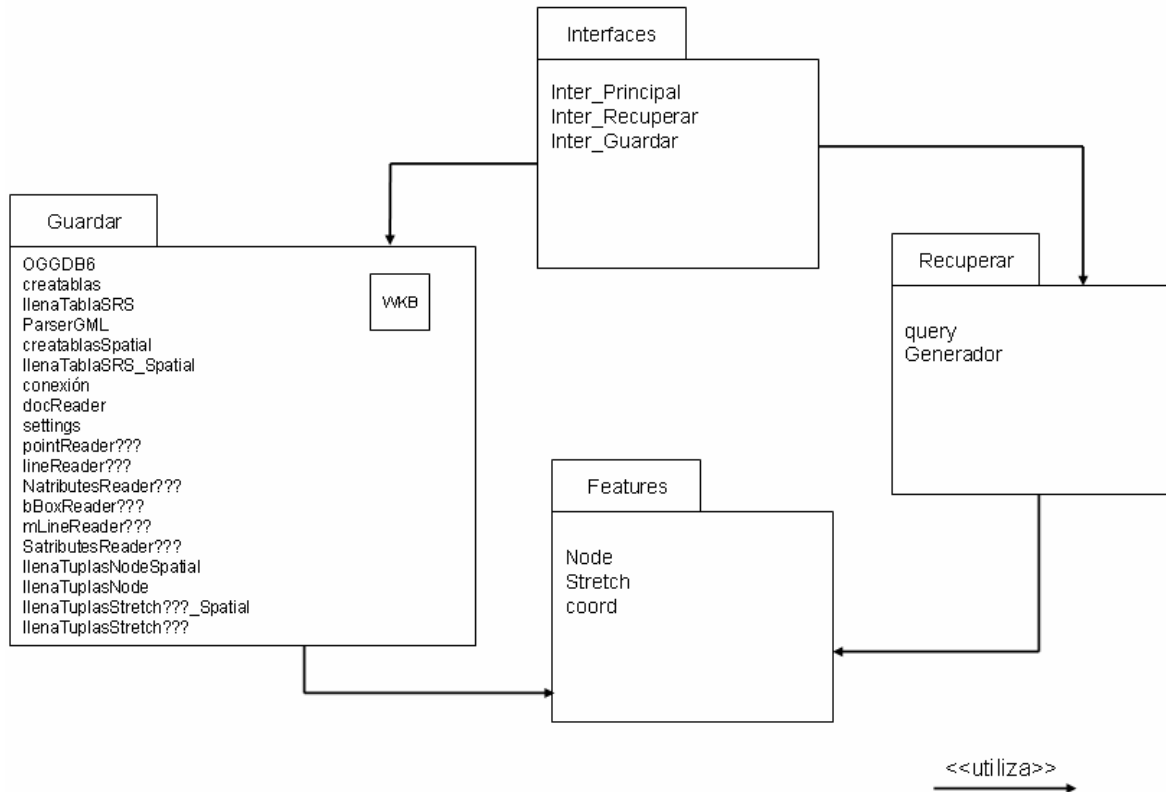


Figura 6.1 Paquetes del sistema OGGDB

### 6.1.2 Relación de clases y la base de datos.

Para la creación, llenado y consulta de la base de datos se crearon clases especializadas. El esquema de la base de datos se muestra en el Apéndice B. A continuación se describirán las clases que manejan alguna acción con la base de datos, todas utilizan la clase *conexion* para establecer la comunicación con la base de datos. La clase *creatablas* tiene definidos enunciados para crear cada una de las tablas. Los enunciados son variables de tipo *String* con la instrucción en SQL para la creación de tablas. A continuación se muestran algunos ejemplos, si se desea se puede revisar el apéndice B para conocer la estructura de cada tabla.

