

Universidad de las Ciencias Informáticas

Facultad 5

**Título: Propuesta de Visualización de Información Geográfica
Tridimensional de Entornos Urbanos para
Plataformas Web.**

**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autora: Loraima Maning Aldana

Tutor: Ing. Leonardo Rafael Fernández Ruiz

Ciudad de La Habana

Junio del 2010

“Año 53 de la Revolución”

“Nadie ha hecho tanto en tan poco tiempo”

Fidel Castro.

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Loraima Maning Aldana

Firma de la Autora

Ing. Leonardo Rafael Fernández Ruiz

Firma del Tutor

DATOS DE CONTACTO

Tutor: Ing. Leonardo Rafael Fernández Ruiz.

Edad: 26 años.

Ciudadanía: cubano.

Institución: Universidad de las Ciencias Informáticas (UCI).

Título: Ingeniero en Ciencias Informáticas.

Categoría docente: Profesor adiestrado.

E-mail: lfernandez@uci.cu.

Graduado de la UCI, experiencia en representación 3D e interacción en la web, pertenece a la Línea de Software Empotrado del CEDIN en la Universidad de las Ciencias Informáticas.

AGRADECIMIENTOS

A mis padres, Maritza Aldana y Carlos Maning por inculcarme el amor al estudio, al trabajo, enseñarme a esforzarme para alcanzar mis metas sin rendirme nunca.

A mi tía Mildred por acogerme como una hija más y estar siempre pendiente de mí.

A mi tutor Leo por toda su ayuda, dedicación por todos los conocimientos que me transmitió, eres el mejor tutor del mundo.

A Lorié por todo su amor y cariño, por comprenderme y ayudarme tanto a lograr mis objetivos.

A Yausel, Rainer, Parra, Yera y Lester porque me han ayudado siempre que lo he necesitado sobretodo en mis primeros años de estudios que fueron bien difíciles.

A mis amigos de toda la vida (Yisel, Yary, Lary, Rosell), a mis compañeras de apartamento por soportarme tanto y compartir conmigo estos 5 años tan importantes.

A mi mejor amiga Yisel Campos por apoyarme siempre en las buenas y en las malas.

A Yanet Quiala que ha sido como una hermana que siempre me ha apoyado cuando más lo he necesitado.

A mis profesores.

A la Revolución y a Fidel por darme la oportunidad de estudiar, formarme como profesional y como persona en esta maravillosa universidad.

A todos los que de una manera u otra influyeron en mi formación, muchas gracias.

Loraima.

DEDICATORIA

A mis padres Maritza Aldana y Carlos Maning

A la memoria de mi abuela Reina.

RESUMEN

La visualización tridimensional cuenta actualmente con gran aceptación y difusión entre los usuarios de INTERNET; su uso y aplicación ha llegado a los usuarios a través de la web formando parte de su cotidianidad.

Esta vista ofrece un gran número de posibilidades destinadas a lograr una escena más realista para el usuario final, lo cual ha desencadenado la búsqueda de una nueva forma de desarrollar una propuesta que permita visualizar la información geográfica tridimensional de entornos urbanos en la web.

Para desarrollar dicha propuesta, es necesario el empleo de tecnologías web como por ejemplo: XML (Extensible Markup Language) y XSLT (Lenguaje extensible de hojas de estilo de transformación), que permiten a partir de datos geográficos almacenados en un gestor de base de datos, se transformen a un estándar de visualización como X3D; logrando visualizar la información geográfica 3D de entornos urbanos en la web.

Los resultados obtenidos demostrarán la validez de la propuesta realizada y los beneficios que se logran al desarrollar la misma.

Palabras clave: Información geográfica, Visualización 3D, Entornos urbanos, Plataformas Web.

TABLA DE CONTENIDOS

INTRODUCCIÓN..... 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA 5

1.1. VISUALIZACIÓN DE INFORMACIÓN GEOGRÁFICA TRIDIMENSIONAL Y SUS APLICACIONES..... 5

1.2. ANTECEDENTES..... 6

1.3. CAPTURA DE DATOS..... 14

1.4. SISTEMAS GESTORES DE BASES DE DATOS..... 17

 1.4.1. Bases de datos espaciales 18

 1.4.2. Gestores de bases de datos espaciales 19

1.5. ESTÁNDARES DE LA OGC 25

1.6. VISUALIZACIÓN GEOGRÁFICA TRIDIMENSIONAL EN LA WEB 29

 1.6.1. Interfaz gráfica de usuario (GUI)..... 29

 1.6.2. VRML/X3D 30

 1.6.3. JAVA3D..... 31

1.7. LENGUAJES DE PROGRAMACIÓN WEB 31

1.8. ESTÁNDARES DE PROGRAMACIÓN WEB..... 34

1.9. ARCGIS..... 36

1.10. SELECCIÓN DE TECNOLOGÍAS PARA DESARROLLAR LA PROPUESTA..... 37

CONCLUSIONES DEL CAPÍTULO 37

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN..... 38

2.1 ARQUITECTURA..... 38

 2.1.1 Capa de datos..... 39

 2.1.2 Capa lógica 45

 2.1.3 Capa de presentación 48

2.2 PROTOTIPO Y DESCRIPCIÓN DEL SISTEMA..... 49

CONCLUSIONES DEL CAPÍTULO 50

CAPÍTULO 3: ANÁLISIS DE LOS RESULTADOS 51

3.1 RESULTADOS..... 51

3.2 EVALUACIÓN DE LA PROPUESTA 55

CONCLUSIONES DEL CAPÍTULO 58

CONCLUSIONES 59

RECOMENDACIONES 60

REFERENCIAS BIBLIOGRÁFICAS..... 61

BIBLIOGRAFÍA CONSULTADA 63

ANEXOS..... 64

GLOSARIO DE TÉRMINOS 67

ÍNDICE DE FIGURAS

| | |
|---|----|
| FIGURA 1: PROTOTIPO SIG WEB (SLATANOVA, 2000). | 7 |
| FIGURA 2: PROTOTIPO SIG WEB (VRIES, 2003). | 9 |
| FIGURA 3: PROTOTIPO SIG 3D WEB (VRIES Y OTROS 2003)(HELD Y OTROS 2004). | 10 |
| FIGURA 4: PROTOTIPO BASADO EN X3D, WFS Y ORACLE (VRIES, 2004). | 11 |
| FIGURA 5: PROTOTIPO DE SIG 3D AVANZADOS BASADOS EN VRML, JAVA, JDBC Y LA EIA (ZHU Y OTROS, 2004). | 12 |
| FIGURA 6: PROTOTIPO SIG 3D BASADO EN X3D, FLUX Y POSTGRES (NINSAWAT, 2006). | 13 |
| FIGURA 7: 3D GIS PROTOTYPE BASED ON X3D, ORACLE, AND JAVA (KUMKE Y OTROS, 2006). | 14 |
| FIGURA 8: JERARQUÍA DE LAS CLASES GML. | 28 |
| FIGURA 9: ARQUITECTURA PROPUESTA. | 39 |
| FIGURA 10: TABLAS DE LA PROPUESTA. | 43 |
| FIGURA 11: SPATIAL_REFERENCE_SYS. EJEMPLO. | 44 |
| FIGURA 12: GEOMETRY_COLUMN. EJEMPLO. | 45 |
| FIGURA 13: MODELO DE LA APLICACIÓN USANDO AJAX3D. | 48 |
| FIGURA 14: PASOS PARA LLEVAR A CABO LA PROPUESTA DE SOLUCIÓN. | 50 |
| FIGURA 15: IMAGEN DE SHAPE. | 52 |
| FIGURA 16: IMAGEN DE LOS RESULTADOS. | 53 |
| FIGURA 17: PROPUESTA CON INTERFAZ INTEGRADA. | 54 |
| FIGURA 18: SHAPE DEL MODELO. | 64 |
| FIGURA 19: MODELO SIN ALTURAS ESPECÍFICAS. | 65 |
| FIGURA 20: MODELO CON ALTURAS ESPECÍFICAS. | 65 |
| FIGURA 21: MODELO CON INTERFAZ. | 66 |

ÍNDICE DE TABLA

| | |
|---|----|
| TABLA 1: ESTÁNDARES DE OGC. | 26 |
| TABLA 2: TIPOS DE SHAPE QUE EXISTEN EN ARCVIEW. | 40 |
| TABLA 3: RESULTADOS DE LAS PRUEBAS DE CARGA Y ESTRÉS. | 57 |

INTRODUCCIÓN

La aparición de Internet, como medio de comunicación, ha supuesto que el acceso a la información sea sencillo y rápido. La mayor parte de esta información reside en las conocidas páginas web que suelen presentar textos e imágenes en dos dimensiones [1]. Como la percepción se organiza de forma natural en un entorno tridimensional, el espacio 2D es cerrado, confinado, por más que en él se empleen recursos para representar la profundidad. Requiere, un proceso mental de carácter analítico, convergente, y de abstracción. El espacio 3D es el ámbito en el que la mente se desenvuelve libremente, abierta, conectada de forma natural al sinfín de estímulos que se relacionan armónicamente con el esquema interpretativo, de ahí la conveniencia de la integración de una tercera dimensión que permita por ejemplo: recorrer la geografía de una ciudad o país y brindar información acerca de ésta.

Si se toma un momento para pensar en algunos de los problemas que afectan actualmente al mundo como superpoblación y desastres naturales, se pueden observar que cada uno de ellos tiene una dimensión geográfica crítica. Por ejemplo, cuando se considera el impacto de un desastre tal como un terremoto necesita analizarse información geológica y sísmica, para luego relacionarse con detalles sobre la población o infraestructuras que han sido o pueden ser afectadas. La solución a muchos de estos problemas frecuentemente, requiere de acceso a varios tipos de información que sólo pueden ser relacionadas por geografía. Sólo la tecnología de SIG permite manipular, utilizar información usando geografía y analizar los patrones, relaciones y tendencias de la información para tomar mejores decisiones.

La visualización tridimensional actualmente cuenta con gran aceptación y difusión entre los usuarios. En los últimos años su uso y aplicaciones, se han difundido a los usuarios de Internet a través de la web formando parte del uso cotidiano.

Lo que un día constituyó un gran avance a través de la visualización de los mapas en 2D, hoy se ha visto incrementado a través de la visualización 3D. Esta vista ofrece un mayor número de posibilidades, permitiendo, además de visualizar en detalle el terreno, cargar y visualizar los modelos de las edificaciones existentes, simular la luz solar, la atmósfera terrestre y otros efectos destinados a proporcionar una escena realista al usuario final.

Es conveniente cuando la visualización de una tercera magnitud, resulta útil para la interpretación de los datos que se quieren mostrar. Proporciona información detallada del entorno construido a diferentes escalas, proveen una buena solución a problemas que no pueden resolverse fuera de un entorno 3D. Ayuda a la comprensión de información compleja.

Los sistemas existentes, si bien están a la altura para hacer frente a la geometría tridimensional, le faltan extendidos de análisis espacial y temático. En este sentido, muchos autores consideran que un SIG 3D, que mantiene en topología tridimensional la información espacial y temática, es la solución más exitosa.

La Universidad de las Ciencias Informáticas (UCI), es un importante centro de estudios que ha incursionado en el desarrollo de entornos virtuales Tridimensional al igual que en el desarrollo de Sistemas de Información Geográfica bidimensional, pero aún no se tiene experiencia en el desarrollo de SIG 3D para plataformas WEB que permita una mejor representación e interacción con la información geográfica, un ejemplo de esto es el Centro de geoinformática y señales digitales (GEISED) de la UCI, en el cual se trabaja en el desarrollo de un sistema de vigilancia y carecen de un sistema que visualice la información geográfica tridimensional, por lo que se hace engorroso poder conocer la visibilidad de una cámara, el perfil de altura, la posición más factible a la hora de distribuir las cámaras de vigilancia, pues con un sistema 2D se hace muy difícil conocer esta serie de aspectos, pues no se cuenta con la altura de los modelos representados.

De la situación problemática anterior se obtuvo el siguiente **problema científico**: ¿Cómo visualizar tridimensionalmente la información geográfica de entornos urbanos en la web?

Dada la situación anterior se propone como **objeto de estudio**: Visualización geográfica 3D para la web y como **campo de acción**: Visualización de información geográfica 3D de entornos urbanos para la web.

El **objetivo general** de la investigación científica es: Desarrollar una propuesta que permita visualizar la información geográfica 3D de entornos urbanos en la web.

Para dar cumplimiento al objetivo general se plantean las siguientes **tareas de investigación**:

- Análisis de las tendencias actuales del desarrollo 3D en la web para seleccionar la más adecuada para una aplicación.

- Análisis de los principales sitios de visualización geográfica existentes para definir el diseño y las funcionalidades principales que debe brindar una aplicación.
- Selección de las tecnologías WEB más adecuadas para la implementación de una aplicación.
- Análisis de estándares para la confección de los modelos geográficos 3D.
- Selección de las tecnologías para la persistencia de los datos geográficos más adecuadas para la implementación de una aplicación.
- Determinación de las herramientas a utilizar para el desarrollo del sistema.
- Implementación de la aplicación que demuestre los resultados de la propuesta.
- Realización de las pruebas de rendimiento de la aplicación.
- Recolección de los resultados de las pruebas.

Idea a defender: Se logrará la visualización de la información geográfica 3D de entornos urbanos mediante la implementación de una aplicación para plataformas web.

Métodos de investigación:

Métodos teóricos:

- **Analítico-sintético:** La utilización de este método teórico posibilitará el análisis de los elementos más importantes que se relacionan con la Visualización geográfica 3D para la web. Dará la facilidad de obtener y procesar la información de lo más importante y significativo relacionado con la utilización de estas técnicas de visualización 3D en plataformas web.
- **Histórico-lógico:** Mediante la utilización de este método se puede estudiar la evolución que ha tenido hasta actualidad todo lo referente a la visualización tridimensional sobre plataforma web.

Métodos empíricos:

- **Observación:** Este método será aplicado en distintos momentos del proceso de investigación con el objetivo de evaluar el impacto de las distintas técnicas de visualización 3D en la WEB en el rendimiento de la computadora, con lo cual se podrá seleccionar en cada momento la más adecuada para emplear en la aplicación.

La presente investigación cuenta con 3 capítulos:

Capítulo 1. Fundamentación teórica: Se define el marco teórico conceptual que posibilita la comprensión metodológica de la solución propuesta para lo cual se realiza un análisis de los trabajos realizados anteriormente sobre visualización de entornos urbanos, las técnicas, lenguajes y estándares de visualización.

Capítulo 2. Propuesta de solución: Aborda una secuencia de pasos lógicos que se deben tener en cuenta para lograr la visualización tridimensional de entornos urbanos en la web. Esta propuesta está basada en trabajos anteriores y tendencias actuales sobre visualización en la web y en el uso de estándares de visualización.

Capítulo 3: Análisis de los resultados: Se analizan los resultados que arrojaron las pruebas realizadas a la propuesta, haciendo uso de la herramienta JMeter para realizar pruebas de carga y estrés.

Introducción

En el presente capítulo se pretende realizar un análisis teórico que facilitará la comprensión de la solución propuesta, para el cual se hace un estudio de las herramientas y de los conceptos fundamentales relacionados con Información geográfica tridimensional, plataformas WEB y entornos urbanos.

1.1. Visualización de información geográfica tridimensional y sus aplicaciones

Entre las aplicaciones de la visualización geográfica tridimensional desde el punto de vista de toma de decisiones y otros aspectos, se encuentran [2]:

- **La planificación comparativa:** Simulación de escenarios para cada una de las alternativas del proyecto y comparación frente al contexto. Este fue el caso que se aplicó en Maryland (Anderson et als.1996) para diseñar el conector InterCounty. Se pudo evaluar y comparar el impacto entre incluir una vía de trenes de alta velocidad, una tradicional o ninguna. El modelo fue construido a partir de datos GIS 2D donde las huellas de los edificios fueron extruidas de acuerdo a su altura.
- **La resolución de conflictos:** Para identificar áreas problemáticas, evaluando distintas variantes de proyecto. Como ejemplo se puede mencionar al Plan Maestro para Leidsche Rijn (Utrecht). En ese caso, el usuario o navegador podía alternar entre las diferentes opciones modificando variables como la inclinación de la autopista y visualizar el impacto ambiental de cada alternativa desde distintos puntos de vista. El usuario no sólo podía ver las alternativas sino también escuchar el ruido provocado por el tráfico según cada opción (van Dipten y van Klaveren, 1996).

