4. MANEJO DE DATOS

4.1 INTRODUCCIÓN

Ya que se ha entendido la estructura y la distribución de los datos en la base de datos es pertinente mencionar los diferentes procedimientos que han sido creados para obtener la óptima manipulación y tratamiento de los datos.

Como se observó en el capítulo anterior, existe una gran diversidad de tipos de datos que pueden ser utilizados para diferentes fines. Tenemos datos descriptivos y datos geográficos, que son los que se utilizaron para efectos de esta investigación. También podemos citar los datos estadísticos que en el caso de este proyecto han sido excluidos, pues por la importancia de dichos datos amerita dedicar un estudio completo y exclusivo sobre esa ramificación.

El proceso de manejo de datos está compuesto por dos diferentes actividades. La primera parte es la obtención y almacenamiento de los datos, y por otra parte el tratamiento de los datos para generar las consultas.

En el primer punto serán abordados temas como la conexión a la base de datos, la creación de los diferentes objetos que se utilizan, los diagramas de clase para poder entender las diferentes jerarquías existentes, así como la descripción detallada de los métodos que se crearon para poder tener acceso a los diferentes atributos de las clases.

En el segundo apartado se describen los diferentes tipos de tratamientos que van de la mano con los datos, los procedimientos necesarios para llevar a cabo las múltiples consultas y la obtención de los resultados.

4.2 OBTENCIÓN DE DATOS Y CREACIÓN DE CLASES

4.2.1 Conexión a la base de datos

Es evidente que para poder obtener los datos, y generar los objetos necesarios se tiene que generar una conexión remota a la base de datos. Para lograr este objetivo, se utiliza JDBC.

4.2.2 Creación de las clases

Como ya se mencionó antes, hubo la necesidad de acotar los datos que se iban a manejar en este proyecto. Para obtener mejores resultados sólo se seleccionaron las siguientes entidades del modelo Entidad-Relación:

a) Estado

b) Municipio

c) Localidad

e) Puntos_Localidad

f) Carreteras

g) Puntos_Carretera

h) Ruta_Evacuacion

i) Puntos_Ruta

j) Refugio

k) Puntos_Refugio

4.2.2.1 Clases Implementadas

Cada entidad tiene su respectiva representación en una clase. Se toman en cuenta todos los atributos de cada entidad, como atributos de cada clase. Por ejemplo, la clase Localidad tiene como atributos su nombre, su clave de localidad, su clave del municipio al que pertenece, el área que ocupa, el perímetro, latitud norte, longitud oeste, entre otros. También tiene todos sus métodos para asignar valores y obtener los valores de los atributos. Cabe mencionar que cada clase tiene programado un algoritmo que automatiza la obtención de los atributos. El proceso por el que pasa dicho algoritmo es el siguiente:

- Creación del nuevo objeto con todos sus atributos inicializados con valores nulos

- Llamado del método obtenNuevo***Objeto***DB, refiriéndome con *Objeto* al nombre de la clase en cuestión.

- Se genera la consulta a la base de datos obteniendo todos sus atributos descriptivos

- Se genera la consulta a la base de datos obteniendo todos sus atributos geográficos de la cartografía 1:20,000 que se utilizó para este estudio. En algunas clases este

proceso de obtención de la representación geográfica, es inteligente, pues decide si su representación es una geometría o una colección de geometrías. Por ejemplo cuando una carretera está formada por múltiples polilíneas, se tiene que formar una colección de éstas.

Ahora se muestra en la figura 4.1 el diagrama general de las clases tomando en cuenta las relaciones que existen entre ellas