1.- Enunciado para la creación de la tabla *Spatial\_Ref\_Sys*.

```
String s_create= "CREATE TABLE SPATIAL_REF_SYS ";
String s_atributo1 = "SRID INTEGER NOT NULL PRIMARY KEY, ";
String s_atributo2 = "AUTH_NAME VARCHAR(255), ";
String s_atributo3 = "AUTH_SRID INTEGER, ";
String s_atributo4 = "SRTEXT VARCHAR(255) ";

createSRS = s_create + " ( " + s_atributo1 + s_atributo2 + s_atributo3;
createSRS += s_atributo4 + " ); " ;
```

2.- Enunciado para la creación de la tabla *Feature\_Table\_Stretch*

```
String ft_create_S = "CREATE TABLE FEATURE_TABLE_STRETCH ";
String ft_atributo1_S = "stretchID INTEGER NOT NULL PRIMARY KEY, ";
String ft_atributo2_S = "stretchName VARCHAR (255), ";
String ft_atributo3_S = "fromNode INTEGER, ";
String ft_atributo4_S = "toNode INTEGER, ";
String ft_atributo5_S = "pavementType VARCHAR (90), ";
String ft_atributo6_S = "numberOfLanes FLOAT, ";
String ft_atributo7_S = "speedLimit INTEGER, ";
String ft_atributo8_S = "length DOUBLE ";

createFeatureTableS = ft_create_S + " ( " + ft_atributo1_S + ft_atributo2_S;
createFeatureTableS += ft_atributo3_S + ft_atributo4_S + ft_atributo5_S ;
createFeatureTableS += ft_atributo6_S + ft_atributo7_S + ft_atributo8_S + " ); " ;
```

5.- Enunciado para la creación de la tabla *Geometry\_Table\_Stretch*

```
String gt_create_S = "CREATE TABLE GEOMETRY_TABLE_STRETCH ";
String gt_atributo1_S = "Xmin DOUBLE, ";
String gt_atributo2_S = "Ymin DOUBLE, ";
String gt_atributo3_S = "Xmax DOUBLE, ";
String gt_atributo4_S = "Ymax DOUBLE, ";
String gt_atributo5_S = "GID INTEGER NOT NULL PRIMARY KEY, ";
String gt_atributo6_S = "WKB_GEOMETRY BLOB ";

createGeometryTableS = gt_create_S + " ( " + gt_atributo1_S + gt_atributo2_S;
createGeometryTableS += gt_atributo3_S + gt_atributo4_S + gt_atributo5_S;
createGeometryTableS += gt_atributo6_S + " ); " ;
```

8.- Enunciado para la creación de la tabla *Geometry\_Columns*

```
String g_create = "CREATE TABLE GEOMETRY_COLUMNS";
String g_atributo1 = "F_TABLE_CATALOG VARCHAR (96) NOT NULL, ";
String g_atributo2 = "F_TABLE_SCHEMA VARCHAR (96) NOT NULL, ";
String g_atributo3 = "F_TABLE_NAME VARCHAR (96) NOT NULL, ";
String g_atributo4 = "F_GEOMETRY_COLUMN VARCHAR (96) NOT NULL, ";
String g_atributo5 = "G_TABLE_CATALOG VARCHAR (255), ";
String g_atributo6 = "G_TABLE_SCHEMA VARCHAR (255), ";
String g_atributo7 = "G_TABLE_NAME VARCHAR (255), ";
String g_atributo8 = "STORAGE_TYPE INTEGER, ";
String g_atributo9 = "GEOMETRY_TYPE INTEGER, ";
String g_atributo10 = "COORD_DIMENSION INTEGER, ";
String g_atributo11 = "MAX_PPR INTEGER, ";
String g_atributo12 = "SRID INTEGER REFERENCES SPATIAL_REFERENCE_SYS, ";
String g_constraint = "CONSTRAINT GC_PK PRIMARY KEY (F_TABLE_CATALOG, ";
String g_constraint2 = "F_TABLE_SCHEMA, F_TABLE_NAME, F_GEOMETRY_COLUMN)";

createGeometryColumns = g_create + " { " + g_atributo1 + g_atributo2;
createGeometryColumns += g_atributo3 + g_atributo4 + g_atributo5 + g_atributo6;
createGeometryColumns += g_atributo7 + g_atributo8 + g_atributo9 + g_atributo10;
createGeometryColumns += g_atributo11 + g_atributo12 + g_constraint + " }";
```

