5. IMPLEMENTACIÓN E INTEGRACIÓN DEL SISTEMA

Ya que se ha hecho una introducción de las clases que están involucradas con nuestro sistema, es necesario a hablar de la aplicación ya como producto compuesto por clases e interfaz para usuario.

Ahora, una aplicación que resuelve múltiples cálculos complejos, o que satisface con las necesidades, no es suficiente. La tecnología permite a los desarrolladores de software generar aplicaciones que a parte de cubrir las expectativas, sean elaboradas lo mejor posibles. Con esto me refiero a que también, uno como ingeniero de software, se tiene que preocupar por aspectos como la forma en que el usuario va a visualizar su aplicación, el tiempo de respuesta que predomina, la facilidad de manejo y el control de errores del usuario. En este capitulo, a parte de ilustrar la arquitectura general de la aplicación se mencionaran las diferentes técnicas que fueron empleadas para liberar una aplicación que cumpliera con todos los requisitos de manera satisfactoria.

5.1 UBICACIÓN DE LA APLICACIÓN DENTRO DEL CONTEXTO GENERAL DE LOS ESTUDIOS ANTERIORES

La aplicación que se modela en esta tesis, es uno de varios estudios que se han generado en relación con un solo proyecto. Como se mencionó a un principio de este documento, existen trabajos realizados como el descrito en [Loyo, 2000] y [Vera, 2000]. Cuando inició dicho proyecto se contaba sólo con una cartografía digitalizada con escala 1:250,000 que aun se conserva. Posteriormente se adquirió la de escala 1:20,000 que es con

la que ahora se trabaja por razón obvia de detalle. Se ha venido manipulando dicha cartografía a conveniencia del rumbo del proyecto. Sobre ella, se han generado nuevas capas como son las rutas de evacuación, y también se le han asociado datos descriptivos como son los nombres de las localidades aledañas al volcán Popocatépetl.

Para ubicar esta aplicación en el orden de aparición de lo que hoy día se tiene realizado, es necesario exponer el siguiente diagrama:

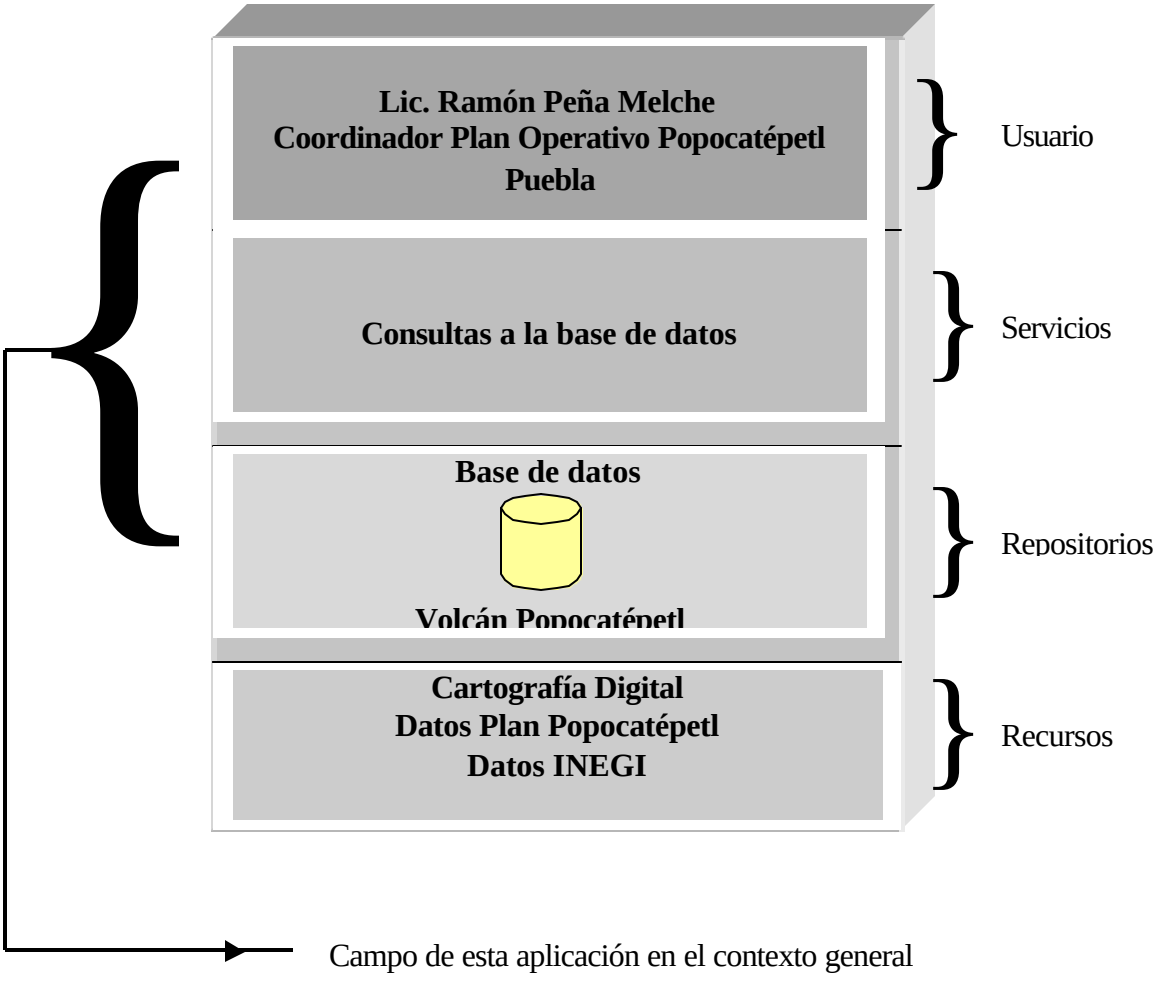

Fig. 5.1 Ubicación de la aplicación en el contexto general

Como se aprecia en la figura 5.1, en la parte inferior se encuentran los recursos o los datos fuente con los que hemos trabajado. Aquí entra toda la información que se tienen, pero que no sirve si no es sometida a un tratamiento previo. Mapas impresos, cartografía digitalizada, datos recolectados entre otros.