- **Los estudios de visibilidad:** En la señalización del tránsito para corroborar su visibilidad desde todas las direcciones (Ranzinger, 1995) o en la verificación de cobertura o visibilidad de antenas transmisoras.
- **Los estudios de morfología y diseño:** Es la comparación de cambios en la volumetría edificada según la aplicación de distintos códigos urbanos.

Otras aplicaciones que tiene la visualización de información geográfica tridimensional son [3]:

- Disminuir el tiempo que lleva responder a requerimientos de clientes.
- Encontrar terrenos adecuados para desarrollo.
- Buscar relaciones entre cultivos, suelos y clima.
- Localizar la posición de cortes en circuitos eléctricos.

1.2. Antecedentes

Recientes implementaciones de sistemas de información geográfica tridimensionales, se dedican principalmente a la visualización tridimensional y simples componentes interactivos; como el acceso a la información adicional. Otros principios generales de los SIG como análisis de los datos aún están desaparecidos. La razón de ello es que la gestión de datos relacionados no es adecuada para la funcionalidad 3D real. Sin embargo hay un par de prototipos disponibles, que apuntan hacia los SIG 3D. A continuación se abordan algunas descripciones breves de trabajos sobre este tema.

Un prototipo de sistema tridimensional de SIG (Slatanova, 2000) [4]

El sistema desarrollado es un VRML (del inglés Virtual Reality Modeling Language) proporciona la típico ejemplo de clientes reducidos basados en HTTP, scripts CGI (realizado en Perl), VRML y los documentos HTML que se crean al vuelo. La información de gráficos tridimensionales obtenidos, como resultado de consultas espaciales, proporciona los medios para consultar gráficamente los objetos observados en la escena tridimensional. Los documentos HTML se utilizan para visualizar texto e imágenes, para especificar las consultas SQL o introducir nuevos valores para los elementos modificados.

Para solicitar información sobre un objeto particular, es necesario escribir su identificación en un formulario HTML o hacer clic en el objeto correspondiente en VRML.

La extracción de un grupo de objetos de acuerdo a un criterio, se completa directamente escribiendo consulta SQL en la sección consulta. El resultado de la consulta se muestra ya sea en una página HTML o en un documento VRML. Estos documentos se crean en la marcha, sólo con la información relacionada a los objetos de interés. El mismo mecanismo se utiliza para crear, eliminar, actualizar e insertar formas para editar datos.

El libre acceso a la base de datos proporciona un mecanismo para especificar y mostrar una amplia gama de consultas espaciales. Ejemplos de tales preguntas son "cuál es el edificio más alto", "mostrar el resultado de los edificios en un área determinada", "mostrar todas las calles", "mostrar todos los edificios administrativos". Una ventaja del sistema es que los clientes prácticamente no utilizan ningún software específico, además de navegador Web y un plug-in.

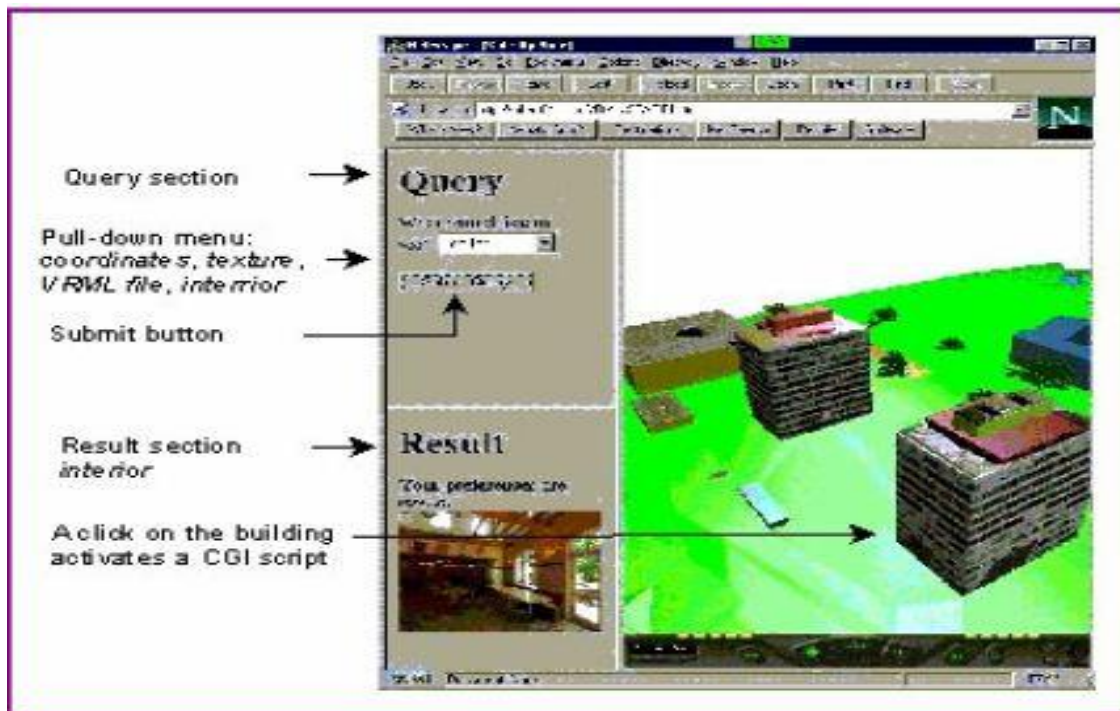


Figura 1: Prototipo SIG Web (Slatanova, 2000)

SALIX (Lammeren y Hoogerwerf; 2003, Wachowicz, 2002) [5]

El sistema está diseñado para la planificación del paisaje interactivo, la planificación de los árboles y arbustos y simular su crecimiento. La interfaz gráfica de usuario se basa en el entorno de Cortona, utilizando VRML y Java para proporcionar toda la funcionalidad. El Sistema de gestión de bases de datos (SGBD) es usado solamente para almacenar los objetos de interés (variedad de árboles y arbustos).

Los objetos se colocan manualmente en el campo de visión. Un gran número de barras de herramientas brindan la posibilidad a los usuarios de inspeccionar ciertas restricciones, la distancia entre los árboles plantados en diferentes etapas de su vida, para simular el crecimiento, la creación de conglomerados de objetos del mismo tipo, etc.

Prototipo SIG 3D basado en VRML, ASP y una base de datos Access (Vries, 2003) [4]

Esta aplicación hace uso de tecnologías web para lograr un SIG tridimensional. Los datos geográficos se encuentran disponibles en un archivo VRML y los atributos están almacenados en un sistema de base de datos Microsoft Access y se pueden consultar de forma dinámica.

La tecnología ASP (del inglés Active Server Pages) combinada con Servicios de información de INTERNET (IIS) como servidor web es usada para la interacción con la base de datos. El mundo VRML está dentro del marco principal de la página HTML. La interacción del usuario se posibilita en forma de consulta de cada atributo de los objetos de datos. Si el usuario hace clic en un objeto en el mundo VRML se envía una petición al servidor. Después de conectarse a la base de datos, ASP es la creación de un adecuado fragmento HTML que contiene los datos de atributo solicitado en una tabla.

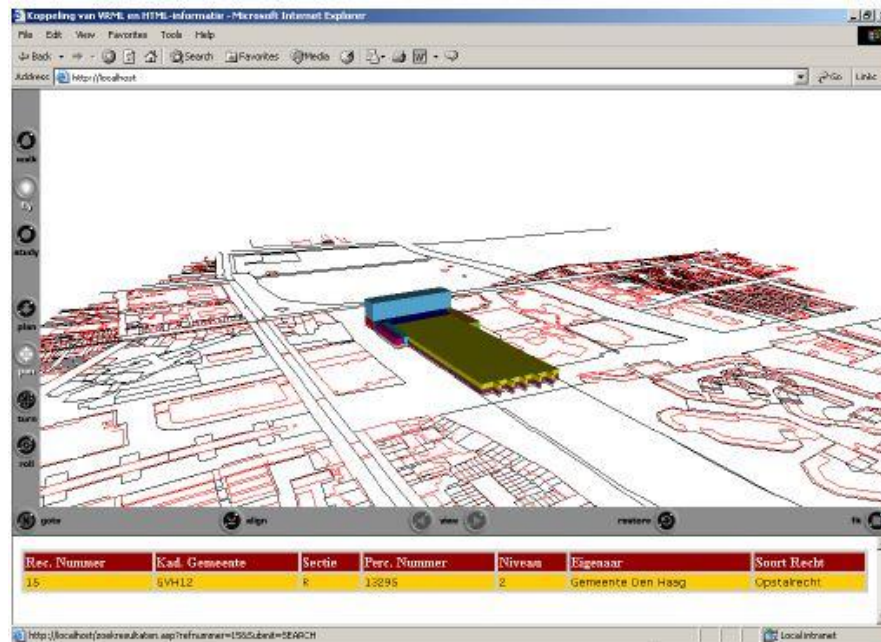


Figura 2: Prototipo SIG Web (Vries, 2003)

X3D, Java Servlets, XSQL and Oracle9i (Vries, 2003) [5]

Este prototipo de sistema está basado en una arquitectura de base de datos integrada. Se usan Oracle9i y su extensión espacial, del lado del servidor, el sistema se basa en un contenedor de servicios Java, como Apache Tomcat y el servidor HTTP Apache. En concreto, el prototipo usa XML, librerías Java para consultas e intercambio de datos. Las bibliotecas son parte de Oracle XML Developer Kit (XDK) que se integran en Tomcat. XDK proporciona un analizador de XML y el procesador de XSLT. Para visualizar las consultas, la respuesta XML de la base de datos se transforma en X3D. El marco principal para mostrar el mundo virtual, por otra para la visualización de los atributos correspondientes del objeto utilizando las tablas HTML. El tercer marco es ofrecer formularios HTML para consultar la base de datos de objetos espaciales. Una vez que se realiza una consulta, el marco principal se visualizará la nueva escena.

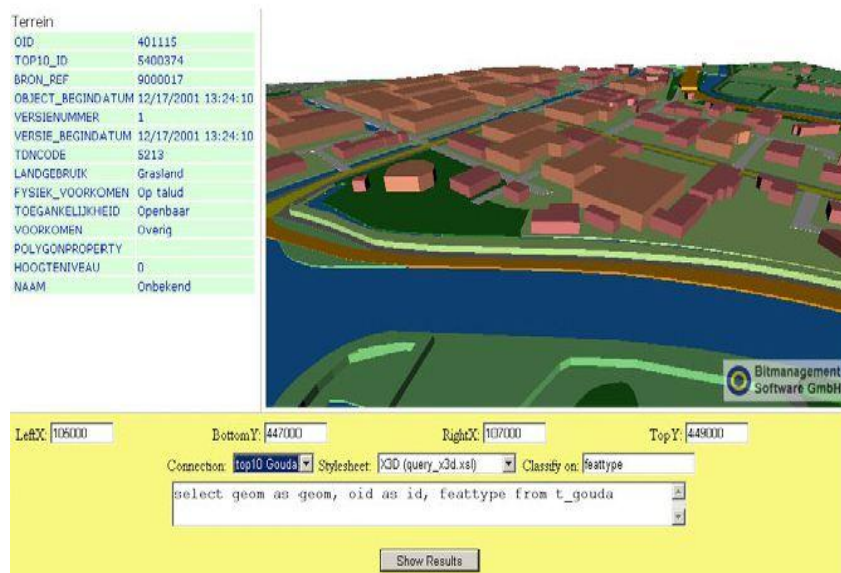


Figura 3: Prototipo SIG 3D Web (Vries y otros 2003)(Held y otros 2004)

Visualización geográfica tridimensional de Tasmania (Beard, 2004) [4]

En Australia (Beard, 2004) se creó un conjunto de datos geológicos de Tasmania utilizando VRML. Diferentes capas se pueden agregar o quitar de la escena mediante un menú a la izquierda. La visualización 3D se altera usando JavaScripts, que puede acceder directamente a los Datos VRML. Este proyecto muestra que es posible modificar una escena en 3D con JavaScripts sin tener que volver a cargar el conjunto de datos completos.

No se utiliza una base de datos para este proyecto, los datos ya se encuentran en un archivo VRML. No se pueden hacer consultas de selección a los datos, aunque diferentes capas 3D se pueden agregar o quitar partir de los datos en función de lo que el usuario quiere ver.

Dentro de sus desventajas están:

- Brinda la posibilidad de seleccionar las características dentro de una capa.
- No hay una solución para hacer frente a grandes conjuntos de datos en forma de capas completo se proporcionan como nodos VRML por lo tanto el usuario debe esperar hasta que se carguen.

Prototipo basado en X3D, WFS y Oracle (Vries, 2004) [4]

Los datos y la geometría son almacenados en un gestor de base datos Oracle espacial. Una Web Feature Service (WFS) lee los datos desde la base de datos mediante un protocolo JDBC (Java Database Connectivity) y salidas GML3, que soporta geometría tridimensional. El GML3 se procesa, se transforma y se devuelve al cliente como X3D, por el lado del cliente, JavaScript y un transformador de hoja de estilo (XSLT). El estándar X3D se visualiza con el plugin BsContact.

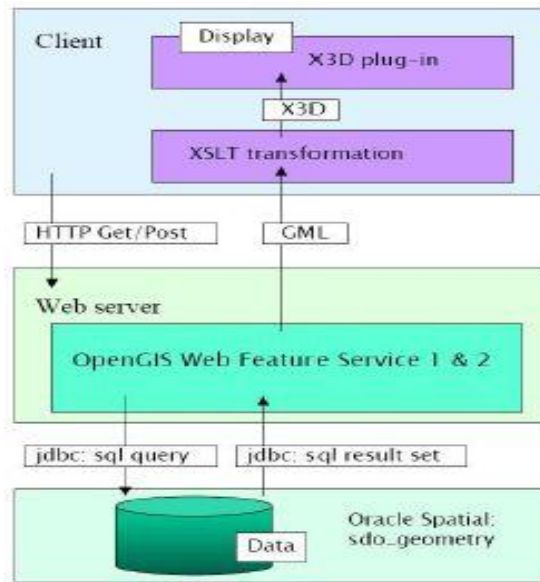


Figura 4: Prototipo basado en X3D, WFS y Oracle (Vries, 2004)

Prototipo de SIG 3D avanzados basados en VRML, Java, JBDC y la EIA (Zhu y otros, 2004) [4]

Continuando el trabajo de Kim en Singapur, Zhu propuso un prototipo basado en Java, JDBC, y EAI. Este prototipo se enfoca principalmente en la visualización del terreno y no en la visualización de otros objetos 3D. El prototipo tiene dos componentes principales, un navegador VRML y una interfaz de usuario Java. Es usado también un servidor web Apache.

En el VRML se hace uso de nodos de secuencia de comandos. Mediante el uso de Java en la secuencia de comandos, la información puede ser intercambiada entre la interfaz gráfica de usuario (GUI) de Java y VRML. Los mapas son traducidos a VRML de la base de datos con el componente EAI mientras JDBC se

utiliza para la entrada de las consultas. La consulta de la ruta en este prototipo es limitada, sino que es utilizada exclusivamente para obtener resultados textuales de las consultas sobre la base de datos.

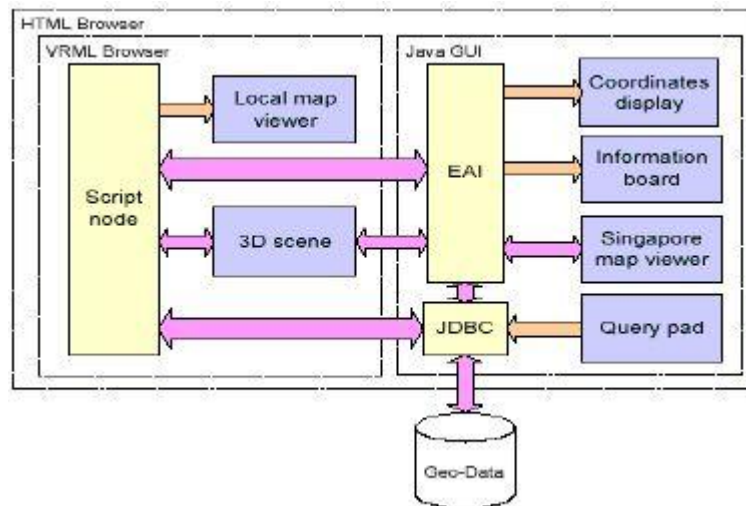


Figura 5: Prototipo de SIG 3D avanzados basados en VRML, Java, JDBC y la EIA (Zhu y otros, 2004)

Prototipo SIG 3D basado en VRML/X3D, Java y la EIA (Racinc y otros, 2005) [4]

La más reciente aplicación SIG web hecha con VRML se desarrolla en Serbia y Montenegro. Ellos presentan un prototipo para la visualización en 3D para la web. Éste es similar al prototipo de ZHU 2004. El prototipo puede visualizar DEMs, 2D, 2.5D, y funciones 3D, que son almacenados en bases de datos diferentes. Las tecnologías usadas son Java, EAI, VRML and X3D. Cuando se utiliza el zoom dentro o fuera del terreno, las diferentes cargas del terreno se han cargado, de esta manera se pueden tratar grandes conjuntos de datos.