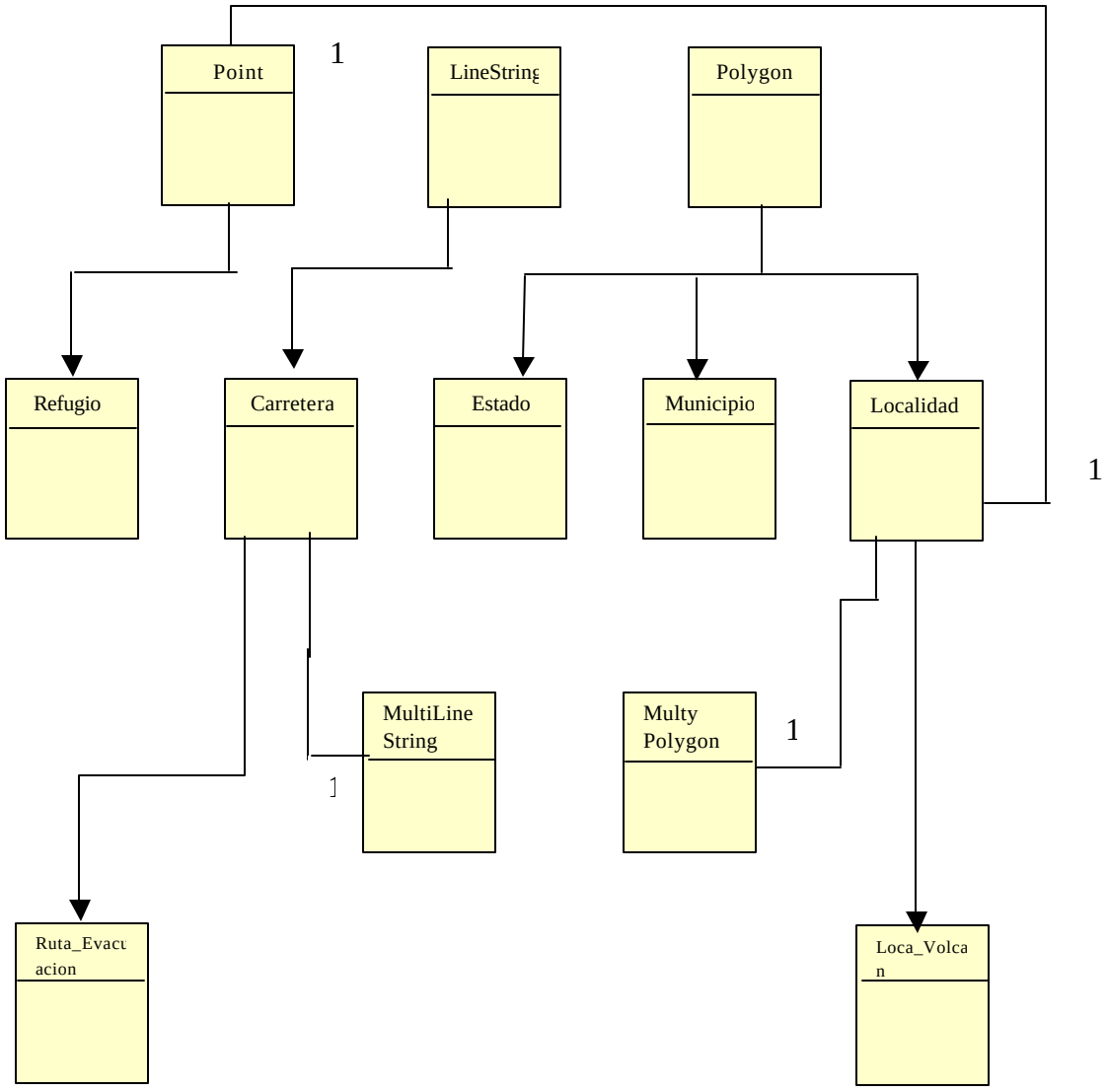

Fig. 4.1 Diagrama de relación de clases

4.2.3 Descripción detallada de las clases y Mapeo con la base de datos

En la figura 4.2 se muestra un ejemplo del mapeo entre la representación geográfica de nuestras clases, y la estructura de la base de datos descrita en el capítulo anterior.