En el segundo nivel identificado como repositorios se encuentra la base de datos. Como se describió en el capítulo III el esquema de la base de datos tiene un enfoque general. Puede ser explotada de múltiples formas, pero en este caso particular es directamente con datos relacionados al volcán Popocatépetl.

En el tercer nivel que son los servicios, están ubicados los queries y el objetivo de esta aplicación. Obviamente dichas consultas son enfocadas a los datos del volcán Popocatépetl.

En el cuarto nivel se encuentran los usuarios. Aquí ubicamos al Licenciado Ramón Peña Melche, el cual es nuestro usuario final y a la vez fungió como colaborador, aportador de datos y como retroalimentador del proyecto. Él seguirá situado en esta posición para algunos desarrollos futuros, como pueden ser la visualización de las consultas en 2 dimensiones así como en tres dimensiones.

5.2 ARQUITECTURA DEL SISTEMA

Antes de mencionar el “Cómo” es necesario describir el “Qué”. Nuestra aplicación está compuesta por dos partes o módulos. El primer módulo es aquel que hace que todos los

procedimientos sean transparentes para el usuario. Con esto me refiero al módulo que está en contacto directo con el usuario final. El segundo módulo es aquel en el que se realizan todos y cada uno de los cálculos generados por las consultas. Llamemos a ambos GUI y QUERIES respectivamente, para identificarlos fácilmente.

En el siguiente esquema, se muestra la arquitectura del sistema, y el procedimiento que se sigue para poder hacer una transacción o consulta.

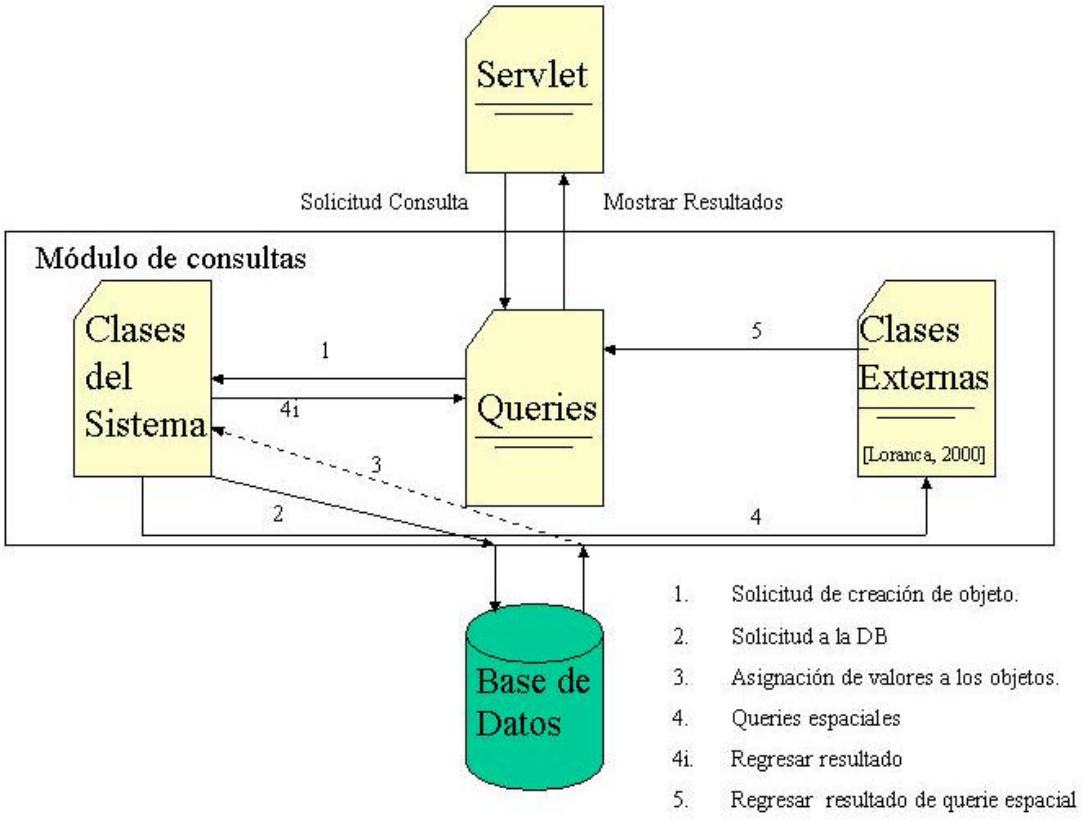

Fig. 5.2 Arquitectura del Sistema

Es notorio que existe un paso opcional (4), pues no en todo momento se hace uso de los métodos de las clases externas [Loranca, 2000]. Solo se utilizan cuando se generan

consultas espaciales y no en todos los casos, y si notamos, los resultados se regresan (5) cuando se hace uso de consultas espaciales [Loranca, 2000] y (4i) cuando se hace uso de las clases propias de este sistema.