La clase *llenaTablaSRS* llena las *tuplas* de la tabla de *Spatial\_Ref\_System* con la información del sistema de referencia espacial con el que se va a estar trabajando en las geometrías. La clase *llenaTuplasStretch* llena las *tuplas* de las tablas *Feature\_Table\_Stretch*, *Illegal\_Turning\_Relations*, *Stretch* y *Geometry\_Table\_Stretch* con la información de los tramos obtenida de la lectura de los documentos. La clase *llenaTuplasNode* llena las *tuplas* de las tablas *Feature\_Table\_Node* y *Geometry\_Table\_Node* con la información de los nodos obtenida de la lectura de los documentos. Todas estas clases definen enunciados de tipo *String*, para las instrucciones de *insertar* en la base de datos, similares a los mostrados para la clase *creatablas*.

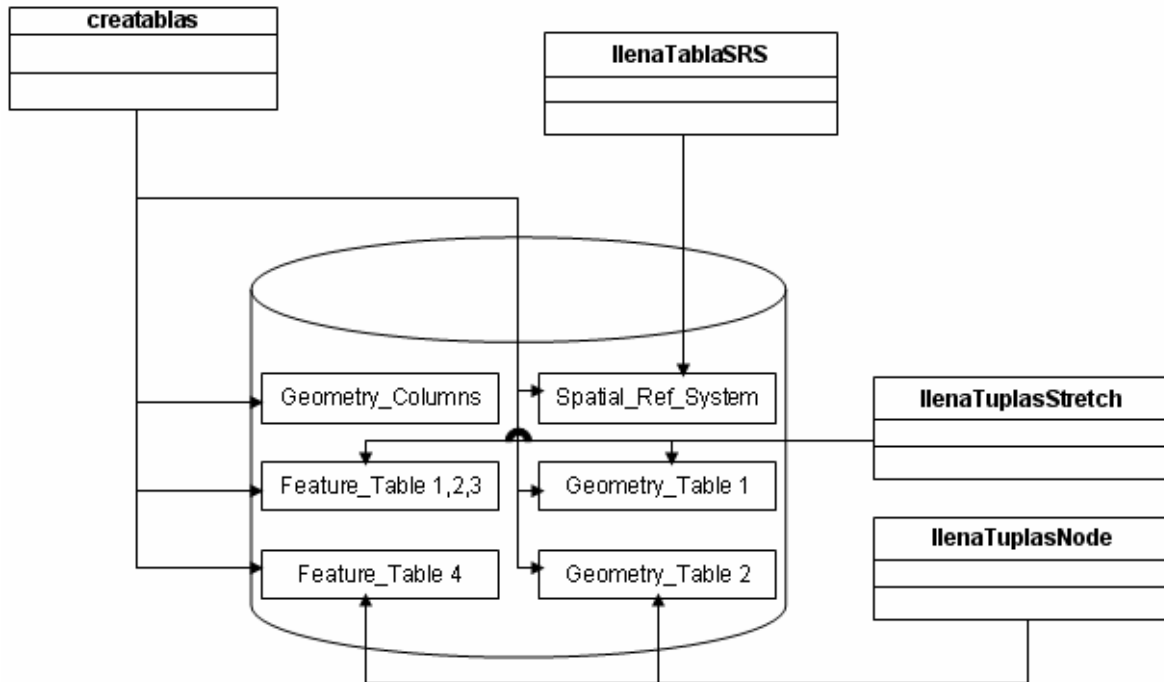


Figura 6.2 Relación de clases con la base de datos

## 6.2 Pruebas

Recordemos los ejemplos de Anthony y la Dra. Candy del capítulo 5. En las siguientes secciones se van a mostrar las pruebas que se realizaron al sistema en el contexto de esos ejemplos, pero las tablas completas de pruebas se pueden consultar en el Apéndice E. Las pruebas están clasificadas en dos: pruebas al proceso de Guardar y pruebas al proceso de Recuperar. De esta forma se estudiaron las respuestas del sistema a las posibles entradas del usuario.