Prototipo SIG 3D basado en X3D, Flux y PostgreSQL (Ninsawat, 2006)

En Japón, Ninsawat creó una solución de código abierto para el desarrollo de SIG 3D para la Web usando X3D. Permite la visualización del terreno usando como medio de resolución DEMs, y la visualización de los datos del pozo o de infraestructura de red de capas. Todo el software utilizado es de código abierto, se utiliza PostgreSQL como gestor de base de datos, Python para secuencias de comandos, X3D como formato de visualización tridimensional y un reproductor de flujo para visualizar los archivos X3D. Los

datos se devuelven al usuario en una interfaz HTML con plugin de flujo integrado. Las consultas pueden ser recogidas por JavaScript, y se remitirán como petición HTTP al servidor web, que recoge, y lo pasa a la base de datos.

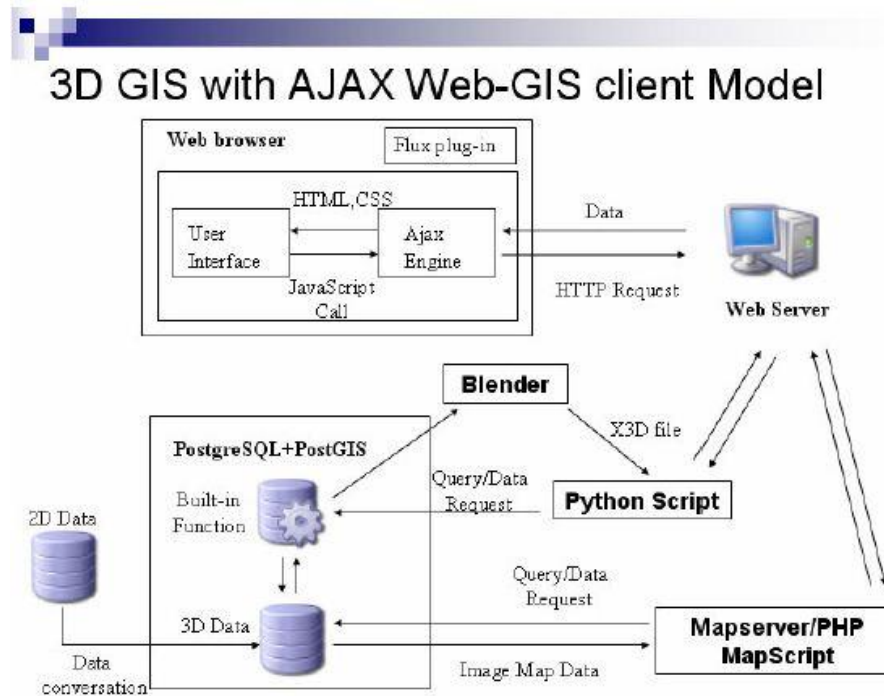


Figura 6: Prototipo SIG 3D basado en X3D, Flux y PostgreSQL (Ninsawat, 2006)

3D GIS prototype based on X3D, Oracle, and Java (Kumke y otros, 2006) [4]

En Alemania, Kumke propuso una aplicación para la visualización de datos 3D de edificio, y la integración de las texturas y las características térmicas. La construcción de modelos 3D y los datos de sensor de infrarrojos se almacenan en una base de datos. El análisis de los datos se divide en el análisis de la geometría y el análisis de las características y propiedades. Estos últimos clasificados. Los resultados de ambos análisis se envían a la base de datos.

Los datos geométricos se almacenan en una base de datos Oracle 9i como polígonos en 2D con un componente z en la representación de Fronteras. Además, las imágenes del infrarrojo y la parte visible del espectro se encuentran almacenadas en la base de datos. La geometría se traduce a X3D utilizando Java

3D y presentado en el cliente. Las imágenes también se envían al cliente y se hace referencia a ellas como texturas en X3D.

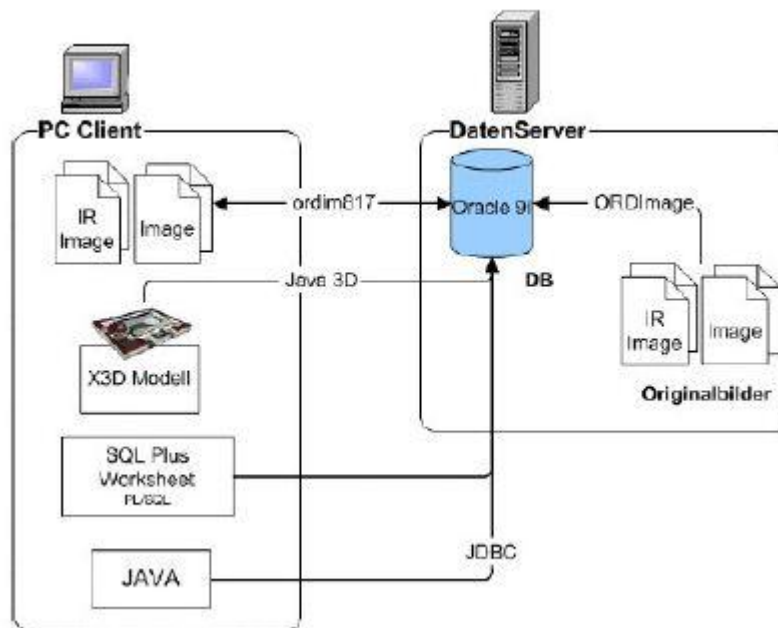


Figura 7:3D GIS prototype based on X3D, Oracle, and Java (Kumke y otros, 2006)

1.3. Captura de datos

Los datos constituyen uno de los elementos más importantes de todo SIG. Los procedimientos para llevar a cabo la captura de datos pueden ser diferentes en dependencia con el modelo de datos que se utilice, ya sea vectorial o raster, aunque es posible cambiar la información de un sistema a otro, ésto se denomina vectorización de información raster (raster - vectorial) o rasterización de información vectorial (vectorial -raster).

En un SIG raster se puede incorporar la información analógica contenida en mapas y planos en soporte papel por medio de barredores ópticos o escáner; la información digital puede incorporarse directamente de ficheros en formato raster o convertirse a partir de ficheros en formato vectorial mediante procedimientos de rasterización.

Escáner

El barredor óptico o escáner es un periférico que permite obtener rápidamente una imagen digital a partir de una imagen analógica. Un dispositivo envía un haz de luz sobre la imagen analógica, que es reflejada con distinta intensidad por el papel. La intensidad luminosa es registrada mediante un sensor óptico capaz de diferenciar un elevado número de píxeles, que se convierte en un fichero raster, y que una vez introducido y reconocido por el sistema, cada píxel, podrá ser gestionado como una tesela o celda que contiene diferente información [6].

Ventajas:

- El borrador óptico o escáner es muy rápido.
- Puede llegar a una alta resolución.

Desventajas:

- En muchas ocasiones es necesario preparar los mapas originales, redibujándolos manualmente, para seleccionar exclusivamente aquella información que debe figurar en forma de capa temática en el SIG.

Imágenes de satélites

Habitualmente, mediante sensores remotos, se registra en formato digital la radiación emitida por la superficie de la tierra. La resolución de estas imágenes es inestable en función de los distintos sensores remotos en activo, pudiendo alcanzar hasta el metro en algunos sensores comerciales de reciente puesta en órbita. Estas imágenes pueden ser tratadas para producir información geográfica [6].

Ventajas:

- La información que se obtiene de los sensores remotos siempre es actual.
- La información que se obtiene para algunos sensores remotos cubre toda la superficie de la tierra.

Desventajas:

- La resolución no siempre es adecuada para determinados tipos de análisis.

- La clasificación de imágenes ofrece resultados que no alcanzan el grado de exactitud requerido en determinadas aplicaciones.

Shapefile

Los “shapefile” son archivos que almacenan por un lado datos geométricos y por otro atributo de la información espacial en un conjunto de datos [7]. Los shapefile soportan los tipos básicos de elementos, es decir: el punto, la línea y el polígono.

Los atributos descriptivos son almacenados en un archivo dBase. Un shapefile consta de un archivo principal, un archivo de índices y una tabla dBase. En el archivo principal se tiene acceso directo a los registros de longitud variable donde cada registro es un “shape” o un objeto geométrico con una lista de vértices. En el archivo de índices, cada registro es el apuntador al registro correspondiente en el archivo principal. La tabla dBase almacena los atributos de los objetos con un registro por objeto [7].

Todos los archivos siguen la convención de 8 caracteres de nombre y 3 de extensión. El archivo de índice, el archivo principal y el archivo de base tienen el mismo nombre y las extensiones de cada uno son las siguientes:

- Archivo principal (.shp)
- Archivo de índice (.shx)
- Tabla dBase (.dbf)

Opcionalmente existen otros como [8]:

- **.sbn** y **.sbx** - Almacena el índice espacial de las entidades.
- **.fbn** y **.fbx** - Almacena el índice espacial de las entidades para los shapefiles que son inalterables (sólo lectura).
- **.ain** y **.aih** - Almacena el índice de atributo de los campos activos en una tabla o el tema de la tabla de atributos.
- **.prj** - Es el archivo que guarda la información referida a sistema de coordenadas.

- **.shp .xml** - Almacena los metadatos del shapefile.

Ventajas [8]:

- Se dibujan más rápidos.
- Se puede convertir un tema existente a shapefile.
- Se puede crear un nuevo tema basado en un shapefile.
- Se pueden editar shapefile.

1.4. Sistemas gestores de bases de datos

Un Sistema de Gestión de Bases de Datos (SGBD) es un programa de ordenador que facilita una serie de herramientas para manejar bases de datos y obtener información de ellas. Además de almacenar la información, se le pueden hacer preguntas sobre esos datos, obtener listados impresos, generar pequeños programas de mantenimiento de la BD, o ser utilizado como servidor de datos para programas más complejos realizados en cualquier lenguaje de programación. Además, ofrece otras herramientas más propias de la gestión de base de datos como sistemas de permisos para autorización de accesos, volcados de seguridad, transferencia de ficheros, recuperación de información dañada, etc. [9].

Los SGBD permiten el acceso directo a la información y son capaces de manipularla, ya que funcionan como una interfaz entre el usuario, la base de datos y las aplicaciones que la utilizan. Éstos brindan un independiente y eficiente acceso a los datos reduciendo el tiempo de ejecución e integridad y seguridad de los mismos.

Además, se puede definir como una base de datos la colección ordenada de datos, que normalmente se almacena en uno o más archivos asociados. Éstos están estructurados en forma de tablas, donde las referencias cruzadas entre las tablas son posibles.

Se puede concluir que una base de datos no es más que un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su uso posterior.

Las bases de datos ofrecen una forma más eficiente para almacenar datos. Las características incluyen el control de redundancia de datos que está reduciendo el riesgo de incompatibilidad, así como la posibilidad de compartir información. Este último aspecto es muy importante porque los SGBD ofrecen acceso multiusuario. En general, la accesibilidad de los datos se mejora a través de la consulta por ejemplo, a través del lenguaje estructurado de Consulta (SQL). Tanto la manipulación de datos, procesos como la actualización y eliminación son mucho más fácil de lograr. A pesar de tener algunos inconvenientes como los costos y la alta complejidad, las ventajas en la información geográfica pueden llegar a ser preponderantes.

Algunos ejemplos de sistemas gestores de bases de datos relacionales son: MySQL, PostgreSQL, Post Oracle, el servidor de Microsoft SQL y DB2 de IBM. En la presente investigación se prestará mayor atención a gestores de bases de datos espaciales.

1.4.1. Bases de datos espaciales

Las bases de datos que tienen agregada una tercera dimensión se denominan Bases de datos espaciales en tercera dimensión. Por consiguiente, un SGBD espacial debe ofrecer una aplicación central de almacenamiento de información geográfica en tercera dimensión. Los requisitos para que una base de datos sea espacial son:

- Ofrecer tipos de datos espaciales y su aplicación.
- Proporcionar un lenguaje de consulta correspondiente.
- Crear índices espaciales.
- Transformar análisis espacial.

Una Base de Datos Espacial permite describir los objetos espaciales que la forman a través de tres características básicas: atributos, localización y topología. Los atributos representan características de los objetos que permiten saber qué son.

La localización, representada por la geometría del objeto y su ubicación espacial de acuerdo a un sistema de referencia, permite saber dónde está y qué espacio ocupa el objeto. Por último, la topología definida

por medio de las relaciones conceptuales y espaciales entre los objetos, permite mejorar la interpretación semántica del contexto y establecer ciertas jerarquías de elementos a través de sus relaciones [10].

Para mirar más de cerca los datos espaciales, la información geográfica debe ser dividida en dos partes, sus propios datos geográficos y sus atributos asociados. Como se mencionó antes, los aspectos de la geometría 3D y la topología 3D se deben tener en cuenta al tratar con datos espaciales tridimensionales. Ésto es una necesidad porque la funcionalidad espacial requiere fuertemente de ellos como base para los cálculos. En comparación con la segunda dimensión, los datos tridimensionales son mucho más complejos.

Un requisito fundamental es la disposición de un lenguaje de consultas asociadas. Dado que SQL es un estándar de consulta general, no hay ninguna característica para realizar consultas espaciales y un lenguaje de consulta espacial debe proporcionar operaciones espaciales fundamentales y formas razonables de representar el resultado. Las normas para el acceso, consultar la base de datos y mostrar los resultados son importantes. Open GIS Consortium ha puesto en marcha la “Especificación de funciones simples para SQL” que propone tipos de datos espaciales y las operaciones.

Hoy en día, las implementaciones de bases de datos espaciales tienen una amplia difusión. Casi todos los proveedores de base de datos proporcionan extensiones espaciales con el fin de penetrar en el mercado de los SIG. Sin embargo, Stoter y Zlatanova plantean que las implementaciones actuales aún no son compatibles con la tercera dimensión [11].

1.4.2. Gestores de bases de datos espaciales

PostgreSQL y PostGIS

PostgreSQL: Es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado de varias formas desde la década de 1980. Sigue, actualmente, un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto. Es ampliamente considerado como una de las alternativas de sistema de bases de datos de código abierto.

Dentro de sus principales características están [12]:

- Cumple completamente con ACID.

- Cumple con ANSI SQL.
- Integridad referencial la cual es utilizada para garantizar la validez de los datos de la base de datos.
- Replicación (soluciones comerciales y no comerciales) que permiten la duplicación de bases de datos maestras en múltiples sitios de réplica.
- Interfaces nativas para ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python y Ruby Reglas Vistas Triggers.
- Múltiples tipos de datos predefinidos, como todos los manejadores de bases de datos, implementa los tipos de datos definidos para el estándar SQL3 y aumenta algunos otros.
- Soporte de tipos y funciones de usuario, PostgreSQL soporta operadores, funciones métodos de acceso y tipos de datos definidos por el usuario.

La forma tradicional de almacenar información espacial digitalmente, es utilizar algún formato de los programas SIG “de escritorio” como ArcView, MapInfo o GRASS. En general, estos formatos son poco flexibles y bastante anticuados, ya que son un conjunto de archivos que almacenan la información sobre las coordenadas, proyección espacial y los datos asociados a cada elemento.

El formato más común utiliza archivos con extensión DBF para almacenar los datos, formato que presenta muchas limitaciones que parecen totalmente arbitrarias en el día de hoy como por ejemplo nombres de campos limitados a 10 caracteres, 3 ó 4 tipos de datos, entre otras. En cambio, PostGIS (y en general cualquier extensión espacial a un RDBMS) soluciona estos problemas y además, permite una mucho mayor flexibilidad porque hace posible realizar operaciones espaciales en la fuente de datos misma, lo que conlleva varias ventajas, entre ellas [\[12\]](#):

- Facilita el trabajo del programador, puesto que evita el tener que implementar esas operaciones en la aplicación (intersección, búsqueda por cajas, proyección geográfica de los datos, etc.). Además se puede optar por desarrollar partes de la lógica de la aplicación vía procedimientos almacenados, o generar nuevos conjuntos de datos a partir de los existentes de manera mucho más fácil a través de vistas, subqueries, joins o tablas temporales.