Como se mencionó antes el sistema esta dividido en dos partes: GUI y QUERIES. GUI es prácticamente el Servlet. El Servlet toma un rol demasiado importante en nuestra aplicación, pues es el medio de comunicación entre el usuario y las aplicaciones que están atrás de todo el funcionamiento de nuestro sistema y los resultados de las consultas que quiere. Pensemos un Servlet como una página web en la cual podemos dar datos, y nos regresará la información que queremos, pero en su interior no es así de simple.

5.2.1 Servlets

Los Servlets son módulos de código Java que funcionan en una aplicación en el servidor para responder solicitudes del cliente. La diferencia entre los Servlets y los Applets es que los Applets corren en el cliente. Los Servlets son comúnmente usados en http para lograr independencia de plataforma. Por lo tanto Servlet por lo regular es relacionado con http Servlet.

El uso que se le da a los http Servlets es de:

- Proceso y/o almacenamiento de datos suministrado desde una forma HTML.
- Generando contenido dinámico. Por ejemplo: Regresando el resultado de una consulta a una base de datos.

- Manejando el estado de la información. Por ejemplo: Para un sistema de compras en línea el cual recibe varios clientes concurrentes, se debe de mapear cualquier petición al cliente correcto.

5.3 EXPLICACIÓN Y JUSTIFICACIÓN DE LA ARQUITECTURA UTILIZADA

Ya que se ha explicado lo que es un Servlet, podemos notar la importancia que este tiene dentro de nuestra aplicación. Con ellos podemos generar páginas web dinámicamente incluyendo los datos que nos regresa una consulta. Por ejemplo; acabamos de seleccionar la consulta que nos obtiene los nombres de las poblaciones que se encuentran en el municipio de Atlixco. El resultado de esta consulta es un vector, y para mostrar este reporte vía web, antes sólo lo podíamos hacer con un Applet, pero ahora con las ventajas de los Servlets es posible con el siguiente conjunto de instrucciones:

```
<%
for (int i = 0;  i < vectorLocalidades.size();  i++)
{
        out.println("Localidad "+ i + "   "+ vectorLocalidades.elementAt( i ));
}
%>
```

Lo único que se hace es mandar llamar un ciclo for y generar una instrucción equivalente al System.out.println en Java, y el Servlet lo traducirá en código html.

El resultado de esto lo podemos ver en la figura 5.3.

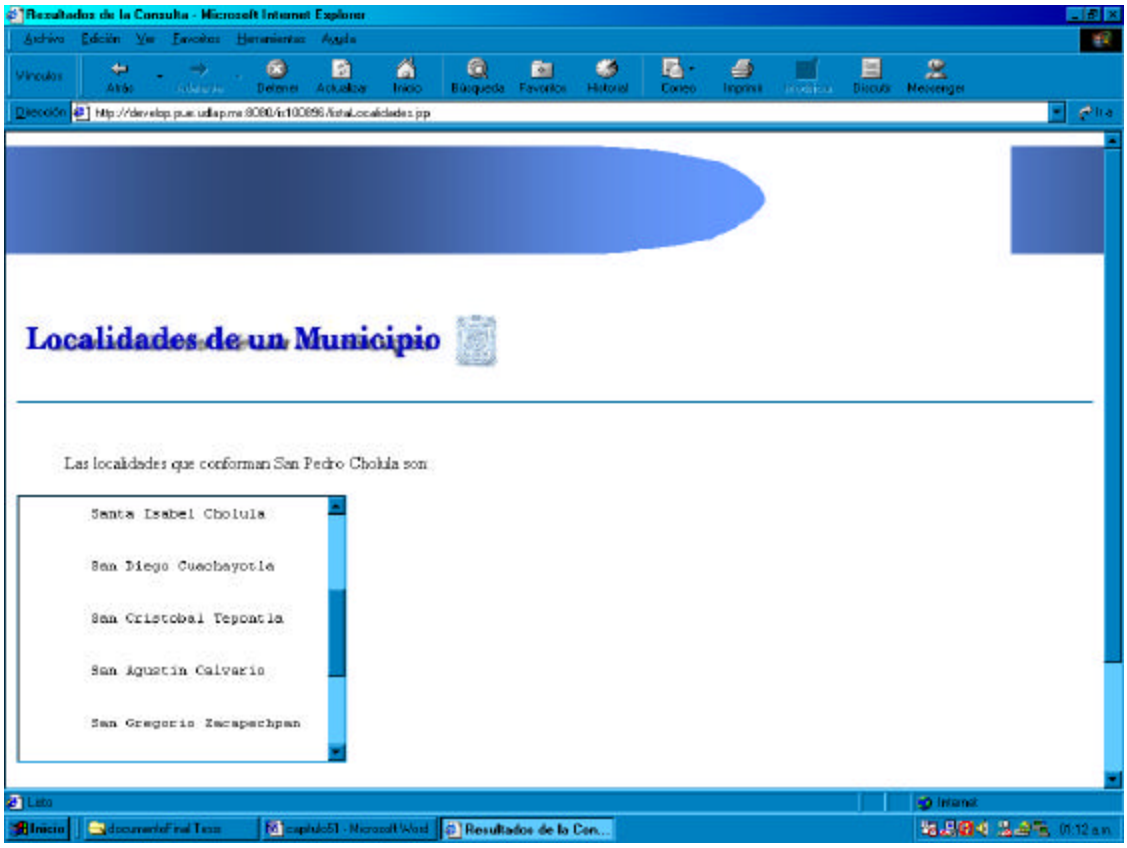

Fig. 5.3 Resultado de la consulta

El código que se mostró anteriormente, lo único que hace es imprimir en el área de texto, y mostrar las diferentes localidades que forman un municipio. Es notorio que la utilización de código java combinado con html es sencilla y eficiente.