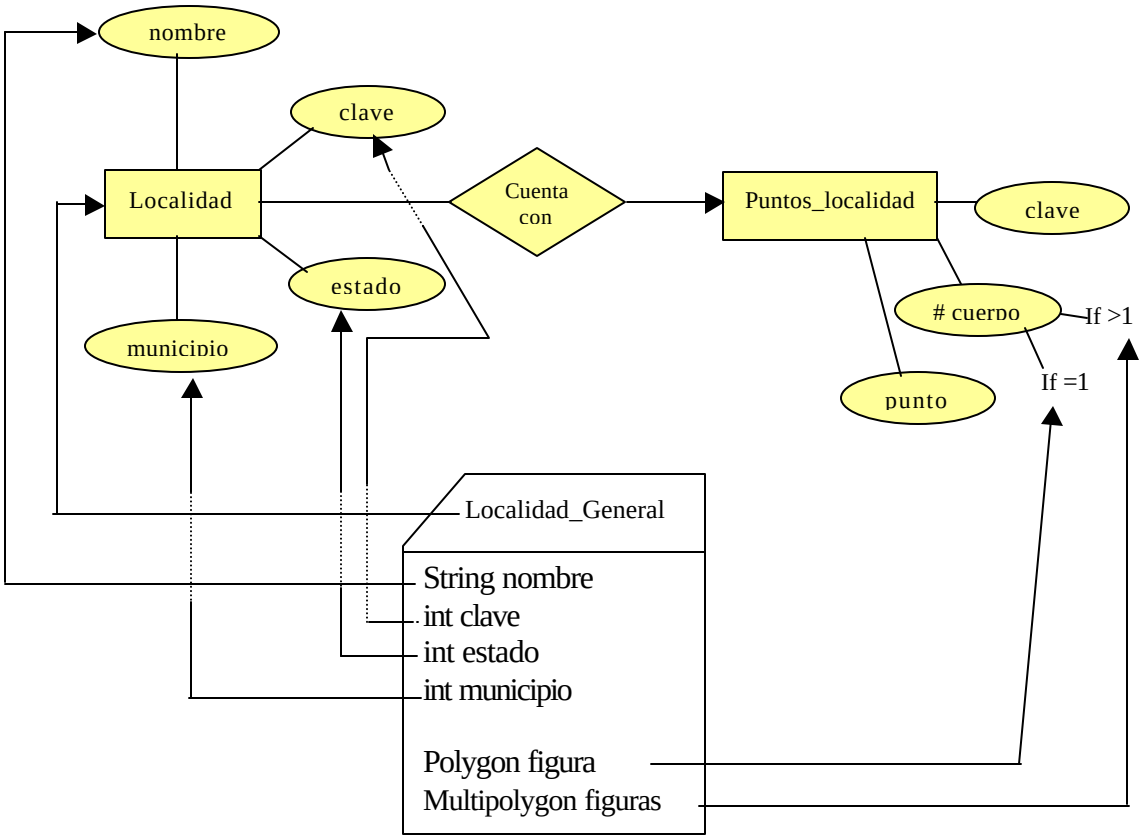

Fig. 4.2 Mapeo de una clase con la base de datos

4.2.3.1 Estado.- Esta clase obtiene de manera automática los datos descriptivos al igual que su representación geográfica utilizando la especificación OpenGis por medio de un método llamado obtenNuevoEstadoDB. Dicho método recibe como parámetro un String que es el nombre del estado y por medio de este, se genera un query definido para obtener

todos los datos que son tomados en cuenta como atributos de la clase. Ya que han sido obtenidos todos los atributos descriptivos, se lanza un método llamado obtenRepresentacionGeometrica que es el encargado de generar su figura espacial.

Tipo de Geometría utilizada: Polígono.

En la siguiente figura se muestra el mapeo con la base de datos.

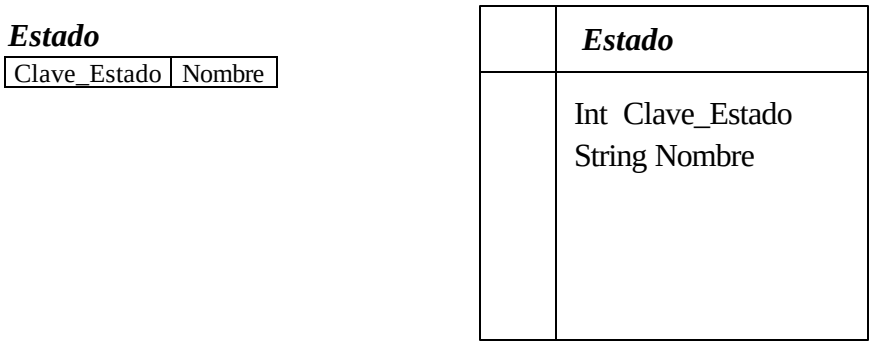

Fig. 4.3 Mapeo entre la entidad Estado y su clase

4.2.3.2 Municipio.- Esta clase obtiene de manera automática los datos descriptivos al igual que su representación geográfica utilizando la especificación OpenGis por medio de un método llamado obtenNuevoMunicipioDB. Dicho método recibe como parámetro un String que es el nombre del municipio y por medio de este, se genera un query definido para obtener todos los datos que son tomados en cuenta como atributos de la clase. Ya que han sido obtenidos todos los atributos descriptivos, se lanza un método llamado obtenRepresentacionGeometrica que es el encargado de generar su representación espacial.