- Permite un mayor control sobre la aplicación, al separar los datos del lugar donde se encuentra la aplicación (y no tener necesariamente que compartir archivos vía un sistema de archivos distribuido si queremos separar los datos de la aplicación). Este control presenta beneficios en varios niveles, como el de poder distribuir la carga hacia el RDBMS, o a la aplicación, según el tipo de aplicación y sus características, o correr en máquinas distintas el servidor espacial (de haberlo) y el RDBMS.

PostGIS: Es una extensión al sistema de base de datos objeto-relacional PostgreSQL. Permite el uso de objetos SIG. PostGIS incluye soporte para índices TSIG basados en R-Tree, y funciones básicas para el análisis de objetos SIG. Está creado por Refractor Research Inc, como un proyecto de investigación de tecnologías de bases de datos espaciales. Está publicado bajo licencia GNU. Con PostGIS se puede usar todos los objetos que aparecen en la especificación OpenGIS como puntos, líneas, polígonos, multilíneas, multipuntos, y colecciones geométricas. Las funciones de PostGIS se enuncian a continuación [13]:

- Soporte de geometrías como: puntos, líneas, polígonos, multipuntos, multilíneas, multipolígonos colección de geometrías.
- Permite determinar las interacciones de las geometrías usando la matrix de Egenhofer
- Posee operadores espaciales que permiten determinar medidas geoespaciales como área, distancia, longitud y perímetro.
- Permite llevar a cabo álgebra de vectores como unión, diferencia, diferencia simétrica, función de buffer, generalización y otras funciones geométricas.
- Permite búsquedas espaciales y no-espaciales.
- Permite almacenamiento de imágenes raster en el formato WKTraster.

Ventajas:

- Instalación limitada.
- Soporte.

- Ahorros considerables en costos de producción.
- Escalabilidad y confiabilidad legendarias.
- Extensible.
- Multiplataforma.
- Diseñado para ambientes de altos volúmenes.
- Herramientas gráficas de diseño y de administración de BD.

Desventajas:

- Limitaciones al escribir funciones y procedimientos en comparación con Oracle's PL/SQL o Sybase's T-SQL.
- Las tablas espaciosas, tablas particionadas, y con bloqueo altamente complicado siguen siendo ofrecidas por los vendedores propietarios de bases de datos.
- Carencia de herramientas de desarrollo propia.

Oracle: Es un sistema gestor de bases de datos relacional (de sus siglas en inglés RDBMS) desarrollado por Oracle Corporation. Es un manejador de base de datos relacional que hace uso de los recursos del sistema informático en todas las arquitecturas de hardware, para garantizar su aprovechamiento al máximo en ambientes cargados de información. Es el conjunto de datos que proporciona la capacidad de almacenar y acudir a éstos de forma recurrente con un modelo definido como relacional. Además es una suite de productos que ofrece una gran variedad de herramientas. Oracle corre en computadoras personales (PC), microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo. Soporta unos 17 idiomas, corre automáticamente en más de 80 arquitecturas de hardware y software distinto sin tener la necesidad de cambiar una sola línea de código. Ésto es porque más el 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de sistemas operativos.

Sus principales características son las siguientes [14]:

- Entorno Cliente/Servidor.
- Gestión de grandes bases de datos.
- Usuarios concurrentes.
- Alto rendimiento en transacciones.
- Sistemas de alta disponibilidad.
- Adaptación a estándares de la industria, como SQL-92.
- Gestión de la seguridad.
- Autogestión de la integridad de los datos.
- Opción distribuida.
- Portabilidad.
- Compatibilidad.
- Replicación de entornos.

Oracle es el mayor y más usado Sistema Manejador de base de datos relacional (SDBMR) del mundo pero al igual que otros sistemas de este tipo tiene ventajas e inconvenientes que se mostraran a continuación.

Ventajas [14]:

- Puede ejecutarse en todas las plataformas, desde una Pc hasta un supercomputador.
- Oracle soporta todas las funciones que se esperan de un servidor "serio": un lenguaje de diseño de bases de datos muy completo (PL/SQL) que permite implementar diseños "activos",

con triggers y procedimientos almacenados, con una integridad referencial declarativa bastante potente.

- Permite el uso de particiones para la mejora de la eficiencia, de replicación e incluso ciertas versiones admiten la administración de bases de datos distribuidas.
- El software del servidor puede ejecutarse en multitud de sistemas operativos.
- Existe incluso una versión personal para Windows 9x, lo cual es un punto a favor para los desarrolladores que se llevan trabajo a casa.
- Ha evolucionado añadiendo tipos de clases, referencias, tablas anidadas, matrices y otras estructuras de datos complejas.
- Oracle es la base de datos con más orientación hacia INTERNET
- Un aceptable soporte.

Desventajas [14]:

- Una de las versiones más recientes de Oracle es la 8 punto algo (Aunque ya está la 9i). Y es que desde el lanzamiento original de la 8 se sucedieron varias versiones con correcciones, hasta alcanzar la estabilidad en la 8.0.3. El motivo de tantos fallos fue, al parecer, la remodelación del sistema de almacenamiento por causa de la introducción de extensiones orientadas a objetos.
- El mayor inconveniente de Oracle es quizás su precio. Incluso las licencias de Personal Oracle son excesivamente caras, en mi opinión. Otro problema es la necesidad de ajustes. Un error frecuente consiste en pensar que basta instalar el Oracle en un servidor y enchufar directamente las aplicaciones clientes. Un Oracle mal configurado puede ser desesperantemente lento.
- También es elevado el coste de la formación, y sólo últimamente han comenzado a aparecer buenos libros sobre asuntos técnicos distintos de la simple instalación y administración.

1.5. Estándares de la OGC

Para que un sistema de cómputo pueda simular algún aspecto de la realidad, es necesario que atraviese por un proceso de abstracción en el cual se identifican las diversas entidades existentes y se obtienen sus características con la finalidad de que mediante estos atributos puedan ser representados posteriormente.

Estos atributos deben entonces ser descritos en un lenguaje particular, creado especialmente para que la información se encuentre debidamente estructurada con la finalidad de que pueda almacenarse, compartirse y recuperarse en cualquier parte del mundo.

El Open Geospatial Consortium (OGC), fundado en 1997, tiene como misión lograr una mayor interoperabilidad entre los servicios geográficos. El catálogo de especificaciones OGC está categorizado en dos grupos. Primero, las Especificaciones Resumen OpenGIS, cuyo objetivo es definir modelos conceptuales a fin de especificar las interfaces de aplicación. Estas especificaciones a su vez están divididas en 16 temas. En segundo lugar, las Especificaciones Aplicación OpenGIS se centran en las especificaciones de software que están adoptando las especificaciones resumen OpenGIS.

La tabla 1 muestra una visión general sobre la investigación relacionada con OpenGIS especificaciones de implementación que ya han sido aceptadas por el comité técnico de la OGC [15].

| Nombre de Volumen | Explicación |
|---|---|
| Web Map Service (WMS) | 1: Define la interfaz para solicitar los mapas de un servidor. 2: El la mirada central de XML para implementaciones de servicios de mapas. |
| Web Feature Service (WFS) | Definición de una interfaz web para insertar, borrar y manipular los datos. |
| Web Coverage Service (WCS) | Servicio de acceso a las imágenes de tramas. |
| Web Terrain Service (WTS) | Ampliar el WMS con funcionalidad 3D como modelos de elevación del terreno, texturas y definiciones punto de vista - basados en tramas. |
| Simple Feature Specification for SQL | Define un esquema SQL para interactuar con las características simples almacenadas en una base de datos. |
| Geographic Markup Language (GML) | Especificación de cambio basando en XML el formato de los datos. |

Tabla 1: Estándares de OGC

GML:

El lenguaje de Marcado Geográfico (GML) es un estándar internacional para el intercambio y almacenamiento de datos geográficos. Ofrece una amplia variedad de objetos para describir la geografía incluyendo entidades, sistemas de coordenadas, geometría, topología, tiempo, unidades de medida y valores generalizados. GML depende de XML como medio de descripción fundamental de los datos utilizados. Requiere de un esquema XML para la definición de la gramática GML, UML para expresar los diagramas que muestran las relaciones de los objetos que lo conforman. Espacios de nombre en XML: para diferenciar los nombres de las etiquetas de diferentes esquemas [16].

Ventajas [16]:

- Verificación automática de la integridad de los datos.
- Puede ser leído por herramientas públicas o genéricas.
- Puede ser fácilmente editado.
- Puede ser integrado fácilmente con datos No-Espaciales. Los formatos binarios dificultan este proceso porque se necesita conocer las estructuras y ser capaces de modificarlos.
- Mapas de mayor calidad.
- Funcionan en los navegadores sin la necesidad de comprar software en el lado del cliente.
- Mejores capacidades de consulta frente a los mapas raster.

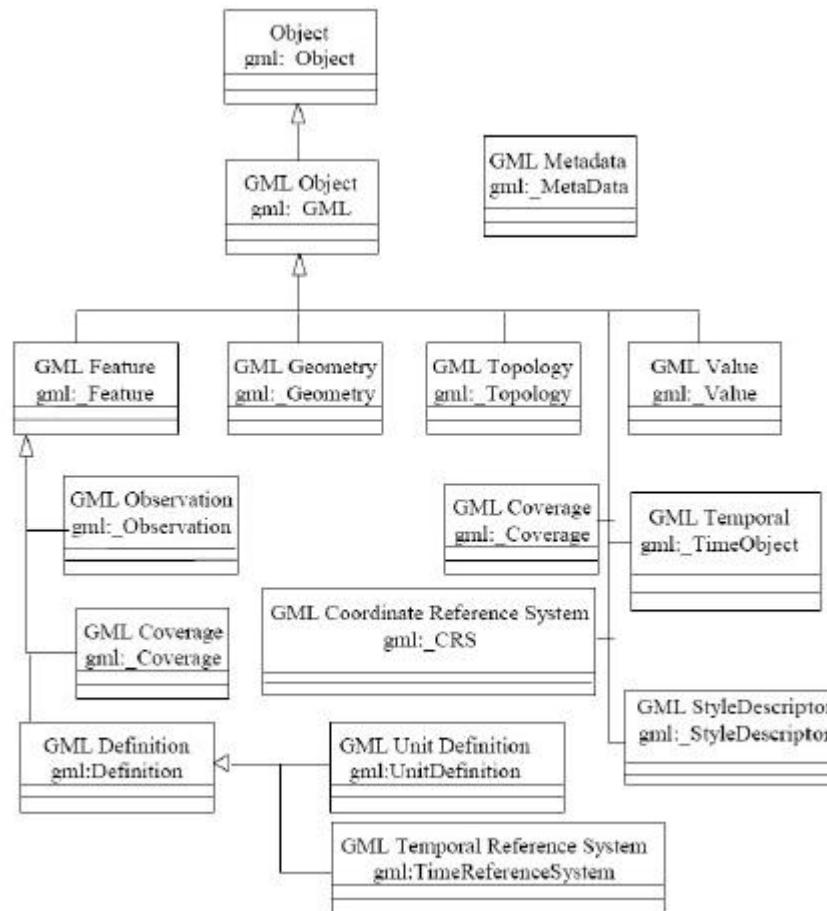


Figura 8: Jerarquía de las clases GML

CityGML:

CityGML es un modelo de información para la representación de los objetos urbanos 3D. Define las clases y las relaciones para los objetos topográficos más relevantes de ciudades y los modelos regionales con respecto a sus características geométricas, topológicas, semánticas y del aspecto. Incluye las jerarquías de la generalización entre las clases temáticas, las agregaciones, las relaciones entre los objetos, y las características espaciales. Esta información temática va más allá de formatos gráficos del intercambio y

permite emplear modelos virtuales de ciudad 3D para el análisis sofisticado las tareas en diversos dominios [17].

Una de las características más importantes de CityGML es su capacidad para contener datos vectoriales 3D complejos y georreferenciados junto a la semántica asociada a esos datos. En contraste con otros formatos 3D, CityGML se basa en modelo de información rico y de propósito general que va más allá del contenido gráfico y geométrico. Para trabajar en dominios específicos, CityGML, también proporciona un mecanismo de extensión con el objetivo de enriquecer los datos con características identificables que permitan la interoperabilidad semántica en esos dominios [18].

1.6. Visualización geográfica tridimensional en la Web

La forma de gestionar y organizar los datos respectivamente, como modelar objetos en 3D para el medio urbano, son la base fundamental para una razonable visualización de modelos 3D de ciudad.

El proceso de visualización está dividido en 4 etapas. Primero está el nivel “Data Source” (fuente de datos) que incluye la gestión de funcionalidad y los datos. La segunda etapa es el nivel “Features” (fondo) en este caso los datos han sido pre-ensamblados mediante la ejecución a través de un filtro como una consulta de datos. En tercer lugar se encuentra el nivel “Display Elements” (Mostrar elementos). En esta etapa ya los datos han sido construidos con los requisitos del formato de visualización. Por último el nivel “Image” (Imagen) en el cual los datos se muestran según las características de la pantalla del dispositivo.

Como se puede apreciar, éste es el proceso mediante el cual se lleva a cabo la visualización 3D, a continuación se analizarán una serie de aspectos necesarios para facilitar la comprensión del tema, es decir se conocerá acerca de las Interfaz gráfica de usuario (de sus siglas en ingles GUI) y de algunas tecnologías importante a la hora de llevar a cabo la visualización 3D en la Web.

1.6.1. Interfaz gráfica de usuario (GUI)

Para interactuar y comunicarse con la información, una interfaz gráfica de usuario (GUI) tiene que ser diseñada y creada. La información geográfica es generalmente muy compleja, y difícil de alcanzar. Por otra parte, la interfaz de usuario es lo más crítico debido a que es la puerta principal de la aplicación, si ésta es implementada pobremente la aplicación no será aceptada por los usuarios o no estará en contraste con la interacción del usuario [19].

Desarrollar una GUI para la visualización 3D, tiene diversos aspectos importantes. Para lograr ésto, son necesarias las características de base para crear un mundo 3D. Lo cual significa modelado apropiado de los objetos físicos, iluminación y sombreado, definición de puntos de vista, texturas. Tan pronto como la interacción tenga que estar implicada con eventos, el script interno y externo llega a ser indispensable [20].

Una GUI se encuentra en la parte superior de la aplicación del usuario. Dado que la información geográfica suele ser muy compleja, esta tarea es difícil de lograr, tiene que ofrecer un acceso eficaz a la lógica de aplicación y por último los datos. Por otra parte, la interfaz de usuario es más crítica debido al hecho de que ésta es la "puerta principal" de la aplicación [15].

Si una interfaz gráfica de usuario se ejecuta de forma incorrecta, la solicitud no será aceptada por el usuario. En contraste con la interacción del usuario en 2D, una interfaz gráfica para la tercera dimensión tiene que ser diferente. Ésta es la razón, porque en la mayoría de los casos los usuarios tienen una imaginación pobre asociadas con las representaciones de múltiples dimensiones [20].

1.6.2. VRML/X3D

VRML y su respectivo sucesor **X3D** fueron presentados por el consorcio Web3D para distribuir los mundos virtuales en la web. Ambos son lenguajes de marcado y estandarizado. X3D es el cumplimiento de los conceptos XML, además se especifica más modular. VRML y X3D están logrando los conceptos fundamentales de una interfaz gráfica de usuario tridimensional [21]. Conceptos de la construcción de un mundo virtual de núcleo y, especialmente, la interfaz de edición externa (EAI) en torno a las posibilidades de clasificación X3D/VRML. Mediante el uso de la EIA, se puede agregar la funcionalidad individual a los mundos virtuales. Producido, por secuencias de comandos o más lenguajes de programación, las escenas en 3D pueden llegar a ser muy interactivas.

Un buen ejemplo es el acceso a una base de datos de los mundos VRML con el fin de recuperar los datos nuevos. Otra prueba de que VRML/X3D son capaces de manejar tareas complejas que se muestra por Diehl. Sus implementaciones distribuidas muestran mundos virtuales, por ejemplo mundos multiusuario [22].

Cientes VRML en combinación con HTML ya han demostrado su capacidad para reaccionar como los agentes de usuario SIG en muchos ejemplos y prototipos. Sin embargo, la aplicación comercial disponible no es bien conocida. El uso más común de VRML está dentro de un navegador en el cliente / plug-in de la aplicación. Desafortunadamente vendedores de plug-in dudan con el envío navegadores X3D

1.6.3. JAVA3D

Java es un lenguaje interpretado, pre-compilado de forma que pueda ser usado en distintas plataformas. Para ejecutar cualquier aplicación java se requiere de otra aplicación que traduzca el código pre-compilado a código entendible por la maquina en la cual se ejecuta, a esto se la llama Java Virtual Machine (JVM).