Una característica importante de los Servlets es la velocidad de respuesta. Esto tiene que ver con que el Servlet o la aplicación relacionada al Servlet, corre directamente en el servidor donde se encuentra almacenado el código de las clases. Antes con los Applets, se perdía mucho tiempo en lo que la JVM (Java Virtual Machine) interpretaba el.class del Applet que ahora puede ser evitado con los Servlets.

Otra ventaja que tienen los Servlets es la de poder limitar al usuario a no cometer errores. En el caso de nuestra aplicación lo hacemos al generar menús colgantes para seleccionar las opciones que puede tener el usuario. En la misma consulta de obtener las localidades que forman un municipio, si no utilizáramos menús colgantes y le diéramos la opción al usuario de escribir el nombre del municipio, podría cometerse el error de escribir el nombre mal, y no obtener los resultados deseados.

En este caso, los menús colgantes también se generan dinámicamente al crear el Servlet. Al lanzar la página, se manda a llamar un proceso para que se incluyan en el menú colgante sólo los nombres de los municipios que están en la base de datos. Esto fue pensado por modificaciones que pudiera tener la base de datos. De tal manera, sólo se le muestra al usuario lo que puede accesar. Y este es el mismo caso de todas las páginas web de consultas. Ver figura 5.4

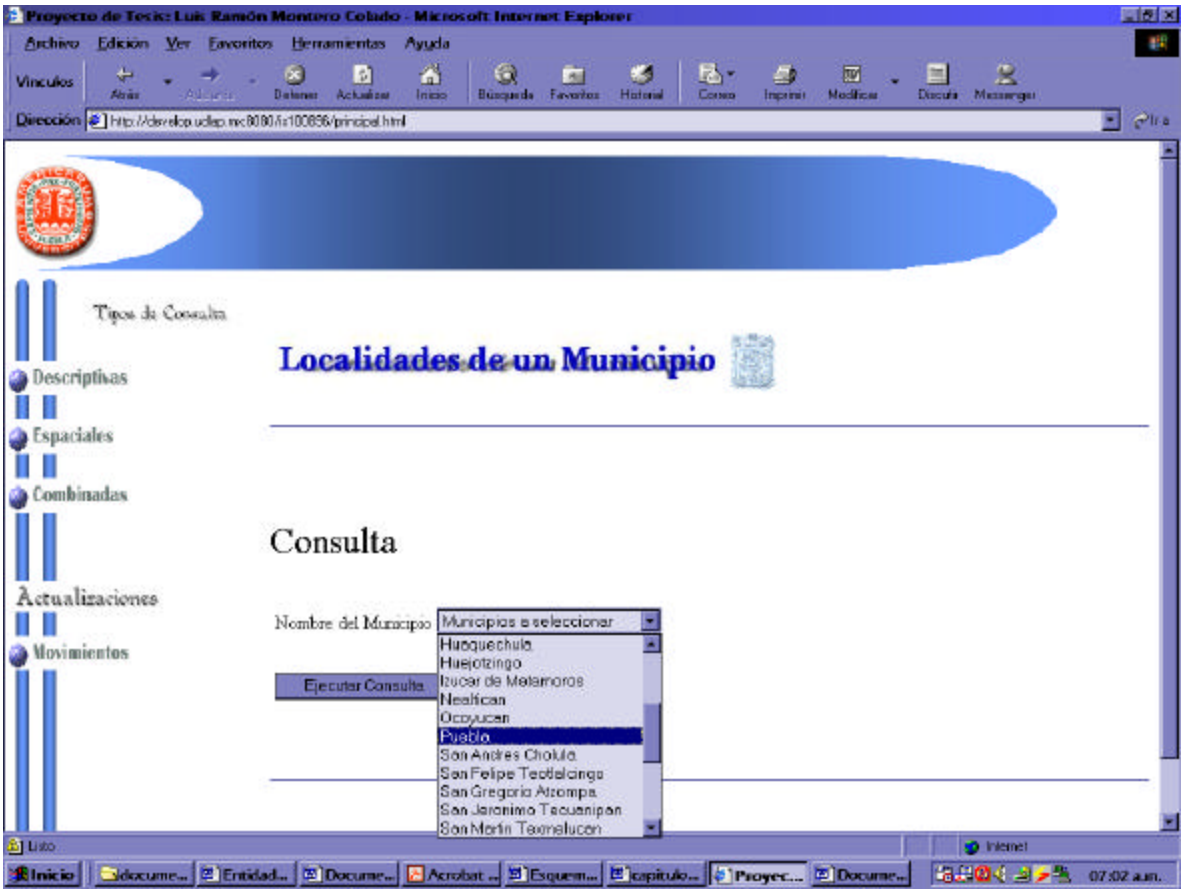

Fig. 5.4 Menús colgantes generados dinámicamente

Cabe señalar que aunque se argumente que los Servlets son independientes de plataforma por ser elaborados en Java, y que no están ligados a algún protocolo en específico, necesitan un compilador especial, y correr sobre un ambiente especial. El ambiente que fue utilizado para compilar y correr la aplicación es TOM CAT v3.2.1. Este tipo de software es Gratis, y se puede encontrar en Internet.