Las tablas de pruebas a interfaces se dividen en 4 columnas, las de pruebas al nivel de la base de datos en 3 columnas. La columna “Opciones” muestra los botones y el campo de texto con los que puede interactuar el usuario. La columna de “Casos” muestra cuales pueden ser las acciones que lleve a cabo el usuario con cada opción, por ejemplo en el caso del campo de texto “Archivo” el usuario pudo haber seleccionado un archivo del directorio

y el nombre aparecerá automáticamente o pudo haberlo tecleado directamente. La columna de “Resultado esperado” muestra cuál es la respuesta que se esperaría del sistema, en algunos casos sólo se puede esperar una cosa, pero en otras se muestran más de una respuesta. Por último, la columna “Logró” muestra si se obtuvo la respuesta esperada, por parte del sistema, o no.

## 6.2.1 Pruebas para el proceso de Guardar

### 6.2.1.1 Pruebas a la interfaz

En esta sección se revisan las opciones con que se puede encontrar la Dra. Candy en el uso de la interfaz de guardar y como le va a responder el sistema. La interfaz de guardar recibe como datos de entrada los nombres de los archivos que contienen la información que se desea almacenar en la base de datos y el tipo de base de datos en el que se desea guardar (MySQL o MySQL *Spatial*). El nombre del archivo lo puede seleccionar de un directorio con el botón de “Abrir” o teclear “Arenas-Huejotzingo.xml” directamente en el campo de texto “Archivo”. La selección de la base de datos es mediante cajas de selección, hay uno para cada opción y sólo se necesita darles clic y una palomita de selección aparecerá en el cuadrado. Para nuestro ejemplo, supongamos que la Dra. Candy decide almacenar la información en la base de datos MySQL *Spatial*

Opciones	Casos	Resultado esperado	Logró
Abrir		Se debe abrir la ventana de "Abrir" en el directorio de "Mis Documentos"	Si
Archivo	Se seleccionó un archivo de la ventana de "Abrir"	Debe aparecer el nombre del archivo, con extensión, en el <i>textField</i> "Archivo"	Si
	Se tecleó un archivo en el <i>textField</i> "Archivo"	Debe permitir teclear y borrar el nombre del archivo	Si
Agregar Archivo	Se seleccionó un archivo	Debe de aparecer el nombre del archivo en la <i>textArea</i>	Si
Procesar	Se tiene un archivo en la lista	Debe aparecer un mensaje de procesando documento y almacenamiento exitoso, o	Si

		que no se encontró el documento.	
MySQL/MySQL <i>Spatial</i>	Se selecciona a lo más una opción	Debe mandarse un indicador de la BD seleccionada al sistema	Si
Salir		Se debe cerrar la ventana de Guardar	Si
X		Se debe cerrar el sistema completo	Si

Tabla 6.1 Tabla de pruebas a la interfaz de Guardar en el contexto de la Dra. Candy

### 6.2.1.2 Pruebas al nivel de la base de datos

En esta sección se muestran las pruebas realizadas en cuanto al desempeño del sistema con diferentes archivos de entrada y en caso de ser correctos, si la información fue guardada exitosamente en la base de datos. Los archivos de entrada deben estar definidos por los esquemas que son aceptados por OGGDB<sup>1</sup>, estos esquemas establecen el formato de los datos y si la información es obligatoria u opcional. Las pruebas que se muestran a continuación se hicieron en base al ejemplo de la Dra. Candy, al ser un usuario experto se asumirá que el único error que ella puede cometer es que modifique el contenido del documento.