Java 3D es una herramienta para crear mundos tridimensionales en la Web. Posee un alto nivel de subsistemas de gráficos 3D [23]. La biblioteca Java 3D es una libre disposición de la API para el desarrollo de mundos virtuales en Java [24] . Por lo tanto las clases de Java 3D pueden ser usadas por Applets de Java en páginas HTML. Las funcionalidades de Java son casi las mismas que están proporcionando VRML/X3D.

Una gran ventaja en comparación con plug-in de soluciones basadas es que los desarrolladores tienen más control sobre la representación e interacción del usuario. Otra es la transformabilidad. Compilando clases Java 3D puede ser utilizado como aplicación independiente o applet. En contraste con los lenguajes de marcado de VRML o X3D, Java3D requiere mucho más conocimiento de programación [22].

Ésta es probablemente una de las razones, por la que sólo algunas soluciones se han realizado utilizando Java 3D.

1.7. Lenguajes de programación Web

PHP

PHP es la sigla de **Hypertext Preprocessor** (Procesador de Hipertextos) es generalmente muy usado por su flexibilidad. Es un lenguaje de código abierto de scripting (escritura) especialmente adecuado para el desarrollo de web.

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Además, permite la conexión a diferentes tipos de servidores de base de datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird, SQLite logrando aplicaciones web robustas y sus archivos cuentan con la extensión (.php) [25] .

Ventajas [25]:

- Es identificado como un lenguaje muy rápido y manuable.
- Soporta de cierta forma la orientación a objeto. Clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos.
- Puede expandir su capacidad de potencial utilizando módulos.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos y su código está disponible bajo la licencia GPL.
- Posee una biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Desventajas [25]:

- Se necesita instalar un servidor WEB para la ejecución del código.

- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto, puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.

Java Script

Es un lenguaje interpretado que está del lado del cliente porque es el navegador el que soporta la carga de procesamiento. Se utiliza integrado en un navegador WEB permitiendo el desarrollo de interfaces de usuario mejoradas y páginas WEB dinámicas.

Este lenguaje es utilizado para acceder a objetos en las aplicaciones y gracias a su compatibilidad con la mayoría de los navegadores modernos es el más utilizado por parte de los clientes. JavaScript posee una sintaxis similar a la del lenguaje Java y el lenguaje C, aunque no es orientado a objeto está destinado al desarrollo de aplicaciones WEB como complemento del HTML. También puede realizar efectos especiales sobre páginas WEB para crear contenidos dinámicos y elementos de la página que tengan movimiento. Además, permite que se ejecuten instrucciones como respuesta a las acciones del usuario, con lo que se crean páginas interactivas [26].

Ventajas [26]:

- El código JavaScript es gestionado en el navegador.
- Los programas JavaScript tienden a ser pequeños y compactos, no requieren mucha memoria ni tiempo adicional de transmisión.
- Funciona en cualquier navegador que soporte JavaScript independientemente de la plataforma o el sistema operativo.
- Es un lenguaje de scripting seguro y fiable.

Desventajas [26]:

- El código es visible por cualquier usuario.

- El código debe descargarse completamente.
- Puede poner en riesgo la seguridad del sitio, con el actual problema llamado XSS (Cross Site Scripting) renombrado a XSS por su similitud con las hojas de estilo.

1.8. Estándares de programación Web

XML

Extensible Markup Language es una tecnología muy sencilla que tiene a su alrededor otras tecnologías que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. XML tiene maneras más fáciles e interesantes de trabajar con los datos y, en definitiva, un avance a la hora de tratar la información, que es en realidad el objetivo de la computación en general. XML no es un lenguaje, sino varios lenguajes, no es una sintaxis, sino varias y no es una manera totalmente nueva de trabajar, sino una manera más refinada que permitirá que todos los lenguajes anteriores se puedan comunicar entre sí sin problemas, ya que los datos cobran sentido [12].

XSLT

El lenguaje de hojas de estilo de transformación (XSLT o XSL Transformation) es la parte más importante del estándar XSL. Proporciona los elementos que definen las reglas de cómo un documento XML se transforma en otro XML, HTML o documento de texto. Una vez transformado el documento se pueden utilizar dispositivos de marcado y DTD del documento original, o puede utilizar los documentos de conjunto completamente diferentes. XSLT puede adicionar elementos dentro de un fichero de salida, remover elementos existentes, reorganizar y ordenar los elementos, prueba y toma decisiones, a través de hojas de estilos apropiados [27].

Una **hoja de estilo** XSLT es básicamente un conjunto de reglas expresadas en Lenguajes de hojas de estilo extensible para la transformación de documentos XML. En el caso de la visualización de datos, el rol de las hojas de estilo XSLT es transformar los datos XML contenidos dentro de un formato de presentación como: HTML, XHTML, SVG, textos o cualquier otro formato estructurado. Pero en el contexto de los SIG, los datos XML que se encuentran en GML tiene que ser presentado en un formato

gráfico como gráficos vectoriales escalables que es un lenguaje basado en XML para describir gráficos 2D [27].

Ventajas:

- No asume un único formato de salida de documentos.
- Permite manipular de muy diversas maneras un documento XML: reordenar elementos, filtrar, añadir, borrar, etc.
- Permite acceder a todo el documento XML.
- XSLT es un lenguaje XML.

Desventajas:

- Su utilización es más compleja que un lenguaje de programación convencional.
- Consume cierta memoria y capacidad de proceso → DOM detrás.

Ajax

Ajax es el acrónimo de **Asynchronous JavaScript And XML** (JavaScript y XML asíncronos). Ajax no es una tecnología porque es la unión de un conjunto de tecnologías para el desarrollo WEB que se ejecutan en el cliente, es decir, en el navegador de los usuarios y mantiene comunicación asíncrona con el servidor en segundo plano.

Ajax tiene una presentación basada en estándares usando XHTML y CSS. Además realiza una interacción dinámica usando el DOM y realiza el intercambio y manipulación de datos usando XML. Esta tecnología posee recuperación de datos asíncrona usando XMLHttpRequest. Los datos adicionales que carga del servidor no sobrecargan la página ni afecta su comportamiento porque no interfiere con la visualización de la misma. También Ajax se ha extendido a la manipulación de entornos 3D en la WEB con Ajax3D [28].

Ventajas [29]:

- Basado en estándares abiertos.
- Usabilidad.
- Válido en cualquier plataforma y navegador.
- Beneficia las aplicaciones web.
- No es difícil su utilización.
- Compatible con Flash.
- Adoptado por los “gordos” de la tecnología web.
- Web 2.0.
- Es independiente del tipo de tecnología de servidor que se utilice.
- Mejora la estética de la web.

1.9. ArcGIS

ArcGIS: es un software GIS para visualizar, crear, manipular y gestionar información geográfica, éstos corresponden a lugares, direcciones, posiciones en terreno, áreas urbanas y rurales; regiones y cualquier tipo de ubicaciones en terrenos determinados.

Esta información es trabajada de manera sistémica, lo que representa una diferencia sustancial a lo relacionado al trabajo con información planos y mapas, permitiéndonos explorar, ver y analizar los datos según parámetros, relaciones y tendencias que presenta nuestra información, teniendo como resultado nuevas capas de información, mapas y nuevas bases de datos.

ArcScene: la aplicación de visualización tridimensional que forma parte de la extensión ArcGIS 3D Analyst, permite crear los mostradores tradicionales y no convencionales en 3D a partir de datos del mundo real. La figura creada a partir de la elevación y valores de profundidad se utilizan para revelar la

superficie de la tierra y exponer el interior. Por otra parte, las medidas para que otros puedan representar una tercera dimensión de los datos científicos de tierra.

1.10. Selección de tecnologías para desarrollar la propuesta

Luego de hacer un análisis de las tecnologías más importantes usadas para el desarrollo de aplicaciones web en el mundo, a continuación se mencionan las tecnologías seleccionadas para desarrollar la propuesta: como lenguaje del lado del servidor utilizaremos PHP, para la interacción de los modelos 3D en la web se utilizará X3D, como servidor WEB se escogió Apache, como gestor de base de datos se puede utilizar PostgreSQL, su versión espacial PostGIS, con el empleo de PDO y como entornos de desarrollo integrado (IDE), Aptana Studio para el trabajo con HTML, JavaScript, PHP y Ajax; y Vivaty Studio para la visualización en el navegador. Por último se seleccionó arquitectura 3 capas.

Conclusiones del capítulo

En este capítulo se abordan los conceptos fundamentales que facilitan la comprensión de todos los aspectos referentes a visualización de información geográfica tridimensional en la Web pero enfocada en los entornos urbanos, dejando entrever las posibilidades y ventajas que tiene la visualización tridimensional de entornos urbanos.

También se hace un exhaustivo análisis de algunas herramientas, lenguajes de programación web y otras tecnologías existentes que se pueden aplicar a un sistema de este tipo y se analizaron las posibilidades de las mismas. Se realizó una selección de tecnologías y herramientas de programación web para lograr desarrollar la propuesta.

Introducción

En este capítulo se realizará la descripción de la solución propuesta para dar respuesta al problema planteado, los aspectos más importantes a tener en cuenta son: la arquitectura del sistema, los pasos a seguir para darle solución al problema, así como el proceso de transformación de los datos y las características del sistema propuesto.

2.1 Arquitectura

La propuesta cuenta con una arquitectura en 3 capas que es un estilo de programación que tiene como objetivo primordial la separación de la capa de presentación, capa de negocio y la capa de datos.

Capa de presentación: En esta capa es donde el usuario interactúa, le comunica la información y captura de la información del usuario en un mínimo proceso. Esta capa se comunica únicamente con la capa de negocio. También puede ser nombrada como interfaz gráfica y debe tener la característica de ser “amigable” para el usuario.

Capa lógica: Es donde, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se le llama capa de negocio porque es donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir solicitudes y presentar resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos.

Capa de datos: Es donde residen los datos y se encargada de acceder a los mismos. Está formada por uno o más gestores de base de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de la información desde la capa lógica.

A continuación se dará una vista de la arquitectura (ver figura 9) y una descripción de la funciones que tienen lugar en cada capa.

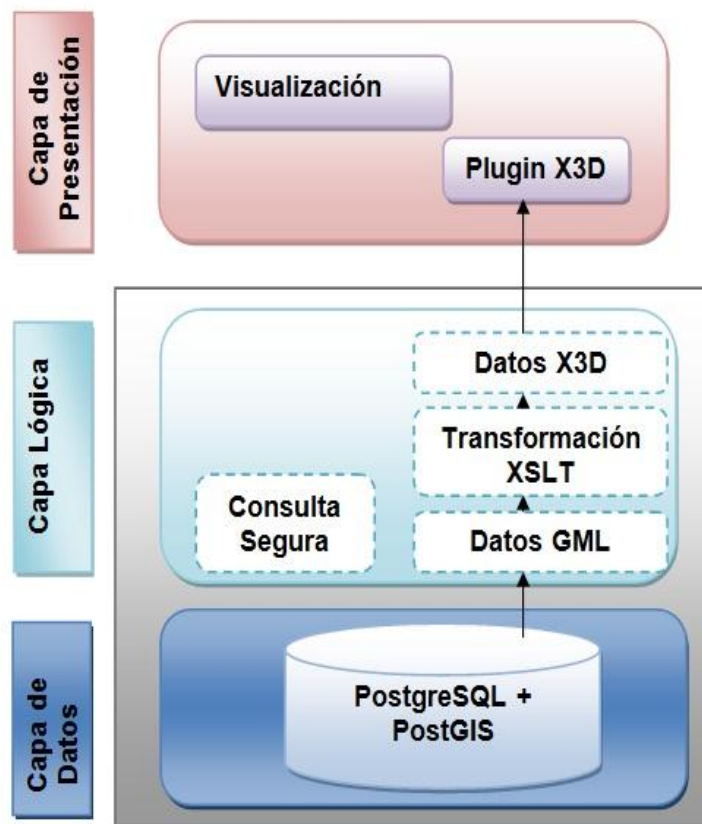


Figura 9: Arquitectura propuesta

2.1.1 Capa de datos

Lograr la visualización de los datos en forma tridimensional en la web constituye un proceso largo y complejo, pues requiere de varios aspectos para efectuarse. Primero, se necesita una fuente de donde se van a adquirir los datos que se adicionaran posteriormente a la base de datos. Estos se van a obtener a

partir a un archivo en formato shape. Los shape son los tipos de geometría que pueden tener un shapefile, cada shapefile estará asociado a un objeto geométrico el cual puede ser cualquiera de los valores que se muestra en el tabla.

| Valor | Tipo de shape |
|-------|---------------|
| 0 | Shape nulo |
| 1 | Punto |
| 3 | PoliLinea |
| 5 | Polígono |
| 8 | MultiPunto |
| 11 | PuntoZ |
| 13 | PoliLineaZ |
| 15 | PolígonoZ |
| 18 | MultiPuntoZ |
| 21 | PuntoM |
| 23 | PoliLineaM |
| 25 | PolígonoM |
| 28 | MultiPuntoM |
| 31 | MultiPatch |

Tabla 2: Tipos de Shape que existen en ArcView

La geometría de estos archivos se obtienen usando el módulo ArcScene de ArcGIS y luego los datos se importan a la base de datos (PostgreSQL/PostGIS) que soporta los tipos geométricos siguientes:

- POINT(0 0 0) -- XYZ
- SRID=32632;POINT(0 0) -- XY with SRID
- POINTM(0 0 0) -- XYM
- POINT(0 0 0 0) -- XYZM
- SRID=4326;MULTIPOINTM(0 0 0,1 2 1) -- XYM with SRID
- MULTILINESTRING((0 0 0,1 1 0,1 2 1),(2 3 1,3 2 1,5 4 1))
- POLYGON((0 0 0,4 0 0,4 4 0,0 4 0,0 0 0),(1 1 0,2 1 0,2 2 0,1 2 0,1 1 0))
- MULTIPOLYGON(((0 0 0,4 0 0,4 4 0,0 4 0,0 0 0),(1 1 0,2 1 0,2 2 0,1 2 0,1 1 0)),((-1 -1 0,-1 -2 0,-2 -2 0,-2 -1 0,-1 -1 0)))
- GEOMETRYCOLLECTIONM(POINTM(2 3 9), LINESTRINGM(2 3 4, 3 4 5))

Se puede observar que la versión espacial del gestor de base de datos PostgreSQL, PostGIS, soporta los tipos de geometría contenidas en la fuente de datos usada (shapefile). Teniendo la fuente de datos el siguiente paso es importar los datos almacenados en dicha fuente al gestor de base de datos.

Importación de datos

El proceso de importación de los datos puede ocurrir de dos formas; usando SQL o el cargador. El cargador de datos **shp2pgsql**, se conecta directamente con la base de datos (PostgreSQL/PostGIS) y convierte un archivo de figuras en una tabla.

El cargador tiene varios modos de operación que se seleccionan con los parámetros desde línea de comando:

- **-d** Elimina la tabla de la base de datos antes de crear la tabla con los datos del archivo de figuras.
- **-a** Añade los datos del archivo de figuras a las tabla de la base de datos. El fichero debe tener los mismos atributos que la tabla.
- **-c** Crea una nueva tabla y llena ésta con el archivo de figuras. Éste es el modo por defecto.
- **-D** Crea una tabla nueva llenándola con los datos del fichero de formas. Pero usa el formato *dump* para la salida de datos que es más rápido que el *insert* de SQL. Se usará éste para conjuntos de datos largos.
- **-s<SRID>** Crea y rellena una tabla geométrica con el SRID que se le pasa como parámetro.