Es evidente que los Servlets a parte de tener todas las características que se han venido mencionado, siguen con una tendencia que hoy día es muy popular. Dicha tendencia

es la facilidad de obtener información vía Internet de manera sencilla y es por esto y por las características siguientes que podemos justificar su utilización para este proyecto.

- Los Servlets son rápidos.
- Se cargan en memoria sólo una vez, y las siguientes veces que se usen se obtienen desde ella.
- Son relativamente fáciles de implementar.
- Por ser generados con código en Java, son independientes de plataforma.

Ya que se tiene generado el Servlet, se produce el tipo de consulta deseada a ejecutar. Esta consulta, como se ha venido mencionando durante todo este documento, puede ser sobre los datos descriptivos o los espaciales. Si es sobre los descriptivos, se generan los objetos propios necesarios directamente y gracias a la consulta. Se hacen los cálculos necesarios y se obtiene el resultado de la consulta representado por un objeto, y finalmente solo se hace un reporte en base al objeto que se haya generado. Existen casos de consultas sobre datos espaciales en las que el procedimiento es el mismo, pero siendo el caso de alguna consulta espacial donde se utilicen las consultas externas el procedimiento cambia. Se genera la consulta en base a la elección hecha en el Servlet. Se generan los objetos propios necesarios para satisfacer la consulta. Dichos objetos son exportados al módulo de consultas externas y ahí se genera el resultado, finalizando con un objeto el cual es interpretado para generar el reporte. Cabe señalar que los objetos generados en todos los tipos de consultas pueden ser desde un Boolean hasta un objeto complejo como puede ser un objeto de tipo Loca_Volcan.

Todos los resultados son traducidos a un reporte presentado en el Servlet.

5.4 INTEGRACIÓN DEL SERVLET CON EL MÓDULO DE CONSULTAS

Para lograr la integración de nuestro sistema, se analizó la posibilidad de extender los Servlets a una técnica aún más nueva que al final se resume en lo mismo. Esta técnica son los JSP (Java Server Pages). Al decir que se resume en lo mismo me refiero que al final todo se traduce en un Servlet, pero su formulación es aún más sencilla.

Con JSP, se puede mezclar código HTML con código de Java. Esto es una gran ventaja porque a diferencia de la programación de Servlets puros, que se programan las clases y resultado de estas son las paginas HTML, los JSP son directos. Solo necesitamos tener el código HTML y si queremos insertar código Java, sólo tenemos que marcar donde empezamos y donde terminamos la inserción con los siguientes indicadores:

<% Este par de caracteres denotan el inicio del fragmento de código Java.

%> Este par de caracteres denotan el fin del fragmento de código Java.

Para que se entienda mejor se muestra ahora un pequeño ejemplo de un archivo JSP.

```
<HTML>
    <BODY>
  <BR> Esto es una prueba de código html <BR>
  <%  String temporal = new String("Universidad de las Americas");
      out.println(" Esta investigación se ha llevado a cabo en la"
                  + temporal);
  %>
    <BODY>
<HTML>
```

El procedimiento que siguen los Servlets es que en el momento que se compila el .java, se genera el .class que es el que va a crear dinámicamente la clase. El problema es que cada vez que se mande a llamar esa clase, se va a estar compilando, que a diferencia de los JSP, se valida solo una vez, y sólo al haber un cambio en el código se vuelve a compilar, de no ser así, el código no se vuelve a compilar.

Como vimos con anterioridad la capacidad de generar objetos de Java dinámicamente dentro del contexto de una página web es relativamente sencillo, y por lo mismo, la integración de los módulos no fue muy compleja. Como las consultas ya están preestablecidas algo importante que se tuvo que hacer fue la obtención de los parámetros

necesarios para generar la consulta, y obviamente el paso de parámetros entre las diferentes páginas.

Cabe señalar que cuando se obtienen los parámetros, aun se está en la página en la que seleccionan los componentes de la consulta. Se tienen que guardar dichos datos en variables, que son pasadas al generar la pagina nueva. Ya que se tiene la página nueva, se recuperan los parámetros, y se hace la consulta. En la pagina donde se muestran los resultados es prácticamente la página en la que se hacen todos los movimientos necesarios para obtenerlos satisfactoriamente

Para definir la presentación de los reportes se tuvo que analizar las interfaces, y ver si realmente eran lo más simple que se podía generar para el manejo por parte del usuario, y en base a ese estimado, se fueron haciendo diferentes pruebas, hasta que se llego a la interfaz final.

El manejo de menús colgantes, a parte de acotar el nivel de error en casi nulo, muestra un manejo de datos muy sencillo. Todo procedimiento de consulta se reduce a seleccionar los elementos necesarios para que la consulta se lleve a cabo y dar clic sobre el botón que inicia el proceso.

5.5 EJEMPLO DE UNA CONSULTA

Ahora se mostrará paso por paso, el procedimiento a seguir para realizar una consulta.