Tipo de Geometría utilizada: Polígono.

En la siguiente figura se muestra el mapeo con la base de datos.

Municipio

Clave_Estado	Clave_Municipio	Nombre	Cabecera	Lat_Norte	Long_Oeste	Clave_Distrito	Altitud

	Municipio
	Int Clave_Estado Int Clave_Municipio String Nombre . . .

Fig. 4.4 Mapeo entre la entidad Municipio y su clase

4.2.3.3 Localidad.- Esta clase obtiene de manera automática los datos descriptivos al igual que su representación geográfica utilizando la especificación OpenGis por medio de un método llamado obtenNuevaLocalidadDB. Dicho método recibe como parámetro un String que es el nombre de la localidad y por medio de este, se genera un query definido para obtener todos los datos que son tomados en cuenta como atributos de la clase. Ya que han sido obtenidos todos los atributos descriptivos, se lanza un método llamado obtenRepresentacionGeometrica que es el encargado de generar su figura espacial.

Tipo de Geometría utilizada: Punto, Polígono, Multipolígono.

Las razones por las que tiene múltiples representaciones son:

Punto.- Toda localidad puede ser referenciada por un punto dependiendo la necesidad que surja. Por ejemplo. Si queremos obtener la distancia entre dos localidades cualesquiera que sean, debemos de tener un punto de referencia, el cual es calculado en el momento que se genera el objeto (centroide), pero también puede ser preestablecido como puede ser un centro de reunión de la población en caso de emergencia o el mismo zócalo de la localidad. Cabe aclarar que no es la representación de mayor trascendencia para el caso de las localidades.

Polígono.- Esta es la representación con mayor importancia para una localidad, puesto que cualquier localidad sin excepción tiene una representación poligonal. Esto significa, que la localidad puede ser representada por el contorno formado por el trazo de la misma localidad.

La clase Localidad como tal extiende a la clase Polygon, por las razones previamente argumentadas y a consecuencia se pueden utilizar los diferentes métodos establecidos en la Super clase como si fueran métodos nativos de la clase en cuestión.

Multipolígono.- Existe el caso en que la representación geográfica de una localidad, puede estar formada por 2 o más polígonos. Pensemos en una localidad la cual tenga una barranca la cual la divida en dos sectores. Si no se manejara esta división con figuras diferentes, seguramente el trazo del contorno sería erróneo. Es por esto que se maneja como posible Multipolígono. Para generar los polígonos independientes, existe un atributo

en la base de datos que es el número de cuerpo. De tal manera, todos los puntos que tengan un mismo número de cuerpo, se toman como parte de un polígono independiente. Si no se contara con este número de cuerpo, no habría forma de definir los polígonos. En la figura 4.5, se muestra la diferencia entre representaciones multipoligonales con número de cuerpo y sin número de cuerpo.

Fig. 4.5 Diferencia en multipolígonos utilizando un número identificador de cuerpo.

Cabe señalar que en esta clase, al momento de obtener la representación geográfica (Método obtenRepresentacionGeografica)se hace un análisis automático e inteligente para saber si es uno o más polígonos y el mismo algoritmo decide por que tipo de representación geométrica optar.

En la figura 4.6 se muestra el mapeo con la base de datos.

Localidad_General

Clave_Municipio	Clave_Localidad	Nombre	Area	Perimetro	Lat_Norte	Long_Oeste

PX	PY	Dist_Crater	Altitud

Puntos_Localidad

Clave_Localidad	No_Punto	X	Y	No_Cuerpo

	Localidad
	Int Clave_Municipio Int Clave_Estado String Nombre . . . Multipolygon Contornos

Fig. 4.6 Mapeo entre la entidad Localidad y su clase

4.2.3.4 Loca_Volcan.- Esta clase extiende de localidad lo que significa que tiene los mismos atributos, más los específicos de información descriptiva de las localidades situadas en zonas de riesgo de erupción del volcán Popocatépetl. Se generaron las dos clases (Localidad y Loca_Volcan) de manera independiente, pensando en estudios futuros que no tengan que ver con el volcán Popocatépetl y que sólo se utilicen localidades.