Ejemplo:

```
shp2pgsql lesoto.shp tablalesoto gisdb >lesotobuilding.sql
```

Tablas del sistema

Para ser compatible con los desarrollos anteriores las tablas de la geobase en openGIS no se alteran, lo único que se agrega es el valor de “3” en el atributo dimensión y en los nuevos tipos de datos. La codificación de la información en 3D sigue siendo binaria, por lo que no hay diferencia con los elementos en 2D. La figura muestra el esquema general de las tablas en el sistema:

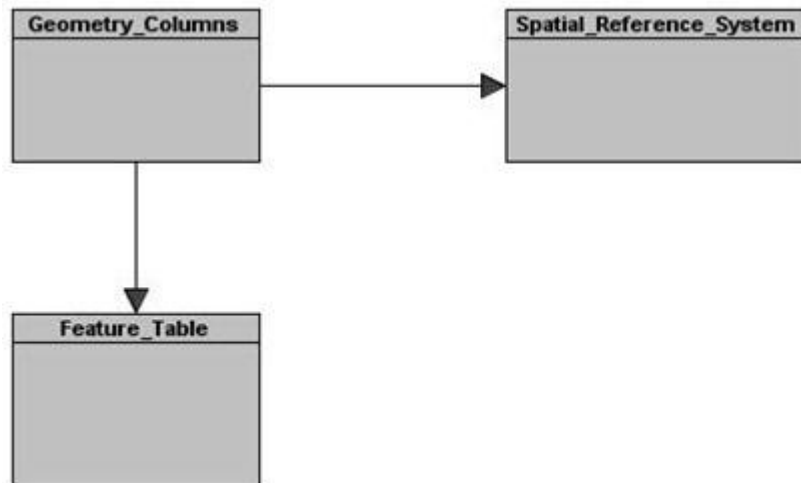


Figura 10: Tablas de la propuesta

A continuación se brinda una descripción de cada tabla y del contenido de sus columnas:

Spatial_Rerefence_Sys: Es una tabla de base de datos incluida en PostGIS compatible con las listas de más de 3.000 sistemas de referencia espacial más conocidos y los detalles necesarios para transformar/reproyectar entre ellos. En conclusión contiene los identificadores numéricos y descripciones textuales de los sistemas de referencia. Las columnas de Spatial_Reference_Sys son las siguientes:

- SRID: Número entero que identifica unívocamente a un sistema de referencia espacial en la base de datos.
- AUTH_NAME: Nombre del estándar para el sistema de referencia.
- AUTH_SRID: Identificador (ID) del sistema de referencia tal y como lo define el estándar en AUTH_NAME.
- SRTEXT: La representación textual conocida del sistema de referencia especial.

| | srid [PK] integ | auth_name character | auth_srid integer | srtext character | proj4text character |
|-----------|------------------------|----------------------------|--------------------------|-------------------------|----------------------------|
| 1 | 2000 | EPSG | 2000 | PROJCS["An | +proj=tmer |
| 2 | 2001 | EPSG | 2001 | PROJCS["An | +proj=tmer |
| 3 | 2002 | EPSG | 2002 | PROJCS["Do | +proj=tmer |
| 4 | 2003 | EPSG | 2003 | PROJCS["Gr | +proj=tmer |
| 5 | 2004 | EPSG | 2004 | PROJCS["Mc | +proj=tmer |
| 6 | 2005 | EPSG | 2005 | PROJCS["St. | +proj=tmer |
| 7 | 2006 | EPSG | 2006 | PROJCS["St. | +proj=tmer |
| 8 | 2007 | EPSG | 2007 | PROJCS["St. | +proj=tmer |
| 9 | 2008 | EPSG | 2008 | PROJCS["NA | +proj=tmer |
| 10 | 2009 | EPSG | 2009 | PROJCS["NA | +proj=tmer |

Figura 11: Spatial_Reference_Sys. Ejemplo

Geometry_Column: Guarda un índice de tablas que contiene algún campo con la geometría. Las columnas de Geometry_Column son las siguientes:

- f_table_catalog, f_table_schema, f_table_name: Nombre completo de la tabla que contiene la geometría.
- f_geometry_column: Nombre del campo que contiene la geometría en la tabla.
- Coord_dimension: Dimensión espacial de la columna (2D o 3D) de la geometría.
- Srid: Identificador del sistema de referencia espacial usada por la geometría en la tabla.
- Type: Tipo de objeto espacial (point, linestring, polygon, multipoint, multilinestring, multipolygon, geometrycollection).

| | oid | f_table_ca [PK] chara | f_table_sc [PK] chara | f_table_na [PK] chara | f_geometr [PK] chara | coord_dir integer | srid integer | type character |
|---|-------|--------------------------|--------------------------|--------------------------|-------------------------|----------------------|-----------------|-------------------|
| 1 | 34725 | " | public | buildings | the_geom | 2 | -1 | MULTIPOLYG |
| 2 | 34767 | " | public | campusw | the_geom | 2 | -1 | MULTIPOLYG |
| 3 | 34781 | " | public | docentes | the_geom | 2 | -1 | MULTIPOLYG |
| 4 | 34795 | " | public | herradura | the_geom | 2 | -1 | MULTIPOLYG |
| 5 | 34753 | " | public | lesoto | the_geom | 2 | -1 | MULTIPOLYG |
| 6 | 18146 | " | public | roads | road_geom | 3 | -1 | GEOMETRY |
| 7 | 18167 | " | public | simpleshap | the_geom | 2 | -1 | MULTIPOLYG |

Figura 12: Geometry_Column. Ejemplo

Feature_table:

Se crea una nueva tabla. Primero se crea una tabla no espacial y luego se añade una columna (campo) espacial a la tabla usando la función *AddGeometryColumn*.

Ejemplo de creación de una tabla:

```
SELECT AddGeometryColumn ('calles_db','calles_geom','geom',423, 'LINESTRING',2)
```

2.1.2 Capa lógica

Como se había mencionado al comenzar este epígrafe, la capa lógica es donde se lleva a cabo todo el proceso referente a las peticiones y las respuestas de éstas, a continuación realizaremos un análisis detallado de cómo se realiza el proceso de transformación de los datos que luego se mostrarán en la capa de presentación.

La transformación de los datos tiene lugar en la capa lógica siguiendo la siguiente secuencia de pasos:

- La petición de los datos se hace a través del lenguaje PHP para convertir parámetros de entrada en una consulta SQL.

- Los datos se convierten a GML a través de la función AsGML que retorna la geometría como un elemento GML, el parámetro versión si se especifica puede ser 2 ó 3, sino se asume el valor por defecto (2).

```
$consulta = "Select asgml (lesoto.the_geom) , heigh from lesoto";
```

- Se escribe en un fichero XML el resultado de la consulta, cumpliendo con el esquema de validación que hace referencia al estándar GML abordado en el capítulo. A continuación se presenta un diagrama UML del esquema de validación para la propuesta:

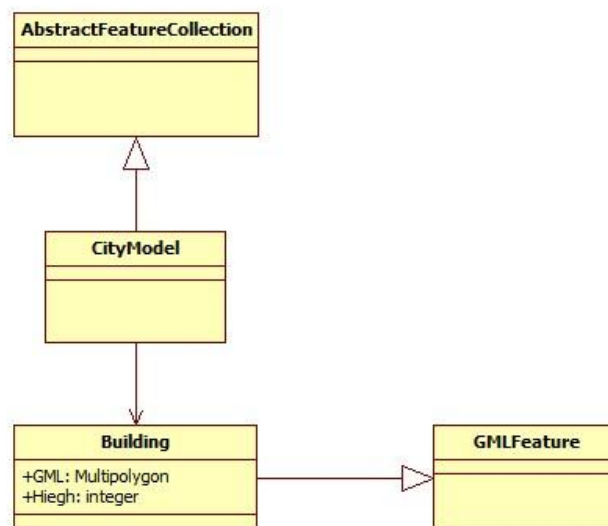


Figura 13: Diagrama UML del esquema de validación

- A los documentos GML, que representan la geometría de un lugar, se le aplica una hoja de estilos XSLT para transformar los datos de GML a X3D. A continuación se muestra una imagen con la transformación:

```

<xsl:element name="Shape">
  <xsl:attribute name="DEF">Shape<xsl:value-of
  select="position()"/></xsl:attribute>
  <xsl:element name="Extrusion" <xsl:attribute
  name="crossSection">
    <xsl:value-of select="$coordinates-list"></xsl:value-of>
  </xsl:attribute>
  <xsl:attribute name="scale">1 1</xsl:attribute>
  <xsl:attribute name="solid">>true</xsl:attribute>
  <xsl:attribute name="spine">0 0 0,0 5 0</xsl:attribute>
  </xsl:element>

  <xsl:element name="Appearance">
    <xsl:element name="Material">
      <xsl:attribute name="diffuseColor">0.2 0.6 0.5</xsl:attribute>
    </xsl:element>
  </xsl:element>
</xsl:element>
</xsl:element>

```

- Se convierten los datos a X3D.

```

<Shape DEF="Shape6">
  <Extrusion containerField="geometry" convex="false" crossSection="85.513 -
  65.490, 86.276 -64.312, 87.997 -65.251, 87.196 -66.465, 85.513 -65.490" scale="1 1 1"
  orientation="0 0 1 0 0 1 0" solid="true" spine="0 0 0 7 0"/>
  <Appearance containerField="appearance">
    <Material DEF="TextMaterial6" containerField="material" emissiveColor="0
  0.250980392156863 0.250980392156863" diffuseColor="0 0.25 0.25"
  transparency="0.25"/>
  </Appearance>
</Shape>

```

- Finalmente con un plug-in instalado en el navegador web se muestran los datos en 3D.

2.1.3 Capa de presentación

A continuación se muestran los pasos a seguir para lograr la visualización de los datos de forma tridimensional y la interacción del usuario con el sistema.

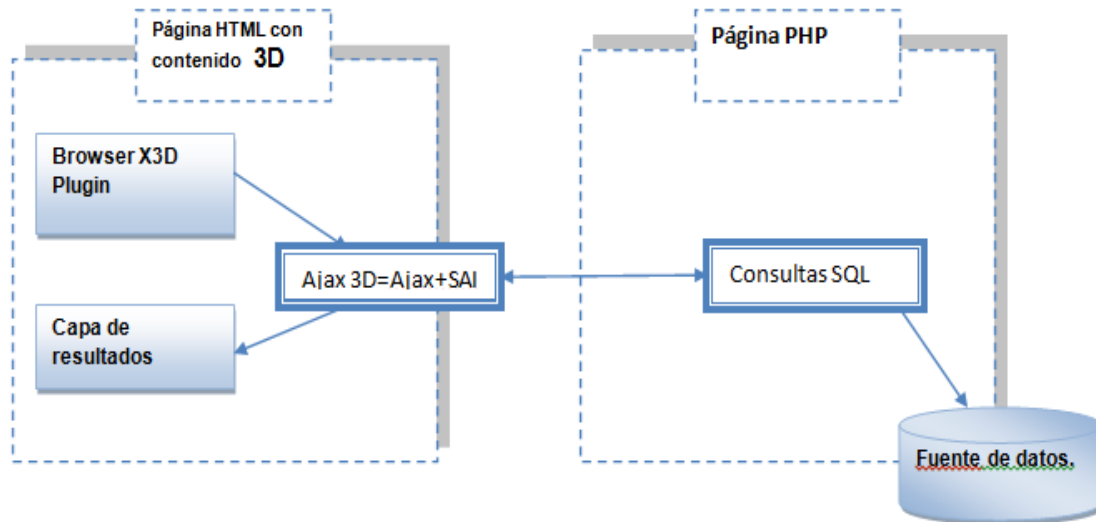


Figura 14: Modelo de la aplicación usando Ajax3D

1. Se define un grupo dentro del modelo .x3d al que se le va a asignar un sensor de contacto para interactuar con el mouse.

```
<Group DEF= 'GrupoModelo'>
```

2. Se definen los objetos que van a estar contenidos dentro de ese grupo.

```
<Shape DEF= 'Modelo'>
```

3. Se define el material que va a tener el objeto.

```
<Material DEF="TextMaterial6" containerField="material" emissiveColor="0
0.250980392156863 0.250980392156863" diffuseColor="0 0.25 0.25"
transparency="0.25"/>
```

4. En caso de ya estar definida una textura dentro de la escena, sólo hay que especificar cuál se desea asignar al objeto.

```
<Material> containerField='material' USE= 'TexturaModelo'/>
```

5. Se le asigna al grupo un sensor de contacto.

```
<TouchSensor DEF= 'SensorModelo' containerField= 'children'/>
```

6. Después de cargar el modelo X3D en la aplicación web, por medio de la utilización del Ajax3D, se realiza el cambio de color.
 - Se carga el modelo en la página WEB.
 - Se detecta cualquier interacción en el navegador.
 - En dependencia del edificio donde se realizó el clic se pregunta qué sensor de contacto tiene asignado ese objeto, en correspondencia del que sea, se envía el id del mismo para realizar la consulta parametrizada a la base de datos para mostrar la información asociada.

2.2 Prototipo y descripción del sistema

La implementación de la propuesta para visualizar tridimensionalmente la información geográfica de entornos urbanos en la web cuenta con un fichero "Shape" para obtener su geometría, a través del ArcGIS.

Como gestor de bases de datos se usó la extensión de datos espaciales de PostgreSQL, PostGIS donde se importan los datos geográficos almacenados en el fichero "Shape". Usando el lenguaje de programación web PHP se obtienen desde SGBD los datos geográficos que son especificados por el programador mediante una consulta. El resultado de dicha consulta se devuelve en un fichero XML cumpliendo con las características del estándar de la OGC, GML al que se le aplica una plantilla XSLT para transformar los datos de GML a X3D. Finalmente con un plug-in instalado en el navegador web es posible visualizar los datos geográficos en forma tridimensional.

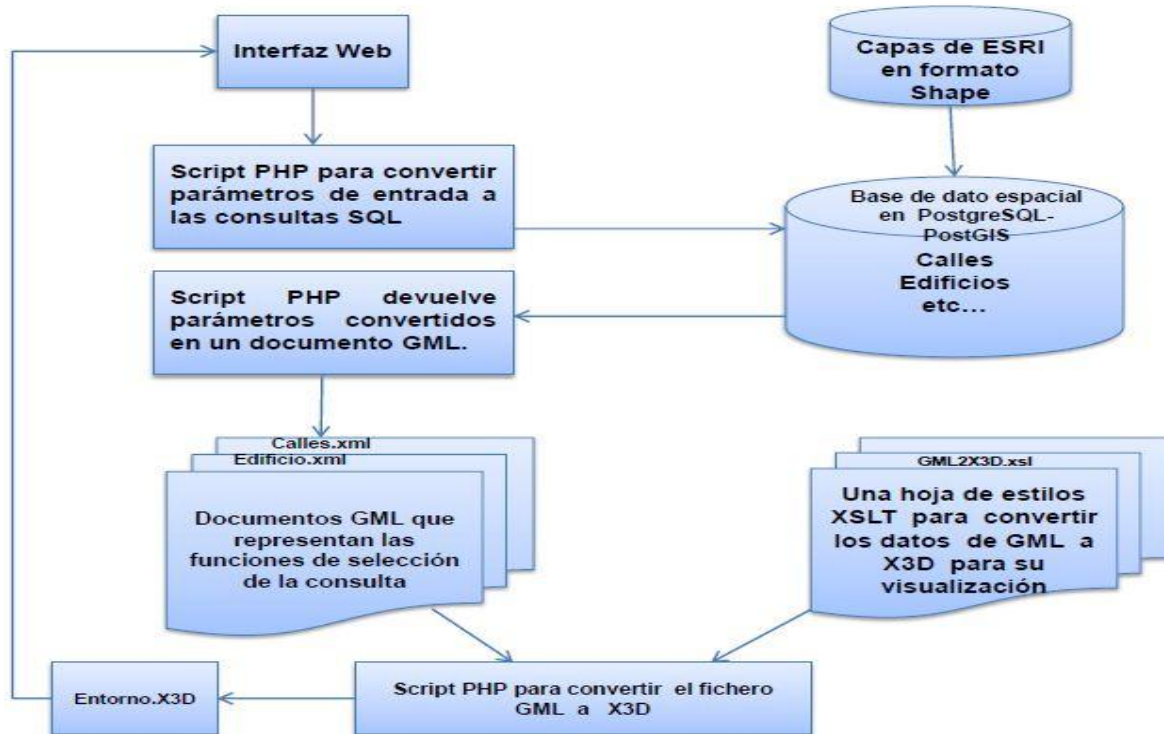


Figura 15: Pasos para llevar a cabo la propuesta de solución

Conclusiones del capítulo

En el presente capítulo se realizó una propuesta de solución a la situación para la visualización de información geográfica tridimensional de entornos urbanos para plataformas web, se dieron en forma detallada una secuencia de pasos a seguir para llevar a cabo la visualización de entornos urbanos en forma tridimensional, además, se explicó de forma general la arquitectura organizacional del sistema y se especificó cómo se lleva a cabo el proceso de transformación de los datos para que se puedan visualizar en forma tridimensional en el cualquier navegador.

ANÁLISIS DE LOS RESULTADOS

Introducción

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, y la evaluación es hecha de algún aspecto del sistema o componente. La prueba de software es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación. A continuación se dará una evaluación de la propuesta realizada atendiendo a los resultados logrados.

3.1 Resultados

Usando la realidad virtual no inmersiva se brinda la posibilidad de interactuar con modelos urbanos los cuales podrán ser rotados brindando diferentes vistas. Además se ofrece información básica relacionada con el entorno representado.