1 En el web browser, en el campo para la dirección escribir la siguiente: http://develop.udlap.mx:8080/is100896/principal.html y aparecerá la siguiente pantalla:

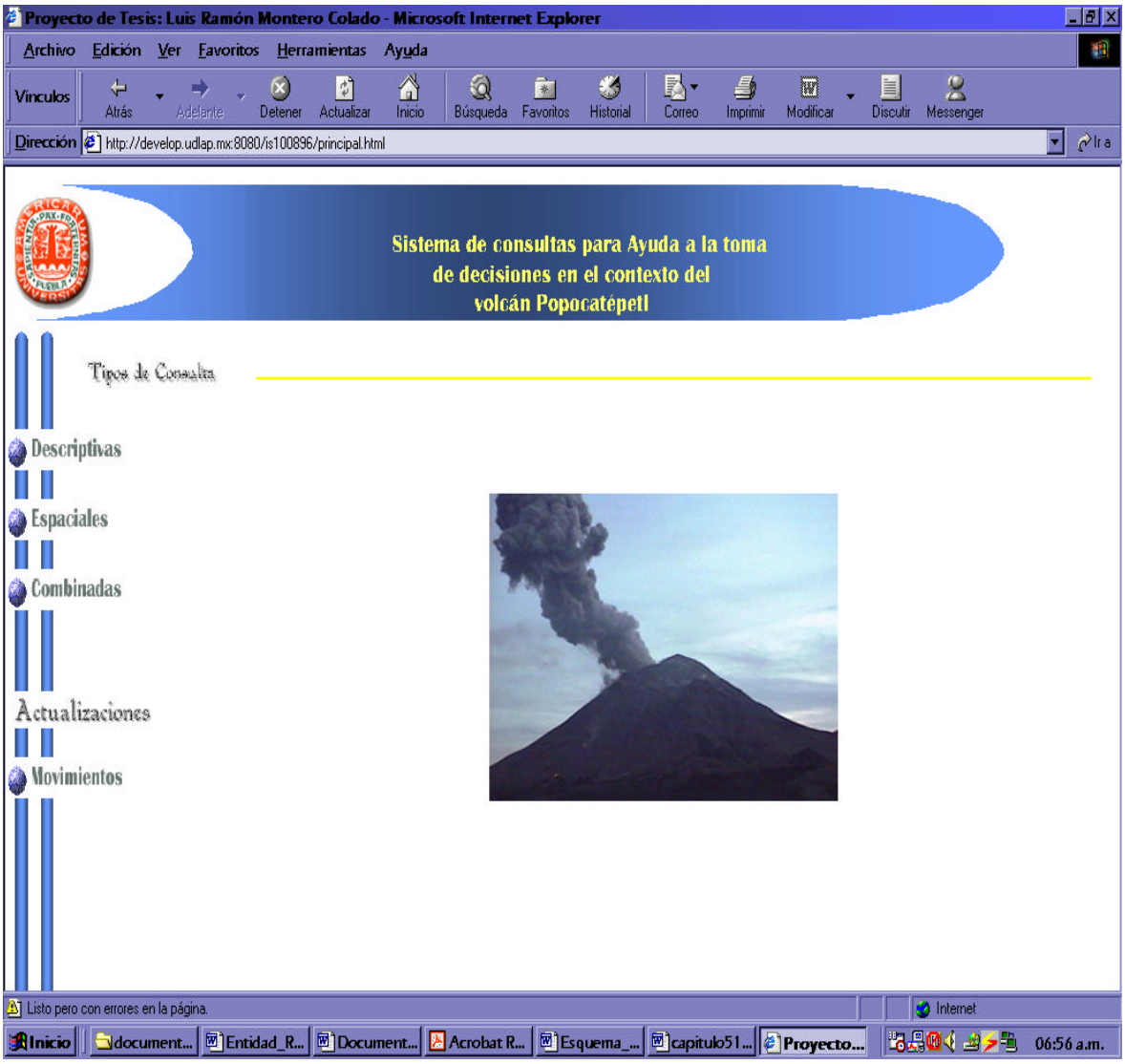

Fig. 5.6 Pantalla principal de la aplicación

Esta es la primera página del conjunto que conforman nuestra aplicación. Del lado izquierdo se tiene el menú de tipo de consultas que se pueden realizar, y una

liga al módulo de actualizaciones y modificaciones a la base de datos realizado por [Morales, 2001]. En el frame del lado derecho es donde aparecen las diferentes páginas donde se seleccionan los parámetros de las consultas y se visualizan los resultados