También cuenta con un método que se llama obtenNuevaLocaVolcanDB, el cual obtiene todos sus atributos y entre ellos los atributos de la localidad. En este caso, la clase Loca_Volcan no tiene un método para obtener su representación geométrica, puesto que se

utiliza el método que ya se tiene en la clase Localidad.

En la figura 4.7 se muestra el mapeo con la base de datos.

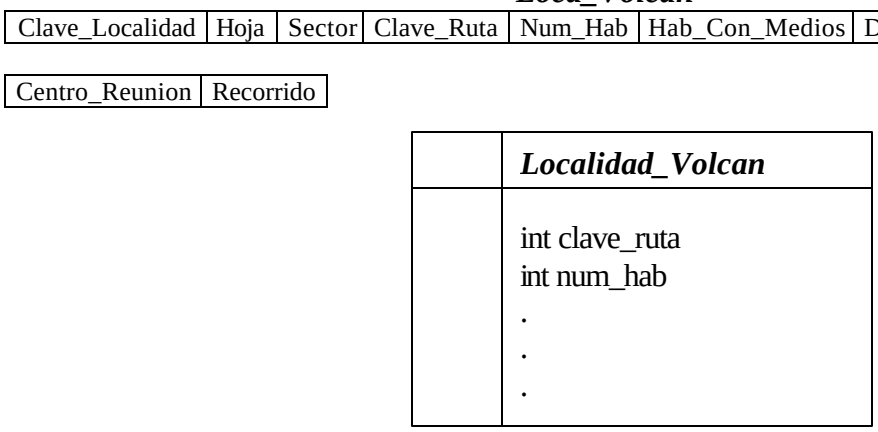

Fig. 4.7 Mapeo entre la entidad Localidad_Volcan y su clase

4.2.3.5 Carreteras.- Esta clase obtiene de manera automática los datos descriptivos al igual que su representación geográfica utilizando la especificación OpenGis por medio de un método llamado obtenNuevaCarreteraDB. Dicho método recibe como parámetro un String que es el nombre de la carretera y por medio de este, se genera un query definido para obtener todos los datos que son tomados en cuenta como atributos de la clase. Ya que han sido obtenidos todos los atributos descriptivos, se lanza un método llamado obtenRepresentacionGeometrica que es el encargado de generar su representación espacial.

Tipo de Geometría utilizada: LineString, MultiLineString.

Las razones por las que tiene múltiples representaciones son:

LineString.- Esta es la representación con mayor importancia para una carretera, puesto que cualquier carretera sin excepción tiene una representación que conste de 1 o más

líneas. Esto significa, que la carretera siempre y en el peor de los casos es representada por el trazo de su misma trayectoria.

La clase Carreteras como tal extiende a la clase LineString, por las razones previamente argumentadas y a consecuencia se pueden utilizar los diferentes métodos establecidos en la Super clase como si fueran métodos nativos de la clase en cuestión.

MultiLineString.- Existe el caso en que la representación espacial de una carretera, puede estar formada por 2 o más LineStrings. Pensemos en una carretera la cual cruce una ciudad, y dentro de la ciudad, ésta carretera se convierte en avenida principal. Si no se manejara esta división con curvas diferentes, seguramente el trazo de la curva sería erróneo. Es por esto que se maneja como posible MultiLineString. De igual manera que las localidades y su representación multipoligonal, dentro de los datos espaciales de la carretera existe uno y sólo un número identificador relacionado a cada punto. Esto para evitar que la representación sea desfasada o errónea.

En la figura 4.8 se hace referencia a la diferencia que puede existir al manejar un numero de cuerpo al formar la representación geométrica de las carreteras.

Sin identificación de número de curvas

Con identificación de número de curvas

Fig. 4.8 Diferencia en multipolilíneas utilizando un número identificador de cuerpo fica

se hace un análisis automático e inteligente para saber si es uno o más polígonos y decide por que tipo de representación geométrica optar.

En la figura 4.9 se muestra el mapeo con la base de datos.

Carretera

Clave_Carretera	Nombre	Estado	Tipo	Amplitud	Longitud

Puntos_Carretera

Clave_Carretera	No_Punto	X	Y	No_Cuerpo

	Carretera
	Int Clave_Carretera String nombre . . . . MultiLineString contorno

Fig. 4.9 Mapeo entre la entidad Carretera y su clase

4.2.3.6 Ruta Evacuación.- Esta clase extiende de Carreteras lo que significa que tiene los mismos atributos, más los específicos de información descriptiva de las rutas de evacuación establecidas y utilizadas por Plan Operativo Popocatépetl.