Casos	Resultado esperado	Logró
El archivo existe, está bien formado y completo	Se debe mandar un mensaje de éxito	Si
	Si se comprueba manualmente, la información debe estar correctamente guardada en la base de datos	Si
El archivo no está bien formado	Se debe mandar un mensaje de documento mal formado y no se debe guardar nada en la base de datos	Si
	Se debe mandar un mensaje indicando específicamente el error	Si

Tabla 6.2 Tabla de pruebas a la base de datos en el contexto de la Dra. Candy

<sup>1</sup> Los esquemas aceptados son los de GISonline [Cepeda, 2003], WFS [López, 2004] y Alemanes (Apéndice A)



## 6.2.2 Pruebas para el proceso de Recuperar

### 6.2.2.1 Pruebas a la interfaz

En esta sección se muestran las pruebas hechas a la interfaz de Recuperar en el contexto del ejemplo de Anthony. Esta interfaz recibe como datos de entrada el *query* y la decisión de incluir o no las geometrías en las respuestas a la consulta. Se puede hacer cualquier tipo de *query* sobre los atributos no espaciales de tramo y nodo. La lista de estos atributos se muestra en los botones que definen las consultas básicas a cada uno de ellos, el usuario puede de ahí modificarlos para hacerlos más específicos. La selección de las geometrías es mediante cajas de selección, hay uno para cada opción y sólo se necesita darles clic y una palomita confirmando su selección aparecerá en el cuadrado. Anthony utilizará el botón de *Stretch Name* y le agregará la condición “*where speedLimit>30*” a la consulta. Ya que sólo desea saber los nombres, seleccionará la opción de sin geometrías y el nombre “BuenaV-Cal\_30” .

Opciones	Casos	Resultado esperado	Logró
<i>Stretch Name</i>		Debe aparecer el <i>query</i> general solicitando el atributo " <i>stretchName</i> "	Si
Sin/Con Geometrías	Se selecciona a lo más una opción	Debe mandarse un indicador al sistema de la opción seleccionada	Si
Procesar	Se selecciona un <i>query</i> definido y se modifica	Debe mandar el <i>query</i> al sistema	Si
	Se tiene un <i>query</i> incorrecto	Debe mandar un mensaje de consulta no válida Debe mostrar el mensaje de error directamente de <i>java.sql.SQLException</i>	No Si
Nombre Archivo	No se teclea un archivo	Debe mandar un mensaje de nombre de Archivo no seleccionado	Si
		Debe tomarse el nombre del Archivo <i>default</i>	No
	Se teclea un nombre de archivo válido	Debe mandar al sistema el nombre del Archivo	Si
Salir		Se debe cerrar la ventana de Guardar	Si
X		Se debe cerrar el sistema completo	Si

Tabla 6.3 Tabla de pruebas hechas a la interfaz de Recuperar en el contexto de Anthony

### 6.2.2.2 Pruebas al nivel de la base de datos

En esta sección se muestran las pruebas hechas sobre la consulta de Anthony. En el caso de error se revisaron sólo los casos de un pequeño error en el formato del *query*, dado que es un usuario experto. Los resultados esperados sólo son dos, que el archivo sea creado con la información correcta o que no sea creado.

Casos	Resultado esperado	Logró
El query sólo pide el " <i>stretchName</i> " con/sin geometrías	Se debe crear un archivo con la información correcta	Si
El <i>query</i> no está bien formado	Se debe mandar un mensaje de error y no se debe crear el archivo	Si
El query tiene " <i>where</i> " y lo que sigue devuelve más de una <i>tupla</i>	Se debe mandar un mensaje de error y no se debe crear el archivo	Si
El <i>query</i> tiene " <i>where</i> " y lo que sigue devuelve una <i>tupla</i>	Se debe crear un archivo con la información correcta	Si
El <i>query</i> está bien formado	Se debe crear un archivo con la información correcta	Si
El query está bien formado pero no regresa nada	Se debe mandar un mensaje de no respuesta	Si
	No se debe crear el archivo	No
	Se debe crear un archivo "vacío" , solo con los <i>namespaces</i>	Si

Tabla 6.4 Tabla de pruebas hechas para consultas en el contexto de Anthony

Las pruebas hechas al sistema mostraron un correcto desempeño del mismo, ya que la información fue almacenada y recuperada de manera correcta. Los casos mostrados en las tablas son representativos de las situaciones reales que podrían ser generadas por un usuario común. Lo anterior prueba que el sistema cumple con los objetivos que se definieron en el análisis para OGGDB. En el siguiente capítulo se muestran las conclusiones y el trabajo a futuro que se puede realizar sobre el proyecto OGGDB.