2 Seleccionar el tipo de consulta que se desea en el frame del lado izquierdo

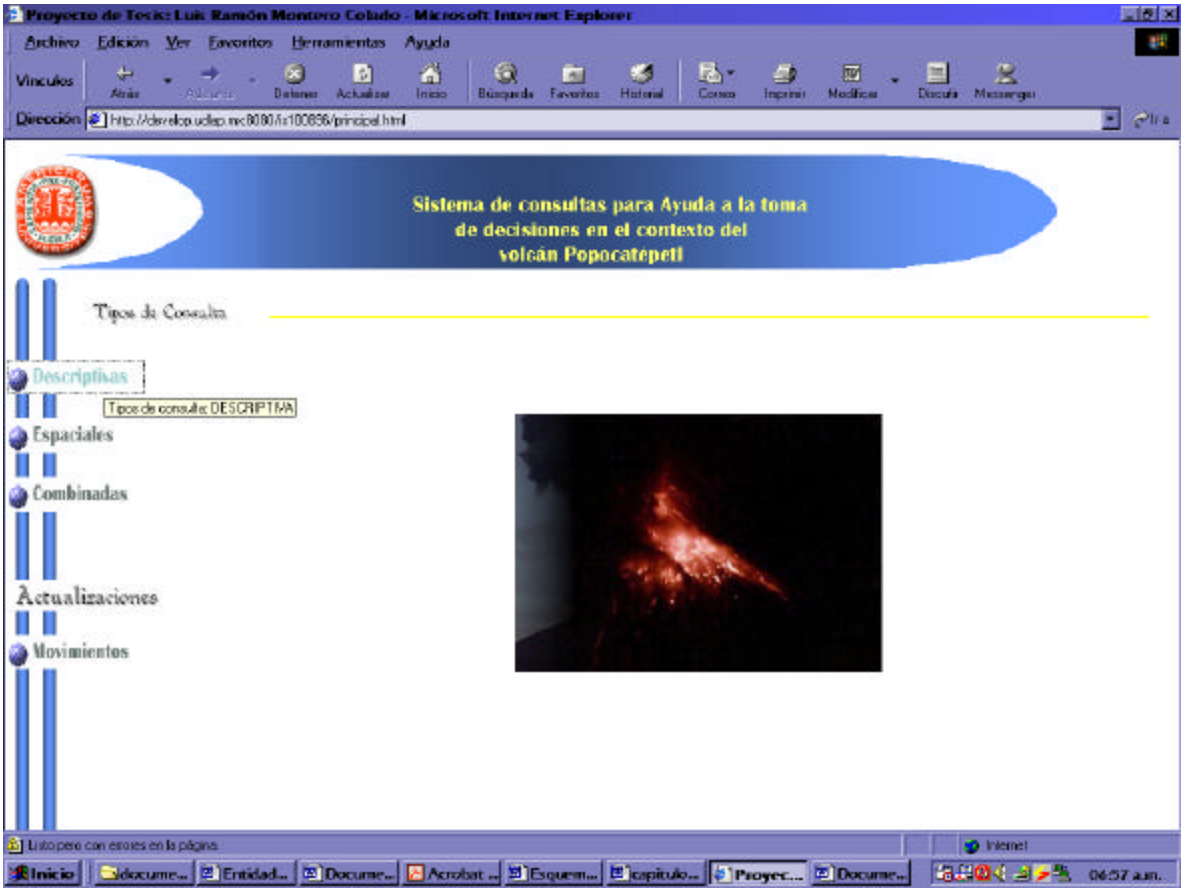

Fig 5.7 Selección de tipo de consulta

Como ya se mencionó, el frame del lado izquierdo es el que controla el flujo entre las diferentes páginas que conforman éste modelo. Se identifica que consulta se está

seleccionando porque cuando se pasa el puntero sobre el tipo, el color preestablecido cambia.