En la figura 4.10 se muestra el mapeo con la base de datos.

Ruta_Evacuación

Clave_Ruta	R	G	B	Clave_Destino	Tipo

Puntos_Ruta

Clave_Ruta	No_Punto	X	Y	No_Cuerpo

	Ruta_Evacuacion
	int clave_ruta int R int G . .

Fig. 4.10 Mapeo entre la entidad Ruta_Evacuacion y su clase

En resumen se nota que para efectos de eficiencia, es de suma importancia generar objetos fundamentales para el manejo de nuestros datos, pues dado que el volumen de éstos es gigantesco, no se pueden manejar directamente de la base de datos.

La ventaja de generar un mapeo total de la base de datos en clases, es en primer punto, la facilidad para conceptuar el esquema de la base de datos. También la facilidad de generar los objetos de forma automática ahorra mucho trabajo al desarrollador, pues el interés general es la manipulación de datos. Resultado de esto será apreciado por desarrolladores futuros que no tengan la necesidad de preocuparse en obtener toda la información de la base de datos.

4.3 TRATAMIENTO DE LOS DATOS Y CONSULTAS

Ya teniendo las clases formuladas sólo surge la necesidad de manipular los datos para obtener los resultados deseados. Partiendo de que los métodos de las clases están diseñados para que dinámicamente se llenen los campos de los atributos de las mismas, el tratamiento se reduce en grado de complejidad.

4.3.1 Tratamiento de los datos

El procedimiento general de tratamiento de datos, es generar la consulta deseada, y con el resultado de dicha consulta, crear un objeto del tipo que se necesite, para que posteriormente, se puedan mostrar los resultados de la consulta. En pocas palabras el tratamiento sobre los datos se reduce en las diferentes consultas que se pueden realizar gracias a la distribución de los datos en la base de datos.

Por ejemplo, el usuario desea obtener la o las carreteras que intercepten la carretera federal a Cholula, primero se tiene que generar el objeto de tipo carretera correspondiente a la carretera federal a Cholula, incluyendo su representación espacial. Posteriormente se hace un análisis en base a diferentes queries de todas las representaciones espaciales de las carreteras, verificando cuál o cuáles carreteras tienen un punto en común, que en este caso seria el punto donde se cruzan. Ya que se obtiene el punto, se genera el nuevo objeto de tipo carretera con los datos de la carretera que cruzó por la federal a Cholula. Esto se hace tantas veces como carreteras que intersecten la federal a Cholula.

4.3.2 Consultas

En el módulo de consultas, que es el módulo principal de nuestra aplicación, se pueden generar todos los cuestionamientos deseados a la base de datos, siempre y cuando se tengan los datos en ella.

También es pertinente aclarar que para todo tipo de cuestionamiento que se quiera hacer a la base de datos, tiene que existir una relación entre las tablas que se encuentran en ella. Lógicamente van a surgir consultas que serán más sencillas que otras, y también existirán consultas en las que se tenga que hacer tratamientos complejos para obtener resultados satisfactorios.

Como ya se había mencionado con anterioridad, las consultas pueden ser de tres tipos, pero de dos fundamentales:

- Consultas sobre datos descriptivos
- Consultas sobre datos espaciales
- Consultas compuestas de datos descriptivos y datos espaciales

Las consultas sobre datos descriptivos, es directamente generar la relación entre las tablas de la base de datos y obtener el resultado de manera directa. Aquí no se genera ningún tipo de calculo, y si hay lugar para alguno, son operaciones triviales entre cantidades.

Por ejemplo para saber el número de habitantes de una localidad que el gobierno del estado tiene que evacuar, solo se tiene que obtener el número total de habitantes y el número de habitantes que pueden evacuar por medios propios. Así se obtiene el número de habitantes sin medios para evacuar y que son responsabilidad de las autoridades.

En las consultas sobre datos espaciales, se introducen algoritmos más complejos para el cálculo de operaciones entre puntos y geometrías. Cabe señalar que el software que se está reutilizando es un repositorio de consultas espaciales generado por [Loranca, 2000] donde incluye distintas opciones para realizar operaciones entre objetos de especificación OpenGis. Se incluyen métodos como Touches, Contain, Crosses, Intersects, entre otros.