En la figura 16 se muestra la imagen del shape antes de realizar el proceso de transformación de los datos:

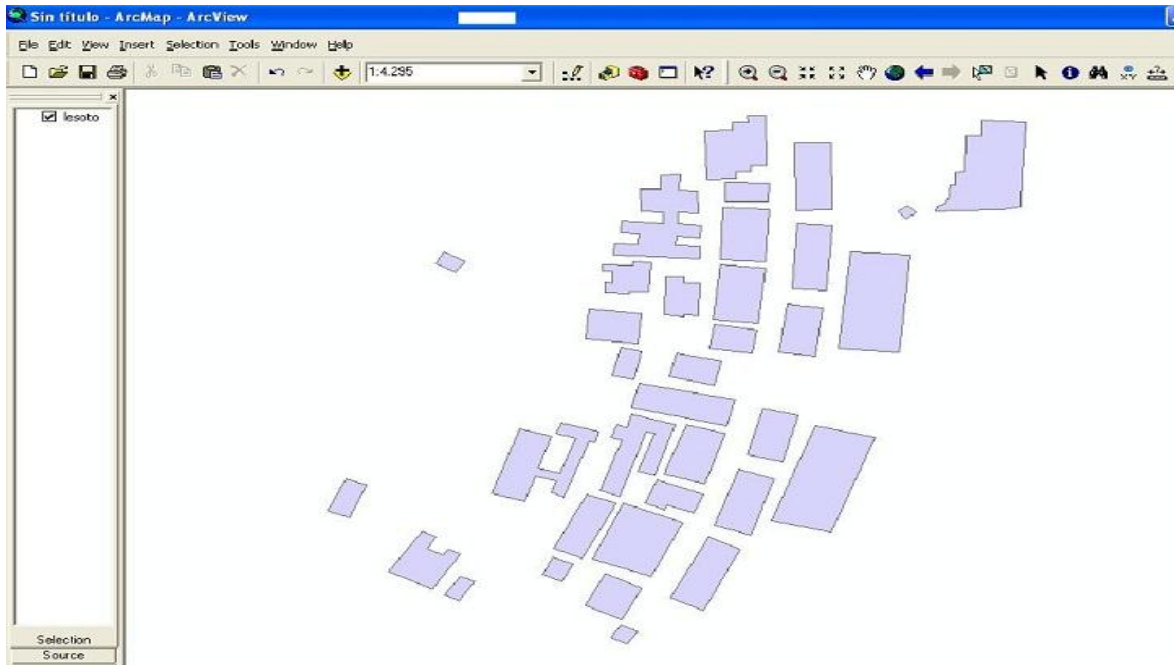


Figura 16: Imagen de shape

La siguiente figura muestra los datos representados en forma tridimensional:

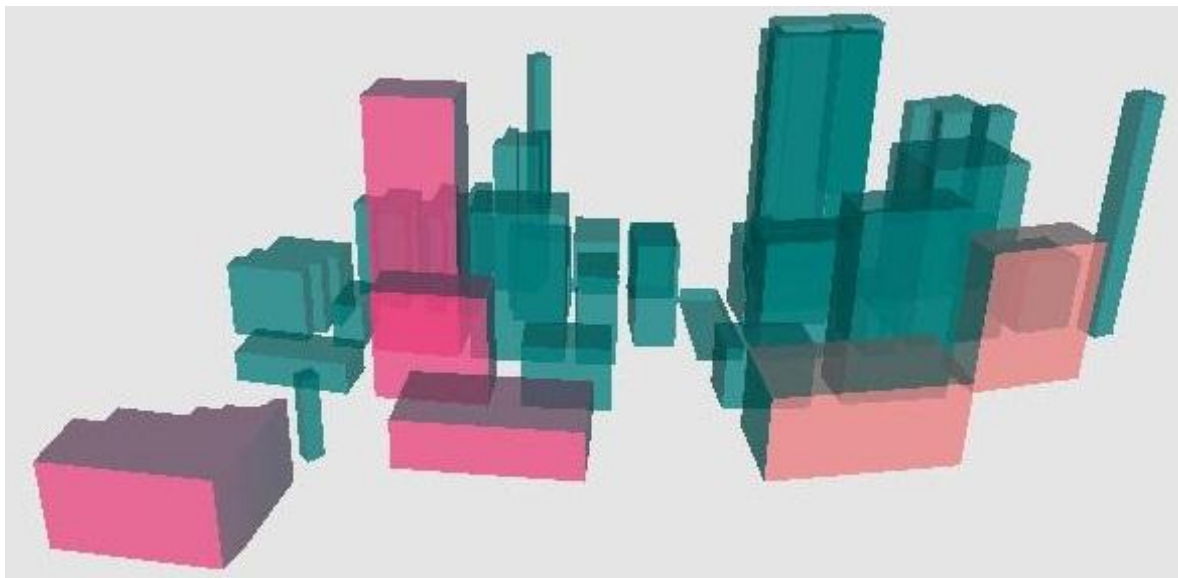


Figura 17: Imagen de los resultados

La figura muestra los datos representados en forma tridimensional con una interfaz de usuario integrada que permite la interacción con el usuario

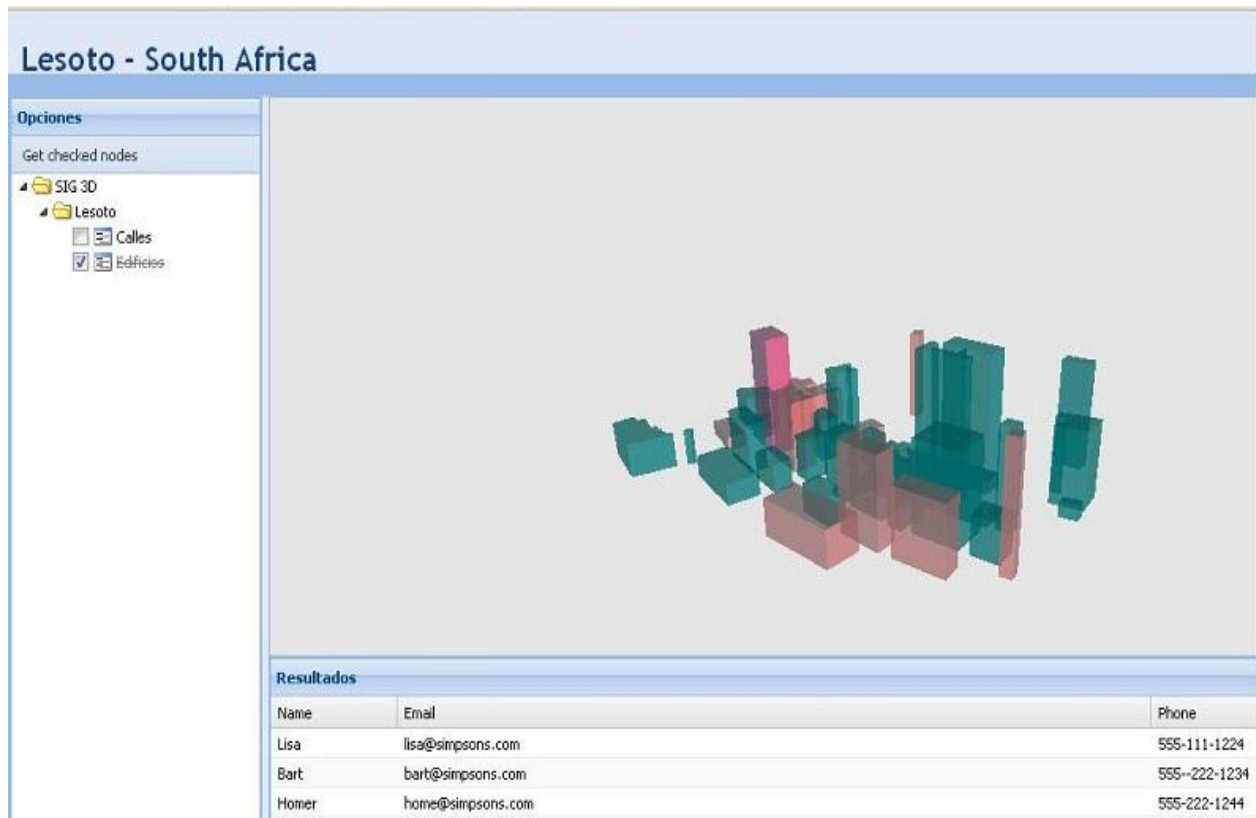


Figura 18: Propuesta con interfaz Integrada

Los modelos podrán ser vistos en detalles con la utilización del scroll del mouse y la página no se sobrecargará debido a la utilización de las técnicas antes descritas que permiten una buena eficiencia en la visualización 3D. La propuesta es multi-plataforma aumentando así la explotación de la misma. La propuesta brinda varias posibilidades con el objetivo de lograr la interacción con entornos urbanos:

- Facilita la interacción con mapas urbanos con información geográfica real.
- Permite, a partir de imágenes de satélite, construir un mapa tridimensional, desplazándose de esta manera el método tradicional.
- Se optimiza el proceso de modelado de mapas ya que se hace automáticamente.

- Permite extender el uso de la representación de datos geográficos tridimensionales en nuestro centro (UCI) especialmente en los centros productivos donde sean necesarios.

3.2 Evaluación de la propuesta

A la propuesta de solución se le realizaron pruebas de carga y estrés para demostrar que la visualización en 3D no constituye una vulnerabilidad, para automatizar las pruebas se hizo uso de la herramienta JMeter. El escenario en el que se realizaron las pruebas fue una PC con las características siguientes:

1. Hardware:

- Tipo de procesador = Pentium 4.
- Memoria del Sistema = 512 MB.
- Tarjeta de Red = Intel (R) PRO/1000 Network Connection.
- Velocidad de Red = 100,0 Mbps.

2. Software:

- Servidor Web Apache = 2.2.12.
- Max de hilos concurrentes = 100.
- Plataforma = Multiplataforma.
- Servidor de base de datos = PostgreSQL + PostGIS.
- Lenguaje = PHP /5.3.0

3. Requerimientos no funcionales.

- Velocidad mínima de transferencia de datos = 1000 kb/seg.
- Cantidad máxima de usuarios concurrentes = 50.

- Tiempo de respuesta = 5 s.

4. Interfaz de usuario

- Interfaz amigable
- Sencilla y práctica para cualquier usuario.
- Presentar información geográfica en 3D.

En la tabla se muestran una serie de datos, los cuales exponen el estado en el que se encuentra en software con respecto al módulo probado.

- Muestras: Cantidad de páginas (Hilos) que simulan la cantidad de usuarios. Que están interactuando con el sistema.
- Mediana: Tiempo promedio que han tardado en cargarse las páginas.
- Min: Tiempo mínimo que ha demorado en cargarse una página en ms.
- Max: Tiempo Máximo que ha tardado en cargarse una página en ms.
- %Error: Por ciento de error de las páginas que no se llegaron a cargar de manera satisfactoria.
- kb /seg: Velocidad de carga de las páginas.
- Rendimiento: Total del solicitudes que se cargan por unidad de tiempo.

Se realizaron dos peticiones http al modelo propuesto, no se detectó error en los 75 hilos independientes lo que demuestra que el modelo se visualizó en su totalidad. En un ambiente con las características de software y de hardware descritas con anterioridad, se obtuvo que para un total de 75 usuarios (hilos) la aplicación es estable ya que se mantuvo prestando servicios todo el tiempo sin incurrir en fallos, generando un promedio de 15,5 kb/seg de transferencia de datos, un rendimiento de 8.1 peticiones por

segundo, que permite calcular que una petición se realiza en aproximadamente 0.108 ms, tiempo que se adecúa al tiempo que se especificaba en los requisitos no funcionales.

| | #Número de polígonos | # Muestras | Mediana | Min | Max | % Error | Rendimiento | KB/seg |
|-----------------|----------------------|------------|---------|-------|------|---------|-------------|--------|
| Caso#1 | 420 | 50 | 1020 | 192 | 1598 | 0.00% | 8,5/seg | 15.9 |
| Caso#2 | 420 | 100 | 2391 | 547 | 3940 | 0.00% | 7.7/seg | 14.4 |
| Promedio | 420 | 75 | 1705.5 | 369.5 | 2769 | 0.00% | 8.1/seg | 15.15 |

Tabla 3: Resultados de las pruebas de carga y estrés

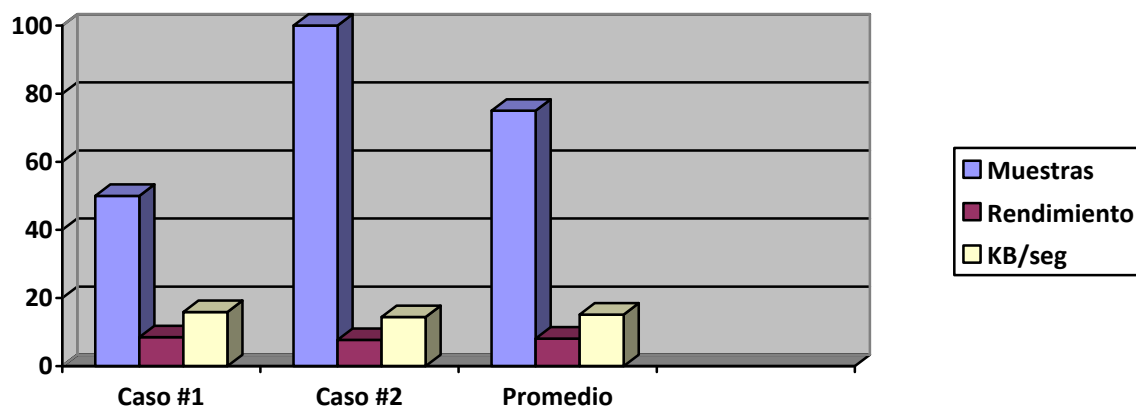


Figura 19: Gráfico de resultados de las pruebas

Conclusiones del capítulo

En el presente capítulo se realizó una evaluación de los resultados de la propuesta, se enuncian sus posibilidades, concluyendo que el desarrollo de esta propuesta es un logro muy importante, pues se ha logrado representar información geográfica tridimensional con datos reales y llevados a la web, lo cual es una de las vías de mayor acceso a la información en la actualidad. La evaluación realizada demuestra que el rendimiento de la propuesta es favorable, para el uso en máquinas de bajas prestaciones.

CONCLUSIONES

- Con el desarrollo de una propuesta de visualización de información geográfica tridimensional para plataformas web, se hizo un estudio de las tendencias actuales sobre la representación de datos geográficos tridimensionales en la web, lo que arrojó la selección y empleo del estándar extensible X3D lográndose la visualización geográfica tridimensional. Seleccionando las técnicas y los lenguajes de programación adecuados, se posibilita que la propuesta garantice el acceso, visualización e interacción con los datos representados en la escena 3D.
- El desarrollo de esta propuesta es un logro muy importante, pues se ha logrado representar información geográfica tridimensional con datos reales y llevados a la web que es una de las vías de mayor acceso a la información en la actualidad. Las pruebas realizadas demuestran que el rendimiento de la propuesta es favorable, para el uso en máquinas de bajas prestaciones.

RECOMENDACIONES

Se recomienda representar la capa de suelo y calles para que el entorno urbano sea más real y hacerlo más funcional.

REFERENCIAS BIBLIOGRÁFICAS

1. Talancón Días, R.R.B., Leandro Raúl, *Sistemas para la Visualización de imágenes y modelos 3D médicos en la WEB con fines educativos*. 2010, Universidad de las Ciencias Informáticas.: Habana. p. 14.
2. Martínez., A.J.A., *Potencialidades del GIS 3D y los Modelos Urbanos Interactivos*. 1999
3. *El Segmento de Capacitación del Gabinete de Teledetección y Sistemas de Información Geográfica*.junio 2002; Available from: <http://www.fcagr.unr.edu.ar/mdt/GTS/Zonaedu/GIS1htm.htm>.
4. Wesselingh, S., *Visualization of a TEN (Tetrahedral Irregular Network) in a web client*. 2007.5. Georg Held, A.A.-R., Siyka Zlatanova, *Web 3D GIS for Urban Environments*. 2005: p. 8-10.
6. Gabriel Dorado Martín, M.J.G.G., José Manuel Nicolás Zabala *Proceso de Captura de Datos en el Modelo Raster*. . 2011.
7. Ontiveros, F.F.G., *Exportación de datos usando OpenGis*, in *Departamento de Ingeniería en Sistemas Computacionales 2000*, Universidad de las Americas Puebla.: Mexico. p. 1-3.
8. Velásquez, S. *Creación y Edición de Shapefiles de ArcView 3.2*.
9. Eva Gómez Ballester, P.M.B., Paloma Moreda Pozo, Armando Suárez Cueto, Andrés Montoyo Guijarro, Estela Saquete Boro, *Bases de datos1*. 2010.
10. Gutiérrez, M., *El Rol de las Bases de Datos Espaciales en una Infraestructura de Datos*. 2006.
11. Zlatanova, J.S.a.S., *Visualisation and editing of 3d objects organised in a DBMS*. 2003.
12. Luna, P.L., *Sistema de Información Geográfica para la ayuda de toma de decisiones en políticas sociales.*, in *DEPARTAMENTO DE COMPUTACIÓN*. 2010, CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL: Mexico.
13. OpenGIS, *Funciones de PostGIS*. 2004.
14. Gómez, A.D.V.B., *Oracle*. 2009.