3 Seleccionar la consulta deseada en el menú de consultas preestablecidas

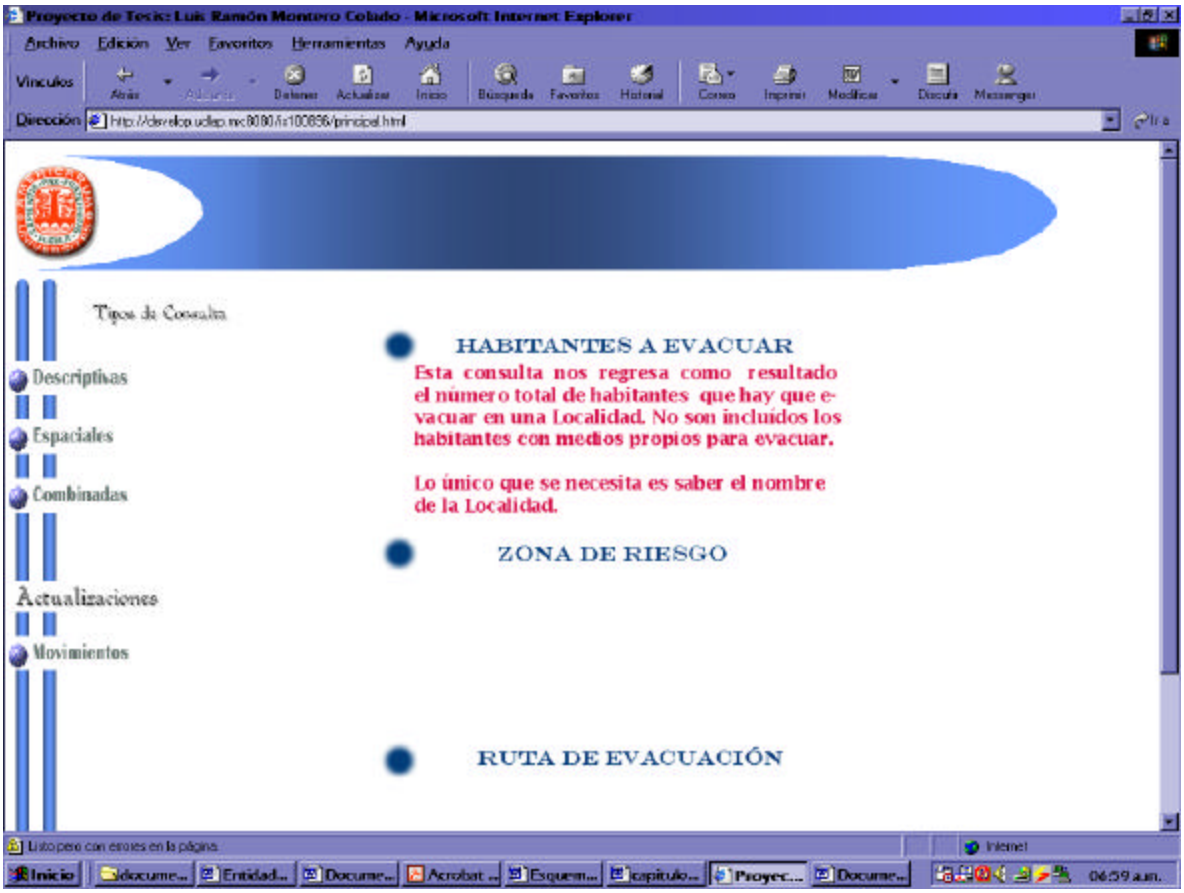

Fig. 5.8 Selección de consulta deseada

Aquí se selecciona la consulta deseada bajo el rubro seleccionado (Descriptiva, Espacial, Combinada). Al pasar el puntero sobre alguna de las opciones, se muestra una descripción del resultado que se obtiene con la consulta.

4 Seleccionar el dato deseado o los datos deseados en el menú colgante y presionar el botón para transmitir la consulta.

Fig.5.9 Selección de dato deseado

Aquí es cuando se generan los menús colgantes de manera dinámica.

5 Visualización y evaluación de los datos obtenidos

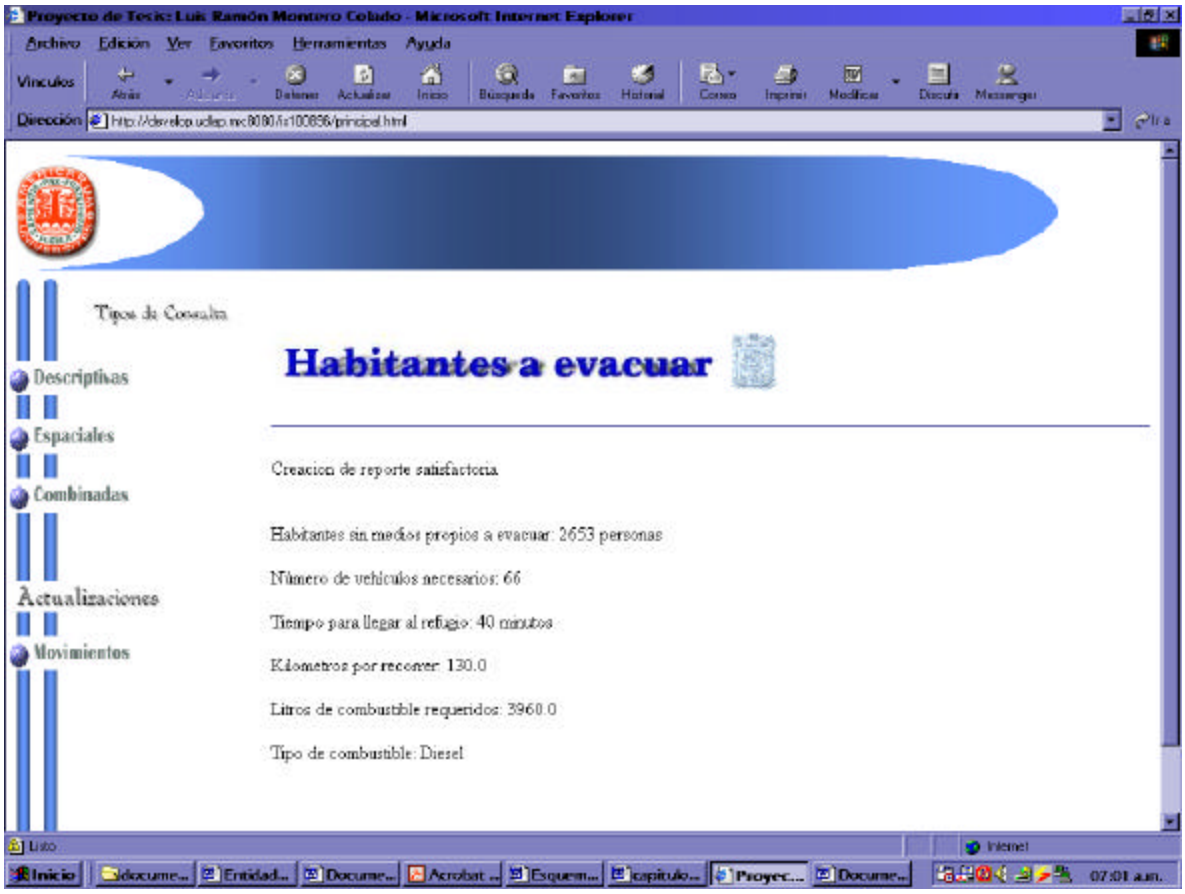

Fig. 5.10 Visualización de resultados

Como se puede apreciar en el procedimiento para hacer una consulta, el manejo de datos que se hace entre las páginas, es muy sencillo y es transparente para el usuario final.

Se considera que se logra el objetivo de satisfacer al usuario final, que en este caso es el Lic. Ramón Peña, pues su necesidad de obtención de datos es satisfecha, así como la facilidad para generar reportes, con la posibilidad de impresión de estos y la sencillez que implica el manejo total del sistema.