Ahora se muestran los diferentes métodos que se incluyen en la clase Geometry incluidas en [Loranca, 2000]:

- Intersects(Geometry g): Regresa un entero 1 si la relación "intersección" es verdadera, 0 para falso y –1 en caso de que sea null. Este método se utiliza por ejemplo cuando quiero saber si una carretera es entrante o saliente de una localidad. Se utiliza este porque en algunos casos, aparentemente algún segmento de la carretera termina exactamente el borde del polígono de la localidad, pero con un zoom excesivo, podemos darnos cuenta que el último punto de ese segmento de carretera está cruzando el borde de la localidad.
- Touches(Geometry g): Regresa un entero 1 si la relación "Toca a" es verdadera, 0 para falso y –1 en caso de que sea null. Este método lo utilizamos cuando se presenta el caso contrario del ejemplo anterior.

Significa que se utiliza cuando el último punto de la carretera, si se encuentra exactamente en el borde de la localidad.

- Crosses(Geometry g): Regresa un entero 1 si la relación "Cruza a" es verdadera, 0 para falso y –1 en caso de que sea null. Este método lo utilizamos cuando queremos saber que las carreteras que cruzan una localidad. Esto quiere decir que se utiliza para saber cuando los segmentos de una carretera, cruzan el polígono formado por los puntos del borde de una localidad de extremo a extremo.
- Contains(Geometry g): Regresa un entero 1 si la relación "Contiene a" es verdadera, 0 para falso y –1 en caso de que sea null. Este método lo utilizamos por ejemplo para saber si un refugio se encuentra dentro del polígono formado por los puntos de una localidad.

En las consultas combinadas, se pueden obtener resultados en base a consultas sobre datos descriptivos y datos espaciales. Un ejemplo interesante es el de la evacuación global. Se tiene que calcular el número de habitantes sin medios, o sea responsabilidad de las autoridades, de todas las poblaciones ubicadas en un cierto radio del cráter del volcán Popocatépetl.

4.3.3 Ejemplos de consultas

Antes de citar los ejemplos de las consultas elaboradas, demos un vistazo a la arquitectura general de nuestro sistema