15. Rahman, A.A., *Developing Three-Dimensional (3D) Topological Model for 3D GIS.*, in *Faculty of Geoinformation Science and Engineering*. 2005: Universiti Teknologi Malaysia
16. Riaza, T.F.d.S., *GML:El lenguaje de marcado extendido (XML) para la Ingeniería Geografica.Ventajas y Aplicaciones*. 2005.
17. OpenGIS. *CityGML*. 2007; Available from:
18. Oscar Fonts, C.G., *Visualización geográfica 3D. Estándares y aplicaciones*. 2009.
19. Çöltekin, A., *An Analysis of VRML-based 3D Interfaces for Online GISs: Current Limitations and Solutions*. 2002.
20. M., K.M.G., *R-trees for organizing and visualizing 3D GIS databases* 2000. 11.
21. Jason Dykes, K.M., and David Fairbairn. *From Chernoff to Imhof and Beyond: VRML and Cartography*. in *Conference on the VRML and Web3D Technologies (VRML99)*. 1999. Paderbon , Alemania.
22. Diehl, S., *Distributed Virtual Worlds – Foundations and Implementation Techniques Using VRML, JAVA, and CORBA*. 2001.
23. Döllner, D.J., *3D-Visualisierung. Proceeding of Webmapping 2002*.
24. Sun, *The Java 3D API*. 2004.
25. Soto, C., *The advantages of php*. 2006.
26. Bradenbaugb, J., *Aplicaciones JavaScript*. 2000: p. 532.
27. *Visualization of GML data using XSLT*. 2003.
28. James Garrett, J., *Ajax: Un nuevo acercamiento a las aplicaciones WEB*. 2005.
29. web., C.d.d. *Ajax*. 2008; Available from: <http://web21.uci cu/blogs/ajax/>.

BIBLIOGRAFÍA CONSULTADA

30. Bianca Schön, D. F. L., Sean W. Morrish and Michela Bertolotto (2009). "Three-Dimensional Spatial Information Systems: State of the Art Review."
31. Georg Held, A. A.-R., Siyka Zlatanova (2005). "Web 3D GIS for Urban Environments." 8-10.
32. Mena, U. (2007) Aplicación de los sistemas de información geográfica en la ingeniería civil. Tendencias tecnológicas

ANEXOS

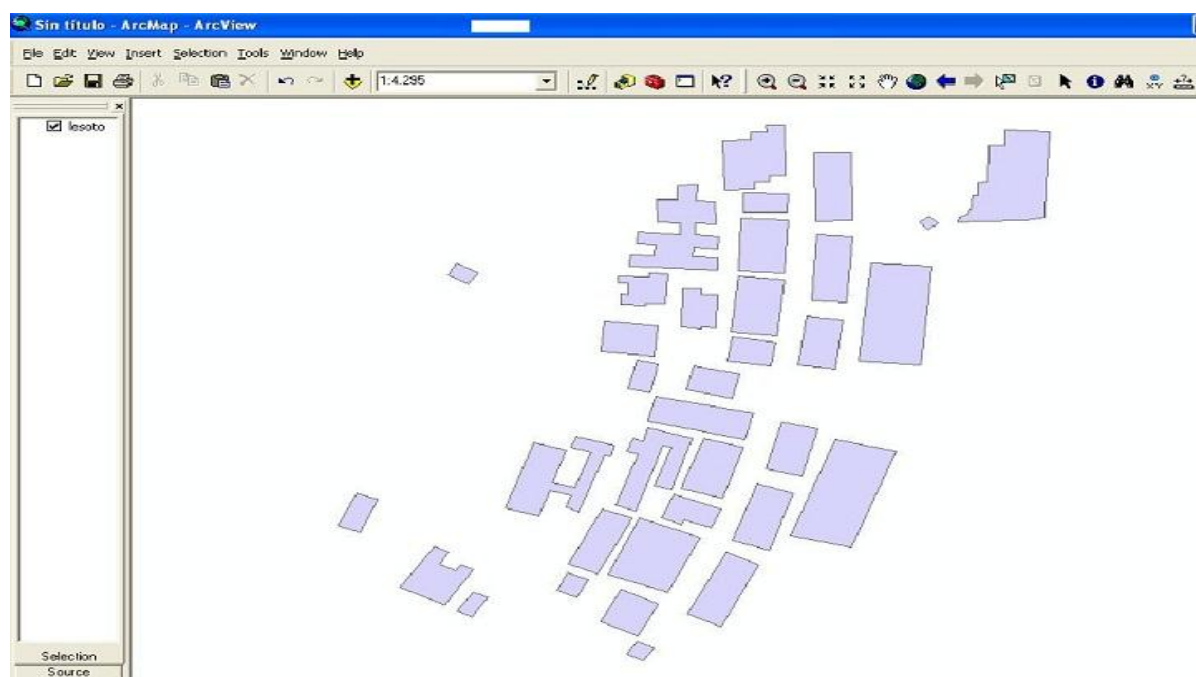


Figura 20: Shape del modelo

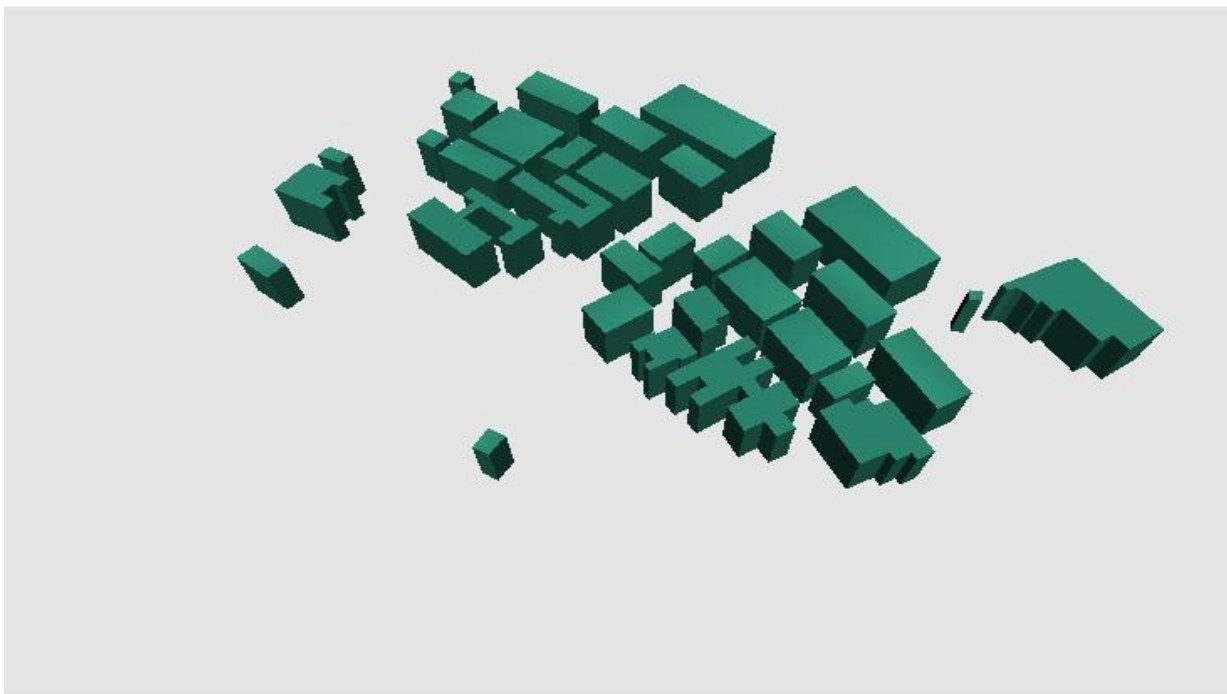


Figura 21: Modelo sin alturas específicas

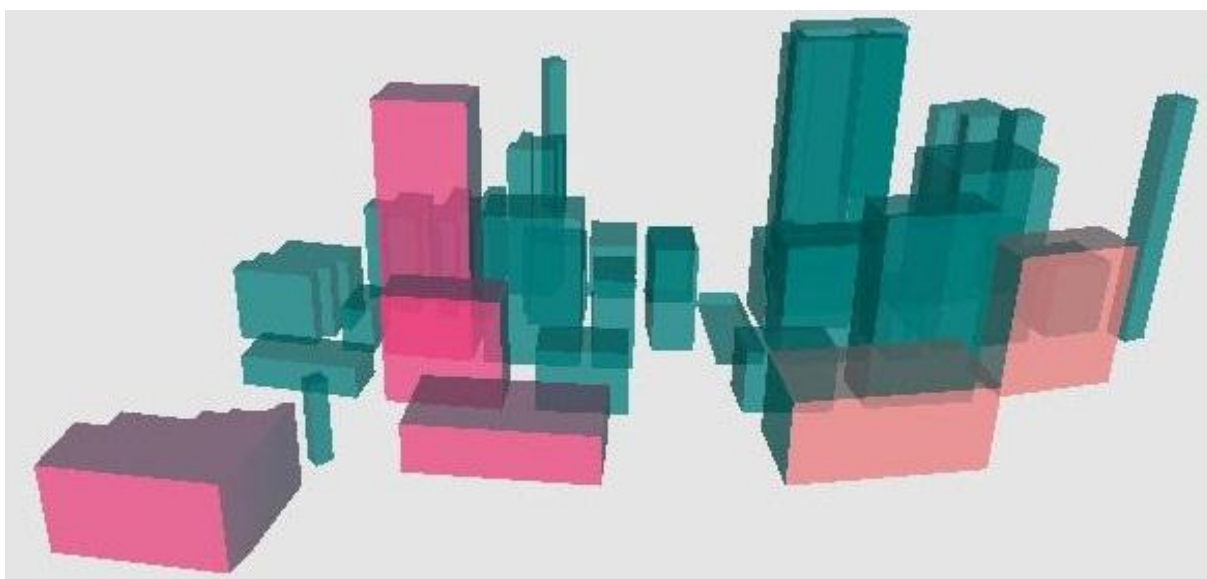


Figura 22: Modelo con alturas específicas

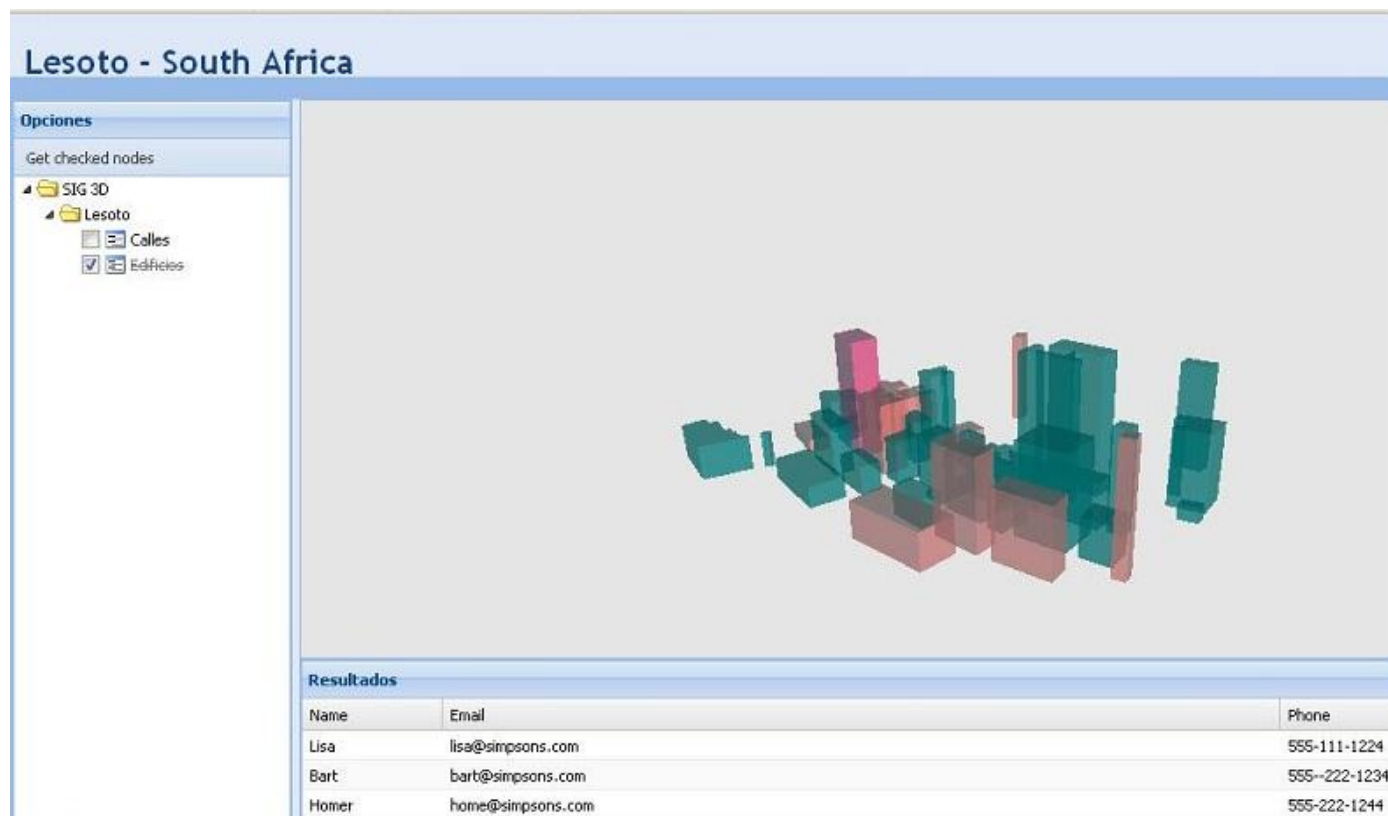


Figura 23: Modelo con interfaz

GLOSARIO DE TÉRMINOS

ACID: Es un acrónimo de Atomicity, Consistency, Isolation and Durability: Atomicidad, Consistencia, Aislamiento y Durabilidad. Conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción.

Ajax: Asynchronous JavaScript And XML no es una tecnología porque es la unión de un conjunto de tecnologías para el desarrollo WEB que se ejecutan en el cliente.

Aptana Studio: Es un IDE de desarrollo para aplicaciones de la WEB 2.0.

BD: Base de datos.

Hardware: Componentes físicos de una computadora o de una red (a diferencia de los programas o elementos lógicos que los hacen funcionar).

HTML: Lenguaje de Marcas de Hipertexto, es un lenguaje para elaborar páginas Web.

Interfaz: Es uno de los componentes más importantes de cualquier sistema computacional, funciona como el vínculo o comunicación entre el humano y la máquina.

JavaScript: Es un lenguaje interpretado que está del lado del cliente.

Java 3D: Es un API (Application Programming Interface) orientado a objetos para el lenguaje Java, que permite la confección de aplicaciones tridimensionales.

PHP: El lenguaje PHP (acrónimo de "PHP: Hypertext Preprocessor") es un lenguaje de script interpretado utilizado para la generación de páginas Web dinámicas, embebido en páginas HTML y ejecutado en el servidor.

Plug-ins: Aditamento para agregar a un equipo.

PostgreSQL: Es el más avanzado sistema de base datos relacional de código abierto con modelo de arquitectura de comunicación cliente servidor.

PosGIS: Es la versión espacial de PostgreSQL.

Realidad virtual no inmersiva: Es aquella que se crea cuando el participante explora diversos ambientes haciendo uso de los dispositivos de hardware comunes: mouse, monitor, tarjeta de sonido y bocinas.

VRML: Virtual Reality Modeling Language es un lenguaje de especificaciones, tridimensional e interactivo, orientado a la modelación y la visualización de objetos, situaciones y mundos virtuales en Red.

XML: Extensible Markup Language es un perfil de aplicación o forma restringida de (SGML).

X3D: Extensible Tridimensional es un estándar abierto XML que tiene un formato de archivo 3D que permite la creación y transmisión de datos 3D entre distintas aplicaciones.