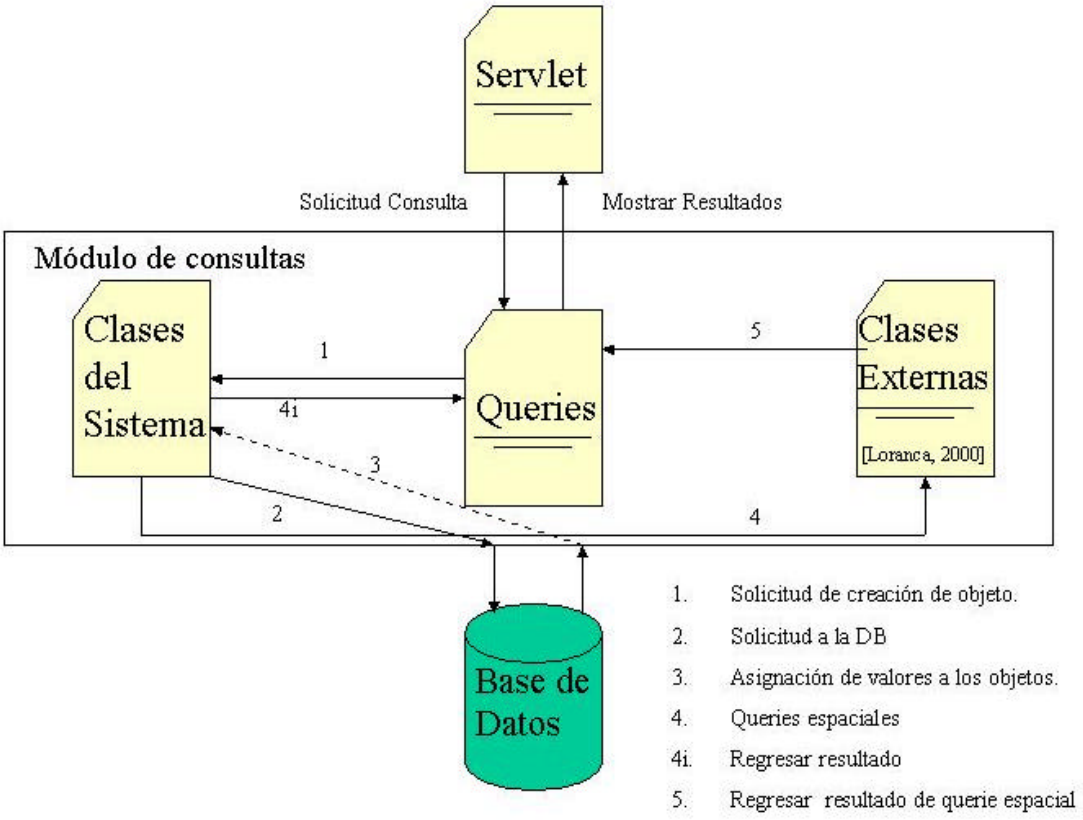

Fig. 4.10 Arquitectura general de la aplicación.

Nuestro sistema esta compuesto por 3 partes. La base de datos descrita en el capítulo anterior, el módulo de consultas y manejo de datos y la parte de visualización que se utiliza un Servlet. En la figura se pueden observar los pasos a seguir del funcionamiento de nuestra arquitectura la cual será descrita a detalle en el siguiente capítulo.

DistanciaCrater.- Esta consulta regresa la distancia que existe entre una población y el cráter del volcán Popocatépetl. El único dato que es necesario, es el nombre de la población. Para esta consulta se tiene el dato descriptivo de distancia_crater, pero podría ser calculado si se señala un punto constante en la cartografía 1:20,000 que se tiene en la UDLAP, y se traza entre el punto del cráter y el punto centroide de la localidad deseada.

LocalidadCercanaCrater.- Esta consulta regresa uno o más nombres de las localidades que se encuentren a x distancia del cráter del volcán. Lo necesario es dar la distancia deseada, y como resultado se obtiene un vector de cadenas donde se incluyen los nombres de todas las localidades.

HabitantesLocalidad.- Esta consulta regresa el número de habitantes total de una localidad, en base al nombre de ésta. Utilizando esta consulta, también se podrían obtener el número total de habitantes de un municipio.

HabitantesMedios.- Esta consulta regresa el número de habitantes de una localidad con medios propios para evacuar. Dicho resultado se obtiene con el nombre de la localidad.

HabitantesEvacuar.- Esta consulta calcula cuantos habitantes sin medios para evacuar son parte de la comunidad de una localidad. Para obtener dicho resultado, solo utilizamos las dos consultas anteriores, y se hace la siguiente operación matemática. HabitantesLocalidad – HabitantesMedios = Habitantes que tienen que evacuar las autoridades.

DistanciaLocalidades.- Esta consulta obtiene la distancia entre dos localidades, o entre dos

puntos cualesquiera que sean. Se calcula la distancia con la fórmula generada del Teorema de Pitágoras. Cabe señalar que no se está haciendo uso de ninguna proyección. La distancia obtenida es un estimado en base a la línea recta que existe entre los dos puntos. No se toma en cuenta la elevación de los puntos. Esta consulta puede ser mejorada tomando en cuenta las opciones mencionadas anteriormente.

LocalidadMunicipio.- Esta consulta nos regresa un listado de todas las comunidades que forman parte de un municipio. Solo se necesita el nombre del municipio para obtener el resultado deseado.

ZonaRiesgo.- Esta consulta nos regresa el indicador de la zona de riesgo en la que se encuentra una localidad. Sólo se necesita el nombre de la localidad. Ya que se tiene el indicador, sólo es necesario checar en una tabla con el indicador, en que zona se encuentra dicha localidad.

RutaRecorrido.- Esta consulta nos regresa el recorrido de la ruta de evacuación desde una localidad hasta su destino.- Sólo se necesita saber el nombre de la comunidad que se quiere relacionar con su ruta.

RefugioPertenecienteAComunidad.- Esta consulta hace un análisis espacial utilizando la representación geométrica de una localidad y la representación puntual de un refugio. Al dar el nombre del refugio y el nombre de la comunidad, regresara un valor booleano el cual indicará si el refugio se encuentra dentro de la localidad. En el caso de que la localidad tenga una representación multipoligonal, se hace el mismo análisis tantas veces como

numero de polígonos tenga la localidad, ya que lo único que se tiene que hacer es descomponer el objeto multipolígono en todos los polígonos que lo componen.

En el apéndice I se encuentra un pequeño tutorial para poder generar consultas, esto pensándose en estudios futuros o consultas que se puedan agregar a nuestra aplicación.